



# ICs for Communications

DSP Oriented PBX Controller  
DOC

PEB 20560 Version 2.1

Preliminary Data Sheet 2003-08

<b>PEB 20560</b>		
<b>Revision History:</b>		<b>Current Version: Preliminary Data Sheet 2003-08</b>
Previous Version:		Preliminary Data Sheet 11.97
Page (in previous Version)	Page (in current Version)	Subjects (major changes since last revision)
		Global Interrupt Status Register (Version 2.1)
		Interrupt Mask Register for GPIO (Version 2.1)
		Trademarks changed

*Note: OCEM<sup>®</sup> and OakDSPCore<sup>®</sup> (OAK<sup>®</sup>) are registered trademarks of ParthusCeva, Inc..*

#### **Edition 2003-08**

**Published by Siemens AG,  
Bereich Halbleiter, Marketing-  
Kommunikation, Balanstraße 73,  
81541 München**

© Siemens AG 1997.  
All Rights Reserved.

#### **Attention please!**

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

#### **Packing**

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

#### **Components used in life-support devices or systems must be expressly authorized for such purpose!**

Critical components<sup>1</sup> of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems<sup>2</sup> with the express written approval of the Semiconductor Group of Siemens AG.

- 1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.
- 2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

<b>Table of Contents</b>		<b>Page</b>
<b>1</b>	<b>Overview</b> .....	1-1
1.1	DOC Features .....	1-5
1.2	Logic Symbol .....	1-7
1.3	Pin Configuration .....	1-8
1.4	Pin Description .....	1-9
1.5	Functional Block Diagram and System Integration .....	1-25
1.6	Example for System Integration .....	1-26
<b>2</b>	<b>Functional Block Description</b> .....	2-1
2.1	ELIC0 and ELIC1 .....	2-1
2.1.1	General Functions and Device Architecture .....	2-1
2.1.2	Functional Blocks .....	2-1
2.1.2.1	Watchdog Timer .....	2-1
2.1.2.2	Reset Logic .....	2-2
2.1.2.3	EPIC <sup>®</sup> -1 .....	2-3
2.1.2.3.1	PCM-Interface .....	2-3
2.1.2.3.2	Configurable Interface .....	2-3
2.1.2.3.3	Memory Structure and Switching .....	2-3
2.1.2.3.4	Pre-processed Channels, Layer-1 Support .....	2-4
2.1.2.3.5	Special Functions .....	2-5
2.1.2.4	SACCO .....	2-6
2.1.2.4.1	Block Diagram .....	2-6
2.1.2.4.2	Parallel Interface .....	2-6
2.1.2.4.3	FIFO-Structure .....	2-10
2.1.2.4.4	Protocol Support .....	2-12
2.1.2.4.5	Special Functions .....	2-27
2.1.2.4.6	Serial Interface .....	2-28
2.1.2.4.7	Serial Port Configuration .....	2-30
2.1.2.4.8	Test Mode .....	2-32
2.1.2.5	D-Channel Arbiter .....	2-32
2.1.2.5.1	Upstream Direction .....	2-33
2.1.2.5.2	Downstream Direction .....	2-38
2.1.2.5.3	Control Channel Delay .....	2-39
2.1.2.5.4	D-Channel Arbiter Co-operating with QUAT-S Circuits .....	2-40
2.2	SIDEC .....	2-41
2.3	Multiplexers .....	2-42
2.3.1	IOM <sup>®</sup> - and PCM-Ports Multiplexers .....	2-45
2.3.1.1	IOM <sup>®</sup> Multiplexer for IOM <sup>®</sup> -2 Ports (CFI Interfaces of EPIC) .....	2-45
2.3.1.2	PCM-Ports Multiplexer for PCM Highways .....	2-46
2.3.2	Multiplexers for Signaling Controllers .....	2-47
2.3.2.1	SACCO-A0 and SACCO-A1 .....	2-47
2.3.2.2	SACCO-B0 .....	2-47

<b>Table of Contents</b>		<b>Page</b>
2.3.2.3	SACCO-B1 .....	2-49
2.3.2.4	SIDEC .....	2-50
2.3.3	ELIC1-Ports Multiplexer .....	2-51
2.3.4	IOM <sup>®</sup> -Multiplexer for DSP Connection to EPICs .....	2-52
2.3.5	PCM/IOM MUX Registers Description .....	2-53
2.3.5.1	PCM/IOM MUX Mode Register (MMODE) .....	2-53
2.3.5.2	CFI Channel Select 0 Register (MCCHSEL0) .....	2-54
2.3.5.3	CFI Channel Select 1 Register (MCCHSEL1) .....	2-55
2.3.5.4	CFI Channel Select 2 Register (MCCHSEL2) .....	2-56
2.3.5.5	PCM Channel Select 0 Register (MPCHSEL0) .....	2-57
2.4	Channel Indication Logic (CHI) .....	2-58
2.4.1	CHI Configuration Register (VMODR) .....	2-58
2.4.2	CHI Control Registers (VDATR0:VDATR3) .....	2-59
2.5	FSC with Delay (FSCD) .....	2-60
2.6	Digital Signal Processor (DSP) .....	2-61
2.6.1	DSP Kernel Block Diagram .....	2-61
2.6.2	DSP Instruction Set .....	2-61
2.7	DSP Control Unit (DCU) .....	2-62
2.7.1	General .....	2-62
2.7.2	DSP Address Decoding .....	2-62
2.7.3	Control of External Memories / Registers .....	2-63
2.7.3.1	Memory Configuration Register (MEMCONFR) .....	2-68
2.7.3.2	Test Configuration Register (TESTCONFR) .....	2-69
2.7.4	Emulation Support .....	2-70
2.7.5	Interrupt Handling and Test Support .....	2-70
2.7.6	Run Time Statistics Counter and Register (STATC and STATR) ...	2-71
2.7.7	Program Write Protection Register (PASSR) .....	2-73
2.7.8	Serial (via JTAG) Emulation Configuration Register (JCONF) .....	2-73
2.7.9	Boot Support .....	2-74
2.7.9.1	Boot Sequence .....	2-74
2.7.9.2	Emulation Boot .....	2-75
2.7.9.3	Boot Procedure .....	2-75
2.7.9.4	Boot ROM .....	2-75
2.7.9.5	μP Boot .....	2-76
2.7.9.6	Mail Box Instructions Format .....	2-77
2.7.9.7	Write Program Memory Command .....	2-77
2.7.9.8	OAK Opcodes .....	2-78
2.7.9.9	Boot Configuration Register (BOOTCONF) .....	2-79
2.7.9.10	The Bootroutine .....	2-79
2.7.10	Sine Table ROM .....	2-87
2.8	PCM-DSP Interface Unit (PEDIU) .....	2-88
2.8.1	General Description .....	2-88

<b>Table of Contents</b>		<b>Page</b>
2.8.2	PEDIU Internal Registers .....	2-93
2.8.2.1	PEDIU Control Register (UCR) .....	2-93
2.8.2.2	PEDIU Status Register (USR) .....	2-97
2.8.2.3	PEDIU Input Stream Bypass Enable Register (UISBPER) .....	2-101
2.8.2.4	PEDIU Output Stream Bypass Enable Register (UOSBPER) ...	2-103
2.8.2.5	PEDIU Tri-State Register (UTSR) .....	2-104
2.8.2.6	PEDIU ROM Test Address Register (UPRTAR) and PEDIU ROM Test Data Register (UPRTDR) .....	2-106
2.8.3	PEDIU Synchronization and Clock Rates .....	2-107
2.8.3.1	PEDIU Synchronization by FSC and DCL .....	2-107
2.8.3.2	Restrictions on PEDIU Clock Rates .....	2-108
2.8.4	PEDIU Address Space .....	2-108
2.8.5	PEDIU Data Processing .....	2-109
2.8.5.1	PEDIU Serial Data Processing .....	2-109
2.8.5.2	PEDIU Parallel Data Processing .....	2-109
2.8.5.3	The Circular Buffer Address Method .....	2-111
2.8.6	a-/μ-law Conversion .....	2-114
2.9	On-chip Emulation (OCEM) .....	2-115
2.10	Mailbox .....	2-115
2.10.1	μP Mailbox .....	2-115
2.10.2	OAK Mailbox .....	2-116
2.11	μP Interface .....	2-118
2.11.1	Compatibility .....	2-118
2.11.2	Memory and I/O Organization .....	2-118
2.12	Clock Generator .....	2-119
2.12.1	Block Diagram .....	2-119
2.12.2	Types of Clock Signals .....	2-120
2.12.2.1	Input/Output Clocks .....	2-120
2.12.2.2	Clock Selection .....	2-120
2.12.3	Clocks Generator Registers Description .....	2-125
2.12.3.1	Clocks Select 0 Register (CCSEL0) .....	2-125
2.12.3.2	Clocks Select 1 Register (CCSEL1) .....	2-126
2.12.3.3	Clocks Select 2 Register (CCSEL2) .....	2-128
2.13	Interrupt Controller .....	2-129
2.13.1	MASK (IMASK0, IMASK1) .....	2-129
2.13.2	Interrupt Sources .....	2-130
2.13.3	Interrupt Priority (IPAR0, IPAR1, IPAR2) .....	2-130
2.13.4	Interrupt Cascading .....	2-132
2.13.4.1	Slave Mode .....	2-133
2.13.4.2	Daisy Chaining .....	2-134
2.13.5	Global Interrupt Status Registers (IGIS0 and IGIS1) .....	2-135
2.14	Universal Asynchronous Receiver/Transmitter (UART) .....	2-136

<b>Table of Contents</b>		<b>Page</b>
2.14.1	Registers Overview . . . . .	2-139
2.14.1.1	Line Control Register (LCR) . . . . .	2-143
2.14.1.2	Programmable Baud Rate Generator (Divisors) . . . . .	2-144
2.14.1.3	Line Status Register (LSR) . . . . .	2-145
2.14.1.4	FIFO Control Register (FCR) . . . . .	2-148
2.14.1.5	Interrupt Identification Register (IIR) . . . . .	2-149
2.14.1.6	Interrupt Enable Register (IER) . . . . .	2-151
2.14.1.7	Modem Control Register (MCR) . . . . .	2-151
2.14.1.8	Modem Status Register (MSR) . . . . .	2-153
2.14.1.9	Scratchpad Register (SCR) . . . . .	2-154
2.14.2	FIFO Interrupt Mode Operation . . . . .	2-154
2.14.3	FIFO Polled Mode Operation . . . . .	2-155
2.15	General Purpose I/O Port (GPIO) . . . . .	2-156
2.15.1	I/O Port Support Lines (Mode 0) . . . . .	2-156
2.15.2	SACCO-B0 Support Lines (Mode 1) . . . . .	2-156
2.15.3	UART Support Lines (Mode 2) . . . . .	2-156
2.15.4	Configuration Register (VCFGR) . . . . .	2-157
2.15.5	Data Register (VDATR) . . . . .	2-157
2.15.6	Version Number Register (VNR) . . . . .	2-158
2.16	Boundary Scan Support (JTAG) . . . . .	2-158
2.16.1	Boundary Scan . . . . .	2-158
2.16.2	TAP-Controller . . . . .	2-159
2.17	Reset Logic . . . . .	2-160
<b>3</b>	<b>Operational Description . . . . .</b>	<b>3-1</b>
3.1	ELIC0 and ELIC1 . . . . .	3-1
3.1.1	Interrupt Structure and Logic . . . . .	3-1
3.1.2	Clocking . . . . .	3-2
3.1.3	EPIC®-1 Operation . . . . .	3-3
3.1.3.1	PCM-Interface . . . . .	3-3
3.1.3.2	Configurable Interface . . . . .	3-5
3.1.3.3	Switching Functions . . . . .	3-6
3.1.3.4	Special Functions . . . . .	3-10
3.1.4	SACCO-A/B . . . . .	3-10
3.1.4.1	Data Transmission in Interrupt Mode . . . . .	3-10
3.1.4.2	Data Transmission in DMA-Mode . . . . .	3-12
3.1.4.3	Data Reception in Interrupt Mode . . . . .	3-13
3.1.4.4	Data Reception in DMA-Mode . . . . .	3-14
3.1.5	D-Channel Arbiter . . . . .	3-15
3.1.5.1	SACCO-A Transmission . . . . .	3-15
3.1.5.2	SACCO-A Reception . . . . .	3-16
3.1.6	Initialization Procedure . . . . .	3-17

<b>Table of Contents</b>		<b>Page</b>
3.1.6.1	Hardware Reset . . . . .	3-17
3.1.6.2	EPIC <sup>®</sup> -1 Initialization . . . . .	3-17
3.1.6.2.1	EPIC <sup>®</sup> Registers Initialization . . . . .	3-17
3.1.6.2.2	Control Memory Reset . . . . .	3-17
3.1.6.2.3	Initialization of Pre-processed Channels . . . . .	3-18
3.1.6.2.4	Initialization of the Upstream Data Memory (DM) Tri-State Field . . . . .	3-19
3.1.6.3	SACCO-Initialization . . . . .	3-20
3.1.6.4	Initialization of D-Channel Arbiter . . . . .	3-21
3.1.6.5	Activation of the PCM- and CFI-Interfaces . . . . .	3-22
3.1.6.6	Initialization Example . . . . .	3-23
3.1.6.6.1	EPIC <sup>®</sup> -1 Initialization Example . . . . .	3-24
3.1.6.6.2	SACCO-A Initialization Example . . . . .	3-26
3.1.6.6.3	D-Channel Arbiter Initialization Example . . . . .	3-26
3.1.6.6.4	PCM- and CFI-Interface Activation Example . . . . .	3-27
3.1.6.6.5	SACCO-B Initialization Example . . . . .	3-27
3.2	SIDEC . . . . .	3-28
<b>4</b>	<b>DSP Core OAK</b> . . . . .	<b>4-1</b>
4.1	Introduction . . . . .	4-1
4.1.1	General Description . . . . .	4-1
4.1.2	Architecture Highlights . . . . .	4-1
4.2	Architecture Features . . . . .	4-3
4.2.1	Features . . . . .	4-3
4.2.2	Buses . . . . .	4-4
4.2.2.1	Data Buses . . . . .	4-4
4.2.2.2	Program Buses . . . . .	4-4
4.2.3	Memory Spaces and Organization . . . . .	4-5
4.2.3.1	Program Memory . . . . .	4-5
4.2.3.2	Data Memory . . . . .	4-6
4.3	Development Tools . . . . .	4-7
4.3.1	COFF Macro Assembler . . . . .	4-7
4.3.2	Linker/Locator . . . . .	4-7
4.3.3	Object Format Converter . . . . .	4-7
4.3.4	ANSI C-Compiler . . . . .	4-7
4.3.5	Simulator . . . . .	4-8
4.3.6	Debugger . . . . .	4-8
<b>5</b>	<b>Description of Registers</b> . . . . .	<b>5-1</b>
5.1	μP Address Space Register Overview . . . . .	5-1
5.1.1	ELIC0 and ELIC1 Registers Description . . . . .	5-22
5.1.1.1	Interrupt Top Level . . . . .	5-22
5.1.1.1.1	Interrupt Status Register (ISTA) . . . . .	5-22

<b>Table of Contents</b>		<b>Page</b>
5.1.1.1.2	Mask Register (MASK) . . . . .	5-23
5.1.1.2	Watchdog Timer (in ELIC0 only) . . . . .	5-23
5.1.1.2.1	Watchdog Control Register (WTC) . . . . .	5-23
5.1.1.3	ELIC <sup>®</sup> Mode Register . . . . .	5-24
5.1.1.4	EPIC <sup>®</sup> -1 . . . . .	5-26
5.1.1.4.1	PCM-Mode Register (PMOD) . . . . .	5-26
5.1.1.4.2	Bit Number per PCM-Frame (PBNR) . . . . .	5-28
5.1.1.4.3	PCM-Offset Downstream Register (POFD) . . . . .	5-28
5.1.1.4.4	PCM-Offset Upstream Register (POFU) . . . . .	5-29
5.1.1.4.5	PCM-Clock Shift Register; within the ELIC (PCSR) . . . . .	5-29
5.1.1.4.6	PCM-Input Comparison Mismatch (PICM) . . . . .	5-30
5.1.1.4.7	Configurable Interface Mode Register 1 (CMD1) . . . . .	5-30
5.1.1.4.8	Configurable Interface Mode Register 2 (CMD2) . . . . .	5-33
5.1.1.4.9	Configurable Interface Bit Number Register (CBNR) . . . . .	5-34
5.1.1.4.10	Configurable Interface Time-Slot Adjustment Register (CTAR) . . . . .	5-35
5.1.1.4.11	Configurable Interface Bit Shift Register (CBSR) . . . . .	5-35
5.1.1.4.12	Configurable Interface Subchannel Register (CSCR) . . . . .	5-37
5.1.1.4.13	Memory Access Control Register (MACR) . . . . .	5-38
5.1.1.4.14	Memory Access Address Register (MAAR) . . . . .	5-42
5.1.1.4.15	Memory Access Data Register (MADR) . . . . .	5-43
5.1.1.4.16	Synchronous Transfer Data Register (STDA) . . . . .	5-43
5.1.1.4.17	Synchronous Transfer Data Register B (STDB) . . . . .	5-43
5.1.1.4.18	Synchronous Transfer Receive Address Register A (SARA) . . . . .	5-44
5.1.1.4.19	Synchronous Transfer Receive Address Register B (SARB) . . . . .	5-44
5.1.1.4.20	Synchronous Transfer Transmit Address Register A (SAXA) . . . . .	5-45
5.1.1.4.21	Synchronous Transfer Transmit Address Register B (SAXB) . . . . .	5-45
5.1.1.4.22	Synchronous Transfer Control Register (STCR) . . . . .	5-46
5.1.1.4.23	MF-Channel Active Indication Register (MFAIR) . . . . .	5-47
5.1.1.4.24	MF-Channel Subscriber Address Register (MFSAR) . . . . .	5-47
5.1.1.4.25	Monitor/Feature Control Channel FIFO (MFFIFO) . . . . .	5-48
5.1.1.4.26	Signaling FIFO (CIFIFO) . . . . .	5-48
5.1.1.4.27	Timer Register (TIMR) . . . . .	5-49
5.1.1.4.28	Status Register EPIC <sup>®</sup> -1 (STAR_E) . . . . .	5-49
5.1.1.4.29	Command Register EPIC <sup>®</sup> -1 (CMDR_E) . . . . .	5-50
5.1.1.4.30	Interrupt Status Register EPIC <sup>®</sup> -1 (ISTA_E) . . . . .	5-52
5.1.1.4.31	Mask Register EPIC <sup>®</sup> -1 (MASK_E) . . . . .	5-53
5.1.1.4.32	Operation Mode Register (OMDR) . . . . .	5-54
5.1.1.4.33	Version Number Status Register (VNSR) . . . . .	5-56
5.1.1.5	SACCO-A Register Description . . . . .	5-56
5.1.1.5.1	Receive FIFO (RFIFO) . . . . .	5-56
5.1.1.5.2	Transmit FIFO (XFIFO) . . . . .	5-57
5.1.1.5.3	Interrupt Status Register (ISTA_A/B) . . . . .	5-57



<b>Table of Contents</b>		<b>Page</b>
5.1.1.5.4	Mask Register (MASK_A/B) . . . . .	5-57
5.1.1.5.5	Extended Interrupt Register (EXIR_A/B) . . . . .	5-58
5.1.1.5.6	Command Register (CMDR) . . . . .	5-59
5.1.1.5.7	Mode Register (MODE) . . . . .	5-60
5.1.1.5.8	Channel Configuration Register 1 (CCR1) . . . . .	5-61
5.1.1.5.9	Channel Configuration Register 2 (CCR2) . . . . .	5-62
5.1.1.5.10	Receive Length Check Register (RLCR) . . . . .	5-62
5.1.1.5.11	Status Register (STAR) . . . . .	5-62
5.1.1.5.12	Receive Status Register (RSTA) . . . . .	5-63
5.1.1.5.13	Receive HDLC-Control Register (RHCR) . . . . .	5-65
5.1.1.5.14	Transmit Address Byte 1 (XAD1) . . . . .	5-65
5.1.1.5.15	Transmit Address Byte 2 (XAD2) . . . . .	5-65
5.1.1.5.16	Receive Address Byte Low Register 1 (RAL1) . . . . .	5-66
5.1.1.5.17	Receive Address Byte Low Register 2 (RAL2) . . . . .	5-66
5.1.1.5.18	Receive Address Byte High Register 1 (RAH1) . . . . .	5-67
5.1.1.5.19	Receive Address Byte High Register 2 (RAH2) . . . . .	5-67
5.1.1.5.20	Receive Byte Count Low (RBCL) . . . . .	5-67
5.1.1.5.21	Receive Byte Count High (RBCH) . . . . .	5-68
5.1.1.5.22	Version Status Register (VSTR) . . . . .	5-68
5.1.1.6	SACCO-B Register Description . . . . .	5-68
5.1.1.6.1	Receive FIFO (RFIFO) . . . . .	5-68
5.1.1.6.2	Transmit FIFO (XFIFO) . . . . .	5-69
5.1.1.6.3	Interrupt Status Register (ISTA_A/B) . . . . .	5-70
5.1.1.6.4	Mask Register (MASK_A/B) . . . . .	5-71
5.1.1.6.5	Extended Interrupt Register (EXIR_A/B) . . . . .	5-71
5.1.1.6.6	Command Register (CMDR) . . . . .	5-72
5.1.1.6.7	Mode Register (MODE) . . . . .	5-74
5.1.1.6.8	Channel Configuration Register 1 (CCR1) . . . . .	5-75
5.1.1.6.9	Channel Configuration Register 2 (CCR2) . . . . .	5-77
5.1.1.6.10	Receive Length Check Register (RLCR) . . . . .	5-78
5.1.1.6.11	Status Register (STAR) . . . . .	5-78
5.1.1.6.12	Receive Status Register (RSTA) . . . . .	5-79
5.1.1.6.13	Receive HDLC-Control Register (RHCR) . . . . .	5-81
5.1.1.6.14	Transmit Address Byte 1 (XAD1) . . . . .	5-81
5.1.1.6.15	Transmit Address Byte 2 (XAD2) . . . . .	5-82
5.1.1.6.16	Receive Address Byte Low Register 1 (RAL1) . . . . .	5-82
5.1.1.6.17	Receive Address Byte Low Register 2 (RAL2) . . . . .	5-83
5.1.1.6.18	Receive Address Byte High Register 1 (RAH1) . . . . .	5-83
5.1.1.6.19	Receive Address Byte High Register 2 (RAH2) . . . . .	5-83
5.1.1.6.20	Receive Byte Count Low (RBCL) . . . . .	5-84
5.1.1.6.21	Receive Byte Count High (RBCH) . . . . .	5-84
5.1.1.6.22	Transmit Byte Count Low (XBCL) . . . . .	5-84

<b>Table of Contents</b>		<b>Page</b>
5.1.1.6.23	Transmit Byte Count High (XBCH) . . . . .	5-85
5.1.1.6.24	Time-Slot Assignment Register Transmit (TSAX) . . . . .	5-85
5.1.1.6.25	Time-Slot Assignment Register Receive (TSAR) . . . . .	5-86
5.1.1.6.26	Transmit Channel Capacity Register (XCCR) . . . . .	5-86
5.1.1.6.27	Receive Channel Capacity Register (RCCR) . . . . .	5-86
5.1.1.6.28	Version Status Register (VSTR) . . . . .	5-87
5.1.1.7	D-Channel Arbiter Registers . . . . .	5-87
5.1.1.7.1	Arbiter Mode Register (AMO) . . . . .	5-87
5.1.1.7.2	Arbiter State Register (ASTATE) . . . . .	5-88
5.1.1.7.3	Suspend Counter Value Register (SCV) . . . . .	5-88
5.1.1.7.4	D-Channel Enable Register IOM <sup>®</sup> -Port 0 (DCE0) . . . . .	5-89
5.1.1.7.5	D-Channel Enable Register IOM <sup>®</sup> -Port 1 (DCE1) . . . . .	5-89
5.1.1.7.6	D-Channel Enable Register IOM <sup>®</sup> -Port 2 (DCE2) . . . . .	5-89
5.1.1.7.7	D-Channel Enable Register IOM <sup>®</sup> -Port 3 (DCE3) . . . . .	5-89
5.1.1.7.8	Transmit D-Channel Address Register (XDC) . . . . .	5-90
5.1.1.7.9	Broadcast Group IOM <sup>®</sup> -Port 0 (BCG0) . . . . .	5-90
5.1.1.7.10	Broadcast Group IOM <sup>®</sup> -Port 1 (BCG1) . . . . .	5-90
5.1.1.7.11	Broadcast Group IOM <sup>®</sup> -Port 2 (BCG2) . . . . .	5-90
5.1.1.7.12	Broadcast Group IOM <sup>®</sup> -Port 3 (BCG3) . . . . .	5-91
5.1.2	SIDEC Register Description . . . . .	5-91
5.1.2.1	Receive FIFO (RFIFO) . . . . .	5-91
5.1.2.2	Transmit FIFO (XFIFO) . . . . .	5-91
5.1.2.3	Interrupt Status Register (ISTA_A/B) . . . . .	5-92
5.1.2.4	Mask Register (MASK_A/B) . . . . .	5-92
5.1.2.5	Extended Interrupt Register (EXIR_A/B) . . . . .	5-93
5.1.2.6	Command Register (CMDR) . . . . .	5-94
5.1.2.7	Mode Register (MODE) . . . . .	5-96
5.1.2.8	Channel Configuration Register 1 (CCR1) . . . . .	5-97
5.1.2.9	Channel Configuration Register 2 (CCR2) . . . . .	5-98
5.1.2.10	Receive Length Check Register (RLCR) . . . . .	5-99
5.1.2.11	Status Register (STAR) . . . . .	5-99
5.1.2.12	Receive Status Register (RSTA) . . . . .	5-100
5.1.2.13	Receive HDLC-Control Register (RHCR) . . . . .	5-102
5.1.2.14	Transmit Address Byte 1 (XAD1) . . . . .	5-102
5.1.2.15	Transmit Address Byte 2 (XAD2) . . . . .	5-103
5.1.2.16	Receive Address Byte Low Register 1 (RAL1) . . . . .	5-103
5.1.2.17	Receive Address Byte Low Register 2 (RAL2) . . . . .	5-104
5.1.2.18	Receive Address Byte High Register 1 (RAH1) . . . . .	5-104
5.1.2.19	Receive Address Byte High Register 2 (RAH2) . . . . .	5-104
5.1.2.20	Receive Byte Count Low (RBCL) . . . . .	5-105
5.1.2.21	Receive Byte Count High (RBCH) . . . . .	5-105
5.1.2.22	Time-Slot Assignment Register Transmit (TSAX) . . . . .	5-105

<b>Table of Contents</b>		<b>Page</b>
5.1.2.23	Time-Slot Assignment Register Receive (TSAR) .....	5-106
5.1.2.24	Transmit Channel Capacity Register (XCCR) .....	5-106
5.1.2.25	Receive Channel Capacity Register (RCCR) .....	5-106
5.1.2.26	Version Status Register (VSTR) .....	5-107
<b>6</b>	<b>Applications</b> .....	<b>6-1</b>
6.1	DOC in a Small PBX .....	6-1
6.1.1	Small PBX with 2.048 Mbit/s Data Rate .....	6-1
6.1.2	Small PBX with 4.096 Mbit/s Data Rate .....	6-3
6.2	DOC on Line Card .....	6-4
6.2.1	Line Card with 2.048 and 4.096 Mbit/s Data Rates .....	6-4
6.2.2	Line Card with 4.096 and 8.192 Mbit/s Data Rates .....	6-5
6.3	Clock Generation .....	6-6
6.3.1	PBX with one DOC .....	6-6
6.3.2	PBX with Multiple DOCs .....	6-7
6.4	Signaling with SIDEK .....	6-8
6.5	Local Network Controller (SACCO-B1 as LNC) .....	6-9
6.6	UART Applications .....	6-9
6.7	IOM <sup>®</sup> -2 Channel Indication Signal (CHI) .....	6-9
6.8	Use of FSCD .....	6-9
6.9	Watch-Dog Activation .....	6-9
6.10	Tone and Voice Processing .....	6-9
6.11	DSP Frequency Recommendation .....	6-11
<b>7</b>	<b>Electrical Characteristics</b> .....	<b>7-1</b>
7.1	Absolute Maximum Ratings .....	7-1
7.2	Operating Range .....	7-1
7.3	DC Characteristics .....	7-2
7.4	Capacitances .....	7-3
7.5	Strap Pins Pull-up Resistors Specification .....	7-3
7.6	40 MHz External Crystal .....	7-3
7.7	General Recommendations and Prohibitions .....	7-4
7.8	AC Characteristics .....	7-4
7.9	Microprocessor Interface Timing .....	7-5
7.9.1	PCM Interface Timing .....	7-13
7.9.2	IOM <sup>®</sup> -2 Interface Timing .....	7-20
7.9.3	Serial Interface Timing .....	7-28
7.9.4	Reset Timing .....	7-33
7.9.5	Boundary Scan Timing .....	7-34
7.9.6	DSP External Memory Interface Timing .....	7-35
7.9.7	Clocks Signals Timing (additional to the IOM <sup>®</sup> -2 and PCM clocks) ..	7-45

<b>Table of Contents</b>		<b>Page</b>
<b>8</b>	<b>Ordering Information and Mechanical Data</b> .....	8-1
8.1	Package Outlines .....	8-1
<b>9</b>	<b>Appendix</b> .....	9-1
9.1	IOM <sup>®</sup> -2 Interface .....	9-1
9.1.1	Signals / Channels .....	9-1
9.1.2	Monitor and C/I Handlers .....	9-2
9.1.3	IOM <sup>®</sup> -2 Interface Timing .....	9-2
9.2	Working Sheets for Multiplexers Programming .....	9-4
9.3	Working Sheets .....	9-4
9.3.1	Register Summary for EPIC <sup>®</sup> Initialization .....	9-5
9.3.2	Switching of PCM Time-Slots to the CFI Interface (Data Downstream) .....	9-9
9.3.3	Switching of CFI Time-Slots to the PCM Interface (Data Upstream)	9-10
9.3.4	Preparing EPICs C/I Channels .....	9-11
9.3.5	Receiving and Transmitting IOM <sup>®</sup> -2 C/I-Codes .....	9-12
9.3.6	DOC Port and Signaling Multiplexers .....	9-13
9.3.7	DOC Clocking Multiplexers .....	9-14
9.4	Development Tools and Software Support .....	9-15
9.4.1	Software .....	9-15
9.4.2	Macro Assembler .....	9-15
9.4.3	Linker/Locator .....	9-15
9.4.4	C Compiler .....	9-15
9.4.5	Object Format Converter .....	9-15
9.4.6	Simulator .....	9-15
9.5	OAK Development / Evaluation Board .....	9-16
9.6	DOC Configurator .....	9-18
<b>10</b>	<b>Acronym</b> .....	10-1

IOM<sup>®</sup>, IOM<sup>®</sup>-1, IOM<sup>®</sup>-2, SICOFI<sup>®</sup>, SICOFI<sup>®</sup>-2, SICOFI<sup>®</sup>-4, SICOFI<sup>®</sup>-4 $\mu$ C, SLICOFI<sup>®</sup>, ARCOFI<sup>®</sup>, ARCOFI<sup>®</sup>-BA, ARCOFI<sup>®</sup>-SP, EPIC<sup>®</sup>-1, EPIC<sup>®</sup>-S, ELIC<sup>®</sup>, IPAT<sup>®</sup>-2, ITAC<sup>®</sup>, ISAC<sup>®</sup>-S, ISAC<sup>®</sup>-S TE, ISAC<sup>®</sup>-P, ISAC<sup>®</sup>-P TE, IDEC<sup>®</sup>, SICAT<sup>®</sup>, OCTAT<sup>®</sup>-P, QUAT<sup>®</sup>-S are registered trademarks of Siemens AG.

<b>List of Figures</b>		<b>Page</b>
Figure 1-1	Functional Blocks of a PBX .....	1-1
Figure 1-2	Application Example PBX for 32 Subscribers with 4 Trunk Lines using one DOC .....	1-2
Figure 1-3	Principle Block Diagram of the DOC .....	1-4
Figure 1-4	DOC Logic Symbol. ....	1-7
Figure 1-5	DOC Pin Configuration (P-MQFP-160 Package) .....	1-8
Figure 1-6	Functional Block Diagram and System Integration .....	1-25
Figure 1-7	Example for a PBX with one DOC .....	1-26
Figure 2-1	EPIC <sup>®</sup> -1 Memory Structure .....	2-5
Figure 2-2	Timing Diagram for DMA-Transfers (fast) Transmit (n < 32, remainder of a long message or n = k × 32) .....	2-7
Figure 2-3	Timing Diagram for DMA-Transfers (slow) Transmit (n < 32, remainder of a long message or n = k × 32) .....	2-8
Figure 2-4	Timing Diagram for DMA-Transfer (fast) Receive (n = k × 32) .....	2-8
Figure 2-5	Timing Diagram for DMA-Transfers (slow) Receive (n = k × 32) .....	2-8
Figure 2-6	Timing Diagram for DMA-Transfers (slow or fast) Receive (n = 4, 8 or 16) .....	2-9
Figure 2-7	DMA-Transfers with Pulsed $\overline{DACK}$ (read or write) .....	2-9
Figure 2-8	Frame Storage in RFIFO (single frame / multiple frames) .....	2-10
Figure 2-9	XFIFO Loading, Continuous Frame Transmission Disabled (CFT = 0)	2-11
Figure 2-10	XFIFO Loading, Continuous Frame Transmission Enabled (CFT = 1)	2-12
Figure 2-11	Support of the HDLC Protocol by the SACCO .....	2-13
Figure 2-12	Polling of up to 64 Bytes Direct Data .....	2-19
Figure 2-13	Polling More than 64 Bytes of Direct Data (e.g. 96 bytes) .....	2-19
Figure 2-14	Re-Transmission of a Frame .....	2-20
Figure 2-15	Re-Transmission of a Frame with Auto-Repeat Function .....	2-21
Figure 2-16	Polling of Prepared Data .....	2-22
Figure 2-17	Receive Data Flow .....	2-26
Figure 2-18	Location of Time-Slots .....	2-29
Figure 2-19	D-Channel Arbiter. ....	2-33
Figure 2-20	Arbiter State Machine (ASM) .....	2-36
Figure 2-21	.....	2-40
Figure 2-22	SIDEC Block Diagram .....	2-41
Figure 2-23	SIDEC Signals .....	2-42
Figure 2-24	Principle Block Diagram of IOM and PCM Multiplexers; Mode 0-0-0-0-1. ....	2-43
Figure 2-25	Modes of HDLC Connection to IOM <sup>®</sup> -2 Interfaces within the Signaling MUX .....	2-44
Figure 2-26	IOM <sup>®</sup> Ports Multiplexer in Mode 1 and ELIC <sup>®</sup> -1-Ports Multiplexer in Mode 0 .....	2-45
Figure 2-27	PCM-Ports Multiplexer in Mode 1. ....	2-46
Figure 2-28	SACCO-B0 Multiplexers. ....	2-48

<b>List of Figures</b>	<b>Page</b>
Figure 2-29 QUAT-S with SACCO-B for Single-Channel LT-T Application . . . . .	2-49
Figure 2-30 SACCO-B1 Multiplexers. . . . .	2-50
Figure 2-31 PEDIU Connection to the ELICs . . . . .	2-52
Figure 2-32 CHI Signal . . . . .	2-58
Figure 2-33 FSCD Behavior . . . . .	2-60
Figure 2-34 External Data/Program Read Access . . . . .	2-64
Figure 2-35 External Data Write Access . . . . .	2-65
Figure 2-36 External Program Write Access Due to MOVD . . . . .	2-66
Figure 2-37 External Boot ROM Read. . . . .	2-67
Figure 2-38 . . . . .	2-71
Figure 2-39 Example: Flow of B-Channels between ELIC1 and PEDIU . . . . .	2-89
Figure 2-40 Accesses to the PEDIU RAM (circular buffer) at two Consecutive Frames . . . . .	2-90
Figure 2-41 Block Diagram of the PCM-DSP Interface Unit (PEDIU) . . . . .	2-92
Figure 2-42 Block Structure of Circular Buffer (PEDIU RAM) in Different PEDIU Work Modes. . . . .	2-96
Figure 2-43 Connection between CB bit and Accesses to the Circular Buffer Blocks in PEDIU work Mode 0, 1 or 2 . . . . .	2-99
Figure 2-44 The Connection between CB bit and Accesses to the Circular Buffer Blocks in PEDIU work Mode 3 or 4 . . . . .	2-100
Figure 2-45 DOC Clock Generator . . . . .	2-119
Figure 2-46 PFS Signal Selection . . . . .	2-121
Figure 2-47 PDC Generation. . . . .	2-122
Figure 2-48 Priority Unit - Block Diagram . . . . .	2-131
Figure 2-49 Interrupt Cascading (Slave Mode) in Siemens/Intel Bus Mode . . . . .	2-133
Figure 2-50 Interrupt Cascading (Daisy Chaining) in Siemens/Intel Bus Mode . . . . .	2-134
Figure 2-51 UART Block Diagram . . . . .	2-138
Figure 3-1 ELIC <sup>®</sup> Interrupt Structure . . . . .	3-1
Figure 3-2 Switching Paths Inside the EPIC <sup>®</sup> -1 . . . . .	3-6
Figure 3-3 Pre-Processed Channel Codes . . . . .	3-9
Figure 3-4 Interrupt Driven Transmission Sequence (Flow Diagram) . . . . .	3-11
Figure 3-5 Interrupt Driven Transmission Sequence Example . . . . .	3-12
Figure 3-6 DMA Driven Transmission Example . . . . .	3-12
Figure 3-7 Interrupt Driven Reception Example . . . . .	3-13
Figure 3-8 DMA-Driven Reception Example . . . . .	3-14
Figure 3-9 DOC/ELIC <sup>®</sup> Interfaces for Initialization Example . . . . .	3-23
Figure 4-1 Program Memory Map . . . . .	4-5
Figure 4-2 Data Memory Map . . . . .	4-6
Figure 5-1 Timing Relation Between Internal and External Clock. . . . .	5-25
Figure 5-2 Position of the FSC-Signal for FC-Modes 3 and 6. . . . .	5-33
Figure 5-3 Position of the FSC-Signal for FC-Mode 6. . . . .	5-33



<b>List of Figures</b>	<b>Page</b>
Figure 5-4 Internal FSC Shift to enable a Synchronization with the Rising Edge of DCL . . . . .	5-36
Figure 5-5 Use of CTS Signal in SIDEC . . . . .	5-99
Figure 6-1 DOC in a Small PBX with 2.048 Mbit/s Data Rate . . . . .	6-2
Figure 6-2 DOC in a Small PBX with 4.096 Mbit/s Data Rate . . . . .	6-3
Figure 6-3 DOC on Line Card with 2.048 and 4.096 Mbit/s Data Rates . . . . .	6-4
Figure 6-4 DOC on Line Card with 4.096 and 8.192 Mbit/s Data Rates . . . . .	6-5
Figure 6-5 Clock Generation in a PBX with one DOC . . . . .	6-6
Figure 6-6 Clock Synchronization in a PBX with Multiple DOCs . . . . .	6-7
Figure 6-7 QUAT-S in LT-T Mode with SIDEC for Four-Channel Trunk Applications . . . . .	6-8
Figure 7-1 . . . . .	7-4
Figure 7-2 I/O-Wave Form for AC-test. . . . .	7-4
Figure 7-3 Address Timing . . . . .	7-5
Figure 7-4 Data Timing . . . . .	7-6
Figure 7-5 Siemens/Intel Interrupt Timing (Slave mode) . . . . .	7-8
Figure 7-6 Siemens/Intel Interrupt Timing (Daisy chaining) . . . . .	7-9
Figure 7-7 Motorola Interrupt Timing (Slave mode) . . . . .	7-11
Figure 7-8 Motorola Interrupt Timing (Daisy chaining) . . . . .	7-12
Figure 7-9 Interrupt inactivation from WR, RD. . . . .	7-13
Figure 7-10 PDC and PFS Timing In Master Mode (PDC & PFS are outputs) . . . . .	7-15
Figure 7-11 PCM-Interface Timing in Master Mode. . . . .	7-17
Figure 7-12 PCM-Interface Timing in Slave Mode. . . . .	7-19
Figure 7-13 IOM <sup>®</sup> -2 Interface Clocks Timing When FSC and DCL are Driven by ELIC0 and the DOC is in Slave Mode . . . . .	7-21
Figure 7-14 IOM <sup>®</sup> -2 Interface Clocks Timing When FSC and DCL are Driven directly by PDC4/8 and PFS, and the DOC is in Slave Mode . . . . .	7-22
Figure 7-15 IOM <sup>®</sup> -2 Interface Clocks Timing when FSC and DCL are Driven Directly by PDC4/8 and PFS, and the DOC is in Master Mode (PDC and PFS are generated by internal clocks generator) . . . . .	7-24
Figure 7-16 IOM <sup>®</sup> -2 Interface Timing . . . . .	7-26
Figure 7-17 FSCD timing . . . . .	7-27
Figure 7-18 Channel Indication (CHI) Timing . . . . .	7-27
Figure 7-19 DRDY Timing . . . . .	7-28
Figure 7-20 Serial Interface Timing . . . . .	7-31
Figure 7-21 Serial Interface Strobe Timing (clock mode 1) . . . . .	7-32
Figure 7-22 Serial Interface Synchronization Timing (clock mode 2) . . . . .	7-33
Figure 7-23 $\overline{\text{DRESET}}$ and RESIN timing . . . . .	7-34
Figure 7-24 Boundary Scan Timing . . . . .	7-35
Figure 7-25 Program Read Access Timing . . . . .	7-36
Figure 7-26 External RAM Data Read Access Timing Diagram . . . . .	7-39
Figure 7-27 Emulation Mail-Box Read Access Timing Diagram . . . . .	7-40

<b>List of Figures</b>	<b>Page</b>
Figure 7-28 Boot ROM Read Access Timing Diagram . . . . .	7-41
Figure 7-29 External Data Write Access . . . . .	7-43
Figure 7-30 Emulation Mail-Box Write Access . . . . .	7-44
Figure 7-31 External Program Write Access Due to MOVD . . . . .	7-45
Figure 7-32 CLK61 (input) Timing . . . . .	7-46
Figure 7-33 CLK16 (input) Timing . . . . .	7-46
Figure 7-34 REFCLK Timing . . . . .	7-47
Figure 7-35 XCLK (input) Timing . . . . .	7-47
Figure 7-36 CLK30 (output) Timing . . . . .	7-48
Figure 7-37 CLK15 (output) Timing . . . . .	7-48
Figure 7-38 CLK7 (output) Timing . . . . .	7-49
Figure 8-1 Package Outline . . . . .	8-1
Figure 9-1 IOM <sup>®</sup> -2 Interface with 4-bit C/I Channel . . . . .	9-1
Figure 9-2 IOM <sup>®</sup> -2 Interface Timing with Single Data Rate DCL . . . . .	9-2
Figure 9-3 Timing of the IOM <sup>®</sup> -2 Interface with Double Data Rate DCL . . . . .	9-3
Figure 9-4 a EPIC <sup>®</sup> Initialization Register Summary (working sheet) . . . . .	9-5
Figure 9-4 b Switching of PCM Time-Slots to the CFI Interface (working sheet) . . . . .	9-9
Figure 9-4 c Switching of CFI Time-Slots to the PCM Interface (working sheet) . . . . .	9-10
Figure 9-4 d Preparing EPIC <sup>®</sup> s C/I Channels (working sheet) . . . . .	9-11
Figure 9-5 Receiving and Transmitting IOM <sup>®</sup> -2 C/I-Codes (working sheet) . . . . .	9-12
Figure 9-6 . . . . .	9-13
Figure 9-7 . . . . .	9-14
Figure 9-8 DOC Evaluation Board - Block Diagram . . . . .	9-16
Figure 9-9 DOC Evaluation System . . . . .	9-17
Figure 9-10 Example for SIDECS and SACCOs Assignment . . . . .	9-18
Figure 9-11 Example for Selection of ELIC Clocks . . . . .	9-19



<b>List of Tables</b>		<b>Page</b>
Table 1-1	IOM <sup>®</sup> -2 Interface .....	1-9
Table 1-2	PCM Interface .....	1-12
Table 1-3	Communication and Signaling Interfaces.....	1-13
Table 1-4	DSP External Memory Interface.....	1-16
Table 1-5	Clock Signals (additional to the IOM <sup>®</sup> -2, PCM and SCC clocks)....	1-18
Table 1-6	μP Interface .....	1-18
Table 1-7	Input / Output Port .....	1-20
Table 1-8	Power Supply .....	1-23
Table 1-9	Test and Emulation Interfaces .....	1-23
Table 2-1	Watchdog Timer Programming .....	2-1
Table 2-2	Reset Activities.....	2-2
Table 2-3	Behavior of the Reset Logic in the Case of Voltage Drop .....	2-3
Table 2-4	Address Recognition .....	2-14
Table 2-5	Auto-Mode Data Types .....	2-15
Table 2-6	HDLC-Control Field in Auto-Mode .....	2-15
Table 2-7	Auto-Mode Command Byte Interpretation .....	2-16
Table 2-8	Auto-Mode Response Generation .....	2-17
Table 2-9	.....	2-23
Table 2-10	Control Channel Implementation .....	2-37
Table 2-11	.....	2-38
Table 2-12	Control Channel Delay Examples .....	2-39
Table 2-13	.....	2-47
Table 2-14	.....	2-51
Table 2-15	DSP Program Address Space .....	2-62
Table 2-16	DSP Data Address Space .....	2-63
Table 2-17	Interrupt Map .....	2-70
Table 2-18	Boot Mode .....	2-74
Table 2-19	.....	2-75
Table 2-20	EPROM/ROM Data Structure.....	2-76
Table 2-21	Work Modes Specifications of the PEDIU .....	2-94
Table 2-22	Address Spaces of Circular Buffer Blocks as a Function of the PEDIU Work Mode.....	2-95
Table 2-23	Specification of the Time-Slot Quadruplet Controlled by Each Bit in UISB-PER, in the Different Work Modes of the PEDIU .....	2-101
Table 2-24	Specification of the Time-Slot Quadruplet that Controlled by Each Bit in UOSBPER, in the Different Work Modes of the PEDIU ..	2-104
Table 2-25	Specification of the Time-Slot Quadruplet Controlled by Each Bit in UOSB-PER, in the Different Work Modes of the PEDIU .....	2-105
Table 2-26	PEDIU Registers Addresses in the DSP Address Space.....	2-108
Table 2-27	Register Contents.....	2-117
Table 2-28	Interrupt Sources .....	2-130
Table 2-29	.....	2-133

<b>List of Tables</b>	<b>Page</b>
Table 2-30 .....	2-135
Table 2-31 Summary of Registers 1.....	2-139
Table 2-32 Summary of Registers 2.....	2-140
Table 2-33 Register Reset Values .....	2-141
Table 2-34 UART Registers and Addresses .....	2-142
Table 2-35 Baud Rates Using 61.44 MHz Crystal .....	2-145
Table 2-36 IIR Register .....	2-149
Table 2-37 Boundary Scan Cell Types.....	2-158
Table 2-38 Instruction Code of 4 Bit TAP Controller .....	2-159
Table 2-39 Data Path of 4 Bit TAP Controller .....	2-159
Table 3-1 .....	3-14
Table 3-2 Pre-processed Channel Options at the CFI .....	3-18
Table 3-3 Mode Dependent Register Set-up .....	3-20
Table 3-4 Feature Dependent Register Set-up .....	3-21
Table 3-5 Mode Dependent Register Set-up .....	3-28
Table 3-6 Feature Dependent Register Set-up .....	3-28
Table 5-1 ELIC0 SACCO-A .....	5-1
Table 5-2 ELIC0 SACCO-B .....	5-2
Table 5-3 ELIC0-EPIC .....	5-3
Table 5-4 ELIC0-MODE REGISTER .....	5-5
Table 5-5 ELIC0-WATCH-DOG TIMER .....	5-5
Table 5-6 ELIC0-INTERRUPT TOP LEVEL.....	5-5
Table 5-7 ELIC0-ARBITER .....	5-5
Table 5-8 ELIC1 SACCO-A .....	5-6
Table 5-9 ELIC1 SACCO-B .....	5-7
Table 5-10 ELIC1-EPIC .....	5-8
Table 5-11 ELIC1-MODE REGISTER .....	5-10
Table 5-12 ELIC1-INTERRUPT TOP LEVEL.....	5-10
Table 5-13 ELIC1-ARBITER .....	5-10
Table 5-14 SIDEC0 .....	5-11
Table 5-15 SIDEC1 .....	5-12
Table 5-16 SIDEC2 .....	5-13
Table 5-17 SIDEC3 .....	5-14
Table 5-18 ICU.....	5-15
Table 5-19 GPIO .....	5-15
Table 5-20 CHI.....	5-15
Table 5-21 OAK MAIL BOX .....	5-16
Table 5-22 CLOCKS .....	5-17
Table 5-23 IOM/PCM MUX .....	5-18
Table 5-24 UART.....	5-18
Table 5-25 PEDIU .....	5-19
Table 5-26 DCU .....	5-19

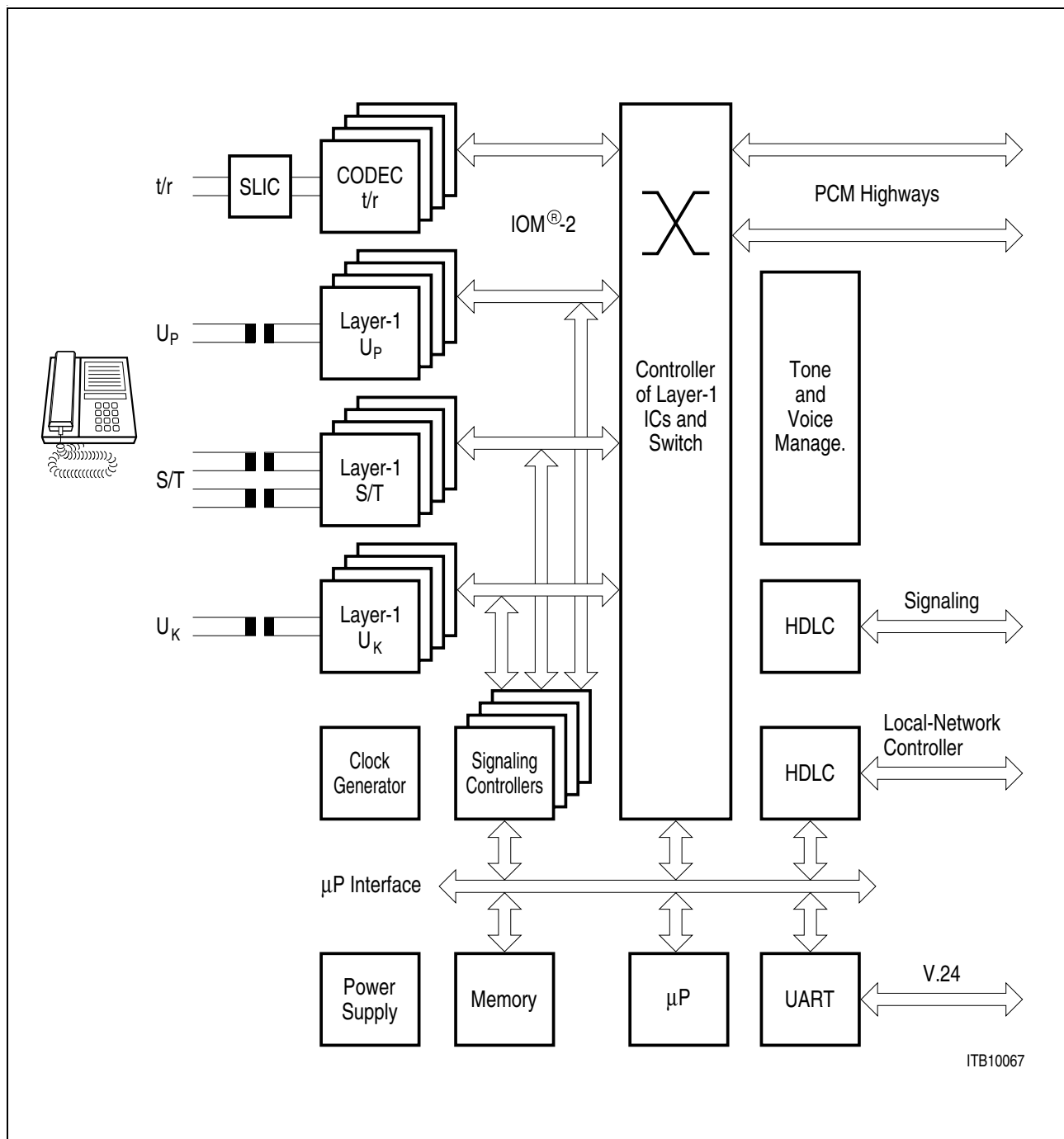
<b>List of Tables</b>		<b>Page</b>
Table 5-27	OAK MAIL BOX .....	5-20
Table 5-28	.....	5-24
Table 5-29	.....	5-26
Table 5-30	.....	5-27
Table 5-31	.....	5-30
Table 5-32	.....	5-31
Table 5-33	.....	5-31
Table 5-34	.....	5-32
Table 5-35	.....	5-34
Table 5-36	.....	5-37
Table 5-37	.....	5-38
Table 5-38	.....	5-39
Table 5-39	Downstream .....	5-40
Table 5-40	Upstream .....	5-41
Table 5-41	Time-Slot Encoding for Data Memory Accesses .....	5-42
Table 5-42	Time-Slot Encoding for Control Memory Accesses .....	5-42
Table 5-43	.....	5-46
Table 5-44	Summary of MF-Channel Commands .....	5-51
Table 5-45	.....	5-54
Table 5-46	.....	5-69
Table 5-47	.....	5-98
Table 6-1	An Example for Required DSP Performance in a Comfort PBX with 30 Subscribers .....	6-10
Table 7-1	.....	7-1
Table 7-2	.....	7-1
Table 7-3	.....	7-2
Table 7-4	.....	7-3
Table 7-5	.....	7-3
Table 7-6	Bus Interface Timing .....	7-5
Table 7-7	Siemens/Intel Interrupt Timing .....	7-7
Table 7-8	Motorola Interrupt Timing .....	7-10
Table 7-9	Interrupt Timing .....	7-12
Table 7-10	PDC and PFS Timing In Master Mode .....	7-13
Table 7-11	PCM-Interface Timing in Master Mode .....	7-15
Table 7-12	PCM Interface Timing In Slave Mode .....	7-18
Table 7-13	IOM <sup>®</sup> -2 Interface Clocks Timing when FSC and DCL are Driven by ELIC0 and the DOC is in Slave Mode .....	7-20
Table 7-14	IOM <sup>®</sup> -2 Interface Clocks Timing When FSC and DCL are Driven directly by PDC4/8 and PFS, and the DOC is in Slave Mode .....	7-22
Table 7-15	IOM <sup>®</sup> -2 Interface Clocks Timing when FSC and DCL are Driven Directly by PDC4/8 and PFS, and the DOC is in Master Mode (PDC and PFS are generated by internal clocks generator) .....	7-23

<b>List of Tables</b>	<b>Page</b>
Table 7-16 IOM <sup>®</sup> -2 Interface Timing . . . . .	7-24
Table 7-17 FSCD (Delayed FSC) Timing . . . . .	7-26
Table 7-18 Channel Indication (CHI) Timing . . . . .	7-27
Table 7-19 DRDY Timing . . . . .	7-28
Table 7-20 Serial Interface Timing . . . . .	7-28
Table 7-21 Reset Timing . . . . .	7-33
Table 7-22 Boundary Scan Timing . . . . .	7-34
Table 7-23 Program Read Access Timing . . . . .	7-35
Table 7-24 External Data Read Access Timing . . . . .	7-37
Table 7-25 External Program/Data Write Access . . . . .	7-41
Table 7-26 CLK61 (input) Timing . . . . .	7-45
Table 7-27 CLK16 (input) Timing . . . . .	7-46
Table 7-28 REFCLK Timing . . . . .	7-46
Table 7-29 XCLK (input) Timing . . . . .	7-47
Table 7-30 CLK30 (output) Timing . . . . .	7-48
Table 7-31 CLK15 (output) Timing . . . . .	7-48
Table 7-32 CLK7 (output) Timing . . . . .	7-49
Table 9-1 Timing Characteristics of the IOM <sup>®</sup> -2 . . . . .	9-3
Table 9-2 Timing Characteristics of the IOM <sup>®</sup> -2 . . . . .	9-4

# 1 Overview

A Small PBX and a Line Card consist of the following main functional blocks: A switching matrix, multiple Layer-1 transceivers for t/r (a/b), S/T, U<sub>P</sub> and U<sub>K</sub> interfaces, IOM-2 interface controller, signaling controllers, a DSP for tone and voice management, a microprocessor, memory, clock generator, power supply and transformers.

Figure 1-1 shows main functional blocks of a PBX.

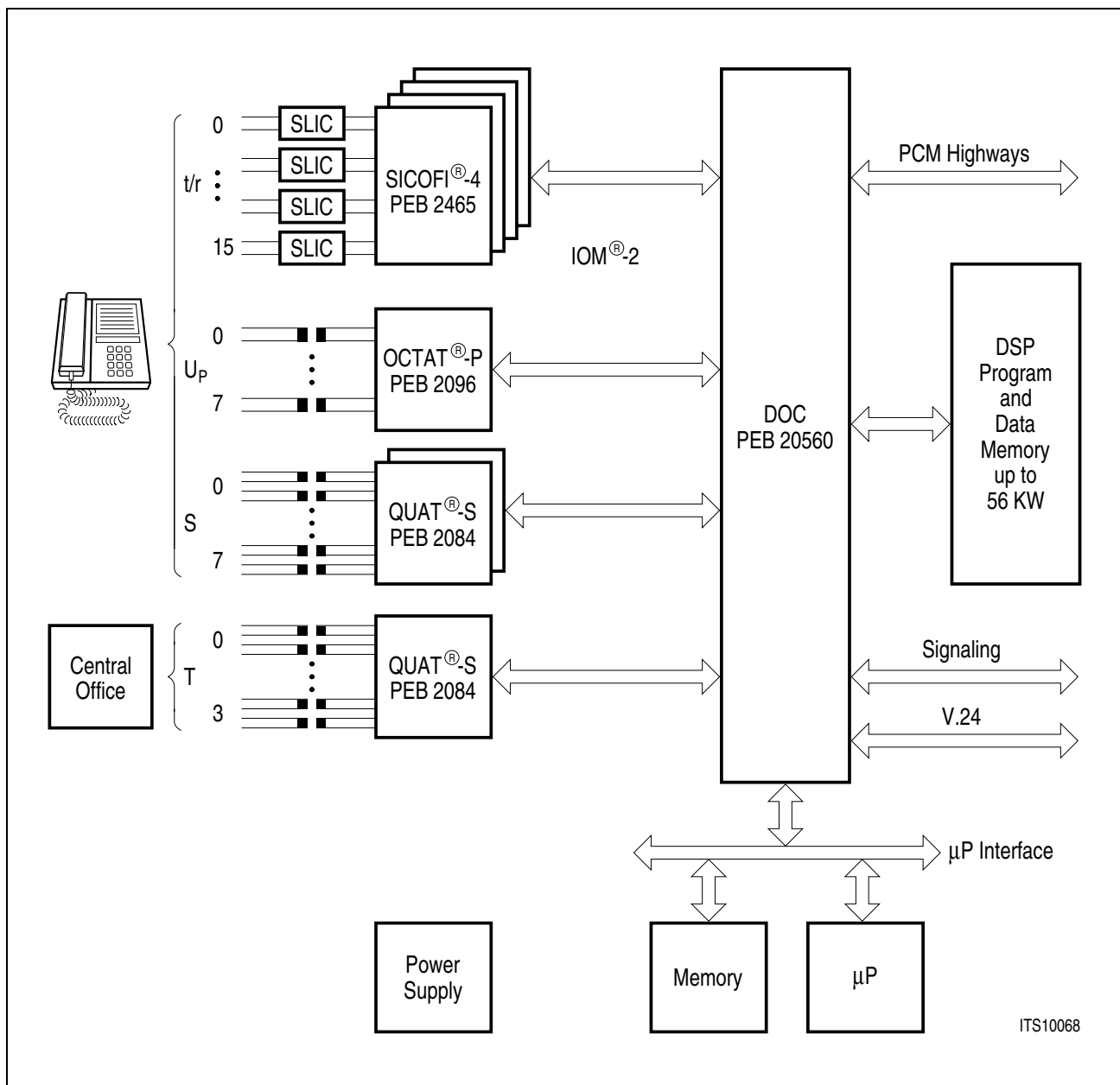


ITB10067

Figure 1-1 Functional Blocks of a PBX

The new Siemens generation of highly integrated ISDN circuits enables design engineers to decrease board size and thus PBX size and its production costs.

**Figure 1-2** shows an example of a PBX for 16 ISDN and 16 analog subscribers with 4 trunk lines realized with a few highly integrated chips of the new Siemens family of PBX and Line Card ICs: DOC, SICOFI-4, OCTAT-P and QUAT-S.



**Figure 1-2 Application Example**  
**PBX for 32 Subscribers with 4 Trunk Lines using one DOC**

**DOC**, DSP Oriented PBX Controller, PEB 20560, The DOC integrates many different functional blocks on a single chip for building small PBXs or PBX Line Cards: Two ELICs, Enhanced Line Card Controller (PEB 20550), one SIDECC, 4-channel signaling controller

(LAP-D), multiple IOM-2 and PCM interfaces, one up-to 40 MIPS DSP with on-chip emulation and a Mailbox, one PCM-DSP interface for fast DSP access, one UART, Interrupt Controller, ...

The DOC is a CMOS device offered in a P-MQFP-160 package.

**QUAT<sup>®</sup>-S**, Quadruple Transceiver for S/T Interfaces, PEB 2084, implements 4 four-wire S/T interfaces to link voice/data digital terminals to PBX subscriber lines and PBX trunk lines to the public ISDN. It can handle up to four S/T interfaces simultaneously in accordance with CCITT I.430, ETSI 300.012, and ANSI T1.605 standards.

The QUAT-S is a CMOS device offered in a P-MQFP-44 package.

**OCTAT<sup>®</sup>-P**, Octal Transceiver for  $U_{PN}$  interfaces, PEB 2096, implements the two-wire  $U_{PN}$  interface used to link voice/data digital terminals to PBX subscriber lines. The OCTAT-P is an optimized device for LT applications and can handle up to eight  $U_{PN}$  interfaces simultaneously. It handles the  $U_{PN}$  interfaces in accordance with the  $U_{P0}$  interface specification except for the reduced loop length.

The OCTAT-P is a CMOS device offered in a P-MQFP-44 package.

**SICOFI<sup>®</sup>-4**, Programmable signaling and CODEC Filter with 4 channels, PEB 2465, implements 4 t/r (a/b) interfaces to link analog voice terminals to PBX subscriber lines and analog PBX trunk lines to public switches. An integrated Digital Signal Processor handles all the algorithms necessary e.g. transhybrid-loss adaptation, gain, frequency response, impedance matching. The IOM-2 Interface handles digital voice transmission, SICOFI-4 feature control and transparent access to the SICOFI-4 command and indication pins. To program the filters, precalculated sets of coefficients are downloaded from the system to the on-chip coefficient RAM. Thus it is possible to use the same line card in different countries.

The SICOFI-4 is a CMOS device offered in P-MQFP-64 package.

### **ISDN-Oriented Modular Interface (IOM<sup>®</sup>-2)**

The "Group of Four", ALCATEL, Siemens, Plessey and ITALTEL systems houses, originally defined a General Circuit Interface (GCI) with the aim of specifying a comprehensive interface which would allow various telecommunication devices to communicate in an efficient manner. The IOM-2 interface is a four-wire interface. It became a standard interface for interchip communication in ISDN applications. All above ICs are compatible and operate from a single 5 V power supply (incl. SICOFI-4).

### DSP Oriented PBX Controller (DOC)

The DOC is comprising all necessary functional blocks like switching, signaling, DTMF/tone handling and conferencing on a single chip.

The transceivers (layer 1 ICs) are not integrated.

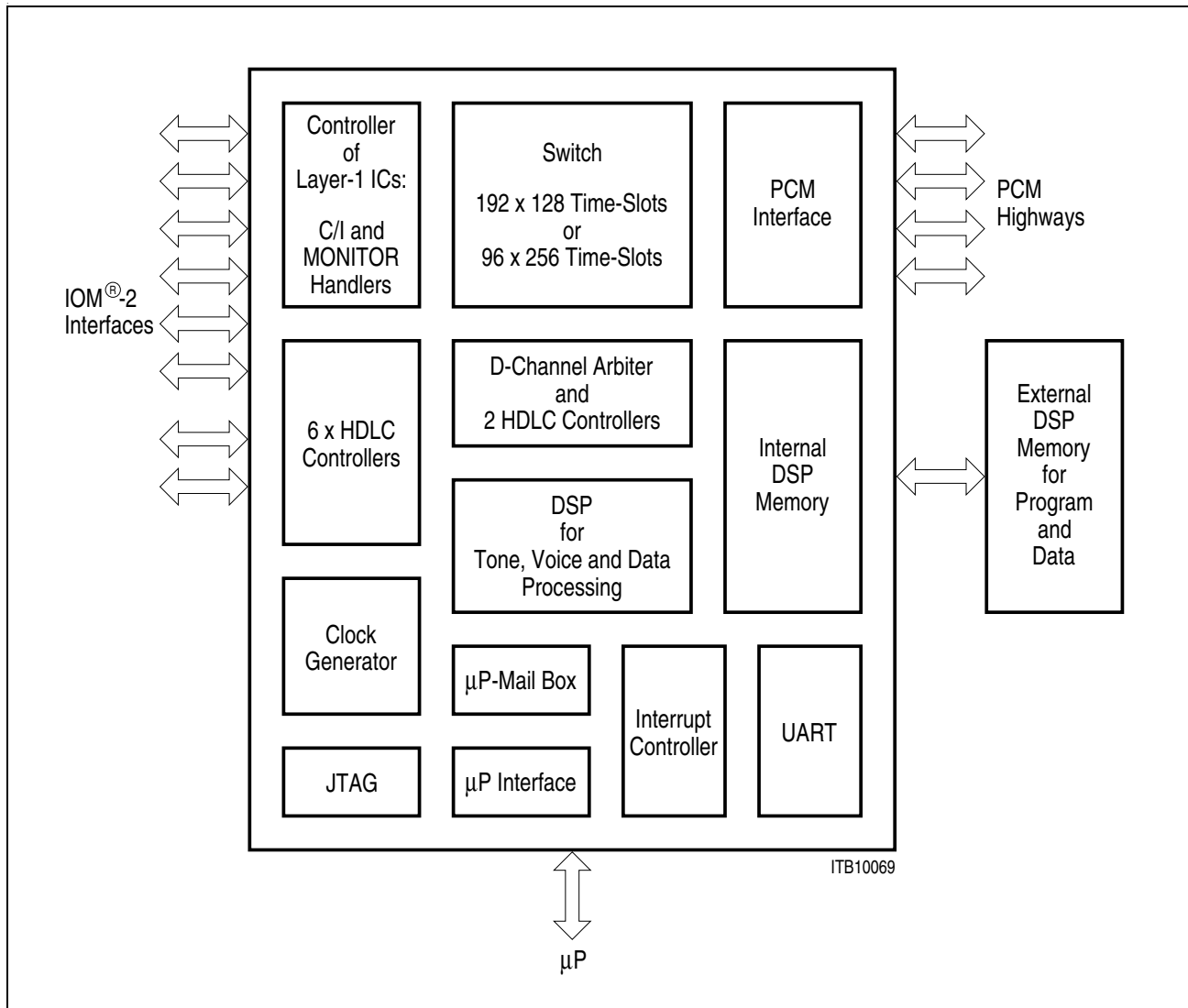


Figure 1-3 Principle Block Diagram of the DOC



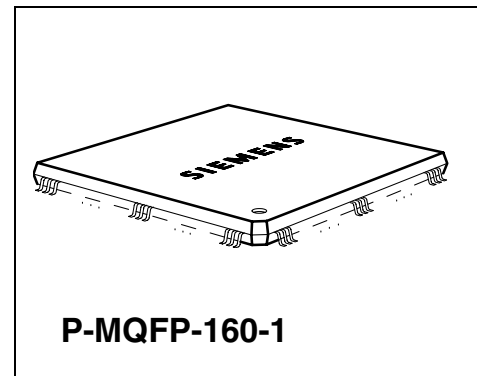
**Version 2.1**

**CMOS**

**1.1 DOC Features**

The DOC provides all necessary features for building PBX (Privat Branch eXchange) systems and Line Cards.

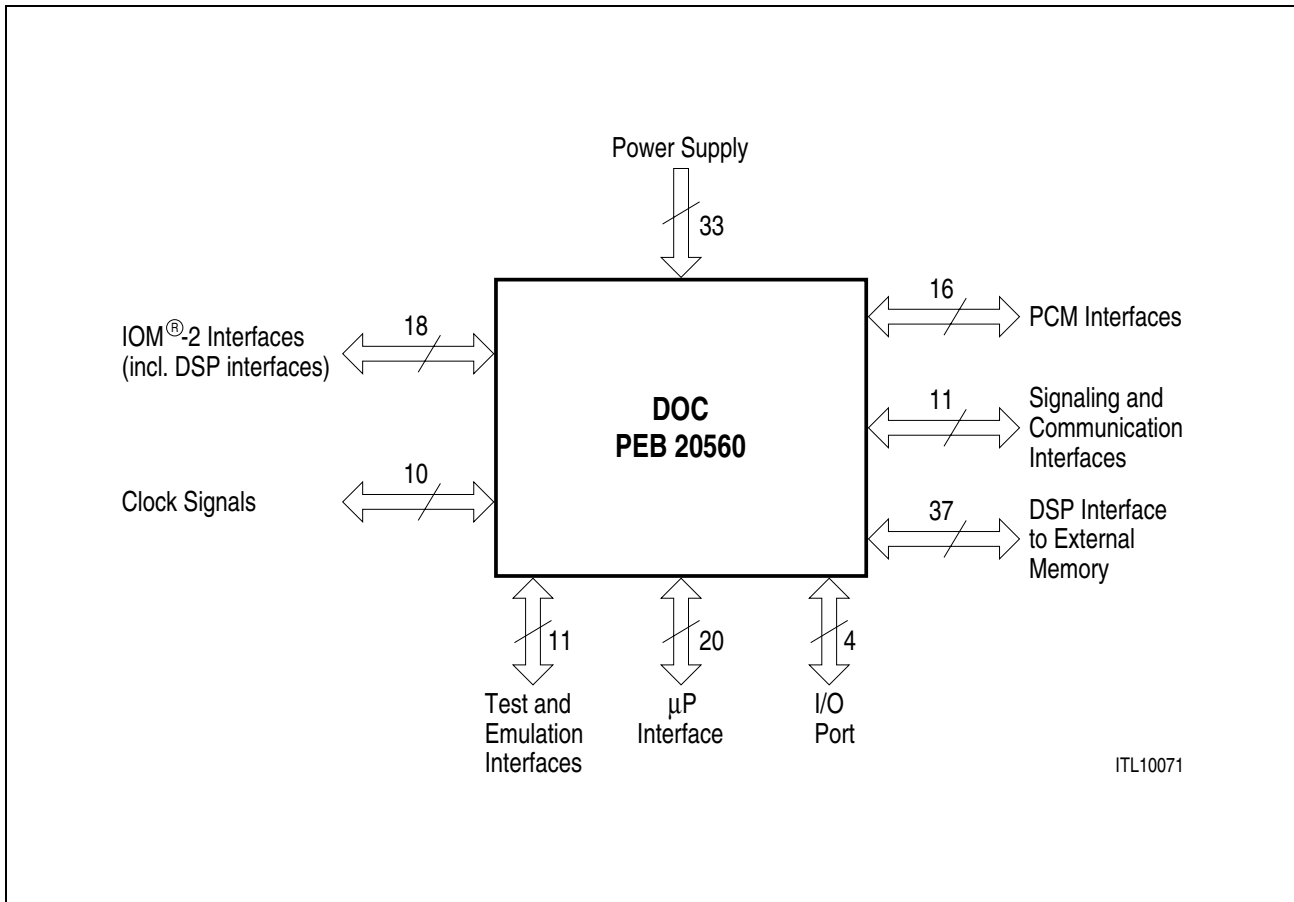
- In the PBX mode (**Figure 2-1**), the DOC provides:
  - 6 fully usable IOM-2 (GCI) interfaces and thus it can control up to 48 ISDN or 96 analog subscribers.
  - 4 PCM highways with 128 time-slots in total.
- In the Line Card mode (**Figure 2-3**), the DOC provides:
  - 2 fully usable IOM-2 interfaces with 16 IOM-2 subframes (2 × 8) and
  - 2 limited IOM-2 interfaces, as two DSP ports are connected, and thus it can control 16 to 24 ISDN (or 32 to 48 analogue) subscribers.
  - 4 PCM highways with 256 time-slots in total.
- Signaling via 8 assignable HDLC controllers, each with a 64-byte data FIFO for transmit and for receive direction.
  - 2 HDLC controllers (SACCO-A) assignable to two of up to 48 ISDN subscribers at a time via two different D-channel Arbiters
  - 4 HDLC controllers (SIDEK) assignable to any D-/B-channel in data upstream or data downstream direction on four IOM-2 interfaces.
  - 2 HDLC controllers (SACCO-B) assignable to any time-slot in data upstream or data downstream direction of the four IOM-2 interfaces or the PCM highways. Optionally, both controllers can be used as stand alone HDLC controllers with up to 8.192 Mbit/s transfer rate. They support DMA operation.
- On-chip user programmable 16-bit Digital Signal Processor, OAK<sup>®</sup>, (with 20, 30 or up to 40 MIPS) with access to 64 time-slots (via one or two internal IOM-2 interfaces) for DTMF/Tone generation and recognition, conferencing, music-on-hold, modem emulation, etc.
  - 1 K × 16-bit on-chip data memory (X)
  - 512 × 16-bit on-chip data memory (Y)



Type	Ordering Code	Package
PEB 20560 V2.1	Q67231-H1007	P-MQFP-160-1

- DSP proprietary interface to an external memory:
  - \* Program memory up to 56 K × 16-bit
  - \* Data memory up to 32 K × 16-bit
- On-chip program ROM (Boot) ~ 0.5 K × 16-bit
- a-/μ-law coding and decoding by hardware (on the fly)
- Firmware for DSP work load measurement (within every 125 μs frame)
- Protection against write into program memory using password
- μP-DSP communication via two Mail-Boxes
- On-chip emulation (OCEM®) for DSP program debugging
- 8-bit μP Interface compatible with Siemens/Intel bus schemes
- Programmable clock generator with built-in logic for Master and Slave configurations
- Watch-Dog timer
- Reset logic
- UART for V.24 Interface
- Multifunctional Input/Output Port configurable as a general I/O Port, DMA lines for SACCO-B0 or as additional UART lines for Modem connection
- Integrated Interrupt controller with vector generation and support for DOC cascading
- Interrupt vector handling compatible with Siemens/Intel/Motorola bus schemes
- Up to 4 external interrupt inputs (via the general I/O Port)
- JTAG Interface for on board tests
- Interface for HW and SW DSP evaluation (debugging)
- Advanced CMOS 0.5 μm technology
- 3.3-V and 5-V Power Supply in 5-V environment
- 3.3-V Power Supply in 3.3-V environment
- TTL driving capability, TTL and CMOS compatible inputs
- P-MQFP-160 package

## 1.2 Logic Symbol



**Figure 1-4 DOC Logic Symbol**

(160 pins are used)

### 1.3 Pin Configuration (top view)

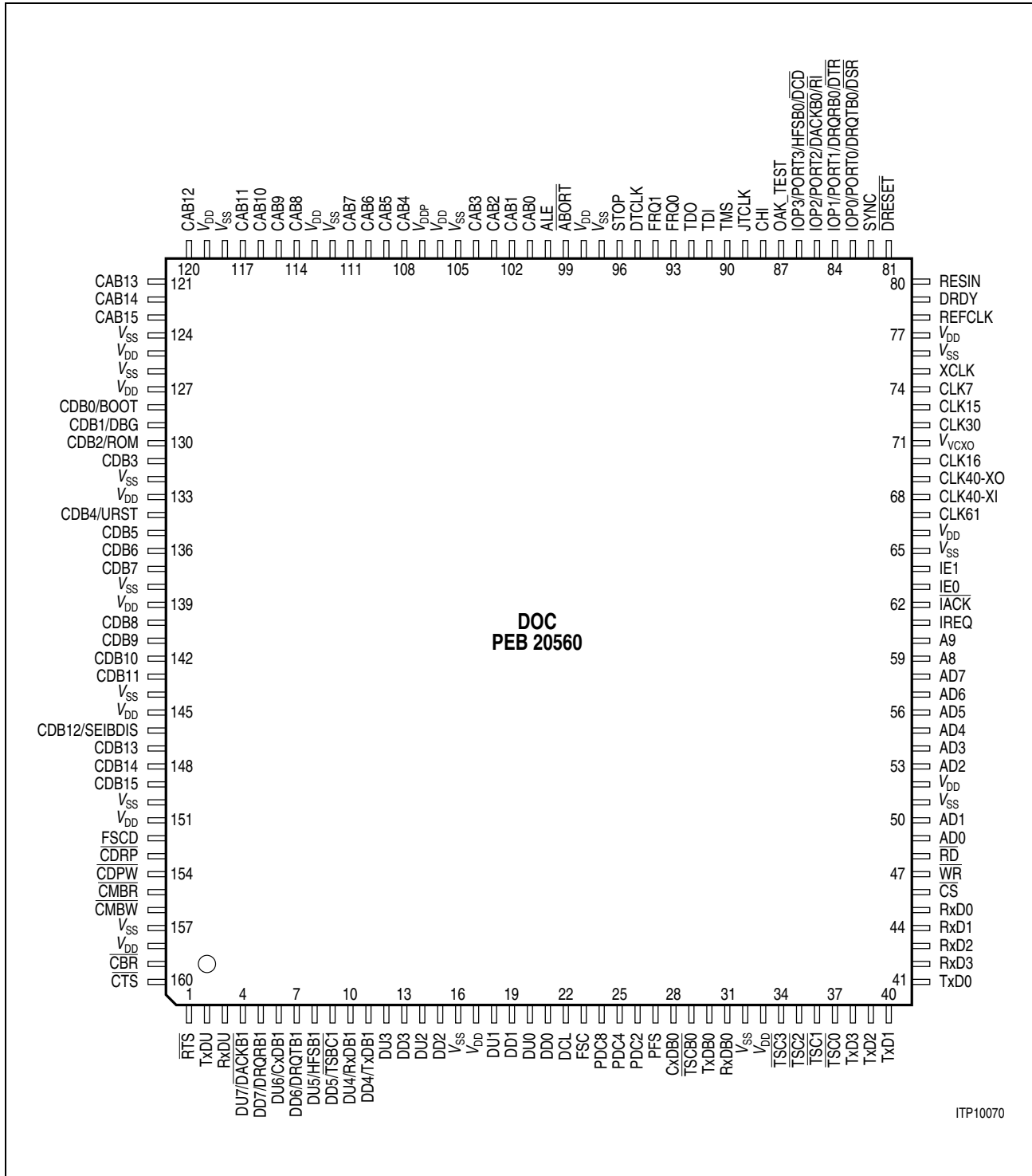


Figure 1-5 DOC Pin Configuration (P-MQFP-160 Package)

## 1.4 Pin Description

**Table 1-1 IOM<sup>®</sup>-2 Interface**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
23	FSC	I/O	I	<b>Frame Synchronization Clock (8 kHz)</b>
22	DCL	I/O	I	<b>Data CLock: Single or double data rate.</b>
21	DD0	O(OD)	High Impedance	<b>Data Downstream IOM-2 Interface 0</b>
20	DU0	I	I	<b>Data Upstream IOM-2 Interface 0</b>
19	DD1	O(OD)	High Impedance	<b>Data Downstream IOM-2 Interface 1</b>
18	DU1	I	I	<b>Data Upstream IOM-2 Interface 1</b>
15	DD2	O(OD)	High Impedance	<b>Data Downstream IOM-2 Interface 2</b>
14	DU2	I	I	<b>Data Upstream IOM-2 Interface 2</b>
13	DD3	O(OD)	High Impedance	<b>Data Downstream IOM-2 Interface 3</b>
12	DU3	I	I	<b>Data Upstream IOM-2 Interface 3</b>
11	DD4/ TxDB1	O(OD) O	High Impedance (from ELIC1)	<b>Data Downstream IOM-2 IOM-2 Interface 4/ Transmit Serial Data SACCO Channel B1.</b> Data output line of the corresponding HDLC-transmit channel. Data is sampled on the bit CCR1:ODS the pins have push pull or open drain characteristic. When transmission is disabled ( $\overline{TSC} = 1$ ) or when bit CCR2:TXDE is reset the pin is in the state high impedance.
10	DU4/ RxDB1	I I	I	<b>Data Upstream IOM-2 Interface 4/ Receive Serial Data SACCO Channel B1.</b> The serial data received on this line is forwarded into the corresponding HDLC-receive channel. Data is sampled on the – falling edge of HDC (CCR2:RDS = 0) or – rising edge of HDC (CCR2:RDS = 1).

**Table 1-1 IOM<sup>®</sup>-2 Interface (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
9	DD5/ TSCB1	O(OD) O	High Impedance (from ELIC1).	<b>Data Downstream IOM-2 Interface 5/ Tri-State Control SACCO Channel B1</b> , active low. Supplies a control signal for an external driver. When low the corresponding TxD-outputs are valid. The detailed functionality is defined programming the SACCO-registers CCR2:SOC1,SOC0.
8	DU5/ HFSB1	I I	I	<b>Data Upstream IOM-2 Interface 5/ HDLC-Interface Frame Synchronization SACCO Channel B1</b> Frame synchronization pulse in clock mode 2, data strobe in clock mode 1.
7	DD6/ DRQTB1	O(OD) O	High Impedance (from ELIC1)	<b>Data Downstream IOM-2 Interface 6/ DMA-Request Transmitter SACCO Channel B1</b> The transmitter of HDLC-Channel SACCO requests a DMA-data transfer by activating this line. The DRQT-pin remains “high” as long as the transmit FIFO requires data transfers. The number of data bytes to be transferred from system memory to the FIFO must be written first into the XBCH, XBCL registers (byte count registers).
6	DU6/ CxDB1	I I	I	<b>Data Upstream IOM-2 Interface 6/ Collision Data SACCO Channel B1</b> In a bus configuration, the external serial bus must be connected to the respective CxD pin for collision detection. In point-to-point configurations the pin provides a “clear to send” function. When ‘0’/‘1’ the transmit channel is enabled/disabled. If this function is not needed the CxDB1 input line has to be tied to $V_{SS}$ .

**Table 1-1 IOM<sup>®</sup>-2 Interface (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
5	DD7/ DRQRB1	O(OD) O	High Impedance (from ELIC1)	<b>Data Downstream IOM-2 Interface 7/ DMA-ReQuest Receiver Channel B1</b> The receiver of SACCO Channel requests a DMA-data transfer by activating this line. The DRQR-pin remains “high” as long as the receiver FIFO requires data transfers. Only blocks of 32, 16, 8 or 4 bytes are transferred.
4	DU7/ $\overline{\text{DACKB1}}$	I I	I	<b>Data Upstream IOM-2 Interface 7/ DMA-ACKnowledge SACCO Channel B1, active low.</b> When “low”, this line notifies the SACCO HDLC-channel, that the requested DMA-cycle is in progress. Together with $\overline{\text{RD}}$ (DRQR) or $\overline{\text{WR}}$ (DRQT) $\overline{\text{DACK}}$ works like $\overline{\text{CS}}$ to enable a read or write operation to the top of the receive or the transmit FIFO. When $\overline{\text{DACK}}$ is active, the address lines are ignored and the FIFOs are implicitly selected. When DACKB1 is not used the input line must be connected to $V_{\text{DD}}$ .

*Note: DU 7...0 pins can be used only as inputs and not as I/O pins.  
 DD 7...0 pins can be used only as outputs and not as I/O pins.  
 The SLD mode, of the ELICs, within the DOC, is not valid.  
 If DCL is programmed to be input, FSC will also be an input.  
 TEST0 and TEST1 pins must be connected to '0' when not used.*

**Table 1-2 PCM Interface**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
27	PFS	I/O	I	PCM-Interface Frames Synchronization Clock/ Master Clock
26	PDC2	I/O	I	PCM-Interface Data Clock / Master Clock 2.048 MHz
25	PDC4	I/O	I	PCM-Interface Data Clock / Master Clock 4.096 MHz
24	PDC8	I/O	I	PCM-Interface Data Clock / Master Clock 8.192 MHz
45	RxD0	I	I	Receive PCM-Interface Data
44	RxD1	I		
43	RxD2	I		
42	RxD3	I		
41	TxD0	O	High Impedance	Transmit PCM-Interface Data
40	TxD1	O		
39	TxD2	O		
38	TxD3	O		
37	$\overline{\text{TSC0}}$	O	"High"	Tri-state Control
36	$\overline{\text{TSC1}}$	O		Supplies a control signal for an external driver.
35	$\overline{\text{TSC2}}$	O		These lines are "low" when the corresponding TxD-outputs are valid.
34	$\overline{\text{TSC3}}$	O		

*Note: The maximal input current on the following DOC input lines will not exceed 2.3 mA at 5.5 V signal when the DOC is without power supply (lines connected with the back plane):*

*PFS, PDC2, PDC4, PDC8, XCLK, REFCLK, RxD0 to RxD3, RxDB0, RxDB1, CxDB0, CxDB1, HFSB0 and HFSB1.*

*(This is a protection for the case that a board with a DOC is plugged into the PCM backplane and the DOC is not yet connected to the power supply pins.*

*Refer to **Figure 6-1.**)*



**Table 1-3 Communication and Signaling Interfaces**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
<b>SACCO-B0</b>				
31	RxDB0	I	I	<b>Receive Serial Data HDLC-Channel B0.</b> The serial data received on this line is forwarded into the corresponding HDLC-receive channel. Data is sampled on the <ul style="list-style-type: none"> <li>– falling edge of HDC (CCR2:RDS = 0) or</li> <li>– rising edge of HDC (CCR2:RDS = 1).</li> </ul>
30	TxDB0	O (OD)	High Impedance	<b>Transmit Serial Data HDLC-Channel B0.</b> Data output line of the corresponding HDLC-transmit channel. Data is sampled on the bit CCR1:ODS the pins have push pull or open drain characteristic. When transmission is disabled ( $\overline{TSCB} = 1$ ) or when bit CCR2:TXDE is reset the pin is in the state high impedance. ( <b>Open Drain output.</b> )
29	$\overline{TSCB0}$	O	“High”	<b>Tri-State Control HDLC-Channel B0,</b> active low. Supplies a control signal for an external driver. When low, the corresponding TxD-outputs are valid. The detailed functionality is defined programming the SACCO-registers CCR2:SOC1,SOC0.
28	CxDB0	I	I	<b>Collision Data HDLC-Channel B0</b> In a bus configuration, the external serial bus must be connected to the respective CxD pin for collision detection. In point-to-point configurations the pin provides a “clear to send” function. When ‘0’/‘1’ the transmit channel is enabled/disabled. If this function is not needed the pin has to be tied to $V_{SS}$ .

**Table 1-3 Communication and Signaling Interfaces (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
<b>UART</b>				
3	RxDU	I	I	<b>Receive Serial Data on UART</b> Serial data input from the communications link (peripheral device, modem, or data set).
2	TxDU	O	“Low”	<b>Transmit Serial Data on UART.</b> This is the composite serial data output to the communications link (peripheral, modem or data set). This signal is set to the marking (logical 1 = ‘0’) state upon a master reset operation.
1	$\overline{\text{RTS}}$	O	“High”	<b>Request To Send</b> When low, this informs the modem or data set that the UART is ready to exchange data. The $\overline{\text{RTS}}$ output signal can be set or reset by programming bit 1 (RTS) of the modem control register. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

**Table 1-3 Communication and Signaling Interfaces (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
160	$\overline{\text{CTS}}$	I	I	<p><b>Clear To Send</b> When low, this indicates that the modem or data set is ready to exchange data. The <math>\overline{\text{CTS}}</math> signal is a modem status input whose conditions can be tested by the <math>\mu\text{P}</math> reading bit 4 (CTS) of the modem status register. Bit 4 is the complement of the <math>\overline{\text{CTS}}</math> signal. Bit 0 (DCTS) of the modem status register indicates whether the <math>\overline{\text{CTS}}</math> input has changed state since the previous reading of the modem status register. <math>\overline{\text{CTS}}</math> has no effect on the transmitter.</p> <p><i>Note: Whenever the CTS bit of the modem status register changes state, an interrupt is generated if the modem status interrupt is enabled.</i></p>

Other Lines

88	CHI	O	"Low"	<b>CH</b> annel <b>I</b> ndication Signal on IOM-2
79	DRDY	I	I	<p><b>D</b>-channel <b>ReaDY</b> controls SIDE<math>\overline{\text{C}}</math> in LT-T mode applications (Collision signal that the S/T Interface is not free for signaling; i.e. QUAT-S signal) when not used this input should be tied to "high".</p>
152	FSCD	O	High Impedance	<p><b>F</b>rame <b>S</b>ynchronization <b>C</b>lock with <b>D</b>elay of 62.5 <math>\mu\text{s}</math> related to the standard FSC (Synchronizes layer-1 devices connected to the second half of an extended IOM-2 interface with 64 time-slots; i.e. OCTAT-P or QUAT-S).</p>

*Note: The optional SACCO-B1 signals which are combined with IOM-2 Interface ports 4 to 7 are described in the **Table 1-1**.*

**Table 1-4 DSP External Memory Interface**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
153	$\overline{\text{CDPR}}$	O	“High”	<b>C-Bus Data or Program Read</b> (active low)
154	$\overline{\text{CDPW}}$	O	“High”	<b>C-Bus Data or Program Write</b> (active low)
155	$\overline{\text{CMBR}}$	O	“High”	<b>C-Bus Mail-Box Boot Read</b> (active low)
156	$\overline{\text{CMBW}}$	O	“High”	<b>C-Bus Mail-Box Boot Write</b> (active low)
159	$\overline{\text{CBR}}$	O	“High”	<b>C-Bus Boot Read</b> (active low)
123 122 121 120 117 116 115 114 111 110 109 108 104 103 102 101	CAB15 CAB14 CAB13 CAB12 CAB11 CAB10 CAB9 CAB8 CAB7 CAB6 CAB5 CAB4 CAB3 CAB2 CAB1 CAB0	O		<b>C-Bus Address Bus</b> (CAB0 = lsb)
128	CDB0/ BOOT	I/O I	I	<b>C-Bus Data Bus bit 0</b> (lsb), During reset this pin is used as a strap, called <b>Boot</b> . It enables the OAK to execute the boot routine from internal boot ROM. This routine loads a DSP program into the external program RAM. Refer to <b>section 2.7.9</b>
129	CDB1/ DBG	I/O I	I	<b>C-Bus Data Bus bit 1</b> , <b>DBG</b> (debug): During reset this pin is used as a strap. It may prevents reset of OCEM registers when a reset is initiated by the debugger Refer to <b>section 2.7.9</b>
130	CDB2/ ROM	I/O I	I	<b>C-Bus Data Bus bit 4</b> <b>ROM</b> : During reset this pin is used as a strap. It enables boot from the CDI ROM. Refer to <b>section 2.7.9</b>

**Table 1-4 DSP External Memory Interface (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
131	CDB3	I/O	I	<b>C-Bus Data Bus bit 3.</b>
134	CDB4/ URST	I/O	I	<b>C-Bus Data Bus bit 2,</b> <b>URST:</b> User ReSeT. During reset this pin is used as a strap. this option is used by the emulator.
135	CDB5	I/O	I	<b>C-Bus Data Bus bit 5</b>
136	CDB6	I/O	I	<b>C-Bus Data Bus bit 6</b>
137	CDB7	I/O	I	<b>C-Bus Data Bus bit 7</b>
140	CDB8	I/O	I	<b>C-Bus Data Bus bit 8</b>
141	CDB9	I/O	I	<b>C-Bus Data Bus bit 9</b>
142	CDB10	I/O	I	<b>C-Bus Data Bus bit 10</b>
143	CDB11	I/O	I	<b>C-Bus Data Bus bit 11</b>
146	CDB12/ SEIBDIS	I/O	I	<b>C-Bus Data Bus bit 12</b> SEIBDIS: SEIB DISable. During reset this pin is used as a strap. '0' - Serial emulation boot, via SEIB. '1' - Parallel emulation boot.
147	CDB13	I/O	I	<b>C-Bus Data Bus bit 13</b>
148	CDB14	I/O	I	<b>C-Bus Data Bus bit 14</b>
149	CDB15	I/O	I	<b>C-Bus Data Bus bit 15 (msb)</b>

**Table 1-5 Clock Signals (additional to the IOM<sup>®</sup>-2, PCM and SCC clocks)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
67	CLK61	I	I	Main Clock of 61.44 MHz <sup>1)</sup>
68	CLK40-XI	I	I	External DSP Clock (0 ... 40 MHz)
69	CLK40-XO	O	O	
70	CLK16	I	I	VCXO Clock of 16.384 MHz <sup>2)</sup>
71	V <sub>VCXO</sub>	O	O	9 Control Oscillator (Quartz) Signal
72	CLK30	O	O	30.72 MHz (μP Clock)
73	CLK15	O	O	15.36 MHz (OCTAT-P)
74	CLK7	O	O	7.68 MHz (QUAT-S)
75	XCLK	I	I	External Synchronization Clock (e.g. 1.536 MHz)
78	REFCLK	I/O	I	Reference Clock (e.g. 512 kHz)

<sup>1)</sup> Shall the VCXO not be used, it is necessary to connect any other clock to this pin during reset. This clock is temporarily needed for resetting the internal units ELIC and SIDEK.

<sup>2)</sup> CLK61 must always be connected as f/2 is needed for the DOC internal topic.

**Table 1-6 μP Interface**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
46	$\overline{CS}$	I	I	<b>Chip Select</b> ; active low. A “low” on this line selects all registers for read/write operations.
47	WR	I	I	<b>Write</b> , active low, identifies a write access.
48	RD	I	I	<b>Read</b> , active low
49	AD0	I/O	I	<b>Address and Data Bus</b> ; multiplexed bus mode.  Handles addresses from the μP-system to the DOC and transfers data between the μP and the DOC.
50	AD1	I/O	I	
53	AD2	I/O	I	
54	AD3	I/O	I	
55	AD4	I/O	I	
56	AD5	I/O	I	
67	AD6	I/O	I	
58	AD7	I/O	I	
59	A8	I	I	
60	A9	I	I	

**Table 1-6**  $\mu$ P Interface (cont'd)

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
100	ALE	I	I	<b>Address Latch Enable</b> ALE controls the on chip address latch in multiplexed bus mode. While ALE is “high”, the latch is transparent. The falling edge latches the current address.
61	IREQ	O (OD)	“High”	<b>Interrupt Request</b> is programmable to active high or low. This signal is activated when the DOC requests a CPU interrupt. Due to open drain (OD) characteristic of the output line multiple interrupt sources can be connected together.
62	$\overline{\text{IACK}}$	I	I	<b>Interrupt Acknowledge</b>
63	IE0	I/O	I	<b>Interrupt Enable 0,1</b> Support lines for IACK evaluation; depends of the selected mode (slave or daisy chain).
64	IE1	I	I	
80	RESIN	O	“High”	<b>RESet INdication</b> This pin is set to “high”, when the DOC executes either a power-up reset, a watchdog timer reset, an external reset (DRESET) or a software system reset.
81	$\overline{\text{DRESET}}$	I	–	<b>DOC RESET</b> A “low” forces the DOC into reset state.

*Note: After reset, IE0 remains in input direction.*

**Table 1-7 Input / Output Port**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
83	IOP0	I/O	I	The I/O port is multifunctional and can be programmed to 3 different modes of use: Mode 0: General Purpose I/O Port Mode 1: SACCO-B0 support lines. Mode 2: UART support lines <i>Note: The signal names change accordingly.</i>
84	IOP1	I/O	I	
85	IOP2	I/O	I	
86	IOP3	I/O	I	

**Mode 0: General Purpose I/O Port**

83	PORT0	I/O	I	General purpose I/O lines During and after reset: Mode 0
84	PORT1	I/O	I	
85	PORT2	I/O	I	
86	PORT3	I/O	I	

**Mode 1: SACCO-B0 Support Lines**

83	DRQTB0	O		<b>DMA-ReQuest Transmitter SACCO Channel B0</b> The transmitter of HDLC-Channel SACCO requests a DMA-data transfer by activating this line. The DRQT-pin remains “high” as long as the transmit FIFO requires data transfers. The number of data bytes to be transferred from system memory to the FIFO must be written first into the XBCH, XBCL registers (byte count registers).
84	DRQRB0	O		<b>DMA-ReQuest Receiver Channel B0</b> The receiver of SACCO Channel requests a DMA-data transfer by activating this line. The DRQR-pin remains “high” as long as the receiver FIFO requires data transfers. Only blocks of 32, 16, 8 or 4 bytes are transferred.



**Table 1-7 Input / Output Port (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
85	$\overline{\text{DACKB0}}$	I		<p><b>DMA-ACK</b>nowledge SACCO Channel <b>B0</b>, active low.</p> <p>When “low”, this line notifies the SACCO HDLC-channel, that the requested DMA-cycle is in progress. Together with <math>\overline{\text{RD}}</math> (DRQR) or <math>\overline{\text{WR}}</math>(DRQT) <math>\overline{\text{DACK}}</math> works like <math>\overline{\text{CS}}</math> to enable a read or write operation to the top of the receive or the transmit FIFO. When <math>\overline{\text{DACK}}</math> is active, the address lines are ignored and the FIFOs are implicitly selected.</p> <p>When DACKB0 is not used the pin must be connected to <math>V_{\text{DD}}</math>.</p>
86	HFSB0	I		<p><b>HDLC-Interface Frame Synchronization SACCO Channel B0</b></p> <p>Frame synchronization pulse in clock mode 2, data strobe in clock mode 1.</p>

**Mode 2: Additional UART Support Lines**

83	$\overline{\text{DSR}}$	I		<p><b>Data Set Ready</b></p> <p>When low, this signal indicates that the modem or data set is ready to establish the communications link with the UART. The <math>\overline{\text{DSR}}</math> signal is a modem status input whose condition can be tested by the CPU reading bit 5 (DSR) of the modem status register. Bit 5 is the complement of the <math>\overline{\text{DSR}}</math> signal. Bit 1 (DDSR) of the modem status register indicates whether the <math>\overline{\text{DSR}}</math> input has changed state since the previous reading of the modem status register.<sup>1)</sup></p>
----	-------------------------	---	--	---

**Table 1-7 Input / Output Port (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
84	$\overline{\text{DTR}}$	O		<p><b>Data Terminal Ready</b> When low, this informs the modem or data set that the UART is ready to establish a communications link. The <math>\overline{\text{DTR}}</math> output signal can be set to an active low by programming bit 0 (<math>\overline{\text{DTR}}</math>) of the modem control register to a high level. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.</p>
85	$\overline{\text{RI}}$	I		<p><b>Ring Indicator</b> When low, this signal indicates that a telephone ringing signal has been received by the modem or data set. The <math>\overline{\text{RI}}</math> signal is a modem status input whose condition can be tested by the CPU reading bit 6 (RI) of the modem status register. Bit 6 is the complement of the <math>\overline{\text{RI}}</math> signal. Bit 2 (TERI) of the modem status register indicates whether the <math>\overline{\text{RI}}</math> input signal has changed from a low to a high state since the previous reading of the modem status register.</p> <p><i>Note: Whenever the RI bit of the modem status register changes from a high to a low state, an interrupt is generated if the modem status interrupt is enabled.</i></p>
86	$\overline{\text{DCD}}$	I		<p><b>Data Carrier Detect</b> When low, it indicates that the data carrier has been detected by the modem or data set. The <math>\overline{\text{DCD}}</math> signal is a modem status input whose condition can be tested by the CPU reading bit 7 (DCD) of the modem status register. Bit 7 is the complement of the <math>\overline{\text{DCD}}</math> signal. Bit 3 (DDCD) of the modem status register indicates whether the <math>\overline{\text{DCD}}</math> input has changed state since the previous reading of the modem status register. <math>\overline{\text{DCD}}</math> has no effect on the receiver.<sup>1)</sup></p>

<sup>1)</sup> Whenever the  $\overline{\text{DSR}}$  bit etc.  $\overline{\text{DCD}}$  bit of the modem status register changes state, an interrupt is generated if the modem status interrupt is enabled

**Table 1-8 Power Supply**

Pin No.	Symbol	In (I) Out (O)	Function
16 pins: 17, 33, 52, 66, 77, 98, 106, 113, 119, 125, 127, 133, 139, 145, 151, 158.	$V_{DD}$	I	Positive Power Supply +3.3 V (for core logic)
107	$V_{DDP}$	I	Positive Power Supply +5 V (for input protection and outputs)
15 pins: 16, 32, 51, 65, 76, 97, 105, 112, 118, 124, 126, 132, 138, 144, 150, 157.	$V_{SS}$	I	Ground (0)

**Table 1-9 Test and Emulation Interfaces**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
---------	--------	-------------------	--------------	----------

**Test Interface** for boundary scan according to IEEE Std. 1149.1

89	JTCLK	I	I	<b>JTAG Test CLoCK</b>
90	TMS	I	I	<b>Test Mode Select</b>
91	TDI	I	I	<b>Test Data Input</b>
92	TDO	O	Spec.	<b>Test Data Output</b>

**Emulation Interface and other Control and Test Pins**

95	DTCLK	O		<b>DSP Test CLoCK.</b> (Used for production test only)
96	STOP	O		<b>STOP</b> for external logic when using the OCEM
99	$\overline{\text{ABORT}}$	I	I	Low signal forces OAK to stop program execution and to return control to the debugger
87	OAK_TEST	I	I	Only for use for production testing. The <b>OAK_TEST</b> pin has to be wired to $V_{SS}$ .

**Table 1-9 Test and Emulation Interfaces (cont'd)**

Pin No.	Symbol	In (I) Out (O)	During Reset	Function
82	SYNC	I	I	Internal DOC <b>SYN</b> Chronization during reset and production tests. The SYNC pin has to be wired to $V_{SS}$ .
93	FRQ0	I	I	DSP <b>FR</b> eQ uency selection: 00: DSP frequency = 20 MHz 01: DSP frequency = 30 MHz 10: DSP frequency = External DSP Clock (e.g. 40 MHz) 11: Only for production testing.
94	FRQ1	I	I	

### 1.5 Functional Block Diagram and System Integration

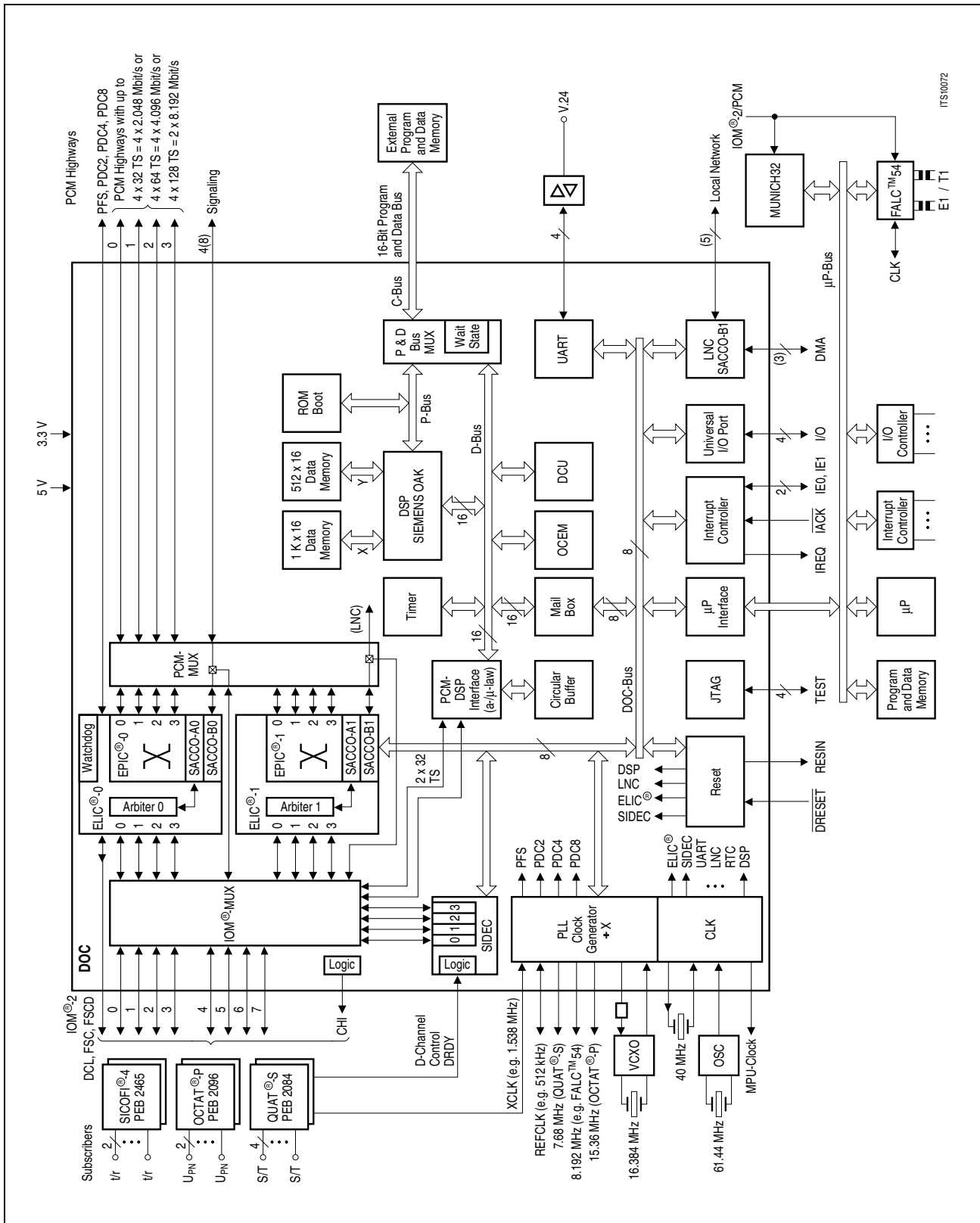
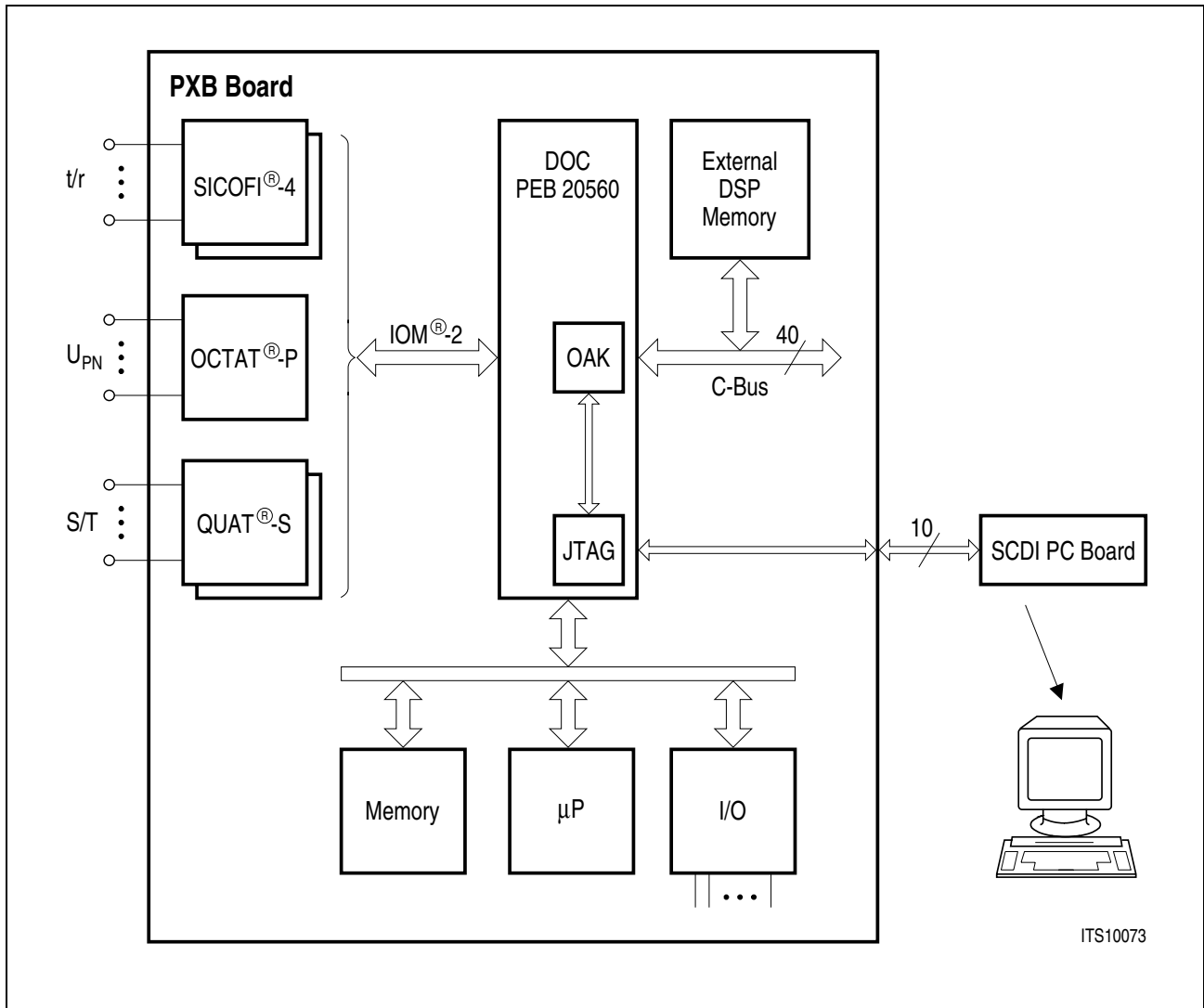


Figure 1-6 Functional Block Diagram and System Integration

1.6 Example for System Integration



ITS10073

Figure 1-7 Example for a PBX with one DOC

## 2 Functional Block Description

DOC Block Diagram and System Integration see **Figure 1-3**.

### 2.1 ELIC0 and ELIC1

#### 2.1.1 General Functions and Device Architecture

The ELIC integrates the existing Siemens device PEB 2055 (EPIC-1), a two channel HDLC-Controller (SACCO: Special Application Communication Controller) with a PEB 2050 (PBC) compatible auto-mode, a D-channel arbiter, a configurable bus interface and typical system glue logic into one chip. It covers all control functions on digital and analog line cards and can be combined via IOM-2 interface with layer-1 circuits or special application devices (e.g. ADPCM/PCM-converters).

#### 2.1.2 Functional Blocks

##### 2.1.2.1 Watchdog Timer

To allow recovery from software or hardware failure, a watchdog timer is provided.

After reset the watchdog timer is disabled. When setting bit SWT in the watchdog timer control register WTC it is enabled. The only possibility to disable the watchdog timer is a ELIC-reset (power-up or DRESET). The timer period is 1024 PFS-cycles assuming that also PDC is active, i.e. a PFS of 8-kHz results in a timer period of 128 ms.

During that period, the bits WTC1 and WTC2 in the register WTC have to be written in the following sequence:

**Table 2-1 Watchdog Timer Programming**

Activity	WTC:WTC1	WTC:WTC2
1.	1	0
2.	0	1

**The minimum required interval between the two write accesses is 2 PDC-periods.**

When the software fails to follow these requirements, a timer overflow occurs and a IWD-interrupt is generated. Additionally an external reset indication (RESIN) is activated. The internal ELIC-status is not changed.

### 2.1.2.2 Reset Logic

After power-up the ELIC is latched into the “Resetting” state. Therefore an integrated power-up reset generator is provided. Additionally an external reset input ( $\overline{\text{DRESET}}$ ) and an reset indication output (RESIN) are available. A microprocessor access is not possible in the “Resetting” state. The ELIC is released from the power-up “Resetting” state when provided with PFS- and PDC-signals for 8 PFS-periods.

The ELIC can also be reset by applying a  $\overline{\text{DRESET}}$ -pulse for at least 4 PDC-periods. Note that such an external  $\overline{\text{DRESET}}$  has priority over a power-on reset. It is thus possible to kill the 8-frame reset duration after power-up. For correct  $\overline{\text{DRESET}}$  a main clock must be applied to CLK61.

During reset all ELIC-outputs with the exception of RESIN and TDO + DRQRA/B + DRQTA/B + SACCO are in the state high impedance. The tri-state control signals of the EPIC-1 PCM-interface ( $\overline{\text{TSC}}[3:0]$ )  $\overline{\text{TSCA/B}}$  are not tri-stated during a chip reset. Instead they are high during reset, thus containing the correct tri-state information for external drivers.

RESIN is set upon power up,  $\overline{\text{DRESET}}$  and the expiring of the watchdog timer. It may be used as a system reset. RESIN is activated for 8 PFS-periods (assuming an active PDC-input) or it has the same pulse width as  $\overline{\text{DRESET}}$ .  $\overline{\text{DRESET}}$  has priority over internal generated resets with respect to the RESIN pulse width. The activation of  $\overline{\text{DRESET}}$  causes an immediate activation of RESIN. Upon the deactivation of  $\overline{\text{DRESET}}$  however, RESIN is deactivated only with the next rising PDC-edge. A PFS-frequency of 8-kHz results in a RESIN-period of 1 ms.

When setting bit VNSR:SWRX RESIN is also activated but the ELIC itself is not reset. This feature supports a proper reset procedure for devices which require dedicated clocking during reset. The sequence required is as follows:

1. Initialize EPIC-1 for a timer interrupt
2. Set bit VNSR:SWRX to ‘1’, RESIN is activated
3. When the timer interrupt occurs, RESIN is deactivated
4. Set bit VNSR:SWRX to ‘0’
5. Read ISTA\_E, in order to deactivate timer interrupt

**Table 2-2 Reset Activities**

	Internal ELIC Reset	RESIN Activation	RESIN Pulse Width
Power up	X	X	8 PFS
Watchdog timer under flow	–	X	8 PFS
External reset ( $\overline{\text{DRESET}}$ )	X	X	$\overline{\text{DRESET}}$
Setting of bit SWRX	–	X	Programmable



When  $V_{DD}$  drops under normal operation the reset logic has the following behavior:

**Table 2-3 Behavior of the Reset Logic in the Case of Voltage Drop**

$V_{DD}$	Behavior
$> 3 V$	No internal reset, no RESIN
$< 1 V$	Internal reset and RESIN after $V_{DD}$ goes up again
$1 V \leq V_{DD} \leq 3 V$	Not defined

*Note: The power-up reset generator must not be used as a supply voltage control element.*

### 2.1.2.3 EPIC<sup>®</sup>-1

#### 2.1.2.3.1 PCM-Interface

The PCM-interface formats the data transmitted or received at the PCM-highways. It can be configured as one (max. 8.192 Mbit/s), two (max. 4.096 Mbit/s) or four (max. 2.048 Mbit/s) PCM-ports, consisting each of a data receive (RxD#), a data transmit (TxD#) and an output tri-state indication line ( $\overline{TSC\#}$ ).

Port configuration, data rates, clock shift and sampling conditions are programmable.

The newly implemented PCM-mode 3 is similar to mode 1 (two PCM-highways). Unlike mode 1 the pins TxD1, TxD3 are not tri-stated but drive the inverted values of TxD0, TxD2.

#### 2.1.2.3.2 Configurable Interface

In order to optimize the on-board interchip communication, a very flexible serial interface is available. It formats the data transmitted or received at the DDn-, DUn- or SIPn-lines. Although it is typically used in IOM-2 or SLD-configuration to connect layer-1 devices, application specific frame structures can be defined (e.g. to interface ADPCM-converters or maintenance blocks).

#### 2.1.2.3.3 Memory Structure and Switching

The memory block of the EPIC-1 performs the switching functionality.

It consists of four sub blocks:

- Upstream data memory
- Downstream data memory
- Upstream control memory
- Downstream control memory

---

## Functional Block Description

The PCM-interface reads periodically from the upstream (writes periodically to the downstream) data memory (cyclical access), see **Figure 2-1**.

The CFI reads periodically the control memory and uses the extracted values as a pointers to write to the upstream (read from the downstream) data memory (random access). In the case of C/I- or signaling channel applications the corresponding data is stored in the control memory. In order to select the application of choice, the control memory provides a code portion.

The control memory is accessible via the  $\mu$ P-interface. In order to establish a connection between CFI time-slot A and PCM-interface time-slot B, the B-pointer has to be loaded into the control memory location A.

### 2.1.2.3.4 Pre-processed Channels, Layer-1 Support

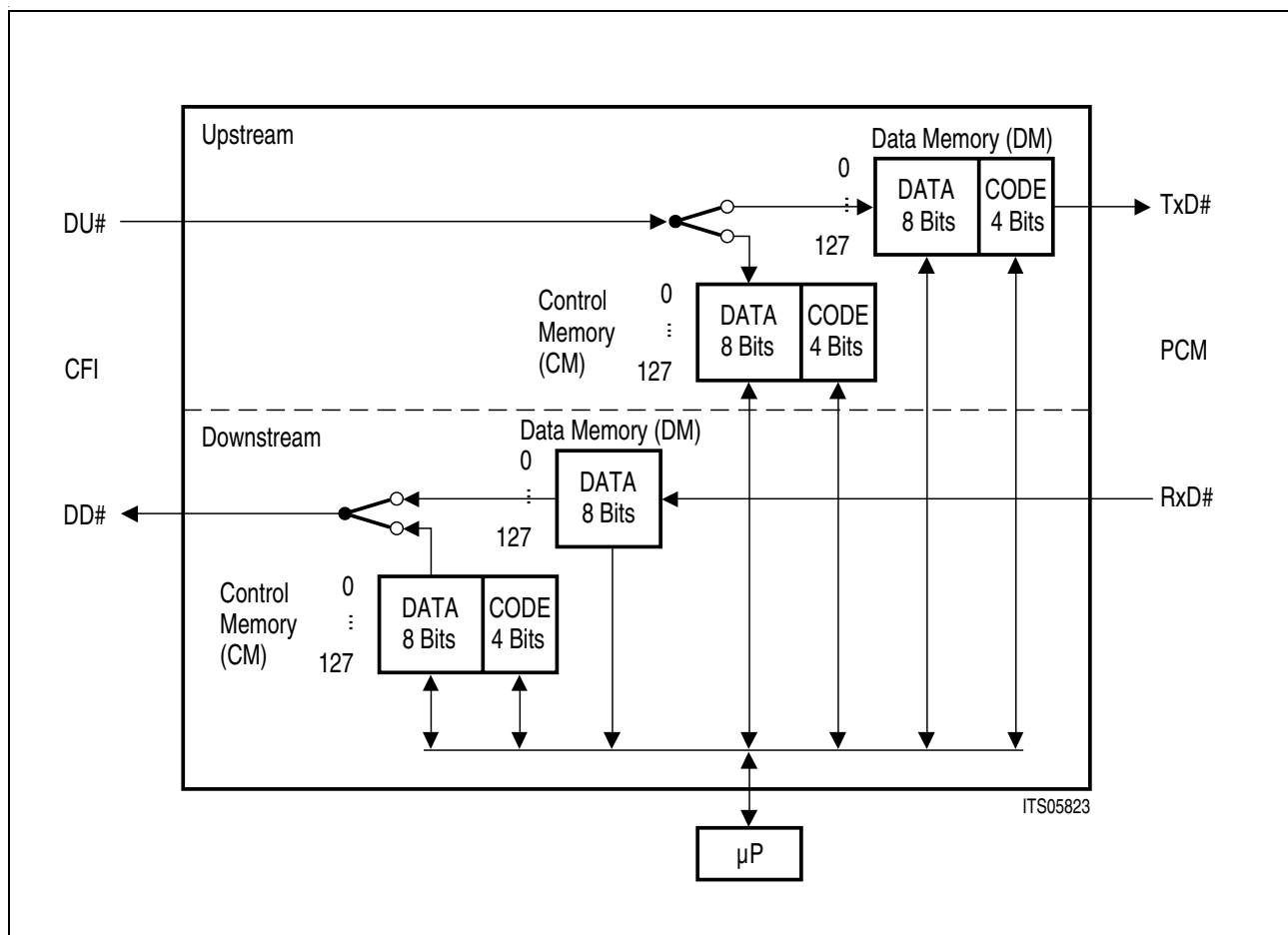
The EPIC-1 supports the monitor/feature control and control/signaling channels according to SLD- or IOM-2 interface protocol.

The monitor handler controls the data flow on the monitor/feature control channel either with or without active handshake protocol. To reduce the dynamic load of the CPU a 16-byte transmit/receive FIFO is provided.

The signaling handler supports different schemes (D-channel +C/I-channel, 6-bit signaling, 8-bit signaling).

In downstream direction the relevant content of the control memory is transmitted in the appropriate CFI time-slot. In the case of centralized ISDN D-channel handling, a 16-kbit/s D-channel received at the PCM-interface is included.

In upstream direction the signaling handler monitors the received data. Upon a change it generates an interrupt, the channel address is stored in the 9-byte deep C/I FIFO and the actual value is stored in the control memory. In 6-bit and 8-bit signaling schemes a double last look check is provided.



**Figure 2-1 EPIC®-1 Memory Structure**

### 2.1.2.3.5 Special Functions

- Synchronous transfer.  
This utility allows the synchronous  $\mu$ P-access to two independent channels on the PCM- or CFI-interface. Interrupts are generated to indicate the appropriate access windows.
- 7-bit hardware timer.  
The timer can be used to cyclically interrupt the CPU, to determine the double last look period, to generate a proper CFI-multiframe synchronization signal or to generate a defined RESIN pulse width.
- Frame length checking.  
The PFS-period is internally checked against the programmed frame length.
- Alternative input functions.  
In PCM-mode 1 and 2, the unused ports can be used for redundancy purposes. In these modes, for every active input port a second input port exists which can be connected to a redundant PCM-line. Additionally the two lines are checked for mismatches.

#### 2.1.2.4 SACCO

The SACCO (Special Application Communication Controller) is a high level serial communication controller consisting of two independent HDLC-channels (A +B). It is a derivative product of the Siemens SAB 82525 (HSCX).

The SACCO essentially reduces the hardware and software overhead for serial synchronous communication. SACCO channel A can be multiplexed by the D-channel arbiter to serve multiple subscribers.

In the following section one SACCO channel is described referring to as "SACCO".

##### 2.1.2.4.1 Block Diagram

The SACCO (one channel) provides two independent 64-byte FIFOs for receive and transmit direction and a sophisticated protocol support. It is optimized for line card applications in digital exchange systems and offers special features to support:

- Communication between a line card and a group controller
- Communication between terminal equipment and a line card

##### 2.1.2.4.2 Parallel Interface

All registers and the FIFOs are accessible via the DOC parallel  $\mu$ P-interface. The FIFOs allocate an address space of 32 bytes each. The data in the FIFOs can be managed by the CPU- or a DMA-controller.

To enable the use of block move instructions, the top of FIFO-byte is selected by any address in the reserved range.

#### Interrupts

The SACCO indicates special events by issuing an interrupt request. The cause of a request can be determined by reading the interrupt status register ISTA\_A/B or EXIR\_A/B. The related register is flagged in the top level ISTA (refer to **Figure 3-1**).

Three indications are available in ISTA\_A/B, another five in the extended interrupt register EXIR\_A/B. An interrupt which is masked in the MASK\_A/B is not indicated in the top level register and the  $\overline{\text{INT}}$ -line is not activated. The interrupt is also not visible in the local registers ISTA\_A/B but remains stored internally and will be indicated again when the corresponding MASK\_A/B-bit is reset.

The SACCO-interrupt sources can be splitted in three logical groups:

- Receive interrupts (RFS, RPF, RME, EHC)
- Transmit interrupts (XPR, XMR)
- Special condition interrupts (XDU/EXE, RFO)

For further information refer to **chapter 3.1.4.1** (Data Transmission in Interrupt Mode) and **chapter 3.1.4.3** (Data Reception in Interrupt Mode).

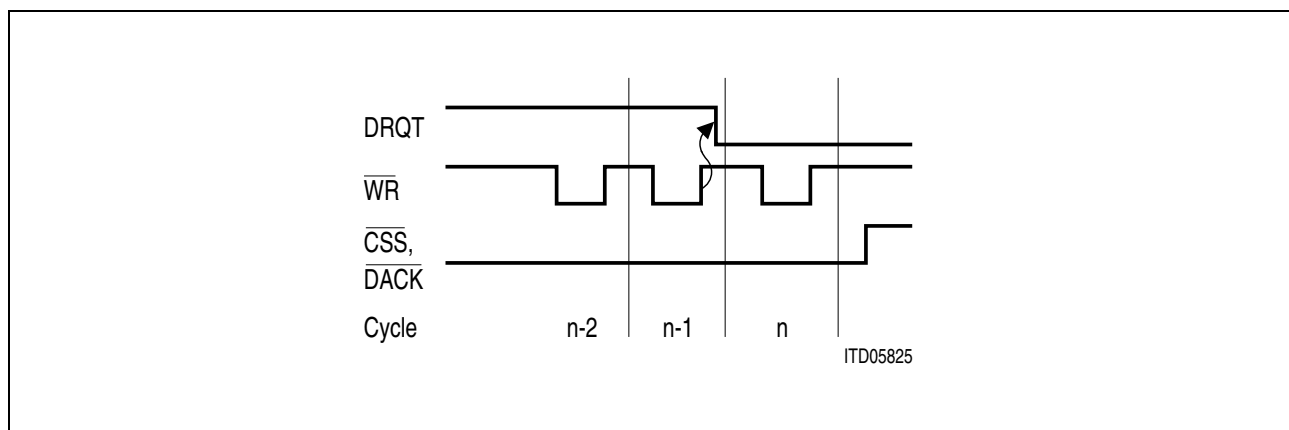
### DMA-Interface

To support efficient data exchange between system memory and the FIFOs an additional DMA-interface is provided. The FIFOs have separate DMA-request lines (DRQRA/B for RFIFO, DRQTA/B for XFIFO) and a common DMA-acknowledge input. The DMA-controller has to operate in the level triggered, demand transfer mode. If the DMA-controller provides a DMA-acknowledge signal, each bus cycle implicitly selects the top of FIFO and neither address nor chip select is evaluated. If no  $\overline{DACK}$  signal is supplied, normal read/write operations (providing addresses) must be performed (memory to memory transfer).

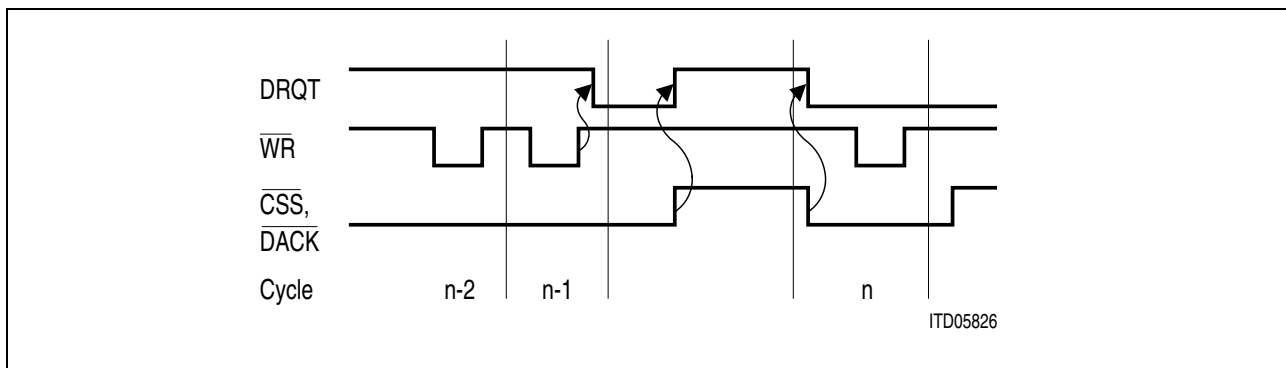
The SACCO activates the DRQT/R-lines as long as data transfers are needed from/to the specific FIFOs.

A special timing scheme is implemented to guarantee safe DMA-transfers regardless of DMA-controller speed.

If in transmit direction a DMA-transfer of  $n$  bytes is necessary ( $n < 32$  or the remainder of a long message), the DRQT-pin is active up to the rising edge of  $\overline{WR}$  of DMA-transfer ( $n-1$ ). If  $n \geq 32$  the same behavior applies additionally to transfers 31, 63, ...,  $((k \times 32) - 1)$ . DRQT is activated again with the next rising edge of  $\overline{DACK}$  (or  $\overline{CSS}$ ), if there are further bytes to transfer (**Figure 2-3**). When a fast DMA-controller is used ( $> 16$  MHz), byte  $n$  (or bytes  $k \times 32$ ) will be transferred before DRQT is deactivated from the SACCO. In this case pin DRQT is not activated any more up to the next block transfer (**Figure 2-2**).



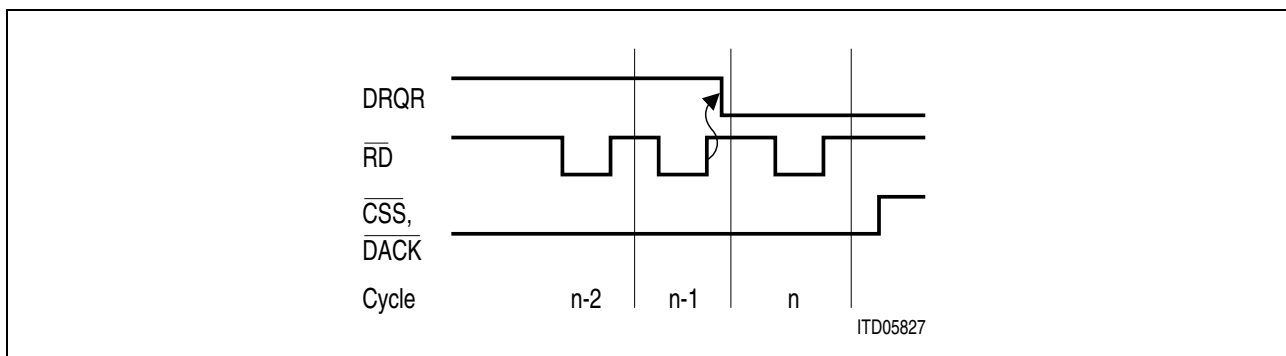
**Figure 2-2 Timing Diagram for DMA-Transfers (fast) Transmit ( $n < 32$ , remainder of a long message or  $n = k \times 32$ )**



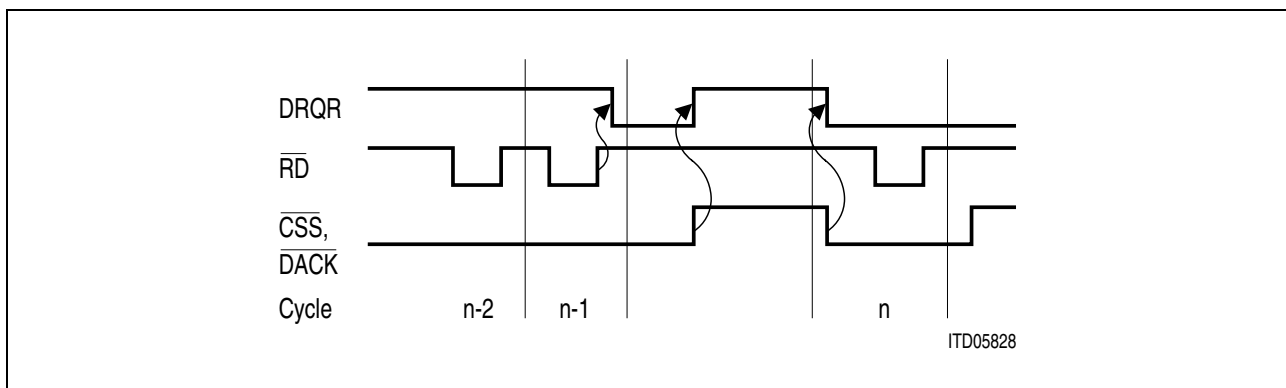
**Figure 2-3 Timing Diagram for DMA-Transfers (slow) Transmit ( $n < 32$ , remainder of a long message or  $n = k \times 32$ )**

In receive direction the behavior of pin DRQR is implemented correspondingly. If  $k \times 32$  bytes are transferred, pin DRQR is deactivated with the rising edge of  $\overline{RD}$  of DMA-transfer ( $(k \times 32) - 1$ ) and it is activated again with the next rising edge of  $\overline{DACK}$  (or  $\overline{CSS}$ ), if there are further bytes to transfer (**Figure 2-5**). When a fast DMA-controller is used ( $> 16$  MHz), byte  $n$  (or bytes  $k \times 32$ ) will be transferred immediately (**Figure 2-4**).

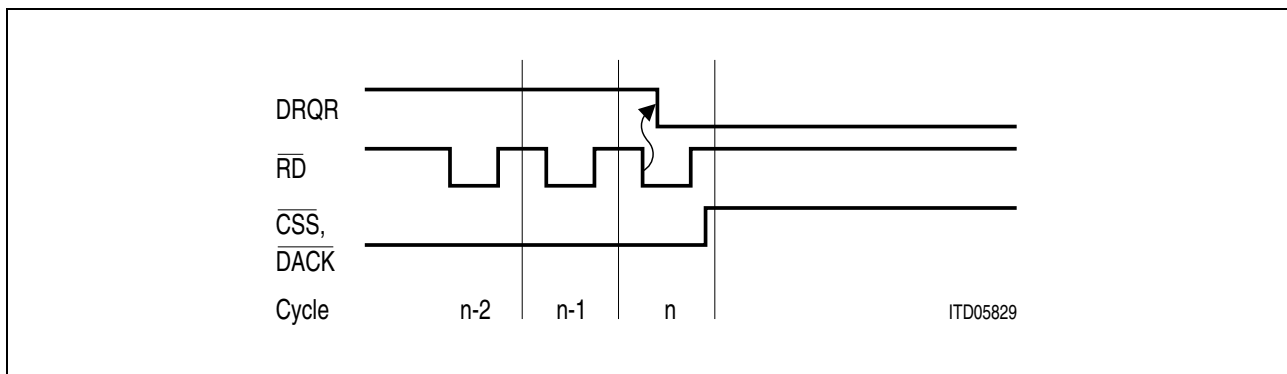
However, if 4, 8, 16 or 32 bytes have to be transferred (only these discrete values are possible in receive direction), DRQR is deactivated with the falling edge of  $\overline{RD}$  (**Figure 2-6**).



**Figure 2-4 Timing Diagram for DMA-Transfer (fast) Receive ( $n = k \times 32$ )**



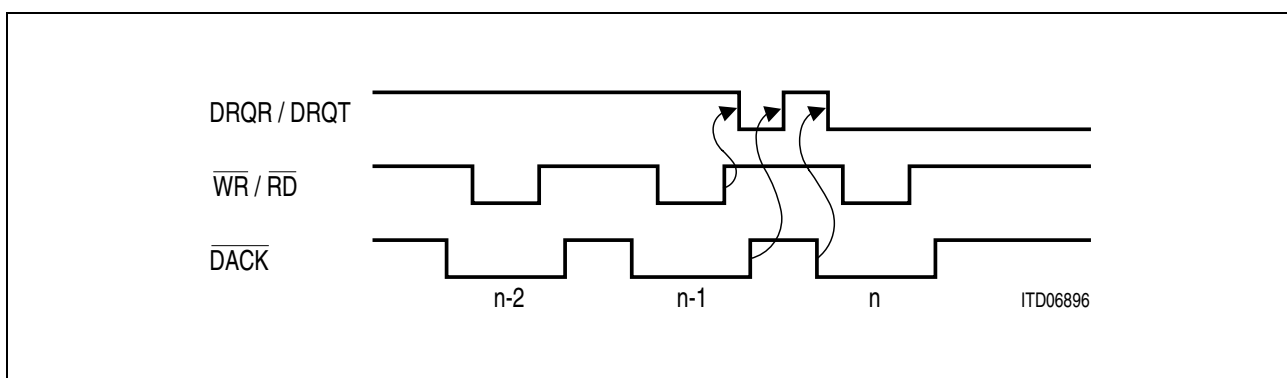
**Figure 2-5 Timing Diagram for DMA-Transfers (slow) Receive ( $n = k \times 32$ )**



**Figure 2-6 Timing Diagram for DMA-Transfers (slow or fast) Receive (n = 4, 8 or 16)**

Generally it is the responsibility of the DMA-controller to perform the correct bus cycles as long as a request line is active.

For further information refer to **chapter 3.1.4.2 (Data Transmission in DMA-Mode)** and **chapter 3.1.4.4 (Data Reception in DMA-Mode)**.



**Figure 2-7 DMA-Transfers with Pulsed DACK (read or write)**

If a pulsed DACK-signal is used the DRQR/DRQT-signal will be deactivated with the rising edge of RD/WR-operation (n – 1) but activated again with the following rising edge of DACK. With the next falling edge of DACK (DACK ‘n’) it will be deactivated again (see **Figure 2-7**).

This behavior might cause a short negative pulse on the DRQR/DRQT-line depending on the timing of DACK vs. RD/WR.

### 2.1.2.4.3 FIFO-Structure

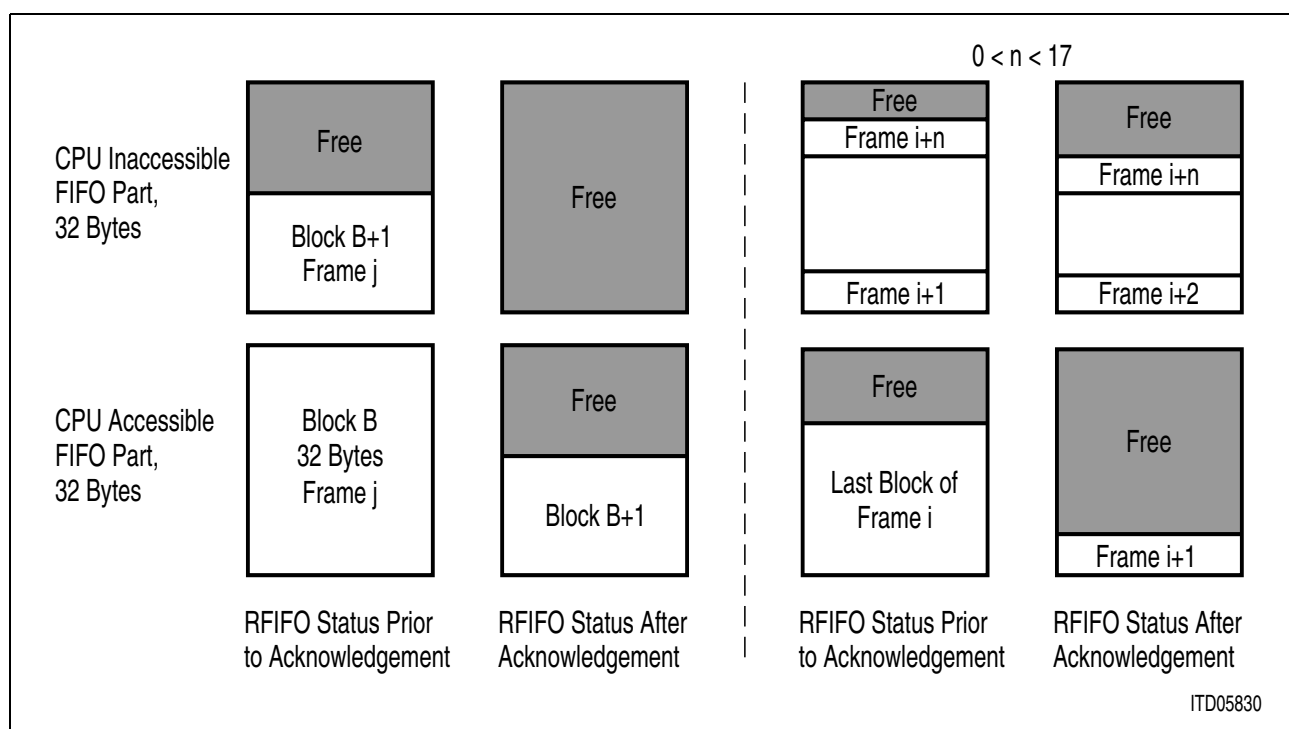
Two independent 64-byte deep FIFOs for transmit and receive direction are provided. They enable an intermediate storage of data between the serial and the parallel (CPU) interface. The FIFOs are divided into two halves of 32 bytes each, where only one half is accessible by the CPU- or DMA-controller.

#### Receive FIFO

The receive FIFO (RFIFO) is organized in two parts of 32 bytes each, of which only one part is accessible for the CPU.

When a frame with **up to 64 bytes** is received, the complete frame may be stored in RFIFO. After the first 32 bytes have been received, the SACCO prompts to read the data block by means of interrupt or DMA-request (RPF-interrupt or activation of DRQR-line).

The data block remains in the RFIFO until a confirmation is given to the SACCO-acknowledging the reception of the data. This confirmation is either a RMC- (Receive Message Complete) command in interrupt mode or it is implicitly achieved in DMA-mode after 32 bytes have been read. As a result it is possible in interrupt mode to read out the data block any number of times until the RMC-command is executed. Upon the confirmation the second data block is shifted into the accessible RFIFO-part and an RME-interrupt is generated. The configuration of the RFIFO prior to and after acknowledgment is shown in **Figure 2-8 (left)**. If frames longer than 64 bytes are received, the SACCO will repeatedly prompt to read out 32-byte data blocks via interrupt or DMA.



**Figure 2-8 Frame Storage in RFIFO (single frame / multiple frames)**



## Functional Block Description

In the case of **several shorter frames**, up to 17 frames may be stored in the RFIFO. Nevertheless, only one frame is stored in the CPU accessible part of the RFIFO. E.g., if frame *i* (or the last part of frame *i*) is stored in the accessible RFIFO-part, up to 16 short frames may be stored in the other half (*i* + 1, *i* + 2, ..., *i* + *n*, *n* ≤ 16). This behavior is illustrated in **Figure 2-8 (right)**.

*Note: After every frame a receive status byte is appended, specifying the status of the frame (e.g. if the CRC-check is o.k.).*

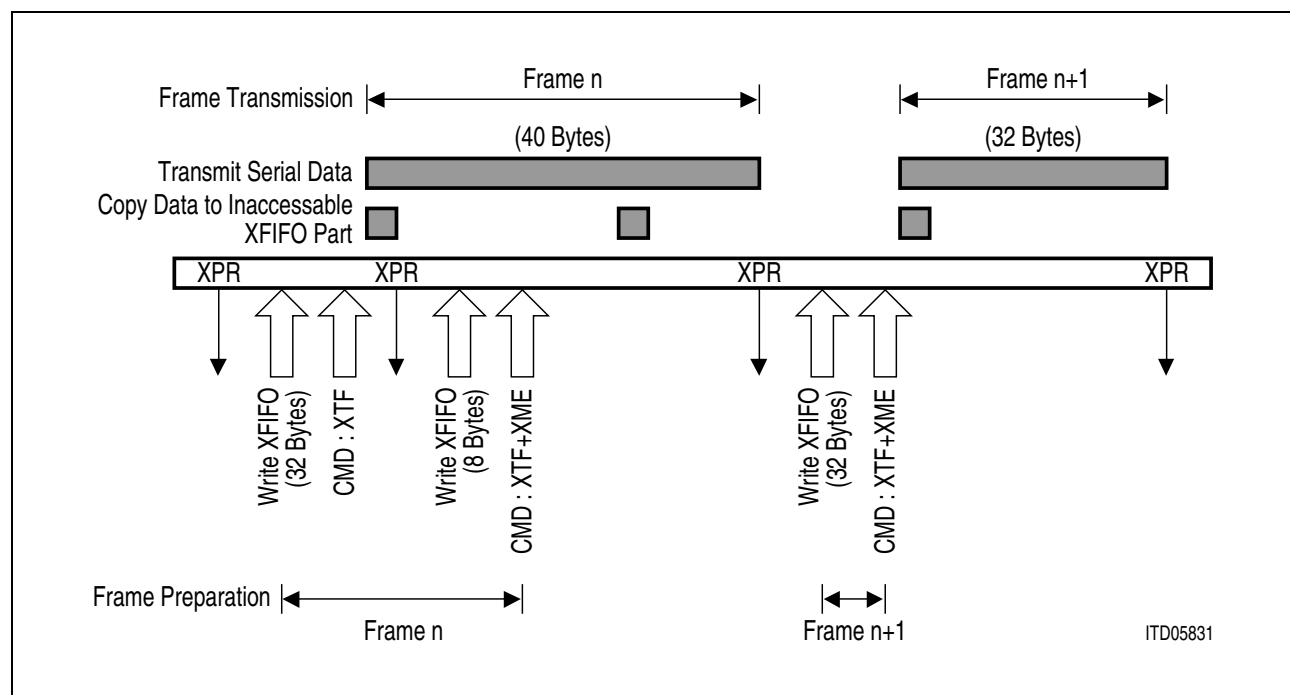
When using the DMA-mode, the SACCO requests fixed size block transfers (4, 8, 16 or 32 bytes). The valid byte count is determined by reading the registers RBCH, RBCL following the RME-interrupt.

### Transmit FIFO

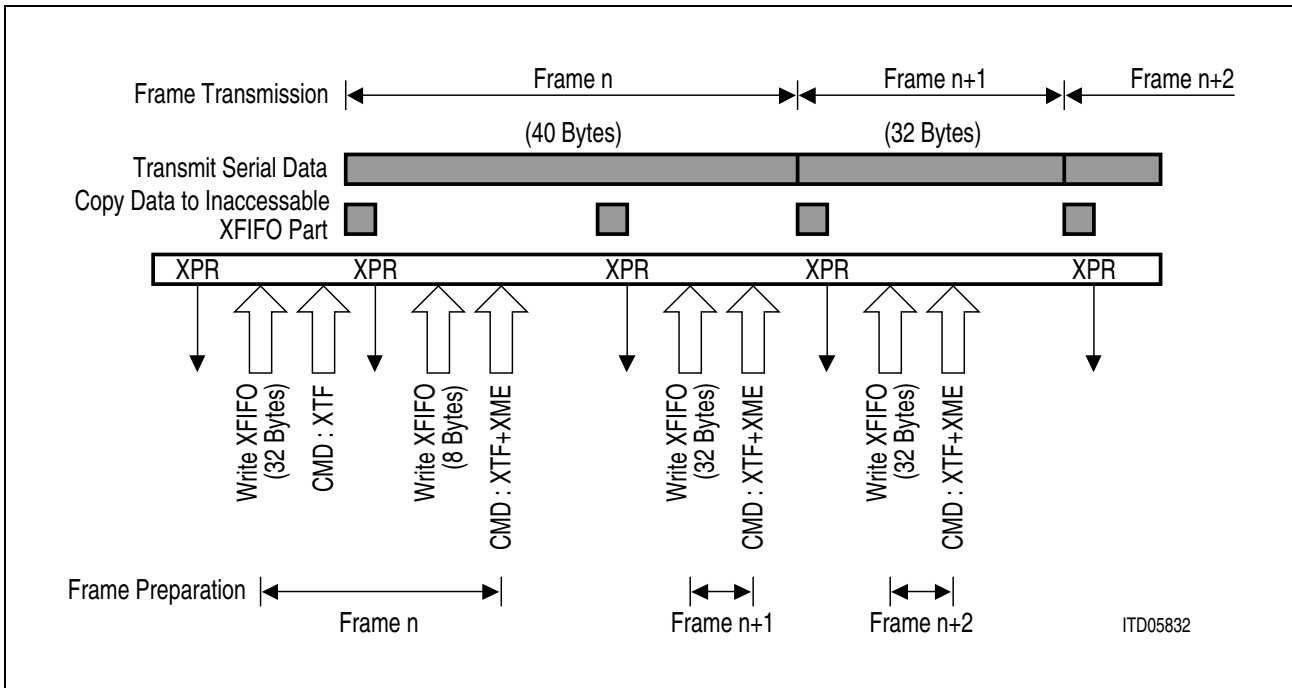
The transmit FIFO (XFIFO) provides a 2 × 32 bytes capability to intermediately store transmit data.

In interrupt mode the user loads the data and then executes a transmit command. When the frames are longer than 32 bytes, a XPR-interrupt is issued as soon as the accessible XFIFO-part is available again.

The status of the bit MODE:CFT (continuous frame transmission) defines whether a new frame can be loaded as soon as the XFIFO is available or after the current transmission was terminated.



**Figure 2-9 XFIFO Loading, Continuous Frame Transmission Disabled (CFT = 0)**



**Figure 2-10 XFIFO Loading, Continuous Frame Transmission Enabled (CFT = 1)**

When using the DMA-mode, prior to the data transfer the actual byte count to be transmitted must be written to the registers XBCH, XBCL (transmit byte count high, low). If the data transfer is initiated via the proper command, the SACCO automatically requests the correct amount of block data transfers ( $n \times 32 + \text{remainder}$ ,  $n = 0, 1, 2, \dots$ ) by activating the DRQT-line.

Refer to **chapter 2.1.2.4.2** for a detailed description of the DMA transfer timing.

#### 2.1.2.4.4 Protocol Support

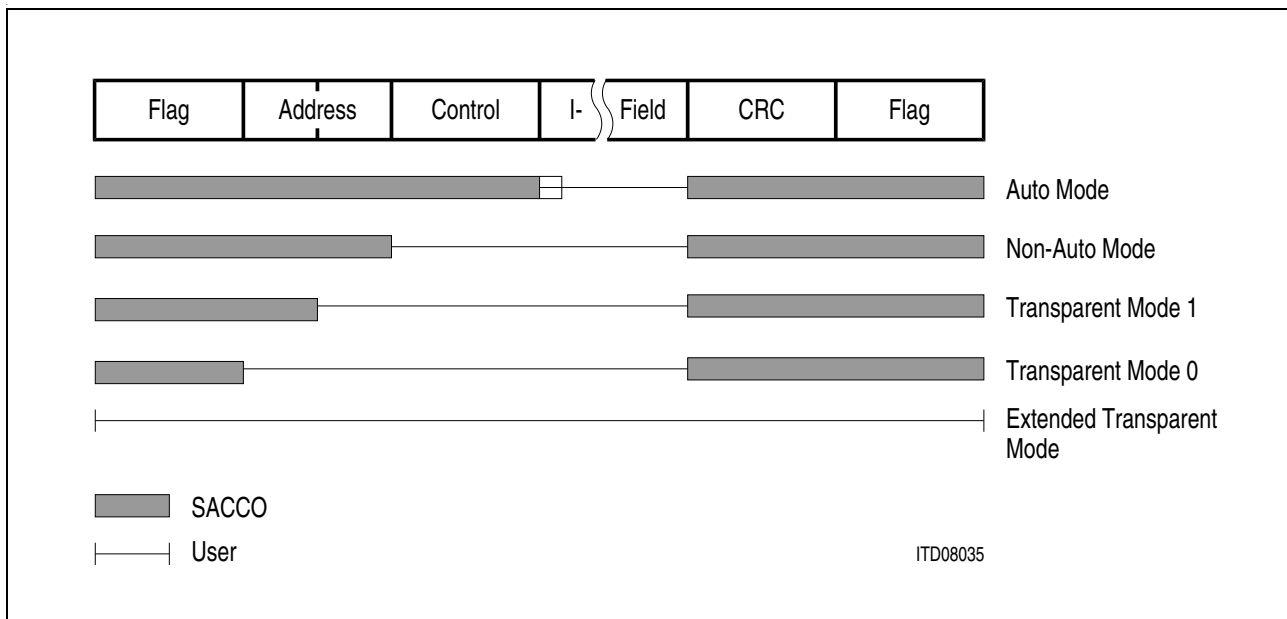
The SACCO supports the following fundamental HDLC functions:

- Flag insertion/deletion,
- Bit stuffing,
- CRC-generation and checking,
- Address recognition.

Further more it provides six different operating modes, which can be set via the MODE register. These are:

- Auto Mode,
- Non-Auto Mode,
- Transparent Mode 0 and 1,
- Extended Transparent Mode 0 and 1.

These modes provide different levels of HDLC processing.



**Figure 2-11 Support of the HDLC Protocol by the SACCO**

### Address Recognition

Address recognition is performed in three operating modes (auto-mode, non-auto-mode and transparent mode 1). Two pairs of compare registers (RAH1, RAH2: high byte compare, RAL1, RAL2: low byte compare) are provided. RAL2 may be used for a broadcast address. In auto-mode and non-auto-mode 1- or 2-byte address fields are supported, transparent mode 1 is restricted on high byte recognition. The high byte address is additionally compared with the LAPD group address (FC<sub>H</sub>, FE<sub>H</sub>).

**Functional Block Description**

Depending on the operating mode the following combinations are considered valid addresses:

**Table 2-4 Address Recognition**

<b>Operating Mode</b>	<b>Compare Value High Byte</b>	<b>Compare Value Low Byte</b>	<b>Activity</b>
Auto-mode, 2-byte address field	<RAH1>	<RAL1>	Processed, following the auto-mode protocol
	<RAH2>	<RAL1>	
	FCH	<RAL1>	
	FEH	<RAL1>	
	<RAH1>	<RAL2>	Frame is stored transparently in RFIFO
	<RAH2>	<RAL2>	
	FCH	<RAL2>	
	FEH	<RAL2>	
Auto-mode, 1-byte address field	–	<RAL1>	Processed, following the auto-mode protocol
	–	<RAL2>	Frame is stored transparently in RFIFO
Non-auto mode, 2-byte address field	<RAH1>	<RAL1>	Frame is stored transparently in RFIFO
	<RAH2>	<RAL1>	
	FCH	<RAL1>	
	FEH	<RAL1>	
	<RAH1>	<RAHL2>	
	<RAH2>	<RAL2>	
	FCH	<RAL2>	
	FEH	<RAL2>	
Non-auto mode, 1-byte address field	–	<RAL1>	Frame is stored transparently in RFIFO
	–	<RAL2>	
Transparent mode 1	<RAH1>	–	Frame is stored transparently in RFIFO
	<RAH2>	–	
	FCH	–	
	FEH	–	

**Functional Block Description**

**Auto-Mode (MODE:MDS1,MDS0 = 00)**

Characteristics: HDLC formatted, NRM-type protocol, 1-byte/2-byte address field, address recognition, any message length, automatic response generation for RR- and I-frames, window size 1.

The auto-mode is optimized to communicate with a group controller following a NRM-(Normal Response Mode) type protocol. Its functionality guarantees a minimum response time and avoids the interruption of the CPU in many cases.

The SACCO auto-mode is compatible to a PEB 2050 (PBC) behavior in secondary mode.

Following the PBC-conventions, two data types are supported in auto-mode.

**Table 2-5 Auto-Mode Data Types**

<b>Data Types</b>	<b>Meaning</b>
Direct data	Data exchanged in normal operation mode between the local $\mu$ P and the group controller, typically signaling data.
Prepared data	Data request by or send to the group controller for maintenance purposes.

*Note: In many applications only direct data is used, nevertheless both data types are supported because of compatibility reasons.*

**Receive Direction**

In auto-mode the SACCO provides address recognition for 2- and 1-byte address fields. The auto-mode protocol is only applied when RAL1 respectively RAH1/RAL1 match. With any other matching combination, the frame is transferred transparently into the RFIFO and an interrupt (RPF or RME) is issued.

If no address match occurs, the frame is skipped. The auto-mode protocol processes RR- and I-frames automatically. On the reception of any other frame type an EHC-interrupt (extended HDLC frame) is generated. No data is stored in the RFIFO but due to the internal hardware structure the HDLC-control field is temporarily stored in register RHCR. In the PBC-protocol an extended HDLC-frame does not contain any data.

**Table 2-6 HDLC-Control Field in Auto-Mode**

<b>HDLC-Control Byte</b>	<b>Frame Type</b>
xxxP xxx0	I-frame
xxxP xx01	RR-frame
xxxx xx11	Extended HDLC-frame

### RR-Frames

RR-frames are processed automatically and are not stored in RFIFO.

When a RR-frame with poll bit set (control field = xxx10001) is received, it is interpreted as a request to transmit direct data.

Depending on the status of the XFIFO an I-frame (data available) or a RR-response (no data available) is issued.

This behavior guarantees minimum response times and supports a fast cyclical polling of signaling data in a point-to-multi-point configuration.

A RR-frame with poll bit = 0 is interpreted as an acknowledgment for a previously transmitted I-frame: the XFIFO is cleared, a XPR interrupt is emitted, no response is generated.

The polling of a frame can be repeated an unlimited number of times until the frame is acknowledged. Depending on the status of the bit MODE:AREP (auto repeat), the transmission is repeated without or with the intervention of the CPU (XMR interrupt). The auto repeat mode must not be selected, when the frame length exceeds 32 bytes. In DMA mode, when using the auto repeat mode, the control response will not be compatible to the PBC.

### I-Frames

When an I-frame is received in auto-mode the first data byte is interpreted as a command byte according to the PEB 2050 (PBC) protocol.

Depending on the value of the command byte one of the following actions is performed.

**Table 2-7 Auto-Mode Command Byte Interpretation**

Command Byte = 1. Data Byte	Stored in RFIFO	Interrupt	Additional Activities	Condition
00 - 9F <sub>H</sub> B0 - CF <sub>H</sub> F0 - FF <sub>H</sub>	yes	RPF, RME	Response generation when poll bit set	–
A0-AF <sub>H</sub>	no	no	Response generation when poll bit set I-frame with XFIFO-Data	– Command XPD executed
D0 - EF <sub>H</sub>	no	XPR	Response generation when poll bit set, reset XFIFO	Command XPD executed
	no	no	Response generation when poll bit set	Command XPD not executed

**Functional Block Description**

When a I-frame is stored in RFIFO the command byte has to be interpreted by software. Depending on the subset of PBC commands used in the individual application, the implementation may be limited to the necessary functions. In case XPD is executed (with or without data in XFIFO) the SACCO will generate an XPR interrupt upon the reception of a command  $D0_H, \dots, EF_H$ , even if the data has not been polled previously.

*Note: In auto-mode I-frames with wrong CRC or aborted frames are stored in RFIFO. In the attached RSTA-byte the CRC and RAB-bits are set accordingly to indicate this situation. In these cases no response is generated.*

**Transmit Direction, Response Generation**

In auto-mode frames are only transmitted after the reception of a RR- or I-frame with poll bit set.

**Table 2-8 Auto-Mode Response Generation**

Received Frame	Response	Condition
RR-poll poll bit set	I-frame with XFIFO-data	Command XDD executed
	RR-response	Command XDD not executed
I-frame, first byte = AxH poll bit set	I-frame with XFIFO-data	Command XPD executed
	I-frame, data byte = control response	Command XPD not executed
I-frame, first data byte not AxH, poll bit set	I-frame, data byte = control response	

**RR-Response**

The RR-response is generated automatically. It has the following structure.

flag	address	control byte	CRC-word	flag
------	---------	--------------	----------	------

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to  $11_H$  (RR-frame, final bit = 1).

**Control Response**

The control response is generated automatically. It has the following structure.

flag	address	control byte	control resp.	CRC-word	flag
------	---------	--------------	---------------	----------	------

**Functional Block Description**

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to 10<sub>H</sub>.

According to the PBC conventions, the control response byte has the following structure:

bit 7				bit 0			
1	0	1	AREP	0	0	DOV	1

- bit7...6 : 10 : response to an I-frame, no further data follows
- bit5 : 1 : μP connected (PBC operates optionally in stand alone mode)
- bit4 : AREP : 1/0: autorepeating is enabled/disabled  
(Read back value of CMDR:AREP)
- bit3...2 : 00 : SACCO FIFO available for data reception
- bit1 : DOV : inverted status of the bit RSTA:RDO (RFIFO overflow)
- bit0 : 1 : fixed value, no functionality.

**I-Frame with Data**

flag	address	control byte	data	CRC-word	flag
------	---------	--------------	------	----------	------

The address is defined by the value stored in XAD1 (1-byte address) or XAD1 and XAD2 (2-byte address). The control byte is fixed to 10<sub>H</sub> (I-frame, final bit = 1). The data field contains the XFIFO contents.

*Note: The control response byte has to be generated by software.*

**Data Transfer**

**Polling of Direct Data**

When direct data was loaded (XDD executed) an I-frame is generated as a response to a RR-poll.

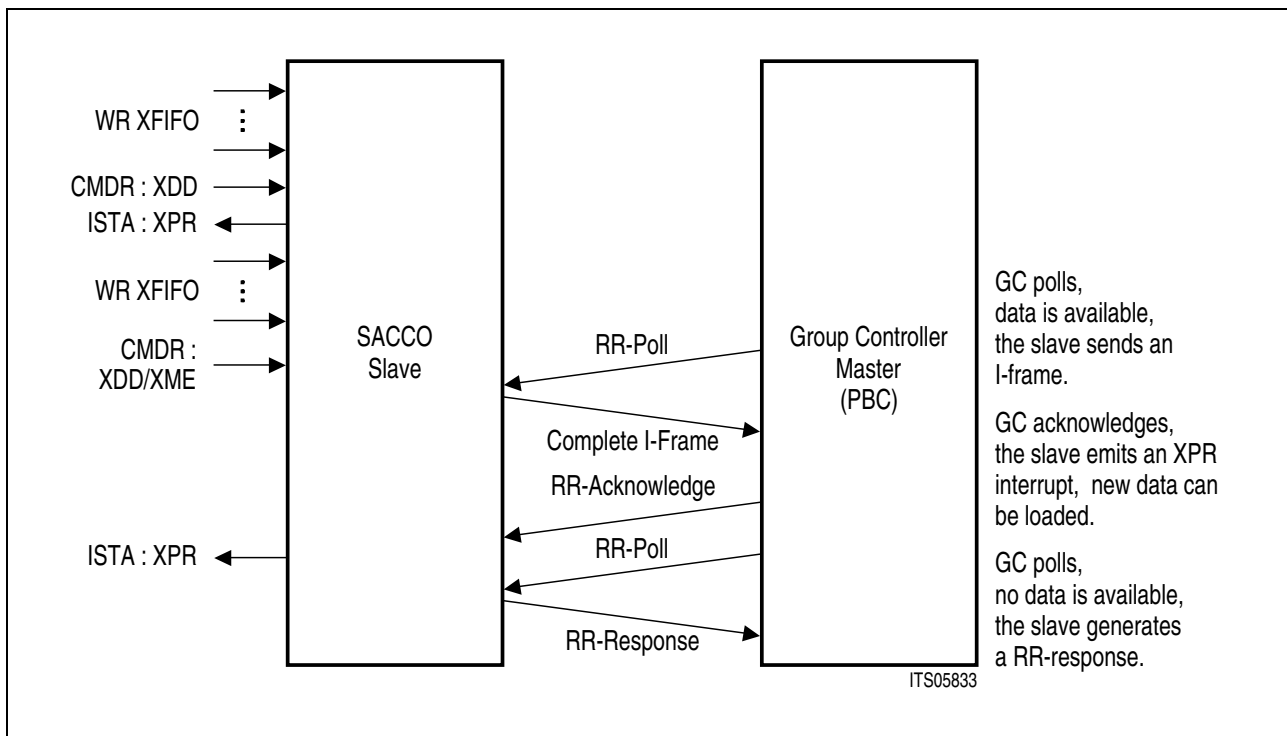
After checking STAR:XFW, blocks of up to 32 bytes may be entered in XFIFO. When more than 32 bytes are to be transmitted the XPR-interrupt is used to indicate that the CPU accessible XFIFO-part is free again. A maximum of 64 bytes may be stored before the actual transmission is started.

A RR-acknowledge (poll bit = 0) causes an ISTA:XPR interrupt, XFIFO is cleared and STAR:XFW is set.

When the SACCO receives a RR-poll frame and no data was loaded in XFIFO it generates automatically a RR-response.

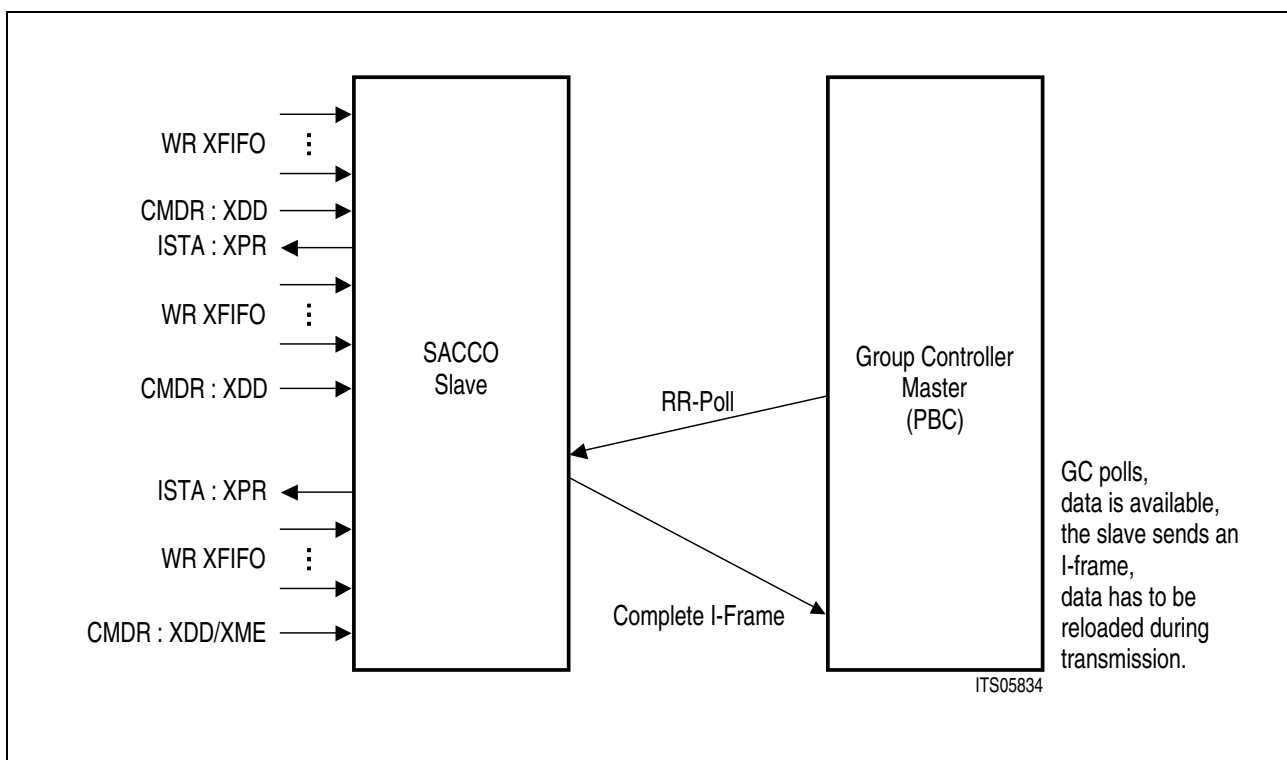


Functional Block Description



**Figure 2-12 Polling of up to 64 Bytes Direct Data**

If more than 64 bytes are transmitted, the XFIFO is used as an intermediate buffer. Data has to be reloaded after transmission was started.



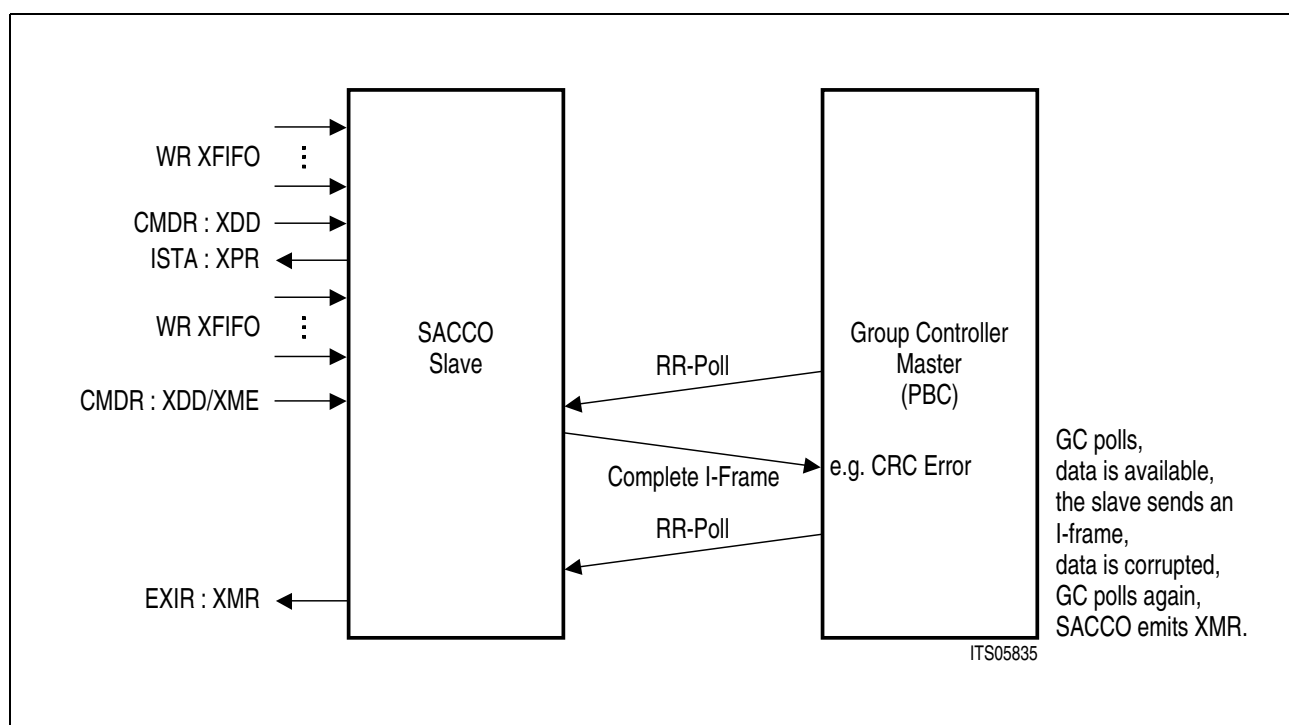
**Figure 2-13 Polling More than 64 Bytes of Direct Data (e.g. 96 bytes)**

### Functional Block Description

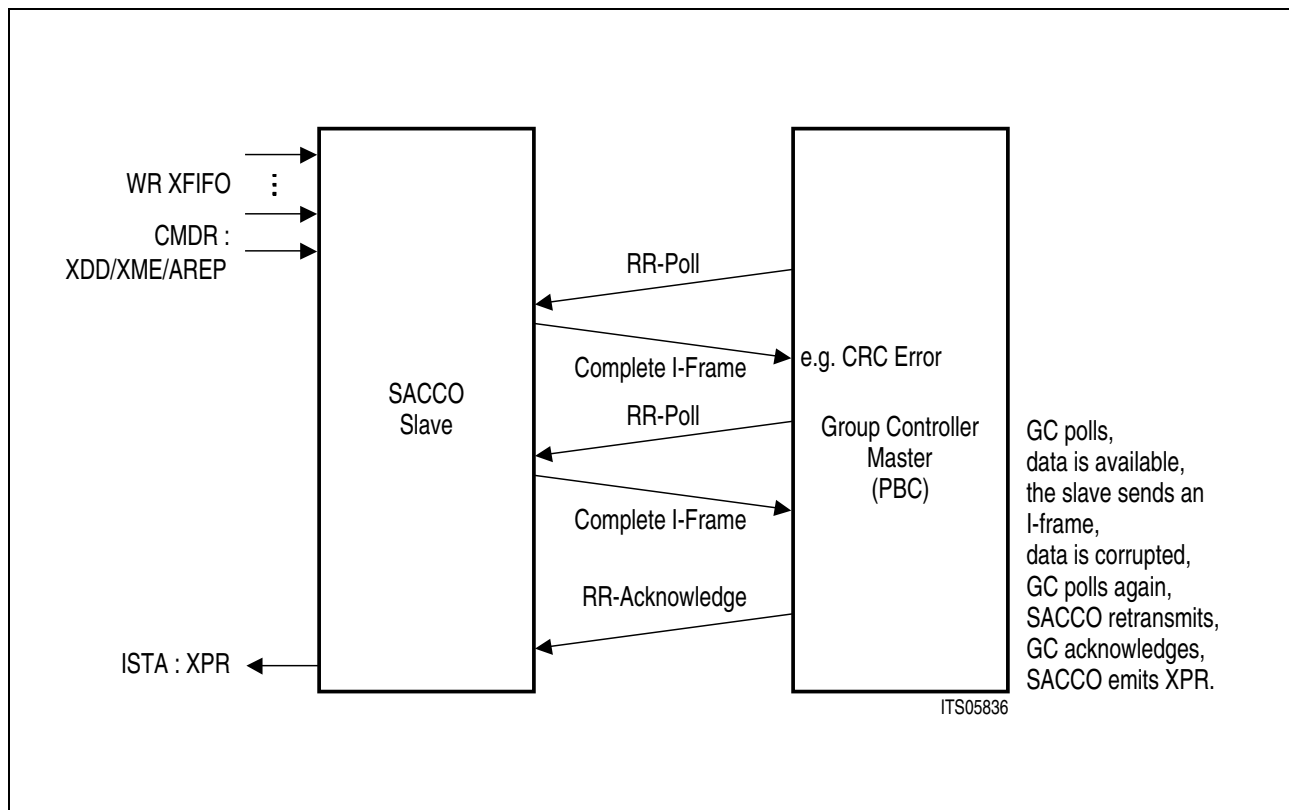
When the group controller wants the SACCO to re-transmit a frame (e.g. due to a CRC-error) it does not answer with a RR-acknowledge but emits a second RR-poll.

The SACCO then generates an XMR-interrupt (transmit message repeat) indicating the CPU that the previously transmitted frame has to be loaded again. For frames which are not longer then 32 bytes the SACCO offers an auto repeat function allowing the automatic re-transmission of a frame without interrupting the CPU.

*Note: For frames which are longer than 32 bytes the auto repeat function must not be used.*



**Figure 2-14 Re-Transmission of a Frame**



**Figure 2-15 Re-Transmission of a Frame with Auto-Repeat Function**

### Polling of Prepared Data

If polling “prepared data” a different procedure is used. The group controller issues an I-frame with a set poll bit and the first data byte is interpreted as command byte.

When prepared data was loaded into the XFIFO (CMDR:XPD/XME was set) the reception of a command byte equal to AxH initiates the transmission of an I-frame.

For “prepared data” the auto repeat function must be selected! Due to this the polling can be repeated without interrupting the CPU.

An I-frame with a data byte equal to D0H-EFH is interpreted as an acknowledgment for previously transmitted data. An XPR-interrupt is issued and the XFIFO is reset.

All other I-frames are stored in the RFIFO and a RME-interrupt is generated. The local  $\mu$ P can read and interpret the received data (e.g. following the PBC-protocol). A PBC compatible control response is generated automatically.

E.g., if the local  $\mu$ P recognizes the request to “prepare data” it may load the XFIFO and set CMDR:XPD/XME.

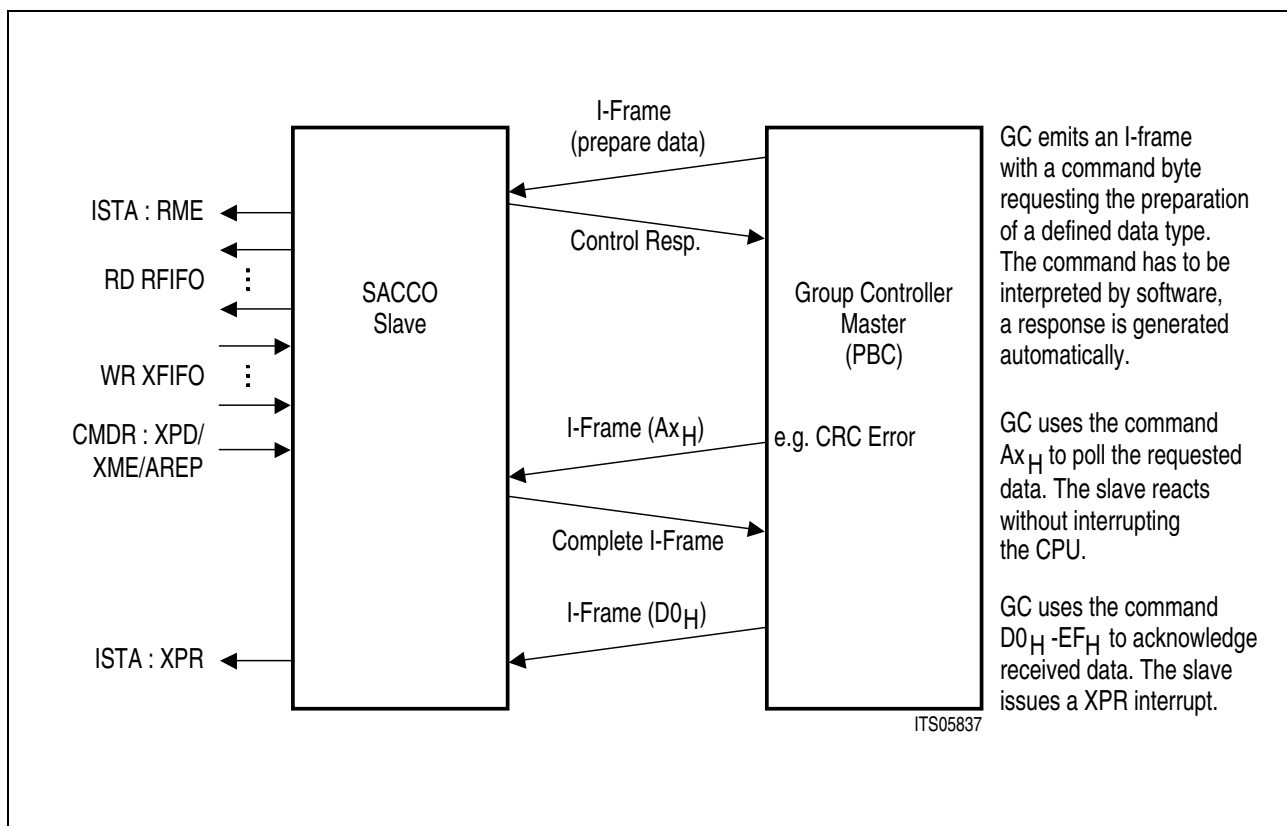


Figure 2-16 Polling of Prepared Data

### Behavior of SACC0 when a RFIFO Overflow Occurs in Auto-Mode

When the RFIFO overflows during the reception of an I-frame, a control response with overflow indication is transmitted, the overflow information is stored in the corresponding receive status byte. When additional poll frames are received while the RFIFO is still occupied, an RFO (receive frame overflow) interrupt is generated. Depending on the type of the received poll frame different responses are generated:

- I-frame – control response with overflow indication  
(exception: when the command “transmit prepared data” ( $Ax_H$ ) is received and prepared data is available in the XFIFO, an I-frame (with data) is issued)
- RR-poll – RR-response, when no direct data was stored in the XFIFO  
– I-frame, when direct data was stored in the XFIFO

**Functional Block Description**

Depending on the number of bytes to be stored in the RFIFO the following behavior occurs:

**Table 2-9**

<b>RFIFO Handling/Steps</b>	<b>Case 1</b>	<b>Case 2</b>	<b>Case 3</b>
Receive frame	Total frame length: 63 data bytes	Total frame length: 64 data bytes	Total frame length: 65 data bytes or more
After 32 bytes are received	A RPF-interrupt is issued, the RFIFO is not acknowledged		
After next 31/32 bytes are received	Control response, no overflow indication	Control response with overflow indication	Control response with overflow indication
Additional I-poll	RFO-interrupt, I-response with overflow indication or I-data if stored in XFIFO as prepared data		
Additional RR-poll	RFO-interrupt, RR-response or I-data if stored in XFIFO as direct data		
Read and acknowledge RFIFO	RME-interrupt	RPF-interrupt	RPF-interrupt
Read and acknowledge RFIFO	RDO-bit is not set, frame is complete	RME-interrupt	RME-interrupt
Read and acknowledge RFIFO		<b>RDO-bit is set, frame is complete but indicated as incomplete</b>	RDO-bit is set, frame is not complete

Multiple shorter frames results in the same behavior, e.g.

frame 1: 1-31 bytes

frame 2 –n: total of 31 bytes including receive status bytes for frame 2 –(n –1) cause the case 1.

**Non-Auto-Mode (MODE:MDS1, MDS0 = 01)**

Characteristics: HDLC formatted, 1-byte/2-byte address field, address recognition, any message length, any window size.

All frames with valid address fields are stored in the RFIFO and an interrupt (RPF, RME) is issued.

The HDLC-control field, data in the I-field and an additional status byte are stored in RFIFO. The HDLC-control field and the status byte can also be read from the registers RHCR, RSTA (currently received frame only!).

According to the selected address mode, the SACCO can perform 2-byte or 1-byte address recognition.

**Transparent Mode 1 (MODE:MDS1, MDS0, ADM = 101)**

Characteristics: HDLC formatted, high byte address recognition, any message length, any window size.

Only the high byte address field is compared with RAH1, RAH2 and the group address ( $FC_H$ ,  $FE_H$ ). The whole frame except the first address byte is stored in RFIFO. RAL1 contains the second and RHCR the third byte following the opening flag (currently received frame only). When using LAPD the high byte address recognition feature can be used to restrict the frame reception to the selected SAPI-type.

**Transparent Mode 0 (MODE:MDS1, MDS0, ADM = 100)**

Characteristics: HDLC formatted, no address recognition, any message length, any window size.

No address recognition is performed and each frame is stored in the RFIFO. RAL1 contains the first and RHCR the second byte following the opening flag (currently received frame only).

*Note: In non-auto-mode and transparent mode I-frames with wrong CRC or aborted frames are stored in RFIFO. In the attached RSTA-byte the CRC and RAB-bits are set accordingly to indicate this situation.*

**Extended Transparent Mode 0 (MODE:MDS1, MDS0, ADM = 110)**

Characteristics: fully transparent without HDLC framing, any message length, any window size.

Data is stored in register RAL1.

In extended transparent mode, fully transparent data transmission/reception without HDLC-framing is performed, i.e. without FLAG-generation/recognition, CRC-generation/check, bit stuffing mechanism. This allows user specific protocol variations or can be used for test purposes (e.g. to generate frames with wrong CRC-words).

Data transmission is always performed out of the XFIFO. Data reception is done via register RAL1, which contains the actual data byte assembled at the RxD pin.

**Extended Transparent Mode 1 (MODE:MDS1, MDS0, ADM = 111)**

Characteristics: fully transparent without HDLC-framing, any message length, any window size. Data is stored in register RAL1 and RFIFO.

Identical behavior as extended transparent mode 0 but the received data is shifted additionally into the RFIFO.

**Receive Data Flow (summary)**

The following figure gives an overview of the management of the received HDLC-frames depending on the selected operating mode.

Functional Block Description

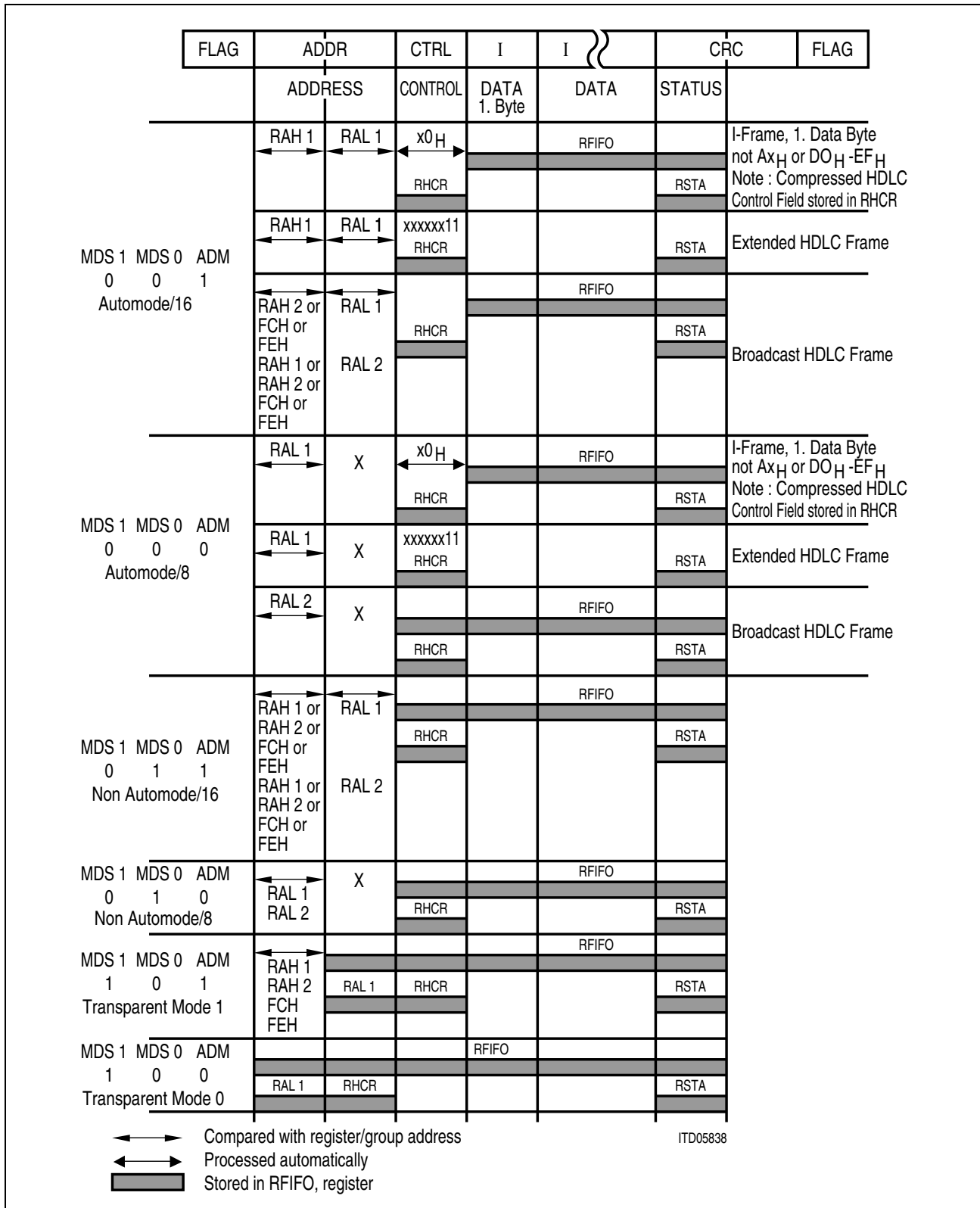


Figure 2-17 Receive Data Flow

Note: RR-frames and I-frame with first data byte equal to Ax<sub>H</sub> or DO<sub>H</sub>-EF<sub>H</sub> are processed automatically. They are not stored in RFIFO and no interrupt is issued.



#### 2.1.2.4.5 Special Functions

##### Cyclical Transmission (fully transparent)

When the extended transparent mode is selected, the SACCO supports the continuous transmission of the XFIFO-contents.

After having written 1 to 32 bytes to the XFIFO, the command XREP/XTF/XME (XREP/XTF in DMA-mode) is executed. Consequently the SACCO repeatedly transmits the XFIFO-data via pin TxD.

The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.

When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

*Note: Bit MODE:CFT must be set to '0'.*

##### Continuous Transmission (DMA-mode only)

If data transfer from system memory to the SACCO is done by DMA (DMA bit in XBCH set), the number of bytes to be transmitted is usually defined via the transmit byte count registers XBCH, XBCL. Setting the "transmit continuously" bit (XC) in XBCH, however, the byte count value is ignored and the DMA-interface of the SACCO will continuously request for transmit data any time 32 bytes can be stored in the XFIFO.

This feature can be used e.g. to

- continuously transmit voice or data onto a PCM-highway (clock mode 2, ext. transp. mode)
- transmit frames exceeding the byte count programmable in XBCH, XBCL (> 4095 bytes).

*Note: If the XC-bit is reset during continuous transmission, the transmit byte count becomes valid again, and the SACCO will request the amount of DMA-transfers programmed in XBC11...XBC0. Otherwise the continuous transmission is stopped when a data underrun condition occurs in the XFIFO, i.e. the DMA-controller does not transfer further data to the SACCO. In this case an abort sequence (min. 7 '1's) followed by the inter frame timefill pattern is transmitted (no CRC-word is appended).*

##### Receive Length Check

The SACCO offers the possibility to supervise the maximum length of received frames and to terminate data reception in case this length is exceeded.

This feature is enabled by setting the RC- (receive check) bit in RLCR and programming the maximum frame length via bits RL6...RL0.

## Functional Block Description

According to the value written to RL6...RL0, the maximum receive length can be adjusted in multiples of 32-byte blocks as follows:  $\text{max. frame length} = (\text{RL} + 1) \times 32$ .

All frames exceeding this length are treated as if they have been aborted from the opposite station, i.e. the CPU is informed via a

- RME-interrupt, and the
- RAB-bit in RSTA register is set (clock mode 0-2)

To distinguish between frames really aborted from the opposite station, the receive byte count (readable from registers RBCH, RBCL) exceeds the maximum receive length (via RL6...RL0) by one or two bytes in this case.

### 2.1.2.4.6 Serial Interface

#### Clock Modes

The SACCO uses a single clock for transmit and receive direction. Three different clock modes are provided to adapt the serial interface to different requirements.

#### Clock Mode 0

Serial data is transferred on RxD/TxD, an external generated clock (double or single data rate) is forwarded via pin HDC.

#### Clock Mode 1

Serial data is transferred on RxD/TxD, an external generated clock (double or single data rate) is forwarded via pin HDC. Additionally a receive/transmit strobe provided on pin HFS is evaluated.

#### Clock Mode 2

This operation mode has been designed for applications in time-slot oriented PCM-systems. The SACCO receives and transmits only during a certain time-slot of programmable width (1...256 bits) and location with respect to a frame synchronization signal, which must be delivered via pin HFS.

The position of the time-slot can be determined applying the formula in **Figure 2-18**.

**TSN** Defines the number of 8 bit time-slots between the start of the frame (HFS edge) and the beginning of the time-slot for the HDLC channel. The values for TSN are written to the registers TSAR:7..2 and TSN:7..2.

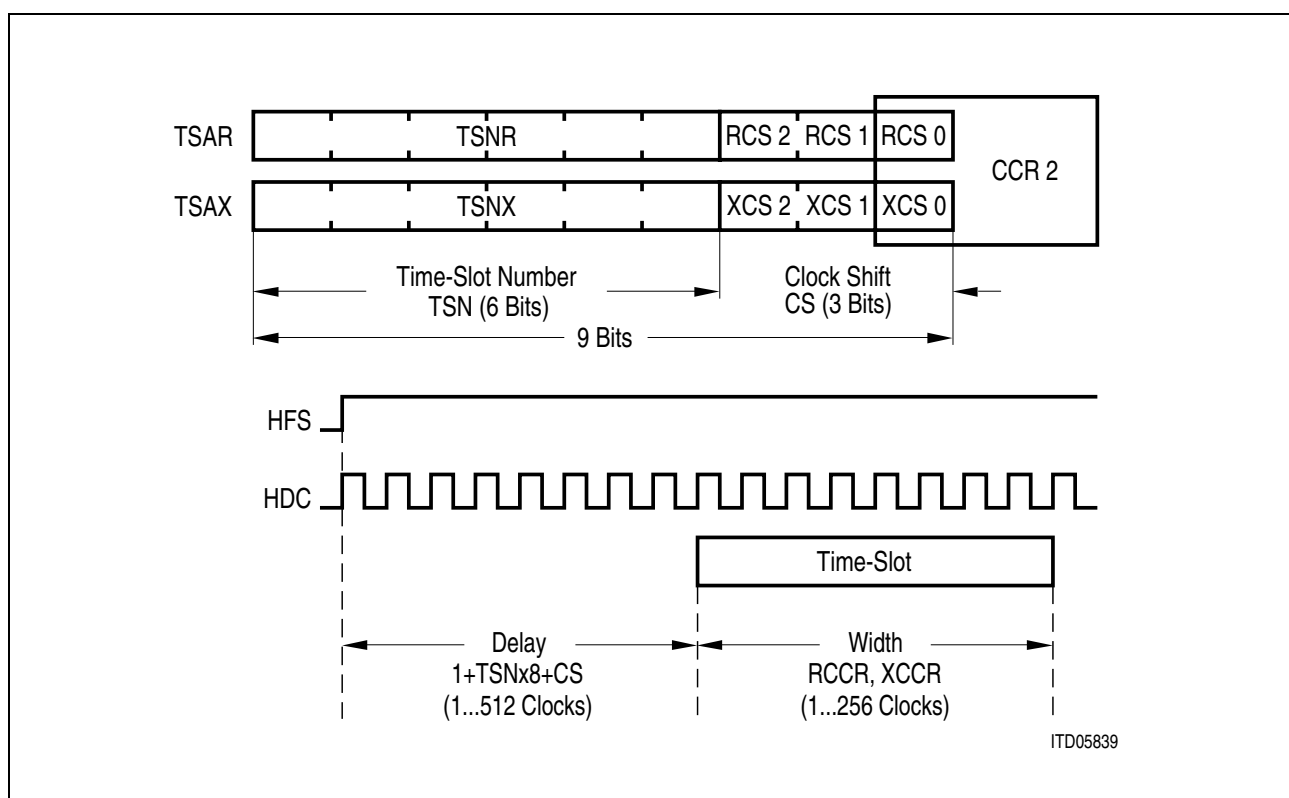
**CS** Additionally a clock shift of 0..7 bits can be defined using register bits TSAR:RSC2...1, TSAX:XCS2..1 and CCR2:XCS0, CCR2:RCS0.

Together TSN and CS provide 9 bits to determine the location of the time-slot for the HDLC channel.

## Functional Block Description

One of up to 64 time-slots can be programmed independently for receive and transmit direction via the registers TSAR and TSAX.

According to the value programmed via those bits, the receive/transmit window (time-slot) starts with a delay of 1 (minimum delay) up to 512 clock periods following the frame synchronization signal and is active during the number of clock periods programmed via RCCR, XCCR (number of bits to be received/transmitted within a time-slot) as shown in **Figure 2-18**.



**Figure 2-18 Location of Time-Slots**

*Note: In extended transparent mode the width of the time-slot has to be  $n \times 8$  bit.*

### Clock Mode 3

In clock mode 3 SACCO-A is multiplexed among multiple subscribers under the control of the D-channel arbiter. It must be used only in combination with transparent mode 0.

Serial data is transferred on (received from) the D-channels of the EPIC-1 IOM-2 interfaces. The data clock is derived from DCL. The D-channel arbiter generates the receive and transmit strobes.

When bit CCR2:TXDE is set, the transmitted D-channel data can additionally be monitored on pin TxDA delayed by 1 bit. The timing is identical to clock mode 1 assuming a transmit strobe during the transmission of the third and fourth bit following the rising FSC-edge.

### Receive Status Byte in Clock Mode 3

In clock mode 3 the receive status byte is modified when it is copied into RFIFO. It contains the following information:

bit 7								bit 0
VFR	RDO	CRC	CHAD4	CHAD3	CHAD2	CHAD1	CHAD0	

**VFR** Valid Frame.  
 Indicates whether the received frame is valid ('1') or not ('0' invalid).  
 A frame is invalid when

- its length is not an integer multiple of 8 bits ( $n \times 8$  bits), e.g. 25 bit,
- it is too short, depending on the selected operation mode (transparent mode 0: 2 bytes minimum),

the frame was aborted from the transmitting station.

**RDO** Receive Data Overflow.  
 A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.

**CRC** CRC Compare Check.  
 0: CRC check failed, received frame contains errors.  
 1: CRC check o.k., received frame is error free.

**CHAD4...0** Channel Address 4...0.  
 CHAD4...0 identifies on with IOM-port/channel the corresponding frame was received:  
 CHAD4...3: IOM-port number (3 - 0) of ELIC ( $\neq$  DOC port number)  
 CHAD2...0: IOM-channel number (7 - 0)

*Note: The contents of the receive status register is not changed.*

#### 2.1.2.4.7 Serial Port Configuration

The SACCO supports different serial port configuration, enabling the use of the circuit in

- point-to-point configurations
- point-to-multi-point configurations
- multi master configurations

#### Point-to-Point Configuration

The SACCO transmits frames without collision detection/resolution.  
 (CCR1:SC1, SC0: 00)

Additionally the input CxD can be used as a "clear to send" strobe. Transmission is inhibited by a '1' on the CxD-input. If "CxD" becomes '1' during the transmission of a frame, the frame is aborted and IDLE is transmitted. The CxD-pin is evaluated with the

falling edge of HDC.

When the “clear to send” function is not needed, CxD must be tied to  $V_{SS}$ .

## Bus Configuration

The SACCO can perform a bus access procedure and collision detection. As a result, any number of HDLC-controllers can be assigned to one physical channel, where they perform statistical multiplexing.

Collisions are detected by automatic comparison of each transmitted bit with the bit received via the CxD input. For this purpose a logical AND of the bits transmitted by parallel controllers is formed and connected to the input CxD. This may be implemented most simply by defining the output line to be open drain. Consequently the logical AND of the outputs is formed by simply tying them together (“wired or”). The result is returned to the CxD-input of all parallel circuits.

When a mismatch between a transmitted bit and the bit on CxD is detected, the SACCO-stops sending further data and IDLE is transmitted. As soon as it detects the transmit bus to be idle again, the controller automatically attempts to re-transmit its frame. By definition, the bus is assumed idle when x consecutive ones are detected in the transmit channel. Normally x is equal to 8.

An automatic priority adjustment is implemented in the multi master mode. Thus, when a complete frame is successfully transmitted, x is increased to 10, and its value is restored to 8 when 10 '1's are detected on the bus (CxD). Furthermore, transmission of new frames may be started by the controller after the 10<sup>th</sup> '1'.

This multi master, deterministic priority management ensures an equal right of access of every HDLC-controller to the transmission medium, thereby avoiding blocking situations.

Compared to the Version 1.2 the Version 1.3 provides new features:

Push-pull operation may be selected in bus configuration (up to Version 1.2 only open drain):

- When active TXDA / TXDB outputs serial data in push-pull-mode.
- When inactive (interframe or inactive time-slots) TXDA / TXDB outputs '1'.

*Note: When bus configuration with direct connection of multiple ELIC's is used open drain option is still recommended.*

*The push-pull option with bus configuration can only be used if an external tri-state buffer is placed between TXDA / TXDB and the bus.*

*Due to the delay of TSCA / TSCB in this mode (see description of bits SOC(0:1) in register CCR2 (**chapter 5.1.1.6.9**) these signals cannot directly be used to enable this buffer.*

## Timing Mode

When the multi master configuration has been selected, the SACCO provides two timing modes, differing in the period between sending data and evaluating the transmitted data for collision detection.

- Timing mode 1 (CCR1:SC1, SC0 = 01)  
Data is output with the rising edge of the transmit clock via TxD and evaluated 1/2 clock period later with the falling clock edge at the CxD pin.
- Timing mode 2 (CCR1:SC1, SC0 = 11)  
Data is output with the falling clock edge and evaluated with the next falling clock edge. Thus a complete clock period is available during data output and their evaluation.

### 2.1.2.4.8 Test Mode

To provide support for fast and efficient testing, the SACCO can be operated in the test mode by setting the TLP-bit in the MODE-register.

The serial input and output pins (TxD, RxD) are connected generating a local loop back. As a result, the user can perform a self-test of the SACCO. Transmit lines TXDA/B are also active in this case, receive inputs RXDA/B are deactivated.

### 2.1.2.5 D-Channel Arbiter

The D-channel arbiter facilitates the simultaneous serving of multiple D-channels with one HDLC-controller (SACCO-A) allowing a full duplex signaling protocol (e.g. LAPD). It builds the interface between the serial input/output of SACCO-channel A and the time-slot oriented D-channels on the EPIC-1 IOM-2 interface.

The SACCO-operation mode “transparent mode 0” has to be selected when using the arbiter.

It is only possible to operate the D-channel arbiter with framing control modes 3, 6 and 7, (refer to register EPIC-1.CMD2:FC(2:0)).

The arbiter consists of three sub blocks:

- Arbiter state machine (ASM): selects one subscriber for upstream D-channel assignment
- Control channel master (CCM): issues the “D-channel available” information from the arbiter in the control channel
- Transmit channel selector (TCHS): selects one or a group of subscribers for D-channel assignment

Functional Block Description

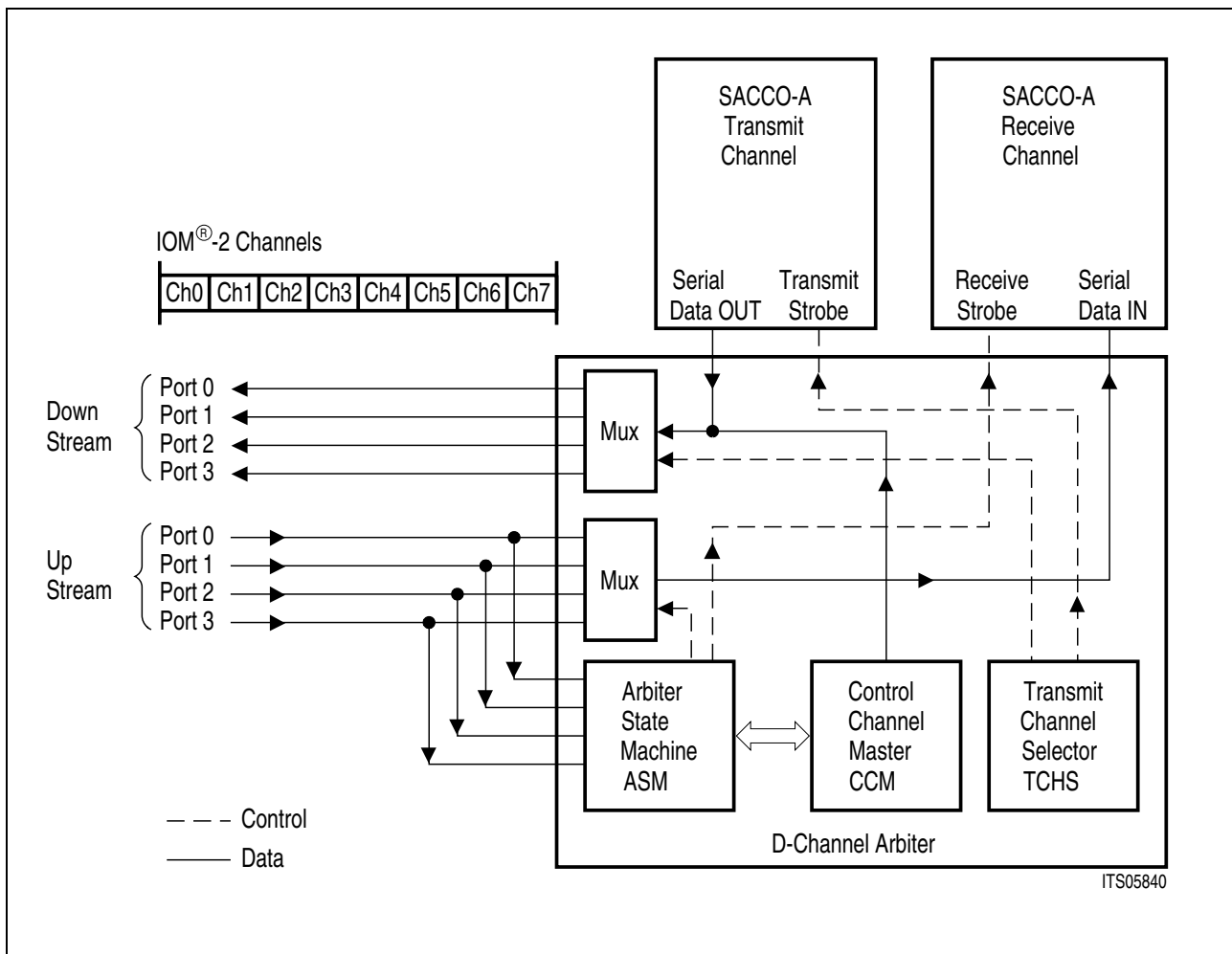


Figure 2-19 D-Channel Arbiter

2.1.2.5.1 Upstream Direction

In upstream direction the arbiter assigns the receive channel of SACC0-A to one subscriber terminal.

It uses an unidirectional control channel to indicate the terminals whether their D-channels are available or blocked. The control channel is implemented using different existing channel structures to close the transmission path between the line card HDLC-controller and the HDLC-controller in the subscriber terminal. On the line card, the control channel is either integrated in the C/I-channel or transmitted in the MR-bit depending on a programming of bit AMO:CCHH (OCTAT-P → C/I channel, IBC → MR-bit).



### Arbiter State Machine

The D-channel assignment is performed by the arbiter state machine (ASM), implementing the following functionality.

- (0) After reset or when SACCO-A clock mode is not 3 the ASM is in the state “**suspended**”. The user can initialize the arbiter and select the appropriate SACCO clock mode (mode 3).
- (1) When the receiver of SACCO-A is reset and clock mode 3 is selected the ASM enters the state “**full selection**”. In this state all D-channels enabled in the D-channel enable registers (DCE) are monitored.
- (2) Upon the detection of the first ‘0’ the ASM enters the state “**expect frame**”. When simultaneously ‘0’s are detected on different IOM-2 channels, the lowest channels number is selected. Channel and port address of the related subscriber are latched in arbiter state register (ASTATE), the receive strobe for SACCO-A is generated and the DCE-values are latched into a set of slave registers (DCES). Additionally a suspend counter is loaded with the value stored in register SCV. The counter is decremented after every received byte (4 IOM-frames).
- (3) When the counter underflows before the state “expect frame” was left, the corresponding D-channel is considered to produce permanent bit errors (typical pattern: ...111011101011...). The ASM emits an interrupt, disables the receive strobe and enters the state “**suspended**” again. The user can determine the affected channel by reading register ASTATE. In order to reactivate the ASM the user has to reset the SACCO-A receiver.
- (4) When seven consecutive ‘1’s are detected in the state “expect frame” before the suspend counter underflows the ASM changes to the state “**limited selection**”. The previously detected ‘0’ is considered a single bit error (typical pattern: ...1111110111111111...). The receive strobe is turned off and the DCES-bit related to the corresponding D-channel is reset, i.e. the subscriber is temporarily excluded of the priority list.
- (5) When SACCO-A indicates the recognition of a frame (frame indication after receiving 3 bytes incl. the flag) before the suspend counter underflows the ASM enters the state “**receive frame**”.
- (6) The ASM-state changes from “receive frame” to “**limited selection**” when SACCO-A indicates “end of frame”. The receive strobe is turned off and the DCES-bit related to the corresponding D-channel is reset. The ASM again monitors the D-channels but limited to the group enabled in the slave registers DCES “anded” with DCE. The “and” function guarantees, that the user controlled disabling of a subscriber has immediate effect.
- (7) When the ASM detects a ‘0’ on the serial input line it enters the state “**expect frame**”. Channel and port address of the related subscriber are latched in the arbiter state register (ASTATE), the receive strobe for SACCO-A is generated and



---

## Functional Block Description

the suspend counter is loaded with the value stored in register SCV. The counter is decremented after every received byte. When simultaneously '0's are detected on different IOM-2 channels, the lowest channel is selected.

- (8) When the ASM does not detect any '0' on the remaining serial input lines during n IOM-frames (n is programmed in the register AMO) it re-enters the state “**full selection**”. The list of monitored D-channels is then increased to the group selected in the user programmable DCE-registers. **In order to avoid arbiter locking n has to be greater than the value described in chapter 2.1.2.5.3 or must be set to 0.**
- (9) If n is set to 0, then the state “limited selection” is skipped.

The described combination of DCE and DCES implements a priority scheme guaranteeing that (almost) simultaneous requesting subscribers are served sequentially before one is selected a second time.

The current ASM-state is accessible in ASTATE7:5.

Functional Block Description

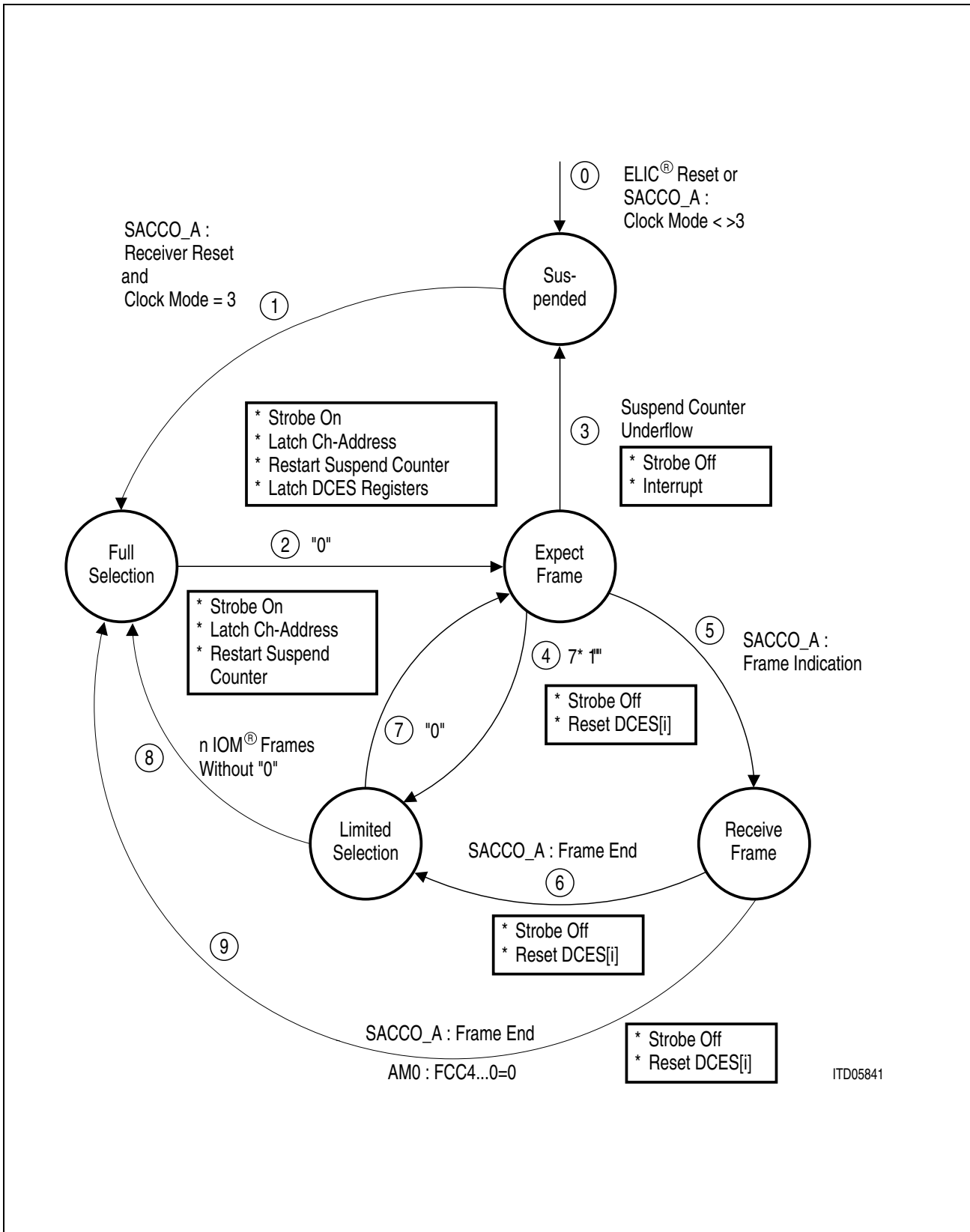


Figure 2-20 Arbiter State Machine (ASM)

### Control Channel Master

The control channel master (CCM) issues the “D-channel available” information in the control channel as shown in **Table 2-10**. If a D-channel is not enabled by the arbiter, the control channel passes the status, stored in the EPIC-1 control memory (C/I, MR). For correct operation of the arbiter this status bit has to contain the “blocked” information for all D-channels under control of the arbiter.

If the ASM is in the state “**suspended**” the arbiter functionality depends on the status of the Control Channel Master:

The CCM is enabled if AMO:CCHM = ‘1’. All subscribers will be sent the “available/blocked” information (C/I or MR) as programmed in the control memory. However, the control memory should be programmed as “blocked”.

The CCM is disabled if AMO:CCHM = ‘0’. All in the DCE-registers enabled subscribers (DCE = ‘1’) will be sent the information “available” (which has a higher priority than the “blocked” information from EPIC-1).

If the ASM is in the state “**full selection**” all D-channels are marked to be available which are enabled in the user programmable DCE-registers. When the user reprograms a DCE-register this has an immediate effect, i.e. a currently transmitting subscriber can be forced to abort its message.

If the ASM is in the state “**limited selection**” the subscribers which are currently enabled in DCE and DCES get the information “available”; they can access the D-channel. The DCE/DCES **anding** is performed in order to allow an immediate disabling of individual subscribers.

In the state “**expect frame**” and “**receive frame**” all channels except one (addressed by ASTATE4:0) have blocked D-channels. The disabling of the currently addressed D-channel in DCE has an immediate effect; the transmitter (HDLC-controller in the subscriber terminal) is forced to abort the current frame.

Depending on the programming of AMO:CCHH the available/blocked information is coded in the C/I-channel or in the MR-bit.

**Table 2-10 Control Channel Implementation**

CCHH	Control via	Available	Blocked
1	MR	1	0
0	C/I	x0xx	x1xx

The CCHM is activated independently of the SACCO-clock mode by programming AMO:CCHM. Even when the ASM is disabled (clock mode not 3) the CCHM can be activated. In this case the content of the DCE-registers defines which D-channels are enabled.

**Functional Block Description**

When a D-channel is enabled in the DCE-register and available, the control channel master takes priority over the C/I- (MR) values stored in the EPIC-1 control memory and writes out either MR = 1 or C/I = x0xx. When a D-channel is enabled but blocked, the control channel master simply passes the C/I- (MR) values which are stored in the EPIC-1 control memory. These values should have been programmed as MR = 0 or C/I = x1xxx.

When a D-channel is disabled in the DCE-register the control channel master simply passes the C/I- (MR) values which are stored in the EPIC-1 control memory. This gives the user the possibility to exclude a D-channel from the arbitration but still decide whether the excluded channel is available or blocked.

Overview of different conditions for control channel handling/information sent to subscribers:

**Table 2-11**

<b>Clock Mode</b>	<b>3</b>		<b>X</b>		<b>X</b>	
<b>ASM State</b>	Not suspended		Suspended		X	
<b>CCHM</b>	'1' = enabled		'1' = enabled		'0' = disabled	
<b>Subscriber in DCEs</b>	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled
<b>Information sent to Subscribers = "available" or "blocked"</b>	According to the D-channel <b>Arbiter</b> State (CCM)	Content of the EPIC-1 Control Memory- (C/I or MR)	Content of the EPIC-1 Control Memory- (C/I or MR)	Content of the EPIC-1 Control Memory- (C/I or MR)	Available!	Content of the EPIC-1 Control Memory- (C/I or MR)

**2.1.2.5.2 Downstream Direction**

In downstream direction no channel arbitration is necessary because the sequentiality of the transmitted frames is guaranteed.

In order to define IOM-channel and port number to be used for a transmission, the transmit channel selector (TCHS) provides a transmit address register (XDC) which the user has to write before a transmit command (XTF) is executed. Depending on the programming of the XDC-register the frame is transmitted in the specified D-channel or send as broadcast message to the broadcast group defined in the registers BCG1-4.

Due to the continuous frame transmission feature of the SACCO, the full 16-kbit/s bandwidth of the D-channel can be utilized, even when addressing different subscribers.

*Note: The broadcast group must not be changed during the transmission of a frame*

Functional Block Description

2.1.2.5.3 Control Channel Delay

Depending on the selected system configuration different delays between the activation of the control channel and the corresponding D-channel response occur.

Table 2-12 Control Channel Delay Examples

System Configuration	Circuit Chain	Number of Frames (= 125 μs)			
		Blocked → Available		Available → Blocked	
		min.	max.	min.	max.
U <sub>PN</sub> line card - - U <sub>PN</sub> phone	ELIC +OCTAT-P +ISAC-P TE	4	8	4	8
U <sub>PN</sub> line card - - S <sub>0</sub> adapter - - S <sub>0</sub> phone	ELIC +OCTAT-P +ISAC-P TE + SBCX + ISAC-S	9	13	5	9
U <sub>PN</sub> line card - - U <sub>PN</sub> adapter - - U <sub>PN</sub> phone	ELIC +OCTAT-P +ISAC-P TE + ISAC-P TE + ISAC-P TE	9	13	9	13
S <sub>0</sub> line card - S <sub>0</sub> phone	ELIC + QUAT-S + ISAC-S TE	4	8	4	8

**Beware of Arbiter Locking!**

In the state “limited selection”, the D-channel arbiter sends the “blocked” information to the terminal from which the last HDLC-frame was received. Since the “blocked” information reaches the terminal with several IOM-frames delay  $t_{CCDD}$  (e.g. after  $5 \times 125 \mu s$ ) the terminal may already have started sending a second HDLC-frame. On reception of the “blocked” information the terminal immediately aborts this frame.

Since the abort sequence of the second frame reaches the ELIC with several frames delay  $t_{DCDU}$ , the full selection counter value must be set so that the D-channel arbiter re-enters the state “full selection” only after the abort sequence of the second frame has reached the ELIC.

If the D-channel arbiter re-enters the “full selection” state (in which it again sends an “available” information to the terminal) before the abort sequence has reached the ELIC, it would mistake a ‘0’ of the second frame as the start of a new frame. When the delayed abort sequence arrives at the ELIC, the D-channel arbiter would then switch back to the state “limited selection” and re-block the terminal. Thus the D-channel arbiter would toggle between sending “available” and “blocked” information to the terminal, forever aborting the terminal’s frame. The arbiter would have locked.

## Functional Block Description

In order to avoid such a locking situation the time  $t_{DFS}$  min. (value in the AMO-register) has to be greater then the maximum delay  $t_{CCDD}$  (for the case "available" → "blocked") plus the delay  $t_{DCDU}$ .

- For the QUAT-S a value of 0 is recommended for the suspend counter (register SCV). For the OCTAT-P it is recommended to program  $SCV = 1$  in the case of 2 terminals  
 $SCV = 0$  if one terminal is used.

See the following diagram:

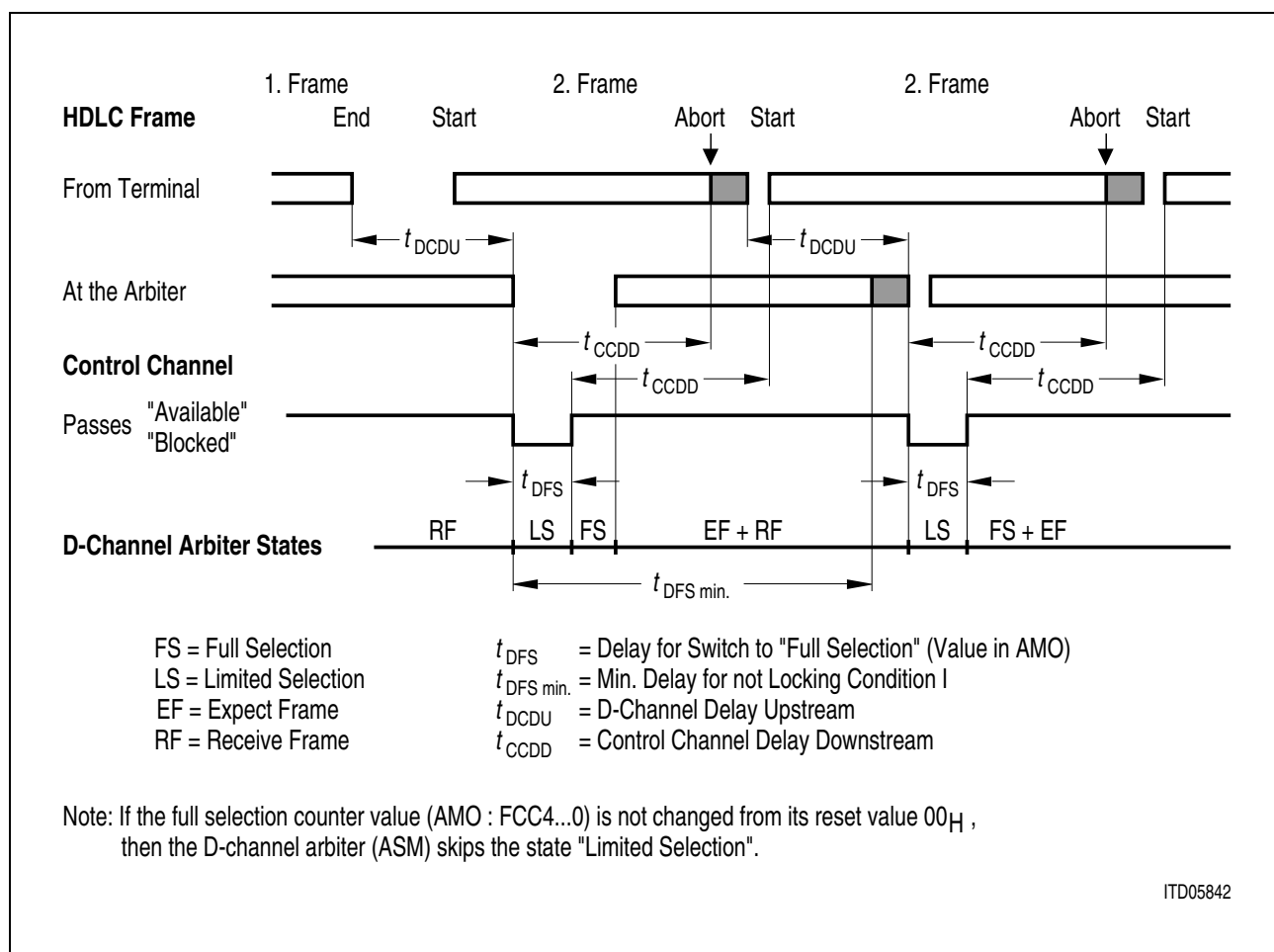


Figure 2-21

### 2.1.2.5.4 D-Channel Arbiter Co-operating with QUAT-S Circuits

When D-channel multiplexing is used on a  $S_0$ -bus line card, only the transmit channel selector of the arbiter is used.

The arbiter state machine can be disabled because the QUAT-S offers a self arbitration mechanism between several  $S_0$ -buses. This feature is implemented by building a wired OR connection between the different E-channels. As a result, the arbitration function

Functional Block Description

does not add additional delays. This means that the priority management on the S<sub>0</sub>-bus (two classes) still may be used, allowing the mixture of signaling and packet data.

Nevertheless, it still can make sense to use the ELIC arbiter in this configuration. The advantage of using the arbiter is, that if one terminal fails the others will not be blocked.

2.2 SIDE C

The SIDE C is a 4-channel signaling controller containing slightly modified SACCO modules and a control logic for DRDY handling (Stop/Go signal from QUAT-S).

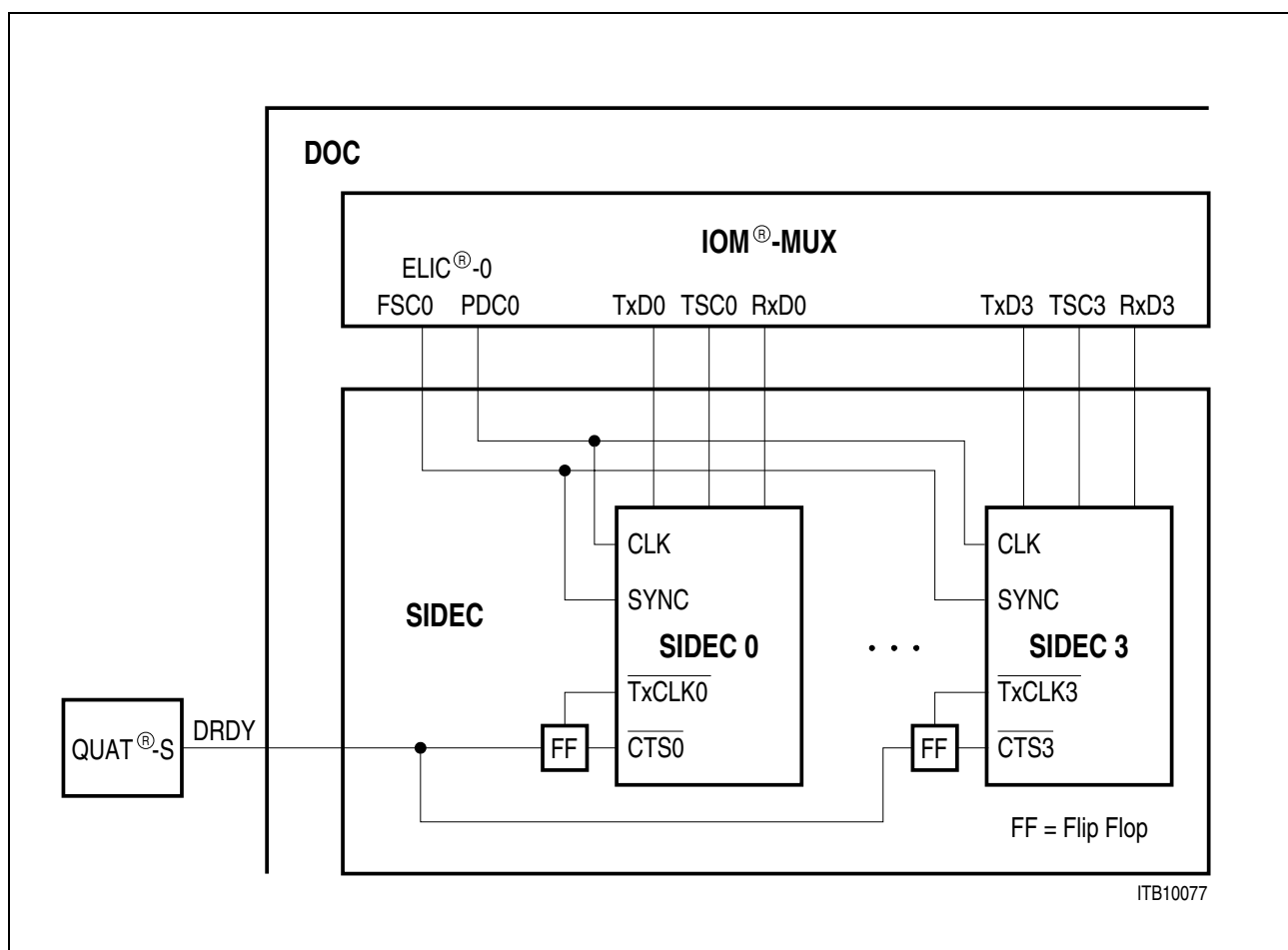
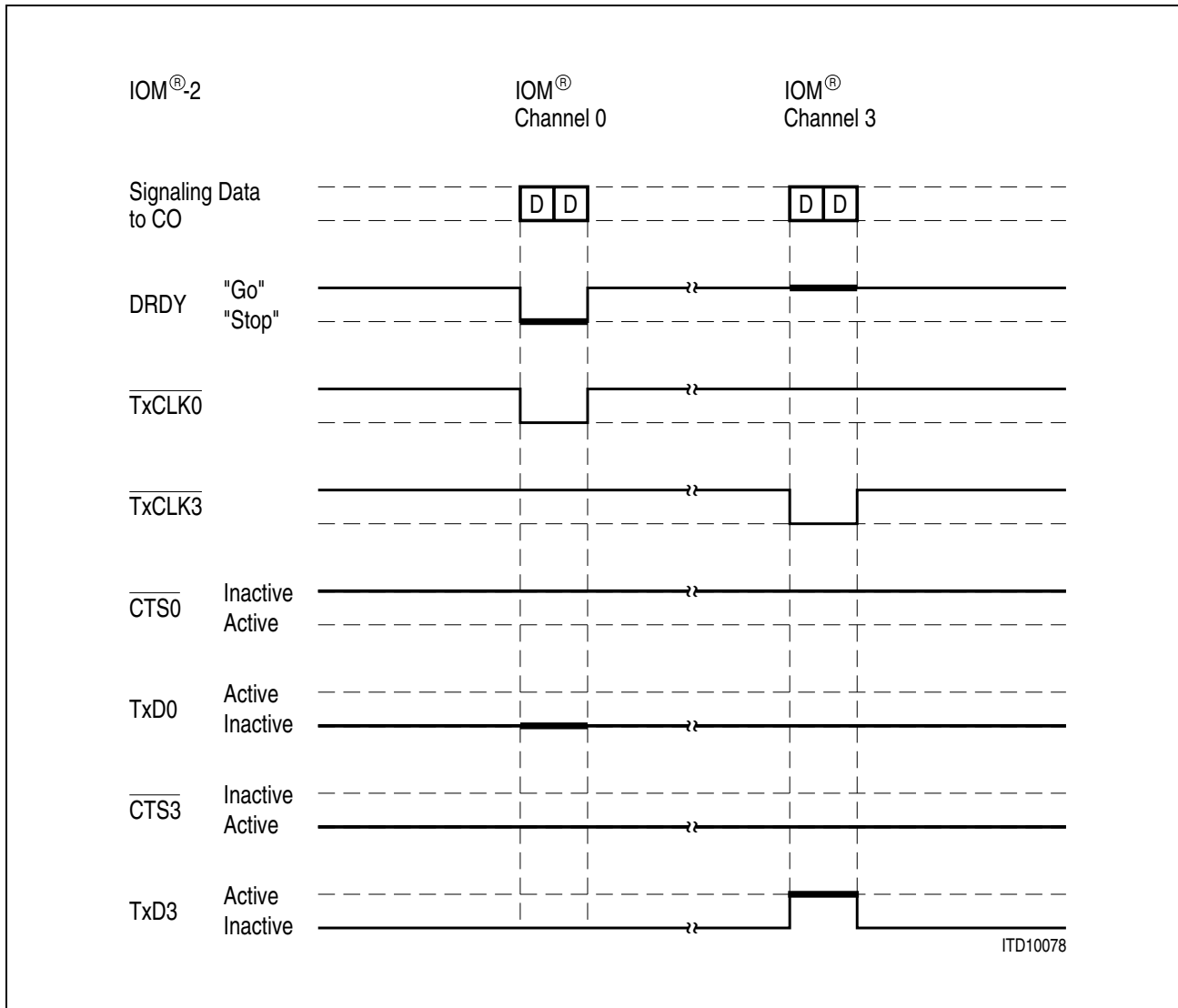


Figure 2-22 SIDE C Block Diagram

Depending on the DRDY signal from QUAT-S (in LT-T mode) the SIDE Cn sends data in the programmed time-slot or waits for a “Go” signal, Figure 2-22 and Figure 2-23.



**Figure 2-23 SIDE C Signals**

The DRDY signal is latched internally using a timeslot indication signal  $\overline{\text{TxCLKx}}$  and switches the transmission on/off via the CTSx pins.

A "Stop" forces the affected channel to abort the HDLC frame; upon a "Go" the affected channel restarts the frame.

### 2.3 Multiplexers

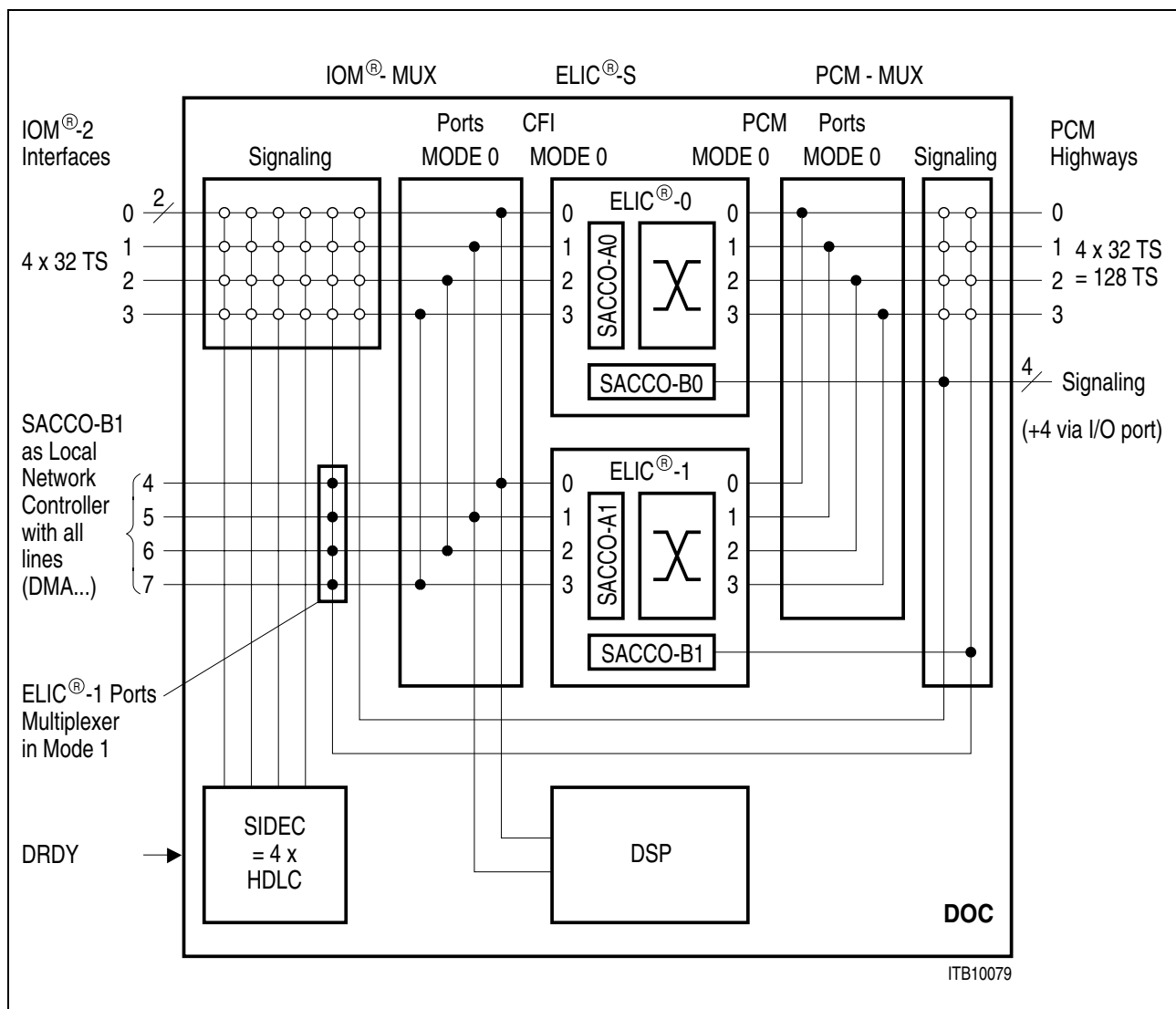
As the DOC contains two independent switches and eight different HDLC controllers, multiple programmable (multimode) multiplexers are implemented (**Figure 2-24**):

- An IOM-Ports multiplexer
- A PCM-Ports multiplexer
- IOM and PCM signaling multiplexers
- An ELIC1-Port multiplexer



Functional Block Description

All multiplexers can be programmed to different modes of operation.



**Figure 2-24 Principle Block Diagram of IOM and PCM Multiplexers; Mode 0-0-0-0-1**

*Note: Mode 0-0-0-0-1 means:*  
*the IOM-Ports Multiplexer is in Mode 0*  
*the ELIC CFI Ports are in Mode 0*  
*the ELIC PCM Ports are in Mode 0*  
*the PCM-Ports Multiplexer is in Mode 0*  
*the ELIC1-Ports Multiplexer is in Mode 1*

The meaning of the circles within the Signaling Multiplexers is explained in **Figure 2-25**.



### 2.3.1 IOM<sup>®</sup>- and PCM-Ports Multiplexers

#### 2.3.1.1 IOM<sup>®</sup> Multiplexer for IOM<sup>®</sup>-2 Ports (CFI Interfaces of EPIC)

The IOM multiplexer connects the 8 IOM-2 ports to the two integrated ELICs.

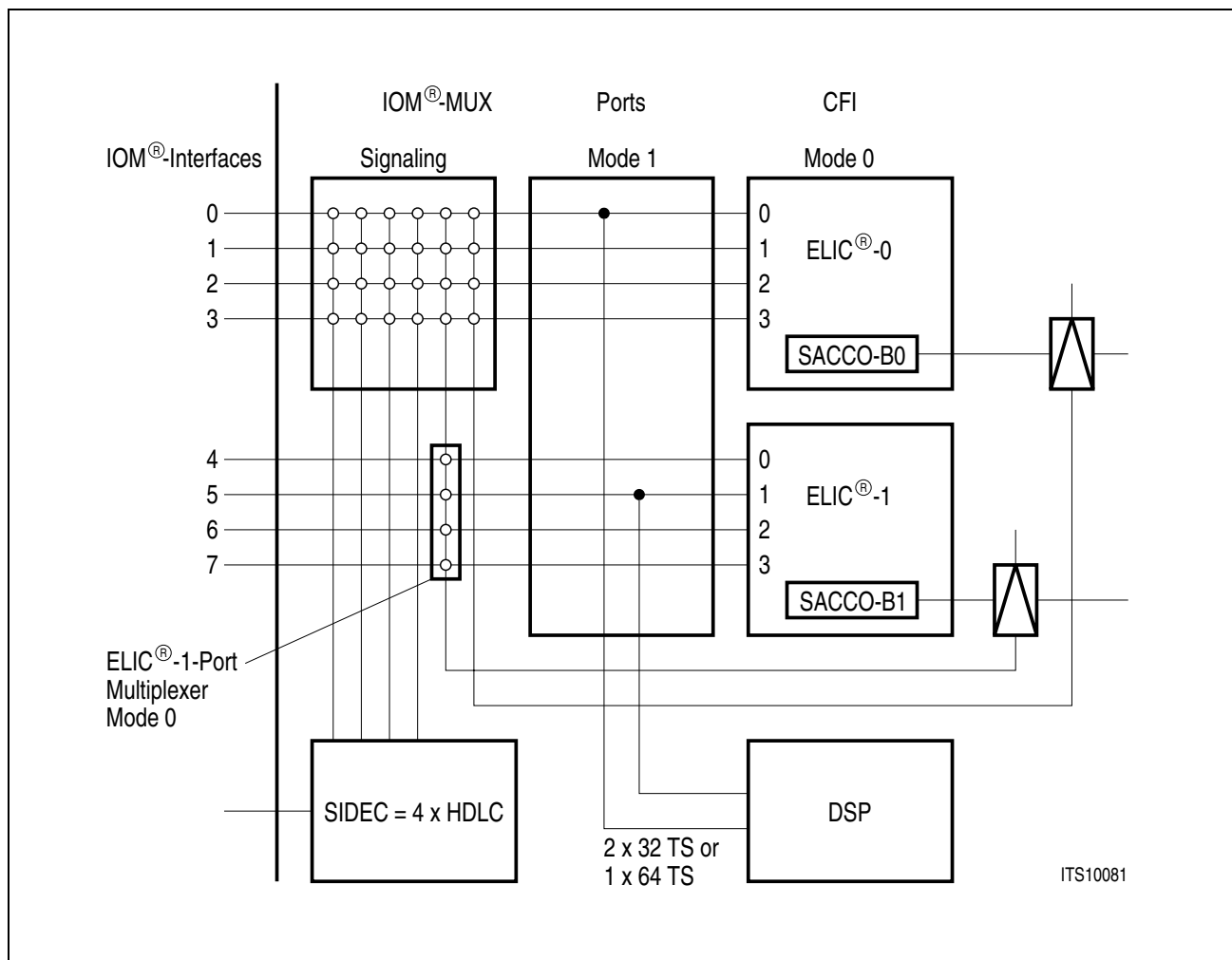
The IOM-MUX can be programmed to one of 2 different modes:

##### MODE 0

- ELIC0 and ELIC1 are connected together: DD00 with DD10, DU00 with DU10, ...

##### MODE 1

- All ELIC0 lines (IOM) reach DOC pins (IOM-2 Ports 0 to 3)
- ELIC1 lines reach ELIC1-Port Multiplexer and are not connected to any ELIC0 line. (State after DOC Reset)



**Figure 2-26 IOM<sup>®</sup> Ports Multiplexer in Mode 1 and ELIC<sup>®</sup>-1-Ports Multiplexer in Mode 0**

### 2.3.1.2 PCM-Ports Multiplexer for PCM Highways

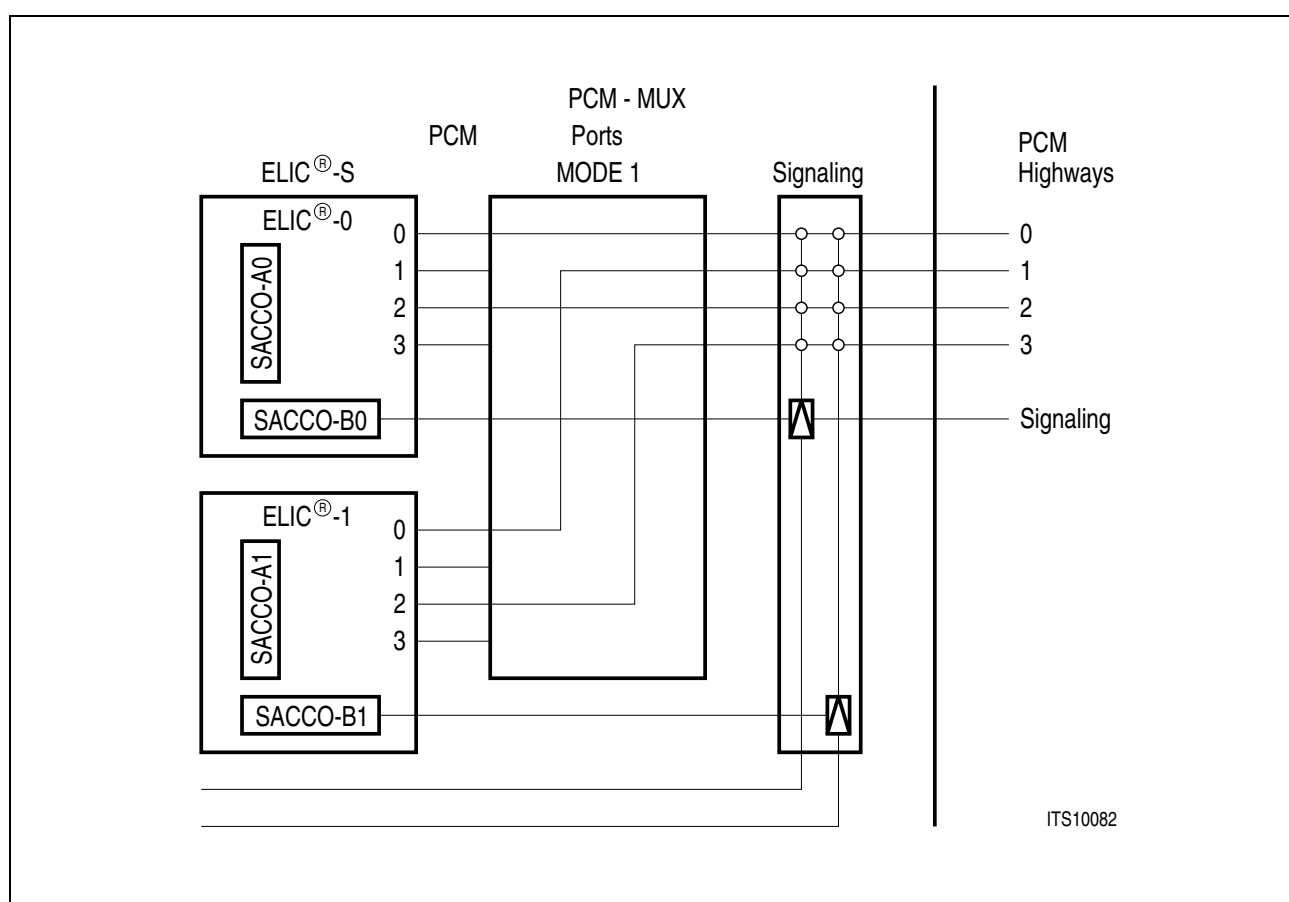
The PCM multiplexer connects the 4 PCM ports of the DOC to the two integrated ELICs. The PCM-MUX can be programmed to one of 2 different modes:

#### MODE 0

- ELIC0 and ELIC1 are connected together: TXD00 with TXD10, RXD00 with RXD10, ... **Figure 2-24** (State after DOC Reset)

#### MODE 1

- Only the Ports 0 and 2 of both ELICs are connected to DOC pins, **Figure 2-27**



**Figure 2-27 PCM-Ports Multiplexer in Mode 1**

The TxD output data lines at the DOC have tri-state capability so that the DOC may be connected in parallel with further devices at the PCM interface. Additional tri-state control lines ( $\overline{TSC}$ ) indicate valid/invalid time-slots on the PCM interface. They can be used as control signals for external drivers. The RxD inputs contain a protection logic limiting the input current to 2.3 mA if the DOC is without power supply.

### 2.3.2 Multiplexers for Signaling Controllers

All 8 integrated HDLC Controllers can be used for D-channel signaling.  
6 HDLC Controllers can also be used for B-channel access.

The following configurations are programmable:

**Table 2-13**

Controllers	Comm. Channels	Connection		Max. Data Rate
		to	via	
SACCO-A0	D	IOM-2	D-Ch. Arbiter0	16 kbit/s
SACCO-A1	D	IOM-2	D-Ch. Arbiter1	16 kbit/s
SACCO-B0	D, B, general	IOM-2, PCM or ext.	Multiplexers	8.192 Mbit/s <sup>1)</sup>
SACCO-B1	D, B, general	IOM-2, PCM or ext.	Multiplexers	8.192 Mbit/s <sup>1)</sup>
SIDEC	D and B	IOM-2	Multiplexer	4 × 64 kbit/s

<sup>1)</sup> In clock mode 0 are only the first 64 time-slots accessible.

#### 2.3.2.1 SACCO-A0 and SACCO-A1

Both SACCO-A are dedicated to work with the D-channel arbiter only. They must always operate in “Transparent mode 0”.

For more details please refer to ELIC, PEB 20550, User’s Manual 1.96.

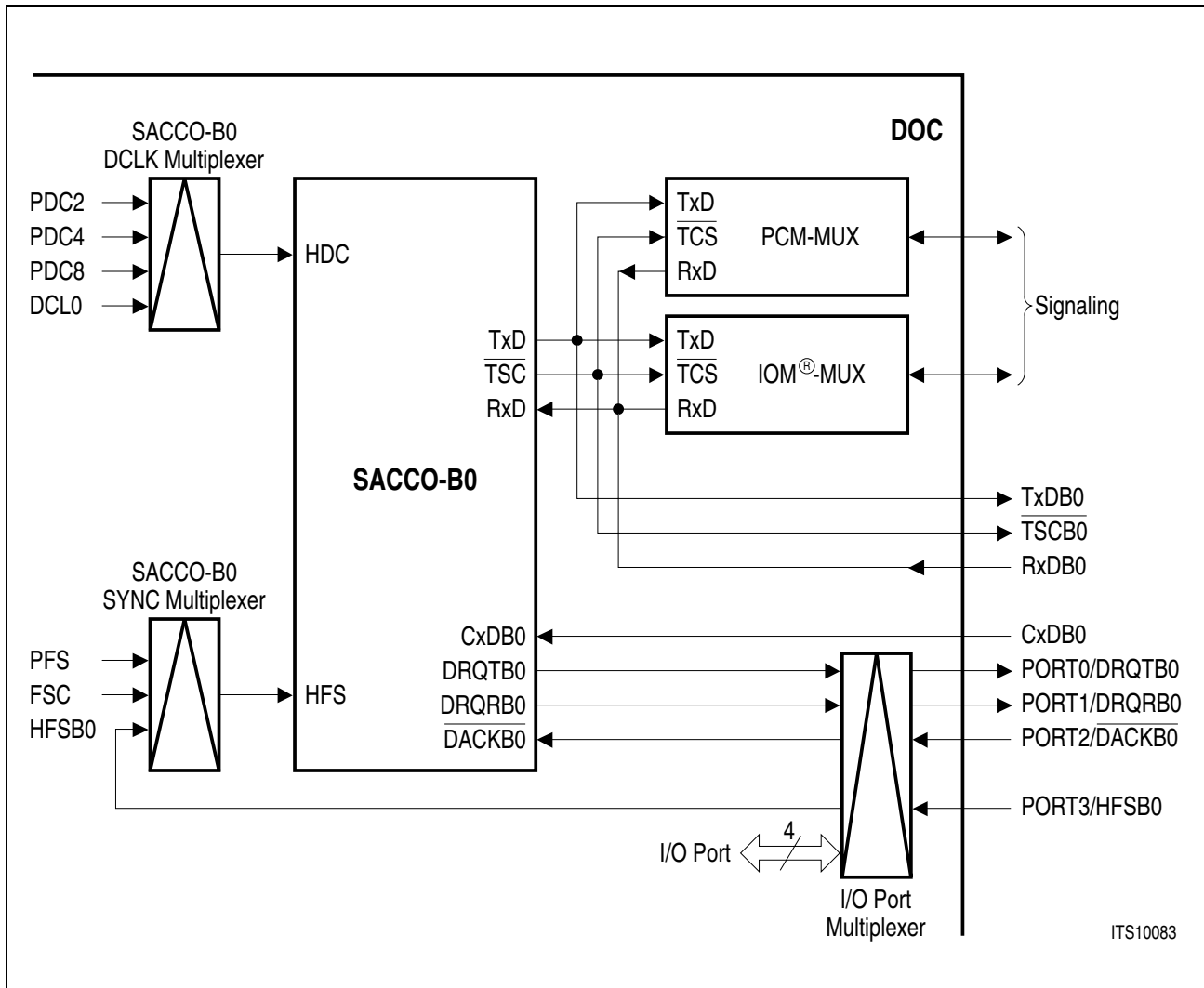
Neither SACCO-A are connected to any DOC pin.

#### 2.3.2.2 SACCO-B0

The serial interface can be assigned by the PCM Signaling Multiplexer to 3 different applications:

1. To any time-slot on the IOM-2 interface Port 0 to 3 in DD or DU direction
2. To the four PCM highways in DD or DU direction
3. As a stand-alone controller

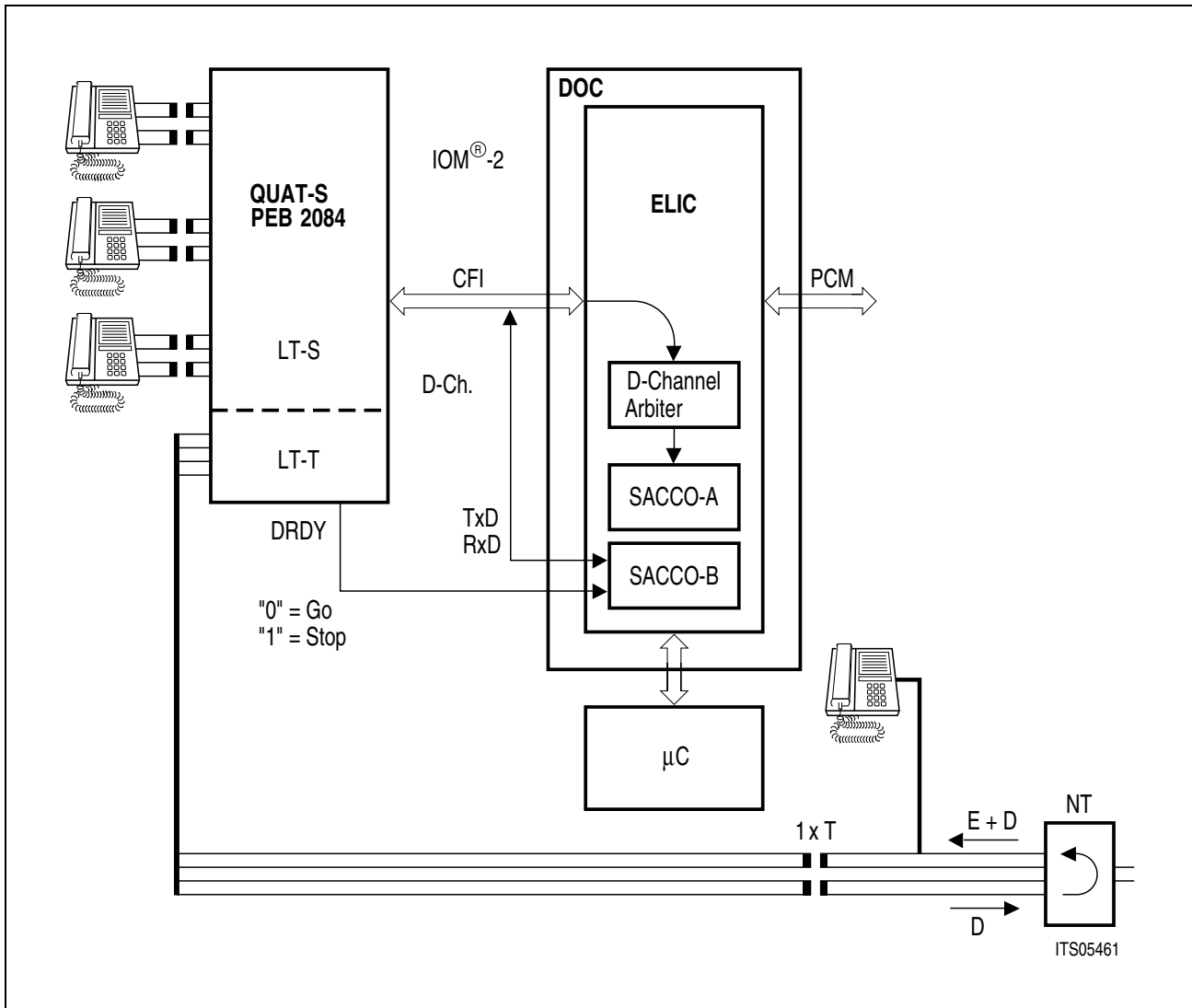
Refer to **Figure 2-24**.



ITS10083

**Figure 2-28 SACCO-B0 Multiplexers**

The SACCO-B0 can also be used as a stand-alone HDLC controller. The I/O Port can be programmed to provide all additional support lines (DMA interface). Different clock sources for transmit and receive data (DCLK) and for frame clock (SYNC) can be selected by the user. The input signal CXDB0 enables/inhibits the HDLC controller assigned to the S/T transceiver (i.e. QUAT-S in LT-T mode); refer to **Figure 2-29**.



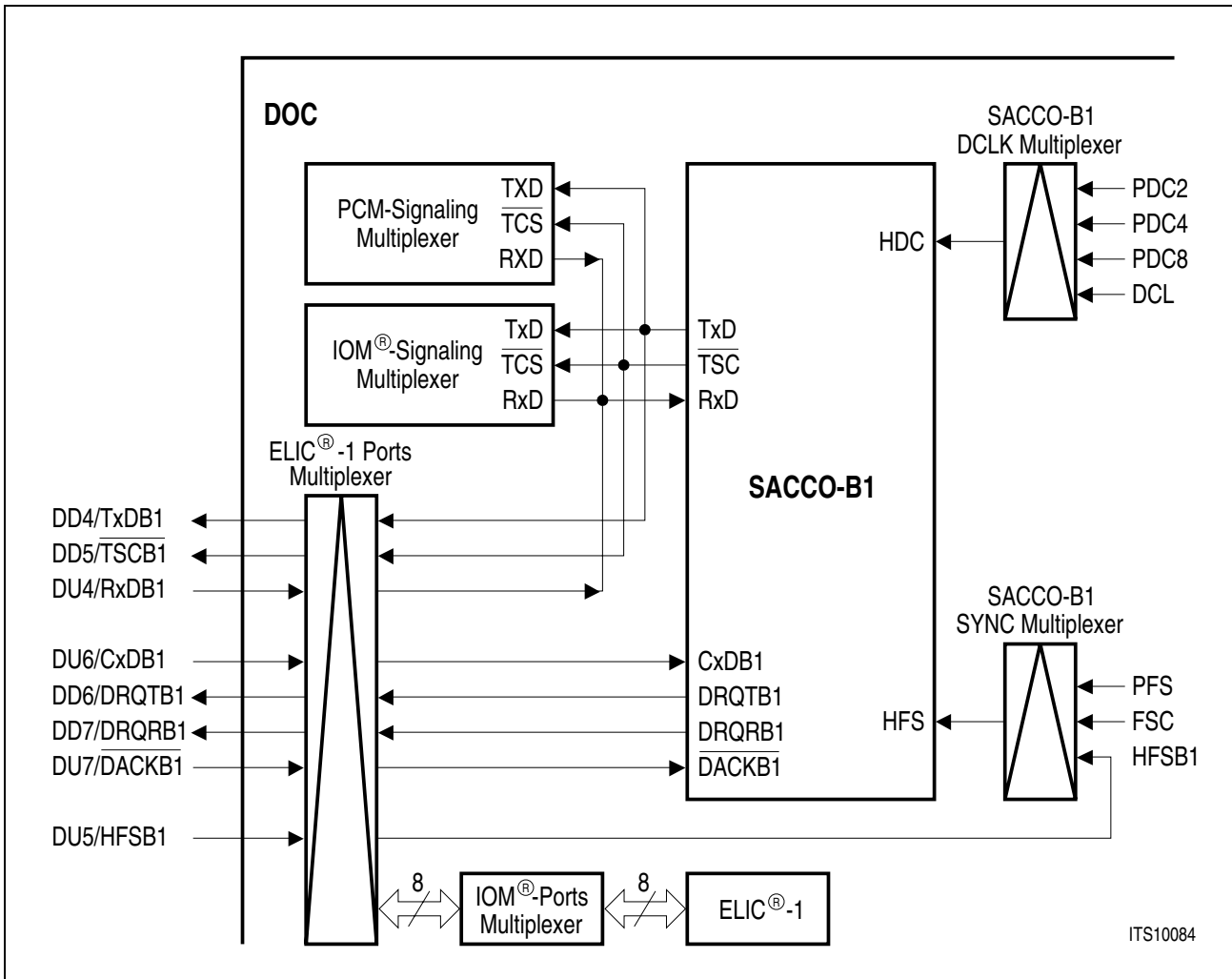
**Figure 2-29 QUAT-S with SACCO-B for Single-Channel LT-T Application**

### 2.3.2.3 SACCO-B1

The serial interface can be assigned by the PCM Signaling Multiplexer to 3 different applications:

1. To any time-slot on the IOM-2 interface Port 0 to 3 in DD or DU directions
2. To the four PCM highways in DD or DU directions
3. As a stand alone controller

Refer to **Figure 2-24**.



**Figure 2-30 SACCO-B1 Multiplexers**

The SACCO-B1 can be used as a stand-alone HDLC controller with DMA control lines if only 4 IOM-2 interfaces are used. The ELIC1 Ports Multiplexer must be in Mode 1 or 2, **Figure 2-30**.

The transmit and receive data clocks and the SYNC clock are selectable.

### 2.3.2.4 SIDEC

The four serial interfaces of the SIDEC can be assigned to:

- Any D-channel or B-channel of the four IOM-2 interfaces of ELIC0 via the IOM-Signaling Multiplexer
- Individually in data downstream or in data upstream direction. SIDEC is connected to FSC and DCL (of ELIC0).

See also **Figure 2-22**.



### 2.3.3 ELIC1-Ports Multiplexer

The ELIC1-Ports Multiplexer can operate in three different modes:

- MODE 0 The ELIC1 is connected to DOC pins; IOM-2 Ports 4 to 7 (after Reset) – **Figure 2-26**
- MODE 1 SACCO-B1 is connected to DOC pins; to IOM-2 Ports 4 to 7  
Thus the SACCO-B1 can be used as a stand-alone controller – **Figure 2-30**.
- MODE 2 Port 0 of ELIC1 is connected to DOC pins (IOM-2 Port 4) IOM-2 Ports 5 to 7 are connected with SACCO-B1.  
In this mode, ELIC1 can be used in EPIC CFI Mode 2 (8.192 Mbit/s on Port 0) and the SACCO-B1 can be used with all lines without restrictions. The TXD, RXD and  $\overline{TSC}$  lines can be assigned to any IOM-2 Port (0 to 3) or as described above.

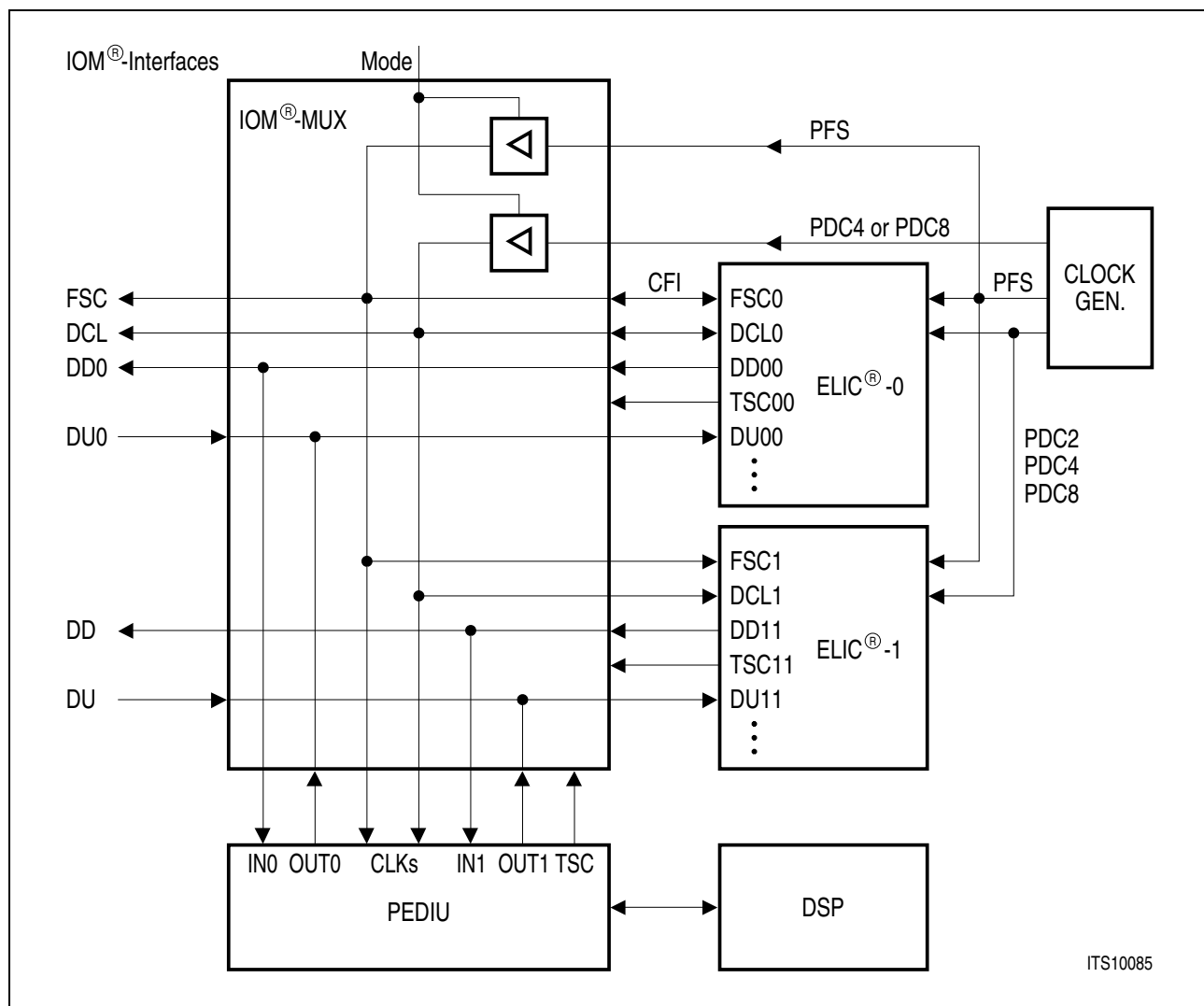
*Note: After reset, the multiplexers around ELICs (Figure 2-24) are in the following states:*

**Table 2-14**

IOM-Ports-Multiplexer	Mode 1.
IOM-Signaling-Multiplexer	No one signaling controller is connected to IOM-2.
PCM-Ports-Multiplexer	Mode 0.
PCM-Signaling-Multiplexer	No one signaling controller is connected to PCM highways
ELIC1-Ports Multiplexer	Mode 0

### 2.3.4 IOM<sup>®</sup>-Multiplexer for DSP Connection to EPICs

The DSP is connected to the ELIC0, Ports 0, and to the ELIC1, Port1, via a PCM-DSP Interface Unit (PEDIU). **Figure 2-31** shows the IOM-Port Multiplexer in Mode 1.



**Figure 2-31 PEDIU Connection to the ELICs**

- The DSP can be connected to both ELICs or to ELIC0 only (via IN0 and OUT0).
- ELIC0 and ELIC1 can be programmed for FSC and DCL be input or output clocks, depending on the system requirements.

*Note: If FSC/DCL are outputs of the ELIC0/1, only the outputs of ELIC0 drive the port pins FSC/DCL of the DOC. The clock outputs of ELIC1 are not connected in this case.*

- PEDIU Data Clock = DCL0 or PDC4 or PDC8  
PDC8 requires a DSP clock of 40 MHz.
- PEDIU Frame Sync Clock = FSC0 or PFS

### 2.3.5 PCM/IOM MUX Registers Description

#### 2.3.5.1 PCM/IOM MUX Mode Register (MMODE)

Address: 380<sub>H</sub>

μP interface mode: read/write

Reset Value: 01<sub>H</sub>

*Note: Bits 7...4 are unused, and are read as '0'.*

bit 7				bit 0			
0	0	0	0	EL1M1	EL1M0	PM	IM

This register defines muxes modes:

- IOM mux: mode0, mode1
- PCM mux: mode0, mode1
- ELIC1 mux: mode0, mode1, mode2

#### **EL1M1...0** - ELIC1 MUX Mode if IM = 1

- 00: ELIC1 mux mode0: ELIC1 is connected to IOM-2 ports 4-7
- 01: ELIC1 mux mode1: SACCO-B1 is connected to IOM-2 ports 4-7
- 10: ELIC1 mux mode2: SACCO-B1 is connected to IOM-2 ports 5-7  
ELIC1 port 0 is connected to IOM-2 port 4  
SACCO-B1 TXD and RXD are connected to one of IOM-2 ports0...3.

*Note: CXDB1 is internally tied to V<sub>SS</sub> in mode 0.*

#### **PM** - PCM MUX mode

- 0: PCM mux mode0: ELIC0 and ELIC1 are connected together to PCM ports0...3
- 1: PCM mux mode1: ELIC0 PCM ports 0, 2 are connected to DOC PCM ports 0, 2  
ELIC1 PCM ports 0, 2 are connected to DOC PCM ports 1, 3

#### **IM** - IOM MUX Mode

- 0: IOM mux mode0: ELIC0 and ELIC1 are connected together to IOM-2 ports0...3. ELIC1 is connected to ELIC1 port MUX
- 1: IOM mux mode1: ELIC0 is connected to IOM ports0...3;  
ELIC1 is connected to ELIC1 mux

*Note: For an overview please refer to Workingsheets for Multiplexers Programming, **chapter 9.2***

### 2.3.5.2 CFI Channel Select 0 Register (MCCHSEL0)

Address: 381<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

bit 7

bit 0

CD1	DIR1	PN11	PN10	CD0	DIR0	PN01	PN00
-----	------	------	------	-----	------	------	------

This register selects the IOM port that SIDE0 and SIDE1 will be connected to.

**CD1:** Connect/Disconnect SIDE1

- 0: SIDE1 is disconnected from IOM signaling mux
- 1: SIDE1 is connected to IOM signaling mux

**DIR1:** The DIRection of SIDE1 connection

- 0: HDLC TXD line connected to IOM DD line  
HDLC RXD line connected to IOM DU line
- 1: HDLC TXD line connected to IOM DU line  
HDLC RXD line connected to IOM DD line

**PN1 1...0:** Port Number which SIDE1 is connected to.

- 00: IOM port: PORT0
- 01: IOM port: PORT1
- 10: IOM port: PORT2
- 11: IOM port: PORT3

**CD0:** Connect/Disconnect SIDE0

- 0: SIDE0 is disconnected from IOM signaling mux
- 1: SIDE0 is connected to IOM signaling mux

**DIR0:** The DIRection of SIDE0 connection

- 0: HDLC TXD line connected to IOM DD line  
HDLC RXD line connected to IOM DU line
- 1: HDLC TXD line connected to IOM DU line  
HDLC RXD line connected to IOM DD line

**PN0 1...0:** Port Number which SIDE0 is connected to.

- 00: IOM port: PORT0
- 01: IOM port: PORT1
- 10: IOM port: PORT2
- 11: IOM port: PORT3

### 2.3.5.3 CFI Channel Select 1 Register (MCCHSEL1)

Address: 382<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

bit 7				bit 0			
CD3	DIR3	PN31	PN30	CD2	DIR2	PN21	PN20

This register selects the IOM port that SIDE2 and SIDE3 will be connected to.

**CD3:** Connect/Disconnect SIDE3

- 0: SIDE3 is disconnected from IOM signaling mux
- 1: SIDE3 is connected to IOM signaling mux

**DIR3:** <sup>1)</sup> The DIRection of SIDE3 connection

- 0: HDLC TXD line connected to IOM DD line  
HDLC RXD line connected to IOM DU line
- 1: HDLC TXD line connected to IOM DU line  
HDLC RXD line connected to IOM DD line

**PN3 1...0:** <sup>1)</sup>Port Number which SIDE3 is connected to.

- 00: IOM port: PORT0
- 01: IOM port: PORT1
- 10: IOM port: PORT2
- 11: IOM port: PORT3

**CD2:** Connect/Disconnect SIDE2

- 0:SIDE2 is disconnected from IOM signaling mux
- 1:SIDE2 is connected to IOM signaling mux

**DIR2:** <sup>1)</sup> The DIRection of SIDE2 connection

- 0: HDLC TXD line connected to IOM DD line  
HDLC RXD line connected to IOM DU line
- 1: HDLC TXD line connected to IOM DU line  
HDLC RXD line connected to IOM DD line

**PN2 1...0:** <sup>1)</sup>Port Number which SIDE2 is connected to.

- 00: IOM port: PORT0
- 01: IOM port: PORT1
- 10: IOM port: PORT2
- 11: IOM port: PORT3

*Note:* <sup>1)</sup> only valid if CDx=1

### 2.3.5.4 CFI Channel Select 2 Register (MCCHSEL2)

Address: 383<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

bit 7

bit 0

ICDB1	IDIRB1	IPNB11	IPNB10	ICDB0	IDIRB0	IPNB01	IPNB00
-------	--------	--------	--------	-------	--------	--------	--------

This register selects the IOM ports that SACCO-B0 and SACCO-B1 will be connected to.

**ICDB1:** Connect/Disconnect SACCO-B1 from IOM signaling MUX  
 0: SACCO-B1 is disconnected from IOM signaling mux  
 1: SACCO-B1 is connected to IOM signaling mux

**IDIRB1:** <sup>1)</sup> The DIRection of SACCO-B1 connection to IOM signaling MUX  
 0: HDLC TXD line connected to IOM DD line  
 HDLC RXD line connected to IOM DU line  
 1: HDLC TXD line connected to IOM DU line  
 HDLC RXD line connected to IOM DD line

**IPNB1 1...0:** <sup>1)</sup> IOM Port Number which SACCO-B1 is connected to.  
 00: IOM port: PORT0  
 01: IOM port: PORT1  
 10: IOM port: PORT2  
 11: IOM port: PORT3

**ICDB0:** Connect/Disconnect SACCO-B0 from IOM signaling MUX  
 0: SACCO-B0 is disconnected from IOM signaling mux  
 1: SACCO-B0 is connected to IOM signaling mux

**IDIRB0:** <sup>1)</sup> The DIRection of SACCO-B0 connection to IOM signaling MUX  
 0: HDLC TXD line connected to IOM DD line  
 HDLC RXD line connected to IOM DU line  
 1: HDLC TXD line connected to IOM DU line  
 HDLC RXD line connected to IOM DD line

**IPNB0 1...0:** <sup>1)</sup> IOM Port Number which SACCO-B0 is connected to.  
 00: IOM port: PORT0  
 01: IOM port: PORT1  
 10: IOM port: PORT2  
 11: IOM port: PORT3

Note: <sup>1)</sup> only valid if ICBx=1

### 2.3.5.5 PCM Channel Select 0 Register (MPCHSEL0)

Address: 384<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

bit 7

bit 0

PCDB1	PDIRB1	PPNB11	PPNB10	PCDB0	PDIRB0	PPNB01	PPNB00
-------	--------	--------	--------	-------	--------	--------	--------

This register selects the PCM ports that SACCO-B0 and SACCO-B1 will be connected to.

**PCDB1:** Connect/Disconnect SACCO-B1 from PCM signaling MUX  
 0: SACCO-B1 is disconnected from PCM signaling mux  
 1: SACCO-B1 is connected to PCM signaling mux

**PDIRB1:**<sup>1)</sup> The DIRection of SACCO-B1 connection to PCM signaling MUX  
 0: HDLC TXD line connected to PCM TXD line  
 HDLC RXD line connected to PCM RXD line  
 1: HDLC TXD line connected to PCM RXD line  
 HDLC RXD line connected to PCM TXD line

**PPNB1 1...0:**<sup>1)</sup> PCM Port Number which SACCO-B1 is connected to.  
 00: PCM port: PORT0  
 01: PCM port: PORT1  
 10: PCM port: PORT2  
 11: PCM port: PORT3

**PCDB0:** Connect/Disconnect SACCO-B0 from PCM signaling MUX  
 0: SACCO-B0 is disconnected from PCM signaling mux  
 1: SACCO-B0 is connected to PCM signaling mux

**PDIRB0:**<sup>1)</sup> The DIRection of SACCO-B0 connection to PCM signaling MUX  
 0: HDLC TXD line connected to PCM TXD line  
 HDLC RXD line connected to PCM RXD line  
 1: HDLC TXD line connected to PCM RXD line  
 HDLC RXD line connected to PCM TXD line

**PPNB0 1...0:**<sup>1)</sup> PCM Port Number which SACCO-B0 is connected to.  
 00: PCM port: PORT0  
 01: PCM port: PORT1  
 10: PCM port: PORT2  
 11: PCM port: PORT3

Note: <sup>1)</sup> only valid if PCDBx=1

## 2.4 Channel Indication Logic (CHI)

An output signal (CHI), as one additional line to the IOM-2 interface, indicates when the fourth byte (byte 3) of each subframe comes. This byte carries 2 D-bits, 4 C/I-bits, MR and MX-bits or 6 C/I- and 2 M-bits in every IOM-2 subframe.

- CHI signal is programmable
- The CHI signal may be activated or masked during the fourth byte of each subframe
- Four 8 bits registers control the CHI line
- Every bit in the CHI control register controls one subframe in IOM-2 frame
- After reset chi is masked until control register programmed.

CFI Mode 0: 32 TS with 8 subframes                      8 bits in one 8-bit register  
DCL = 2 x or 1 x data rate

CFI Mode 1: 64 TS with 16 subframes                    16 bits in two 8-bit registers  
DCL = 1 x data rate

CFI Mode 2: 128 TS with 32 subframes                 32 bits in four 8-bit registers  
DCL = 1 x data rate.

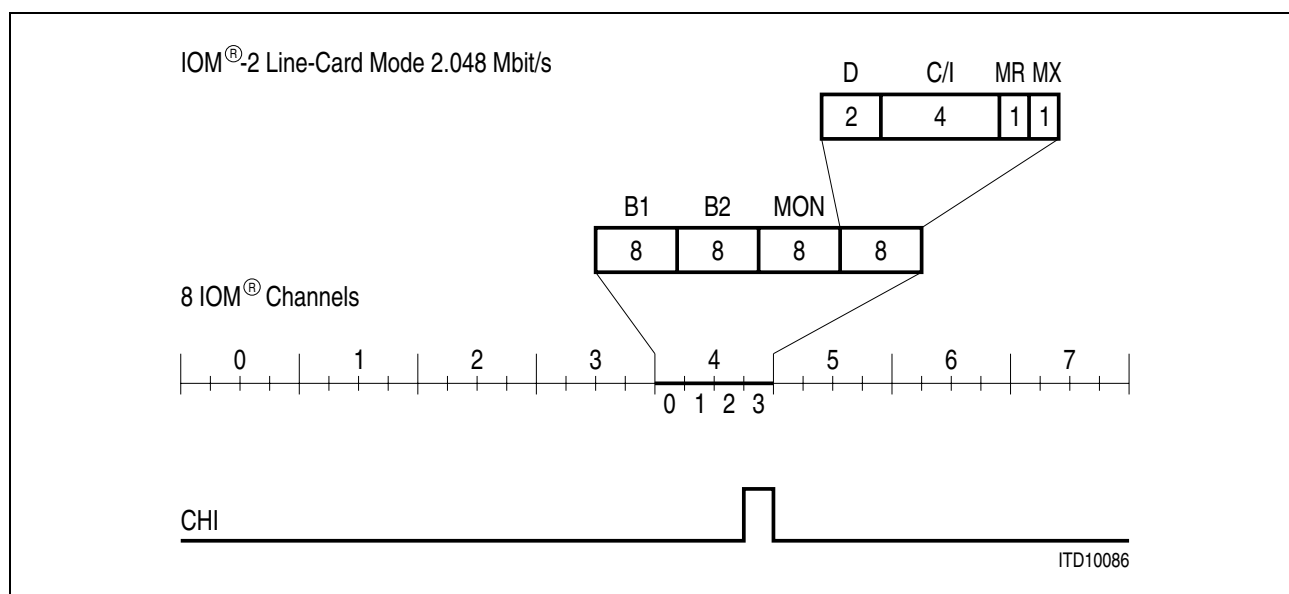


Figure 2-32 CHI Signal

### 2.4.1 CHI Configuration Register (VMODR)

Address 323<sub>H</sub>

reset value 00<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VMODR</b>	unused	unused	unused	unused	unused	CHIA	MOD1	MOD0

**CHIA** - flag for CHI logic activation ('0'-CHI logic is inactive/'1'-CHI logic is active)



**Functional Block Description**

<b>MOD1 - MOD0</b>	00	32 TS with 8 subchannels (single data rate), VDTR0 register is used as data register for CHI logic programming. Used when data frequency and data rate is 2.048 MHz
	01	64 TS with 16 subchannels, VDTR0 & VDTR1 registers are used for CHI logic programming. Used when data frequency and data rate is 4.096 MHz
	11	32 TS with 8 subchannels (double data rate), VDTR0 register is used as data register for CHI logic programming. Used when data frequency is 4.096 MHz and data rate 2.048 MHz

**2.4.2 CHI Control Registers (VDATR0:VDATR3)**

Reset value: Unchanged upon chip reset

μP interface mode: read/write

Address: 324<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VDATR0</b>	CTRL7	CTRL6	CTRL5	CTRL4	CTRL3	CTRL2	CTRL1	CTRL0

Address: 325<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VDATR1</b>	CTRL15	CTRL14	CTRL13	CTRL12	CTRL11	CTRL10	CTRL9	CTRL8

Address: 326<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VDATR2</b>	CTRL23	CTRL22	CTRL21	CTRL20	CTRL19	CTRL18	CTRL17	CTRL16

Address: 327<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VDATR3</b>	CTRL31	CTRL30	CTRL29	CTRL28	CTRL27	CTRL26	CTRL25	CTRL24

**CTRLn** control bit for subframe n in IOM-2 frame:  
 0 subframe is masked  
 1 subframe is enabled

## 2.5 FSC with Delay (FSCD)

**Frame Synchronization Clock with Delay**, which indicates the 32'nd time-slot start, related to the standard FSC. Used for synchronization of layer-1 devices connected to the second half of an extended IOM-2 interface with 64 time-slots; (i.e. two OCTAT-P connected to one 4 Mbit/s IOM-2 port).

The FSCD depends on the work mode of the PEDIU (PCM DSP Interface Unit):

PEDIU Work Mode 0/1: 32 TS with 8 subchannels.  
FSCD is not used (constantly '0').

PEDIU Work Mode 2/3: 64 TS with 16 subchannels.  
FSCD indicates the start of time-slot 32. It is delayed by 62.5  $\mu$ s relative to FSC.

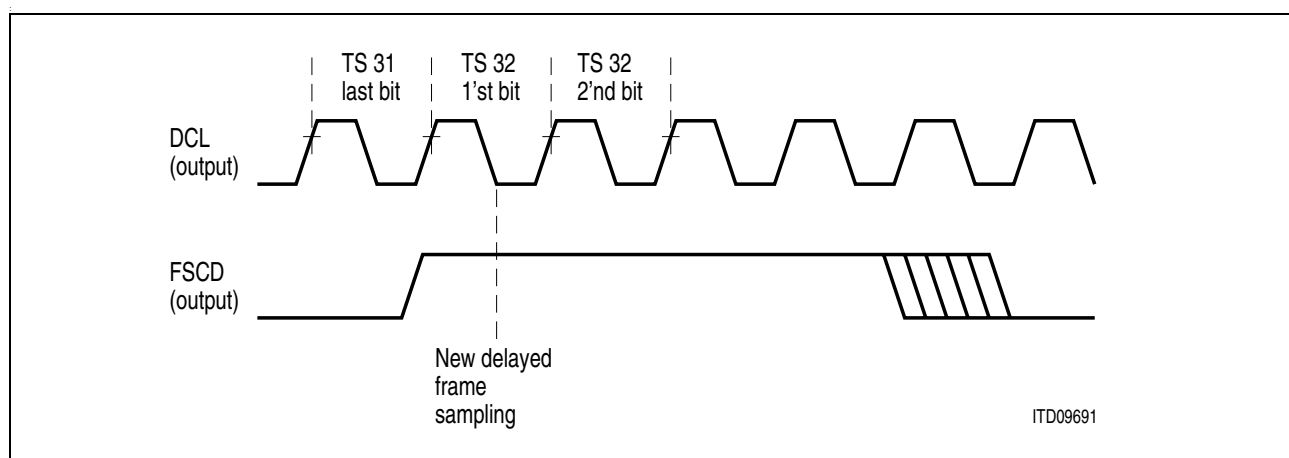
PEDIU Work Mode 4: 128 TS with 32 subchannels.  
FSCD indicates the start of time-slot 32. It is delayed by 125/4  $\mu$ s (31.25  $\mu$ s).

For more details about the PEDIU work modes, refer to **section 2.8.2.1: PEDIU Control Register (UCR)**

Two additional points should be emphasized about FSCD:

1. FSCD can be used only when FSC direction is configured as output. Otherwise, if FSC direction is configured as input, FSCD will stay in tri-state. For more details on configuring FSC direction, refer to **sections 2.10** and **2.12.3.1**. When FSC is configured as output and the PEDIU work mode is 0 or 1, or when the PEDIU is in IDLE mode, FSCD will be driven as constant '0'.
2. FSCD is designed to be sampled by external devices with DCL falling edge.

The next figure (**Figure 2-33**) demonstrate the behavior of FSCD, when the PEDIU Works in Mode 2, 3 or 4 and it is not in IDLE mode, and when FSC direction is output.



**Figure 2-33 FSCD Behavior**

For more details about using FSCD, refer to Applications, **section 6**.

## **2.6 Digital Signal Processor (DSP)**

Based on the cooperation between DSP Group (USA) and Siemens a Signal Processing Core with Extensions (OAK) will be integrated in the DOC.

OAK Features:

- Up to 40 MHz / 40 MIPS
- 7 instruction groups with totally 72 instructions (including Bit Manipulation)
- Two independent 16-bit data busses (X and Y) each with demultiplexed address and data busses

### **2.6.1 DSP Kernel Block Diagram**

The DSP Kernel consists of four units: Computation Unit (CU), Bit Manipulation Unit (BMU), Data Addressing Arithmetic Unit (DAAU) and Program Control Unit (PCU)

### **2.6.2 DSP Instruction Set**

The integrated DSP (OAK) executes the following instructions:

- 20 Arithmetic and Logical Instructions: ADD, ADDL, ADDH, ADDV, SUB, SUBL, SUBH, SUBV, OR, AND, XOR, CMP, CMPV, MODA, NORM, DIVS, MAX, MAXD, MIN and LIM.
- 12 Multiply Instructions: MPY, MPYSU, MAC, MACSU, MACUS, MACUU, MAA, MAASU, MSU, MPYI, SQR and SQRA.
- 10 Bit Manipulation Instructions: SET, RST, CHNG, TST0, TST1, TSTB, CHFC, SHFI, MODB and EXP.
- 10 Move Instructions: MOV, MOVP, MOVD, MOVS, MOVSI, MOVR, PUSH, POP, SWAP and BANKE.
- 3 Loop Instructions: REP, BREP and BREAK.
- 10 Branch and Call Instructions: BR, BRR, CALL, CALLR, CALLA, RET, RETD, RETI, RETID and RETS.
- 7 Control and Miscellaneous Instructions: NOP, MODR, EINT, DINT, TRAP, LOAD and CNTX.

## 2.7 DSP Control Unit (DCU)

### 2.7.1 General

The DCU is responsible for the following tasks:

- DSP Address decoding
- Control of external memories
- Emulation support
- Interrupt handling and test support
- Data Bus and Program Bus arbitration
- DSP run time statistics
- Program write protection
- Boot support

### 2.7.2 DSP Address Decoding

The DCU decodes the DSP data address bus (DXAP) and the DSP program address bus (PPAP) for performing the following tasks:

- Generating DSP memory mapped I/O read and write signals based upon decoding of the 8 MSB lines of the data address bus.
- Generating the circular buffer RAM read and write controls.
- Generating external program and data RAM controls upon detection of their address.
- Generating read signal for the internal program ROM.

**Table 2-15 DSP Program Address Space**

Address	Size	Number of Wait States	Description
0000 <sub>H</sub> -DFFF <sub>H</sub>	56 KW	0 for read <sup>1)</sup> 3 for write	External Program
E000 <sub>H</sub> -EFFF <sub>H</sub>	4 KW	0 for read <sup>1)</sup> 3 for write	Internal Program RAM (devided to four) - Option For Future implementation
F000 <sub>H</sub> -FDFF <sub>H</sub>	3.5 KW	–	Not used
FE00 <sub>H</sub> -FEFF <sub>H</sub>	0.25 KW	0	Syne Table ROM
FF00 <sub>H</sub> -FFFF <sub>H</sub>	0.25 KW	0	Internal Boot ROM

<sup>1)</sup> When accessing program memory, three more DSP cycles are added to the program read due to the multiplexed nature of the external program/data bus.

**Table 2-16 DSP Data Address Space**

Address	Size	Number of Wait States	Number of DSP Cycles	Description
0000 <sub>H</sub> -03FF <sub>H</sub>	1 KW	0	1	Internal XRAM
0400 <sub>H</sub> -3FFF <sub>H</sub>	15 KW	–	–	unused
4000 <sub>H</sub> -BFFF <sub>H</sub>	32 KW	0-7	4-11 <sup>1)</sup>	External data memory
C000 <sub>H</sub> -DFFF <sub>H</sub>	8 KW	0	1	OAK memory mapped registers
E000 <sub>H</sub> -EFFF <sub>H</sub>	4 KW	–	–	unused
F000 <sub>H</sub> -F0FF <sub>H</sub>	256 W	0 <sup>2)</sup>	1 <sup>2)</sup>	Circular RAM buffer
F100 <sub>H</sub> -F3FF <sub>H</sub>	0.75 KW	–	–	unused
F400 <sub>H</sub> - F7EE <sub>H</sub>	1 KW-16	0-7	4-11 <sup>1)</sup>	Emulation mail box (on CDI)
F7F0 <sub>H</sub> -F7FF <sub>H</sub>	16 W	1	2	OCEM Registers
F800 <sub>H</sub> -FDFF <sub>H</sub>	1.5 KW	–	–	unused
FE00 <sub>H</sub> -FFFF <sub>H</sub>	0.5 KW	0	1	Internal YRAM

1) For accessing the external data memory 0 to 7 waitstates can be selected. This leads to three to ten more DSP cycles for external data read and write.

2) If the DSP tries to write to the circular buffer in the same time that the PEDIU uses it, the PEDIU has higher priority. In this case, up to four more DSP cycles will be added to the access.  
The User should dedicate 0.5 KWord of Program RAM for the monitor (the routine used by the emulator).

### 2.7.3 Control of External Memories / Registers

The external data and program memories are accessed through a multiplexed data and program bus. This bus includes a 16-bit address bus (CA) and 16-bit data bus (CD). The DCU is generating the read and write controls for these memories and controls the input and output pads.

The external bus is usually used for fetching program instructions, therefore it's defined with program priority. It means that every external bus sequence starts with a program fetch (always zero wait states). If there is need for data access, the external bus sequence is extended to a minimum of 4 cycles in the following way:

- cycle 1 - Program fetch
- cycle 2 - IDLE1 cycle
- cycle 3 - Data access (can be extended up to 8 cycles = 7 wait states)
- cycle 4 - IDLE2 cycle

If instead of a program fetch there is a program write, the OAK receives three wait states (see program write diagram). The program write is always performed by the OAK using the MOVD instruction (it ensures that a program write will never be in parallel to a data access). The MOVD instruction is a four cycle instruction therefore, due to the extra wait

Functional Block Description

cycles, this instruction is actually performed in seven cycles in the DOC. If a MOVD instruction reads the data from an external data RAM, it's length will be increased in 3-11 cycles more, according to the external bus sequence.

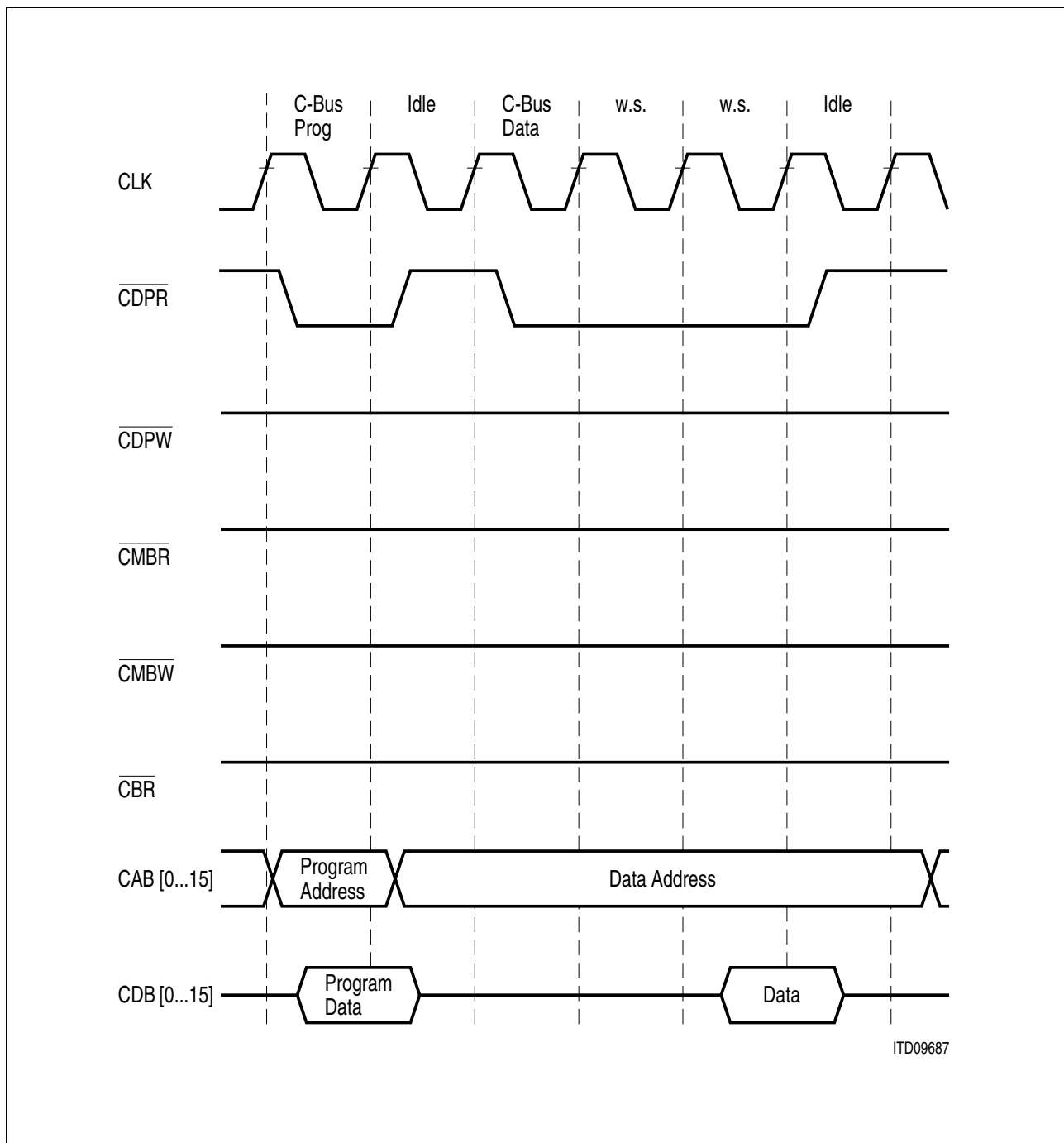


Figure 2-34 External Data/Program Read Access

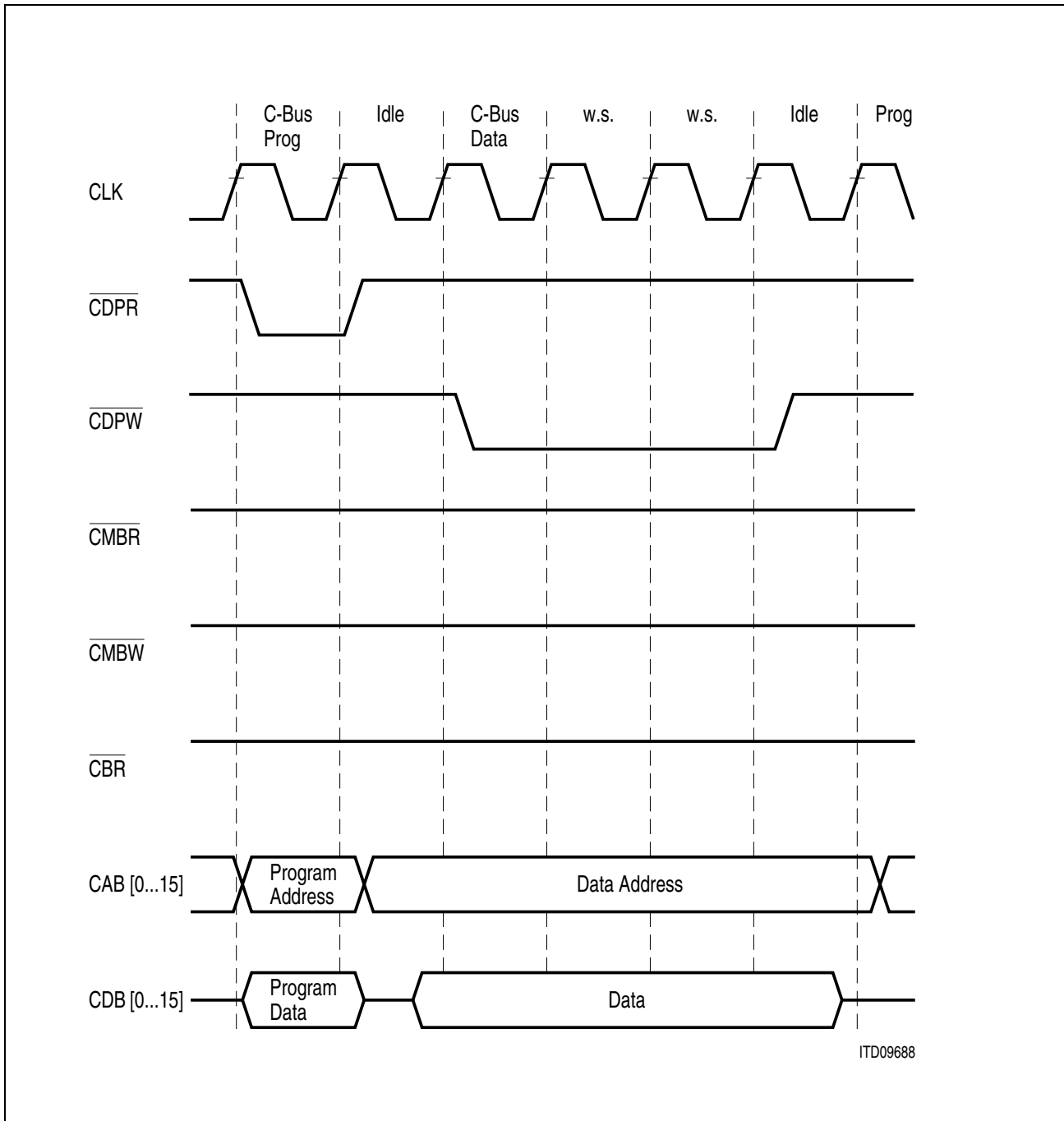


Figure 2-35 External Data Write Access

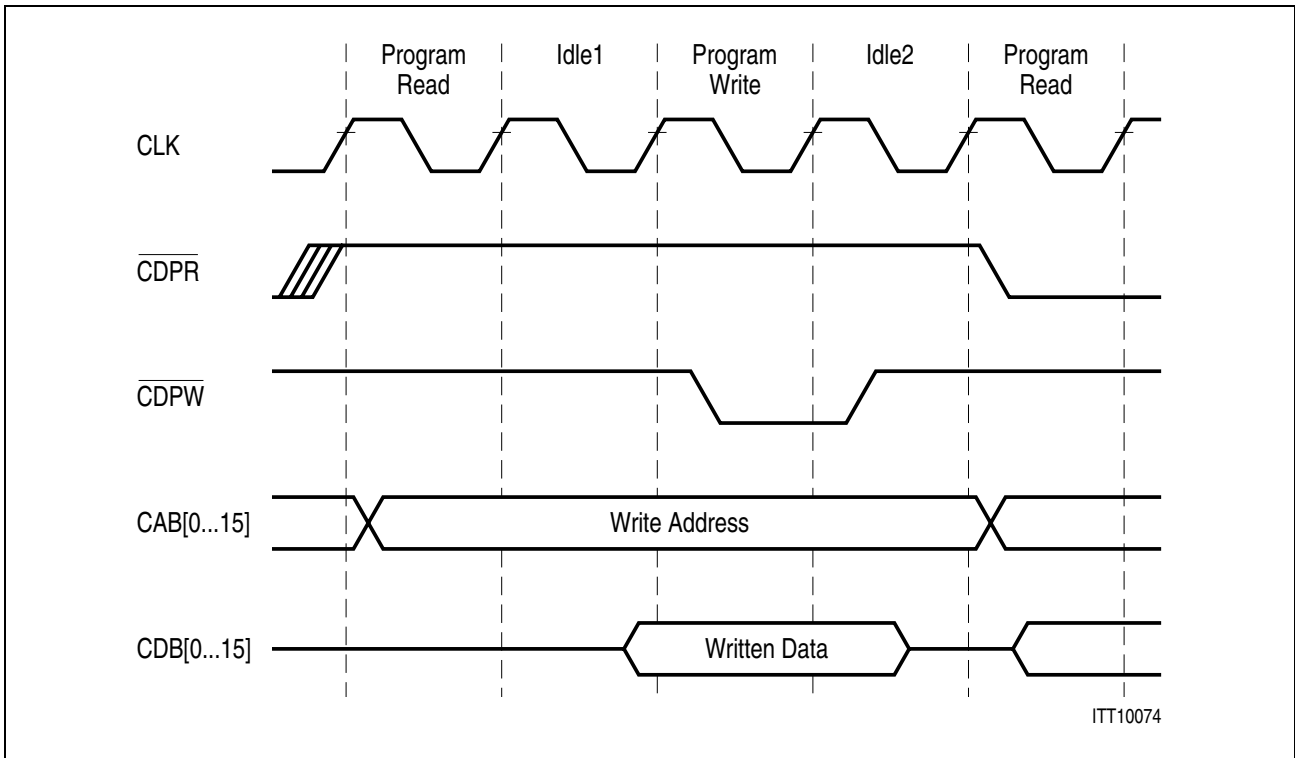


Figure 2-36 External Program Write Access Due to MOVD



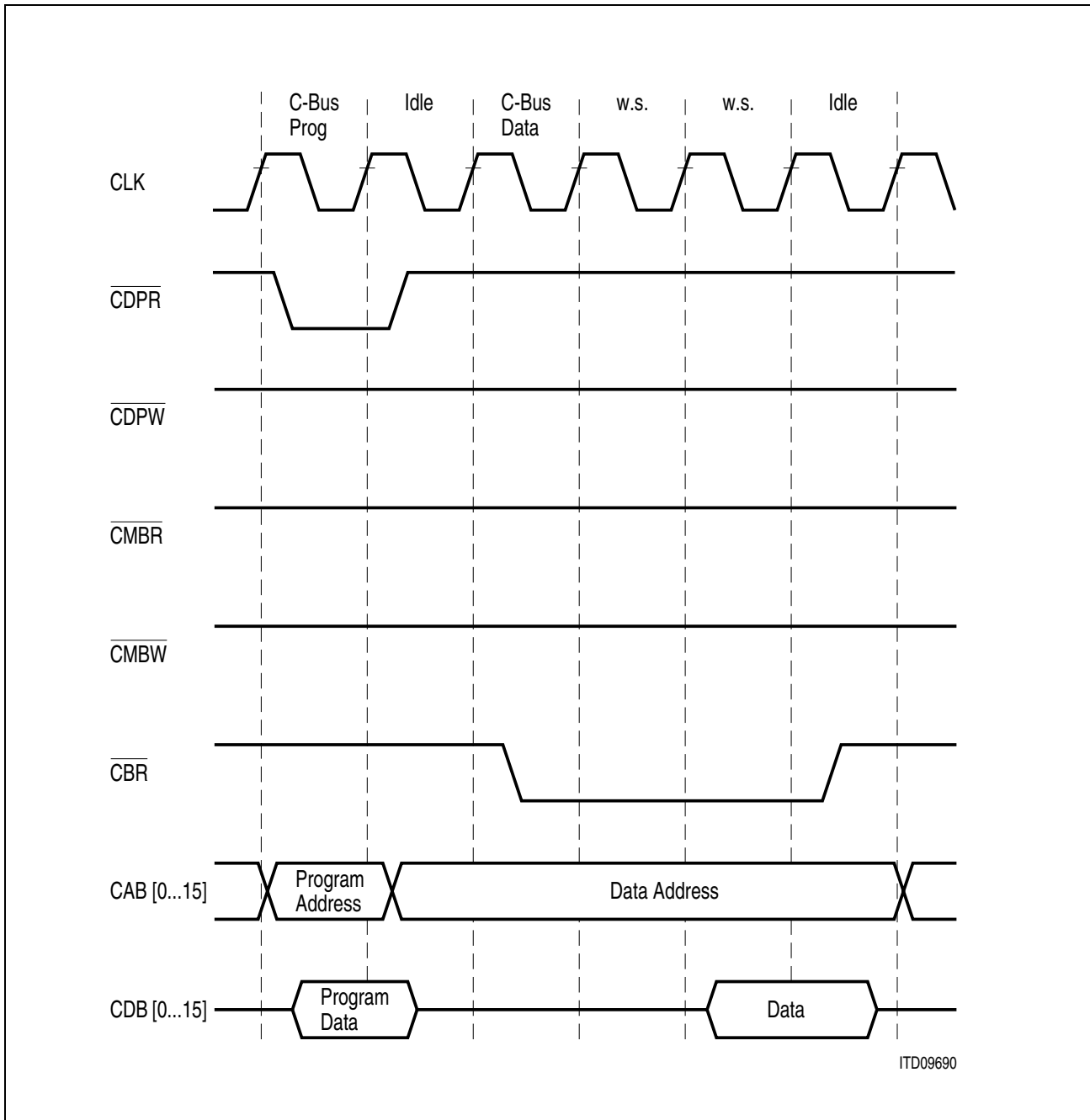


Figure 2-37 External Boot ROM Read

### 2.7.3.1 Memory Configuration Register (MEMCONFR)

The memory configuration register, MEMCONFR, is a memory mapped register, at address C002<sub>H</sub>, which controls the memory interface. In the following are its bit assignments:

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>MEMCONFR</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DWS	

*Note: Bits 15...3 are unused and are read as '0'.*

**DWS** - sets the number of wait states on external data space (0-7).

DWS can be read or write by the DSP.

After reset it is set to 0007<sub>H</sub>.

#### WAIT-STATES

##### External Data Space

The wait-state generator is capable of generating programmable number (0-7) of w.s. when accessing the external data. The w.s. number can be programmed by setting the DWS field, in the MEMCONFR register, to the number of the desired w.s. When DWS = 0 no wait states will be implemented. After reset the DWS is set to 7.

##### External Boot ROM (can be on CDI)

The access to the EPROM, during stand-alone boot, is with the same number of wait-states as written in DWS.

### 2.7.3.2 Test Configuration Register (TESTCONFR)

The test configuration register is memory mapped register, at address C003<sub>H</sub>

bit	15	14	13	12	11	10	9	8
<b>TESTCONFR</b>	PWB	OAKE	CIRE	P3E	P2E	P1E	P0E	0

bit	7	6	5	4	3	2	1	0
<b>TESTCONFR</b>	0	0	0	0	0	0	0	0

*Note: Bits 8...0 are unused and are read as '0'.*

**PWB** Used as RAM'S Power Down Bypass. This bit can be read/write by the SW. Cleared after reset.

**OAKE** Memory Enable of the OAK. This bit is the value of the OAK memory enable signal (PMEMENP) in the previous cycle. It enables observability of this signal. This bit is read only.

**CIRE** Circular buffer enable. This bit is the previous cycle value of the circular buffer enable signal which is combination of the OAK PMEMENP and the PEDIU request because they both can use the RAM. This bit is read only.

**P3E, P2E, P1E, P0E** Program RAM Enables reflect the value of the BSN input of the four Program RAMs for enabling observability. If the BSN inputs of the Program RAMs are stuck to zero, the RAM will work properly and the only effect will be more current.

### 2.7.4 Emulation Support

The CDI board communicates with the COMBO with two main interfaces:

- Control pins (BOOT, DBG, ABORTN, RSTN).
- C-bus interface (for Program RAM, Mail Box and Boot EPROM)

The control strap pins are inserted to the DCU which is responsible to send the right signals to the OCEM module and the OAK Emulation interface pins.

The communication protocol between the debugger and the monitor program is not protected from the case where user program, by mistake writes to the monitor space.

In such a case the monitor routine will be corrupted with no way to recover.

A write to the emulation mail box is prevented except for the following cases:

- During monitor routine (a write which is done by the emulator).
- During boot routine execution (code download).

### 2.7.5 Interrupt Handling and Test Support

The OAK user interrupts (INT0, INT1, INT2, NMI) are asserted by the DCU.

Usually, the sources for the interrupts are the functional blocks but for testing, there is a way of inserting the interrupts from DOC input pins. Therefore, the interrupt sources are as follows:

(TIF If the Test Flag - see the TESTCONF register).

**Table 2-17 Interrupt Map**

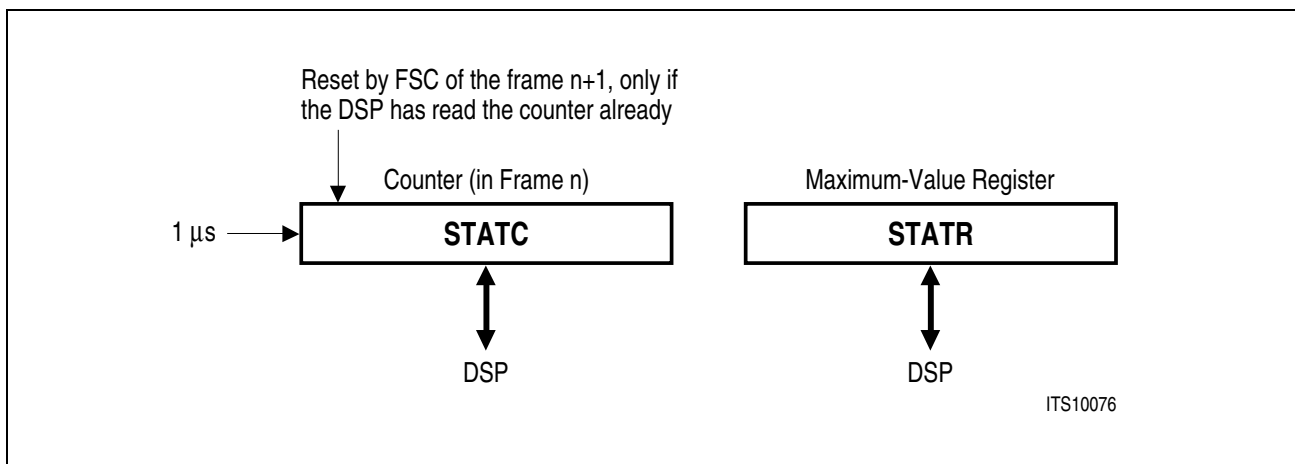
Interrupt	Source (TIF = 0)	Source (TIF = 1)
INT0	PCM-DSP interface = FSC	AD8
INT1		AD9
INT2	μP Mail Box	CS
NMI		WR
BI	OCEM	Abort
Wait	DCU	AD7

### 2.7.6 Run Time Statistics Counter and Register (STATC and STATR)

The DSP performs its activities in a cyclic manner, which start with frame sync clock (FSC). It must finish the execution of the current activities before the beginning of the next frame sync.

The DSP run time statistics is used by the user to estimate the work load on the DSP. By using this HW, the user can find very accurately the maximum time spent by the DSP from the FSC until it finished its job.

The DSP statistics include an eight bit counter STATC which is counting up every 1  $\mu$ s.



**Figure 2-38**

Usually, the STATC is reset when there is a frame sync (FSC) rising edge. When the DSP finishes its tasks it reads the STATC value.

The time between two consecutive frame syncs is always 125  $\mu$ s, therefore, if the DSP is working properly, the counter value should always be less than 125.

If the DSP failed to read the counter value and a new FSC rising edge has arrived, the counter is not reset. Therefore, the DSP would read a value greater than 125. It means that the DSP failed to finish its tasks within the time frame of 125  $\mu$ s.

The STATR register is added for helping the user to perform the statistics. STATR is a general purpose 8-bit read/write register (its 8 most significant bits are always read as '0').

The user program should perform statistics in the following way:

- The STATC is reset upon detection of FSC rising edge.
- The DSP finishes its activities and reads the value of STATC and STATR.
- The DSP compares STATC to the previous maximum value saved in STATR.
- If the new value is larger, it should be written to STATR.

The system programmer can get the counter value via  $\mu$ P-Mail Box and thus can change the DSP program (For example, more conferences may be implemented).

**Registers Definition:**

**STATC**

Memory mapped address C004<sub>H</sub>.  
read only counter.

bit	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
bit	7	6	5	4	3	2	1	0
	STATC7	STATC6	STATC5	STATC4	STATC3	STATC2	STATC1	STATC0

**STATC7...0** Statistics Count.

The 8 MSBs are unused and read as '0'.

Reset upon detection of FSC edge if STATC was read by the DSP since the last FSC.

Unchanged upon chip reset

**STATR**

Memory mapped address C005<sub>H</sub>.

8-bit read/write general purpose register.

bit	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
bit	7	6	5	4	3	2	1	0
	MSC7	MSC6	MSC5	MSC4	MSC3	MSC2	MSC1	MSC0

**MSC7...0** Max Statistics Count.

The 8 MSBs are unused and read as '0'.

Unchanged upon chip reset.

### 2.7.7 Program Write Protection Register (PASSR)

If the user writes by mistake to the program RAM, the DSP program may collapse. A protection is provided by means of a password which the user must write to the password register (PASSR) before writing into the program RAM. PASSR must be written with the value  $F236_H$  for enabling a program write. If the value in PASSR is different from  $F236_H$ , the  $\overline{CDPW}$  (external program RAM write) is not issued during a MOVD instruction.

Memory mapped address  $C006_H$ .

16-bit read/write register

bit	15											0
<b>PASSR</b>												

Reset value -  $0000_H$

*Note: The boot routine automatically sets the value of PASSR to  $F236_H$  for enabling the download of data to the program RAM. After the download, the boot routine writes  $0000_H$  to PASSR for enabling the program protection. This mechanism is bypassed during trap/bi handler in order to enable the monitor to change the program.*

### 2.7.8 Serial (via JTAG) Emulation Configuration Register (JCONF)

This register determines the emulation configuration.

Memory mapped address  $C007_H$ .

2-bit read/1 bit write register

bit	15	14	13								0
<b>JCONF</b>	ABORTI	ACTIVE									

Reset value -  $8000_H$  (except bit 14 which is determined by strap)

**ABORTI** This bit is read/write. When set to '1' the abort is as in "parallel emulation" (i.e. via  $\overline{ABORT}$  pin). When resets to '0' the abort function is implemented via TDI pin (see SEIB chapter for more details).

**ACTIVE** This bit is read only, it gets the inverted value of bit 12 of c-bus ( $\overline{CDB12/SEIBDIS}$ ) at the rising edge of  $\overline{DRESET}$ , when it used as a strap. When '1' it means that SEIB is active (i.e. emulation is serial via JTAG). When '0' SEIB is not active, i.e. emulation is parallel.

### 2.7.9 Boot Support

The boot is a process which loads the external program RAM with the DSP program. There are three methods of booting the system:

- Emulation boot  
A boot sequence which loads the monitor (BI routine) to the program RAM. This routine enables the PC emulator to control the DSP.
- ROM boot  
Downloading code directly from an external boot ROM to the program RAM.
- $\mu$ P boot  
A boot sequence which loads the Program RAM from the  $\mu$ P through the  $\mu$ P mail box.

The boot is controlled by a boot routine which resides in the internal DSP program ROM. This routine is starting to be executed upon chip reset according to the condition of the strap pins:

**Table 2-18 Boot Mode**

	Boot Strap	DBG Strap	ROM Strap	
No boot	0	Don't care	Don't care	Usual reset which starts fetching from the address 0000 <sub>H</sub> of the external program RAM.
Boot	1	1	0	Emulation Boot
Boot	1	0	1	Rom Boot
Boot	1	0	0	$\mu$ P Boot
Not defined	1	1	1	Forbidden

*Note: When the strap pins, CDB0/BOOT, CDB1/DBG and CDB2/ROM, are not driven externally during reset, they are driven internally by an internal pull-downs. If a fixed external pull-up is applied on a strap, a pull-up resistor of 5 K $\Omega$  is required.*

#### 2.7.9.1 Boot Sequence

The boot always starts immediately after reset with an execution of the instruction Brr-3 by the OAK. Due to the fact that during reset the OAK PC register points to address 0000<sub>H</sub>, the Brr-3 instruction performs a jump to address FFFE<sub>H</sub>.

This address is on the top of the program ROM. At this address there is a branch instruction to the beginning of the boot routine (Br FF00<sub>H</sub>).

At the beginning of the boot, the status of the strap pins is checked by reading the contents of the BOOTCONF and JCONF register.



**Functional Block Description**

According to the strap pins, the download of the program RAM starts, and also the type of emulation (in case of emulation boot i.e. parallel/serial).

**2.7.9.2 Emulation Boot**

The emulation boot supports the down-loading of software code from an external dual ported RAM (Mail Box buffer), loaded earlier by the debugger host, to the program RAM. Two specified addresses in the Mail Box have to be with following data

**Table 2-19**

Address	Contents	Value
0xF400	The number of words to be loaded.	
0xF401	The first program RAM address to load to.	

**2.7.9.3 Boot Procedure**

The debugger should activate the BOOT and DBG strap pins, on reset negation. As a result, the program jumps to the boot routine. The Mail Box is accessed (read) using the data memory control signal. The software uses the movd command of the OAK to write to the program RAM. The last instruction of the boot routine is TRAP to pass control to the monitor program.

**2.7.9.4 Boot ROM**

The purpose of this mode is to down-load the application program from a slow memory device to the program RAM in order to execute the program after boot from the RAM with zero w.s. The external hardware will include EPROM or ROM which will be connected to the CBR signal and a program RAM which will be connected to the  $\overline{CDPW}$ .

The specified address in the EPROM will include:

- Control word
- The number of words to be loaded.
- The first program RAM address to load to.

If the BOOT and ROM strap pins are active during reset negation and the DBG is not active, the boot routine starts performing the Boot ROM download. The EPM bit in the BOOTCONF register, is set by the HW. As a result, all read transactions, in movp instructions, will be from the BOOT EPROM which means that the boot EPROM will be located in the program address space. The movp instruction will put the data in a temporary location in the data memory, from which it will be transferred to the external RAM using movd instruction.

The ROM data structure is illustrated bellow, this structure will enable the user to down-load a few blocks of code from the EPROM into different locations in the

**Functional Block Description**

destination RAM. The control word will tell the down-loading program whether to finish loading (FFFF<sub>H</sub>) or to continue with the next block (0000<sub>H</sub>) except for the first block where the control word will contain program RAM address to jump to at the end of the down-loading.

At the end of the down - load procedure the EPM bit must be cleared by the boot routine, by writing to it. As a result, resuming to the normal memory mapping and the regular control signal will be generated to the external data memory. The next instruction will be a branch to the address, specified in the EPROM first address.

**Table 2-20 EPROM/ROM Data Structure**

Start address
Number of words
RAM address
Data to be loaded
Control word
Number of words
RAM address
Data to be loaded
Control word
Number of words
RAM address
Data to be loaded
Control word

**2.7.9.5 μP Boot**

The μP boot procedure is as follows:

- OAK boot routine: write to OCMD (mail box command register) the command “start loading program RAM”
- μP: As a result, the μP writes the μP mail box command “start boot procedure”.
- OAK boot routine: clears the MBUSY bit
- μP: performs the command “write program memory command”.
- OAK boot routine: performs the write to the program RAM and verifies that the data was written correctly. Then the MBUSY is cleared.
- μP: performs the next write command.

This process continues until all the code is loaded. Then, instead of a write command, the μP issues the command: “Finish boot procedure”. As a result, the OAK boot routine clears the MBUSY bit and branches to address 0000<sub>H</sub>.

**Functional Block Description**

If something went wrong during the download, the OAK sends the command “Error during boot” and then waits for the command “Start boot procedure from the  $\mu$ P”. When receiving this command, the OAK boot routine restarts the boot procedure and waits for the “write program memory command”.

**2.7.9.6 Mail Box Instructions Format**

For enabling the  $\mu$ P boot, there are some predefined opcodes used by the boot routine. These opcodes are valid only for the boot routine. When the user program is executed later, these opcodes have no affect.

There are two separate sets of opcodes, one for the  $\mu$ P mail box and the other for the OAK mail box.

**$\mu$ P Opcodes**

**Finish Boot Procedure**

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

Description: Finish the boot procedure (the  $\mu$ P has finished loading the program RAM).

**2.7.9.7 Write Program Memory Command**

Description: Write the content of registers MDT1- MDT<sub>NUM</sub>(NUM=1...5) to the program RAM address (MDT0) –(MDT0 +NUM –1), consecutively.

0	0	1	0	1	N	U	M
---	---	---	---	---	---	---	---

Operation: Write to program memory from address within MDT0 up to (MDT0 + NUM –1). The data to be written is taken from MDT1 to MDT5. The content of MDT1 is written to the address which resides within MDT0, the content of MDT2 is written to the address (MDT0 +1), and so on. Notice that allowed values of NUM are 1...5.

Example: MDT0 = 20A0<sub>H</sub> opcode 00101011 (NUM = 3). It will result with:  
 Program RAM address 20A0<sub>H</sub>  $\leq$ MDT1  
 Program RAM address 20A1<sub>H</sub>  $\leq$ MDT2  
 Program RAM address 20A2<sub>H</sub>  $\leq$ MDT3

*Note: The program RAM is write protected. It can be written only if the write is enabled by writing a special code to a register within the DCU (see the DCU description for more details).*

### Start Boot Procedure

0	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---

Description: Start the boot procedure.

Used for starting the boot procedure (as response to the OAK command “start loading program RAM”) and for restarting the boot if something failed during the boot (as response to the OAK command “Error During boot”).

### 2.7.9.8 OAK Opcodes

The OAK opcodes enable the OAK to send commands to the  $\mu$ P.

#### Start Loading Program RAM

Description: This command requests the  $\mu$ P to start loading the program RAM. It is used by the boot routine to initiate the load of the program RAM.

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

*Note: The INT2 vector is used by the boot routine in a pending method (reading the ST2), therefore, the INT2 interrupt vector can be written without influencing the boot procedure.*

#### Error During Boot Procedure

Description: This command requests the  $\mu$ P to stop the download due to an error (which may be fixed by starting the download from the beginning).

0	1	1	1	0	E	R	R
---	---	---	---	---	---	---	---

The ERR field is the error status as follows:

- 000** A  $\mu$ P command different then nop, finish boot or write program arrived.
- 001** The data written to the program was not read back (result of a test of the first few loaded words to the program RAM).

### 2.7.9.9 Boot Configuration Register (BOOTCONF)

The boot configuration register, BOOTCONF, is a memory mapped register at address C001<sub>H</sub>

bit	15	14	13	12	11	10	9	8
<b>BOOTCONF</b>	EPM	*	*	*	*	*	*	*

bit	7	6	5	4	3	2	1	0
<b>BOOTCONF</b>	*	*	*	*	*	*	*	*

\* written as “don’t care”, read as ‘0’

**EPM** This bit indicates that the chip is in EPROM boot mode. It is set when BOOT strap pins are active, the DBG strap pin is inactive and the ROM strap is active on reset negation. This bit can be read/write by the user. While EPM bit is set, every read during MOVP instruction is done from the boot ROM instead of the program space.

#### 2.7.9.10 The Bootroutine

Following is the bootroutine which resides in the internal bootrom.

```
.FORMAT 10000,1000
.EQU IOPAGE          0xc0      ; Memory mapped I/O page
.EQU BOOTCONF        0x01      ; BOOTCONF register offset address
.EQU MEMCONF         0x02      ; MEMCONF register offset address
.EQU TESTCONF        0x03      ; TESTCONF register offset address
.EQU STATUS1         0xf7fe    ; OCEM STATUS1 register
.EQU EMUMB           0xf400    ; Emulation mail box address
.EQU OcemTraceBuff   0x0010    ; Ocem Trace Buffer length
.EQU OCEMadd         0x45      ; <<<< TEMPORARY NUMBER
.EQU Version         0xd0c0    ; chip Version
.EQU MONITOR         0x7c00    ; monitor address (31K-32K)
.EQU BOOTCONFadd     (IOPAGE<<8) | BOOTCONF
.EQU BR_CODE         0x4180    ; Branch opcode
.EQU OCMD            0x50      ; OAK mail box command register
.EQU OBUSY           0x51      ; OAK mail box busy bit
.EQU ODT0            0x52      ; OAK mail box data reg 0
.EQU MCMD            0x40      ; micro-processor mail box command
                             register
```

**Functional Block Description**

```
.EQU MBUSY          0x41      ; micro-processor mail box busy bit
.EQU MDT0          0x42      ; micro-processor mail box data reg 0
.EQU MDT0_ADD     0xc042    ; micro-processor mail box data reg 0
.EQU MDT1_ADD     0xc044    ; micro-processor mail box data reg 1
.EQU MB_NOP       0x0       ; mail-box NOP
.EQU FINISH_BOOT  0x7       ; mail-box Finish boot command
.EQU START_BOOT   0x55      ; mail-box Start boot procedure
                           command
.EQU WRITE_PROG   0x28      ; mail-box write program
.EQU START_LOAD   0x7       ; mail-box start prog. load command
.EQU MCMD_ERROR   0x70      ; Error in micro-processor command
.EQU PROG_ERROR   0x71      ; Program data verification failed
.EQU PASSR_REG    0xc006    ; PASSR register address

.CODE S_Main
.USE S_Main
.ORG 0xff00
bootrtn:
;#####
; Initialization
; The mail box on the CDI is a slow memory, therefore, wait
; states are needed. If the DOC will run in a frequency higher
; than 40 Mhz 7 WS will be needed. Therefore, the default of
; 7 WS in DWS field of MEMCONF is not changed.
    lpg    #IOPAGE
    mov    #0x0,r5          ; r5=0
    mov    ##0x200,sp      ; Put a default value in SP
    mov    ##PASSR_REG,r1  ; PASSR register address
    set    ##0xf236,(r1)   ; disable program protection

;#####
; This check decides what kind of boot is it.
; If it's a ROM boot. EPM bit is active. EPM is the MSB
; of the BOOTCONF, therefore, after reading it to the
; accumulator, the sign extension will create a negative
; value if EPM is set (M flag will be activated)
    mov    BOOTCONF,a0     ; Read the EPM bit.
    br     rom_boot,lt
```

**Functional Block Description**

```

; If the DBG flag in the OCEM is active, it is an emulation
; boot. The dbg flag is in STATUS1 register, bit 15 (MSB).
; Therefore, if it is set, a transfer to the accumulator will
; make the accumulator negative.
    mov    ##STATUS1,r0      ; STATUS1 address to r0
    mov    (r0),a0          ; Read the DBG bit.
    br     emu_boot,lt

#####
; micro-processor boot
;
; After the read of the micro-processor mail-box, the MBUSY bit
; is set and then the micro-processor is loading the next data.
; In parallel to this load, the boot routine verifies that the
; data loaded to the program RAM is correct.
    load  #0x2,stepi        ; Step1 = 2
    mov   ##START_LOAD,r0  ; Code of start load command
    mov   r0,OCMD          ; command to OCMD register

; The micro-processor should respond with 'Start boot procedure
; command'.
; The 'start boot procedure' command is added always but it is
; really needed in a special case. It is when the download
; verification fails. In this case, the boot has to restart
; but meanwhile, the micro-processor has started writing the
; next data and may write the write command. Therefore, in
; this case the OAK waits until the micro-processor receives
; the error command and reacts with the 'start boot procedure'
; command.

int2_poll1:
    rep   #0x6              ; Wait until INT2 is reset after
    nop                               ; a clear of busy bit
    tstl  ##0x2000,st2       ; check if int2 is set
    brr   int2_poll1,neq     ; Wait for int2
    mov   MCMD,a0           ; Read micro-processor command
    mov   r5,MBUSY          ; Clear busy bit

```

Functional Block Description

```

cmp    ##START_BOOT,a0    ; Check if it's "start boot"
brr    int2_poll1,neq
brr    int2_poll2

```

clear\_busy:

```

mov    r5,MBUSY           ; Clear busy bit
rep    #0x6
nop                                     ; Delay for making sure that
                                     ; the int2 signal was reset
                                     ; due to the clear of MBUSY

```

int2\_poll2:

```

tstl   ##0x2000,st2      ; check if int2 is set
brr    int2_poll2,neq    ; Wait for int2
mov    MCMD,a0           ; Read micro-processor command
mov    MCMD,a1           ; Copy also to a1
and    #0xf8,a0          ; Clear 3 LSB of a0
cmp    ##0x28,a0         ; Check if it's Write mem. command
brr    write_mem,eq
cmp    #0x7,a1           ; Check if it's finish boot
br     finish_eboot,eq   ; goto emulation boot finish
mov    ##MCMD_ERROR,r0   ; Error has occurred if reached here
mov    r0,OCMD
brr    clear_busy

```

; a1 include the micro-processor command contents. The 3 LSB are  
; the number of data registers to be loaded. The first data  
; register (MDT0) contains the address of the first write data.

write\_mem:

```

mov    #0x0,r3           ; First address of XRAM
and    #0x7,a1           ; leave only the number of words to load
dec    a1                ; prepare a1 for the rep (number of loops
                        ; is a1 + 1)
mov    ##MDT0_ADD,r1     ; MDT0 address to r1
mov    (r1)+s,r4         ; mov start load address from MDT0
                        ; to r4 and advance r1 to MDT1

```



**Functional Block Description**

```

mov    r4,a0                ; save load address in a0
bkrep  all,>%end_copy-1
mov    (r1)+s,y             ; copy mail box registers to XRAM
mov    y, (r3)+

```

%end\_copy:

```

mov    ##MDT1_ADD,r1       ; MDT0 address-value to r1
rep    all
movd   (r1)+s,(r4)+        ; Download to program RAM
mov    r5,MBUSY            ; Clear busy bit for enabling the
                                ; the micro processor to load
                                ; the next data

```

; The time until the micro-processor loads the next data is used  
; for verifying that the data was loaded correctly

```

mov    a11,r2              ; number of load words-1 to r2
modr   (r2)+               ; Number of load words in r2
mov    #0x0,r3             ; First address of XRAM

```

check\_ram:

```

movp   (a01),a1            ; read loaded program RAM
cmp    (r3)+,a1            ; Compare to saved mail box in XRAM
brr    check_err,neq      ; Data in Prog. RAM not equal to MB reg.
inc    a0                  ; Point to next Prog. RAM address
modr   (r2)-
brr    check_ram,nr
brr    int2_poll2

```

; An error has occurred during program RAM verification

check\_err:

```

mov    ##PROG_ERROR,r0     ; Program data error has occurred
mov    r0,OCMD
brr    int2_poll1         ; Wait for resume of boot procedure
                                ; by the micro processor 'start boot
                                ; procedure command'

```

; Finish the micro-processor boot

Functional Block Description

```

finish_eboot:
; Enable program write protection
;*****
    mov    ##PASSR_REG,r1    ; PASSR register address
    mov    #0x0,r2          ; r2 = 0
    mov    r2,(r1)          ; enable program write protection

    mov    r5,MBUSY         ; Clear busy bit
    br     0x0

;#####
; ROM boot
;
; During ROM boot the MOVDP command is reading from the Boot
; ROM.
rom_boot:
; Read the ROM parameters:
;*****
    mov    #0x0, a11        ; ROM first address
    movdp (a11), lc         ; Start address to jump to at the end.
    mov    #0x0,r3         ; First address of XRAM

rom_load:
    inc    a1
    movdp (a11), r2        ; Number of words to load.
    inc    a1
    movdp (a11), r4        ; RAM address to load to.
    inc    a1

; Load user program from ROM to the program RAM:
;*****
    mov    a11, r5         ; load the pointer

rom_block:
    movdp (r5)+, (r3)      ; down load one word
    movdp (r3),(r4)+
    modr (r2)-            ; Decrement number of words

```

Functional Block Description

```

brr    rom_block,nr
mov    r5, a11
movp   (a11), a01      ; Load the control word
brr    rom_load, eq    ; if equal to zero load the next block
rst    #0x8000,BOOTCONF; Reset the EPM bit in BOOTCONF
                                register

```

; Enable program write protection

;\*\*\*\*\*

```

mov    #PASSR_REG,r1    ; PASSR register address
mov    #0x0,r2          ; r2 = 0
mov    r2,(r1)          ; enable program write protection
mov    lc, pc           ; Branch to the address specified in the
                                ; EPROM parameters
nop                                ; NOP 1 - Has to be added after mov to pc
nop                                ; NOP 2 - Has to be added after mov to pc

```

;#####

; EMULATION mode

;

; The emulation boot loads the monitor routine for the emulator.

; The start addresses of the mail box are as follows:

; F400 - Number of words to load

; F401 - The first program RAM address to load to.

emu\_boot:

```

mov    #EMUMB, r0       ; Emulation mail box address
mov    (r0)+, r3        ; Number of words to load.
mov    (r0)+, r5        ; RAM address to load to.

```

; Load nop to the reset vector

;\*\*\*\*\*

```

mov    #0x0, r1
mov    #0x0, r4         ; 0 is NOP opcode
mov    r4,(r1)         ; Put 0 in Xram address 0
movd   (r1),(r4)+
movd   (r1),(r4)+

```

Functional Block Description

```

; Load the branch instruction opcode to BI/TRAP vector
;*****
    mov    ##BR_CODE, r2
    mov    r2, (r1)
    movd   (r1), (r4)+
    mov    r5, (r1)
    movd   (r1), (r4)+

; Load the monitor program from MailBox to the program RAM:
;*****
    modr   (r3)-          ; loop count
    rep    r3
    movd   (r0)+, (r5)+

; Push on the stack the following information:
;*****
    push   #OCEMadd
    push   #OcemTraceBuff
    push   #BOOTCONFadd
    push   #Version

; Enable program write protection
;*****
    mov    ##PASSR_REG, r1    ; PASSR register address
    mov    #0x0, r2          ; r2 = 0
    mov    r2, (r1)          ; enable program write protection

    trap

;.ORG 0xffff8
    nop                ; should be fff8
    nop                ; should be fff9
    nop                ; should be fffa
    nop                ; should be fffb
    nop                ; should be fffc
    nop                ; should be fffd
    br     bootrtn

```

**2.7.10 Sine Table ROM**

The Sine Table ROM is a  $256 \times 16$  bit word ROM, which resides in the DSP program space, between the addresses:  $FE00_H$  -  $FEFF_H$ . It contains 256 samples of one sine cycle. These samples were taken every fixed step of  $2\pi/256$ , and their value is represented in fixed-point.

*Note: For details please refer to the Application Note "How to use the DOC Sine Table".*

## 2.8 PCM-DSP Interface Unit (PEDIU)

### 2.8.1 General Description

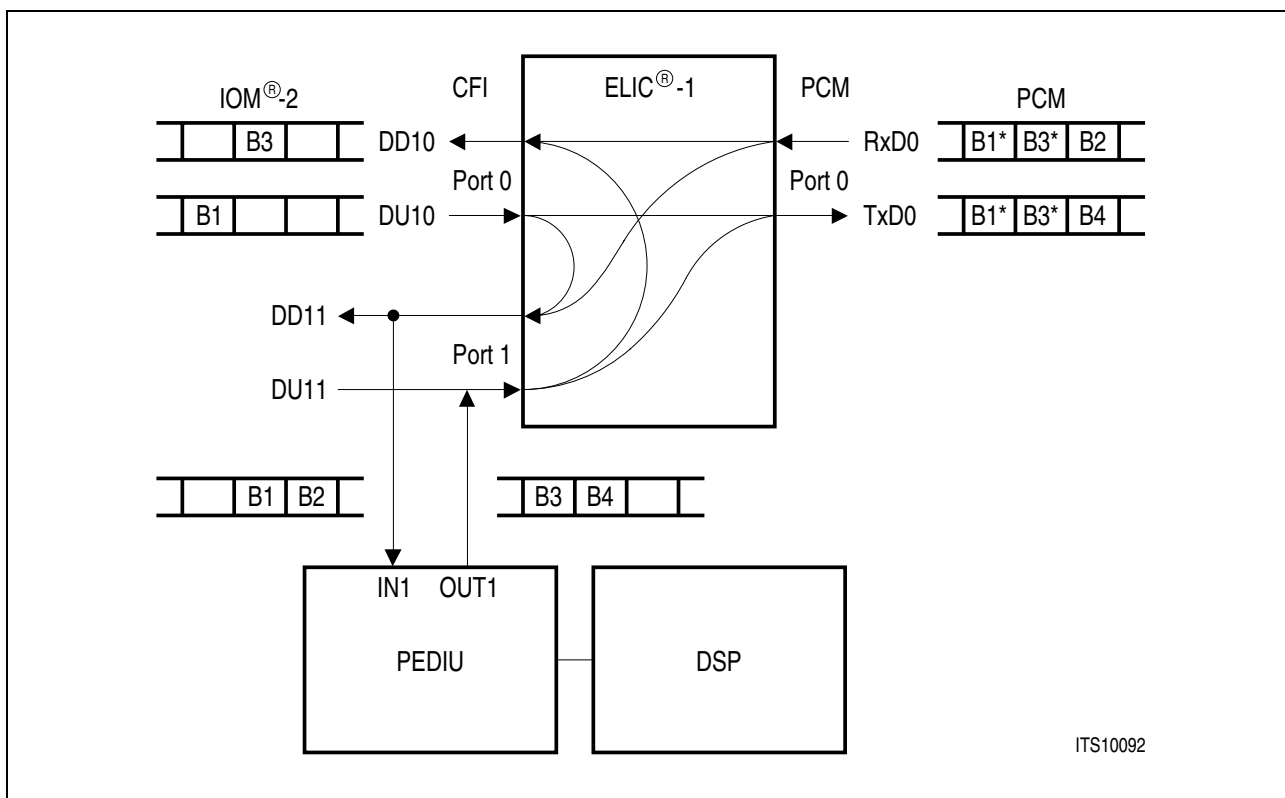
The PEDIU gives the DSP access possibility to 64 down-stream PCM B-channels, that come from the two ELICs, and the possibility to drive upstream 64 PCM B-channels that go back to the ELICs. In general, the PEDIU feeds a bidirectional circular buffer with B-channels. They are written into the buffer by the PEDIU and read by the DSP in downstream direction, and written into the buffer by the DSP and read out by the PEDIU in the upstream direction.

A possible flow of B-channels between ELIC1 and PEDIU Port1 is demonstrated as an example in **Figure 2-39**:

- The B-channel B1 coming from IOM-2 interface on DU00 is switched by ELIC1 via internal loop to CFI Port1 DD11 line. Each loop requires one time-slot on the PCM highway (B\*).
- The B-channel B2 coming from PCM highway on the RXD0 line is switched by ELIC1 to CFI Port1 DD11 line.
- The PEDIU reads the B-channels, B1 and B2, in.
- The PEDIU writes the B-channels B3 and B4 out on the DU11 line.
- ELIC1 switches B3 to the IOM-2 interface on DD00 via its internal loop.
- ELIC1 switches B4 via TXD0 to the PCM highway.

*Note: Every above ELIC loop utilizes one time-slots on the PCM highway (i.e. B1\* and B3\*). If the ELIC is programmed to be in high impedance state during those time-slots, they (B1\* and B3\*) can be used for other purpose (e.g. for signaling by a SACCO).*

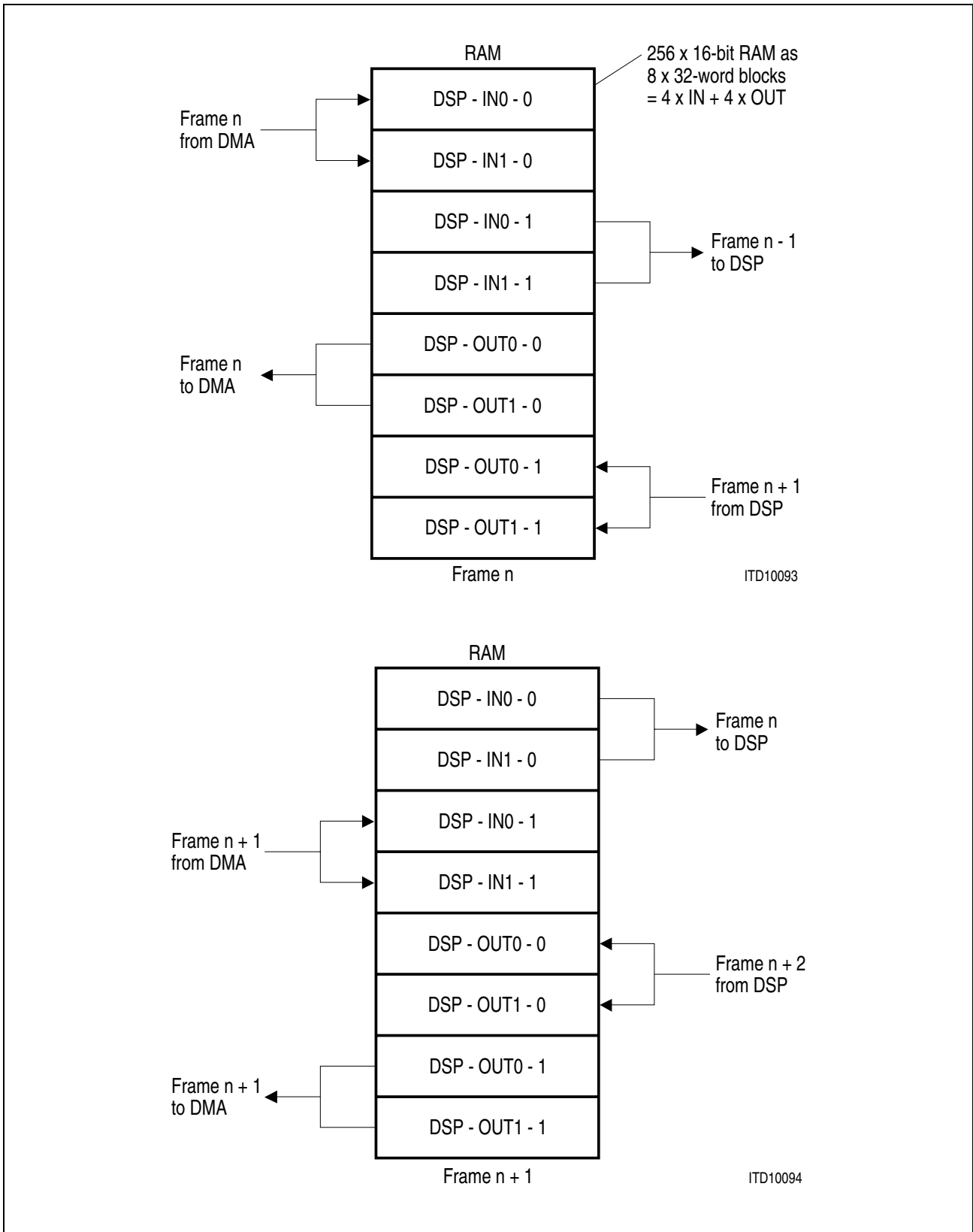
The PEDIU is connected with both ELICs: With Port0 of ELIC0 and with Port1 of ELIC1.



**Figure 2-39 Example: Flow of B-Channels between ELIC1 and PEDIU**

Two data streams with 32 successive time-slots each, or one data stream of 64 time-slots, come every 125  $\mu$ s from EPIC0 and EPIC1 in data downstream direction. They are converted into 8-bit parallel values, or into 16-bit a-/ $\mu$ -law linear values, and stored via a DMA controller in a RAM (the circular buffer). The RAM contains 128 words for received data (DSP-IN) configured in 4 IN blocks, and 128 words for data to be transmitted back to the upstream (DATA-OUT), configured in 4 OUT blocks. While the DMA controller writes the data of the current downstream frame (frame n) into one half of DSP-IN block, the DSP accesses the second half of DSP-IN, which contains B-channels values of the previous frame (frame n-1). At the same time the DMA reads data from one half of the DSP-OUT block into the upstream, while the DSP writes into the other half of DSP-OUT; the data to be read into the upstream in the next frame (frame n+1). All accesses to the circular buffer are synchronized by FSC (Frame Synchronization Clock) and by DCL (Data Clock). For more information on the process, which takes place every frame, see **Figure 2-40**.

Functional Block Description



**Figure 2-40** Accesses to the PEDIU RAM (circular buffer) at two Consecutive Frames



---

## Functional Block Description

The index of each frame ( $n$ ,  $n + 1$ , etc.) indicates when the frame would be written to the up stream or read from the down stream. The process described in the picture recurrences every 2 frames.

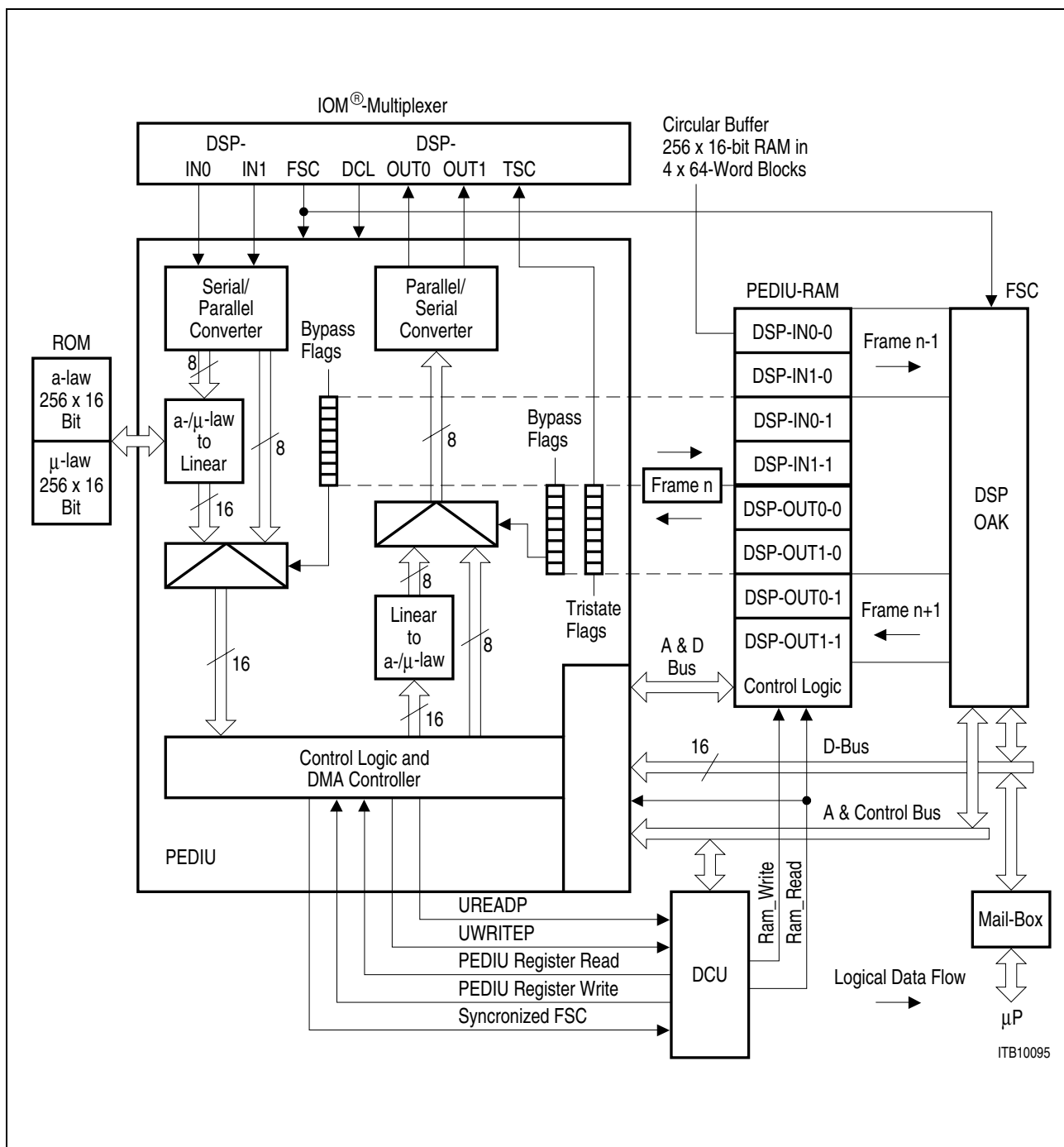
The PEDIU contains:

- A serial-parallel and an a-/ $\mu$ -law to linear converters for receive (IN) direction
- A linear to a-/ $\mu$ -law and a parallel-serial converter for transmit (OUT) direction
- A multichannel DMA controller for accessing the RAM
- A bypass and tri-state logic
- ROM, RAM and DSP interface

The bypass logic of the DSP-IN stream selects which B-channels will be converted into linear values, and which B-channels will bypass the a-/ $\mu$ -law to linear converter. Every consecutive 4 B-channels are controlled by one Bypass Flag in a special register. One 16-bit register controls 64 DSP-IN B-channels. The OUT streams values (8-bit coded or 16-bit linear) are controlled similarly in DSP-OUT direction. Another similar 16-bit register is dedicated for controlling the tri-state buffers of the two OUT streams (OUT1 and OUT2). This register defines which OUT-stream B-Channels are valid, and which are not. Here the resolution is also of 4 B-channels per bit. These three registers can be accessed by the DSP, for read and write.

The PEDIU-ROM is of 512 words. It contains the linear values of all the possible a-law and  $\mu$ -law values. It is used to convert words from a-/ $\mu$ -law to linear in the DSP-IN direction. The opposite conversion, from linear to a-/ $\mu$ -law in the DSP-OUT direction, is done by a special logic circuit.

Functional Block Description



**Figure 2-41 Block Diagram of the PCM-DSP Interface Unit (PEDIU)**

- Note: a) After reset, both converters (a-μ-law to linear and linear to a-μ-law) are enabled, and all the DSP-OUT time-slots are in tri-state.
- b) The PEDIU access priority to the circular buffer, is higher than access priority of the DSP. If the DSP tries to access the circular buffer concurrently, with the PEDIU, it will be delayed by a wait signal from the DCU (DSP bus control unit), till the end of the PEDIU's access.

## 2.8.2 PEDIU Internal Registers

The following section describes the PEDIU internal registers. The following details are specified for each register: name, function of each bit, registers schema, read and write addresses and method of write into the register or of read the content.

### 2.8.2.1 PEDIU Control Register (UCR)

This register determines the work mode of the PEDIU. It also determines whether the PEDIU is in a working or idle mode. The UCR has 5 bits.

After reset all bits are '0'.

Address: 0xC100

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	ACTIVEP	AMUL	M2	M1	M0

**M2...0** PEDIU Work Mode  
The definition of each bit can be seen in **Table 2-21**

**AMUL** a-law or  $\mu$ -law

**ACTIVEP** Active (Idle Not).

*Note: Bits 15...5 are unused and are read as '0'.*

A detailed description of the UCR follows on the next pages:

Functional Block Description

Table 2-21 Work Modes Specifications of the PEDIU

Mode Bits			Mode	Data Rate [Mbit/s]	DCL [MHz]	No. of input-streams	Time-Slots per frame (in each input stream)	Number of IN and OUT blocks in circular buffer (PEDIU RAM)	Notes
M2	M1	M0							
0	0	0	0	2	4 double clock	2	32	4 IN blocks 4 OUT blocks	IOM-2 Compatible
0	0	1	1	2	2 single clock	2	32	4 IN blocks 4 OUT blocks	
0	1	0	2	4	4 single clock	2	first 32 from 64 time-slots	4 IN blocks 4 OUT blocks	PEDIU can handle only first 32 time-slots of each input stream frame.
0	1	1	3	4	4 single clock	1	64	2 IN blocks 2 OUT blocks	PEDIU can handle all 64 time-slots of the frame, but only of IN-stream 0.
1	0	0	4	8	8 single clock	1	first 64 from 128 time-slots	2 IN blocks 2 OUT blocks	PEDIU can only handle the first 64 time-slots of the frame, and only of IN-stream 0.
1	0	1	5	PEDIU test mode. Used to test PEDIU ROM and PEDIU RAM. PEDIU IN and OUT streams are in idle mode.					

**Functional Block Description**

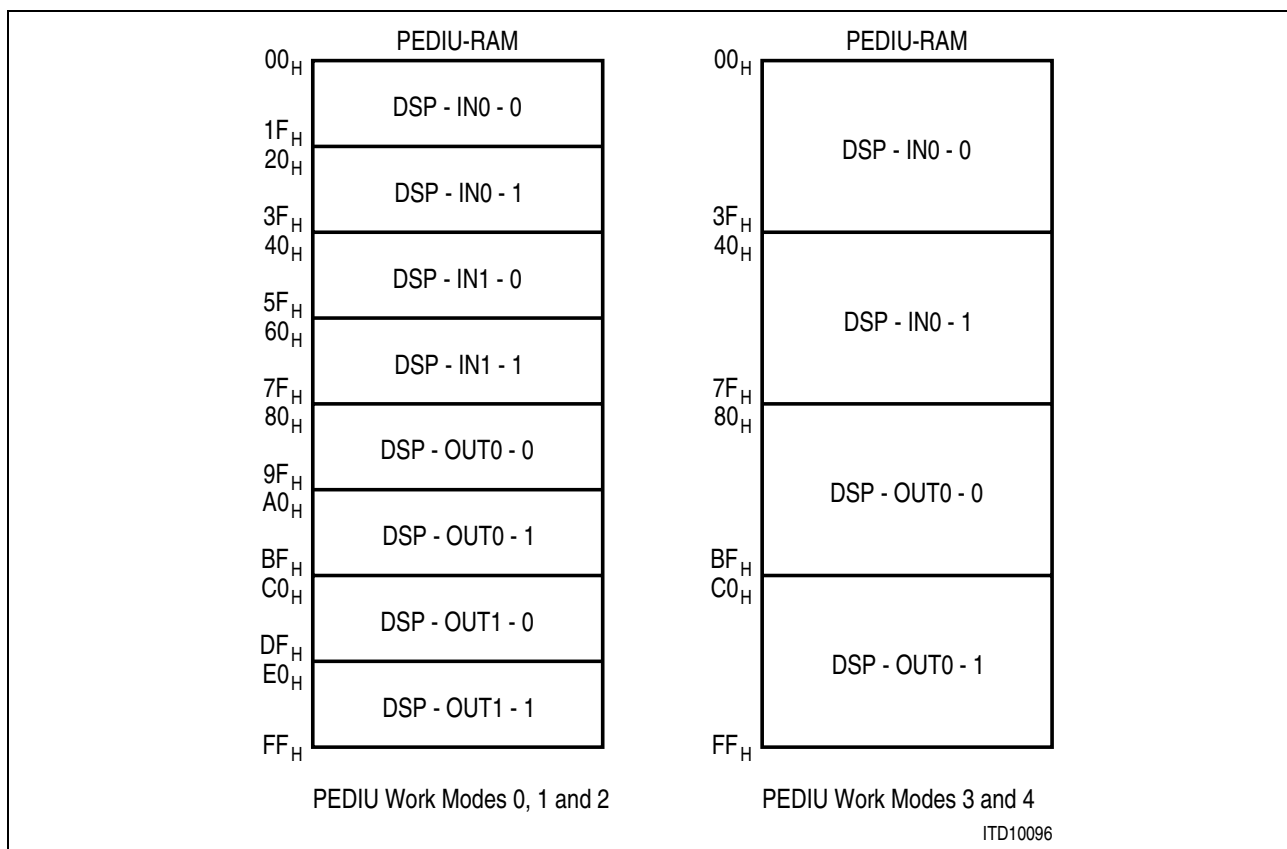
*Note:*

- The data rate of each output stream is identical to the one of an input stream, and the number of output streams is identical to the number of input streams in any mode.
- In Mode0 DCL is a double clock, which means that a new bit streams in every 2 DCL cycles. In modes 1, 2 and 3 DCL is a single clock.
- In all modes FSC = 8 kHz.
- In modes 0, 1, 2, 3 DSP clock rate can be 20 MHz, 30 MHz or 40 MHz. In mode 4 DSP clock rate should be 40 MHz
- DCL rate may be 2.048 MHz, 4.096 MHz or 8.192 MHz.
- In Modes 2 and 3 there are 64 time-slots in each frame. In mode 2 the circular buffer is divided into 4 IN blocks and 4 OUT blocks of 32 words each (in the same way as in mode0), so only the first 32 time-slots, coming in each IN-stream, can be handled by the PEDIU. Unlike mode 2, in mode 3 the circular buffer divided into 2 IN blocks and 2 OUT blocks of 64 words each, so handling of all 64 time-slots of the frame is possible, but only of the IN0 input-stream. The division of the circular buffer in Mode 4, is as in Mode 3.

The exact address space of each circular buffer block, as a function of the PEDIU work mode, can be seen in **Table 2-22**.

**Table 2-22 Address Spaces of Circular Buffer Blocks as a Function of the PEDIU Work Mode**

Mode	IN Blocks				OUT Blocks			
	IN0 - 0	IN0 - 1	IN1 - 0	IN1 - 1	OUT0 - 0	OUT0 - 1	OUT1 - 0	OUT1 - 1
Modes 0, 1 & 2	0x00 - 0x1F	0x20 - 0x3F	0x40 - 0x5F	0x60 - 0x7F	0x80 - 0x9F	0xA0 - 0xBF	0xC0 - 0xDF	0xE0 - 0xFF
Mode 3 & 4	0x00 - 0x3F	–	0x40 - 0x7F	–	0x80 - 0xBF	–	0xC0 - 0xFF	–



**Figure 2-42 Block Structure of Circular Buffer (PEDIU RAM) in Different PEDIU Work Modes**

### AMUL - bit 3 in UCR

AMUL bit defines whether PEDIU applies a-low to linear conversion on the input stream and linear to a-low conversion on the output stream, or  $\mu$ -low to linear conversion on the input stream and linear to  $\mu$ -low conversion on the output stream.

AMUL = 0: a-low conversion.

AMUL = 1:  $\mu$ -low conversion.

*Note: a/ $\mu$ -low to linear conversion and linear to a/ $\mu$ -low conversion can be bypassed by programming registers UBPIR and UBPOR. For details see **section 2.8.2.3**.*

### ACTIVEP - bit 4 in UCR

ACTIVEP bit defines whether or not the PEDIU is in idle mode. When in idle mode the PEDIU is paralyzed, and no activities may take place inside it –including DMA accesses to the PEDIU RAM.

ACTIVEP = 0: PEDIU is in idle mode.

ACTIVEP = 1: PEDIU is active.

*Note: When the PEDIU is in idle mode (ACTIVEP = 0), access of the DSP to the PEDIU RAM is enabled.*

### 2.8.2.2 PEDIU Status Register (USR)

The USR is the status register of the PEDIU. Its content may be used by the programmer to find out the status of the PEDIU. This register has only one bit –the most significant bit: CB.

DSP reads USR from the address 0xC101. The 15 lsbs will be read as ‘0’.

bit	15																0
USR	CB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

It is not possible to write to USR.

After reset the USR register is “0000<sub>H</sub>”.

**CB** Current Block

The CB bit indicates the state of the circular buffer during the current frame. CB indicates the blocks of the circular buffer the accessed by the PEDIU in this frame and those that should be accessed by the DSP. The exact meaning of CB depends on the work mode of the PEDIU, as this mode defines the circular buffer structure, as can be seen in **Table 2-22**.

CB = ‘0’: When PEDIU mode is 0, 1 or 2:

During the current frame the PEDIU reads from circular buffer blocks DSP-OUT0 - 0 and DSP-OUT1 - 0 and writes into DSP-in0 - 0 and DSP-in1 - 0. When PEDIU mode is 3 or 4:

During the current frame the PEDIU reads from circular buffer blocks DSP-OUT0 - 0 and writes into DSP-in0 - 0. For more details see **Table 2-22**, which explains the circular buffer block.

CB = ‘1’: When PEDIU mode is 0, 1 or 2:

During the current frame the PEDIU reads from circular buffer blocks DSP-OUT0 - 1 and DSP-OUT1 - 1 and writes into DSP-in0 - 1 and DSP-in1 - 1. When PEDIU mode is 3 or 4:

During the current frame the PEDIU reads from circular buffer blocks DSP-OUT0 - 1 and writes into DSP-in0 - 1. For more details see **Table 2-22**, which explains the circular buffer block structure in each mode.

CB is inverted at the start of every new frame, immediately after rising edge sampling of FSC by the PEDIU. After reset or when the PEDIU is in idle mode (UCR:ACTIVEP = 0), CB is updated to ‘0’. In the first frame after PEDIUs activation (set UCR:ACTIVEP to ‘1’), CB will be ‘1’.

**Figure 2-43** describes the function of CB when PEDIU work mode is 0, 1 or 2 and **Figure 2-44** describes the function of CB when PEDIU work mode is 3 or 4.

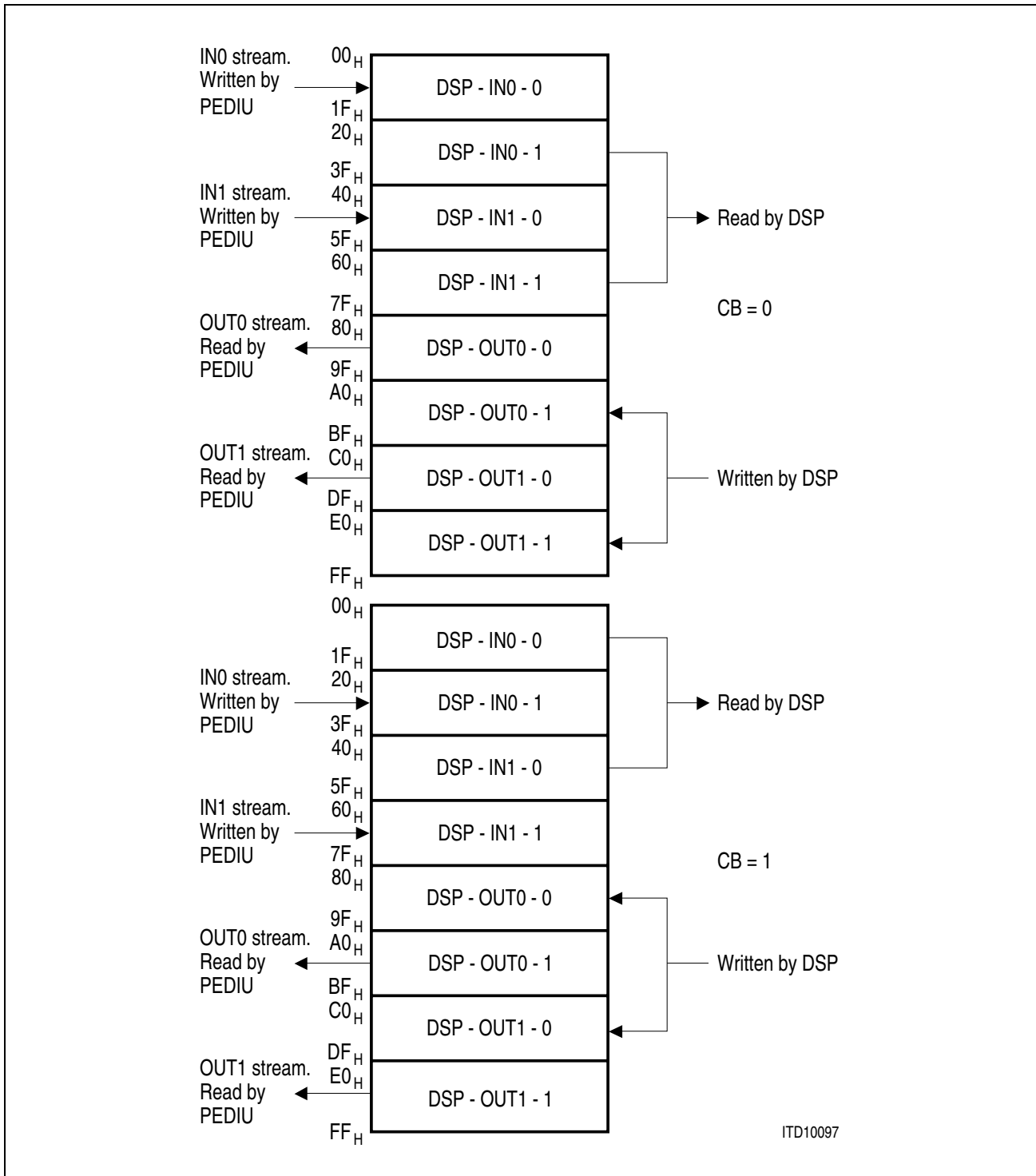
---

**Functional Block Description**

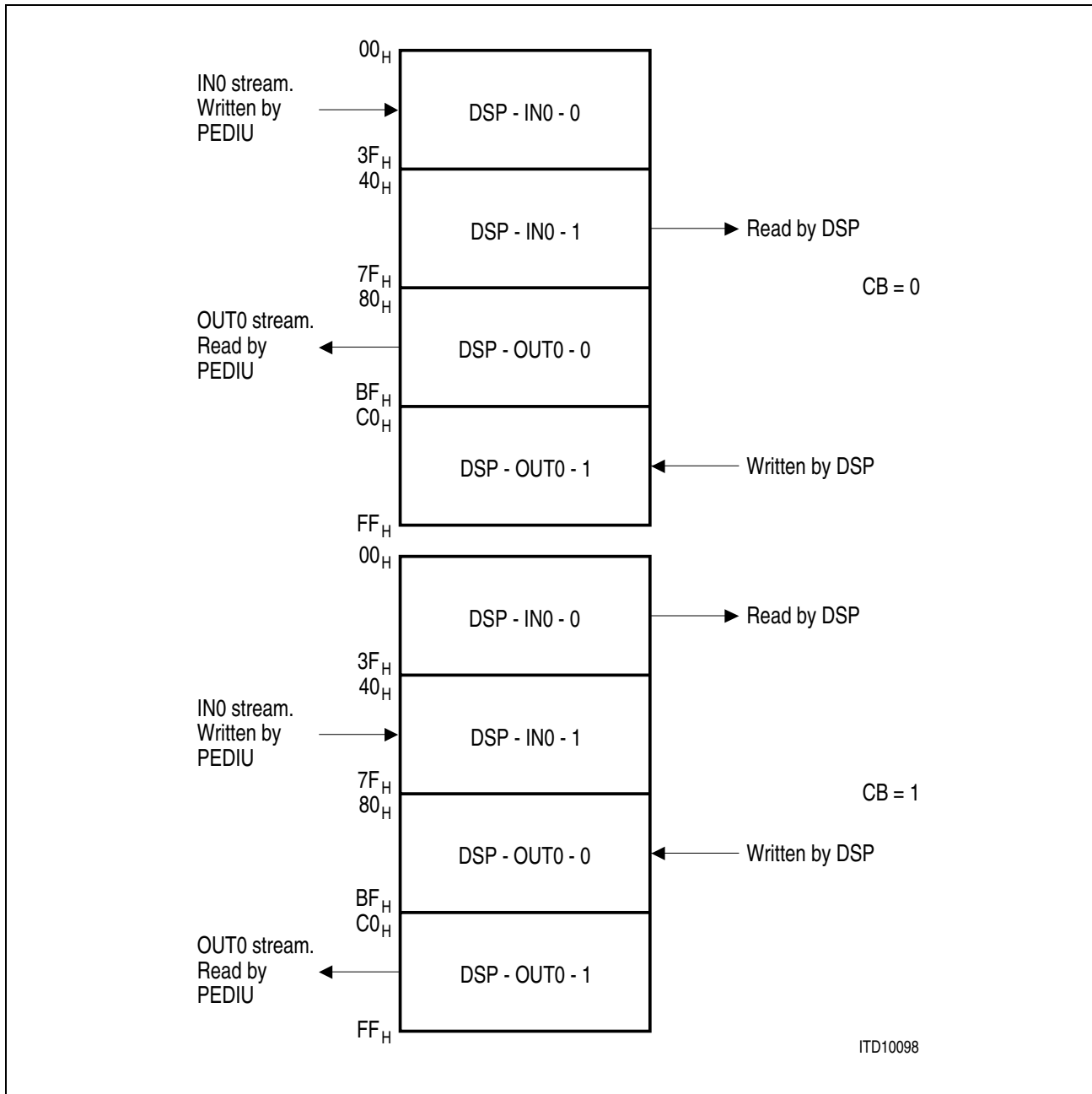
*Note: It is **unnecessary** for the user to read the CB bit during regular work with the PEDIU, in purpose to access the right block in the circular buffer. During regular work, (in modes 0, 1, 2, 3 or 4, when the PEDIU is active), the PEDIU uses the CB signal internally for both DMA accesses to the circular buffer and DSP accesses to the circular buffer. In the case of DSP access to the circular buffer, the PEDIU disregards one of the address bits supplied by the DSP, and use  $\overline{CB}$  instead. In this way any DSP access to the circular buffer accesses a permissible block, automatically. For more details refer to **section 2.8.5.3**. Actually, the PEDIU Status Register is used only for PEDIU testing.*



Functional Block Description



**Figure 2-43 Connection between CB bit and Accesses to the Circular Buffer Blocks in PEDIU work Mode 0, 1 or 2**



**Figure 2-44 The Connection between CB bit and Accesses to the Circular Buffer Blocks in PEDIU work Mode 3 or 4**

### 2.8.2.3 PEDIU Input Stream Bypass Enable Register (UISBPER)

This register defines:

- Which time-slots, coming in the input streams, will be converted to linear values by the a-/μ-law to linear converter
- Which time-slots will bypass this converter.

The UISBPER is a 16-bit register, where each bit controls 4 sequential time-slots and determines which time-slots will be converted or not. After setting any of UISBPER bits to '1', its related 4 sequential time-slots will bypass the a-/μ-law to linear converter. After resetting any of UISBPER bits (to '0'), its related 4 sequential time-slots will be converted by the a-/μ-law to linear converter.

15	14	13	12	11	10	9	8
UISBPE15	UISBPE14	UISBPE13	UISBPE12	UISBPE11	UISBPE10	UISBPE9	UISBPE8
7	6	5	4	3	2	1	0
UISBPE7	UISBPE6	UISBPE5	UISBPE4	UISBPE3	UISBPE2	UISBPE1	UISBPE0

For each bit in UISBPER

- '0' **Convert** the related sequential time-slot quadruplet from a-/μ-law to linear word.
- '1' Let the related sequential time-slot quadruplet to **bypass** the a-/μ-law to linear converter.

The PEDIU work mode determines the quadruplet (4 sequential time-slots) that each UISBPER bit is in charge of. When the PEDIU works in mode 0 or 1 and 2 the UISBPER is divided into two parts. The 8 lsbs controls the 32 time-slots per frame (when working in mode 2 these are the first 32 time-slots from 64), coming in IN0 input stream. The 8 msbs controls the 32 time-slots per frame, coming in IN1 input stream. When working in mode 3 or 4 UISBPER is not divided, and all its 16 bits controls the 64 time-slots, coming in IN0 input stream. In mode 4 these are the first 64 timeslots from 128.

**Table 2-23** describes which time-slot quadruplet is controlled by each UISBPER bit in each PEDIU work mode.

**Table 2-23 Specification of the Time-Slot Quadruplet Controlled by Each Bit in UISBPER, in the Different Work Modes of the PEDIU**

UISBPER Bit	Mode 0/1/2		Mode 3/4	
	Input Stream	Time-Slots	Input Stream	Time-Slots
ISBPE0	IN0	0-3	IN0	0-3
ISBPE1	IN0	4-7	IN0	4-7
ISBPE2	IN0	8-11	IN0	8-11
ISBPE3	IN0	12-15	IN0	12-15

Functional Block Description

**Table 2-23 Specification of the Time-Slot Quadruplet Controlled by Each Bit in UISBPER, in the Different Work Modes of the PEDIU (cont'd)**

UISBPER Bit	Mode 0/1/2		Mode 3/4	
	Input Stream	Time-Slots	Input Stream	Time-Slots
ISBPE4	IN0	16-19	IN0	16-19
ISBPE5	IN0	20-23	IN0	20-23
ISBPE6	IN0	24-27	IN0	24-27
ISBPE7	IN0	28-31	IN0	28-31
ISBPE8	IN1	0-3	IN0	32-35
ISBPE9	IN1	4-7	IN0	36-39
ISBPE10	IN1	8-11	IN0	40-43
ISBPE11	IN1	12-15	IN0	44-47
ISBPE12	IN1	16-19	IN0	48-51
ISBPE13	IN1	20-23	IN0	52-55
ISBPE14	IN1	24-27	IN0	56-59
ISBPE15	IN1	28-31	IN0	60-63

After reset all bits of UISBPER are '0'.

**Setting Resetting and Reading UISBPER**

UISBPER has a special method of setting and resetting its bits. Both UISBPER reset instructions and UISBPER set instructions are implemented by the DSP write instruction of a word that indicates the bits that should be set when it is a UISBPER set instruction, and which bits should be reset when it is a UISBPER reset instruction. The only difference between the set instruction and the reset instruction is the address that this word is being written to:

- For a set instruction the address of the DSP write instruction should be 0xC102
- For a reset instruction the address of the DSP write instruction should be 0xC103.

Written word values are such that:

- Every '1' in the written word indicates that the compatible bit in UISBPER should be set or reset, depending on the address
- Every '0' in the written word indicates that the compatible bit in UISBPER should not be changed.

*Note: The term "compatible bit" means that bit0 (lsb) of the written word determines if bit ISBPE0 of UISBPER will be set/reset or unchanged; bit1 determines ISBPE1 and so on.*

Reading UISBPER is standard, and accomplished by a DSP read instruction from address 0xC104. The bit sequence in the read word will be the same as in UISBPER.

### 2.8.2.4 PEDIU Output Stream Bypass Enable Register (UOSBPER)

UOSBPER functions exactly like UISBPER, except that it controls the output streams instead of the input streams. It is also a 16 bit register. Each bit in UISBPER determines whether a sequential quadruplet of time-slots, read from the circular RAM into the output streams, will bypass the linear to a-/μ-law converter, or will be converted by it. The related sequential time-slot quadruplet of each UOSBPER bit, and the output stream (OUT0 or OUT1) that this quadruplet belongs to, are different when PEDIU is in work mode 0 or 1 and when PEDIU is in work mode 2, as is the case in UISBPER. For more details see **section 2.8.2.3**.

The setting and resetting method of UOSBPER is exactly like the one of UISBPER. Only the addresses are different. Every ‘1’ in the written word indicates a bit that should be set or reset, depending on the address, and every ‘0’ indicates a bit that should not be changed.

The addresses for setting and resetting of UOSBPER:

0xC106: Set UOSBPER.

0xC107: Reset UOSBPER.

The address for reading UOSBPER is 0xC105.

15	14	13	12	11	10	9	8
UOSBPE15	UOSBPE14	UOSBPE13	UOSBPE12	UOSBPE11	UOSBPE10	UOSBPE9	UOSBPE8
7	6	5	4	3	2	1	0
UOSBPE7	UOSBPE6	UOSBPE5	UOSBPE4	UOSBPE3	UOSBPE2	UOSBPE1	UOSBPE0

For each bit in UOSBPER:

- ‘0’ **Convert** the related sequential time-slot quadruplet from linear to a-/μ-law bite.
- ‘1’ Let the related sequential time-slot quadruple to **bypass** the linear to a-/μ-law converter.

After reset all bits of UOSBPER are ‘0’.

**Table 2-24** describes which time-slot quadruplet controlled by each UOSBPER bit in each PEDIU work mode.

**Table 2-24 Specification of the Time-Slot Quadruplet that Controlled by Each Bit in UOSBPER, in the Different Work Modes of the PEDIU**

UOSBPER Bit	Mode 0/1/2		Mode 3/4	
	Output Stream	Time-Slots	Output Stream	Time-Slots
OSBPE0	OUT0	0-3	OUT0	0-3
OSBPE1	OUT0	4-7	OUT0	4-7
OSBPE2	OUT0	8-11	OUT0	8-11
OSBPE3	OUT0	12-15	OUT0	12-15
OSBPE4	OUT0	16-19	OUT0	16-19
OSBPE5	OUT0	20-23	OUT0	20-23
OSBPE6	OUT0	24-27	OUT0	24-27
OSBPE7	OUT0	28-31	OUT0	28-31
OSBPE8	OUT1	0-3	OUT0	32-35
OSBPE9	OUT1	4-7	OUT0	36-39
OSBPE10	OUT1	8-11	OUT0	40-43
OSBPE11	OUT1	12-15	OUT0	44-47
OSBPE12	OUT1	16-19	OUT0	48-51
OSBPE13	OUT1	20-23	OUT0	52-55
OSBPE14	OUT1	24-27	OUT0	56-59
OSBPE15	OUT1	28-31	OUT0	60-63

The bit sequence in the read word will be the same as in UOSBPER.

### 2.8.2.5 PEDIU Tri-State Register (UTSR)

This 16-bit register determines the selection of output stream time-slots of every frame in which the PEDIU will or will not drive the DU0 (Data Upstream input) of ELIC0 and DU1 of ELIC1. The time-slots in which the PEDIU does not drive DU-lines give external IOM-2-devices the opportunity to drive these lines. Each bit in UTSR controls whether the PEDIU will drive the DU-lines during quadruplet of 4 sequential time-slots. During its related time-slot quadruplet, every such bit is driven into the IOM-2-MUX, where it controls which unit the DU-lines will be driven by.

The related sequential time-slot quadruplet of each UTSR bit and the output stream that this quadruplet belongs to (OUT0 or OUT1) are different when PEDIU is in work mode 0 or 1 than when PEDIU is in work mode 2 –as in UISBPER and UOSBPER. For more details see **section 2.8.2.3**.

**Functional Block Description**

**Table 2-25** describes which time-slot quadruplet is controlled by each UOSBPER bit in each PEDIU work mode.

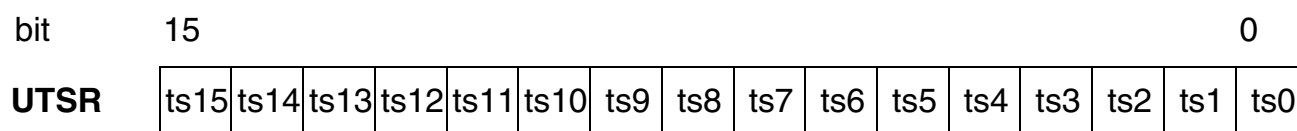
The setting and resetting methods of UTSR are exactly like those of UISBPER and UOSBER; only the addresses are different. Every '1' in the written word indicates a bit that should be set or reset, depending on the address, and every '0' indicates a bit that should not be changed.

The addresses for setting and resetting of UTSR:

0xC108: Set UTSR.

0xC109: Resetet UTSR.

The address for reading UTSR is 0xC10A.



For each bit in UTSR:

**'0'** The PEDIU does not drive the related DU-line (ELIC0-DU0 or ELIC1-DU1) during the related sequential quadruplet of time-slots.

**'1'** The PEDIU drives the related DU-line (ELIC0-DU0 or ELIC1-DU1) during the related sequential quadruplet of time-slots. The bit sequence in the read word will be the same as in UTSR.

After reset all bits of UTSR are '0'.

**Table 2-25 Specification of the Time-Slot Quadruplet Controlled by Each Bit in UOSBPER, in the Different Work Modes of the PEDIU**

UTSR Bit	Mode 0/1/2		Mode 3/4	
	Output Stream	Time-Slots	Output Stream	Time-Slots
TS0	OUT0	0-3	OUT0	0-3
TS1	OUT0	4-7	OUT0	4-7
TS2	OUT0	8-11	OUT0	8-11
TS3	OUT0	12-15	OUT0	12-15
TS4	OUT0	16-19	OUT0	16-19
TS5	OUT0	20-23	OUT0	20-23
TS6	OUT0	24-27	OUT0	24-27
TS7	OUT0	28-31	OUT0	28-31
TS8	OUT1	0-3	OUT0	32-35
TS9	OUT1	4-7	OUT0	36-39

**Table 2-25 Specification of the Time-Slot Quadruplet Controlled by Each Bit in UOSBPER, in the Different Work Modes of the PEDIU (cont'd)**

UTSR Bit	Mode 0/1/2		Mode 3/4	
	Output Stream	Time-Slots	Output Stream	Time-Slots
TS10	OUT1	8-11	OUT0	40-43
TS11	OUT1	12-15	OUT0	44-47
TS12	OUT1	16-19	OUT0	48-51
TS13	OUT1	20-23	OUT0	52-55
TS14	OUT1	24-27	OUT0	56-59
TS15	OUT1	28-31	OUT0	60-63

**2.8.2.6 PEDIU ROM Test Address Register (UPRTAR) and PEDIU ROM Test Data Register (UPRTDR)**

The function of these two registers is to enable efficient testing of the PEDIU ROM. This is a 512 word ROM (9 bit address), and it holds linear values of all 8 bits a-law values and 8-bits  $\mu$ -low values.

Accessing UPRTAR and UPRTDR is possible only when the PEDIU is in work mode 5 - testing mode. In this mode the input streams into the PEDIU are not handled by it, so accessing the PEDIU ROM via these test registers is enabled.

Reading of a PEDIU ROM content is accomplished by writing the 8 lsbs of the demanded address into UPRTAR, and then reading the 16 bit ROM linear content from UPRTDR. The msb of the ROM address is the AMUL bit of register UCR (see **section 2.8.2.1**). This bit also determines which part of the ROM is being tested - the a-law segment ('0') or the  $\mu$ -low segment ('1').

Address off UPRTAR for read and write: 0xC10B.

bit 15 0

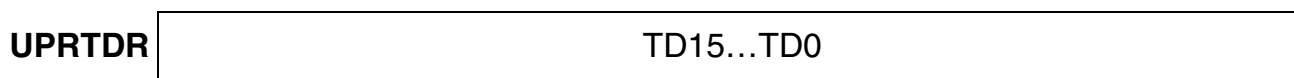


\* - Written as "don't care" and read as '0'.

**TA7...0** Test Address 8 least significant bits. The most significant bit is determined by the bit UCR:AMUL (see **section 2.8.2.1**).

Address off UPRTDR for read only: 0xC10C.

bit 15 0





## Functional Block Description

**TD15...0** Test Data. The PEDIU ROM content, which its address consists of UPRTAR:TA7...0, as the lsb, and from UCR:AMUL, as the msb.

*Note: After writing an address to UPRTAR, at least 1 cycle should pass before trying to read the content of this address from UPRTDR. A nop can be placed between the write and read instructions, in order to sustain it.*

### 2.8.3 PEDIU Synchronization and Clock Rates

#### 2.8.3.1 PEDIU Synchronization by FSC and DCL

The sampling of ELIC0-DD0, ELIC1-DD1 and driving ELIC0-DU0, ELIC1-DU1 by the PEDIU must be synchronized to DCL and FSC. These two signals can be inputs to the DOC, or can be driven by ELIC0 or ELIC1, or by the DOC's internal clocks generator.

The PEDIU is designed to sample ELIC0-DD0 and ELIC1-DD1 in falling edges of DCL:

- In work mode 0 (IOM-2), DCL is a double data-rate clock, and the PEDIU samples the DD signals every second DCL falling edge.
- In work modes 1, 2, 3 and 4 the sampling occurs every DCL falling edge.
- In order to make the DD-signals sampling work correctly under these conditions, the ELICs CFI ports must transmit at DCL rising edge.
- The transmission of the next bit on ELIC0-DU0 and/or on ELIC1-DU1 by the PEDIU is done every rising edge of DCL:
- In work mode 0 the transmission occurs every second DCL rising edge
- In order to make the ELICs sample the DU-lines correctly, the ELICs CFI ports must be programmed to sample the DU signals at DCL falling edge.

The PEDIU synchronizes its sampling of DD-lines and transmission on DU-lines by FSC and DCL. This is done according to IOM-2 specifications, similarly to the way in which the QUAT-S does it (see spec of the QUAT-S, PEB 2084 Version 1.2, Data Sheet 07.95, figures 33 and 34 at pages 64-65):

- When working in PEDIU work mode 0 (double data rate DCL), PEDIU samples the first bit of a frame at the first falling edge of DCL after DCL falling edge, in which active FSC was sampled, and the next samples occur every second DCL falling edge.
- When working in PEDIU work modes 1-4 (single data rate DCL), PEDIU samples the first bit of a frame at the same DCL falling edge, in which active FSC was sampled,

Every rising edge sampling of FSC by the PEDIU starts a new PEDIU frame, and resets its bit-counter and its time-slot counter. Reset of these counters will occur, even if the FSC is not synchronized to the end of the former frame. In cases where the FSC rising edge sampling is too soon and occurs before the end of the frame, the counters will be reset, and the PEDIU starts a new frame. In such cases the output streams, which drive the up streams into the ELIC, will not be defined during the first time-slot of the new frame. In cases where the FSC rising edge sampling is too late, and comes after the end of the last frame, the PEDIU will go into idle state, from the last frame end until the FSC rising edge sampling.

### 2.8.3.2 Restrictions on PEDIU Clock Rates

The PEDIU should synchronize FSC DCL and input data streams (down-streams) to the DSP 2-phase system clock. The DSP clock rate can be 20 MHz, 30 MHz, or 40 MHz:

- When working with DCL rate of 4.096 MHz, the PEDIU will work correctly with any of these DSP rates.
- When working with DCL rate of 8.192 MHz, the DSP rate is restricted to 40 MHz.

### 2.8.4 PEDIU Address Space

The PEDIU address space includes 2 address sub-spaces:

- The PEDIU-RAM (circular-buffer) address space
- The PEDIU register address space.

The PEDIU distinguishes between accesses to these two sub-spaces by 4 signals, which come from the DCL:

- Read and write signals to the RAM space
- Read and write signals to the register space

Accesses of DSP to the register space will be done immediately without any wait states. During DSP accesses to the RAM address space, some wait states might be necessary, when the PEDIU DMA accesses the RAM at the same time.

The PEDIU-RAM address-space in the DSP address space is between the addresses: 0xF000 - 0xF3FF. The value in the 8 lsbs define the shift in the PEDIU RAM.

The PEDIU register space in the DSP address space is between the addresses: 0xC100 - 0xC10F. **Table 2-26** specifies the use of each address in this space.

**Table 2-26 PEDIU Registers Addresses in the DSP Address Space**

Address	Use
C100 <sub>H</sub>	read and write UCR
C101 <sub>H</sub>	read and write USR
C102 <sub>H</sub>	set UISBPER
C103 <sub>H</sub>	reset UISBPER
C104 <sub>H</sub>	read UISBPER
C105 <sub>H</sub>	read UOSBPER
C106 <sub>H</sub>	set UOSBPER
C107 <sub>H</sub>	reset UOSBPER
C108 <sub>H</sub>	set UTSR
C109 <sub>H</sub>	reset UTSR
C10A <sub>H</sub>	read UTSR

**Table 2-26 PEDIU Registers Addresses in the DSP Address Space (cont'd)**

Address	Use
C10B <sub>H</sub>	read and write UPRTAR
C10C <sub>H</sub>	read UPRTDR
C10D <sub>H</sub>	reserved
C10E <sub>H</sub>	reserved
C10F <sub>H</sub>	reserved

### 2.8.5 PEDIU Data Processing

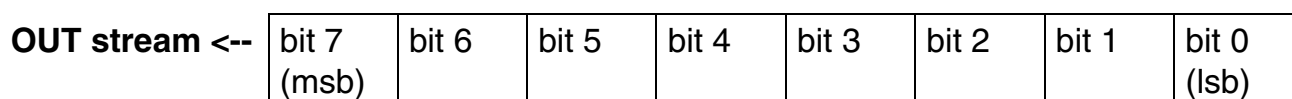
This section describes the way in which the PEDIU should receive and transmit data.

#### 2.8.5.1 PEDIU Serial Data Processing

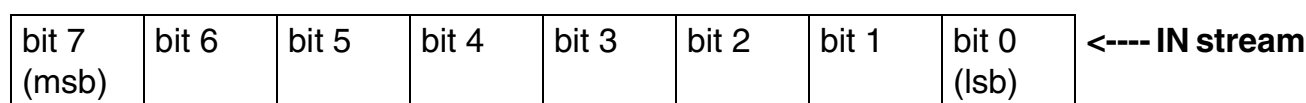
The mode field of UCR defines the following:

- The length of each frame in time-slots is defined by UCR:M.
- The internal structure of each time-slot can not be changed. In all modes each time-slot is constructed from sequential 8 bits, when their sequence is from msb to lsb. The first bit of each transmitted or received time-slot is always bit 7, and the last is bit 0.

Transmitted time-slot form the PEDIU into the OUT streams:



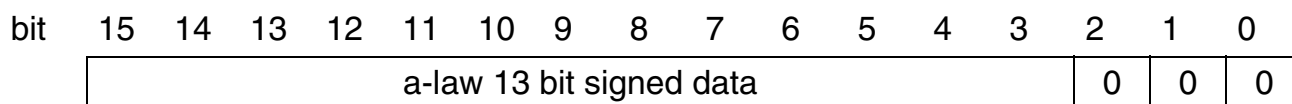
Received time-slot by the PEDIU from the IN streams:



#### 2.8.5.2 PEDIU Parallel Data Processing

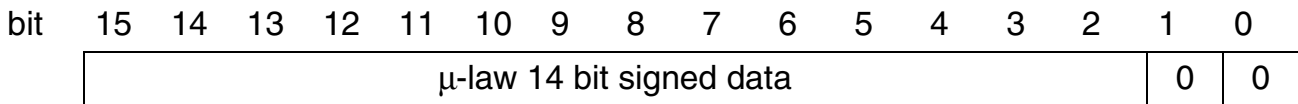
The PEDIU converts the received serial data into parallel data, and writes it into the PEDIU RAM in the following manner:

- If the written data is a linear word which was generated by a-law decoding, this word will be stored in the 13 MSBs. The 3 lsbs will be '0'.

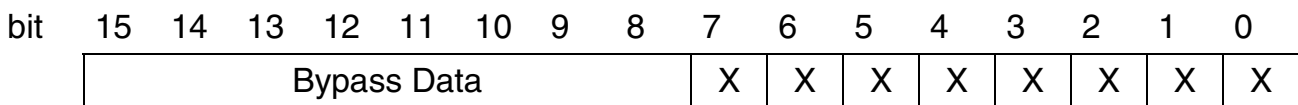


**Functional Block Description**

- If the written data is a linear word which was generated by  $\mu$ -law decoding, it will be stored in the 14 MSBs. The 2 lsbs will be '0'



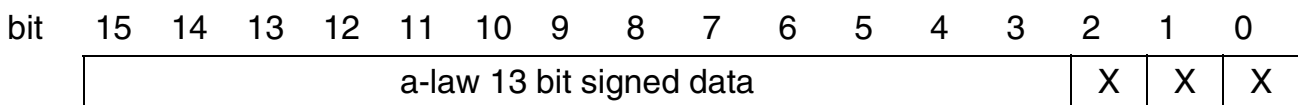
- If the data bypassed the a-/ $\mu$ -law to linear converter, it will be stored in the most significant byte (8 MSBs). The 8 lsbs will be '0'.



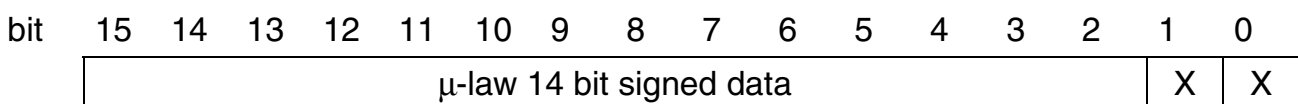
The same rules are valid also when the PEDIU handles words, which were written to the PEDIU RAM by the DSP.

Data should be stored in the following manner:

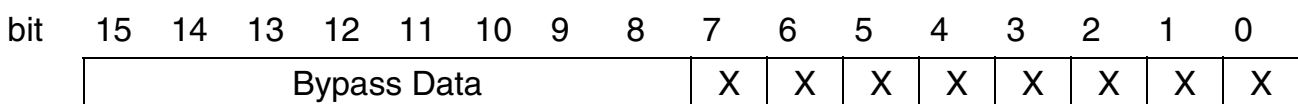
- Linear data which should be encoded according to a-law should be stored in the 13 MSBs, by the DSP. The 3 lsbs should be "don't care".



- Linear data which should be encoded according to  $\mu$ -law should be stored in the 14 MSBs. The 2 lsbs will be '0'



- Data which should bypass the linear to a-/ $\mu$ -law converter should be stored in the most significant byte. The 8 lsbs are "don't care"



**2.8.5.3 The Circular Buffer Address Method**

As can be seen in **Table 2-22** and in **Figure 2-42**, the block structure of the circular buffer is different in each mode. **Section 2.8.2.2** includes a description of which circular buffer blocks the PEDIU should access to, and which blocks should be accessed by the DSP, according to the mode and Current Block (CB) signal.

A special method is used to simplify the DSP access to the circular buffer, during regular work of the PEDIU (when the PEDIU is active and in mode 0, 1, 2, 3 or 4). According to

**Functional Block Description**

this method, DSP SW access to the circular buffer should not be aware the value of Current Block (CB) signal, or the correct circular buffer blocks to which the DSP should access during the current frame:

- For any DSP access to the circular buffer, the most significant byte of the address should be F0<sub>H</sub>.
- The decoding of the least significant byte of the address is different in each mode, and during write operation or read operation to/from the circular buffer.

**Circular-buffer write/read address in modes 0, 1 and 2**

In modes 0, 1, 2 the circular buffer is divided into 4 IN blocks and 4 OUT blocks—32 words each. This is shown in **Table 2-22** and **Figure 2-42**.

When working in one of these modes, decoding of circular buffer **write** address is described as follows:

bit	15	14	13	12	11	10	9	8
	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'
bit	7	6	5	4	3	2	1	0
	'1'	stream 0/1	'1'	indeX (4)	index (3)	index (2)	index (1)	index (0)

**Index4...0** This field defines the index of the specific word in a 32 words circular-buffer block.

**Stream 0/1** 0 = stream 0  
1 = stream 1  
This field defines if the access is to a block of OUT-stream 0 (DSP-OUT0-0, DSP-OUT0-1) or of OUT-stream 1 (DSP-OUT1-0 or DSP-OUT1-1).

**Bit 7, Bit 5** These should always be '1' during a write operation.

When working in modes 0-2, decoding of circular buffer read address is as follows:

bit	15	14	13	12	11	10	9	8
	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'
bit	7	6	5	4	3	2	1	0
	'0'	stream 0/1	'0'	indeX (4)	index (3)	index (2)	index (1)	index (0)

**Index4...0** This field defines the index of the specific word in a 32 words circular-buffer block.

**Functional Block Description**

**Stream 0/1** 0 = stream 0  
 1 = stream 1  
 This field defines if the access is to a block of IN-stream 0 (DSP-IN0-0, DSP-IN0-1) or of IN-stream 1 (DSP-IN1-0, DSP-IN1-1).

**Bits 7:5** These should always be '0' during a read operation.

**Bits 15:8** These should contain the prefix of the circular buffer: F0<sub>H</sub>.

The PEDIU gets only the 8 lsbs of the address (DXAP7...0). In order to access the correct word in the circular buffer, the PEDIU converts the 7 lsbs of the address into an 8 bits address.

For write operation the conversion is as follows:

bit	7	6	5	4	3	2	1	0
<b>DSP</b>	1	stream0/1	CB	index(4)	index(3)	index(2)	index(1)	index(0)

**Bit 7** This is always '1' during a write operation

**Stream0/1** This field is shifted to bit 6

**CB** Current Block ( $\overline{CB}$ ) signal negation is inserted as bit 5.  $\overline{CB}$  is used here because the PEDIU uses CB signal for its own accesses to the circular buffer. Therefore, in order to access the other blocks during DSP accesses,  $\overline{CB}$  must be used. For read operation the conversion is as follows:

bit	7	6	5	4	3	2	1	0
<b>DSP</b>	0	stream0/1	CB	index(4)	index(3)	index(2)	index(1)	index(0)

**Bit 7** This is always '0' during a read operation.

**Stream0/1** This field is shifted to bit 6.

**CB** Current Block ( $\overline{CB}$ ) signal negation is inserted as bit 5.  $\overline{CB}$  is used here because the PEDIU uses CB signal for its own accesses to the circular buffer. Therefore, in order to access the other blocks during DSP accesses,  $\overline{CB}$  must be used.

**Circular-buffer write/read address in modes 3, 4**

In modes 3, 4 the circular buffer is divided into 2 IN blocks and 2 OUT blocks – 64 words each. This is depicted in **Table 2-22** and **Figure 2-42**.

**Functional Block Description**

When working in one of these modes the decoding of circular buffer **write** address is as follows:

bit	15	14	13	12	11	10	9	8
	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'
bit	7	6	5	4	3	2	1	0
	'1'	'1'	indeX (5)	indeX (4)	index (3)	index (2)	index (1)	index (0)

**Index5...0** The stream0/1 field is not required here, because in modes 3 and 4 only stream 0 (IN0 and OUT0) is active. Instead, the index field is 6 bits wide, because each circular-buffer block is of 64 words in modes 3 and 4.

**Bits 7:6** These are always '1' during a write operation.

When working in one of these modes the decoding of circular buffer **read** address is as follows:

bit	15	14	13	12	11	10	9	8
	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'
bit	7	6	5	4	3	2	1	0
	'0'	'0'	indeX (5)	indeX (4)	index (3)	index (2)	index (1)	index (0)

**Bits 7:6** These are always '0' during a write operation.

The PEDIU gets only the 8 lsbs of the address (DXAP7...0). In order to access the correct word in the circular buffer, the PEDIU converts the 7 lsbs of the address into an 8 bits address.

Circular-buffer converted write address in modes 3, 4:

bit	7	6	5	4	3	2	1	0
<b>DSP</b>	1	CB	index(5)	index(4)	index(3)	index(2)	index(1)	index(0)

Bit 7 is '1'.

Circular-buffer converted read address in modes 3, 4:

bit	7	6	5	4	3	2	1	0
<b>DSP</b>	0	CB	index(5)	index(4)	index(3)	index(2)	index(1)	index(0)

Circular-buffer converted read address in modes 3, 4:

Bit 7 is '0'.

### Circular-buffer write/read address in mode 5

Mode 5 is a test mode for testing the PEDIU ROM and the PEDIU RAM (circular buffer). In this mode the DSP can access any address of the circular buffer without any limitations for read or write operation. The internal circular-buffer address, in this mode, is the 8 lsbs of DXAP (DXAP7...0) without any conversion, and it can be any address from 0 to FF<sub>H</sub>. here, also, bits 15...8 of the DSP address should contain the prefix of the circular buffer: F0<sub>H</sub>.

### 2.8.6 a-/μ-law Conversion

In voice systems, two companding/expanding laws are used:

- μ-law following the BELL specification is used in USA, Canada, Japan and Philippines,
- a-law following the CCITT specification is used in Europe and in other countries.

The DOC supports both conversion techniques by an integrated hardware logic. This hardware logic is a part of the PEDIU, and enables conversion from linear to a-/μ-law and from a-/μ-law to linear. This section specifies the a-/μ-law coding/decoding, as they are implemented in the DOC, within the PEDIU.

In both, a-law and μ-law, the 8 bit digital code (the logarithmic data) has a sign-bit, P, three bits of segment, S2S1S0, and 4 bits, Q3Q2Q1Q0, for step selection within the chosen segment.

bit	7	6	5	4	3	2	1	0
	P	S2	S1	S1	Q3	Q2	Q1	Q0

In μ-law, the 8 bit digital code is encoded/decoded from/to a 14 bit linear data, when the msb is the sign-bit. Since the OAK (DSP) data word is 16 bit long, these 14 bit are inserted in the 14 msbs:.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	μ-law 14 bit signed data														0	0

In a-law, the 8 bit digital code is encoded/decoded from/to a 13 bit linear data, when the msb is the sign-bit. Since the OAK (DSP) data word is 16 bit long, these 13 bit are inserted in the 13 msbs:.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	a-law 13 bit signed data													0	0	0



## 2.9 On-chip Emulation (OCEM)

The On-chip Emulation (Debugger) allows “bugs” in the application software to be found and corrected early in design cycle. It is implemented partly in hardware and partly in software. It interprets functions such as Go, Abort, Stop, Data Read/Write and also performs instruction and data breakpoints, program flow trace buffering and breakpoint on event at operation cycle speed without additional off-chip hardware. Different breakpoints can be set on:

- Program Address
- Data Address
- Data Value
- Single Step
- Interrupt

Once a condition is met the On-Chip Emulation activates the TRAP mechanism causing the kernel to suspend any action and jump into the service routine. An additional external signal (STOP pin) is provided to stop in parallel any connected processing element.

Program flow tracing includes dynamic recording of program addresses. Those addresses provide a full program flow graph of instruction being executed.

## 2.10 Mailbox

The  $\mu$ P and the DSP communicate via a bidirectional mailbox according to a user definable protocol. This mailbox includes two separate parts:

- $\mu$ P mailbox - enables transfer from the  $\mu$ P to the OAK.
- OAK mailbox - enable transfer from the OAK to the  $\mu$ P.

Both parts includes a command register,  $6 \times 16$  bits registers and a busy bit. The commands syntax will be defined by the user. An example for some commands can be found in the boot sequence definition within the DCU description.

### 2.10.1 $\mu$ P Mailbox

The  $\mu$ P mailbox includes six general purpose sixteen bit registers (MDTx), an 8-bit command register (MCMD) and one 8-bit busy register (MBUSY). The registers MDTx and MCMD can be written by the  $\mu$ P and read by the OAK. MBUSY can be read by  $\mu$ P.

A write of the  $\mu$ P to the  $\mu$ P mailbox command register generates an interrupt to the OAK (INT2). Therefore, when the user wants to transfer more data the command register must be written last.

A busy bit which can be read by the  $\mu$ P (MBUSY) is set automatically after a write to the  $\mu$ P command register and reset by a direct OAK write operation to it.

Users can define own opcodes (up to 256 opcodes) for the transfer direction from the  $\mu$ P to the OAK.

### Data Transfer from the $\mu$ P to the OAK

- The  $\mu$ P tests the MBUSY bit.
- The  $\mu$ P writes to the data registers (optional).
- The  $\mu$ P writes to the  $\mu$ P-command register (MCMD) (must be performed) - this write sets automatically the  $\mu$ P mailbox busy bit (MBUSY).
- An OAK interrupt (INT2) is activated due to the write to the command register.
- The OAK INT2 routine reads MCMD and performs the command (the read of the command register stops the INT2 activation automatically).
- If the command is asking for data (like a read of an OAK register), the interrupt routine puts the data in the OAK mailbox registers.
- When finished, the INT2 routine resets MBUSY for enabling the  $\mu$ P to send the next command.

The  $\mu$ P can write consecutive writes to the  $\mu$ P mailbox and it's up to the user to make sure that the data has been transferred to the OAK correctly (the busy bit has been reset) before writing new data to the  $\mu$ P mailbox.

#### 2.10.2 OAK Mailbox

The OAK mailbox includes:

- Six general purpose 16-bit registers (ODTx)
- An 8-bit command register (OCMD)
- One bit busy register (OBUSY). All the registers can be written by the OAK and read by the  $\mu$ P.

A write of the OAK to the OAK mailbox command register generates an interrupt to the  $\mu$ P (INT Source No.6). Therefore, when the user wants to transfer more data then the command register must be written last.

A busy bit (OBUSY) which can be read by the OAK is set automatically after a write of the OAK to the OAK command register and is reset by a direct  $\mu$ P write to it (when the  $\mu$ P has finished reading the OAK mail box contents).

Users can define own opcodes (up to 256 opcodes) for the transfer direction from the OAK to the  $\mu$ P.

### Data Transfer from the OAK to the $\mu$ P

- The OAK tests the OBUSY bit.
- The OAK writes to the data registers (optional).
- The OAK writes to the command register (OCMD) (must be performed). This write operation sets the OAK mailbox busy bit (OBUSY).
- A  $\mu$ P interrupt is activated due to the write operation to the command register.
- The  $\mu$ P reads the command register and performs the command.
- If the command is asking for data (like a read of a  $\mu$ P register), the interrupt routine puts the data in the  $\mu$ P mailbox registers.

**Functional Block Description**

- When finished, the  $\mu$ P resets OBUSY bit for enabling the OAK to send the next command. This operation also deactivates the  $\mu$ P-mailbox interrupt.

*Note: 1) The OBUSY bit is set only 4 DSP cycles after a OAK write operation to OCMD register. Therefore, the first polling-read cycle of OBUSYR should take place at least 5 DSP clock cycles after the write cycle to OCMD.*

*2) The OAK can write consecutive writes to the OAK mailbox and it's up to the user to make sure that the data has been transferred to the  $\mu$ P correctly (OBUSY has been reset) before writing new data to the OAK mailbox*

**Table 2-27 Register Contents**

Register	Description	Reset Value	Bit	OAK Access	$\mu$ P Access	$\mu$ P Add. for MSB	$\mu$ P Add. for LSB	OAK Addr.
MCMD	$\mu$ P command	00H	8	R	W	none	340 <sub>H</sub>	C040 <sub>H</sub>
MBUSYR	$\mu$ P MB busy	0H	1	W	R	none	341 <sub>H</sub>	C041 <sub>H</sub>
MDT0	$\mu$ P data reg 0	un-changed	16	R	W	343 <sub>H</sub>	342 <sub>H</sub>	C042 <sub>H</sub>
MDT1	$\mu$ P data reg 1	un-changed	16	R	W	345 <sub>H</sub>	344 <sub>H</sub>	C044 <sub>H</sub>
MDT2	$\mu$ P data reg 1	un-changed	16	R	W	347 <sub>H</sub>	346 <sub>H</sub>	C046 <sub>H</sub>
MDT3	$\mu$ P data reg 1	un-changed	16	R	W	349 <sub>H</sub>	348 <sub>H</sub>	C048 <sub>H</sub>
MDT4	$\mu$ P data reg 1	un-changed	16	R	W	34B <sub>H</sub>	34A <sub>H</sub>	C04A <sub>H</sub>
MDT5	$\mu$ P data reg 1	un-changed	16	R	W	34D <sub>H</sub>	34C <sub>H</sub>	C04C <sub>H</sub>
OCMD	OAK command	00H	8	W	R	none	350 <sub>H</sub>	C050 <sub>H</sub>
OBUSYR	OAK MB busy	0H	1	R	W	none	351 <sub>H</sub>	C051 <sub>H</sub>
ODT0	OAK data reg 0	un-changed	16	W	R	353 <sub>H</sub>	352 <sub>H</sub>	C052 <sub>H</sub>
ODT1	OAK data reg 1	un-changed	16	W	R	355 <sub>H</sub>	354 <sub>H</sub>	C054 <sub>H</sub>
ODT2	OAK data reg 2	un-changed	16	W	R	357 <sub>H</sub>	356 <sub>H</sub>	C056 <sub>H</sub>
ODT3	OAK data reg 3	un-changed	16	W	R	359 <sub>H</sub>	358 <sub>H</sub>	C058 <sub>H</sub>
ODT4	OAK data reg 4	un-changed	16	W	R	35B <sub>H</sub>	35A <sub>H</sub>	C05A <sub>H</sub>
ODT5	OAK data reg 5	un-changed	16	W	R	35D <sub>H</sub>	35C <sub>H</sub>	C05C <sub>H</sub>

*Note: The busy bit within MBUSYR and OBUSYR is always read or written at the MSB.*

**Functional Block Description**

OAK reads of address C051<sub>H</sub> will therefore result with:

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>OBUSYR</b>	OBUSY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*Note: The OBUSY bit is set only 4 DSP cycles after a OAK write operation to OCMD register. Therefore, the first polling-read cycle of OBUSYR should take place at least 5 DSP clock cycles after the write cycle to OCMD.*

μP read of address 341<sub>H</sub> will result with:

bit	7	6	5	4	3	2	1	0
<b>MBUSYR</b>	MBUSY	0	0	0	0	0	0	0

**2.11 μP Interface**

The System Data Interface is a passive interface adaptable to different Microprocessor bus schemes:

- 8-bit data bus multiplexed with lower 8 bits of the address bus
- Upper address bus for an access to all accessible DOC registers.
- Control lines

**2.11.1 Compatibility**

The bus is compatible to the following types of μPs:

- Siemens C16x
- Intel 80x86/88 or

*Note: In 32-bit μP systems (e.g. based on i80386, M68xxx or MIPC R3000), an external logic for control signals generation is necessary (e.g. sequencer PALs).*

**2.11.2 Memory and I/O Organization**

The DOC is a slave to the Microprocessor. The μP can access all DOC registers but the DSP memory, the DSP registers, the DSP memory mapped registers and the PEDIU. The μP can communicate with the DSP via the μP Mailbox only. For more details see register overview, **section 5.1**.

## 2.12 Clock Generator

An integrated clock generator provides all required clocking frequencies.

### 2.12.1 Block Diagram

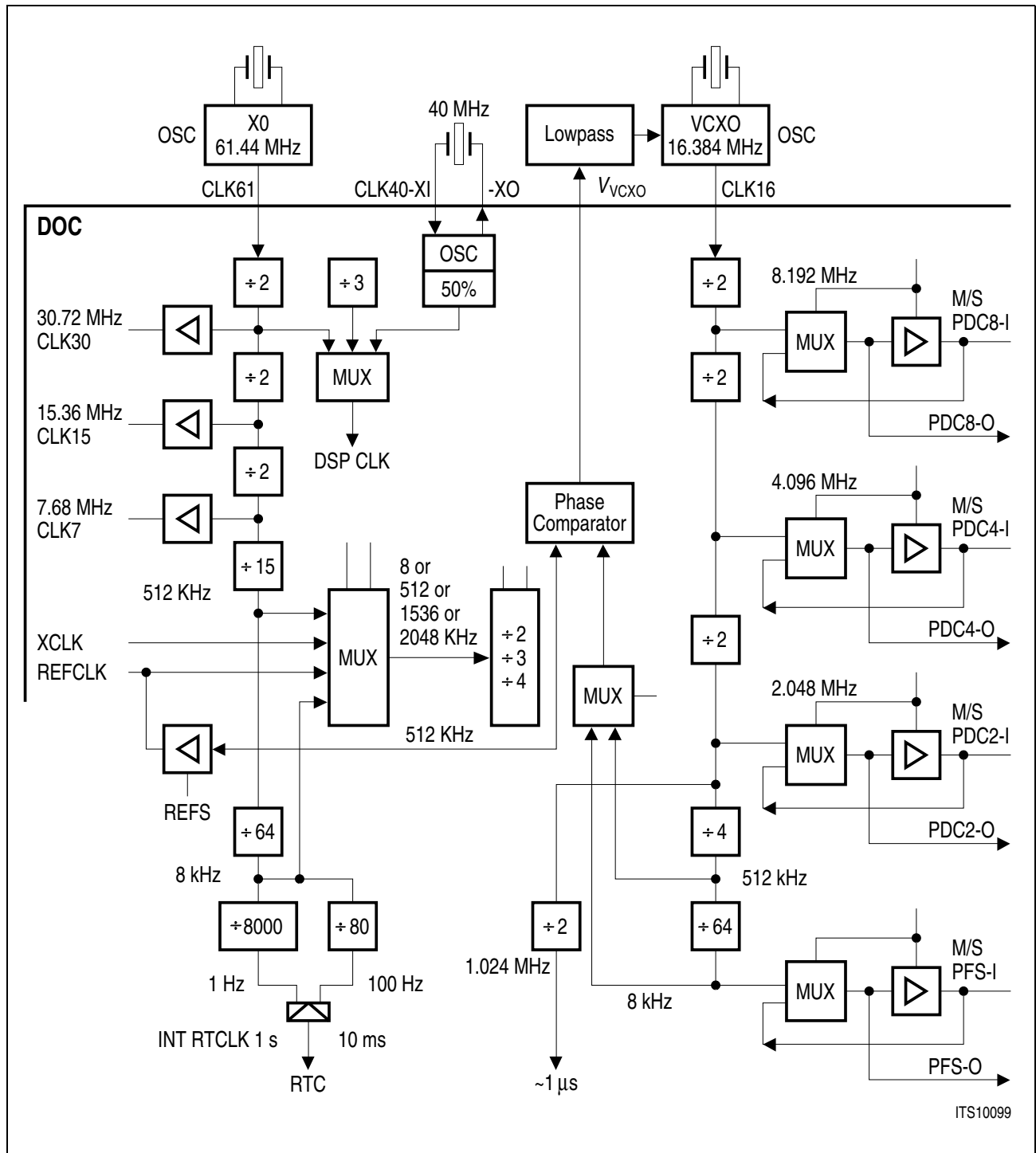


Figure 2-45 DOC Clock Generator

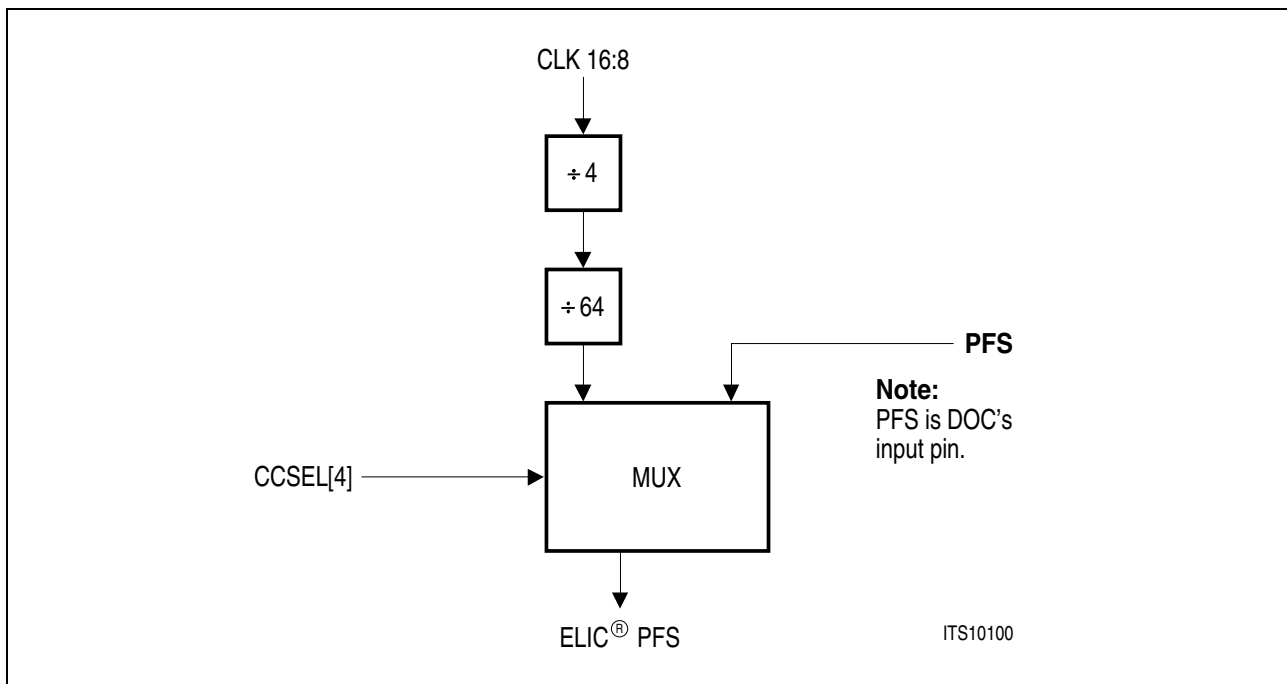
## 2.12.2 Types of Clock Signals

### 2.12.2.1 Input/Output Clocks

- CLK16: 16.384 MHz VCXO clock voltage oscillator input
- CLK61: 61.44 MHz ext. oscillator clock input
- CLK40-XI: up to 40 MHz ext quartz clock input
- CLK40-XO: up to 40 MHz ext quartz clock output
- CLK30: 30.72 MHz (i.e. for  $\mu$ P) output
- CLK15: 15.36 MHz (i.e. for OCTAT-P) output
- CLK7: 7.68 MHz (i.e. for QUAT-S) output
- PFS: Frame synchronization on PCM 8 kHz input/output
- PDC2: PCM data clock 2.048 MHz input/output
- PDC4: PCM data clock 4.096 MHz input/output
- PDC8: PCM data clock 8.192 MHz (i.e. for FALC54) input/output
- FSC: Frame synchronization on IOM-2 interface 8 kHz input/output
- DCL: IOM-2 data clock input/output
- XCLK: 8 kHz or 512 kHz or 1.536 MHz or 2.048 MHz (selectable) input
- REFCLK (i.e. 512 kHz) input/output

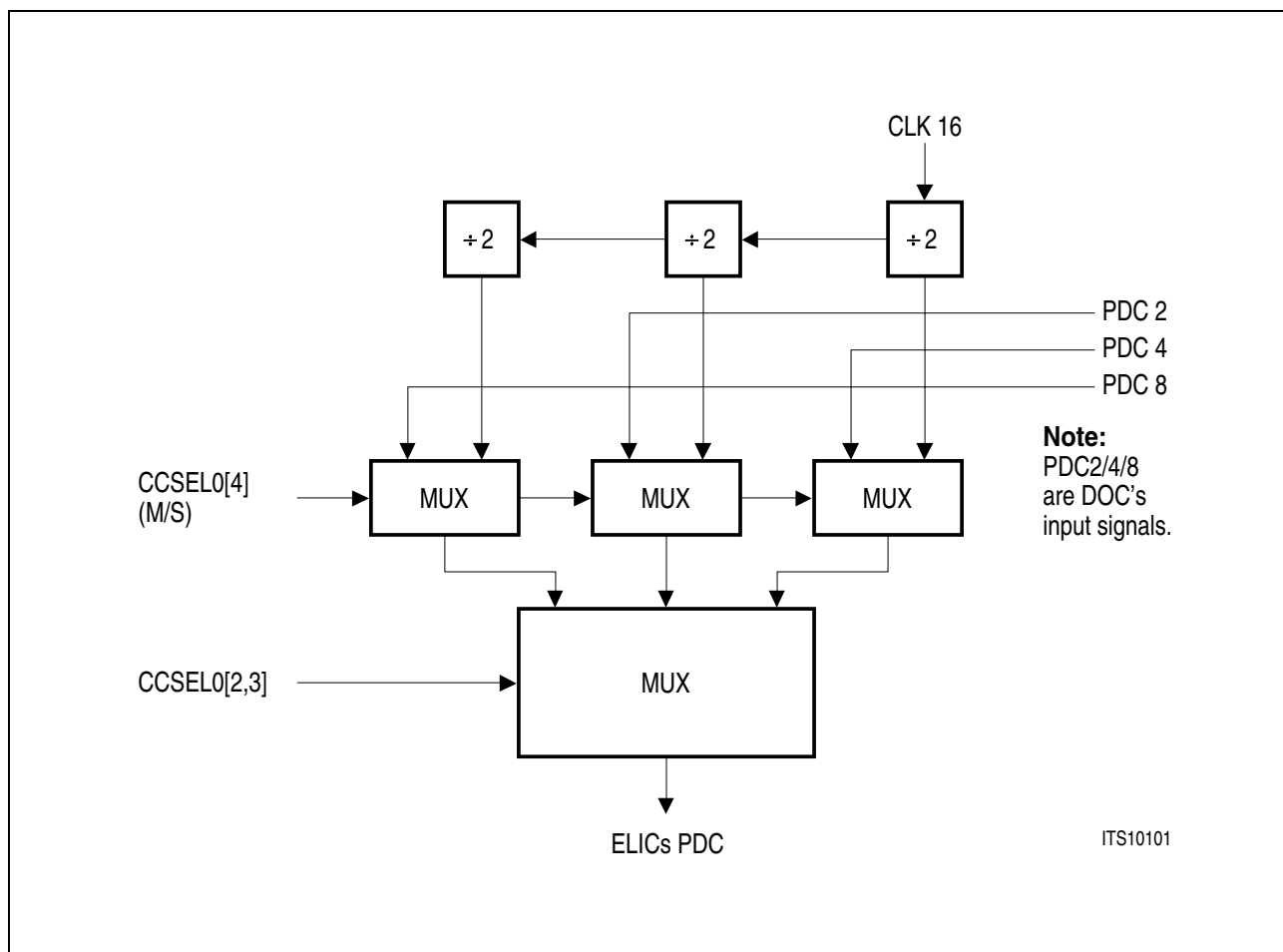
### 2.12.2.2 Clock Selection

- DSP clock: Selectable clock ~ 20, 30 or 40 MHz internal clock
  - The DSP clock can be selected from 3 possible frequencies: 20, 30 or 40 MHz.
  - The source of the clock can be one of two:
    - 61.44 MHz external oscillator, which by dividing by 2,3 produces the 20, 30 MHz clock
    - 40 MHz crystal with an internal oscillator which produces the 40 MHz clock.
  - DSP clock frequency is determined by the input pins, FREQ1...0. The value of these pins can be read from CCSEL0 register, bits1...0 (read only bits).
- RTCLK: Real time clock
  - Clock for Real Time Interrupt: 1 Hz (1 s) or 100 Hz (10 ms) internal clock
  - The RTC interrupt is a periodic interrupt activated every 10 ms/1 s and it is programmable via CCSEL1:RTCP.
- Programmable UART clock internal clock
  - The UART clock is CLK61 divided by 5. It can be further divided by programming the UART. (For more details see **Table 2-34** in **section 2.14.1.2**)
- PFS: Frame synchronization on PCM 8 kHz input/output
  - The PFS signal can be derived from either the CLK16 input clock or be driven by an external signal via the IO pads. The selection between external and internal signals is performed by CCSEL0:MS (master/slave bit). See **Figure 2-46**.



**Figure 2-46 PFS Signal Selection**

- ELIC0 and ELIC1 PCM interface clocks: internal clocks
  - ELIC0 and ELIC1 receive the same PDC and PFS clocks.
  - PDC has 3 optional frequencies: 2.048 MHz, 4.096 MHz and 8.192 MHz, which are produced from the 16.384 MHz input clock or driven by external clocks via IO pads. See **Figure 2-47**.
  - For correct functionality, the PDC frequency must be selected according to the ELICs operation mode.
  - Both ELICs are connected to the same PDC.
  - PDC frequency and configuration are determined by register CCSEL0 bits 4...2. During reset, PDC frequency is driven from an internal 8.192 MHz clock. No PDC will be driven on IO (even though CCLSE0[4] =1) before CCSEL0 is written.
  - The selected PDC functions as the ELIC0 watch-dog timer clock, which also serves as the DOC watch-dog timer (the ELIC1 watch-dog timer exists but unused).



**Figure 2-47 PDC Generation**

- IOM-2 Interface selected FSC clock input/output
  - The FSC signal can be selected from driven by 3 sources:
    - Output: FSC produced by ELIC0 (derived from the selected PFS clock, internally, in ELIC0). FSC drives the output signal. (ELIC0:CMD1:CSS = '0')
    - Note:** *In this case ELIC1 should be programmed to produce its own internal FSC even though it has no effect beyond the ELIC1 block. ELIC0:CMD1:CSS and ELIC1:CMD1:CSS must be programmed with the same value.*
    - Input: FSC driven by IO ((ELIC0:CMD1:CSS = '1') and (CCSEL1[0] = '1'))
    - Output: FSC driven by selected PFS ((ELIC0:CMD1:CSS = '1') and (CCSEL1[0] = '0'))
- IOM-2 Interface selected DCL clock input/output
  - The DCL signal can be selected from 3 sources:
    - Output: DCL produced by ELIC0; derived from the selected PDC clock; drives the output signal. (ELIC0:CMD1:CSS = '0')
    - Note:** *In this case ELIC1 should be programmed to produce its own internal DCL even though it has no effect beyond the ELIC1 block. ELIC0:CMD1:CSS and ELIC1:CMD1:CSS must be programmed with the same value.*



**Functional Block Description**

Input: DCL driven by IO ((ELIC0:CMD1:CSS = '1') and (CCSEL1[2] = '1'))

Output: DCL driven by selected PDC (PDC4, PDC8). ((ELIC0:CMD1:CSS = '1') and (CCSEL1[2] = '0')) PDC4, 8 can be inputs or outputs, depends if the DOC is master or slave.

**Note:** When DCL is input or driven directly from PDC4/8, it should be used as an input by ELIC0 and ELIC1.

- SACCO-B0 input clocks internal clocks
  - There are two SACCO-B0 clocks, HFS and HDC. Each one may be selected from several possible sources as SACCO-B0 can be connected to the IOM signaling mux, to the PCM signaling mux or directly to the IO when in stand-alone mode.
  - HFS:
    - HFS sources:
    - PORT3/HFSB0 input pin, when disconnected from both PCM and IOM signaling muxes.
    - FSC, when SACCO-B0 is connected to the IOM signaling mux.
    - PFS, when SACCO-B0 is connected to the PCM signaling mux.
  - HDC:
    - HDC sources:
    - PDC2, PDC4, PDC8, DCL – when disconnected from both signaling muxes (CCSEL1[5:4]). Each of these signals can be an inputs or an outputs, according to the configuration.
    - PDC when SACCO-B0 is connected to the PCM signaling mux.
    - DCL when SACCO-B0 is connected to the IOM signaling mux.
- SACCO-B1 clocks internal clocks
  - There are two SACCO-B1 clocks, HFS and HDC. Each one may be selected from several possible sources, as SACCO-B1 can be connected to the IOM signaling mux, to the PCM signaling mux or directly to the IO when in stand-alone mode.
  - HFS:
    - HFS sources:
    - DU5/HFSB1 input pin – when disconnected from both signaling muxes.
    - FSC when SACCO-B1 is connected to the IOM signaling mux.
    - PFS when SACCO-B1 is connected to the PCM signaling mux.
  - HDC:
    - HDC sources:
    - PDC2,PDC4,PDC8,DCL0 – when disconnected from both signaling muxes (CCSEL1[7:6]). Each of these signals can be inputs or outputs, according to the configuration.
    - PDC when SACCO-B1 is connected to the PCM signaling mux.
    - DCL when SACCO-B1 is connected to the IOM signaling mux.
- SIDEC clocks internal clocks
  - The SIDEC module includes four HDLC controllers. Each one receives HDC and HFS clocks from the IOM-2 interface.
  - The HDC clocks are driven by DCL.

---

**Functional Block Description**

- The HFS clocks are driven by FSC.
- PEDIU clocks internal clocks
  - Data clock is driven by IOM-2 interface selected DCL
  - Frame clock is driven by IOM-2 interface selected FSC
- Clock for Run Time Statistics: 1.024 MHz (~ 1µs) internal clock
  - The run time statistics pulse is a periodic clock toggling at 1.024 MHz. It is derived from the CLK16 input clock and is used in DCU for DSP statistics.
- SACCO-A0 input clocks: internal clocks
  - There are two SACCO-A0 clocks, HFS and HDC. These clocks can be provided only, internally, by ELIC0. (Only clock mode 3 of SACCO-A0 is available.)
- SACCO-A1 input clocks: internal clocks
  - There are two SACCO-A1 clocks, HFS and HDC. These clocks can be provided only, internally, by ELIC1. (Only clock mode 3 of SACCO-A1 is available.)

### 2.12.3 Clocks Generator Registers Description

#### 2.12.3.1 Clocks Select 0 Register (CCSEL0)

Address: 360<sub>H</sub>

μP interface mode: read/write

Reset Value: 0001\_00xx - where xx depends on input pins.

*Note: bits 1...0 are read only Bits 7...5 are unused, and are read as '0'.*

bit 7				bit 0			
0	0	0	MS	PDCF1	PDCF0	DSPF1	DSPF0

This register fulfils the following functions:

- DSP frequency indication.
- PCM Data Clock (PDC) frequency selection.
- Definition if the DOC functions as PCM clocks master (PDC and PFS generator) or as PCM clocks slave (the DOC gets PDC and PFS, as inputs).

**DSPF1...0** DSP frequency  
 00: 20 MHz  
 01: 30 MHz  
 10: 40 MHz  
 11: 61 MHz (used only for testing)

*Note: These bits are read only, and their value is determined by the input pins: FRQ1...0.*

**PDCF1...0** PDC frequency for ELIC0 and ELIC1  
 00: 8.192 MHz  
 01: 4.096 MHz  
 10: 2.048 MHz  
 11: reserved

**MS** Master/Slave (PDC & PFS internal/external)  
 0: slave - PDC and PFS are external.  
 1: master - PDC and PFS are internal and driven on IO

*Note: After reset, PDC/PFS are internally by the clocks generator, but the PDC/PFS pins stay in tri-state. PDC/PFS are driven by the DOC, only after the first write access to CCSEL0, if the DOC was configured as master.*

### 2.12.3.2 Clocks Select 1 Register (CCSEL1)

Address: 361<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

bit 7

bit 0

SB1DC1	SB1DC0	SB0DC1	SB0DC0	RTCP	DCLS	DCLF	FSCS
--------	--------	--------	--------	------	------	------	------

This register fulfils the following functions:

- IOM-2-interface-DCL source and frequency.
- IOM-2-interface-FSC source.
- SACCO-B0/1 HDC source when does not connected to any of the signaling muxes.
- Real Time Clock Interrupt Period.

**SB1DC1...0** SACCO-B1 HDC Selection

Select SACCO-B1 HDC when it does not connected to PCM/IOM signaling muxes.

- 00: Select PDC8
- 01: Select PDC4
- 10: Select PDC2
- 11: Select DCL0

**SB0DC1...0** SACCO-B0 HDC Selection

Select SACCO-B0 HDC when it does not connected to PCM/IOM signaling muxes.

- 00: Select PDC8
- 01: Select PDC4
- 10: Select PDC2
- 11: Select DCL0

**RTCP** Real Time Clock Interrupt Period

- 0: 10ms period
- 1: 1 s period

**DCLS** IOM-2 Interface Data Clock Select

DCL from IO/PDC (See the note at the end of this section)

- 0: DCL from PDC (DCL pin is used as an output)
- 1: DCL from IO (DCL pin is used as an input)

**DCLF** –IOM2 Interface Data Clock Frequency.

Select DCL from PDC4/PDC8 (valid only when DCLS = 0)

- 0: DCL is PDC8
- 1: DCL is PDC4

---

**Functional Block Description**

**FSCS** –IOM2 Interface Frame Synchronization Clock Select.  
Select FSC from IO/PFS.  
0: FSC from PFS  
1: FSC from IO

- Note: 1) Although the FSC/DSL ports are tri-state after reset, FSC/DCL are driven internally by the input signal CLK16 divided by 2 for DCL, by 2048 for FSC. FSC pin is driven by the DOC, only after write access to CCSEL1, if FSC was configured as an output.*
- 2) Bits 0,2 are valid only when ELIC0:CMD1:CSS = '1' (internal ELIC0 gets FSC and DCL as inputs).*
- 3) After reset ELIC0:CSS=0 (internal ELIC provides DCL, FSC), therefore DCLs and FSCs should be set to '1'.  
Otherwise (if programmed to '0') the internal PDC (8 MHz), PFS (8 kHz) are output on DCL/FSC until the CFI interface is enabled (ELIC0:OMDR:CSB=1). The result would be unpredictable behavior of the connected layer 1 devices.*
- 4) Bits 0 and 2 should be programmed to the same value to ensure the correct functionality of the DOC.*

### 2.12.3.3 Clocks Select 2 Register (CCSEL2)

Address: 362<sub>H</sub>

μP interface mode: read/write

Reset Value: 00<sub>H</sub>

*Note: Bits 7...6 are unused, and are read as '0'.*

bit 7							bit 0
0	0	RCD1	RCD0	VCXOF	RCS1	RCS0	RCDIR

This register fulfils the following functions:

- REFCLK IO control.
- VCO frequency selection.
- Reference clock source selection.
- Reference clock divisor selection.

**RCD1...0** Reference Clock Divider  
 00: No division of reference clock  
 01: Forbidden  
 10: Divide reference clock by 3  
 11: Divide reference clock by 4

**VCXOF** Voltage Control Oscillator Frequency  
 0: Select 512 KHz clock  
 1: Select 8 KHz clock

**RCS1...0** Reference Clock Source  
 00: Select internal 512 KHz as reference clock  
 01: Select REFCLK as reference clock  
 10: Select XCLK as reference clock  
 11: Select internal 8 KHz clock as reference clock

**RCDIR** REFCLK direction  
 0: REFCLK is input  
 1: REFCLK is output

*Note: The first write access to the DOC (after Reset) should not be addressed to CCSEL2 register.*

## 2.13 Interrupt Controller

All DOC interrupt sources send their interrupt request to the  $\mu\text{P}$  through the interrupt control unit. This unit checks if the  $\mu\text{P}$  allows an interrupt (IE).

- If the interrupt is not masked, the controller sends an interrupt request to the  $\mu\text{P}$ . As a result, the  $\mu\text{P}$  sends interrupt acknowledge through the  $\overline{\text{IACK}}$  input and the interrupt controller puts the address of the interrupt source on the  $\mu\text{P}$  data bus.
- Interrupt acknowledge from the  $\mu\text{P}$  ( $\overline{\text{IACK}}$ ). According to the interrupt-acknowledge protocol defined by the selected  $\mu\text{P}$  bus mode this signal has different behavior. In Siemens/Intel mode the vector is driven by the falling edge of the 2nd (last) pulse of  $\overline{\text{IACK}}$ . In Motorola mode the vector is driven by the falling edge of the 1st (and only) pulse.
- If more than one unit sends an interrupt request at the same time, the interrupt control unit sends the highest prior source interrupt request first.

The  $\mu\text{P}$  can also read the Global Interrupt Status Register (IGIS).

The DOC Version 2.1 provides two new 8-bit interrupt status registers (IGIS0 and IGIS1) for applications in which the generated interrupt vector can not be used. The pending interrupt status is displayed by reading the registers.

### 2.13.1 MASK (IMASK0, IMASK1)

Each interrupt source has a bit in a mask register (IMASKR).

- If the bit is 1, the interrupt is disregarded.
- If the bit is 0, the interrupt is handled

#### Interrupt Mask Registers (IMASKR)

There are two 8-bit mask registers in the DOC: IMASKR0 and IMASKR1.

Value in both registers after Reset:  $\text{FF}_{\text{H}}$

#### Interrupt Mask Register 0 (IMASKR0)

Address  $302_{\text{H}}$

Read and Write

Reset Value  $\text{FF}_{\text{H}}$

bit	7							0
<b>IMASKR0</b>	M7	M6	M5	M4	M3	M2	M1	M0

**Interrupt Mask Register 1 (IMASKR1)**

Address 303<sub>H</sub>

Read and Write

Reset Value 07<sub>H</sub>

bit	7					0		
<b>IMASKR1</b>	0	0	0	0	0	M10	M9	M8

**M10 to M0 Mask bits**

‘0’ = not masked interrupt (the interrupt is enabled)

‘1’ = masked interrupt (the interrupt is disabled)

*Note: After reset, all interrupts are disabled.*

**2.13.2 Interrupt Sources**

**Table 2-28 Interrupt Sources**

Source Number	Interrupt Sources	Interrupt Name	Fixed Source Identifier	DOC Address Base
S0	ELIC0	EINT0P	0000	
S1	ELIC1	EINT1P	0001	
S2	SIDEC: SACC0	S0INTP	0010	
S3	SIDEC: SACC1	S1INTP	0011	
S4	SIDEC: SACC2	S2INTP	0100	
S5	SIDEC: SACC3	S3INTP	0101	
S6	OAK-Mail Box	DINTP	0110	
S7	GPIO Port	VINTP	0111	
S8	FSC	UINTP	1000	
S9	UART	AINTP	1001	
S10	RTC (1 s or 10 ms CLK)	CINTP	1010	

**2.13.3 Interrupt Priority (IPAR0, IPAR1, IPAR2)**

All interrupt sources can be assigned by the user into four different priority groups:

- Group no. ‘11’ has the highest priority
- Group no. ‘00’ the lowest priority.

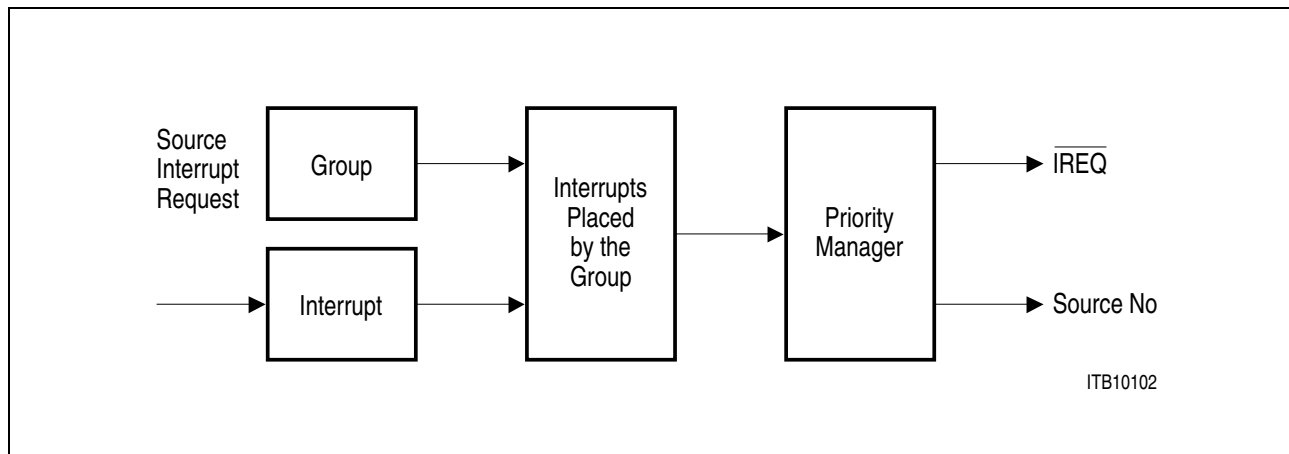
The interrupt priority within the group is defined by the physical source number:

- S0 has the highest priority
- S10 has the lowest priority.



**Functional Block Description**

**For Example:** S2, S3, S9 and S10 are programmed into the same priority group S2 will have highest priority.



**Figure 2-48 Priority Unit - Block Diagram**

**Registers for Priority Assignment (IPAR)**

**Interrupt Priority Assignment Register 0 (IPAR0)** Address 304<sub>H</sub>, Read and Write

bit	7	6	5	4	3	2	1	0
IPAR0	S3		S2		S1		S0	

**Interrupt Priority Assignment Register 1 (IPAR1)** Address 305<sub>H</sub>, Read and Write

bit	7	6	5	4	3	2	1	0
IPAR1	S7		S6		S5		S4	

**Interrupt Priority Assignment Register 2 (IPAR2)** Address 306<sub>H</sub>, Read and Write

bit	7	6	5	4	3	2	1	0
IPAR2			S10		S9		S8	

- S10 to S0 Interrupt Sources**
- '11' = highest priority group
  - '10' = second priority group
  - '01' = third priority group
  - '00' = lowest priority group

**Interrupt DOC Address Base (IDOC)** Address 300<sub>H</sub>, Read and Write

bit	7		4	3		0
IDOC	0	0	0	0	DOC Interrupt Address Base	

**Functional Block Description**

**Interrupt Vector (INT\_VEC)**

This is the vector which is being read during interrupt acknowledge cycle.

bit	7	4	3	0
<b>INT_VEC</b>	DOC Interrupt Address Base			Source Identifier

The 8-bit vector contains a 4-bit programmable interrupt address base, which can be programmed via IDOC, and a 4-bit interrupt source identifier (see **Table 2-28**).

**2.13.4 Interrupt Cascading**

The DOC Interrupt Controller supports two cascading schemes which can be selected by programming the IPC register.

**Interrupt Port Configuration Register (IPC)**

Reset Value: 00<sub>H</sub>

μP interface mode: read/write

Address: 301<sub>H</sub>

bit	7							0
<b>IPC</b>	0	0	MODE	SLA1	SLA0	CASM	IC1	IC0

*Note: Bits 7...6 are unused and are read as '0'.*

**MODE**      Interrupt Handling Mode  
 0...Intel scheme  
 1...Motorola scheme

**SLA1...0**    Slave Address  
 Used only in Slave Cascading Mode (refer to CASM).

**CASM**      Cascading Mode  
 0...Slave Cascading Mode  
 Pins IE0, IE1 are used as inputs. Interrupt acknowledge is accepted if an interrupt signal has been generated and the values on pins IE1...0 correspond to the programmed values in SLA1...0 (Slave Address).  
 1...Daisy Chaining Mode  
 Pin IE0, as Interrupt Enable Output, and pin IE1, as Interrupt Enable input, are used for building a Daisy Chain. Interrupt acknowledge is accepted if an interrupt signal has been generated and Interrupt Enable Input, IE1, is active "high" during a subsequent  $\overline{IACK}$  cycle(s). If pin INT goes active, Interrupt Enable Output, IE0, is immediately set to "low".



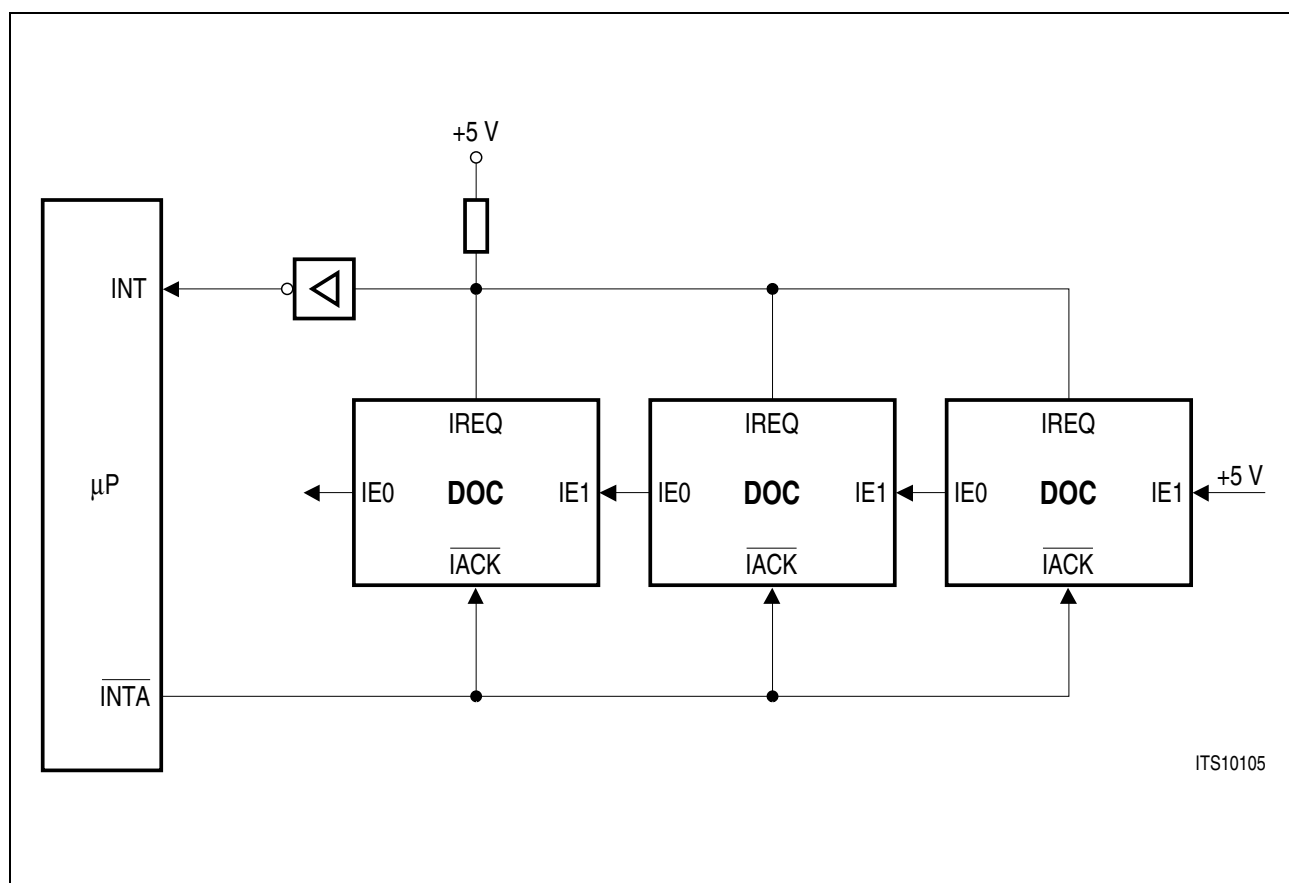
For Intel type microprocessor systems the 2-cycle interrupt acknowledge scheme is supported (80x86 mode).

### 2.13.4.2 Daisy Chaining

If selected via IPC register the Interrupt Enable pins IE0, IE1 are used for building a Daisy Chain by connecting the Interrupt Enable Output (IE0) of the higher priority device to the Interrupt Enable Input (IE1) of the lower priority device. The highest priority device has IE1 pulled high.

Refer to:

**Figure 2-50** for interrupt cascading in Siemens/Intel bus mode.



**Figure 2-50 Interrupt Cascading (Daisy Chaining) in Siemens/Intel Bus Mode**

For Intel type microprocessor systems the 2-cycle interrupt acknowledge scheme is supported 80X86 mode. Maximum available settling time for the chain: from the beginning of the first  $\overline{INTA}$  cycle to the beginning of the second.

For Motorola type  $\mu P$  systems the maximum available setting time for the chain is: from the beginning of the  $\overline{IACK}$  cycle to the falling edge of the  $\overline{RD}/\overline{DS}$  signal.

Refer to Motorola Timing, **Table 7-8**.

### 2.13.5 Global Interrupt Status Registers (IGIS0 and IGIS1)

The DOC Version 2.1 provides two new 8-bit interrupt status registers (IGIS0 and IGIS1) for applications in which the generated interrupt vector can not be used. The pending interrupt status is displayed by reading the registers.

#### Global Interrupt Status Registers (IGIS0)

Address 30AH  
Read only  
Reset Value 00H

bit	7							0
<b>IGIS0</b>	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0

#### Global Interrupt Status Registers (IGIS1)

Address 30BH  
Read only  
Reset Value 00H

bit	7			3			0	
<b>IGIS1</b>	0	0	0	0	0	don't care	IS9	don't care

*Note: Bits 7...3 are not used, and are read as '0'.*

For interrupts description see the following table with Interrupts Sources.

**Table 2-30**

Interrupt Sources Number	Interrupt Status	Interrupt Source	Reset Control for Bits in IGIS0 and IGID1
S0	IS0	ELIC0	Read access to registers: ELIC0: ISTA, ELIC0-EPIC: ISTA-E, CIFIFO ELIC0-SACCOA: ISTA, EXIR ELIC0-SACCOB: ISTA, EXIR
S1	IS1	ELIC1	Read access to registers: ELIC1: ISTA, ELIC1-EPIC: ISTA-E, CIFIFO ELIC1-SACCOA: ISTA, EXIR ELIC1-SACCOB: ISTA, EXIR
S2	IS2	SIDEC0	SIDEC0: ISTA, EXIR

**Table 2-30** (cont'd)

Interrupt Sources Number	Interrupt Status	Interrupt Source	Reset Control for Bits in IGIS0 and IGID1
S3	IS3	SIDEC1	SIDEC1: ISTA, EXIR
S4	IS4	SIDEC2	SIDEC2: ISTA, EXIR
S5	IS5	SIDEC3	SIDEC3: ISTA, EXIR
S6	IS6	OAK-Mail Box	write access to OBUSY
S7	IS7	GPIO Port	read access VDATA
S8	–	FSC	not available (see note below)
S9	IS9	UART	see <b>Table 2-36 on page 149</b>
S10	–	RTC	not available (see note below)

*Note: The FSC and RTC interrupts sources are reset by Interrupt Acknowledge line (IACK) when the appropriate Interrupt Vector is driven on the data bus. The other interrupts are reset by reading from or writing to the appropriate register in the interrupt source module.*

*Thus the FSC and RTC interrupts are not supported in the pending interrupt status register. It is recommended to mask the FSC and RTC interrupts in the Interrupt Mask Register, when working with the pending interrupt status (not using the interrupt vector).*

*Both interrupts can be generated via the  $\mu$ P-Mailbox interrupt by the DSP software as the DSP uses FSC interrupts internally. It may also count the FSC interrupts to e.g. 1 ms and then send a message to the  $\mu$ P.*

## 2.14 Universal Asynchronous Receiver/Transmitter (UART)

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem, and parallel-to-serial conversion on data characters received from the CPU. The  $\mu$ P can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error condition (parity, overrun, framing, or break interrupt).

The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by 1 to  $(2^{16} - 1)$ , and of producing a  $16 \times$  clock for driving the internal transmitter logic. Provisions have also been made to use this  $16 \times$  clock for driving the receiver logic.

The UART features full modem-control capability and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required for handling the communications link.

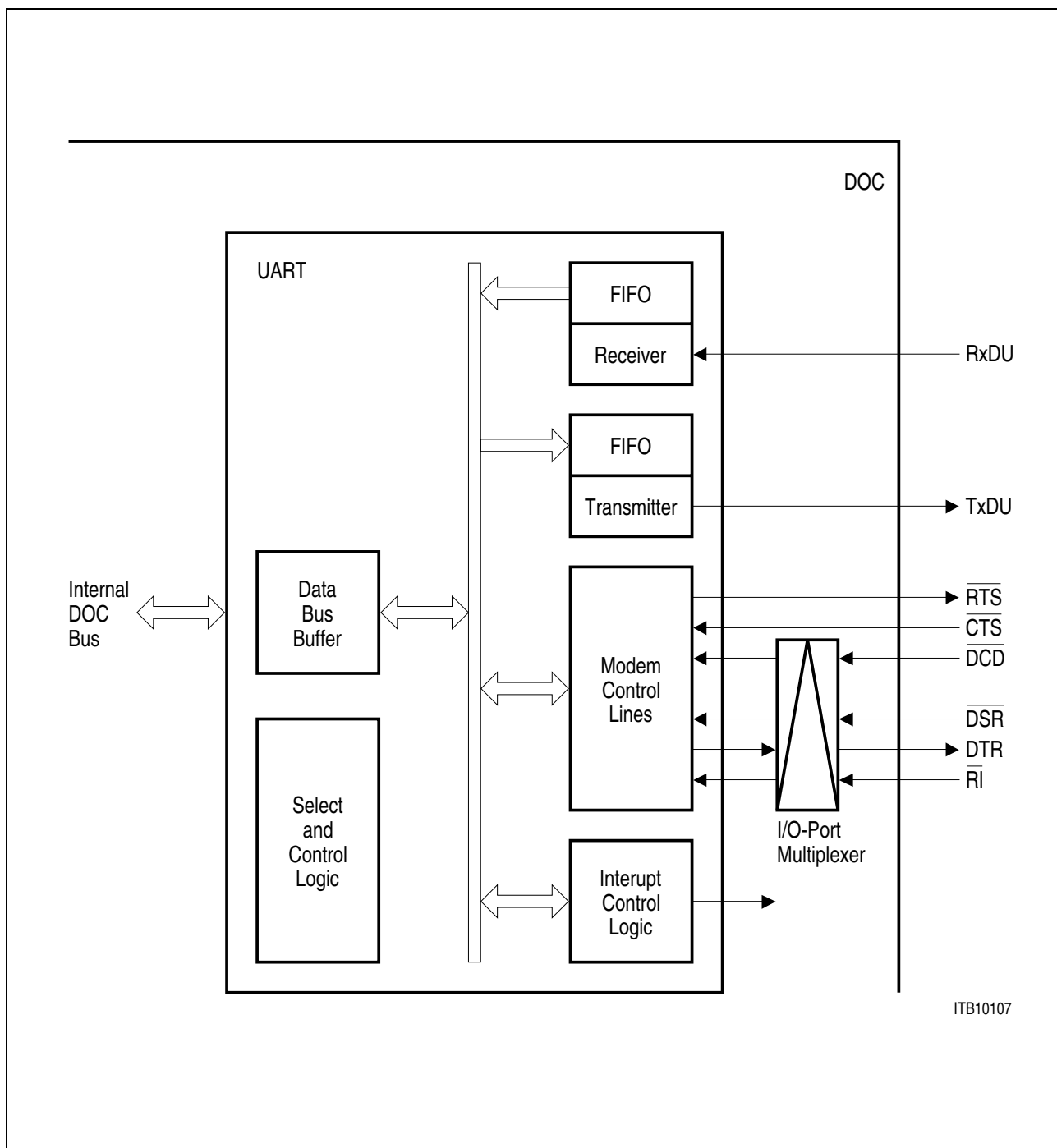
---

## Functional Block Description

The integrated UART is compatible to the standard 16C550A UART, **Figure 2-51**.

It has the following features:

- Receiver and transmitter are each buffered with 16-byte FIFO's (in the FIFO mode) to reduce the number of interrupts presented to the  $\mu\text{P}$
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from the serial data stream
- Holding and shift registers eliminate the need for precise synchronization between the  $\mu\text{P}$  and the serial data
- Independent control of transmit, receive, line status and data set interrupts
- Programmable baud rate generator for 300 Baud to 256 kBaud; it generates the internal  $16 \times$  clock using an internal clock source.
- Modem control functions ( $\overline{\text{CTS}}$ ,  $\overline{\text{RTS}}$ , DSR, DTR, RI and  $\overline{\text{DCD}}$ )
- False start bit detection
- Complete status reporting capabilities
- Fully programmable serial-interface characteristics:
  - 5-, 6-, 7-, 8- bit characters
  - Even, odd, or non-parity generation and detection
  - 1-, 1½-, or 2-stop bit generation
  - Baud generation (DC to 256 kBaud)
- Tri-state TTL drive capability for bidirectional data bus and control bus
- Line break generation and detection
- Internal diagnostic capabilities:
  - Loopback controls for communication link fault isolation
  - Break, parity, overrun, framing error simulation
- Fully prioritized interrupt system controls



ITB10107

**Figure 2-51 UART Block Diagram**

The system programmer may access any of the UART registers summarized in **Table 2-31** via the  $\mu\text{P}$ . These registers control UART operations including transmission and reception of data. Each register bit in **Table 2-31** has its name and reset state shown in **Table 2-33**.



2.14.1 Registers Overview

Table 2-31 Summary of Registers 1

Bit No.	Register Address					
	0 (DLAB = 0)	0 (DLAB = 0)	1 (DLAB = 0)	2	2	3
	Receiver Buffer Register (read only)	Transmitter Holding Register (write only)	Interrupt Enable Register	Interrupt Ident. Register (read only)	FIFO Control Register (write only)	Line Control Register
	RBR	THR	IER	IIR	FCR	LCR
0	Data bit 0 <sup>1)</sup>	Data bit 0 <sup>1)</sup>	Enable received data available interrupt (ERBFI)	'0' if interrupt is pending	FIFO enable (FEWO)	Word length select bit 0 (WLS0)
1	Data bit 1	Data bit 1	Enable transmitter holding register empty (ETBEI)	Interrupt ID bit 0 (IIDB0)	Receiver FIFO reset (RFR)	Word length select bit 1 (WLS1)
2	Data bit 2	Data bit 2	Enable receiver line status interrupt (ERLSI)	Interrupt ID bit 1 (IIDB1)	Transmitter FIFO reset (TFR)	Number of stop bits (STB)
3	Data bit 3	Data bit 3	Enable modem status interrupt (EDDSSI)	Interrupt ID bit 2 (IIDB2)	DMA mode select (DMS)	Parity enable (PEN)
4	Data bit 4	Data bit 4	0	0	Reserved	Even parity select (EPS)
5	Data bit 5	Data bit 5	0	0	Reserved	Stick parity (STP)

Functional Block Description

Table 2-31 Summary of Registers 1 (cont'd)

Bit No.	Register Address					
	0 (DLAB = 0)	0 (DLAB = 0)	1 (DLAB = 0)	2	2	3
	Receiver Buffer Register (read only)	Transmitter Holding Register (write only)	Interrupt Enable Register	Interrupt Ident. Register (read only)	FIFO Control Register (write only)	Line Control Register
	RBR	THR	IER	IIR	FCR	LCR
6	Data bit 6	Data bit 6	0	FIFOs enabled (FE) <sup>2)</sup>	RCVR FIFO trigger level (LSB)	Set break (SBR)
7	Data bit 7	Data bit 7	0	FIFOs enabled (FE) <sup>2)</sup>	RCVR FIFO trigger level (MSB)	Divisor latch access bit (DLAB)

Table 2-32 Summary of Registers 2

Bit No.	Register Address					
	4	5	6	7	0 (DLAB = 1)	1 (DLAB = 1)
	Modem Control Register	Line Status Register	Modem Status Register	Scratch Register	Divisor Latch (LS)	Divisor Latch (MS)
	MCR	LSR	MSR	SCR	DLL	DLM
0	Data terminal ready (DTR)	Data ready (DR)	Delta clear to send (DCTS)	Bit 0	Bit 0	Bit 8
1	Request to send (RTS)	Overrun error (OE)	Delta data set ready (DDSR)	Bit 1	Bit 1	Bit 9
2	Out 1	Parity error (PE)	Trailing edge ring indicator (TERI)	Bit 2	Bit 2	Bit 10
3	Out 2	Framing error (FE)	Delta data carrier detect (DDCD)	Bit 3	Bit 3	Bit 11

Functional Block Description

**Table 2-32 Summary of Registers 2 (cont'd)**

Bit No.	Register Address					
	4	5	6	7	0 (DLAB = 1)	1 (DLAB = 1)
	Modem Control Register	Line Status Register	Modem Status Register	Scratch Register	Divisor Latch (LS)	Divisor Latch (MS)
	MCR	LSR	MSR	SCR	DLL	DLM
4	Loop	Break interrupt (BI)	Clear to send (CTS)	Bit 4	Bit 4	Bit 12
5	0	Transmitter holding register (THRE)	Data set ready (DSR)	Bit 5	Bit 5	Bit 13
6	0	Transmitter empty (TEMT)	Ring indicator (RI)	Bit 6	Bit 6	Bit 14
7	0	Error in RCVR FIFO (EIRF) <sup>2)</sup>	Data carrier detect (DCD)	Bit 7	Bit 7	Bit 15

<sup>1)</sup> Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

<sup>2)</sup> These bits are always 0 in the SAB 16C450 compatible mode.

**Table 2-33 Register Reset Values**

Register/Signal	Reset Control	Reset State
Interrupt enable register	Master reset	0000 0000 <sup>1)</sup>
Interrupt identification register	Master reset	0000 0001
FIFO control register	Master reset	0000 0000
Line control register	Master reset	0000 0000
Modem control register	Master reset	0000 0000
Line status register	Master reset	0110 0000
Modem status register	Master reset	XXXX 0000 <sup>2)</sup>
SOUT	Master reset	High
INTR (RCVR errors)	Read LSR/MR	Low
INTR (RCVR data ready)	Read LSR/MR	Low

**Table 2-33 Register Reset Values** (cont'd)

Register/Signal	Reset Control	Reset State
INTR (THRE)	Read IIR/write THR/MR	Low
INTR (Modem status changes)	Read MSR/MR	Low
OUT2	Master reset	High
RTS	Master reset	High
DTR	Master reset	High
OUT1	Master reset	High
RCVR FIFO	MR/RFR • FEWO/ $\Delta$ FEWO	All bits low
XMIT FIFO	MR/T FR • FEWO/ $\Delta$ FEWO	All bits low

1) Boldface bits are permanently low.

2) Bits 7-4 are driven by the input signals.

### UART Address 2-0

Address signals connected to these 3 inputs select a UART register for the  $\mu$ P to read from or write to during data transfer. A table of registers and their addresses is shown below. Note that the state of the divisor latch access bit (DLAB), which is the most significant bit of the line control register, affects the selection of certain UART registers. The DLAB must be set high by the system software to access the baud rate generator divisor latches

**Table 2-34 UART Registers and Addresses**

DLAB	A2	A1	A0	Register
0	0	0	0	Receiver buffer (read), Transmitter holding register (write)
0	0	0	1	Interrupt enable
X	0	1	0	Interrupt identification (read)
X	0	1	0	FIFO control (write)
X	0	1	1	Line control
X	1	0	0	Modem control
X	1	0	1	Line status
X	1	1	0	Modem status
X	1	1	1	Scratch
1	0	0	0	Divisor latch (least significant byte)
1	0	0	1	Divisor latch (most significant byte)

**2.14.1.1 Line Control Register (LCR)**

bit	7	6	5	4	3	2	1	0
<b>LCR</b>	DLAB	SBR	STP	EPS	PEN	STB	WLS	

The system programmer specifies the format of the asynchronous data communications exchange and sets the divisor latch access bit via the line control register (LCR). The programmer can also read the contents of the line control register. The read capability simplifies system programming and eliminates the need for separate storage of the line characteristics in system memory.

**WLS0, WLS1** These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

Bit 1	Bit 0	Character Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

**STB** This bit specifies the number of stop bits transmitted and received in each serial character. If bit 2 is a logic 0, one stop bit is generated or checked in the transmitted data. If bit 2 is a logic 1 when a 5-bit word length is selected via bits 0 and 1, one and a half stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7-, or 8-bit word length is selected, two stop bits are generated. The receiver checks the first stop-bit only, regardless of the number of stop bits selected.

**PEN** This bit is the parity enable bit. When bit 3 is a logic 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and stop bit of the serial data. (The parity bit is used to produce an even or odd number of 1's when the data word bits and the parity bit are summed).

**EPS** This bit is the even parity select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1's is transmitted or checked in the data word bits and parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1's is transmitted or checked.

---

**Functional Block Description**

**STP** This bit is the stick parity bit. When bits 3, 4, and 5 are logic 1 the parity bit is transmitted and checked as a logic 0. If bits 3 and 5 are 1 and bit 4 is logic 0 then the parity bit is transmitted and checked as a logic 1. If bit 5 is a logic 0 stick parity is disabled.

**SBR** This bit is the break control bit. It causes a break condition to be transmitted by the UART. When it is set to a logic 1, the serial output (SOUT) is forced to the spacing (logic 0) state. The break is disabled by clearing bit 6 to a logic 0. The break control bit acts only on SOUT and has no effect on the transmitter logic.

*Note: This feature enables the  $\mu P$  to alert a terminal in a computer communications system. If the following sequence is used, no erroneous or extraneous characters will be transmitted because of the break*

1. Load on all O's pad character in response to THRE.
2. Set break after the next THRE.
3. Wait for the transmitter to be idle, (TEMT = 1) and clear break when normal transmission is to be restored. During the break, the transmitter can be used as a character timer to accurately establish the break duration.

**DLAB** This bit is the divisor latch access bit. It must be set high (logic 1) to access the divisor latches of the baud generator during a read or write operation. It must be set low (logic 0) to access the receiver buffer, the transmitter holding register, or the interrupt enable register.

### 2.14.1.2 Programmable Baud Rate Generator (Divisors)

The UART contains a programmable baud rate generator. The output frequency of the baud rate generator is  $16 \times$  the baud rate [divisor =  $(61.44 \text{ MHz} \div 5) \div (\text{baud rate} \times 16)$ ]. These divisor latches must be loaded during initialization to ensure proper operation of the baud rate generator. Upon loading of the divisor latch, a 16-bit baud counter is immediately loaded.

**Table 2-35** provides decimal divisors to use with crystal frequency of 61.44 MHz.

Using a divisor of zero is not recommended.

**Table 2-35 Baud Rates Using 61.44 MHz Crystal**

Desired Baud Rate	Decimal Divisor	Actual Baud Rate	Percentual Error Difference between the Desired and Actual Baud Rates
50	15360	50	0
300	2560	300	0
600	1280	600	0
1200	640	1200	0
2400	320	2400	0
4800	160	4800	0
9600	80	9600	0
19200	40	19200	0
38400	20	38400	0
76800	10	76800	0
256000	3	256000	0

**2.14.1.3 Line Status Register (LSR)**

This 8-bit register provides the  $\mu$ P with status information concerning the data transfer.

bit	7	6	5	4	3	2	1	0
<b>LSR</b>	EIRF	TEMT	THRE	BI	FE	PE	OE	DR

**DR** This bit is the receiver **Data Ready** indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the receiver buffer register or the FIFO. Bit 0 is reset to a logic 0 by reading all of the data in the receiver buffer register or the FIFO.

---

**Functional Block Description**

- OE** This bit is the **O**verrun **E**rror indicator. Bit 1 indicates that data in the receiver buffer register was not read by the  $\mu$ P before the next character was transferred into the receiver buffer register, thereby destroying the previous character. The OE indicator is set to a logic 1 upon detection of an overrun condition, and reset whenever the  $\mu$ P reads the contents of the line status register. If in the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is full and the next character has been completely received in the shift register. OE is indicated to the  $\mu$ P as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.
- PE** This bit is the **P**arity **E**rror indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the  $\mu$ P reads the contents of the line status register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the  $\mu$ P when its associated character is at the top of the FIFO.
- FE** This bit is the **F**raming **E**rror indicator. Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic 1 whenever the stop bit following the last data bit or parity bit is detected as a logic 0 bit (spacing level). The FE indicator is reset whenever the  $\mu$ P reads the contents of the line status register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the  $\mu$ P when its associated character is at the top of the FIFO. The UART will try to resynchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this “start” bit twice and then takes in the “data”.



---

**Functional Block Description**

**BI** This bit is the **B**reak **I**nterrupt indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the spacing (logic 0) state for longer than a full word transmission time (that is, the total time of start bit + data bits + parity + stop bits). The BI indicator is reset whenever the  $\mu$ P reads the contents of the line status register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the  $\mu$ P when its associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. The next character transfer is enabled after SIN goes to the marking state and receives the next valid start bit.

*Note: Bits 1 through 4 are the error conditions that produce a receiver line status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.*

**THRE** This bit is the **T**ransmitter **H**olding **R**egister **E**mpy indicator. Bit 5 indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the  $\mu$ P when the transmit holding register empty interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the transmitter holding register into the transmitter shift register. The bit is reset to logic 0 concurrently with the loading of the transmitter holding register by the CPU. In the FIFO mode this bit is set when the XMIT FIFO is empty; it is cleared when at least 1 byte is written to the XMIT FIFO.

**TEMT** This bit is the **T**ransmitter **E**mpy indicator. Bit 6 is set to a logic 1 whenever the transmitter holding register (THR) and the transmitter shift register (TSR) are both empty. It is reset to a logic 0 whenever either the THR or TSR contains a data character. In the FIFO mode this bit is set to one whenever the transmitter FIFO and shift register are both empty.

**EIRF** In the 16450 mode this is a 0. In the FIFO mode EIRF is set when there is at least one parity error, framing error or break indication in the FIFO. EIRF is cleared when the  $\mu$ P reads the LSR, if there are no subsequent errors in the FIFO.

*Note: The line status register is intended for read operations only. Writing to this register is not recommended as this operation is only used for factory testing.*

**2.14.1.4 FIFO Control Register (FCR)**

bit	7	6	5	4	3	2	1	0
<b>FCR</b>	FCR7	FCR6			DMS	TFR	RFR	FEWO

This is a write only register at the same address location as the IIR (the IIR is a read only register). This register is used to enable the FIFOs, clear the FIFOs, set the RCVR FIFO trigger level, and select the type of DMA signaling.

- FEWO** Writing a 1 to FEWO enables both the XMIT and RCVR FIFOs. Resetting FEWO will clear all bytes in both FIFOs. When changing from FIFO mode to 16450 mode and vice versa, data is automatically cleared from the FIFOs. This bit must be a 1 when other FCR bits are written to, or they will not be programmed.
- RFR** Writing a 1 to RFR clears all bytes in the RCVR FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing.
- TFR** Writing a 1 to TFR clears all bytes in the XMIT FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing.
- DMS** Setting DMS to a 1 will cause the RXRDY and TXRD pins to change from mode 0 to mode 1 if FEWO = 1 (see description of RXRDY and TXRDY pins).
- Bit 4 and 5** These bits are reserved for future use.
- FCR7:6** FCR6 and FCR7 are used to set the trigger level for the RCVR FIFO interrupt.

Bit 7	Bit 6	RCVR FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

**2.14.1.5 Interrupt Identification Register (IIR)**

bit	7	6	5	4	3	2	1	0
IIR	FE		0	0	IIDB2	IIDB1	IIDB0	

In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the interrupt identification register. The four levels of interrupt conditions in order of priority are receiver line status, received data ready, transmitter holding register empty, and modem status.

When the  $\mu$ P accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this  $\mu$ P access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete.

**Table 2-31** shows the contents of the IIR. Details on each bit follow:

**Table 2-36 IIR Register**

FIFO Mode Only	Interrupt Identification Register			Interrupt Set and Reset Functions			
	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	–	None	None	–
0	1	1	0	Highest	Receiver line status	Overrun error or parity error or framing error or break interrupt	Reading the line status register
0	1	0	0	Second	Receiver data available	Receiver data available or trigger level reached	Reading the receiver buffer register or the FIFO drops below the trigger level

Functional Block Description

Table 2-36 IIR Register (cont'd)

FIFO Mode Only	Interrupt Identification Register			Interrupt Set and Reset Functions				
	Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
1	1	0	0		Second	Character time-out indication	No characters have been removed from or input to the RCVR FIFO during the last 4 char. times and there is at least 1 char. in it during this time	Reading the receiver buffer register
0	0	1	0		Third	Transmitter holding register empty	Transmitter holding register empty	Reading the IIR register (if source of interrupt) or writing into the transmitter holding register
0	0	0	0		Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect	Reading the modem status register

- Bit 0** This bit can be used in an interrupt environment to indicate whether an interrupt condition is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending.
- IIDB0; IIDB1** These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in **Table 2-36**.
- IIDB2** In the 16C450 mode this bit is 0. In the FIFO mode this bit is set along with bit 2 when a time-out interrupt is pending.
- Bits 4 and 5** These two bits of the IIR are always logic 0.
- FE** These two bits are set when FEWO = 1.

**2.14.1.6 Interrupt Enable Register (IER)**

This register enables the five types of UART interrupts. Each interrupt can individually activate the interrupt (INTR) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the interrupt enable register (IER). Similarly, setting bits of this register to a logic 1 enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTR output signal. All other system functions operate in their normal manner, including the setting of the line status and modem status registers.

bit	7	6	5	4	3	2	1	0
<b>IER</b>	0	0	0	0	EDSSI	ERLSI	ETBEI	ERBFI

- ERBFI** This bit enables the received data available interrupt (and time-out interrupts in the FIFO mode) when set to logic 1.
- ETBEI** This bit enables the transmitter holding register empty interrupt when set to logic 1.
- ERLSI)** This bit enables the receiver status interrupt when set to logic 1.
- EDSSI** This bit enables the modem status interrupt when set to logic 1.
- Bit 4 through 7** These four bits are always logic 0.

**2.14.1.7 Modem Control Register (MCR)**

This register controls the interface with the modem or data set (or a peripheral device emulating a modem). The contents of the modem control register (MCR) are indicated in **Table 2-31**.

Functional Block Description

bit	7	6	5	4	3	2	1	0
<b>MCR</b>	0	0	0	LOOP0	OUT2	OUT1	RTS	DTR

**DTR** This bit controls the data terminal ready output. When bit 0 is set to a logic 1, the  $\overline{DTR}$  output is forced to a logic 0. When bit 0 is reset to a logic 0, the  $\overline{DTR}$  output is forced to a logic 1.

*Note: The  $\overline{DTR}$  output of the UART may be applied to an EIA inverting line driver (such as the 1488) to obtain the proper polarity input at the succeeding modem or data set.*

**RTS** This bit controls the request to send output. Bit 1 affects the  $\overline{RTS}$  output in a manner identical to that described above for bit 0.

**OUT1** This bit controls the output 1 signal, which is an auxiliary user-designated output. Bit 2 affects the  $\overline{OUT1}$  output in a manner identical to that described above for bit 0.

**OUT2** This bit controls the output 2 signal, which is an auxiliary user-designated output. Bit 3 affects the  $\overline{OUT2}$  output in a manner identical to that described above for bit 0.

**LOOP** This bit provides a local loopback feature for diagnostic testing of the UART. When bit 4 is set to logic 1, the following occurs: the transmitter serial output (SOUT) is set to the marking (logic 1) state; the receiver serial input (SIN) is disconnected; the output of the transmitter shift register is 'looped back' into the receiver shift register input; the four modem control inputs ( $\overline{CTS}$ ,  $\overline{DSR}$ ,  $\overline{RI}$ , and  $\overline{DCD}$ ) are disconnected; and the four modem control outputs ( $\overline{DTR}$ ,  $\overline{RTS}$ ,  $\overline{OUT1}$ , and  $\overline{OUT2}$ ) are internally connected to the four modem control inputs. The modem control output pins are forced to their inactive state (high). In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit and received data paths of the UART. In the diagnostic mode, the receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but the interrupt's sources are now the lower four bits of the modem control register instead of the four modem control inputs. The interrupts are still controlled by the interrupt enable register.

**Bits 5 through 7** These bits are permanently set to logic 0.

Details on each bit follow:

### 2.14.1.8 Modem Status Register (MSR)

This register provides the current state of the control lines from the modem (or peripheral device) to the CPU. In addition to this current-state information, four bits of the modem status register provide change information. These bits are set to a logic 1 whenever a control input from the modem changes state. They are reset to logic 0 whenever the  $\mu P$  reads the modem status register.

bit	7	6	5	4	3	2	1	0
<b>MSR</b>	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

**Table 2-31** shows the contents of the MSR. Details on each bit follow.

- DCTS** This bit is the delta clear to send indicator. Bit 0 indicates that the  $\overline{\text{CTS}}$  input to the chip has changed state since the last time it was read by the CPU.
- DDSR** This bit is the delta data set ready indicator. Bit 1 indicates that the  $\overline{\text{DSR}}$  input to the chip has changed state since the last time it was read by the CPU.
- TERI** This bit is the trailing edge of ring indicator detector. Bit 2 indicates that the  $\overline{\text{RI}}$  input to the chip has changed from a low to a high state.
- DDCD** This bit is the delta data carrier detect indicator. Bit 3 indicates that the  $\overline{\text{DCD}}$  input to the chip has changed state:  
*Note: Whenever bit 0, 1, 2, or 3 is set to logic 1, a modem status interrupt is generated.*
- CTS** This bit is the complement of the clear to send input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.
- DSR** This bit is the complement of the data set ready input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.
- RI** This bit is the complement of the ring indicator input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT1 in the MCR.
- DCD** This bit is the complement of the data carrier detect input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT2 in the MCR.

### 2.14.1.9 Scratchpad Register (SCR)

This 8-bit read/write register does not control the UART in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

bit							7				0	
SCR	X	X	X	X	X	X	X	X	X	X	X	

### 2.14.2 FIFO Interrupt Mode Operation

When the RCVR FIFO and receiver interrupts are enabled (FEWO = 1, ERBFI = 1) RCVR interrupts will occur as follows:

- a) The receive data available interrupt will be issued to the  $\mu$ P when the FIFO has reached its programmed trigger level; it will be cleared as soon as the FIFO drops below its programmed trigger level.
- b) The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt it is cleared when the FIFO drops below the trigger level.
- c) The receiver line status interrupt (IIR = 06), as before, has higher priority than the received data available (IIR = 04) interrupt.
- d) The data ready bit (DR) is set as soon as a character is transferred from the shift register to the RCVR FIFO. It is reset when the FIFO is empty.

When RCVR FIFO and receiver interrupts are enabled, RCVR FIFO time-out interrupts will occur as follows:

- a) A FIFO time-out interrupt will occur, if the following conditions exist:
  - at least one character is in the FIFO
  - the most recent serial character received was longer than 4 continuous character times ago (if 2 stop bits are programmed the second one is included in this time delay).
  - the most recent  $\mu$ P read of the FIFO was longer than 4 continuous character times ago.  
This will cause a maximum character received to interrupt issued delay of 160 ms at 300 Baud with a 12-bit character.
- b) Character times are calculated by using the RCLK input for a clock signal (this makes the delay proportional to the baudrate).
- c) When a time-out interrupt has occurred it is cleared and the timer reset when the  $\mu$ P reads one character from the RCVR FIFO.
- d) When a time-out interrupt has not occurred the time-out timer is reset after a new character is received or after the  $\mu$ P reads the RCVR FIFO.



---

## Functional Block Description

When the XMIT FIFO and transmitter interrupts are enabled (FEWO = 1, ETBEI = 1), XMIT interrupts will occur as follows:

- a) The transmitter holding register interrupt (02) occurs when the XMIT FIFO is empty; it is cleared as soon as the transmitter holding register is written to (1 to 16 characters may be written to the XMIT FIFO while servicing this interrupt) or the IIR is read.
- b) The transmitter FIFO empty indication will be delayed 1 character time minus the last stop bit time whenever the following occurs: THRE = 1, and there have not been at least two bytes at the same time in the transmit FIFO since the last THRE = 1. The first transmitter interrupt after changing FEWO will be immediate, if it is enabled.

Character time-out and RCVR FIFO trigger level interrupts have the same priority as the current received data available interrupt; XMIT FIFO empty has the same priority as the current transmitter holding register empty interrupt.

### 2.14.3 FIFO Polled Mode Operation

With FEWO = 1 resetting ERBFI, ETBEI, ERLSI, EDSSI or all to zero puts the UART in the FIFO polled mode of operation. Since the RCVR and XMITTER are controlled separately either one or both can be in the polled mode of operation.

In this mode the user's program will check RCVR and XMITTER status via the LSR. As stated previously:

- DR will be set as long as there is one byte in the RCVR FIFO.
- LSR1 to LSR4 will specify which error(s) has occurred. Character error status is handled the same way as when in the interrupt mode, the IIR is not affected since ERLSI = 0.
- THRE will indicate when the XMIT FIFO is empty.
- TEMT will indicate that both the XMIT FIFO and shift register are empty.
- EIRF will indicate whether there are any errors in the RCVR FIFO.

There is no trigger level reached or time-out condition indicated in the FIFO polled mode, however, the RCVR and XMIT FIFOs are still fully capable of holding characters.

## 2.15 General Purpose I/O Port (GPIO)

The General Purpose I/O Port (GPIO) is multifunctional system. GPIO has three working modes. GPIO can provide the I/O port unit, the SACCO-B0 unit or the UART unit. GPIO has four ports. Each port has special purpose

### 2.15.1 I/O Port Support Lines (Mode 0)

In I/O port mode each port can be configure as input port or as output port. When the data changes on input port, the DOC sends an interrupt.

The Version 2.1 provides a mask register VINTMASK thus changes of the port values do not lead to CPU interrupt generation.

#### Interrupt Mask Register for GPIO (VINTMASK)

Address 328A<sub>H</sub>

Read / Write

Reset Value 00<sub>H</sub>

bit	7	6	5	4	3	2	1	0
<b>VINTMASK</b>	0	0	0	0	MP3	MP2	MP1	MP0

*Note: Bits 7...4 are not used, and are read as '0'.*

*The GPIO interrupt is reset when the VDATR register is read.*

**MP3...0** Mask I/O Port 3 to 0 bits  
 '0' = the interrupt is enabled (the interrupt is not masked)  
 '1' = the interrupt is disabled (the interrupt is masked)

### 2.15.2 SACCO-B0 Support Lines (Mode 1)

When GPIO is configured to SACCO-B0 mode:

- SACCO-B0 drives DRQTB0 signal (dma transmit request) trough port0.
- SACCO-B0 drives DRQRB0 (dma receive request) signal trough port1.
- SACCO-B0 gets DACKB0 (dma acknowledge) signal trough port2.
- SACCO-B0 gets HFSCB0 (external frame synchronization clock) signal trough port3.

### 2.15.3 UART Support Lines (Mode 2)

When GPIO configure to uart mode:

- UART gets DSR (data set ready) signal trough port0.
- UART drives DTR (data transmit ready) signal trough port1.
- UART gets RI (ring indicator) signal trough port2.
- UART gets DCD (data carrier detect) signal trough port3.

### 2.15.4 Configuration Register (VCFGR)

Address: 320<sub>H</sub>

µP access mode: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
unused	MODE2	MODE1	MODE0	DIR3	DIR2	DIR1	DIR0

**MODE2...0** GPIO mode.  
 100...IO mode  
 010...SACCO-B0  
 001 - UART  
 All other values are not defined.  
 After reset GPIO is configured to IO mode, eventhough VCFGR reset mode is 00H.

**DIR 3...0** This field is valid only when in IO mode.  
 Each bit in the field defines if the respective port direction is input or output.  
 0 - input direction.  
 1 - output direction.

### 2.15.5 Data Register (VDATR)

Address: 321<sub>H</sub>

µP access mode: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
unused	unused	unused	unused	DAT3	DAT2	DAT1	DAT0

**DAT 3...0** Data which is written to this field drives the ports, when GPIO is configured to IO mode and the ports are configured as outputs. Each bit in the field drives the respective port, when it's configured as output. When a port is configured as input, the respective DATn-field bit is not used. When the GPIO is not configured to IO mode VDATR is not used. During read instruction of VDATR the value read, is the value which is driven on the ports, independently of the GPIO mode, and the configured direction of each port. Each unused bit will be read as '0'.

### 2.15.6 Version Number Register (VNR)

Address: 322<sub>H</sub>

μP access mode: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

unused	unused	unused	unused	VNR3	VNR2	VNR1	VNR0
--------	--------	--------	--------	------	------	------	------

**VNR3...0** DOC version number.  
 V1.1 0000  
 V2.1 0001  
 Each unused bit will be read as '0'.

### 2.16 Boundary Scan Support (JTAG)

The DOC provides a complete boundary scan support according to IEEE Std. 1149.1 specification for a cost effective board testing.

It consists of:

- Test access port controller (TAP)
- Four dedicated pins: JTCLK, TMS, TDI, TDO
- Tri-state of DOC output lines for board tests in production
- 32-bit DOC ID Register

All DOC-pins except the power supply pins ( $V_{DD}$ ,  $V_{DDP}$ ), the ground pins ( $V_{SS}$ ) and the external quartz clock pins (CLK40-XI, CLK40-XO), are included in the boundary scan. Depending on the pin functionality, one, two or three boundary scan cells are provided. A BSDL (Boundary Scan Description Language) file for the DOC is available.

#### 2.16.1 Boundary Scan

**Table 2-37 Boundary Scan Cell Types**

Pin Type	Number of Boundary Scan Cells	Usage
Input	1	Input
Output	2	Output, enable
I/O	3	Input, output, enable

When the TAP-controller is in the appropriate mode data is shifted into/out of the boundary scan via the pins TDI/TDO using the 6.25-MHz clock on pin TCK.

The ELIC-pins are included in the following sequence in the boundary scan.

### 2.16.2 TAP-Controller

The TAP controller implements a state machine defined in the JTAG standard IEEE1149.1.

The instruction register of the controller is extended to 4 bits in order to increase the number of instructions. This is necessary for the use of the Serial Emulation via the boundary scan interface:

**Table 2-38 Instruction Code of 4 Bit TAP Controller**

Instruction	Code
EXTEST	0000
INTEST	0001
SAMPLE / PRELOAD	0010
IDCODE	0011
Serial Emulation	01xx
Unused	10xx
BYPASS	11xx

The extended TAP controller uses a modified data path:

**Table 2-39 Data Path of 4 Bit TAP Controller**

Instruction Code	Input	Data Path	Output
11xx	TDI	→	TDO
000x, 0010	BSOUT	→	TDO
0011	BSOUT_ID	→	TDO
01xx	TDI2: SEIB:TDO (internal)	→	TDO
10xx	TDI3: V <sub>SS</sub> (not used, internal)	→	TDO

The next paragraph specifies the functionality of each instruction:

- **IDCODE**, the 32-bit identification register is serially read out via TDO. It contains the version number (4 bit), the device code (16 bit) and the manufacturer code (11 bits). The LSB is fixed to '1'.

Ver. No.	Device Code (Part Number)	Manufacturer Code (ID)	
0001	0000 0000 0011 1000	0000 1000 001	1 →TDO
0000 = DOC V1.1			
0001 = DOC V2.1			

Siemens provides for the DOC upon request a Boundary Scan Description Language, so called DSDL File.

## **2.17 Reset Logic**

During DOC HW reset:

- All signal lines with input/output (I/O) capability are switched to input direction.
- The state of each output pin is as defined in the pin description tables (**Table 1-1** to **Table 1-7**).
- The integrated clock generator provides the necessary clocks to both ELICs. Both ELICs PDC and DCL are driven by the internal PDC8 (= CLK16 div. by 2). Both ELICs PFS and FSC are driven by the internal PFS (= CLK16 div. by 2048).
- PDC2, PDC4, PDC8, PFS, DCL and FSC are driven internally, but these I/O pins are not driven by the DOC, and stay in tri-state. PDC2, PDC4, PDC8 and PFS starts to be driven by the DOC after the first write access to CCSEL0, and only if the DOC were configured as MASTER. DCL and FSC starts to be driven by the DOC after the first write access to CCSEL1, and only if these signals were configured as outputs.
- The OAK clock frequency is defined according to FRQ1...0 input pins. If FRQ1...0 pins are '10' (OAK clock frequency = 40 MHz), then a 40 MHz crystal must be connected to pins CLK40-XI and CLK40-XO.
- CLK 61.44 MHz must be provided to the DOC via CLK61 input pin.
- CLK30, CLK15 and CLK7 are driven by the DOC, as usual
- All the registers get the reset value, that defined in the register overview, **section 5**.

### 3 Operational Description

#### 3.1 ELIC0 and ELIC1

The ELIC, designed as a flexible line-card controller, has the following main applications:

- Digital line cards, with the CFI typically configured as IOM-2, IOM-1 (MUX) or SLD.
- Analog line cards, with the CFI typically configured as IOM-2 or SLD.
- Key systems, where the ELIC's ability to mix CFI-configurations is utilized.

To operate the ELIC the user must be familiar with the device's microprocessor interface, interrupt structure and reset logic. Also, the operation of the ELIC's component parts should be understood.

The devices major components are the EPIC-1, the SACCO-A and SACCO-B, and the D-channel arbiter. While EPIC-1 and SACCO-B may all be operated independently of each other, the D-channel arbiter can be used to interface the SACCO-A to the CFI of the EPIC-1. This mode of operation may be considered to utilize the ELIC most extensively. The initialization example, with which this operational description closes, will therefore set the ELIC to operate in this manner.

##### 3.1.1 Interrupt Structure and Logic

The ELIC-signals events that the  $\mu P$  should know about immediately by emitting an interrupt request on the  $\overline{INT}$ -line. To indicate the detailed cause of the request a tree of interrupt status registers is provided.

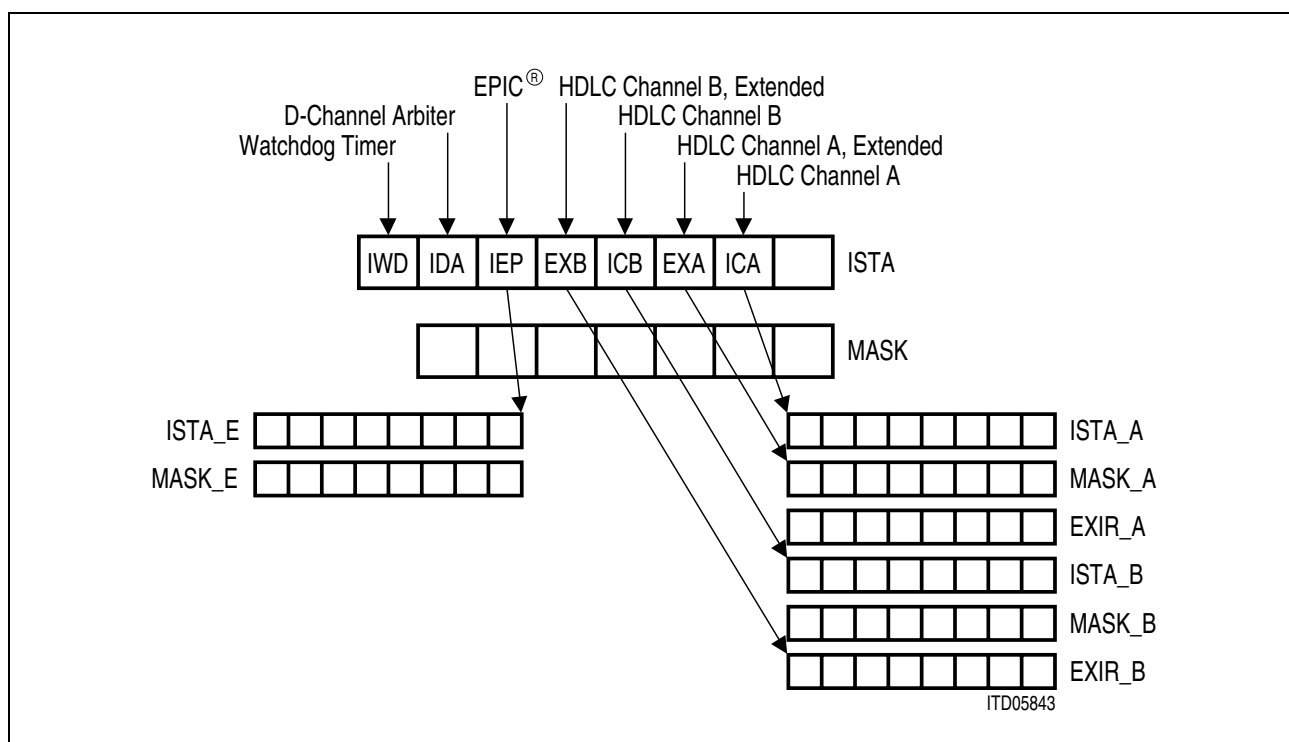


Figure 3-1 ELIC<sup>®</sup> Interrupt Structure

## Operational Description

When serving an ELIC-interrupt, the user first reads the top level interrupt status register (ISTA). This register flags which subblock has generated the request. If a subblock can issue different interrupt types a local ISTA/EXIR exists.

A read of the top level ISTA-register resets bits IWD and IDA. The other bits are reset when reading the corresponding local ISTA- or EXIR-registers.

The  $\overline{\text{INT}}$ -output is level active. It stays active until all interrupt sources have been serviced. If a new status bit is set while an interrupt is being serviced, the  $\overline{\text{INT}}$  stays active. However, for the duration of a write access to the MASK-register the  $\overline{\text{INT}}$ -line is deactivated. When using an edge-triggered interrupt controller, it is thus recommended to rewrite the MASK-register at the end of any interrupt service routine.

### Masking Interrupts

The watchdog timer interrupt can not be masked. Setting the MASK.IDA-bit masks the ISTA.IDA-interrupt: a D-channel arbiter interrupt will then neither activate the  $\overline{\text{INT}}$ -line nor be indicated in the ISTA-register. Setting the MASK.IEP/EXB/ICB/EXA or ICA-bits only masks the  $\overline{\text{INT}}$ -line; that is, with a set top level MASK bit these EPIC-1 and SACCO interrupts are indicated in the ISTA-register but they will not activate the  $\overline{\text{INT}}$ -line.

For the ISTA\_E, ISTA\_A and ISTA\_B registers local masking is also provided. Every interrupt source indicated in these registers can be selectively masked by setting the respective bit of the local MASK-register. Such locally masked interrupts will not be indicated in the local or the top ISTA-register, nor will they activate the  $\overline{\text{INT}}$ -line.

Locally masked interrupts are internally stored. Thus, resetting the local mask will release the interrupt to be indicated in the local interrupt register, flagged in the top level ISTA-register, and to activate the  $\overline{\text{INT}}$ -line.

### 3.1.2 Clocking

To operate properly, the ELIC always requires a PDC-clock.

To synchronize the PCM-side, the ELIC should normally also be provided with a PFS-strobe. In most applications, the DCL and FSC will be output signals of the ELIC, derived from the PDC via prescalers.

If the required CFI-data rate cannot be derived from the PDC, DCL and FSC can also be programmed as input signals. This is achieved by setting the EPIC-1 CMD1:CSS-bit. Frequency and phase of DCL and FSC may then be chosen almost independently of the frequency and phase of PDC and PFS. However, the CFI-clock source **must** still be synchronous to the PCM-interface clock source; i.e. the clock source for the CFI-interface and the clock source for the PCM-interface must be derived from the same master clock.



### 3.1.3 EPIC<sup>®</sup>-1 Operation

The EPIC-1 component of the ELIC is principally an intelligent switch of PCM-data between two serial interfaces, the system interface (PCM-interface) and the configurable interface (CFI). Up to 128 channels per direction can be switched dynamically between the CFI and the PCM-interfaces. The EPIC-1 performs non-blocking space and time switching for these channels which may have a bandwidth of 16, 32 or 64 kbit/s.

Both interfaces can be programmed to operate at different data rates of up to 8.192 Mbit/s. The PCM-interface consists of up to four duplex ports with a tri-state control signal for each output line. The configurable interface can be selected to provide either four duplex ports or 8 bi-directional (I/O) ports.

The configurable interface incorporates a control block (layer-1 buffer) which allows the  $\mu$ P to gain access to the control channels of an IOM- (ISDN-Oriented Modular) or SLD- (Subscriber Line Data) interface. The EPIC-1 can handle the layer-1 functions buffering the C/I and monitor channels for IOM compatible devices and the feature control and signaling channels for SLD compatible devices. One major application of the EPIC-1 is therefore as line card controller on digital and analog line cards. The layer-1 and codec devices are connected to the CFI, which is then configured to operate as, IOM-2, SLD or multiplexed IOM-1 interface.

The configurable interface of the EPIC-1 can also be configured as plain PCM-interface i.e. without IOM- or SLD-frame structure. Since it's possible to operate the two serial interfaces at different data rates, the EPIC-1 can then be used to adapt two different PCM- systems.

The EPIC-1 can handle up to 32 ISDN-subscribers with their 2B +D channel structure or up to 64 analog subscribers with their 1B channel structure in IOM-configuration. In SLD- configuration up to 16 analog subscribers can be accommodated.

The system interface is used for the connection to a PCM-back plane. On a typical digital line card, the EPIC-1 switches the ISDN B-channels and, if required, also the D-channels to the PCM-back plane. Due to its capability to dynamically switch the 16-kbit/s D-channel, the EPIC-1 is one of the fundamental building blocks for networks with either central, decentral or mixed signaling and packet data handling architecture.

#### 3.1.3.1 PCM-Interface

The serial PCM-interface provides up to four duplex ports consisting each of a data transmit (TxD#), a data receive (RxD#) and a tri-state control ( $\overline{\text{TSC\#}}$ ) line. The transmit direction is also referred to as the upstream direction, whereas the receive direction is referred to as the downstream direction.

Data is transmitted and received at normal TTL / CMOS-levels, the output drivers being of the tri-state type. Unassigned time-slots may be either be tri-stated, or programmed to transmit a defined idle value. The selection of the states "high impedance" and "idle value" can be performed with a two bit resolution. This tri-state capability allows several

---

## Operational Description

devices to be connected together for concentrator functions. If the output driver capability of the EPIC-1 should prove to be insufficient for a specific application, an external driver controlled by the  $\overline{TSC\#}$  can be connected.

The **PCM-standby function** makes it possible to switch all PCM-output lines to high impedance with a single command. Internally, the device still works normally. Only the output drivers are switched off.

The number of time-slots per 8-kHz frame is programmable in a wide range (from 4 to 128). In other words, the **PCM-data rate can range between 256 kbit/s up to 8.192 Mbit/s**. Since the overall switching capacity is limited to 128 time-slots per direction, the number of PCM-ports also depends on the required number of time-slots: in case of 32 time-slots per frame (2.048 Mbit/s) for example, four highways are available, in case of 128 time-slots per frame (8.192 Mbit/s), only one highway is available.

The partitioning between number of ports and number of bits per frame is defined by the **PCM-mode**. There are four PCM-modes.

The timing characteristics at the PCM-interface (data rate, bit shift, etc.) can be varied in a wide range, but they are the same for each of the four PCM-ports, i.e. if a data rate of 2.048 Mbit/s is selected, all four ports run at this data rate of 2.048 Mbit/s.

The PCM-interface has to be clocked with a **PCM-Data Clock (PDC)** signal having a frequency equal to or twice the selected PCM-data rate. In **single clock rate** operation, a frame consisting of 32 time-slots, for example, requires a PDC of 2.048 MHz. In **double clock rate** operation, however, the same frame structure would require a PDC of 4.096 MHz.

For the synchronization of the time-slot structure to an external PCM-system, a **PCM-Framing Signal (PFS)** must be applied. The EPIC-1 evaluates the rising PFS edge to reset the internal time-slot counters. In order to adapt the PFS-timing to different timing requirements, the EPIC-1 can latch the PFS-signal with either the rising or the falling PDC- edge. The PFS-signal defines the position of the first bit of the internal PCM-frame. The actual position of the external upstream and downstream PCM-frames with respect to the framing signal PFS can still be adjusted using the **PCM-offset function** of the EPIC-1. The offset can then be programmed such that PFS marks any bit number of the external frame.

Furthermore it is possible to select either the rising or falling PDC-clock edge for transmitting and sampling the PCM-data.

Usually, the repetition rate of the applied framing pulse PFS is identical to the frame period (125  $\mu$ s). If this is the case, the **loss of synchronism indication function** can be used to supervise the clock and framing signals for missing or additional clock cycles. The EPIC-1 checks the PFS-period internally against the duration expected from the programmed data rate. If, for example, double clock operation with 32 time-slots per frame is programmed, the EPIC-1 expects 512 clock periods within one PFS-period. The synchronous state is reached after the EPIC-1 has detected two consecutive correct

## Operational Description

frames. The synchronous state is lost if one bad clock cycle is found. The synchronization status (gained or lost) can be read from an internal register and each status change generates an interrupt.

### 3.1.3.2 Configurable Interface

The EPIC-1 provides up to four ports consisting each of a data output (DD#) and a data input (DU#) line. The output pins are called “Data Downstream” pins and the input pins are called “Data Upstream” pins. These modes are especially suited to realize a standard serial PCM-interface (PCM-highway) or to implement an IOM (ISDN-Oriented Modular) interface. The IOM-interface generated by the EPIC-1 offers all the functionality like C/I- and monitor channel handling required for operating all kinds of IOM compatible layer-1 and codec devices.

Data is transmitted and received at normal TTL/CMOS-levels at the CFI. **Tri-state or open-drain output drivers** can be selected. In case of open-drain drivers, external pull-up resistors are required. Unassigned output time-slots may be switched to high impedance or be programmed to transmit a defined idle value. The selection between the states “high impedance” or “idle value” can be performed on a per time-slot basis.

The **CFI-standby function** switches all CFI-output lines to high impedance with a single command. Internally the device still works normally, only the output drivers are switched off.

The number of time-slots per 8-kHz frame is programmable from 2 to 128. In other words, the **CFI-data rate can range between 128 kbit/s up to 8.192 Mbit/s**. Since the overall switching capacity is limited to 128 time-slots per direction, the number of CFI-ports also depends on the required number of time-slots: in case of 32 time-slots per frame (2.048 Mbit/s) for example, four highways are available, in case of 128 time-slots per frame (8.192 Mbit/s), only one highway is available. Usually, the number of bits per 8-kHz frame is an integer multiple of the number of time-slots per frame (1 time-slot = 8 bits).

The timing characteristics at the CFI (data rate, bit shift, etc.) can be varied in a wide range, but they are the same for each of the four CFI-ports, i.e. if a data rate of 2.048 Mbit/s is selected, all four ports run at this data rate of 2.048 Mbit/s. It is thus not possible to have one port used in IOM-2 line card mode (2.048 Mbit/s) while another port is used in IOM-2 terminal mode (768 kbit/s)!

*Note: The integrated PCM-DSP interface unit (PEDIV) works correctly only at 2.048 Mbit/s or 4.096 Mbit/s data rate.*

The clock and framing signals necessary to operate the configurable interface may be derived either from the clock and framing signals of the PCM-interface (PDC and PFS pins), or may be fed in directly via the DCL- and FSC-pins.

In the first case, the CFI-data rate is obtained by internally dividing down the PCM-clock signal PDC. Several prescaler factors are available to obtain the most commonly used

**Operational Description**

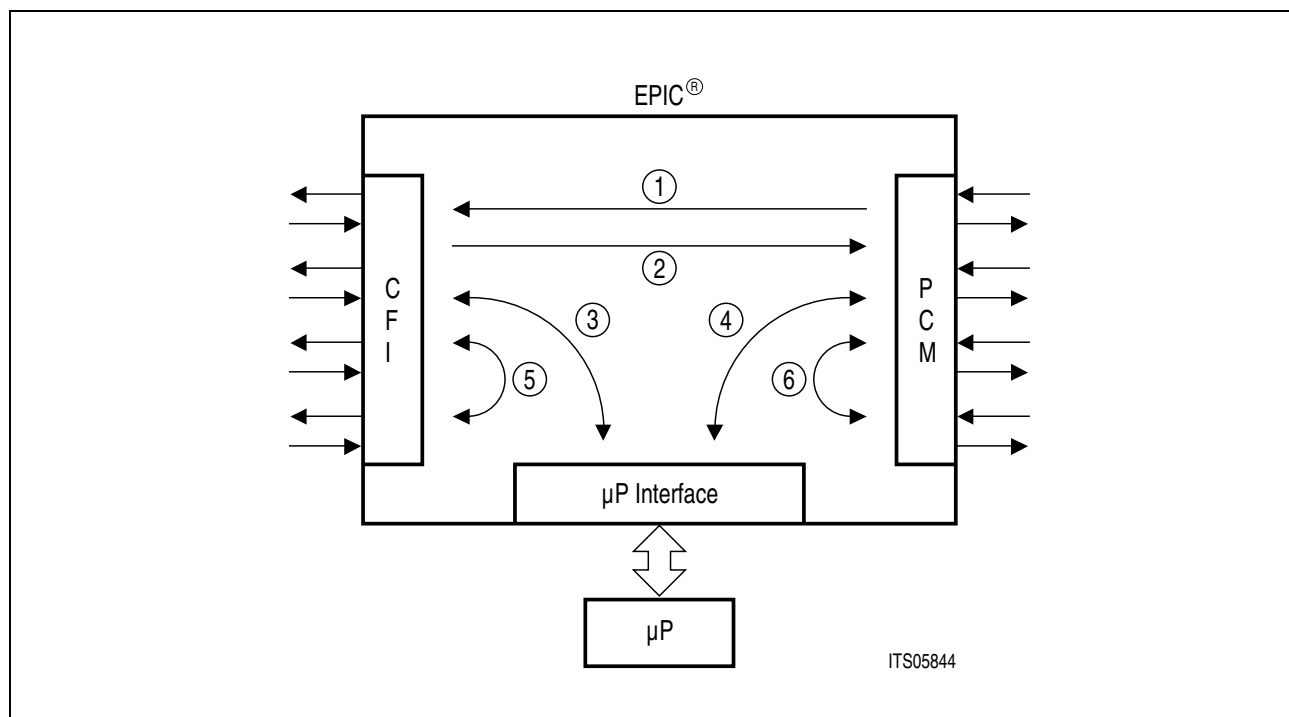
data rates. A CFI reference clock (CRCL) is generated out of the PDC-clock. The PCM-framing signal PFS is used to synchronize the CFI-frame structure. Additionally, the EPIC-1 generates clock and framing signals as outputs to operate the connected subscriber circuits such as layer-1 and codec filter devices. The generated data clock DCL has a frequency equal to or twice the CFI-data rate.

Note that if PFS is selected as the framing signal source, the FSC-signal is an output with a fixed timing relationship with respect to the CFI-data lines. The relationship between FSC and the CFI-frame depends only on the selected FSC-output wave form (CMD2- register). The CFI-offset function shifts both the frame and the FSC-output signal with respect to the PFS-signal.

In the second case, the CFI-data rate is derived from the DCL-clock, which is now used as an input signal. The DCL-clock may also first be divided down by internal prescalers before it serves as the CFI reference clock CRCL and before defining the CFI-data rate. The framing signal FSC is used to synchronize the CFI-frame structure.

**3.1.3.3 Switching Functions**

The major tasks of the EPIC-1 part is to dynamically switch PCM-data between the serial PCM-interface, the serial configurable interface (CFI) and the parallel  $\mu$ P-interface. All possible switching paths are shown in **Figure 3-2**.



**Figure 3-2 Switching Paths Inside the EPIC<sup>®</sup>-1**

Note that the time-slot selections in upstream direction are completely independent of the time-slot selections in downstream direction.

### CFI - PCM Time-Slot Assignment

Switching paths 1 and 2 of **Figure 3-2** can be realized for a total number of 128 channels per path, i.e. 128 time-slots in upstream and 128 time-slots in downstream direction. To establish a connection, the  $\mu\text{P}$  writes the addresses of the involved CFI and PCM time-slots to the control memory. The actual transfer is then carried out frame by frame without further  $\mu\text{P}$ -intervention.

The switching paths 5 and 6 can be realized by programming time-slot assignments in the control memory. The total number for such loops is limited to the number of available time-slots at the respective opposite interface, i.e. looping back a time-slot from CFI to CFI requires a spare upstream PCM time-slot and looping back a time-slot from PCM to PCM requires a spare downstream and upstream CFI time-slot.

time-slot switching is always carried out on 8-bit time-slots, the actual position and number of transferred bits can however be limited to 4-bit or 2-bit sub time-slots within these 8-bit time-slots. On the CFI-side, only one sub time-slot per 8-bit time-slot can be switched, whereas on the PCM-interface up to 4 independent sub time-slots can be switched.

### Sub Time-Slot Switching

Sub time-slot positions at the PCM-interface can be selected at random, i.e. each single PCM time-slot may contain any mixture of 2- and 4-bit sub time-slots. A PCM time-slot may also contain more than one sub time-slot. On the CFI however, two restrictions must be observed:

- Each CFI time-slot may contain one and one only sub time-slot.
- The sub-slot position for a given bandwidth within the time-slot is fixed on a per port basis.

### $\mu\text{P}$ -Transfer

Switching paths 3 and 4 of **Figure 3-2** can be realized for all available time-slots. Path 3 can be implemented by defining the corresponding CFI time-slots as “ $\mu\text{P}$ -channels” or as “pre-processed channels”.

Each single time-slot can individually be declared as “ $\mu\text{P}$ -channel”. If this is the case, the  $\mu\text{P}$  can write a static 8-bit value to a downstream time-slot which is then transmitted repeatedly in each frame until a new value is loaded. In upstream direction, the  $\mu\text{P}$  can read the received 8-bit value whenever required, no interrupts being generated.

The “**pre-processed channel**” option must always be applied to two consecutive time-slots. The first of these time-slots must have an even time-slot number. If two time-slots are declared as “pre-processed channels”, the first one can be accessed by the monitor/feature control handler, which gives access to the frame via a 16-byte FIFO. Although this function is mainly intended for IOM- or SLD-applications, it could also be used to transmit or receive a “burst” of data to or from a 64-kbit/s channel. The second

---

## Operational Description

pre-processed time-slot, the odd one, is also accessed by the  $\mu\text{P}$ . In downstream direction a 4-, 6- or 8-bit static value can be transmitted. In upstream direction the received 8-bit value can be read. Additionally, a change detection mechanism will generate an interrupt upon a change in any of the selected 4, 6 or 8 bits.

Pre-processed channels are usually programmed after Control Memory (CM) reset during device initialization. Resetting the CM sets all CFI time-slots to unassigned channels (CM code '0000'). Of course, pre-processed channels can also be initialized or re-initialized in the operational phase of the device.

To program a pair of pre-processed channels the correct code for the selected handling scheme must be written to the CM. **Figure 3-3** gives an overview of the available pre-processing codes and their application.

*Note: To operate the D-channel arbiter, an IOM-2 configuration with central-, or decentral D-channel handling should be programmed. With the D-channel arbiter enabled, D-channel bits are handled by the SACCO-A.*

Operational Description

DD Application	Even Control Memory Address MAAR = 0.....0		Odd Control Memory Address MAAR = 0.....1		Output at the Configurable Interface Downstream Preprocessed Channels	
	Code Field MACR = 0111...	Data Field MADR = .....	Code Field MACR = 0111...	Data Field MADR = .....	Even Time-Slot	Odd Time-Slot
Decentral D Channel Handling	1 0 0 0	1 1 C/I 1 1	1 0 1 1	X X X X X X X X	m m m m m m m m   - - C/I   m m	Monitor Channel Control Channel
Central D Channel Handling	1 0 1 0	1 1 C/I 1 1	PCM Code for a 2 Bit Sub. Time-Slot	Pointer to a PCM Time-Slot	m m m m m m m m   D D C/I   m m	Monitor Channel Control Channel
6 Bit Signaling (e.g. analog IOM <sup>®</sup> )	1 0 1 0	SIG 1 1	1 0 1 1	X X X X X X X X	m m m m m m m m   SIG   m m	Monitor Channel Control Channel
8 Bit Signaling (e.g. SLD)	1 0 1 0	SIG	1 0 1 1	X X X X X X X X	m m m m m m m m   SIG	Feature Control Channel Signaling Channel
SACCO_A D Channel Handling	1 0 1 0	1 1 C/I M <sup>r</sup> R 1	1 0 1 1	X X X X X X X X	m m m m m m m m   D D C/I   m m	Monitor Channel Control Channel

ITD05845

DU Application	Even Control Memory Address MAAR = 1.....1		Odd Control Memory Address MAAR = 1.....1		Input from the Configurable Interface Upstream Preprocessed Channels	
	Code Field MACR = 0111...	Data Field MADR = .....	Code Field MACR = 011...	Data Field MADR = .....	Even Time-Slot	Odd Time-Slot
Decentral D Channel Handling	1 0 0 0	1 1 C/I 1 1	0 0 0 0	X X X X X X X X	m m m m m m m m   - - C/I   m m	Monitor Channel Control Channel
Central D Channel Handling	1 0 0 0	1 1 C/I 1 1	PCM Code for a 2 Bit Sub. Time-Slot	Pointer to a PCM Time-Slot	m m m m m m m m   D D C/I   m m	Monitor Channel Control Channel
6 Bit Signaling (e.g. analog IOM <sup>®</sup> )	1 0 1 0	SIG Actual Value X X	1 0 1 0	SIG Stable Value X X	m m m m m m m m   SIG   m m	Monitor Channel Control Channel
8 Bit Signaling (e.g. SLD)	1 0 1 1	SIG Actual Value	1 0 1 1	SIG Stable Value	m m m m m m m m   SIG	Feature Control Channel Signaling Channel

- m : Monitor channel bits, these bits are treated by the monitor/feature control handler
- : Inactive sub. time-slot, in downstream direction these bits are tristated (OMDR : COS = 0) or set to logical 1 (OMDR : COS = 1)
- C/I : Command/Indication channel, these bits are exchanged between the CFI in/output and the CM data field. A change of the C/I bits in upstream direction causes an interrupt (ISTA : SFI). The address of the change is stored in the CIFIFO
- D : D channel, these D channel bit switched to and from the PCM interface, or handled by the SACCO\_A, if the D channel arbiter is enabled.
- SIG : Signaling Channel, these bits are exchanged between the CFI in/output and the CM data field. The SIG value which actual value was present in the last frame is stored as the actual value in the even address CM location. The stable value is updated if a valid change in the actual value has been detected according to the last look algorithm. A change of the SIG stable value in upstream direction causes an interrupt (ISTA : CFI). The address of the change is stored in the CIFIFO.

ITD05846

Figure 3-3 Pre-Processed Channel Codes



## Synchronous Transfer

For two channels, all switching paths of **Figure 3-2** can also be realized using Synchronous Transfer. The working principle is that the  $\mu\text{P}$  specifies an input time-slot (source) and an output time-slot (destination). Both source and destination time-slots can be selected independently from each other at either the PCM- or CFI-interfaces. In each frame, the EPIC-1 first transfers the serial data from the source time-slot to an internal data register from where it can be read and if required overwritten or modified by the  $\mu\text{P}$ . This data is then fed forward to the destination time-slot.

### 3.1.3.4 Special Functions

#### Hardware Timer

The EPIC-1 provides a hardware timer which continuously interrupts the  $\mu\text{P}$  after programmable time periods. The timer period can be selected in the range of 250  $\mu\text{s}$  up to 32 ms in multiples of 250  $\mu\text{s}$ . Beside the interrupt generation, the timer can also be used to determine the last look period for 6 and 8-bit signaling channels on IOM-2 and SLD-interfaces and for the generation of an FSC-multiframe signal.

#### Power and Clock Supply Supervision

The +3.3 V power supply line and the clock lines are continuously checked by the EPIC-1 for spikes that may disturb its proper operation. If such an inappropriate clocking or power failure occurs, the  $\mu\text{P}$  is requested to reinitialize the device.

### 3.1.4 SACCO-A/B

**Chapter 2.1.2.5** provides a detailed functional SACCO-description. This operational section will therefore concentrate on outlining how to run these HDLC-controllers.

With the SACCO initialized as outlined in **chapter 3.1.6.3**, it is ready to transmit and receive data. Data transfer is mainly controlled by commands from the CPU to the SACCO via the CMDR-register, and by interrupt indications from SACCO to CPU. Additional status information, which need not trigger an interrupt, is available in the STAR-register.

#### 3.1.4.1 Data Transmission in Interrupt Mode

In transmit direction  $2 \times 32$ -byte FIFO-buffers (transmit pools) are provided for each channel. After checking the XFIFO-status by polling the Transmit FIFO Write Enable bit (XFW in STAR-register) or after a Transmit Pool Ready (XPR) interrupt, up to 32 bytes may be entered by the CPU to the XFIFO.

The transmission of a frame can then be started issuing a XTF/XPD or XDD command via the CMDR-register. If prepared data is sent, an end of message indication (CMDR:XME) must also be set. If transparent or direct data is sent, CMDR:XME may but

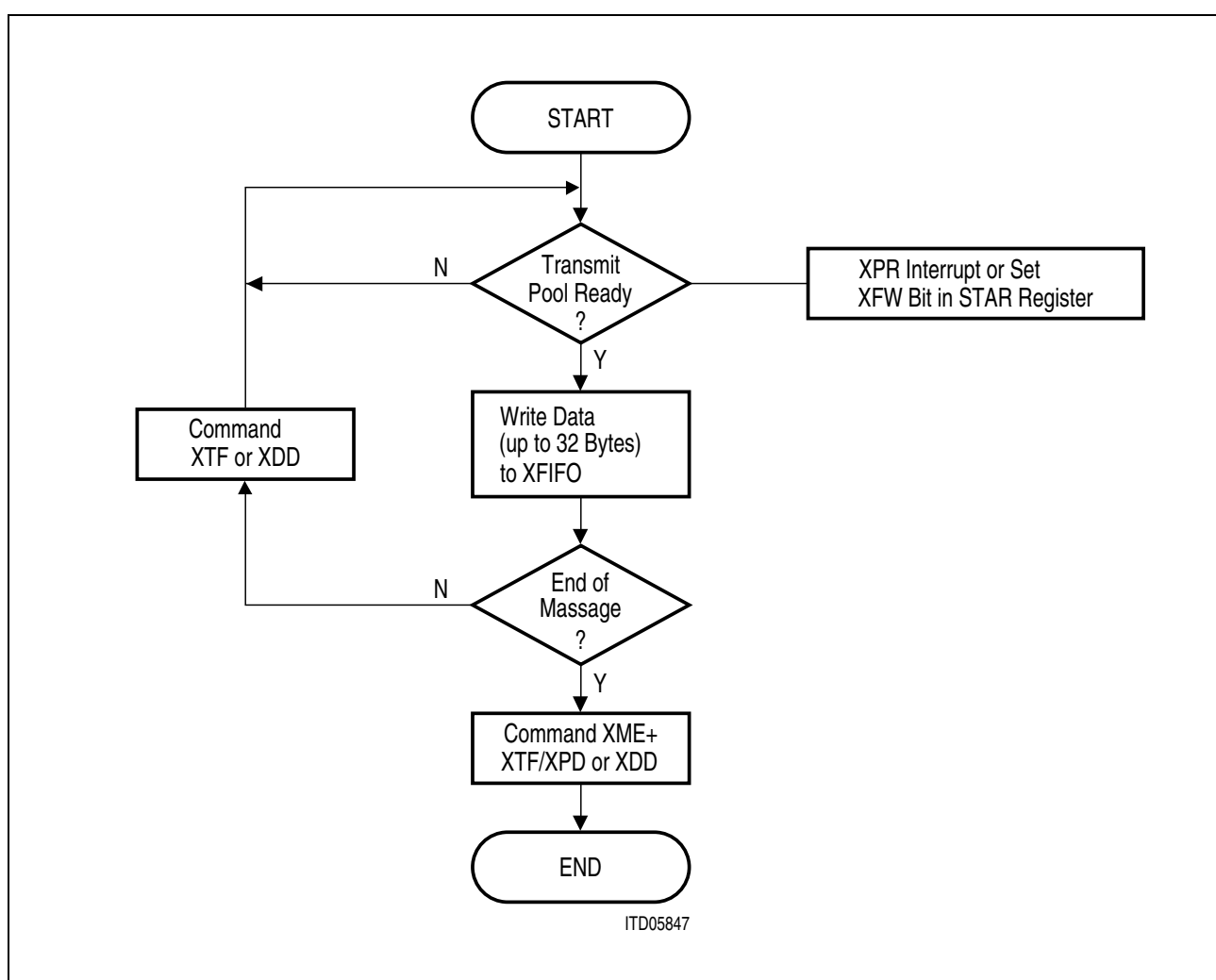


Operational Description

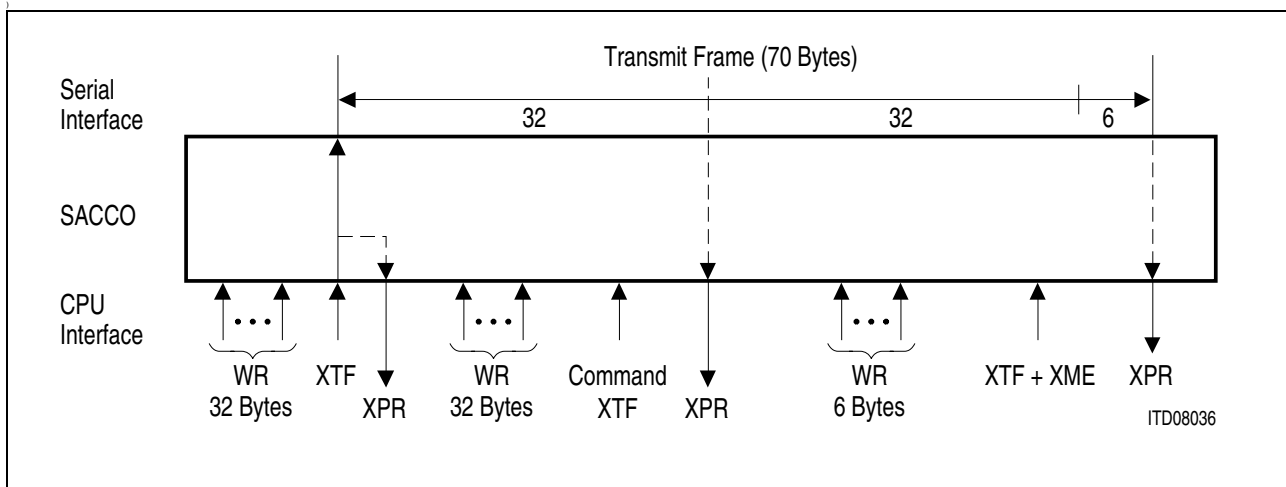
need not be set. If CMDR:XME is not set, the SACCO will repeatedly request for the next data block by means of a XPR-interrupt as soon as the CPU accessible part of the XFIFO is available. This process will be repeated until the CPU indicates the end of message per command, after which frame transmission is ended by appending the CRC and closing flag sequence.

If no more data is available in the XFIFO prior to the arrival of XME, the transmission of the frame is terminated with an abort sequence and the CPU is notified per interrupt (EXIR:XDU). The frame may also be aborted per software (CMDR:XRES).

**Figure 3-4** outlines the data transmission sequence from the CPU's point of view:



**Figure 3-4** Interrupt Driven Transmission Sequence (Flow Diagram)



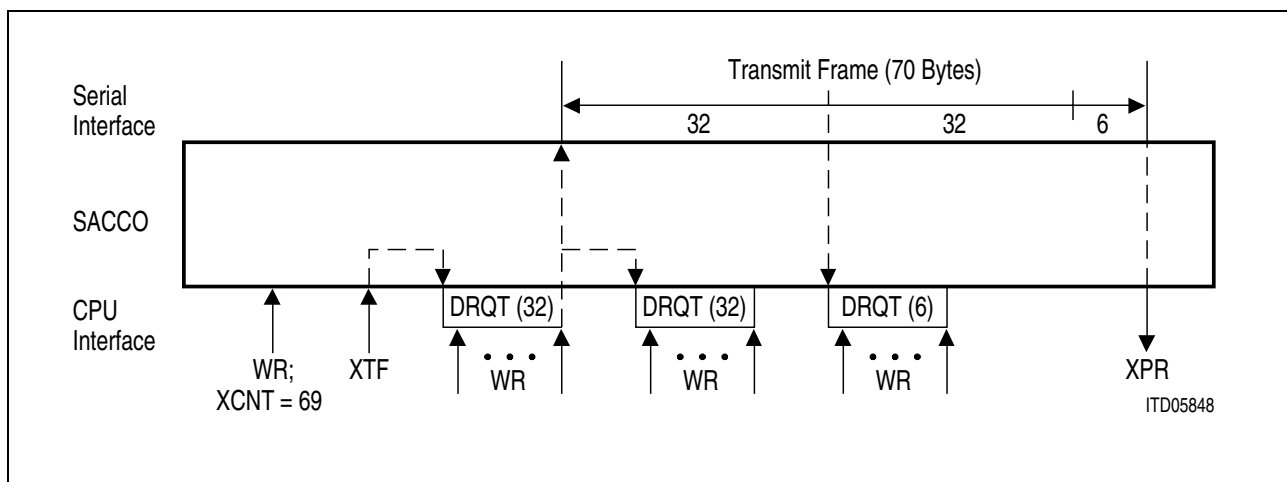
**Figure 3-5 Interrupt Driven Transmission Sequence Example**

### 3.1.4.2 Data Transmission in DMA-Mode

Prior to data transmission, the length of the frame to be transmitted must be programmed via the Transmit Byte Count Registers (XBCH, XBCL). The resulting byte count equals the programmed value plus one byte. Since 12 bits are provided via XBCH, XBCL (XBC11..XBC0) a frame length between 1 and 4096 bytes can be selected.

Having written the Transmit Byte Counter Registers, data transmission can be initiated by command XTF/XPD or XDD. The SACCO will then autonomously request the correct amount of write bus cycles by activating the DRQT-line. Depending on the programmed frame length, block data transfers of  $n \times 32$ -bytes + remainder are requested every time the 32 byte transmit pool is accessible to the DMA-controller.

The following figure gives an example of a DMA driven transmission sequence with a frame length of 70 bytes, i.e. programmed transmit byte count (XCNT) equal 69 bytes.



**Figure 3-6 DMA Driven Transmission Example**

### 3.1.4.3 Data Reception in Interrupt Mode

In receive direction  $2 \times 32$ -byte FIFO-buffers (receive pools) are also provided for each channel. There are two different interrupt indications concerned with the reception of data:

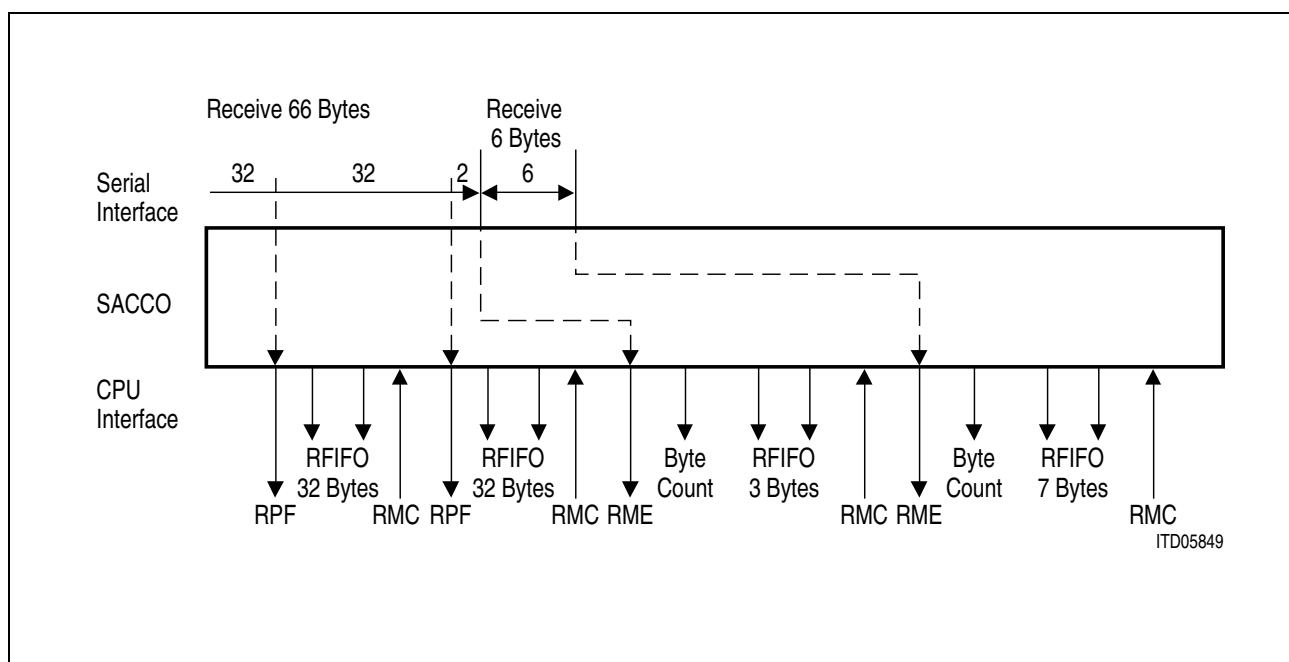
- A RPF (Receive Pool Full) interrupt indicates that a 32-byte block of data can be read from the RFIFO with the received message not yet complete.
- A RME (Receive Message End) interrupt indicates that the reception of one message is completed, i.e. either
  - one message with less than 32 bytes, or the
  - last part of a message with more than 32 bytes is stored in the CPU accessible part of the RFIFO.

The CPU must handle the RPF-interrupt before additional 32 bytes are received via the serial interface, as failure to do so causes a RDO (Receive Data Overflow).

Status information about the received frame is appended to the frame in the RFIFO. This status information follows the format of the RSTA-register, unless using the SACCO-A in clock mode 3. The CPU can read the length of the received message (including the appended Receive Status byte) from the RBCH- and RBCL-registers.

After the received data has been read from the RFIFO, this must be explicitly acknowledged by the CPU issuing a RMC- (Receive Message Complete) command!

The following figure gives an example of an interrupt controlled reception sequence, supposing that a long frame (66 bytes) followed by a short frame (6 bytes) are received.



**Figure 3-7 Interrupt Driven Reception Example**

### 3.1.4.4 Data Reception in DMA-Mode

If the RFIFO contains 32 bytes, the SACCO autonomously requests a block DMA-transfer by activating the DRQR-line. This forces the DMA-controller to continuously perform bus cycles until 32 bytes are transferred from the SACCO to the system memory.

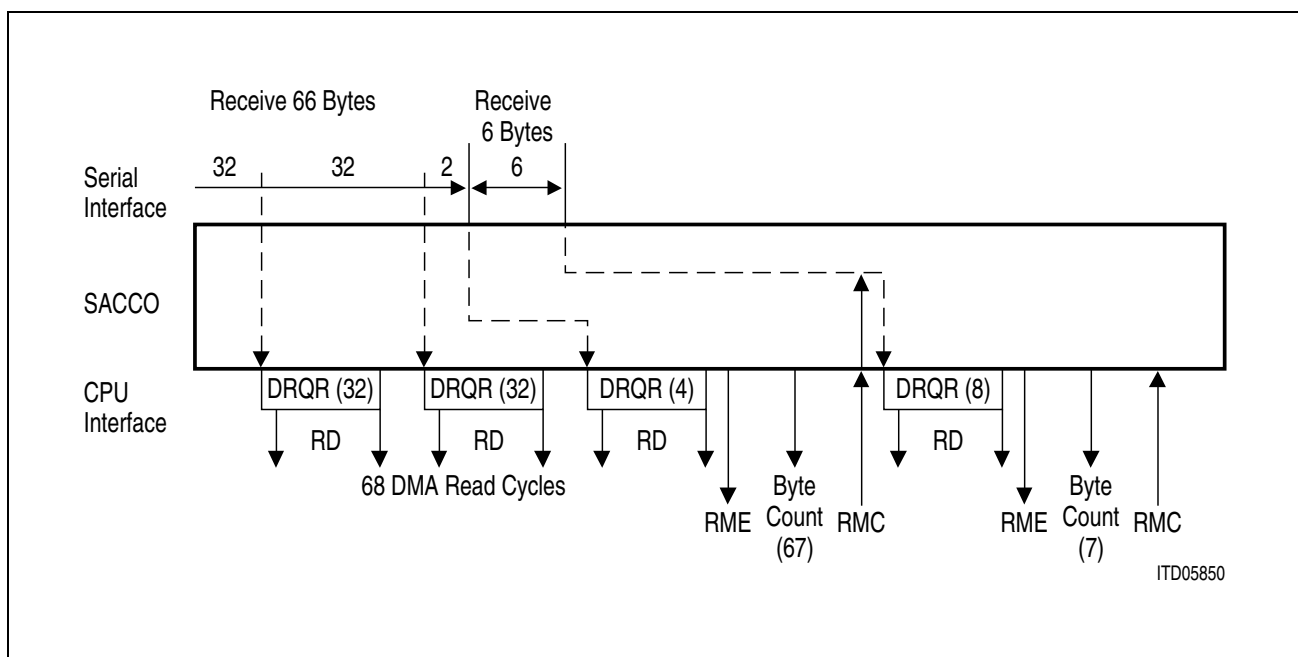
If the RFIFO contains less than 32 bytes (one short frame or the last part of a long frame) the SACCO requests a block data transfer depending on the contents of the RFIFO according to the following table:

**Table 3-1**

RFIFO Contents (in bytes)	DMA Request (in bytes)
1, 2, 3	4
4, 5, 6, 7	8
8-15	16
16-32	32

After the DMA-controller has been set up for the reception of the next frame, the CPU must issue a RMC-command to acknowledge the completion of the receive frame processing. Prior to the reception of this RMC, the SACCO will not initiate further DMA-cycles by activating the DRQR-line.

The following figure gives an example of a DMA controlled reception sequence supposing that a long frame (66 bytes) followed by a short frame (6 byte) are received.



**Figure 3-8 DMA-Driven Reception Example**

### 3.1.5 D-Channel Arbiter

The D-channel arbiter links the SACCO-A to the CFI of the EPIC-1. EPIC-1 and SACCO-A should therefore be initialized before setting up the D-channel arbiter, as demonstrated in **chapter 3.1.6**.

In downstream direction, the D-channel arbiter distributes data from the SACCO-A to the selected subscribers. In upstream direction, the D-channel arbiter ensures that the SACCO-A receives data from only a single correspondent at a time. Given proper initialization, the operation of the D-channel arbiter is largely transparent. The user of the ELIC can thus concentrate on operating the SACCO-A as described in **chapters 2.1.2.5 and 3.1.4**.

For the D-channel arbiter to operate as desired, the SACCO-A must be set clock mode 3 and inter frame timefill set to all '1's. It is also recommended that the SACCO-A **not** be set into auto-mode when communicating with downstream subscribers. The EPIC-1's CFI should be configured to follow the line card IOM-2 protocol, i.e.:

- CFI mode 0
- 2-Mbit/s data rate (usually with a double rate clock)
- 256 bits per frame and port (8 subscribers per port)
- 16-kbit/s D-channels positioned as bits 7,6 of time-slots ( $n \times 4$ ) – 1 for  $n = 1..8$

#### 3.1.5.1 SACCO-A Transmission

Sending data from the SACCO-A to downstream subscribers is handled by the transmit channel master of the D-channel arbiter. The downstream Control Memory (CM) Code for subscribers who may be sent data by the SACCO-A must be set to '1010'<sub>B</sub> for the even time-slot and to '1011'<sub>B</sub> for the odd time-slot. The CM-data of the even time-slot should be programmed to "11 C/I-code 11". For example, a CM-data entry of '11000011' would set the C/I-code to '0000'. Refer to **Figure 3-3**.

If data is to be sent to a single subscriber (no broadcasting), this subscriber must be selected in the XDC-register. Whenever the subscribers D-channel is to be output at the ELIC's CFI, the transmit channel master provides a 2-bit transmit strobe to the SACCO-A. Every frame, 2 data bits are thus strobed from the SACCO-A into the subscriber's D-channel, when the SACCO-A has been commanded to send data. As the subscribers D-channel recurs every 125  $\mu$ s, the data is transmitted from the SACCO-A to the subscriber at a rate of 16 kbit/s. If the SACCO-A has no data to send, it sends its inter frame timefill ('1's) to the subscriber when strobed by the transmit channel master.

With the XDC.BCT bit set (broadcasting), the BCG-registers are used to select the subscribers to whom the SACCO's data is to be sent. The SACCO's output is first copied to an internal buffer. From this buffer, the data is strobed, 2 bits at a time, to all selected subscribers. When the SACCO-A has no data to send, its inter frame timefill ('1's) is copied to the buffer and strobed into the D-channels of the selected subscribers.

### 3.1.5.2 SACCO-A Reception

Subscribers who are to participate in the D-channel arbitration for the SACCO-A must send “all 1s” as inter frame timefill of their D-channels. Flags or idle codes other than “all 1s” are not permitted as inter frame timefill. For any participating subscriber, the “blocked” code must be programmed into the downstream Control Memory (CM). Also, the subscriber’s D-channel must be enabled in the DCE-register.

In the full selection state, the D-channel arbiter overwrites the downstream “blocked” code of enabled subscribers with the “available” code. On the upstream CFI-input lines, the D-channel arbiter monitors all D-channels enabled in the DCE-registers.

When the D-channel arbiter detects a ‘0’ on any monitored D-channel it assumes this to be the start of an opening flag. It therefore strobes the D-channel data of this subscriber to the SACCO-A and starts the Suspend Counter. For this selected subscriber, the D-channel arbiter continues to overwrite the downstream “blocked” code with the “available” code. However, all other enabled subscribers are now passed the “blocked” code from the downstream CM.

If the SACCO-A does indeed receive an HDLC-frame – complete or aborted – from the selected subscriber, the Suspend Counter is reset. While the SACCO-A receives data from the selected subscriber, the “blocked” code stops all other subscribers from sending data to the SACCO-A. After the SACCO-A has received a closing flag or abort sequence for the subscribers frame, the D-channel arbiter stops strobing the subscriber’s data to the SACCO-A and enters the limited selection state.

If, after the initial ‘0’, the SACCO-A does not receive an HDLC-frame – complete or aborted – from the selected subscriber, it does not reset the Suspend Counter. Eventually, the Suspend Counter under flows, setting off the ISTA.IDA-interrupt. The subscriber who sent the erroneous ‘0’ can then be identified in the ASTATE-register. Any subscriber who frequently sends erroneous ‘0’s should be disabled from the DCE, and the cause of the error investigated. After the ISTA.IDA-interrupt, the SACCO-A receiver must be reset to resume operation in the full selection state.

The limited selection state is identical to the full selection state, except that the subscriber who last sent data to the SACCO-A is excluded from the arbitration. This prevents any single subscriber from constantly keeping the SACCO-A busy. The “blocked” code of the CM is passed to the excluded subscriber, while the D-channel arbiter sends all other enabled subscribers the “available” code. All enabled subscribers – except the one excluded – are monitored for the starting ‘0’ of an opening flag. How long the exclusion lasts can be programmed in the AMO-register. If none of the monitored subscribers has started sending data during this time, the D-channel Arbiter re-enters the full selection state.

### 3.1.6 Initialization Procedure

For proper initialization of the DOC the following procedure is recommended:

#### 3.1.6.1 Hardware Reset

Refer to **chapter 2.1.2.2** “Reset Logic”.

#### 3.1.6.2 EPIC<sup>®</sup>-1 Initialization

##### 3.1.6.2.1 EPIC<sup>®</sup> Registers Initialization

The PCM- and CFI-configuration registers (PMOD, PBNR, .., CMD1, CMD2, ..) should be programmed to the values required for the application. The correct setting of the PCM- and CFI-registers is important in order to obtain a reference clock (RCL) which is consistent with the externally applied clock signals.

The state of the operation mode (OMDR:OMS1...0 bits) does not matter for this programming step.

- PMOD = PCM-mode, timing characteristics, etc.
- PBNR = Number of bits per PCM-frame
- POFD = PCM-offset downstream
- POFU = PCM-offset upstream
- PCSR = PCM-timing
- CMD1 = CFI-mode, timing characteristics, etc.
- CMD2 = CFI-timing
- CBNR = Number of bits per CFI-frame
- CTAR = CFI-offset (time-slots)
- CBSR = CFI-offset (bits)
- CCSR = CFI-sub channel positions

##### 3.1.6.2.2 Control Memory Reset

Since the hardware reset does not affect the EPIC-1 memories (Control and Data Memories), it is mandatory to perform a “software reset” of the CM. The CM-code ‘0000’ (unassigned channel) should be written to each location of the CM. The data written to the CM-data field is then don’t care, e.g. FF<sub>H</sub>.

OMDR:OMS1...0 must be to ‘00’<sub>B</sub> for this procedure (reset value).

- MADR = FF<sub>H</sub>
- MACR = 70<sub>H</sub>
- Wait for EPIC.STAR:MAC = 0

The resetting of the complete CM takes 256 RCL-clock cycles. During this time, the EPIC.STAR:MAC-bit is set to logical 1.

### 3.1.6.2.3 Initialization of Pre-processed Channels

After the CM-reset, all CFI time-slots are unassigned. If the CFI is used as a plain PCM-interface, i.e. containing only switched channels (B-channels), the initialization steps below are not required. The initialization of pre-processed channels applies only to IOM- or SLD-applications.

An IOM- or SLD- “channel” consists of four consecutive time-slots. The first two time-slots, the B-channels need not be initialized since they are already set to unassigned channels by the CM-reset command. Later, in the application phase of the software, the B-channels can be dynamically switched according to system requirements. The last two time-slots of such an IOM- or SLD-channel, the pre-processed channels must be initialized for the desired functionality. There are five options that can be selected:

**Table 3-2 Pre-processed Channel Options at the CFI**

Even CFI Time-Slot	Odd CFI Time-Slot	Main Application
Monitor/feature control channel	4-bit C/I-channel, D-channel handled by SACCO-A and D-ch. arbiter	IOM-1 or IOM-2 digital subscriber
Monitor/feature control channel	4-bit C/I-channel, D-channel not switched (decentral D-ch. handling)	IOM-1 or IOM-2 digital subscriber
Monitor/feature control channel	4-bit C/I-channel, D-channel switched (central D-ch. handling)	IOM-1 or IOM-2 digital subscriber
Monitor/feature control channel	6-bit SIG-channel	IOM-2, analog subscriber
Monitor/feature control channel	8-bit SIG/channel	SLD, analog subscriber

Also refer to **Figure 3-4**.

#### Example

In CFI-mode 0 all four CFI-ports shall be initialized as IOM-2 ports with a 4-bit C/I-field and D-channel handling by the SACCO-A.

CFI time-slots 0, 1, 4, 5, 8, 9..28, 29 of each port are B-channels and need not to be initialized.



CFI time-slots 2, 3, 6, 7, 10, 11..30, 31 of each port are pre-processed channels and need to be initialized:

**CFI-port 0, time-slot 2 (even), downstream**

MADR = FF<sub>H</sub> ; the C/I-value '1111' will be transmitted upon CFI-activation  
 MAAR = 08<sub>H</sub> ; addresses ts 2 down  
 MACR = 7A<sub>H</sub> ; CM-code '1010'  
 Wait for STAR:MAC = 0

**CFI-port 0, time-slot 3 (odd), downstream**

MADR = FF<sub>H</sub> ; don't care  
 MAAR = 09<sub>H</sub> ; addresses ts 3 down  
 MACR = 7B<sub>H</sub> ; CM-code '1011'  
 Wait for STAR:MAC = 0

**CFI-port 0, time-slot 2 (even), upstream**

MADR = FF<sub>H</sub> ; the C/I-value '1111' is expected upon CFI-activation  
 MAAR = 88<sub>H</sub> ; address ts 2 up  
 MACR = 78<sub>H</sub> ; CM-code '1000'  
 Wait for STAR:MAC = 0

**CFI-port 0, time-slot 3 (odd), upstream**

MADR = FF<sub>H</sub> ; don't care  
 MAAR = 89<sub>H</sub> ; address ts 3 up  
 MACR = 70<sub>H</sub> ; CM-code '0000'  
 Wait for STAR:MAC = 0

Repeat the above programming steps for the remaining CFI-ports and time-slots.

This procedure can be speeded up by selecting the CM-initialization mode (OMDR:OMS1...0 = 10). If this selection is made, the access time to a single memory location is reduced to 2.5 RCL-cycles. The complete initialization time for 32 IOM-2 channels is then reduced to  $128 \times 0.61 \mu\text{s} = 78 \mu\text{s}$ .

**3.1.6.2.4 Initialization of the Upstream Data Memory (DM) Tri-State Field**

For each PCM time-slot the tri-state field defines whether the contents of the DM-data field are to be transmitted (low impedance), or whether the PCM time-slot shall be set to high impedance. The contents of the tri-state field is not modified by a hardware reset. In order to have all PCM time-slots set to high impedance upon the activation of the PCM- interface, each location of the tri-state field must be loaded with the value '0000'. For this purpose, the "tri-state reset" command can be used:

**Operational Description**

OMDR = C0<sub>H</sub> ; OMS1...0 = 11, normal mode  
MADR = 00<sub>H</sub> ; code field value '0000'<sub>B</sub>  
MACR = 68<sub>H</sub> ; MOC-code to initialize all tri-state locations (1101<sub>B</sub>)  
Wait for STAR:MAC = 0

The initialization of the complete tri-state field takes 1035 RCL-cycles.

*Note: 1) It is also possible to program the value '1111' to the tri-state field in order to have all time-slots switched to low impedance upon the activation of the PCM-interface.*

*2) While OMDR:PSB = 0, all PCM-output drivers are set to high impedance, regardless of the values written to the tri-state field.*

**3.1.6.3 SACCO-Initialization**

To initialize the SACCO, the CPU has to write a minimum set of registers. Depending on the operating mode and on the features required, an optional set of register must also be initialized.

As the first register to be initialized, the MODE-register defines operating and address mode. If data reception shall be performed, the receiver must be activated by setting the RAC-bit. Depending on the mode selected, the following registers must also be defined:

**Table 3-3 Mode Dependent Register Set-up**

	<b>1 Byte Address</b>	<b>2 Byte Address</b>
Transparent mode 1		RAH1 RAH2
Non-auto mode	RAH1 = 00 <sub>H</sub> RAH2 = 00 <sub>H</sub> RAL1 RAL2	RAH1 RAH2 RAL1 RAL2
Auto-mode	XAD1 XAD2 RAH1 = 00 <sub>H</sub> RAH2 RAL1 RAL2	XAD1 XAD2 RAH1 RAH2 RAL1 RAL2

The second minimum register to be initialized is the CCR2. In combination with the CCR1, the CCR2 defines the configuration of the serial port. It also allows enabling the RFS-interrupt.

If bus configuration is selected, the external serial bus must be connected to the C×D-pin for collision detection. In point-to-point configuration, the C×D-pin must be tied to ground if no “clear to send” function is provided via a modem.

Depending on the features desired, the following registers may also require initializing before powering up the SACCO:

**Table 3-4 Feature Dependent Register Set-up**

Feature	Register(s)
Clock mode 2	TSAR, TSAC, XCCR, RCCR
Masking selected interrupts	MASK
DMA controlled data transfer	XBCH
Check on receive length	RLCR

The CCR1 is the final minimum register that has to be programmed to initialize the SACCO. In addition to defining the serial port configuration, the CCR1 sets the clock mode and allows the CPU to power-up or power-down the SACCO.

In power-down mode all internal clocks are disabled, and no interrupts are forwarded to the CPU. This state can be used as standby mode for reduced power consumption. Switching between power-up or power-down mode has no effect on the contents of the register, i.e. the internal state remains stored.

After power-up of the SACCO, the CPU should bring the transmitter and receiver to a defined state by issuing a XRES (transmitter reset) and RHR (receiver reset) command via the CMDR-register. The SACCO will then be ready to transmit and receive data.

The CPU controls the data transfer phase mainly by commands to the SACCO via the CMDR-register, and by interrupt indications from the SACCO to the CPU. Status information that does not trigger an interrupt is constantly available in the STAR-register.

### 3.1.6.4 Initialization of D-Channel Arbiter

The D-channel arbiter links the SACCO-A to the CFI of the EPIC-1 part of the ELIC. Thus the EPIC-1 and SACCO-parts of the ELIC should be initialized before initializing the D-channel arbiter.

For subscribers wishing to communicate with the SACCO-A, the correct pre-processed channel code must have been programmed in the EPIC-1's control memory. In downstream direction, this code is CMC = 1010 for the even time-slot and CMC = 1011 for the odd time-slot. In upstream direction, any pre-processed channel code is also valid for arbiter operation. This is shown in **Figure 3-3** of **chapter 3.1.3.3**. For an example refer to **chapter 3.1.6.2.3**.

If the MR-bit is used to block downstream subscribers, the blocking code  $MR = '0'_B$  can be written as  $MADR = '11xxxx01'_B$  when initializing the even downstream time-slot. The 'x' stand for the C/I-code. This also is shown in **Figure 3-3**.

If the C/I-code is used to block downstream subscribers, such subscribers must be activated with the C/I-code  $'1100'_B$ , not  $'1000'_B$ .

The SACCO-A must be initialized to clock mode 3 to communicate with downstream subscribers. In clock mode 3, the SACCO-A receives its input and transmit its output via the D-channel arbiter. If the CCR2.T×DE-bit is set, the SACCO-A's output is transmitted at the T×DA-pin in addition to being transmitted via the D-channel arbiter.

Once EPIC-1 and SACCO-A have been correctly initialized, writing the subscriber's address into the XDC-register allows the SACCO-A to send the subscriber data. By setting the XDC.BCT-bit and programming the BCG-registers, the SACCO-A can transmit its data to several subscribers.

To strobe upstream data from the CFI-interface to the SACCO-A's receiver, the AMO-register must be programmed for the desired functionality. Subscribers who are to be allowed to send data must be enabled via the DCE-registers. If a subscriber tries to send data during the initialization of the upstream D-channel arbiter, a ISTA.IDA-interrupt may occur. This interrupt can be cleared by resetting the SACCO-A receiver.

*Note: 1) The EPIC-1 and SACCO-A must be initialized correctly before the D-channel arbiter can operate properly. Particular care must be given to programming the EPIC-1's Control Memory (CM) with the required CM-Codes (CMCs).*

*2) The upstream and downstream D-channel arbiter initializations are independent of each other.*

### 3.1.6.5 Activation of the PCM- and CFI-Interfaces

With EPIC-1, SACCO-A and D-channel arbiter all configured to the system requirements, the PCM- and CFI-interface can be switched to the operational mode.

The OMDR:OMS1...0 bits must be set (if this has not already be done) to the normal operation mode (OMS1...0 = 11). When doing this, the PCM-framing interrupt (ISTA:PFI) will be enabled. If the applied clock and framing signals are in accordance with the values programmed to the PCM-registers, the PFI-interrupt will be generated (if not masked). When reading the status register, the STAR:PSS-bit will be set to logical 1.

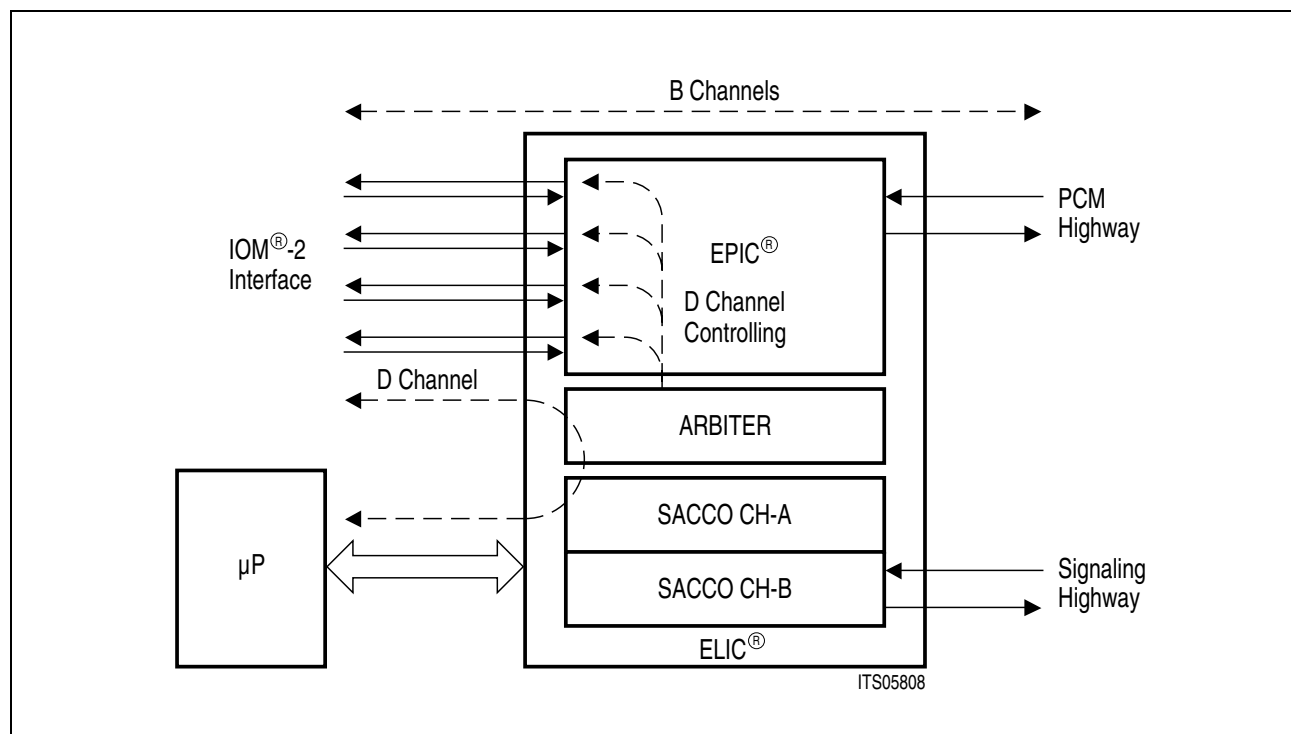
To enable the PCM-output drivers set OMDR:PSB = 1. The CFI-interface is activated by programming OMDR:CSB = 1. This enables the output clock and framing signals (DCL and FSC), if these have been programmed as outputs. It also enables the CFI-output drivers. The output driver type can be selected between "open drain" and "tri-state" with the OMDR:COS-bit.

**Example:** Activation of the EPIC-1 part of the ELIC for a typical IOM-2 application:

OMDR = EE<sub>H</sub>; Normal operation mode (OMS1...0 = 11)  
 PCM-interface active (PSB = 1)  
 PCM-test loop disabled (PTL = 0)  
 CFI-output drivers: open drain (COS = 1)  
 Monitor handshake protocol selected (MFPS = 1)  
 CFI active (CSB = 1)  
 Access to EPIC-1 registers via address pins A4...A0 (RBS = 0)

### 3.1.6.6 Initialization Example

In this sample initialization the ELIC is set up to handle a digital IOM-2 subscriber. The interfaces of the DOC/ELIC are shown below:



**Figure 3-9 DOC/ELIC<sup>®</sup> Interfaces for Initialization Example**

The subscriber uses the ELIC's CFI-port 0, channel 0 (time-slots 0-3). The subscriber's upstream B<sub>1</sub>-channel is to be switched to PCM-port 0, time-slot 5. The subscriber's upstream B<sub>2</sub>-channel is to be looped back to the subscriber on the downstream B<sub>1</sub>-channel. The subscriber's downstream B<sub>2</sub>-channel is to be switched from PCM-port 0, time-slot 1. The subscriber's HDLC-data is exchanged via the D-channel with the SACCO-A. Monitor and C/I-channels are to be handled via the ELIC.

The SACCO-B communicates via a dedicated signaling highway with a non-PBC group controller. A 4-MHz clock is input as PDC and HDCB.

Port 1 of the ELIC is to be used as active low output. Thus the port should be linked to pull-up resistors.

Write PCON1 = FF<sub>H</sub>

Write PORT1 = FF<sub>H</sub>

### 3.1.6.6.1 EPIC<sup>®</sup>-1 Initialization Example

Configure PCM-side of ELIC:

- Write PMOD = 44<sub>H</sub> PCM-mode 1, single clock rate, PFS evaluated with falling edge of PDC, R×D0 = logical input port 0
- Write PBNR = FF<sub>H</sub> 512 bits per PCM-frame
- Write POFD = F0<sub>H</sub> the internal PFS marks downstream bit 6, ts 0 (second bit of frame)
- Write POFU = 18<sub>H</sub> the internal PFS marks upstream bit 6, ts 0 (second bit of frame)
- Write PCSR = 45<sub>H</sub> no clock shift; PCM-data sampled with falling, transmitted with rising PDC

Configure CFI-side of ELIC:

- Write CMD1 = 20<sub>H</sub> PDC and PFS used as clock and framing source for the CFI; CRCL = PDC; CFI-mode 0
- Write CMD2 = D0<sub>H</sub> FSC shaped for IOM-2 interface; DCL = 2 × data rate; CFI-data received with falling, transmitted with rising CRCL
- Write CBNR = FF<sub>H</sub> 256 bits per CFI-frame
- Write CTAR = 02<sub>H</sub> PFS is to mark CFI time-slot 0
- Write CBSR = 20<sub>H</sub> PFS is to mark bit 7 of CFI time-slot 0; no shift of CFI-upstream data relative to CFI-downstream data
- Write CSCR = 00<sub>H</sub> 2-bit channels located in position 7, 6 on all CFI-ports

Reset EPIC-1 Control Memory (CM) to FF<sub>H</sub>:

- Write MADR = FF<sub>H</sub>
- Write MACR = 70<sub>H</sub>

Initialize EPIC-1 CM:

- Write OMDR = 80<sub>H</sub> set EPIC-1 from CM-reset mode into CM-initialization mode

The subscriber's upstream B<sub>1</sub>-channel is switched to PCM-port 0, time-slot 5

- Write MADR = 89<sub>H</sub> connection to PCM-port 0, time-slot 5
- Write MAAR = 80<sub>H</sub> from upstream CFI-port 0, time-slot 0
- Write MACR = 71<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'<sub>B</sub> (code for a simple 64-kbit/s connection)

- Read STAR Wait for STAR:MAC = 0

The subscriber's upstream B<sub>2</sub>-channel is internally looped via PCM-port 1, time-slot 1

- Write MADR = 85<sub>H</sub> loop to PCM-port 1, time-slot 1
- Write MAAR = 81<sub>H</sub> from upstream CFI-port 0, time-slot 1
- Write MACR = 71<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'<sub>B</sub> (code for a simple 64 kbit/s connection)

- Read STAR Wait for STAR:MAC = 0

**Operational Description**

The subscriber's upstream time-slots 2 and 3 are initialized as monitor and C/I-channels with decentral D-channel handling

- Write MADR = FF<sub>H</sub> received C/I-code to be compared to '1111'<sub>B</sub>
- Write MAAR = 88<sub>H</sub> from upstream CFI-port 0, time-slot 2
- Write MACR = 78<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '1000'<sub>B</sub> (even address code for decentral monitor and C/I-channels)
- Read STAR Wait for STAR:MAC = 0
- Write MAAR = 89<sub>H</sub> from upstream CFI-port 0, time-slot 3
- Write MACR = 70<sub>H</sub> write CM-code addressed by MAAR with '0000'<sub>B</sub> (odd address code for decentral monitor and C/I-channels)
- Read STAR Wait for STAR:MAC = 0

The subscriber's downstream B<sub>1</sub>-channel is internally looped via PCM-port 1, time-slot 1

- Write MADR = 85<sub>H</sub> internal loop from PCM-port 1, time-slot 1
- Write MAAR = 00<sub>H</sub> to downstream CFI-port 0, time-slot 0
- Write MACR = 71<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'<sub>B</sub> (code for a simple 64-kbit/s connection)
- Read STAR Wait for STAR:MAC = 0

The subscriber's downstream B<sub>2</sub>-channel is switched from PCM-port 0, time-slot 1

- Write MADR = 01<sub>H</sub> connection from PCM-port 0, time-slot 1
- Write MAAR = 01<sub>H</sub> to downstream CFI-port 0, time-slot 1
- Write MACR = 71<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '0001'<sub>B</sub> (code for a simple 64-kbit/s connection)
- Read STAR Wait for STAR:MAC = 0

The subscriber's downstream time-slots 2 and 3 are initialized as monitor and C/I-channels with D-channel handling by the SACCO-A

- Write MADR = FF<sub>H</sub> C/I-code to be transmitted = '1111'<sub>B</sub>  
(MADR = F3<sub>H</sub> D-channel blocking code '1100'<sub>B</sub> to be transmitted.)
- Write MAAR = 08<sub>H</sub> to downstream CFI-port 0, time-slot 2
- Write MACR = 7A<sub>H</sub> write CM-data addressed by MAAR with content of MADR; write CM-code addressed by MAAR with '1010'<sub>B</sub> (even address code for monitor and C/I-channels with D-channel handling by SACCO-A)
- Read STAR Wait for STAR:MAC = 0
- Write MAAR = 09<sub>H</sub> from upstream CFI-port 0, time-slot 3
- Write MACR = 7B<sub>H</sub> write CM-code addressed by MAAR with '1011'<sub>B</sub> (odd address code for monitor and C/I-channels with D-channel handling by SACCO-A)
- Read STAR Wait for STAR:MAC = 0



Set EPIC-1 to normal mode

Write OMDR = C0<sub>H</sub> set EPIC-1 to CM-normal mode; Interrupt line will go active

Read ISTA = 20<sub>H</sub> EPIC-1 interrupt

Read ISTA\_E = 08<sub>H</sub> PFI-interrupt: PCM-synchronisity status has changed

Read STAR\_E = 25<sub>H</sub> ELIC is synchronized to PCM-interface; MFIFO ready

Reset tri-state field of Data Memory (DM)

Write MADR = 00<sub>H</sub> all bits of time-slot set to high impedance

Write MACR = 68<sub>H</sub> write MADR to all locations of PCM-tri-state field

Read STAR Wait for STAR:MAC = 0

### 3.1.6.6.2 SACCO-A Initialization Example

Configure the SACCO-A for communication with downstream subscribers

Write MODE = A8<sub>H</sub> set SACCO-B to transparent mode 1; switch receiver active

Write RAH1 = 00<sub>H</sub> response SAPI1: Signaling data

Write RAH2 = 40<sub>H</sub> response SAPI 2: Packet-switched data

(Write CCR2 = 00<sub>H</sub>) reset value: T×DA pin disabled; standard data sampling;  
RFS-interrupt disabled

Write CCR1 = 87<sub>H</sub> power-up SACCO-A in point to point configuration and clock  
mode 3 with double rate clock; inter frame timefill = all '1's

Reset the SACCO-A's FIFOs

Write CMDR = C1<sub>H</sub> reset CPU accessible and CPU inaccessible part of RFIFO,  
and reset XFIFO; the interrupt line will go active

Read ISTA = 02<sub>H</sub> interrupt of SACCO-A

Read ISTA\_A = 10<sub>H</sub> transmit pool ready

### 3.1.6.6.3 D-Channel Arbiter Initialization Example

Enable D-channel transmission to CFI-port 0, channel 0

(Write XDC = 00<sub>H</sub>) reset value: broadcasting disabled; transmit to channel 0  
of port 0

Enable D-channel reception on CFI-port 0, channel 0

Write AMO = F9<sub>H</sub> start with maximum selection delay; suspend counter active;  
control of D-channel to take place via C/I-bit; control  
channel master enabled

Write DCE0 = 01<sub>H</sub> enable CFI-port 0, channel 0 for data reception



### 3.1.6.6.4 PCM- and CFI-Interface Activation Example

Write OMDR = EE<sub>H</sub> see chapter 3.1.6.5.  
 Enable upstream PCM-port 0, time-slot 5  
 Write MADR = 0F<sub>H</sub> set all bits of time-slot to low impedance  
 Write MAAR = 89<sub>H</sub> PCM-port 0, time-slot 5  
 Write MACR = 60<sub>H</sub> write only single tri-state control position  
 Read STAR Wait for STAR:MAC = 0

### 3.1.6.6.5 SACCO-B Initialization Example

Configure the SACCO-B as secondary station for an upstream (non-PBC) group controller

Write MODE = 48<sub>H</sub> set SACCO-B to 8-bit non-auto mode; switch receiver active  
 Write RAH1 = 00<sub>H</sub> the high-byte comparison registers should be set to 00<sub>H</sub> when using non-auto mode

Write RAH2 = 00<sub>H</sub>  
 Write RAL1 = 89<sub>H</sub> 8-bit address of SACCO-B  
 Write RAL2 = FF<sub>H</sub> 8-bit group address (broadcast by group controller)  
 Write CCR2 = 08<sub>H</sub> TxDB pin enabled; standard data sampling; RFS-interrupt disabled  
 Write CCR1 = 98<sub>H</sub> power-up SACCO-B in point-to-point configuration and clock mode 0 with single rate clock; inter frame timefill = flags; T×DB is push-pull output

Reset the SACCO-B's FIFOs

Write CMDR = C1<sub>H</sub> reset CPU accessible and CPU inaccessible part of RFIFO, and reset XFIFO; the interrupt line will go active  
 Read ISTA = 08<sub>H</sub> interrupt of SACCO-B  
 Read ISTA\_B = 10<sub>H</sub> transmit pool ready

### 3.2 SIDEC

The SIDEC is a 4-channel signaling controller containing slightly SACCO modules and a control logic for DRDY handling (Stop/Go signal from QUAT-S).

To initialize one of the SIDEC's, the CPU has to write a minimum set of registers. Depending on the operating mode and on the features required, an optional set of registers must also be initialized.

As the first register to be initialized, the MODE-register defines operating and address mode. If data reception shall be performed, the receiver must be activated by setting the RAC-bit. Depending on the mode selected, the following registers must also be defined:

**Table 3-5 Mode Dependent Register Set-up**

	1 Byte Address	2 Byte Address
Transparent mode 1		RAH1 RAH2
Non-auto mode	RAH1 = 00 <sub>H</sub> RAH2 = 00 <sub>H</sub> RAL1 RAL2	RAH1 RAH2 RAL1 RAL2
Auto-mode	XAD1 XAD2 RAH1 = 00 <sub>H</sub> RAH2 RAL1 RAL2	XAD1 XAD2 RAH1 RAH2 RAL1 RAL2

The second minimum register to be initialized is the CCR2. In combination with the CCR1, the CCR2 defines the configuration of the serial port. It also allows enabling the RFS-interrupt. Another function of CCR2 is to define the operating mode for the D-channel ready input (DRDY).

Depending on the features desired, the following registers may also require initializing before powering up the SIDEC:

**Table 3-6 Feature Dependent Register Set-up**

Feature	Register(s)
Time-slot assignment	TSAR, TSAC, XCCR, RCCR
Masking selected interrupts	MASK
Check on receive length	RLCR

---

## Operational Description

The CCR1 is the final minimum register that has to be programmed to initialize the SIDEC. In addition to defining the serial port configuration, the CCR1 defines the characteristic of all IOM-2 ports allows the CPU to power-up or power-down a SIDEC.

In power-down mode all internal clocks are disabled, and no interrupts are forwarded to the CPU. This state can be used as standby mode for reduced power consumption. Switching between power-up or power-down mode has no effect on the contents of the register, i.e. the internal state remains stored.

After power-up of the SIDEC, the CPU should bring the transmitter and receiver to a defined state by issuing a XRES (transmitter reset) and RHR (receiver reset) command via the CMDR-register. The SIDEC will then be ready to transmit and receive data.

The CPU controls the data transfer phase mainly by commands to the SIDEC via the CMDR-register, and by interrupt indications from the SIDEC to the CPU. Status information that does not trigger an interrupt is constantly available in the STAR-register.

## 4 DSP Core OAK

### 4.1 Introduction

#### 4.1.1 General Description

OAK DSP Core is a 16-bit (data and program) busses high performance fixed-point DSP core. OAK is designed for the mid to high-end telecommunications and consumer electronics applications, where low-power and portability are still major requirements. Among the applications supported by OAK are not only PBX but also digital cellular telephones (like JDC, GSM, USDC), fast modems (like V.fast), advanced fax machines, etc.

OAK is aimed at achieving the best cost-performance factor for a given (small) silicon area. Taking into account ALL elements of "system-on-chip" requirements, like: program size, data memory size, glue logic, power management, etc.

Based on the proven philosophy of its predecessor SPC, OAK is also designed to be used as an engine for DSP-based application specific ICs. It is specified with several levels of modularity, in RAM, ROM, and I/O, allowing efficient DSP-based ASIC development. The core consists of the main blocks of a high performance central processing unit, including a full featured bit-manipulation unit, RAM and ROM addressing units, and Program control logic. All other peripheral blocks, which are application specific, are defined as part of the user specified logic implemented around the OAK core on the same silicon die.

OAK has an improved set of DSP and general microprocessor functions to meet the applications requirements. The OAK programming model and instruction-set is aimed at straightforward generation of efficient and compact code. It has an enhanced instruction set which is upward compatible with the SPC instruction set.

OAK also features a wide range of operating voltage, down to 2.7 V. OAK is available as a core in a standard cell library, to be utilized as a part of the user's custom chip design. OAK is the second member in a family of standard DSP core cells.

#### 4.1.2 Architecture Highlights

The OAK DSP core architecture is based on its predecessor SPC. In the following description, the **OAK new features are bold-faced**.

The OAK core consists of four parallel execution units:

- the Computation Unit (CU)
- the **Bit Manipulation Unit (BMU)**,
- the Data Addressing Arithmetic Unit (DAAU),
- the Program Control Unit (PCU).

It has two blocks of data RAM/ROM for parallel feeding of the two inputs of the multiplier.

The CU has a 16 by 16 bit multiplier (which performs signed by signed, signed by unsigned or unsigned by unsigned multiplications) supporting single and double precision multiplication. The CU has also a 36-bit ALU, and two 36-bit accumulators with access to the two additional accumulators of the BMU.

The BMU consists of a full 36-bit barrel shifter, a bit-field operations (BFO) unit including a special hardware for exponent calculation, and two 36-bit accumulators with access to the two accumulators of the CU. Context switching (swapping) between the two sets of accumulators is supported.

The DAAU, the PCU, the data and program memory organization, and buses structure are basically similar to those of SPC.

Powerful zero-overhead looping, enables four levels of block-repeat (BKREP), in addition to an interruptable single word Repeat. The pipeline structure has been improved to achieve minimal cycle time. An index-based addressing capability was added. Shadow registers for parts of the status registers and alternative bank of registers for four of the DAAU registers were added to improve interrupt handling and subroutine nesting. Option for automatic context switching during interrupts is also included.

The hardware stack of SPC is substituted with a more flexible software stack residing in the data memory space. This improvement, as well as the indexed-based addressing capability and the additional accumulators, will also support a C-compiler implementation for OAK.

An additional maskable interrupt has been added to the core and a NMI interrupt (used in SPC for emulation by the name BPI) was released to the user as a non-maskable interrupt. For emulation support a Breakpoint interrupt (BI) was added, sharing the same interrupt vector of TRAP.

The core is designed to interface with external memories and peripherals having different speeds, using a wait-state mechanism. It supports DMA mode and hold mode operation. It also has support for an automatic boot procedure, and it has support for on-chip emulation module, residing off-core.

## 4.2 Architecture Features

### 4.2.1 Features

- 16-bit fixed-point DSP CORE with high level of modularity:
  - Expandable internal program ROM.
  - Expandable internal data RAM and/or ROM.
  - User-defined registers.
- $16 \times 16$  bit 2's complement parallel multiplier with 32-bit product. Multiplication of signed by signed, signed by unsigned, and unsigned by unsigned.
- Single cycle multiply-accumulate instructions.
- 36-bit ALU.
- 36-bit left/right Barrel Shifter.
- Four 36-bit accumulators.
- Memory organization:
- 64 K word maximum addressable data space, organized in:
  - local and external data memory space.
  - 64 K word maximum program memory space.
  - Data RAMs can also be viewed as a single continuous RAM.
  - User definable data ROM on the same address space of the data RAM.
  - Alternative registers bank for 3 of the DAAU pointers (and 1 configuration register) with individual selectable bank exchanging.
- Software stack (with stack pointer) residing in the data RAM.
- Index-based addressing capability.
- Automatic context switching by interrupts (with enable/disable feature for each interrupt) using Shadow registers for parts of status registers and swapping between two 36-bit accumulators.
- All general and most special purpose registers are arranged as a global register set of 34 registers that can be referenced in most data moves and core operations.
- Bit-Field (up to 16 bits) Operations (BFO): Set, Reset, Change, Test.  
These operations are executed directly on registers and data memory content, with no affect on accumulators' content.
- Single cycle exponent evaluation of up to 36-bit values.
- Enables full normalization operation in 2 cycles.
- Double precision multiplication support.
- Max/Min single cycle instruction with pointer latching and modification.  
Optimized for codebook search and Viterbi decoding.
- Single cycle Division step support.
- Single cycle data move & shift capability.
- Arithmetic and logical shifting capability, according to a shift value stored in a special register, or embedded in the instruction opcode. Conditional shift is also available, as well as rotate left and right operations.

- The product register is transferred to the accumulator without scaling, or after scaling by shifting the product 1 bit to the left, 2 bits to the left or 1 bit to the right.
- Add/Subtract/Compare of a long immediate value with registers or data memory, with no affect on the accumulator content.
- Powerful swapping (14 options) between two sets of accumulators.
- Automatic saturation mode on overflow while reading content of accumulators, or using a special instruction.
- Zero-overhead looping by two interruptable mechanisms: Single word Repeat and Block Repeat with four levels of nesting.
- Memory mapped I/O.
- Wait state support.
- Support for Program memory protection.
- Upward compatibility with the SPC instruction-set.
- On-chip emulation support.
- Automatic boot procedure support.

## 4.2.2 Buses

### 4.2.2.1 Data Buses

Data is transferred on the following 16-bit buses:

two bidirectional buses –the X Data Bus (XDB); the internal core general data bus (GDB) and one unidirectional bus –the Y Data Bus (YDB).

The XDB is the main data bus, where most of the data transfers occur. It is a buffered extension of the internal general data bus (GDB). Data transfer between the Y Data Memory (YRAM) and the Multiplier (Y register) occurs over the YDB, when a multiply instruction uses two data memory location simultaneously.

The bus structure supports register to register, register to memory, memory to register, program to data, program to register and data to program data movements. It can transfer up to two 16-bit words within one instruction cycle.

Addresses are specified for the local X\_RAM and Y\_RAM on two unidirectional buses: the 16-bit X Address Bus (XAB), and the 11-bit Y Address Bus (YAB).

### 4.2.2.2 Program Buses

Program memory addresses are specified on the 16-bit unidirectional Program Address Bus (PAB) while the Instruction word fetches take place in parallel over Program Data bus (PDB).

### 4.2.3 Memory Spaces and Organization

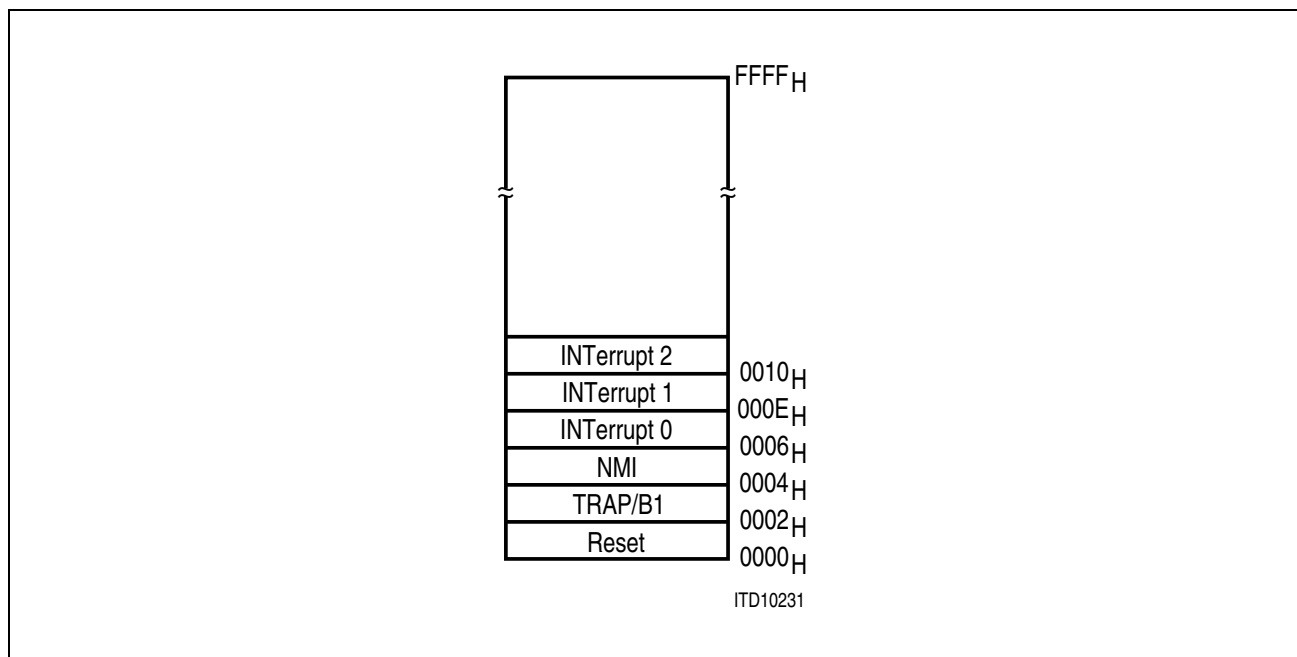
Two independent memory spaces are available: the data space (XRAM and YRAM) and the program space. Each of them is 64 K words.

#### 4.2.3.1 Program Memory

Addresses 0x0000-0x0016 (see **Figure 4-1**) are used as interrupt vectors for Reset, TRAP/BI (software interrupt/breakpoint interrupt), NMI (Non-maskable interrupt) and three maskable interrupts (INT0, INT1, INT2). The RESET, TRAP/BI and NMI vectors have been separated by two locations so that branch instructions can be accommodated in those locations if desired. The maskable interrupts have been separated by eight locations so that branch instructions, or small and fast interrupt service routines, can be accommodated in those locations.

The program memory can be implemented on-chip, and/or off-chip. The OAK supports a wait-state generator for interfacing slow program memory.

The program memory addresses are generated by the PCU.



**Figure 4-1 Program Memory Map**

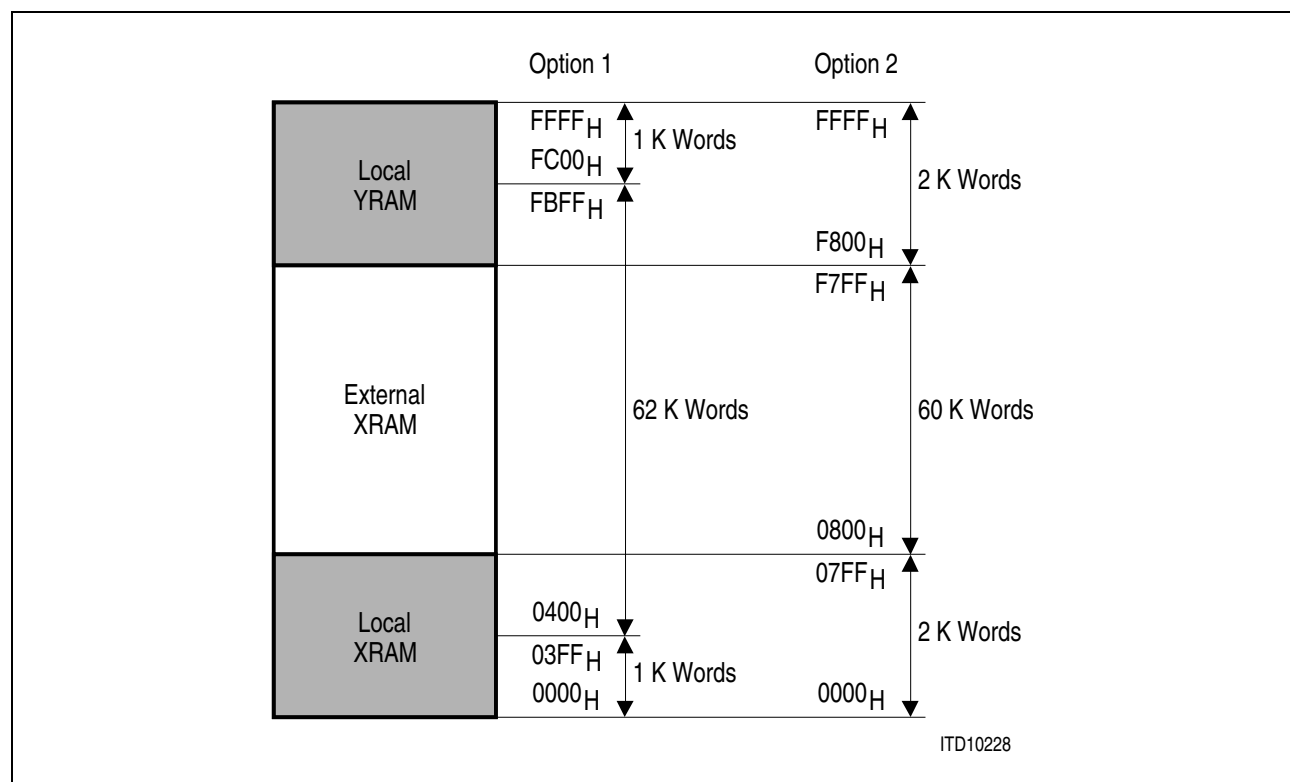


### 4.2.3.2 Data Memory

The data space is divided into a Y data space for the local YRAM, and a X data space for the XRAM. The XRAM space is divided into local data RAM/ROM of 1 K or 2 K, and an external space of 62 K or 60 K, respectively. The configuration is determined according to a core input pin. The local YRAM space and the local XRAM space were mapped to allow a continuous data space. The mapping is described in **Figure 4-2**. The data space partition allows expansion of the local XRAM and YRAM, and at the same time enables looking at the two RAMs as a single continuous data RAM. The XRAM and YRAM can contain RAM or ROM. The X data memory can be expanded external (with no additional wait state cycles) up to the YRAM boundary.

Memory-mapped I/O is used, thus several addresses of the data space locations may be reserved for peripherals depending on the specific application configuration. Wait states generation is possible.

The external space, within the XRAM space, may contain slow memories and peripherals as well as fast memories and peripherals. When using slow memories, additional wait cycles have to be inserted.



**Figure 4-2 Data Memory Map**

For more details on OAK architecture please refer to “**DOC DSP Programmer’s Reference Manual, 11.97**”.

## **4.3 Development Tools**

### **4.3.1 COFF Macro Assembler**

The COFF Macro Assembler translates DSP assembly language source files into machine language object files. It consists of a macro pre-processor, has a complete programming restriction checking and prepares the object for full symbolic debugging.

Besides it is sensitive, contains C-like operators and conventions that allow easy development of code and data structures. The object files generated are compatible to the Common Object File Format (COFF).

### **4.3.2 Linker/Locator**

The Linker/Locator combines object files generated by the COFF Macro Assembler into a single executable COFF object file. As it creates the executable object file, it performs relocation which means map them to the target systems memory map. Besides it supports user defined memory classes and enables to locate segments at absolute locations or relative to other segments and to overlay segments. The linking capability is very flexible and modular.

### **4.3.3 Object Format Converter**

Most EPROM programmers do not accept executable COFF object files as input. Therefore the Object Format Converter translates the COFF file into Intel hex file format that can be downloaded to any ordinary EPROM programmer.

### **4.3.4 ANSI C-Compiler**

The C-Compiler is full featured ANSI Standard C-Compiler which accepts C source code and produces assembler language source code, that can be processed by the COFF Macro Assembler. It uses a sophisticated optimization pass that employs several advanced techniques for generating efficient, compact code from C source. Beside the mixed language environment allows time critical routines to be replaced with very fast assembly routines for optimum performance with C language environment. Both, in-line assembly and out-line assembly are supported. For stand alone applications, the compiler enables to link all code and initialization data into ROM, allowing C code to run from reset. The C-compiler is fully integrated into the development environment, which allows the user complete control over C or assembly code during debugging. The compiler is based on the technology developed by the Free Software Foundation (GNU).

#### **4.3.5 Simulator**

The Simulator simulates the operation for program verification and debugging purposes. It simulates the entire instruction set and accepts executable COFF object code, generated from the Linker/Locator. The Simulator allows verification and monitoring the DSP states without the requirement of hardware. Besides a windowed, mouse driven interface which can be user customized, it also contains a high level language debug interface. To simulate external signals and hardware logic, it is possible to connect DOS files and integrate C functions using the Dynamic Link Library (DLL) mechanism of WINDOWS. During program execution, the internal registers and memory of the simulated DSP are modified as each instruction is interpreted by the host. Execution is suspended when either a breakpoint or an error is encountered or when the user halts execution. Then the DSP internal registers and both program and data memory can be inspected and modified.

#### **4.3.6 Debugger**

To debug a program in realtime the simulator contains an emulation mode that directly interfaces to the On Chip Emulation Module (OCEM). Compared with the simulation mode the user interface will be unchanged.

All software development tools can be used on a standard PC.

Different tools are provided for Win 3.1, Win 95 and Win NT.

## 5 Description of Registers

### 5.1 $\mu$ P Address Space Register Overview

The following table contains an overview of all DOC registers within the microprocessor address space.

**Table 5-1 ELIC0 SACCO-A**

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	000 <sub>H</sub> -01F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-56
XFIFO	WR	000 <sub>H</sub> -01F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	020 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	020 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	021 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	021 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	022 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	024 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	024 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	025 <sub>H</sub>	X0 <sub>H</sub>	transmit address 2	5-65
RBCL	RD	025 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	026 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	027 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	027 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	028 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	029 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	029 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	02A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	02C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	02D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	02D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	02E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	02E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	02F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61

Description of Registers

**Table 5-1 ELIC0 SACCO-A (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
TSAX	WR	030 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	031 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receive	5-86
XCCR	WR	032 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	033 <sub>H</sub>	00 <sub>H</sub>	receive channel cap.	5-86

**Table 5-2 ELIC0 SACCO-B**

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	040 <sub>H</sub> -05F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-56
XFIFO	WR	040 <sub>H</sub> -05F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	060 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	060 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	061 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	061 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	062 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	064 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	064 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	065 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	065 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	066 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	067 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	067 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-64
RAL1	RD/WR	068 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	069 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	069 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	06A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-67
CCR2	RD/WR	06C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	06D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67

Description of Registers

**Table 5-2 ELIC0 SACCO-B (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
XBCH	WR	06D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	06E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	06E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	06F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	070 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	071 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receive	5-86
XCCR	WR	072 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	073 <sub>H</sub>	00 <sub>H</sub>	receive channel cap.	5-86

**Table 5-3 ELIC0-EPIC**

Reg Name	Access	Address	Reset Value	Comment	Page No.
MACR	RD/WR	080 <sub>H</sub>	XX <sub>H</sub>	memory access control register	5-38
MAAR	RD/WR	081 <sub>H</sub>	XX <sub>H</sub>	memory access address register	5-42
MADR	RD/WR	082 <sub>H</sub>	XX <sub>H</sub>	memory access data register	5-43
STDA	RD/WR	083 <sub>H</sub>	XX <sub>H</sub>	synchron transfer data reg. A	5-43
STDB	RD/WR	084 <sub>H</sub>	XX <sub>H</sub>	synchron transfer data reg. B	5-43
SARA	RD/WR	085 <sub>H</sub>	XX <sub>H</sub>	synchron transfer receive address reg. A	5-43
SARB	RD/WR	086 <sub>H</sub>	XX <sub>H</sub>	synchron transfer receive address reg. B	5-44
SAXA	RD/WR	087 <sub>H</sub>	XX <sub>H</sub>	synchron transfer transmit address reg. A	5-45
SAXB	RD/WR	088 <sub>H</sub>	XX <sub>H</sub>	synchron transfer transmit address reg. B	5-45

Description of Registers

**Table 5-3 ELIC0-EPIC (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
STCR	RD/WR	089 <sub>H</sub>	00xx <sub>H</sub> xxxx <sub>H</sub>	synchron transfer control reg.	5-46
MFAIR	RD	08A <sub>H</sub>	0xxx <sub>H</sub> xxxx <sub>H</sub>	MF-channel active indication reg.	5-47
MFSAR	WR	08A <sub>H</sub>	00 <sub>H</sub>	MF-channel subscriber address reg.	5-47
MFFIFO	RD/WR	08B <sub>H</sub>	XX <sub>H</sub>	MF-channel FIFO	5-48
CIFIFO	RD	08C <sub>H</sub>	00 <sub>H</sub>	signaling channel FIFO	5-48
TIMR	WR	08C <sub>H</sub>	00 <sub>H</sub>	timer register	5-49
STAR_E	RD	08D <sub>H</sub>	05 <sub>H</sub>	status register EPIC	5-49
CMDR_E	WR	08D <sub>H</sub>	00 <sub>H</sub>	command register EPIC	5-50
ISTA_E	RD	08E <sub>H</sub>	00 <sub>H</sub>	interrupt status EPIC	5-52
MASK_E	WR	08E <sub>H</sub>	00 <sub>H</sub>	mask register EPIC	5-53
OMDR	RD/WR	08F <sub>H</sub>	00 <sub>H</sub>	operation mode register	5-54
PMOD	RD/WR	090 <sub>H</sub>	00 <sub>H</sub>	PCM-mode register	5-26
PBNR	RD/WR	091 <sub>H</sub>	FF <sub>H</sub>	PCM bit-number reg.	5-28
POFD	RD/WR	092 <sub>H</sub>	00 <sub>H</sub>	PCM-offset downstream register	5-28
POFU	RD/WR	093 <sub>H</sub>	00 <sub>H</sub>	PCM-offset upstream register	5-29
PCSR	RD/WR	094 <sub>H</sub>	00 <sub>H</sub>	PCM-clock shift register	5-29
PICM	RD	095 <sub>H</sub>	XX <sub>H</sub>	PCM-input comparison mismatch reg.	5-30
CMD1	RD/WR	096 <sub>H</sub>	00 <sub>H</sub>	CFI-mode reg.1	5-30
CMD2	RD/WR	097 <sub>H</sub>	00 <sub>H</sub>	CFI-mode reg. 2	5-33
CBNR	RD/WR	098 <sub>H</sub>	FF <sub>H</sub>	CFI-bit number reg.	5-34
CTAR	RD/WR	099 <sub>H</sub>	00 <sub>H</sub>	CFI time-slot adjustment register	5-35
CBSR	RD/WR	09A <sub>H</sub>	00 <sub>H</sub>	CFI-bit shift reg.	5-35
CSCR	RD/WR	09B <sub>H</sub>	00 <sub>H</sub>	CFI-subchannel register	5-37
VNSR	RD/WR	09D <sub>H</sub>	01 <sub>H</sub>	version No. status register	5-56

Description of Registers

**Table 5-4 ELIC0-MODE REGISTER**

Reg Name	Access	Address	Reset Value	Comment	Page No.
EMOD	RD/WR	0BF <sub>H</sub>	XF <sub>H</sub>	ELIC mode version No.	5-24

**Table 5-5 ELIC0-WATCH-DOG TIMER**

Reg Name	Access	Address	Reset Value	Comment	Page No.
WTC	RD/WR	0C0 <sub>H</sub>	1F <sub>H</sub>	watchdog timer control reg.	5-23

**Table 5-6 ELIC0-INTERRUPT TOP LEVEL**

Reg Name	Access	Address	Reset Value	Comment	Page No.
ISTA	RD	0C1 <sub>H</sub>	00 <sub>H</sub>	interrupt status register	5-22
MASK	WR	0C1 <sub>H</sub>	00 <sub>H</sub>	mask register	5-23

**Table 5-7 ELIC0-ARBITER**

Reg Name	Access	Address	Reset Value	Comment	Page No.
AMO	RD/WR	0E0 <sub>H</sub>	00 <sub>H</sub>	arbiter mode register	5-87
ASTATE	RD	0E1 <sub>H</sub>	XX <sub>H</sub>	arbiter state register	5-87
SCV	RD/WR	0E2 <sub>H</sub>	00 <sub>H</sub>	suspend counter value register	5-87
DCE0	RD/WR	0E3 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 0	5-89
DCE1	RD/WR	0E4 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg.1	5-89
DCE2	RD/WR	0E5 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 2	5-89
DCE3	RD/WR	0E6 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 3	5-89
XDC	RD/WR	0E7 <sub>H</sub>	00 <sub>H</sub>	transmit D-channel address register	5-90
BCG0	RD/WR	0E8 <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 0	5-90



Description of Registers

**Table 5-7 ELIC0-ARBITER (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
BCG1	RD/WR	0E9 <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 1	5-90
BCG2	RD/WR	0EA <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 2	5-90
BCG3	RD/WR	0EB <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 3	5-91

**Table 5-8 ELIC1 SACCO-A**

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	100 <sub>H</sub> -11F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-56
XFIFO	WR	100 <sub>H</sub> -11F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	120 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	120 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	121 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	121 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	122 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	124 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	124 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	125 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	125 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	126 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	127 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	127 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	128 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	129 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	129 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	12A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	12C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	12D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	12D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85

Description of Registers

**Table 5-8 ELIC1 SACCO-A (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
VSTR	RD	12E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	12E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	12F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	130 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	131 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receiver	5-86
XCCR	WR	132 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	133 <sub>H</sub>	00 <sub>H</sub>	receive channel cap.	5-86

**Table 5-9 ELIC1 SACCO-B**

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	140 <sub>H</sub> -15F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-56
XFIFO	WR	140 <sub>H</sub> -15F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	160 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	160 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	161 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	161 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	162 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	164 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	164 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	165 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	165 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	166 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	167 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	167 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-64
RAL1	RD/WR	168 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	169 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	169 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	16A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-67

Description of Registers

**Table 5-9 ELIC1 SACCO-B (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
CCR2	RD/WR	16C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	16D <sub>H</sub>	000xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	16D <sub>H</sub>	000xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	16E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	16E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	16F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	170 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmitter	5-85
TSAR	WR	171 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receiver	5-86
XCCR	WR	172 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	173 <sub>H</sub>	00 <sub>H</sub>	receive channel cap.	5-86

**Table 5-10 ELIC1-EPIC**

Reg Name	Access	Address	Reset Value	Comment	Page No.
MACR	RD/WR	180 <sub>H</sub>	XX <sub>H</sub>	memory access control register	5-38
MAAR	RD/WR	181 <sub>H</sub>	XX <sub>H</sub>	memory access address register	5-42
MADR	RD/WR	182 <sub>H</sub>	XX <sub>H</sub>	memory access data register	5-43
STDA	RD/WR	183 <sub>H</sub>	XX <sub>H</sub>	synchron transfer data reg. A	5-43
STDB	RD/WR	184 <sub>H</sub>	XX <sub>H</sub>	synchron transfer data reg. B	5-43
SARA	RD/WR	185 <sub>H</sub>	XX <sub>H</sub>	synchron transfer receive address reg. A	5-44
SARB	RD/WR	186 <sub>H</sub>	XX <sub>H</sub>	synchron transfer receive address reg. B	5-44
SAXA	RD/WR	187 <sub>H</sub>	XX <sub>H</sub>	synchron transfer transmit address reg. A	5-45
SAXB	RD/WR	188 <sub>H</sub>	XX <sub>H</sub>	synchron transfer transmit address reg. B	5-45

Description of Registers

**Table 5-10 ELIC1-EPIC (cont'd)**

<b>Reg Name</b>	<b>Access</b>	<b>Address</b>	<b>Reset Value</b>	<b>Comment</b>	<b>Page No.</b>
STCR	RD/WR	189 <sub>H</sub>	00xxxxxx <sub>H</sub>	synchron transfer control reg.	5-46
MFAIR	RD	18A <sub>H</sub>	0xxxxxxx <sub>H</sub>	MF-channel active indication reg.	5-47
MFSAR	WR	18A <sub>H</sub>	00 <sub>H</sub>	MF-channel subscriber address reg.	5-47
MFFIFO		18B <sub>H</sub>	XX <sub>H</sub>	MF-channel FIFO	5-48
CIFIFO	RD	18C <sub>H</sub>	00 <sub>H</sub>	signaling channel FIFO	5-48
TIMR	WR	18C <sub>H</sub>	00 <sub>H</sub>	timer register	5-49
STAR_E	RD	18D <sub>H</sub>	05 <sub>H</sub>	status register EPIC	5-49
CMDR_E	WR	18D <sub>H</sub>	00 <sub>H</sub>	command register EPIC	5-50
ISTA_E	RD	18E <sub>H</sub>	00 <sub>H</sub>	interrupt status EPIC	5-52
MASK_E	WR	18E <sub>H</sub>	00 <sub>H</sub>	mask register EPIC	5-53
OMDR	RD/WR	18F <sub>H</sub>	00 <sub>H</sub>	operation mode register	5-54
PMOD	RD/WR	190 <sub>H</sub>	00 <sub>H</sub>	PCM-mode register	5-26
PBNR	RD/WR	191 <sub>H</sub>	FF <sub>H</sub>	PCM bit-number reg.	5-28
POFD	RD/WR	192 <sub>H</sub>	00 <sub>H</sub>	PCM-offset downstream register	5-28
POFU	RD/WR	193 <sub>H</sub>	00 <sub>H</sub>	PCM-offset upstream register	5-29
PCSR	RD/WR	194 <sub>H</sub>	00 <sub>H</sub>	PCM-clock shift register	5-29
PICM	RD	195 <sub>H</sub>	XX <sub>H</sub>	PCM-input comparison mismatch reg.	5-30
CMD1	RD/WR	196 <sub>H</sub>	00 <sub>H</sub>	CFI-mode reg.1	5-30
CMD2	RD/WR	197 <sub>H</sub>	00 <sub>H</sub>	CFI-mode reg. 2	5-33
CBNR	RD/WR	198 <sub>H</sub>	FF <sub>H</sub>	CFI-bit number reg.	5-34
CTAR	RD/WR	199 <sub>H</sub>	00 <sub>H</sub>	CFI time-slot adjustment register	5-35
CBSR	RD/WR	19A <sub>H</sub>	00 <sub>H</sub>	CFI-bit shift reg.	5-35
CSCR	RD/WR	19B <sub>H</sub>	00 <sub>H</sub>	CFI-subchannel register	5-37
VNSR	RD/WR	19D <sub>H</sub>	01 <sub>H</sub>	version No. status register	5-56

**Table 5-11 ELIC1-MODE REGISTER**

Reg Name	Access	Address	Reset Value	Comment	Page No.
EMOD	RD/WR	1BF <sub>H</sub>	XF <sub>H</sub>	ELIC mode version No.	5-24

**Table 5-12 ELIC1-INTERRUPT TOP LEVEL**

Reg Name	Access	Address	Reset Value	Comment	Page No.
ISTA	RD	1C1 <sub>H</sub>	00 <sub>H</sub>	interrupt status register	5-22
MASK	WR	1C1 <sub>H</sub>	00 <sub>H</sub>	mask register	5-23

**Table 5-13 ELIC1-ARBITER**

Reg Name	Access	Address	Reset Value	Comment	Page No.
AMO	RD/WR	1E0 <sub>H</sub>	00 <sub>H</sub>	arbiter mode register	5-87
ASTATE	RD	1E1 <sub>H</sub>	XX <sub>H</sub>	arbiter state register	5-88
SCV	RD/WR	1E2 <sub>H</sub>	00 <sub>H</sub>	suspend counter value register	5-88
DCE0	RD/WR	1E3 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 0	5-89
DCE1	RD/WR	1E4 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg.1	5-89
DCE2	RD/WR	1E5 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 2	5-89
DCE3	RD/WR	1E6 <sub>H</sub>	00 <sub>H</sub>	D-channel enable reg. 3	5-89
XDC	RD/WR	1E7 <sub>H</sub>	00 <sub>H</sub>	transmit D-channel address register	5-90
BCG0	RD/WR	1E8 <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 0	5-90
BCG1	RD/WR	1E9 <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 1	5-90
BCG2	RD/WR	1EA <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 2	5-90
BCG3	RD/WR	1EB <sub>H</sub>	00 <sub>H</sub>	broadcast group reg. 3	5-91

Description of Registers

Table 5-14 SIDE0

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	200 <sub>H</sub> -21F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-57
XFIFO	WR	200 <sub>H</sub> -21F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	220 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	220 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	221 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	221 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	222 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	224 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	224 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	225 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	225 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	226 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	227 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	227 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	228 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	229 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	229 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	22A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	22C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	22D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	22D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	22E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	22E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	22F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	230 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	231 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receiver	5-86
XCCR	WR	232 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	233 <sub>H</sub>	00 <sub>H</sub>	receive channel cap.	5-86

Description of Registers

**Table 5-15 SIDEC1**

<b>Reg Name</b>	<b>Access</b>	<b>Address</b>	<b>Reset Value</b>	<b>Comment</b>	<b>Page No.</b>
RFIFO	RD	240 <sub>H</sub> -25F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-57
XFIFO	WR	240 <sub>H</sub> -25F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	260 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	260 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	261 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	261 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	262 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	264 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	264 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	265 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	265 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	266 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	267 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	267 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	268 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	269 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	269 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	26A <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	26C <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	26D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	26D <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	26E <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	26E <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	26F <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	270 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	271 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receive	5-86
XCCR	WR	272 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	273 <sub>H</sub>	00 <sub>H</sub>	receive channell capacity	5-86

Description of Registers

Table 5-16 SIDEC2

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	280 <sub>H</sub> -29F <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-57
XFIFO	WR	280 <sub>H</sub> -29F <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	2A0 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	2A0 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	2A1 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	2A1 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	2A2 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	2A4 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	2A4 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	2A5 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	2A5 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	2A6 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	2A7 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	2A7 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	2A8 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	2A9 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	2A9 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	2AA <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	2AC <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	2AD <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	2AD <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	2AE <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	2AE <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	2AF <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	2B0 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	2B1 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receiver	5-86
XCCR	WR	2B2 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	2B3 <sub>H</sub>	00 <sub>H</sub>	receive channell capacity	5-86



Description of Registers

Table 5-17 SIDE3

Reg Name	Access	Address	Reset Value	Comment	Page No.
RFIFO	RD	2C0 <sub>H</sub> -2DF <sub>H</sub>	XX <sub>H</sub>	receive FIFO	5-57
XFIFO	WR	2C0 <sub>H</sub> -2DF <sub>H</sub>	XX <sub>H</sub>	transmit FIFO	5-57
ISTA	RD	2E0 <sub>H</sub>	00 <sub>H</sub>	int. status reg.	5-57
MASK	WR	2E0 <sub>H</sub>	00 <sub>H</sub>	mask reg.	5-57
STAR	RD	2E1 <sub>H</sub>	48 <sub>H</sub>	status reg.	5-62
CMDR	WR	2E1 <sub>H</sub>	00 <sub>H</sub>	command reg.	5-59
MODE	RD/WR	2E2 <sub>H</sub>	00 <sub>H</sub>	mode reg.	5-60
XAD1	WR	2E4 <sub>H</sub>	XX <sub>H</sub>	transmit address 1	5-65
EXIR	RD	2E4 <sub>H</sub>	00 <sub>H</sub>	extended int.	5-58
XAD2	WR	2E5 <sub>H</sub>	XX <sub>H</sub>	transmit address 2	5-65
RBCL	RD	2E5 <sub>H</sub>	00 <sub>H</sub>	receive byte count low	5-67
RAH1	WR	2E6 <sub>H</sub>	XX <sub>H</sub>	receive address high 1	5-66
RAH2	WR	2E7 <sub>H</sub>	XX <sub>H</sub>	receive address high 2	5-67
RSTA	RD	2E7 <sub>H</sub>	XX <sub>H</sub>	receive status reg.	5-63
RAL1	RD/WR	2E8 <sub>H</sub>	XX <sub>H</sub>	receive address low 1	5-65
RAL2	WR	2E9 <sub>H</sub>	XX <sub>H</sub>	receive address low 2	5-66
RHCR	RD	2E9 <sub>H</sub>	XX <sub>H</sub>	receive HDLC control byte	5-64
XBCL	WR	2EA <sub>H</sub>	XX <sub>H</sub>	transmit byte count low	5-84
CCR2	RD/WR	2EC <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 2	5-62
RBCH	RD	2ED <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	receive byte count high	5-67
XBCH	WR	2ED <sub>H</sub>	000x <sub>H</sub> xxxx <sub>H</sub>	transmit byte count high	5-85
VSTR	RD	2EE <sub>H</sub>	80 <sub>H</sub>	version stat. reg.	5-68
RLCR	WR	2EE <sub>H</sub>	XX <sub>H</sub>	receive frame length check	5-62
CCR1	RD/WR	2EF <sub>H</sub>	00 <sub>H</sub>	channel config. reg. 1	5-61
TSAX	WR	2F0 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment transmit	5-85
TSAR	WR	2F1 <sub>H</sub>	XX <sub>H</sub>	time-slot assignment receive	5-86
XCCR	WR	2F2 <sub>H</sub>	00 <sub>H</sub>	transmit channel capacity	5-86
RCCR	WR	2F3 <sub>H</sub>	00 <sub>H</sub>	receive channell capacity	5-86

Description of Registers

**Table 5-18 ICU**

Reg Name	Access	Address	Reset Value	Comment	Page No.
IDOC	RD/WR	300 <sub>H</sub>	00 <sub>H</sub>		2-131
IPC	RD/WR	301 <sub>H</sub>	00 <sub>H</sub>		2-132
IMASKR0	RD/WR	302 <sub>H</sub>	FF <sub>H</sub>		2-129
IMASKR1	RD/WR	303 <sub>H</sub>	07 <sub>H</sub>		2-130
IPAR0	RD/WR	304 <sub>H</sub>	00 <sub>H</sub>		2-131
IPAR1	RD/WR	305 <sub>H</sub>	00 <sub>H</sub>		2-131
IPAR2	RD/WR	306 <sub>H</sub>	00 <sub>H</sub>		2-131

**Table 5-19 GPIO**

Reg Name	Access	Address	Reset Value	Comment	Page No.
VCFGR	RD/WR	320 <sub>H</sub>	X0 <sub>H</sub>	config. register (7-bit only)	2-157
VDATR	RD/WR	321 <sub>H</sub>	X0 <sub>H</sub>	data register (4-bit only)	2-157
VNR	RD	322 <sub>H</sub>	X <sub>H</sub>	Version No. (4-bit read-only)	2-158

**Table 5-20 CHI**

Reg Name	Access	Address	Reset Value	Comment	Page No.
VMODR	RD/WR	323 <sub>H</sub>	xxxx x000	mode reg. (3-bit only)	2-58
VDTR0	RD/WR	324 <sub>H</sub>	XX <sub>H</sub>	data reg. 0	2-59
VDTR1	RD/WR	325 <sub>H</sub>	XX <sub>H</sub>	data reg. 1	2-59
VDTR2	RD/WR	326 <sub>H</sub>	XX <sub>H</sub>	data reg. 2	2-59
VDTR3	RD/WR	327 <sub>H</sub>	XX <sub>H</sub>	data reg. 3	2-59

Description of Registers

**Table 5-21 OAK MAIL BOX**

Reg Name	Access	Address	Reset Value	Comment	Page No.
MCMD	μP: WR OAK: RD	340 <sub>H</sub>	00 <sub>H</sub>	microprocessor command	2-117
MBUSY	OAK: WR μP: RD	341 <sub>H</sub>	00 <sub>H</sub>	microprocessor mail box busy	2-117
MDT0 (LSB)	μP: WR OAK: RD	342 <sub>H</sub>	unch'd	microprocessor data reg. 0	2-117
MDT0 (MSB)	μP: WR OAK: RD	343 <sub>H</sub>	unch'd		2-117
MDT1 (LSB)	μP: WR OAK: RD	344 <sub>H</sub>	unch'd	microprocessor data reg. 1	2-117
MDT1 (MSB)	μP: WR OAK: RD	345 <sub>H</sub>	unch'd		2-117
MDT2 (LSB)	μP: WR OAK: RD	346 <sub>H</sub>	unch'd	microprocessor data reg. 2	2-117
MDT2 (MSB)	μP: WR OAK: RD	347 <sub>H</sub>	unch'd		2-117
MDT3 (LSB)	μP: WR OAK: RD	348 <sub>H</sub>	unch'd	microprocessor data reg. 3	2-117
MDT3 (MSB)	μP: WR OAK: RD	349 <sub>H</sub>	unch'd		2-117
MDT4 (LSB)	μP: WR OAK: RD	34A <sub>H</sub>	unch'd	microprocessor data reg. 4	2-117
MDT4 (MSB)	μP: WR OAK: RD	34B <sub>H</sub>	unch'd		2-117
MDT5 (LSB)	μP: WR OAK: RD	34C <sub>H</sub>	unch'd	microprocessor data reg. 5	2-117
MDT5 (MSB)	μP: WR OAK: RD	34D <sub>H</sub>	unch'd		2-117
OCMD	μP: RD OAK: WR	350 <sub>H</sub>	00 <sub>H</sub>	OAK command	2-117
OBUSY	OAK: RD μP: WR	351 <sub>H</sub>	00 <sub>H</sub>	OAK mail box busy	2-117

Description of Registers

**Table 5-21 OAK MAIL BOX (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
ODT0 (LSB)	μP: RD OAK: WR	352 <sub>H</sub>	unch'd	OAK data reg. 0	2-117
ODT0 (MSB)	μP: RD OAK: WR	353 <sub>H</sub>	unch'd		2-117
ODT1 (LSB)	μP: RD OAK: WR	354 <sub>H</sub>	unch'd	OAK data reg. 1	2-117
ODT1 (MSB)	μP: RD OAK: WR	355 <sub>H</sub>	unch'd		2-117
ODT2 (LSB)	μP: RD OAK: WR	356 <sub>H</sub>	unch'd	OAK data reg. 2	2-117
ODT2 (MSB)	μP: RD OAK: WR	357 <sub>H</sub>	unch'd		2-117
ODT3 (LSB)	μP: RD OAK: WR	358 <sub>H</sub>	unch'd	OAK data reg. 3	2-117
ODT3 (MSB)	μP: RD OAK: WR	359 <sub>H</sub>	unch'd		2-117
ODT4 (LSB)	μP: RD OAK: WR	35A <sub>H</sub>	unch'd	OAK data reg. 4	2-117
ODT4 (MSB)	μP: RD OAK: WR	35B <sub>H</sub>	unch'd		2-117
ODT5 (LSB)	μP: RD OAK: WR	35C <sub>H</sub>	unch'd	OAK data reg. 5	2-117
ODT5 (MSB)	μP: RD OAK: WR	35D <sub>H</sub>	unch'd		2-117

**Table 5-22 CLOCKS**

Reg Name	Access	Address	Reset Value	Comment	Page No.
CCSEL0	RD/WR	360 <sub>H</sub>	0001 00xx <sup>(1)</sup>	(1) bits [1:0] are read-only and depends on values of freq0, freq1 pins during reset	2-125
CCSEL1	RD/WR	361 <sub>H</sub>	00 <sub>H</sub>	clocks select reg. 1	2-126

Description of Registers

**Table 5-22 CLOCKS** (cont'd)

Reg Name	Access	Address	Reset Value	Comment	Page No.
CCSEL2	RD/WR	362 <sub>H</sub>	00 <sub>H</sub>	clocks select reg. 2	2-128
TESTEN	RD/WR	363 <sub>H</sub>	00 <sub>H</sub>	clocks select reg. 1	2-126

**Table 5-23 IOM/PCM MUX**

Reg Name	Access	Address	Reset Value	Comment	Page No.
MMODE	RD/WR	380 <sub>H</sub>	01 <sub>H</sub>	MUX mode bits[7:4] reserved	2-53
MC CHSEL0	RD/WR	381 <sub>H</sub>	00 <sub>H</sub>	channel selection reg. 0	2-54
MC CHSEL1	RD/WR	382 <sub>H</sub>	00 <sub>H</sub>	channel selection reg. 1	2-55
MC CHSEL2	RD/WR	383 <sub>H</sub>	00 <sub>H</sub>	channel selection reg. 2	2-56
MP CHSEL0	RD/WR	384 <sub>H</sub>	00 <sub>H</sub>		2-57

**Table 5-24 UART**

Reg Name	Access	Address	Reset Value	Comment	Page No.
RBR	RD	3A0 <sub>H</sub>		LCR:DLAB = 0	2-139
THR	WR	3A0 <sub>H</sub>		LCR:DLAB = 0	2-139
IER	RD/WR	3A1 <sub>H</sub>		LCR:DLAB = 0	2-151
IIR	RD	3A2 <sub>H</sub>			2-149
FCR	WR	3A2 <sub>H</sub>			2-148
LCR	RD/WR	3A3 <sub>H</sub>			2-143
MCR	RD/WR	3A4 <sub>H</sub>			2-151
LSR	RD	3A5 <sub>H</sub>		written during production testing	2-145
MSR	RD/WR	3A6 <sub>H</sub>			2-153

Description of Registers

**Table 5-24 UART (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
SCR	RD/WR	3A7 <sub>H</sub>			2-154
DLL	RD/WR	3A0 <sub>H</sub>		Div. Lat. (LS); LCR: DLAB = 1	2-140
DLM	RD/WR	3A1 <sub>H</sub>		Div. Lat. (MS); LCR:DLAB = 1	2-140

**Table 5-25 PEDIU**

Reg Name	Access	Address	Reset Value	Comment	Page No.
UCR	RD/WR	C100 <sub>H</sub>	00 <sub>H</sub>	PEDIU Control Register	2-93
USR	RD	C101 <sub>H</sub>		PEDIU Status Register	2-97
UISBPER	SET	C102 <sub>H</sub>	00 <sub>H</sub>	PEDIU Input Stream ByPass Enable Register	2-101
	RESET	C103 <sub>H</sub>			
	RD	C104 <sub>H</sub>			
UOSBPR	RD	C105 <sub>H</sub>	00 <sub>H</sub>	PEDIU Output Stream ByPass Enable Register	2-103
	SET	C106 <sub>H</sub>			
	RESET	C107 <sub>H</sub>			
UTSR	SET	C108 <sub>H</sub>	00 <sub>H</sub>	PEDIU Tri-State Register	2-104
	RESET	C109 <sub>H</sub>			
	RD	C10A <sub>H</sub>			
UPRTAR	RD/WR	C10B <sub>H</sub>		PEDIU-ROM Test Address Register	2-106
UPRTDR	RD	C10C <sub>H</sub>		PEDIU-ROM Test Data Register	2-106

**Table 5-26 DCU**

Reg Name	Access	Address	Reset Value	Comment	Page No.
BOOTCONF		C001 <sub>H</sub>		Boot Configuration Register	2-79
MEMCONFR		C002 <sub>H</sub>	0007 <sub>H</sub>	Memory Configuration Register	2-68
TESTCONFR		C003 <sub>H</sub>		Test Configuration Register	2-69
STATC		C004 <sub>H</sub>	unch'd	Run Time Statistics Counter	2-71

Description of Registers

**Table 5-26 DCU (cont'd)**

Reg Name	Access	Address	Reset Value	Comment	Page No.
STATR		C005 <sub>H</sub>	unch'd	Run Time Statistics Register	2-71
PASSR		C006 <sub>H</sub>	0000 <sub>H</sub>	Program Write Protection Register	2-73
JCONF		CO07 <sub>H</sub>	8000 <sub>H</sub>	Serial Emulation configuration register. Bit 14 is determined by strap	2-73

**Table 5-27 OAK MAIL BOX**

Reg Name	Access	Address	Reset Value	Comment	Page No.
MCMD	μP: WR OAK: RD	C040 <sub>H</sub>	0000 <sub>H</sub>	microprocessor command	2-117
MBUSY	μP: WR OAK: RD	C041 <sub>H</sub>	0000 <sub>H</sub>	microprocessor mail box busy	2-117
MDT0	μP: WR OAK: RD	C042 <sub>H</sub>	unch'd	microprocessor data reg. 0	2-117
MDT1	μP: WR OAK: RD	C044 <sub>H</sub>	unch'd	microprocessor data reg.1	2-117
MDT2	μP: WR OAK: RD	C046 <sub>H</sub>	unch'd	microprocessor data reg. 2	2-117
MDT3	μP: WR OAK: RD	C048 <sub>H</sub>	unch'd	microprocessor data reg. 3	2-117
MDT4	μP: WR OAK: RD	C04A <sup>H</sup>	unch'd	microprocessor data reg. 4	2-117
MDT5	μP: WR OAK: RD	C04C <sub>H</sub>	unch'd	microprocessor data reg. 5	2-117
OCMD	μP: RD OAK: WR	C050 <sub>H</sub>	0000 <sub>H</sub>	OAK command	2-117
OBUSY	μP: RD OAK: WR	C051 <sub>H</sub>	0000 <sub>H</sub>	OAK mail box busy	2-117
ODT0	μP: RD OAK: WR	C052 <sub>H</sub>	unch'd	OAK data reg. 0	2-117

**Table 5-27 OAK MAIL BOX (cont'd)**

<b>Reg Name</b>	<b>Access</b>	<b>Address</b>	<b>Reset Value</b>	<b>Comment</b>	<b>Page No.</b>
ODT1	μP: RD OAK: WR	C054 <sub>H</sub>	unch'd	OAK data reg. 1	<b>2-117</b>
ODT2	μP: RD OAK: WR	C056 <sub>H</sub>	unch'd	OAK data reg. 2	<b>2-117</b>
ODT3	μP: RD OAK: WR	C058 <sup>H</sup>	unch'd	OAK data reg. 3	<b>2-117</b>
ODT4	μP: RD OAK: WR	C05A <sub>H</sub>	unch'd	OAK data reg. 4	<b>2-117</b>
ODT5	μP: RD OAK: WR	C05C <sub>H</sub>	unch'd	OAK data reg. 5	<b>2-117</b>



### 5.1.1 ELIC0 and ELIC1 Registers Description

#### 5.1.1.1 Interrupt Top Level

##### 5.1.1.1.1 Interrupt Status Register (ISTA)

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

IWD	IDA	IEP	EXB	ICB	EXA	ICA	0
-----	-----	-----	-----	-----	-----	-----	---

**IWD** Interrupt Watchdog Timer (ELIC0 only).

The watchdog timer is expired and an external reset (RESIN) was generated. The software failed to program the bits WTC1 and WTC2 in WTC register in the correct sequence.

**IDA** Interrupt D-channel Arbiter.

The suspend counter expired while the arbiter was in the state “expect frame”. The affected D-channel can be determined by reading register ASTATE.

**IEP** Interrupt EPIC-1,  
detailed information is indicated in register ISTA\_E.

**EXB** Extended interrupt SACCO-B,  
detailed information is indicated in register EXIR\_B.

**ICB** Interrupt SACCO-B,  
detailed information is indicated in register ISTA\_B.

**EXA** Extended interrupt SACCO-A,  
detailed information is indicated in register EXIR\_A.

**ICA** Interrupt SACCO-A,  
detailed information is indicated in register ISTA\_A.

IWD and IDA are reset when reading ISTA. The other bits are reset when reading the corresponding local ISTA- or EXIR-register.

### 5.1.1.1.2 Mask Register (MASK)

Access: write

Reset value: 00<sub>H</sub> (all interrupts enabled)

bit 7						bit 0	
0	IDA	IEP	EXB	ICB	EXA	ICA	0

**IDA** enables(0)/disables(1) the D-Channel Arbiter interrupt

**IEP** enables(0)/disables(1) the EPIC-1 Interrupts

**EXB** enables(0)/disables(1) the SACCO-B Extended interrupts

**ICB** enables(0)/disables(1) the SACCO-B Interrupts

**EXA** enables(0)/disables(1) the SACCO-A Extended interrupts

**ICA** enables(0)/disables(1) the SACCO-A Interrupts

Each interrupt source/group can be selectively masked by setting the respective bit in the MASK-register (bit position corresponding to the ISTA-register). A masked IDA-interrupt is not indicated when reading ISTA. Instead it remains internally stored and will be indicated after the respective MASK-bit is reset. The watchdog timer interrupts is not maskable.

Even with a set MASK-bit EPIC-1 and SACCO-interrupts are indicated but no interrupt signal is generated.

When writing the MASK-register while an interrupt is indicated,  $\overline{INT}$  is temporarily set into the inactive state.

### 5.1.1.2 Watchdog Timer (in ELIC0 only)

#### 5.1.1.2.1 Watchdog Control Register (WTC)

Access: read/write

Reset value: 1F<sub>H</sub>

bit 7						bit 0	
WTC1	WTC2	SWT	1	1	1	1	1

**SWT** Start Watchdog Timer.

When set, the watchdog timer is started. The only way to disable it, is a ELIC-reset (power-up or  $\overline{DRESET}$ ).

Description of Registers

**WTC1...2** Watchdog Timer Control.

Once the watchdog timer has been started WTC1, WTC2 have to be written once every 1024 PFS-cycles in the following sequence in order to prevent the watchdog expiring.

WTC1    WTC2

1) 1 0

2) 0 1

The minimum required interval between the two write accesses is 2 PDC-periods.

**5.1.1.3    ELIC® Mode Register**

Access: read/write

Reset value: XF<sub>H</sub>

bit 7	“don’t care”	1	1	ECMD2	bit 0 1
-------	--------------	---	---	-------	------------

**ECMD2**    ELIC CFI-Mode Bit 2.

If set to ‘0’, the CFI-mode 0 with a 2.048-Mbit/s data rate can be used with a 2.048-MHz PDC-input clock.

This mode requires further restrictions of the current ELIC-specification:

1) EPIC-1 PMOD:PCR must be set to ‘1’.

*Note: Although the PCM clock PDC is set to double clock rate by this bit, the data rate must always be equal to the clock rate.*

2) EPIC-1 CMD2:COC must be programmed to ‘0’, i.e. it is not possible to output a DCL-clock with a frequency of twice the CFI-data rate.

3) EPIC-1 CMD1:CSS must be programmed to ‘0’, i.e. it is not possible to select DCL as clock and FSC as framing signal source for the configurable interface.

4) The timing of the PCM-interface is expanded:

**Table 5-28**

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Clock period	$T_{CP}$	480	–	ns	EMOD:ECMD2 = ‘0’
Clock period low	$T_{CPL}$	200	–	ns	
Clock period high	$T_{CPH}$	200	–	ns	

- 5) PCSR:DRE has to be set to '1'.  
 PCSR:URE has to be set to '1'.  
 When provided with a 2 MHz PDC, the ELIC internally generates a 4 MHz clock.  
 Since the clock shift capabilities (provided by register bits PCSR:DRCS and PCSR:ADSR0) apply to the internal 4 MHz clock, the frame can thus be shifted with a resolution of a half bit.

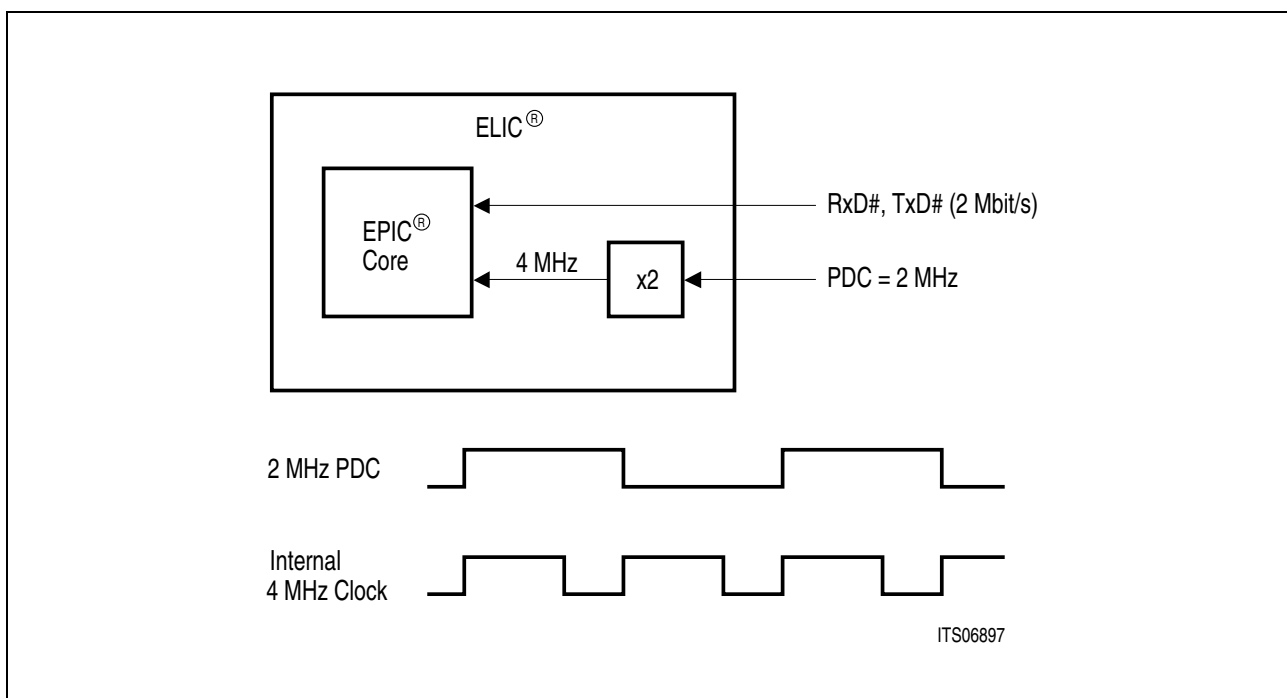


Figure 5-1 Timing Relation Between Internal and External Clock

- 6) PMOD:PSM has to be set to '1'.  
 The frame signal PFS must always be sampled with the rising edge of PDC. The set-up and hold times of PFS are still valid respected to external PDC.

5.1.1.4 EPIC®-1

5.1.1.4.1 PCM-Mode Register (PMOD)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7

bit 0

PMD1	PMD0	PCR	PSM	AIS1	AIS0	AIC1	AIC0
------	------	-----	-----	------	------	------	------

Note: If EMOD:ECMD2 is set to '0' some restrictions apply to the setting of register PMOD (see chapter 5.1.1.3).

**PMD1...0** PCM-Mode. Defines the actual number of PCM-ports, the data rate range and the data rate stepping.

Table 5-29

PMD1...0	PCM-Mode	Port Count	Data Rate [Mbit/s]		Data Rate Stepping [kbit/s]
			min.	max.	
00	0	4	256	2048	256
01	1	2	512	4096	512
10	2	1	1024	8192	1024
11	3	2	512	4096	512

The actual selection of physical pins is described below (AIS1/0).

**PCR**

PCM-Clock Rate.

0...single clock rate, data rate is identical with the clock frequency selected for ELIC PDC input.

1...double clock rate, data rate is half the clock frequency selected for ELIC PDC input.

Note: Only single clock rate is allowed in PCM-mode 2!

**PSM**

PCM Synchronization Mode.

A rising edge on PFS synchronizes the PCM-frame. PFS is not evaluated directly but is sampled with PDC.

0...the external PFS is evaluated with the falling edge of PDC.

The internal PFS (internal frame start) occurs with the next rising edge of PDC.

1...the external PFS is evaluated with the rising edge of PDC.

The internal PFS (internal frame start) occurs with this rising edge of PDC.

Description of Registers

**AIS1...0** Alternative Input Selection (only in PCM-Ports-MUX Mode 0 possible).  
 These bits determine the relationship between the physical pins and the logical port numbers. The logical port numbers are used when programming the switching functions.

*Note: In PCM-mode 0 these bits may not be set!*

**Table 5-30**

PCM	Port 0			Port 1			Port 2			Port 3		
Mode	RxD0	TxD0	TSC0	RxD1	TxD1	TSC1	RxD2	TxD2	TSC2	RxD3	TxD3	TSC3
0	IN0	OUT0	val0	IN1	OUT1	val1	IN2	OUT2	val2	IN3	OUT3	val3
1	IN0 (AIS0=1)	OUT0	val0	IN0 (AIS0=0)	tri-state	AIS0	IN1 (AIS1=1)	OUT1	val1	IN1 (AIS1=0)	tri-state	AIS1
2	not active	OUT	val	not active	tri-state	AIS0	IN (AIS1=1)	undef.	undef.	IN (AIS1=0)	tri-state	AIS1
3	IN0 (AIS0=1)	OUT0	val0	IN0 (AIS0=0)	OUT0	val0	IN1 (AIS1=1)	OUT1	val1	IN1 (AIS1=0)	OUT1	val1

**AIC1** Alternate Input Comparison 1.  
 0...input comparison of port 2 and 3 is disabled  
 1...the inputs of port 2 and 3 are compared

**AIC0** Alternate Input Comparison 0.  
 0...input comparison of port 0 and 1 is disabled  
 1...the inputs of port 0 and 1 are compared

*Note: The comparison function is operational in all PCM-modes; however, a redundant PCM-line which can be switched over to by means of the PMOD: AIS-bits is only available in PCM-modes 1, 2 and 3.*

### 5.1.1.4.2 Bit Number per PCM-Frame (PBNR)

Access: read/write

Reset value: FF<sub>H</sub>

bit 7

bit 0

BNF7	BNF6	BNF5	BNF4	BNF3	BNF2	BNF1	BNF0
------	------	------	------	------	------	------	------

**BNF7...0** Bit Number per PCM Frame.

PCM-mode 0: BNF7...0 = number of bits – 1

PCM-mode 1: BNF7...0 = (number of bits – 2) / 2

PCM-mode 2: BNF7...0 = (number of bits – 4) / 4

PCM-mode 3: BNF7...0 = (number of bits – 2) / 2

The value programmed in PBNR is also used to check the PFS-period.

### 5.1.1.4.3 PCM-Offset Downstream Register (POFD)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7

bit 0

OFD9	OFD8	OFD7	OFD6	OFD5	OFD4	OFD3	OFD2
------	------	------	------	------	------	------	------

**OFD9...2** Offset Downstream bit 9...2.

These bits together with PCSR:OFD1...0 determine the offset of the PCM-frame in downstream direction. The following formulas apply for calculating the required register value. BND is the bit number in downstream direction marked by the rising internal PFS-edge. BPF denotes the actual number of bits constituting a frame.

PCM-mode 0: OFD9...2 = mod<sub>BPF</sub> (BND – 17 + BPF)  
PCSR:OFD1..0 = 0

PCM-mode 1,3: OFD9...1 = mod<sub>BPF</sub> (BND – 33 + BPF)  
PCSR: OFD0 = 0

PCM-mode 2: OFD9...0 = mod<sub>BPF</sub> (BND – 65 + BPF)

#### 5.1.1.4.4 PCM-Offset Upstream Register (POFU)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
OFU9	OFU8	OFU7	OFU6	OFU5	OFU4	OFU3	OFU2

##### OFU9...2 Offset Upstream bit 9...2.

These bits together with PCSR:OFU1...0 determine the offset of the PCM-frame in upstream direction. The following formulas apply for calculating the required register value. BNU is the bit number in upstream direction marked by the rising internal PFS-edge.

PCM-mode 0:  $OFU9...2 = mode_{BPF} (BNU + 23)$

$PCSR:OFU1...00 = 0$

PCM-mode 1, 3:  $OFU9...1 = mod_{BPF} (BNU + 47)$

$PCSR:OFU0 = 0$

PCM-mode 2:  $OFU9...0 = mod_{BPF} (BNU + 95)$

#### 5.1.1.4.5 PCM-Clock Shift Register; within the ELIC (PCSR)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
DRCS	OFD1	OFD0	DRE	ADSRO	OFU1	OFU0	URE

##### DRCS Double Rate Clock Shift.

0...the PCM-input and output data are not delayed

1...the PCM-input and output data are delayed by one PDC-clock cycle

##### OFD1...0 Offset Downstream bits 1...0, see POFD-register.

##### DRE Downstream Rising Edge.

0...the PCM-data is sampled with the falling edge of PDC

1...the PCM-data is sampled with the rising edge of PDC

##### ADSRO Add Shift Register on Output.

0...the PCM-output data are not delayed

1...the PCM-output data are delayed by one PDC-clock cycle

*Note: If both DRCS and ADSRO are set to logical 1, the PCM-output data are delayed by two PDC-clock cycles.*

##### OFU1...0 Offset Upstream bits 1...0, see POFU-register.



**URE** Upstream Rising Edge.  
 0...the PCM-data is transmitted with the falling edge of PDC  
 1...the PCM-data is transmitted with the rising edge of PDC

*Note: If EMOD:ECMD2 is set to '0' some restrictions apply to the setting of PCSR (see chapter 5.1.1.3).*

**5.1.1.4.6 PCM-Input Comparison Mismatch (PICM)**

Access: read/write

Reset value: xx<sub>H</sub>

bit 7 bit 0

IPN	TSN6	TSN5	TSN4	TSN3	TSN2	TSN1	TSN0
-----	------	------	------	------	------	------	------

**IPN** Input Pair Number.  
 This bit denotes the pair of ports, where a bit mismatch occurred.  
 0...mismatch between ports 0 and 1  
 1...mismatch between ports 2 and 3

**TSN6...0** Time-Slot Number.  
 When a bit mismatch occurred these bits identify the affected bit position.

**Table 5-31**

PCM-Mode	Time-Slot Identification	Bit Identification
0	TSN6...TSN2 +2	TSN1, 0: 00 bits 6,7 01 bits 4,5 10 bits 2,3 11 bits 0,1
1, 3	TSN6...TSN1 +4	TSN0: 0 bits 4...7 1 bits 0...3
2	TSN6...TSN0 +8	

**5.1.1.4.7 Configurable Interface Mode Register 1 (CMD1)**

Access: read/write

Reset value: 00<sub>H</sub>

bit 7 bit 0

CSS	CSM	CSP1	CSP0	CMD1	CMD0	CIS1	CIS0
-----	-----	------	------	------	------	------	------

**CSS** Clock Source Selection.  
 0...PDC and PFS are used as clock and framing source for the CFI. Clock

**Description of Registers**

and framing signals derived from these sources are output on DCL and FSC. 1...DCL and FSC are selected as clock and framing source for the CFI. If EMOD:ECMD2 is set to '0', then CSS has to be set to '0' (see **chapter 5.1.1.3**).

*Note: Both ELICs must be programmed in the same way.*

**CSM** CFI-Synchronization Mode.  
 The rising FSC-edge synchronizes the CFI-frame.  
 0...FSC is evaluated with every falling edge of DCL.  
 1...FSC is evaluated with every rising edge of DCL.

*Note: If CSS = 0 is selected, CSM and PMOD:PSM must be programmed identical.*

**CSP1...0** Clock Source Prescaler 1,0.  
 The clock source frequency is divided according to the following table to obtain the CFI-reference clock CRCL.

**Table 5-32**

<b>CSP1,0</b>	<b>Prescaler Divisor</b>
00	2
01	1.5
10	1
11	not allowed

**CMD1...0** CFI-Mode1,0.  
 Defines the actual number and configuration of the CFI-ports.

**Table 5-33**

<b>CMD 1...0</b>	<b>CFI-Mode</b>	<b>Number of Logical Ports</b>	<b>CFI-Data Rate [Mbit/s]</b>		<b>Min. Required CFI-Data Rate [kbit/s] Relative to PCM-Data Rate</b>	<b>Necessary Reference Clock (RCL)</b>	<b>DCL-Output Frequencies CMD1: CSS0 = 0</b>
			<b>min.</b>	<b>max.</b>			
00	0	4 DU (0...3)	128	2048	32N/3	2xDR	DR, 2xDR <sup>1)</sup>
01	1	2 DU (0...1)	128	4096	64N/3	DR	DR
10	2	1 DU	128	8192	64N/3	0.5xDR	DR

Description of Registers

**Table 5-33** (cont'd)

CMD 1...0	CFI- Mode	Number of Logical Ports	CFI-Data Rate [Mbit/s]		Min. Required CFI-Data Rate [kbit/s] Relative to PCM-Data Rate	Necessary Reference Clock (RCL)	DCL-Output Frequencies CMD1: CSS0 = 0
			min.	max.			
11	3	8 bit (0...7)	128	1024	16N/3	4xDR	DR, 2xDR

<sup>1)</sup> In CFI-mode 0 data rate of 2.048 Mbit/s can be used with a 2.048-Mbit/s PDC-input clock, if EMOD:ECMD2 = '0'. Refer to **chapter 5.1.1.3 ELIC-Mode Register (EMOD)**.

where N = number of time-slots in a PCM-frame

**CIS1...0**

CFI Alternative Input Selection.

In CFI-mode 1 and 2 CIS1...0 controls the assignment between logical and physical receive pins. In CFI-mode 0 and 3 CIS1,0 should be set to '0'.

**Table 5-34**

CFI- Mode	Port 0		Port 1		Port 2		Port 3	
	DU0	DD0	DU1	DD1	DU2	DD2	DU3	DD3
0	IN0	OUT0	IN1	OUT1	IN2	OUT2	IN3	OUT3
1	IN0 CIS0 = 0	OUT0	IN1 CIS1 = 0	OUT1	IN0 CIS0 = 1	tri- state	IN1 CIS1 = 1	tri- state
2	IN CIS0 = 0	OUT	not active	tri- state	IN CIS0 = 1	tri- state	not active	tri- state
3	This mode is not allowed in the DOC..							

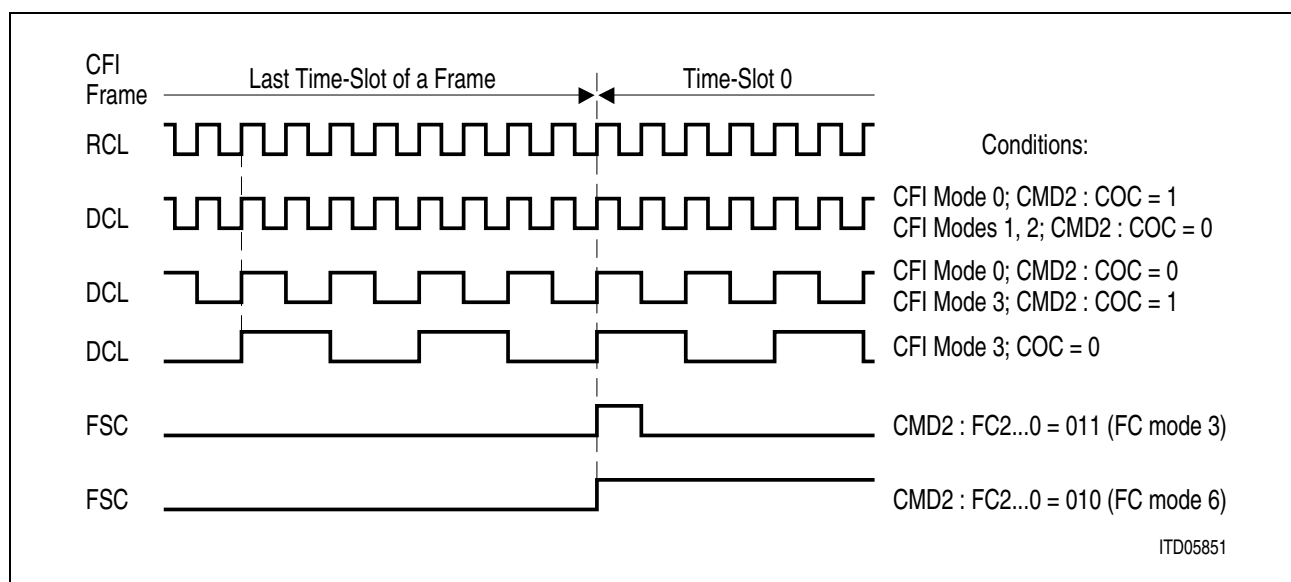
### 5.1.1.4.8 Configurable Interface Mode Register 2 (CMD2)

Access: read/write

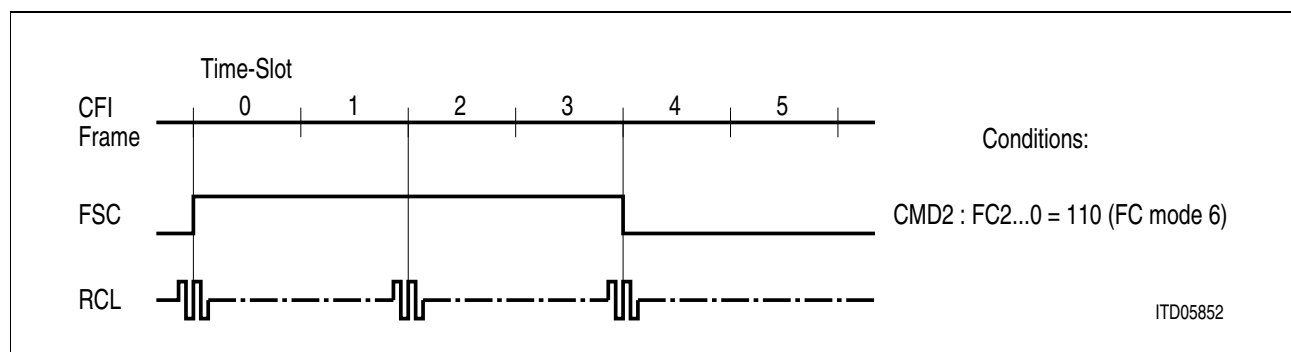
Reset value: 00<sub>H</sub>

bit 7				bit 0			
FC2	FC1	FC0	COC	0	0	CBN9	CBN8

**FC2...0** Framing output Control.  
 Given that CMD1:CSS = 0, these bits determine the position of the FSC-pulse relative to the CFI-frame, as well as the type of FSC-pulse generated. The position and width of the FSC-signal with respect to the CFI-frame can be found in the following two **Figure 5-2** and **Figure 5-3**.



**Figure 5-2** Position of the FSC-Signal for FC-Modes 3 and 6



**Figure 5-3** Position of the FSC-Signal for FC-Mode 6

Application examples:

**Table 5-35**

FC2	FC1	FC0	FC-Mode	Main Applications
0	0	0	0	not allowed
0	0	1	1	not allowed
0	1	0	2	not allowed
0	1	1	3	general purpose
1	0	0	4	not allowed
1	0	1	5	reserved
1	1	0	6	IOM-2
1	1	1	7	software timed multiplexed applications

For further details on the framing output control please refer to **chapter 5.1.1.4.8**.

**COC** CFI-Output Clock rate.  
 0...the frequency of DCL is identical to the CFI-data rate (all CFI-modes),  
 1...the frequency of DCL is twice the CFI-data rate (CFI-modes 0 and 3 only!)

*Note:* 1) *Applies only if CMD1:CSS = 0.*  
*If EMOD:ECMD2 is set to '0' then CMD2:COC must be set to '0' (see chapter 5.1.1.3).*  
 2) *CRR must be set to 0 in CFI-mode 3.*

**CBN9...8** CFI-Bit Number 9...8  
 these bits, together with the CBNR:CBN7...0, hold the number of bits per CFI-frame.

**5.1.1.4.9 Configurable Interface Bit Number Register (CBNR)**

Access: read/write

Reset value: FF<sub>H</sub>

bit 7						bit 0	
CBN7	CBN6	CBN5	CBN4	CBN3	CBN2	CBN1	CBN0

**CBN7...0** CFI-Bit Number 7...0.  
 The number of bits that constitute a CFI-frame must be programmed to CMD2, CBNR:CBN9...0 as indicated below.

CBN9...0 = number of bits – 1

For a 8-kHz frame structure, the number of bits per frame can be derived from the data rate by division with 8000.

### 5.1.1.4.10 Configurable Interface Time-Slot Adjustment Register (CTAR)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
0	TSN6	TSN5	TSN4	TSN3	TSN2	TSN1	TSN0

**TSN6...0** Time-Slot Number.  
 The CFI-framing signal (PFS if CMD1:CSS = 0 or FSC if CMD1:CSS = 1) marks the CFI time-slot called TSN according to the following formula:  
 $TSN6...0 = TSN + 2$   
 E.g.: If the framing signal is to mark time-slot 0 (bit 7), CTAR must be set to 02<sub>H</sub> (CBSR to 20<sub>H</sub>).

*Note: If CMD1:CSS = 0, the CFI-frame will be shifted - together with the FSC-output signal - with respect to PFS. The position of the CFI-frame relative to the FSC-output signal is not affected by these settings, but is instead determined by CMD2:FC2..0. If CMD1:CSS = 1, the CFI-frame will be shifted with respect to the FSC-input signal.*

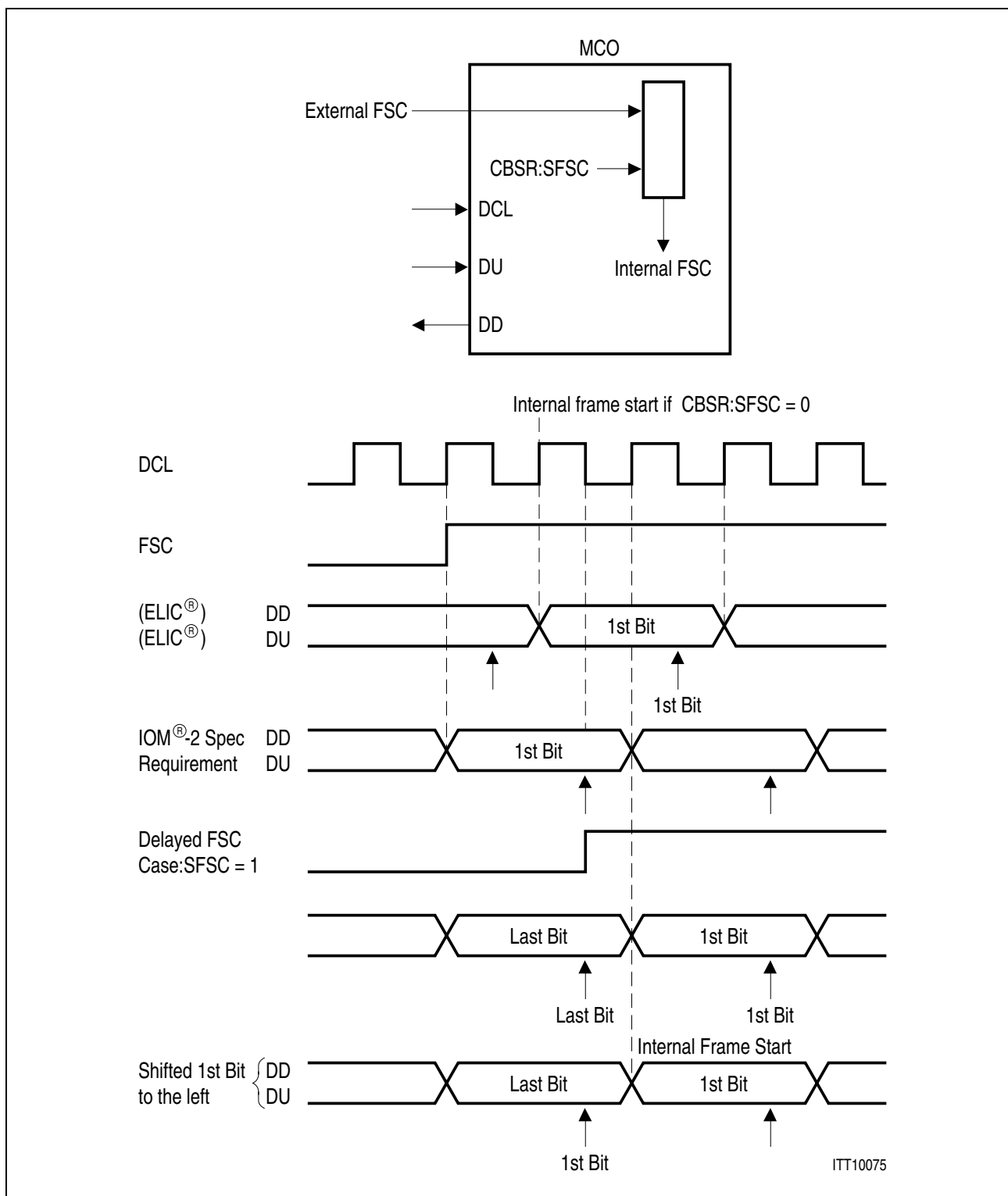
### 5.1.1.4.11 Configurable Interface Bit Shift Register (CBSR)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
SFSC	CDS2	CDS1	CDS0	CUS3	CUS2	CUS1	CUS0

**SFSC** Shift FSC  
 0...default (behaviour like EPIC-1 PEB 2055)  
 1...with double clock rate the FSC input is delayed by one and a half CFI clock cycle (IOM-2 compatibility)



**Figure 5-4 Internal FSC Shift to enable a Synchronization with the Rising Edge of DCL**

Description of Registers

**CDS2...0** CFI-Downstream bit Shift 2...0.  
 From the zero offset bit position (CBSR = 20<sub>H</sub>) the CFI-frame (downstream and upstream) can be shifted by up to 6 bits to the left (within the time-slot number TSN programmed in CTAR) and by up to 2 bits to the right (within the previous time-slot TSN – 1) by programming the CBSR:CDS2...0 bits:

**Table 5-36**

CBSR:CDS2...0	Time-Slot No.	Bit No.
000	TSN – 1	1
001	TSN – 1	0
010	TSN	7
011	TSN	6
100	TSN	5
101	TSN	4
110	TSN	3
111	TSN	2

The bit shift programmed to CBSR:CDS2...0 affects both the upstream and downstream frame position in the same way.

**CUS3...2** CFI-Upstream bit Shift 3...0.  
 These bits shift the upstream CFI-frame relative to the downstream frame by up to 15 bits. For CUS3...0 = 0000, the upstream frame is aligned with the downstream frame (no bit shift).

**5.1.1.4.12 Configurable Interface Subchannel Register (CSCR)**

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
SC31	SC30	SC21	SC20	SC11	SC10	SC01	SC00

**SC#1...#0** CFI-Subchannel Control for logical port #.  
 The subchannel control bits SC#1...SC#0 specify separately for each logical port the bit positions to be exchanged with the data memory (DM) when a connection with a channel bandwidth as defined by the CM-code has been established:



**Table 5-37**

SC#1	SC#0	Bit Positions for CFI Subchannels having a Bandwidth of		
		64 kbit/s	32 kbit/s	16 kbit/s
0	0	7...0	7...4	7...6
0	1	7...0	3...0	5...4
1	0	7...0	7...4	3...2
1	1	7...0	3...0	1...0

Note: In CFI-mode 1: SC21 = SC01; SC20 = SC00; SC31 = SC11; SC30 = SC10  
 In CFI-mode 2: SC31 = SC21 = SC11 = SC01; SC30 = SC20 = SC10 = SC00

#### 5.1.1.4.13 Memory Access Control Register (MACR)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RWS	MOC3	MOC2	MOC1	MOC0 CMC3	CMC2	CMC1	CMC0
-----	------	------	------	--------------	------	------	------

With the MACR the  $\mu$ P selects the type of memory (CM or DM), the type of field (data or code) and the access mode (read or write) of the register access. When writing to the control memory code field, MACR also contain the 4 bit code (CMC3..0) defining the function of the addressed CFI time-slot.

**RWS** Read/Write Select.  
 0...write operation on control or data memories  
 1...read operation on control or data memories

**MOC3...0** Memory Operation Code.

**CMC3...0** Control Memory Code.

These bits determine the type and destination of the memory operation as shown below.

Note: Prior to a new access to any memory location (i.e. writing to MACR) the STAR:MAC bit must be polled for '0'.

Description of Registers

**1. Writing data to the upstream DM-data field (e.g. PCM-idle code).  
Reading data from the upstream or downstream DM-data field.**

MACR:

RWS	MOC3	MOC2	MOC1	MOC0	0	0	0
-----	------	------	------	------	---	---	---

**MOC3...0** defines the bandwidth and the position of the subchannel as shown below:

**Table 5-38**

MOC3...0	Transferred Bits	Channel Bandwidth
0000	–	–
0001	bits 7...0	64 kbit/s
0011	bits 7...4	32 kbit/s
0010	bits 3...0	32 kbit/s
0111	bits 7...6	16 kbit/s
0110	bits 5...4	16 kbit/s
0101	bits 3...2	16 kbit/s
0100	bits 1...0	16 kbit/s

*Note: When reading a DM-data field location, all 8 bits are read regardless of the bandwidth selected by the MOC-bits.*

**2. Writing to the upstream DM-code (tri-state) field.  
Control-reading the upstream DM-code (tri-state).**

MACR:

RWS	MOC3	MOC2	MOC1	MOC0	0	0	0
-----	------	------	------	------	---	---	---

MOC = 1100 Read/write tri-state info from/to single PCM time-slot

MOC = 1101 Write tri-state info to all PCM time-slots

*Note: The tri-state field is exchanged with the 4 least significant bits (LSBs) of the MADR.*

**3. Writing data to the upstream or downstream CM-data field (e.g. signaling code).  
Reading data from the upstream or downstream CM-data field.**

MACR:

RWS	1	0	0	1	0	0	0
-----	---	---	---	---	---	---	---

Description of Registers

**4. Writing data to the upstream or downstream CM-data field and code field (e.g. switching a CFI to/from PCM-connection).**

MACR:

0	1	1	1	CMC3	CMC2	CMC1	CMC0
---	---	---	---	------	------	------	------

The 4-bit code field of the control memory (CM) defines the functionality of a CFI time-slot and thus the meaning of the corresponding data field. This 4-bit code, written to the MACR:CMC3...0 bit positions, will be transferred to the CM-code field. The 8-bit MADR value is at the same time transferred to the CM-data field. There are codes for switching applications, pre-processed applications and for direct  $\mu$ P-access applications, as shown below:

**a) Switching Applications**

CMC = 0000	Unassigned channel (e.g. cancelling an assigned channel)
CMC = 0001	Bandwidth 64 kbit/s PCM time-slot bits transferred: 7...0
CMC = 0010	Bandwidth 32 kbit/s PCM time-slot bits transferred: 3...0
CMC = 0011	Bandwidth 32 kbit/s PCM time-slot bits transferred: 7...4
CMC = 0100	Bandwidth 16 kbit/s PCM time-slot bits transferred: 1...0
CMC = 0101	Bandwidth 16 kbit/s PCM time-slot bits transferred: 3...2
CMC = 0110	Bandwidth 16 kbit/s PCM time-slot bits transferred: 5...4
CMC = 0111	Bandwidth 16 kbit/s PCM time-slot bits transferred: 7...6

*Note: The corresponding CFI time-slot bits to be transferred are chosen in the CSCR-register.*

**b) Pre-processed Applications**

**Table 5-39 Downstream**

Application	Even CM Address	Odd CM Address
Decentral D-channel handling	CMC = 1000	CMC = 1011
Central D-channel handling	CMC = 1010	CMC = PCM-code for a 2-bit subtime-slot
6-bit Signaling (e.g. analog IOM)	CMC = 1010	CMC = 1011
8-bit Signaling (e.g. SLD)	CMC = 1010	CMC = 1011
D-Channel handling by SACCO-A with ELIC-arbiter	CMC = 1010	CMC = 1011

**Table 5-40 Upstream**

Application	Even CM Address	Odd CM Address
Decentral D-channel handling	CMC = 1000	CMC = 0000
Central D-channel handling	CMC = 1000	CMC = PCM-code for a 2-bit subtime-slot
6-bit Signaling (e.g. analog IOM)	CMC = 1010	CMC = 1010
8-bit Signaling (e.g. SLD)	CMC = 1011	CMC = 1011
All code combinations are also valid for ELIC-arbiter operation.		

**c)  $\mu$ P-access Applications**

MACR:

0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

Setting CMC = 1001, initializes the corresponding CFI time-slot to be accessed by the  $\mu$ P. Concurrently, the datum in MADR is written (as 8-bit CFI-idle code) to the CM-data field. The content of the CM-data field is directly exchanged with the corresponding time-slot.

*Note: That once the CM-code field has been initialized, the CM-data field can be written and read as described in **chapter 3**.*

**5. Control-reading the upstream or downstream CM-code.**

MACR:

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

The CM-code can then be read out of the 4 LSBs of the MADR-register.

**5.1.1.4.14 Memory Access Address Register (MAAR)**

Access: read/write

Reset value: xx<sub>H</sub>

bit 7	bit 0						
U/ $\bar{D}$	MA6	MA5	MA4	MA3	MA2	MA1	MA0

The Memory Access Address Register MAAR specifies the address of the memory access. This address encodes a CFI time-slot for control memory (CM) and a PCM time-slot for data memory (DM) accesses. Bit 7 of MAAR (U/ $\bar{D}$ -bit) selects between upstream and downstream memory blocks. Bits MA6...0 encode the CFI- or PCM-port and time-slot number as in the following tables:

**Table 5-41 Time-Slot Encoding for Data Memory Accesses**

Data Memory Address		
PCM-mode 0	bit U/ $\bar{D}$ bits MA6...MA3, MA0 bits MA2...MA1	direction selection time-slot selection logical PCM-port number
PCM-mode 1,3	bit U/ $\bar{D}$ bits MA6...MA3, MA1, MA0 bit MA2	direction selection time-slot selection logical PCM-port number
PCM-mode 2	bit U/ $\bar{D}$ bits MA6...MA0	direction selection time-slot selection

**Table 5-42 Time-Slot Encoding for Control Memory Accesses**

Control Memory Address		
CFI-mode 0	bit U/ $\bar{D}$ bits MA6...MA3, MA0 bits MA2...MA1	direction selection time-slot selection logical CFI-port number
CFI-mode 1	bit U/ $\bar{D}$ bits MA6...MA3, MA2, MA0 bit MA1	direction selection time-slot selection logical CFI-port number
CFI-mode 2	bit U/ $\bar{D}$ bits MA6...MA0	direction selection time-slot selection
CFI-mode 3	bit U/ $\bar{D}$ bits MA6...MA4, MA0 bits MA3...MA1	direction selection time-slot selection logical CFI-port number

**5.1.1.4.15 Memory Access Data Register (MADR)**

Access: read/write

Reset value:  $xx_H$

bit 7							bit 0
MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

The Memory Access Data Register MADR contains the data to be transferred from or to a memory location. The meaning and the structure of this data depends on the kind of memory being accessed.

**5.1.1.4.16 Synchronous Transfer Data Register (STDA)**

Access: read/write

Reset value:  $xx_H$

bit 7							bit 0
MTDA7	MTDA6	MTDA5	MTDA4	MTDA3	MTDA2	MTDA1	MTDA0

The STDA-register buffers the data transferred over the synchronous transfer channel A. MTDA7 to MTDA0 hold the bits 7 to 0 of the respective time-slot. MTDA7 (MSB) is the bit transmitted/received first, MTDA0 (LSB) the bit transmitted/received last over the serial interface.

**5.1.1.4.17 Synchronous Transfer Data Register B (STDB)**

Access: read/write

Reset value:  $xx_H$

bit 7							bit 0
MTDB7	MTDB6	MTDB5	MTDB4	MTDB3	MTDB2	MTDB1	MTDB0

The STDA-register buffers the data transferred over the synchronous transfer channel A. MTDA7 to MTDA0 hold the bits 7 to 0 of the respective time-slot. MTDA7 (MSB) is the bit transmitted/received first, MTDA0 (LSB) the bit transmitted/received last over the serial interface.

#### 5.1.1.4.18 Synchronous Transfer Receive Address Register A (SARA)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
ISRA	MTRA6	MTRA5	MTRA4	MTRA3	MTRA2	MTRA1	MTRA0

The SARA-register specifies for synchronous transfer channel A from which input interface, port and time-slot the serial data is extracted. This data can then be read from the STDA-register.

- ISRA** Interface Select Receive for channel A.  
 0...selects the PCM-interface as the input interface for synchronous channel A.  
 1...selects the CFI-interface as the input interface for synchronous channel A.

**MTRA6...0**  $\mu$ P-Transfer Receive Address for channel A; selects the port and time-slot number at the interface selected by ISRA according to **Table 2-7** and **Table 2-8**: MTRA6...0 = MA6...0.

#### 5.1.1.4.19 Synchronous Transfer Receive Address Register B (SARB)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
ISRB	MTRB6	MTRB5	MTRB4	MTRB3	MTRB2	MTRB1	MTRB0

The SARB-register specifies for synchronous transfer channel B from which input interface, port and time-slot the serial data is extracted. This data can then be read from the STDB register.

- ISRB** Interface Select Receive for channel B.  
 0...selects the PCM-interface as the input interface for synchronous channel B.  
 1...selects the CFI-interface as the input interface for synchronous channel B.

**MTRB6...0**  $\mu$ P-Transfer Receive Address for channel B; selects the port and time-slot number at the interface selected by ISRB according to **Table 2-7** and **Table 2-8**: MTRB6...0 = MA6...0.

### 5.1.1.4.20 Synchronous Transfer Transmit Address Register A (SAXA)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7							bit 0
ISXA	MTXA6	MTXA5	MTXA4	MTXA3	MTXA2	MTXA1	MTXA0

The SAXA-register specifies for synchronous transfer channel A to which output interface, port and time-slot the serial data contained in the STDA-register is sent.

- ISXA** Interface Select Transmit for channel A.  
 0...selects the PCM-interface as the output interface for synchronous channel A.  
 1...selects the CFI-interface as the output interface for synchronous channel A.

**MTXA6...0** μP-Transfer Transmit Address for channel A; selects the port and time-slot number at the interface selected by ISXA according to **Table 2-7** and **Table 2-8**: MTXA6...0 = MA6...0.

### 5.1.1.4.21 Synchronous Transfer Transmit Address Register B (SAXB)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7							bit 0
ISXB	MTXB6	MTXB5	MTXB4	MTXB3	MTXB2	MTXB1	MTXB0

The SAXB-register specifies for synchronous transfer channel B to which output interface, port and time-slot the serial data contained in the STDB-register is sent.

- ISXB** Interface Select Transmit for channel B.  
 0...selects the PCM-interface as the output interface for synchronous channel B.  
 1...selects the CFI-interface as the output interface for synchronous channel B.

**MTXB6...0** μP-Transfer Transmit Address for channel B; selects the port and time-slot number at the interface selected by ISXB according to **Table 2-7** and **Table 2-8**: MTXB6...0 = MA6...0.



### 5.1.1.4.22 Synchronous Transfer Control Register (STCR)

Access: read/write

Reset value: 00xxxxxx<sub>B</sub>

bit 7				bit 0			
TBE	TAE	CTB2	CTB1	CTB0	CTA2	CTA1	CTA0

The STCR-register bits are used to enable or disable the synchronous transfer utility and to determine the subtime slot bandwidth and position if a PCM-interface time-slot is involved.

**TAE, TBE** Transfer Channel A (B) Enable.

1...enables the  $\mu$ P transfer of the corresponding channel.

0...disables the  $\mu$ P transfer of the corresponding channel.

**CTA2...0** Channel Type A (B); these bits determine the bandwidth of the channel and the position of the relevant bits in the time-slot according to the table below.

**CTB2...0** Note that if a CFI time-slot is selected as receive or transmit time-slot of the synchronous transfer, the 64-kbit/s bandwidth must be selected (CT#2...CT#0 = 001).

**Table 5-43**

CT#2	CT#1	CT#0	Bandwidth	Transferred Bits
0	0	0	not allowed	–
0	0	1	64 kbit/s	bits 7...0
0	1	0	32 kbit/s	bits 3...0
0	1	1	32 kbit/s	bits 7...4
1	0	0	16 kbit/s	bits 1...0
1	0	1	16 kbit/s	bits 3...2
1	1	0	16 kbit/s	bits 5...6
1	1	1	16 kbit/s	

### 5.1.1.4.23 MF-Channel Active Indication Register (MFAIR)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7	0	SO	SAD5	SAD4	SAD3	SAD2	SAD1	bit 0	SAD0
-------	---	----	------	------	------	------	------	-------	------

This register is only used in IOM-2 applications (active handshake protocol) in order to identify active monitor channels when the “Search for active monitor channels” command (CMDR:MFSO) has been executed.

**SO** MF Channel Search On.  
 0...the search is completed.  
 1...the EPIC-1 is still busy looking for an active channel.

**SAD5...0** Subscriber Address 5...0; after an ISTA:MAC-interrupt these bits point to the port and time-slot where an active channel has been found. The coding is identical to MFSAR:SAD5...SAD0.

### 5.1.1.4.24 MF-Channel Subscriber Address Register (MFSAR)

Access: read/write

Reset value: xx<sub>H</sub>

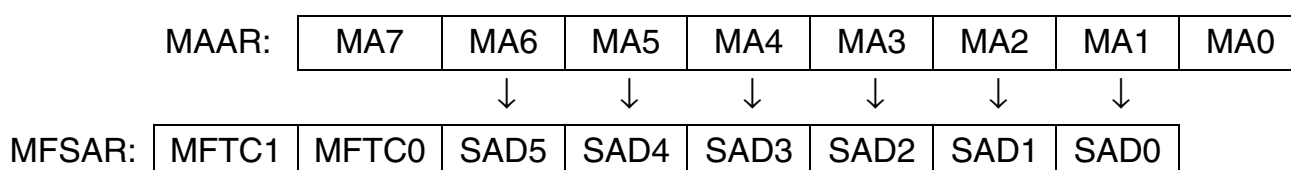
bit 7	MFTC1	MFTC0	SAD5	SAD4	SAD3	SAD2	SAD1	bit 0	SAD0
-------	-------	-------	------	------	------	------	------	-------	------

The exchange of monitor data normally takes place with only one subscriber circuit at a time. This register serves to point the MF-handler to that particular CFI time-slot.

**MFTC1...0** MF Channel Transfer Control 1...0; these bits, in addition to CMDR:MFT1,0 and OMDR:MFPS control the MF-channel transfer as indicated in **Table 5-44**.

**SAD5...0** Subscriber Address 5..0; these bits define the addressed subscriber. The CFI time-slot encoding is similar to the one used for Control Memory accesses using the MAAR-register (**Table 5-41 and Table 5-42**):

CFI time-slot encoding of MFSAR derived from MAAR:



Description of Registers

MAAR:MA7 selects between upstream and downstream CM-blocks. This information is not required since the transfer direction is defined by CMDR (transmit or receive).

MAAR:MA0 selects between even and odd time-slots. This information is also not required since MF-channels are always located on even time-slots.

**5.1.1.4.25 Monitor/Feature Control Channel FIFO (MFFIFO)**

Access: read/write  
Reset value: empty

bit 7							bit 0
MFD7	MFD6	MFD5	MFD4	MFD3	MFD2	MFD1	MFD0

The 16-byte bi-directional MFFIFO provides intermediate storage for data bytes to be transmitted or received over the monitor or feature control channel.

**MFD7...0** MF Data bits 7...0; MFD7 (MSB) is the first bit to be sent over the serial CFI, MFD0 (LSB) the last.

*Note: The byte n + 1 of an n-byte transmit message in monitor channel is not defined.*

**5.1.1.4.26 Signaling FIFO (CIFIFO)**

Access: read  
Reset value: 0xxxxxxx<sub>B</sub>

bit 7							bit 0
SBV	SAD6	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0

The 9 byte deep CIFIFO stores the addresses of CFI time-slots in which a C/I- and/or a SIG-value change has taken place. This address information can then be used to read the actual C/I- or SIG-value from the control memory.

**SBV** Signaling Byte Valid.  
0...the SAD6..0 bits are invalid.  
1...the SAD6..0 bits indicate a valid subscriber address. The polarity of SBV is chosen such that the whole 8 bits of the CIFIFO can be copied to the MAAR register in order to read the upstream C/I- or SIG-value from the control memory.

**SAD6...0** Subscriber Address bits 6...0; The CM-address which corresponds to the CFI time-slot where a C/I- or SIG-value change has taken place is encoded in these bits. For C/I-channels SAD6...0 point to an even CM-address (C/I-value), for SIG-channels SAD6...0 point to an odd CM-address (stable SIG-value).

### 5.1.1.4.27 Timer Register (TIMR)

Access: write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
SSR	TVAL6	TVAL5	TVAL4	TVAL3	TVAL2	TVAL2	TVAL0

The EPIC-1 timer can be used for 3 different purposes: timer interrupt generation (ISTA:TIG), FSC multiframe generation (CMD2:FC2...0 = 111) and last look period generation.

**SSR** Signaling Sampling Rate.  
 0...the last look period is defined by TVAL6...0.  
 1...the last look period is fixed to 125 μs.

**TVAL6...0** Timer Value bits 6...0; the timer period, equal to  $(1+TVAL6...0) \times 250 \mu s$ , is programmed here. It can thus be adjusted within the range of 250 μs up to 32 ms.

The timer is started as soon as CMDR:ST is set to 1 and stopped by writing the TIMR-register or by selecting OMDR:OMS0 = 0.

### 5.1.1.4.28 Status Register EPIC<sup>®</sup>-1 (STAR\_E)

Access: read

Reset value: 05<sub>H</sub>

bit 7						bit 0	
MAC	TAC	PSS	MFTO	MFAB	MFAE	MFRW	MFFE

The status register STAR displays the current state of certain events within the EPIC-1. The STAR register bits do not generate interrupts and are not modified by reading STAR.

**MAC** Memory Access  
 0...no memory access is in operation.  
 1...a memory access is in operation. Hence, the memory access registers may not be used.

*Note: MAC is also set and reset during synchronous transfers.*

**TAC** Timer Active  
 0...the timer is stopped.  
 1...the timer is running.

**PSS** PCM-Synchronization Status.  
 1...the PCM-interface is synchronized.  
 0...the PCM-interface is not synchronized. There is a mismatch between the

Description of Registers

- PBNR-value and the applied clock and framing signals (PDC/PFS) or OMDR:OMS0 = 0.
- MFTO** MF-Channel Transfer in Operation.  
0...no MF-channel transfer is in operation.  
1...an MF-channel transfer is in operation.
  - MFAB** MF-Channel Transfer Aborted.  
0...the remote receiver did not abort a handshake message transfer.  
1...the remote receiver aborted a handshake message transfer.
  - MFAE** MFFIFO-Access Enable.  
0...the MFFIFO may not be accessed.  
1...the MFFIFO may be either read or written to.
  - MFRW** MFFIFO Read/Write.  
0...the MFFIFO is ready to be written to.  
1...the MFFIFO may be read.
  - MFFE** MFFIFO Empty  
0...the MFFIFO is not empty.  
1...the MFFIFO is empty.

**5.1.1.4.29 Command Register EPIC®-1 (CMDR\_E)**

Access: write  
Reset value: 00<sub>H</sub>

bit 7							bit 0
0	ST	TIG	CFR	MFT1	MFT0	MFSO	MFFR

Writing a logical 1 to a CMDR-register bit starts the respective operation.

- ST** Start Timer.  
0...not action. If the timer shall be stopped, the TIMR-register must simply be written with a random value.  
1...starts the timer to run cyclically from 0 to the value programmed in TIMR:TVAL6...0.
- TIG** Timer Interrupt Generation.  
0...setting the TIG-bit to logical 0 together with the CMDR:ST-bit set to logical 1 disables the interrupt generation.  
1...setting the TIG-bit to logical 1 together with CMDR:ST-bit set to logical 1 causes the EPIC-1 to generate a periodic interrupt (ISTA:TIN) each time the timer expires.

Description of Registers

- CFR** CIFIFO Reset.  
 0...no action.  
 1...resets the signaling FIFO within 2 RCL-periods, i.e. all entries and the ISTA:SFI-bit are cleared.
- MFT1...0** MF-channel Transfer Control Bits 1,0; these bits start the monitor transfer enabling the contents of the MFFIFO to be exchanged with the subscriber circuits as specified in MFSAR. The function of some commands depends furthermore on the selected protocol (OMDR:MFPS). **Table 5-44** summarizes all available MF-commands.
- MFSO** MF-channel Search On.  
 0...no action.  
 1...the EPIC-1 starts to search for active MF-channels. Active channels are characterized by an active MX-bit (logical 0) sent by the remote transmitter. If such a channel is found, the corresponding address is stored in MFAIR and an ISTA:MAC-interrupt is generated. The search is stopped when an active MF-channel has been found or when OMDR:OMS0 is set to 0.
- MFFR** MFFIFO Reset.  
 0...no action  
 1...resets the MFFIFO and all operations associated with the MF-handler (except for the search function) within 2 RCL-periods. The MFFIFO is set into the state "MFFIFO empty", write access enabled and any monitor data transfer currently in process will be aborted.

**Table 5-44 Summary of MF-Channel Commands**

Transfer Mode	CMDR: MFT1,MFT0	MFSAR	Protocol Selection	Application
Inactive	00	xxxxxxx	HS, no HS	idle state
Transmit	01	00 SAD5...0	HS, no HS	IOM-2, IOM-1, SLD
Transmit broadcast	01	01xxxxxx	HS, no HS	IOM-2, IOM-1, SLD
Test operation	01	10-----	HS, no HS	IOM-2, IOM-1, SLD
Transmit continuous	11	00 SAD5...0	HS	IOM-2

**Table 5-44 Summary of MF-Channel Commands (cont'd)**

Transfer Mode	CMDR: MFT1,MFT0	MFSAR	Protocol Selection	Application
Transmit + receive same time-slot				
Any # of bytes	10	00 SAD5...0	HS	IOM-2
1 byte expected	10	00 SAD5...0	no HS	IOM-1
2 bytes expected	10	01 SAD5...0	no HS	(IOM-1)
8 bytes expected	10	10 SAD5...0	no HS	(IOM-1)
16 bytes expected	10	11 SAD5...0	no HS	(IOM-1)
Transmit + receive same line				
1 byte expected	11	00 SAD5...0	no HS	SLD
2 bytes expected	11	01 SAD5...0	no HS	SLD
8 bytes expected	11	10 SAD5...0	no HS	SLD
16 bytes expected	11	11 SAD5...0	no HS	SLD

HS: handshake facility enabled (OMDR:MFPS = 1)

no HS: handshake facility disable (OMDR:MFPS = 0)

#### 5.1.1.4.30 Interrupt Status Register EPIC®-1 (ISTA\_E)

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

TIN	SFI	MFFI	MAC	PFI	PIM	SIN	SOV
-----	-----	------	-----	-----	-----	-----	-----

The ISTA-register should be read after an interrupt in order to determine the interrupt source.

**TIN** Timer interrupt; a timer interrupt previously requested with CMDR:ST, TIG = 1 has occurred. The TIN-bit is reset by reading ISTA. It should be noted that the interrupt generation is periodic, i.e. unless stopped by writing to TIMR, the ISTA:TIN will be generated each time the timer expires.

**SFI** Signaling FIFO-Interrupt; this interrupt is generated if there is at least one valid entry in the CIFIFO indicating a change in a C/I- or SIG-channel. Reading ISTA does not clear the SFI-bit. Instead SFI is cleared if the CIFIFO is empty which can be accomplished by reading all valid entries of the CIFIFO or by resetting the CIFIFO by setting CMDR:CFR to 1.

**Description of Registers**

- MFFI** MFFIFO-Interrupt; the last MF-channel command (issued by CMDR:MFT1, MFT0) has been executed and the EPIC-1 is ready to accept the next command. Additional information can be read from STAR:MFT0...MFFE. MFFI is reset by reading ISTA.
- MAC** Monitor Channel Active interrupt; the EPIC-1 has found an active monitor channel. A new search can be started by reissuing the CMDR:MFSO-command. MAC is reset by reading ISTA.
- PFI** PCM-Framing Interrupt; the STAR:PSS-bit has changed its polarity. To determine whether the PCM-interface is synchronized or not, STAR must be read. The PFI-bit is reset by reading ISTA.
- PIM** PCM-Input Mismatch; this interrupt is generated immediately after the comparison logic has detected a mismatch between a pair of PCM-input lines. The exact reason for the interrupt can be determined by reading the PICM-register. Reading ISTA clears the PIM-bit. A new PIM-interrupt can only be generated after the PICM-register has been read.
- SIN** Synchronous transfer Interrupt; The SIN-interrupt is enabled if at least one synchronous transfer channel (A and/or B) is enabled via the STCR:TAE, TBE-bits. The SIN-interrupt is generated when the access window for the  $\mu$ P opens. After the occurrence of the SIN-interrupt the  $\mu$ P can read and/or write the synchronous transfer data registers (STDA, STDB). The SIN-bit is reset by reading ISTA.
- SOV** Synchronous transfer Overflow; The SOV-interrupt is generated if the  $\mu$ P fails to access the data registers (STDA, STDB) within the access window. The SOV-bit is reset by reading ISTA.

**5.1.1.4.31 Mask Register EPIC®-1 (MASK\_E)**

Access: write

Reset value: 00<sub>H</sub>

bit 7								bit 0
TIN	SFI	MFFI	MAC	PFI	PIM	SIN	SOV	

A logical 1 disables the corresponding interrupt as described in the ISTA-register.

A masked interrupt is stored internally and reported in ISTA immediately if the mask is released. However, an SFI-interrupt is also reported in ISTA if masked. In this case no interrupt is generated. When writing register MASK\_E while ISTA\_E indicates a non masked interrupt INT is temporarily set into the inactive state.



5.1.1.4.32 Operation Mode Register (OMDR)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7

bit 0

OMS1	OMS0	PSB	PTL	0 or 1	MFPS	CSB	RBS
------	------	-----	-----	--------	------	-----	-----

**OMS1...01** Operational Mode Selection; these bits determine the operation mode of the EPIC-1 is working in according to the following table:

**Table 5-45**

OMS1...0	Function
00	The <b>CM-reset mode</b> is used to reset all locations of the control memory code and data fields with a single command within only 256 RCL-cycles. A typical application is resetting the CM with the command MACR = 70 <sub>H</sub> which writes the contents of MADR (xx <sub>H</sub> ) to all data field locations and the code '0000' (unassigned channel) to all code field locations. A CM-reset should be made after each hardware reset. In the CM-reset mode the EPIC-1 does not operate normally i.e. the CFI- and PCM-interfaces are not operational.
10	The <b>CM-initialization mode</b> allows fast programming of the control memory since each memory access takes a maximum of only 2.5 RCL-cycles compared to the 9.5 RCL-cycles in the normal mode. Accesses are performed on individual addresses specified by MAAR. The initialization of control/signaling channels in IOM- applications can for example be carried out in this mode. In the CM- initialization mode the EPIC-1 does also not work normally.
11	In the <b>normal operation mode</b> the CFI- and PCM-interfaces are operational. Memory accesses performed on single addresses (specified by MAAR) take 9.5 RCL-cycles. An initialization of the complete data memory tri-state field takes 1035 RCL-cycles.
01	In <b>test mode</b> the EPIC-1 sustains normal operation. However memory accesses are no longer performed on a specific address defined by MAAR, but on all locations of the selected memory, the contents of MAAR (including the U/D-bit!) being ignored. A test mode access takes 2057 RCL-cycles.

Description of Registers

- PSB** PCM-Standby.  
 0...the PCM-interface output pins TxD0...3 are set to high impedance and those  $\overline{TSC}$ -pins that are actually used as tri-state control signals are set to logical 1 (inactive).  
 1...the PCM-output pins transmit the contents of the upstream data memory or may be set to high impedance via the data memory tri-state field.
- PTL** PCM-Test Loop.  
 0...the PCM-test loop is disabled.  
 1...the PCM-test loop is enabled, i.e. the physical transmit pins TxD# are internally connected to the corresponding physical receive pins RxD#, such that data transmitted over TxD# are internally looped back to RxD# and data externally received over RxD# are ignored. The TxD# pins still output the contents of the upstream data memory according to the setting of the tri-state field (only modes 0 and 1; mode 1 with AIS-bit set).
- COS** Don't care. The input driver is determined in register CCR1:ODS of SIDE0.
- MFPS** Monitor/Feature control channel Protocol Selection.  
 0...handshake facility disabled (SLD and IOM-1 applications)  
 1...handshake facility enabled (IOM-2 applications)
- CSB** CFI-Standby.  
 0...the CFI-interface output pins DD0...3, DU0...3, DCL and FSC are set to high impedance.  
 1...the CFI-output pins are active.
- RBS** Register Bank Selection. Used in demultiplexed data/address modes only. The RBS-bit is internally ORed with the A4 address pin. The EPIC-1 registers can therefore be accessed using two different methods:  
 1) If RBS is always set to logical 0, the registers can be accessed using all 5 address pins A4...A0.

### 5.1.1.4.33 Version Number Status Register (VNSR)

Access: write

Reset value: 0x<sub>H</sub>

bit 7							bit 0
IR	0	0	SWRX	X	X	X	X

The VNSR-register bits do not generate interrupts and are not modified by reading VNSR. The IR and VN3...0 bits are read only bits, the SWRX-bit is a write only bit.

**IR** Initialization Request; this bit is set to logical 1 after an inappropriate clocking or after a power failure. It is reset to logical 0 after a control memory reset command: OMDR:OMS1...0 = 00, MACR = 7X.

**SWRX** Software Reset External.  
When set, the pin RESIN is activated. RESIN is reset with the next EPIC-1 interrupt, i.e. the EPIC-1 timer may be used to generate a RESIN-pulse without generating an internal ELIC-reset.

**X** Don't care

### 5.1.1.5 SACCO-A Register Description

#### 5.1.1.5.1 Receive FIFO (RFIFO)

Access: read)

Reset value: xx<sub>H</sub>

bit 7							bit 0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

**RD7...0** Receive Data 7...0, data byte received on the serial interface.

#### Interrupt controlled data transfer

Up to 32 bytes of received data can be read from the RFIFO following an RPF or an RME interrupt.

RPF-interrupt: exactly 32 bytes to be read.

RME-interrupt: the number of bytes can be determined reading the registers RBCL, RBCH.

### 5.1.1.5.2 Transmit FIFO (XFIFO)

Access: write

Reset value: xx<sub>H</sub>

bit 7				bit 0			
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0

**TD7...0** Transmit Data 7...0, data byte to be transmitted on the serial interface. Interrupt controlled data transfer. Up to 32 bytes of transmit data can be written to the XFIFO following an XPR-interrupt.

### 5.1.1.5.3 Interrupt Status Register (ISTA\_A/B)

Access: read

Reset value: 00<sub>H</sub>

bit 7				bit 0			
RME	RPF	0	XPR	0	0	0	0

**RME** Receive Message End.  
A message of up to 32 bytes or the last part of a message greater than 32 bytes has been received and is now available in the RFIFO. The message is complete! The actual message length can be determined by reading the registers RBCL, RBCH. RME is not generated when an extended HDLC- frame is recognized in auto-mode (EHC interrupt).

**RPF** Receive Pool Full.  
A data block of 32 bytes is stored in the RFIFO. The message is not yet completed!

**XPR** Transmit Pool Ready.  
A data block of up to 32 bytes can be written to the XFIFO.

### 5.1.1.5.4 Mask Register (MASK\_A/B)

Access: write

Reset value: 00<sub>H</sub> (all interrupts enabled)

bit 7				bit 0			
RME	RPF	0	XPR	0	0	0	0

**RME** enables(0)/disables(1) the Receive Message End interrupt.

**RPF** enables(0)/disables(1) the Receive Pool Full interrupts.

**XPR** enables(0)/disables(1) the Transmit Pool Ready interrupt.

Each interrupt source can be selectively masked by setting the respective bit in the MASK\_A/B-register (bit position corresponding to the ISTA\_A/B-register). Masked interrupts are internally stored but not indicated when reading ISTA\_A/B and also not flagged into the top level ISTA. After releasing the respective MASK\_A/B-bit they will be indicated again in ISTA\_A/B and in the top level ISTA.

When writing register MASK\_A/B while ISTA\_A/B indicates a non masked interrupt the  $\overline{\text{INT}}$ -pin is temporarily set into the inactive state. In this case the interrupt remains indicated in the ISTA\_A/B until these registers are read.

### 5.1.1.5.5 Extended Interrupt Register (EXIR\_A/B)

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

XMR	XDU/EXE	EHC	RFO	0	RFS	0	0
-----	---------	-----	-----	---	-----	---	---

**XMR** Transmit Message Repeat.

The transmission of a frame has to be repeated because:

- A frame consisting of more than 32 bytes is polled a second time in auto-mode.
- Collision has occurred after sending the 32nd data byte of a message in a bus configuration.
- CTS (transmission enable) has been withdrawn after sending the 32nd data byte of a message in point-to-point configuration.

**XDU/EXE** Transmission Data Underrun/Extended transmission End.

The actual frame has been aborted with IDLE, because the XFIFO holds no further data, but the frame is not yet complete according to registers XBCH/XBCL.

In extended transparent mode, this bit indicates the transmission end condition.

*Note: It is not possible to transmit frames when a XMR- or XDU-interrupt is indicated.*

**EHC** Extended HDLC-frame.

The SACCO has received a frame in auto-mode which is neither a RR- nor an I-frame. The control byte is stored temporarily in the RHCR-register but not in the RFIFO.

Description of Registers

- RFO** Receive Frame Overflow.  
A frame could not be stored due to the occupied RFIFO (i.e. whole frame has been lost). This interrupt can be used for statistical purposes and indicates, that the CPU does not respond quickly enough to an incoming RPF- or RME-interrupt.
- RFS** Receive Frame Start.  
This is an early receiver interrupt activated after the start of a valid frame has been detected, i.e. after a valid address check in operation modes providing address recognition, otherwise after the opening flag (transparent mode 0), delayed by two bytes.  
After a RFS-interrupt the contents of
- RHCR
  - RAL1
  - RSTA bit3-0
- are valid and can be read by the CPU.  
The RFS-interrupt is maskable by programming bit CCR2:RIE.

**5.1.1.5.6 Command Register (CMDR)**

Access: write  
Reset value: 00<sub>H</sub>

bit 7							bit 0
RMC	RHR	XREP	0	XPD/ XTF	XDD	XME	XRES

*Note: The maximum time between writing to the CMDR-register and the execution of the command is 2.5 HDC-clock cycles. Therefore, if the CPU operates with a very high clock speed in comparison to the SACCO-clock, it is recommended that the bit STAR:CEC is checked before writing to the CMDR-register to avoid losing of commands.*

- RMC** Receive Message Complete.  
A '1' confirms, that the actual frame or data block has been fetched following a RPF- or RME-interrupt, thus the occupied space in the RFIFO can be released.
- RHR** Reset HDLC-Receiver.  
A '1' deletes all data in the RFIFO and in the HDLC-receiver.
- XREP** Extended transparent mode 0,1: XREP  
Together with XTF- and XME-set (CMDR = 2A<sub>H</sub>) the SACCO repeatedly transmits the contents of the XFIFO (1...32 bytes) fully transparent without HDLC-framing, i.e. without flag, CRC-insertion, bit stuffing.

**Description of Registers**

The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.

When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

*Note: MODE:CFT must be set to '0' when using cyclic transmission.*

**XPD/XTF** Transmit Prepared Data/Transmit Transparent Frame.

- Non-auto-mode, transparent mode 0,1: XTF  
The transmission of the XFIFO contents is started, an opening flag sequence is automatically added.
- Extended transparent mode 0,1: XTF  
The transmission of the XFIFO contents is started, no opening flag sequence is added.

**XME** Transmit Message End (interrupt mode only).  
A '1' indicate that the data block written last to the XFIFO completes the actual frame. The SACCO can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data. XME is used only in combination with XPD/XTF or XDD.

**XRES** Transmit Reset.  
The contents of the XFIFO is deleted and IDLE is transmitted. This command can be used by the CPU to abort a frame currently in transmission. After setting XRES a XPR-interrupt is generated in every case.

**5.1.1.5.7 Mode Register (MODE)**

Access: read/write

Reset value: 00<sup>H</sup>

bit 7

bit 0

MDS1	MDS0	ADM	CFT	RAC	0	0	TLP
------	------	-----	-----	-----	---	---	-----

**MDS1...0** Mode Select.  
The operating mode of the HDLC-controller is selected.  
01...non-auto-mode  
10...transparent mode (D-channel arbiter)  
11...extended transparent mode

**ADM** Address Mode. The meaning of this bit varies depending on the selected operating mode:

**Description of Registers**

- Transparent mode
  - 0...no address recognition: transparent mode 0 (D-channel arbiter)
  - 1...high byte address recognition: transparent mode 1
- Extended transparent mode
  - 0...receive data in RAL1: extended transparent mode 0
  - 1...receive data in RFIFO and RAL1: extended transparent mode 1

*Note: In extended transparent mode 0 and 1 the bit MODE:RAC must be reset to enable fully transparent reception.*

**CFT** Continuous Frame Transmission.  
 1...When CFT is set the XPR-interrupt is generated immediately after the CPU accessible part of XFIFO is copied into the transmitter section.  
 0...Otherwise the XPR-interrupt is delayed until the transmission is completed (D-channel arbiter).

**RAC** Receiver Active.  
 Via RAC the HDLC-receiver can be activated/deactivated.  
 0...HDLC-receiver inactive  
 1...HDLC-receiver active  
 In extended transparent mode 0 and 1 RAC must be reset (HDLC-receiver disabled) to enable fully transparent reception.

**TLP** Test Loop.  
 When set input and output of the HDLC-channel are internally connected. (transmitter channel A - receiver channel A  
 transmitter channel B - receiver channel B)  
 TXDA/B are active, RXDA/B are disabled.

**5.1.1.5.8 Channel Configuration Register 1 (CCR1)**

Access: read/write

Reset value: 00<sub>H</sub>

	bit 7						bit 0
	PU	0	0	0	0	CM2	1
							1

**PU** Power-Down Mode.  
 0...power-down (standby), the internal clock is switched off. Nevertheless, register read/write access is possible.  
 1...power-up (active).

**CM2** Clock rate  
 0...single clock  
 1...double clock



### 5.1.1.5.9 Channel Configuration Register 2 (CCR2)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
0	0	0	0	0	0	RIE	0

**RIE** Receive frame start Enable.  
When set, the RFS-interrupt in register EXIR\_A/B is enabled.

### 5.1.1.5.10 Receive Length Check Register (RLCR)

Access: write

Reset value: 0xxxxxxx<sub>H</sub>

bit 7						bit 0	
RC	RL6	RL5	RL4	RL3	RL2	RL1	RL0

**RC** Receive Check enable.  
A '1' enables, a '0' disables the receive frame length feature.

**RL6...0** Receive Length.  
The maximum receive length after which data reception is suspended can be programmed in RL6...0. The maximum allowed receive frame length is  $(RL + 1) \times 32$  bytes. A frame exceeding this length is treated as if it was aborted by the opposite station (RME-interrupt, RAB-bit set (VFR)). In this case the receive byte count (RBCH, RBCL) is greater than the programmed receive length.

### 5.1.1.5.11 Status Register (STAR)

Access: read

Reset value: 48<sub>H</sub>

bit 7						bit 0	
XDOV	XFW	AREP/ XREP	RFR	RLI	CEC	XAC	AFI

**XDOV** Transmit Data Overflow.  
A '1' indicates, that more than 32 bytes have been written into the XFIFO.

**XFW** XFIFO Write enable.  
A '1' indicates, that data can be written into the XFIFO.

*Note: XFW is only valid when CEC = 0.*

**Description of Registers**

- XREP**      Read back value of the corresponding command bit CMDR:XREP.
- RFR**      RFIFO Read enable.  
A '1' indicates, that valid data is in the RFIFO and read access is enabled.  
RFR is set with the RME- or RPF-interrupt and reset when executing the RMC-command.
- RLI**      Receiver Line Inactive.  
Neither flags as inter frame time fill nor frames are received via the receive line.

*Note: Significant in point-to-point configurations!*

- CEC**      Command Execution.  
When '0' no command is currently executed, the CMDR-register can be written to.  
When '1' a command (written previously to CMDR) is currently executed, no further command must temporarily be written to the CMDR-register.
- XAC**      Transmitter Active.  
A '1' indicates, that the transmitter is currently active.
- AFI**      Additional Frame Indication.  
A '1' indicates, that one or more completely received frames or the last part of a frame are in the CPU inaccessible part of the RFIFO.  
In combination with the bit STAR:RFR multiple frames can be read out of the RFIFO without interrupt control.

**5.1.1.5.12 Receive Status Register (RSTA)**

Access: read

Reset value: xx<sub>H</sub>

bit 7				bit 0			
VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA

RSTA always displays the momentary state of the receiver. Because this state can differ from the last entry in the FIFO it is reasonable to always use the status bytes in the FIFO.

- VFR**      Valid Frame.  
Indicates whether the received frame is valid ('1') or not ('0' invalid).  
A frame is invalid when
  - its length is not an integer multiple of 8 bits ( $n \times 8$  bits), e.g. 25 bit,
  - its is to short, depending on the selected operation mode:
    - transparent mode 1: 3 bytes
    - transparent mode 0: 2 bytes
  - a frame was aborted (note: VFR can also be set when a frame was aborted)

*Note: Shorter frames are not reported.*

- RDO** Receive Data Overflow.  
A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.
- CRC** CRC-Compare Check.  
0: CRC check failed, received frame contains errors.  
1: CRC check o.k., received frame is error free.
- RAB** Receive message Aborted.  
When '1' the received frame was aborted from the transmitting station. According to the HDLC-protocol, this frame must be discarded by the CPU.
- HA1...0** High byte Address compare.  
In operating modes which provide high byte address recognition, the SACCO compares the high byte of a 2-byte address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FEH and FCH (group address). Depending on the result of the comparison, the following bit combinations are possible:  
10...RAH1 has been recognized.  
00...RAH2 has been recognized.  
01...group address has been recognized.

*Note: If RAH1, RAH2 contain the identical value, the combination 00 will be omitted. HA1..0 is significant only in 2-byte address modes.*

- C/R** Command/Response; significant only, if 2-byte address mode has been selected. Value of the C/R bit (bit of high address byte) in the received frame.
- LA** Low byte Address compare.  
The low byte address of a 2-byte address field or the single address byte of a 1-byte address field is compared with two programmable registers (RAL1, RAL2). Depending on the result of the comparison LA is set.  
0...RAL2 has been recognized,  
1...RAL1 has been recognized.  
In non-auto mode, according to the X.25 LAP B-protocol, RAL1/RAL2 may be programmed to differ between COMMAND/RESPONSE frames.

*Note: A modified receive status byte is copied into the RFIFO following the last byte of the corresponding frame. So contains the IOM-port and channel address of the received frame. Please refer to **chapter 2.1.2.4.6** the RFIFO.*

### 5.1.1.5.13 Receive HDLC-Control Register (RHCR)

Access: read

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RHCR7	RHCR6	RHCR5	RHCR4	RHCR3	RHCR2	RHCR1	RHCR0

**RHCR7...0** Receive HDLC-Control Register.

The contents of the RHCR depends on the selected operating mode.

- Non-auto mode (1-byte address field): 2nd byte after flag
- Non-auto mode (2-byte address field): 3rd byte after flag
- Transparent mode 1: 3rd byte after flag
- Transparent mode 0: 2nd byte after flag

*Note: The value in RHCR corresponds to the last received frame.*

**5.1.1.5.14 Transmit Address Byte 1 (XAD1)**

Access: write

Reset value: xx<sub>H</sub>

bit 7							bit 0
XAD17	XAD16	XAD15	XAD14	XAD13	XAD12	XAD11	XAD10

**XAD17...10** Transmit Address byte 1.

The value stored in XAD1 is included automatically as the address byte (high address byte in case of 2-byte address field) of all frames transmitted in auto mode. Using a 2 byte address field, XAD11 and XAD10 have to be set to '0'.

**5.1.1.5.15 Transmit Address Byte 2 (XAD2)**

Access: write

Reset value: xx<sub>H</sub>

bit 7							bit 0
XAD27	XAD26	XAD25	XAD24	XAD23	XAD22	XAD21	XAD20

**XAD27...20** Transmit Address byte 2.

The value stored in XAD2 is included automatically as the low address byte of all frames transmitted in auto-mode (2-byte address field only).

**5.1.1.5.16 Receive Address Byte Low Register 1 (RAL1)**

Access: read/write

Reset value: xx<sub>H</sub>

bit 7							bit 0
RAL17	RAL16	RAL15	RAL14	RAL13	RAL12	RAL11	RAL10

**RAL17...10** Receive Address byte Low register 1.

The general function (read/write) and the meaning or contents of this register depends on the selected operating mode:

- Non-auto mode (address recognition) - write only:  
compare value 1, address recognition (low byte in case of 2-byte address field).
- Transparent mode 1 (high byte address recognition) - read only:  
RAL1 contains the byte following the high byte of the address in the received frame (i.e. the second byte after the opening flag).
- Transparent mode 0 (no address recognition) - read only:  
contains the first byte after the opening flag (first byte of the received frame).
- Extended transparent mode 0,1 - read only:  
RAL1 contains the actual data byte currently assembled at the RxD-pin by passing the HDLC-receiver (fully transparent reception without HDLC-framing).

*Note: In auto-mode and non-auto mode the read back of the programmed value is inverted.*

**5.1.1.5.17 Receive Address Byte Low Register 2 (RAL2)**

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RAL27	RAL26	RAL25	RAL24	RAL23	RAL22	RAL21	RAL20
-------	-------	-------	-------	-------	-------	-------	-------

**RAL27...20** Receive Address byte Low register 1.

- Non-auto mode (address recognition):  
compare value 2, address recognition (low byte in case of 2-byte address field).

*Note: Normally used for broadcast address.*

**5.1.1.5.18 Receive Address Byte High Register 1 (RAH1)**

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RAH17	RAH16	RAH15	RAH14	RAH13	RAH12	0	0
-------	-------	-------	-------	-------	-------	---	---

**RAL17...12** Receiver Address byte High register 1.

Description of Registers

- Non-auto mode transparent mode 1, (2-byte address field). Compare value 1, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH1 must be set to 00<sub>H</sub>.*

**5.1.1.5.19 Receive Address Byte High Register 2 (RAH2)**

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAH27	RAH26	RAH25	RAH24	RAH23	RAH22	0	0

**RAL27...22** Receiver Address byte High register 2.

- Non-auto mode transparent mode 1, (2-byte address field). Compare value 2, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH2 must be set to 00<sub>H</sub>.*

**5.1.1.5.20 Receive Byte Count Low (RBCL)**

Access: read

Reset value: 00<sub>H</sub>

bit 7						bit 0	
RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0

**RBC7...0** Receive Byte Count.

Together with RBCH (bits RBC11 - RBC8), the length of the actual received frame (0...4095 bytes) can be determined. These registers must be read by the CPU following a RME interrupt.

**5.1.1.5.21 Receive Byte Count High (RBCH)**

Access: read

Reset value: 000xxxxx<sub>H</sub>

bit 7				bit 0			
0	0	0	OV	RBC11	RBC10	RBC9	RBC8

**DMA**

DMA-mode status indication.

Read back value representing the DMA-bit programmed in register XBCH.

Description of Registers

**OV** Counter Overflow.  
 A '1' indicates that more than 4095 bytes were received.  
 The received frame exceeded the byte count in RBC11...RBC0.

**RBC11...8** Receive Byte Count high.  
 Together with RBCL (bits RBC7...RBC0) the length of the received frame can be determined.

**5.1.1.5.22 Version Status Register (VSTR)**

Access: read

bit 7				bit 0			
1	0	0	0	VN3	VN2	VN1	VN0

**VN3...0** 82<sub>H</sub>: SACCO-A in DOC V1.1  
 83<sub>H</sub>: SACCO-A in DOC V2.1.

**5.1.1.6 SACCO-B Register Description**

**5.1.1.6.1 Receive FIFO (RFIFO)**

Access: read

Reset value: xx<sub>H</sub>

bit 7				bit 0			
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

**RD7...0** Receive Data 7...0, data byte received on the serial interface.

**Interrupt controlled data transfer** (interrupt mode, selected if DMA-bit in register XBCH is reset).

Up to 32 bytes of received data can be read from the RFIFO following an RPF or an RME interrupt.

RPF-interrupt: exactly 32 bytes to be read.

RME-interrupt: the number of bytes can be determined reading the registers RBCL, RBCH.

**DMA controlled data transfer** (DMA-mode, selected if DMA-bit in register XBCH is set).

If the RFIFO contains 32 bytes, the SACCO autonomously requests a block data transfer by activating the DRQRA/B-line as long as the 31st read cycle is finished. This forces the DMA-controller to continuously perform bus cycles until 32 bytes are transferred from the SACCO to the system memory (DMA-controller mode: demand transfer, level triggered).

**Description of Registers**

If the RFIFO contains less than 32 bytes (one short frame or the last bytes of a long frame) the SACCO requests a block data transfer depending on the contents of the RFIFO according to the following table:

**Table 5-46**

<b>RFIFO Contents (bytes)</b>	<b>DMA Transfers (bytes)</b>
(1), 2, 3	4
4 - 7	8
8 - 15	16
16 - 32	32

Additionally an RME-interrupt is issued after the last byte has been transferred. As a result, the DMA-controller may transfer more bytes as actually valid in the current received frame. The valid byte count must therefore be determined reading the registers RBCH, RBCL following the RME-interrupt.

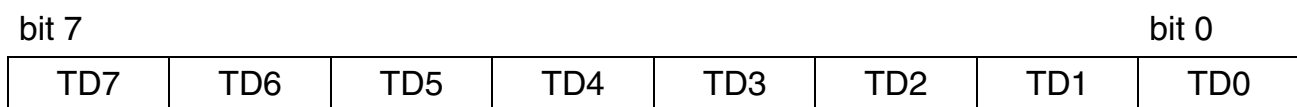
The corresponding DRQRA/B pin remains “high” as long as the RFIFO requires data transfers. It is deactivated upon the rising edge of the 31st DMA-transfer or, if  $n < 32$  or  $n$  is the remainder of a long frame, upon the falling edge of the last DMA-transfer.

If  $n \geq 32$  and the DMA-controller does not perform the 32nd DMA-cycle, the DRQRA/B-line will go high again as soon as  $\overline{\text{CSS}}$  goes high, thus indicating further bytes to fetch.

**5.1.1.6.2 Transmit FIFO (XFIFO)**

Access: write

Reset value:  $\text{xx}_H$



**TD7...0** Transmit Data 7...0, data byte to be transmitted on the serial interface.

**Interrupt controlled data transfer** (interrupt mode, selected if DMA-bit in register XBCH is reset).

Up to 32 bytes of transmit data can be written to the XFIFO following an XPR-interrupt.

**DMA controlled data transfer** (DMA-mode, selected if DMA-bit in register XBCH is set).

Prior to any data transfer, the actual byte count of the frame to be transmitted must be written to the registers XBCH, XBCL:

1 byte: XBCL = 0

n bytes: XBCL = n - 1



**Description of Registers**

If a data transfer is then initiated via the CMDR-register (commands XPD/XTF or XDD), the SACCO autonomously requests the correct amount of block data transfers ( $n \times 32 + \text{remainder}$ ,  $n = 0, 1, \dots$ ).

The corresponding DRQTA/B pin remains “high” as long as the XFIFO requires data transfers. It is deactivated upon the rising edge of  $\overline{WR}$  in the DMA-transfer 31 or  $n - 1$  respectively. The DMA-controller must take care to perform the last DMA-transfer. If it is missing, the DRQTA/B-line will go active again when  $\overline{CSS}$  is raised.

**5.1.1.6.3 Interrupt Status Register (ISTA\_A/B)**

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

RME	RPF	0	XPR	0	0	0	0
-----	-----	---	-----	---	---	---	---

**RME** Receive Message End.  
 A message of up to 32 bytes or the last part of a message greater than 32 bytes has been received and is now available in the RFIFO. The message is complete! The actual message length can be determined by reading the registers RBCL, RBCH. RME is not generated when an extended HDLC- frame is recognized in auto-mode (EHC interrupt).  
 In DMA-mode a RME-interrupt is generated after the DMA-transfer has been finished correctly, indicating that the processor should read the registers RBCH/RBCL to determine the correct message length.

**RPF** Receive Pool Full.  
 A data block of 32 bytes is stored in the RFIFO. The message is not yet completed!

*Note: This interrupt is only generated in interrupt mode (not in DMA-mode).*

**XPR** Transmit Pool Ready.  
 A data block of up to 32 bytes can be written to the XFIFO.

**5.1.1.6.4 Mask Register (MASK\_A/B)**

Access: write

Reset value: 00<sub>H</sub> (all interrupts enabled)

bit 7

bit 0

RME	RPF	0	XPR	0	0	0	0
-----	-----	---	-----	---	---	---	---

**RME** enables(0)/disables(1) the Receive Message End interrupt.

**RPF** enables(0)/disables(1) the Receive Pool Full interrupts.

Description of Registers

**XPR** enables(0)/disables(1) the Transmit Pool Ready interrupt.

Each interrupt source can be selectively masked by setting the respective bit in the MASK\_A/B-register (bit position corresponding to the ISTA\_A/B-register). Masked interrupts are internally stored but not indicated when reading ISTA\_A/B and also not flagged into the top level ISTA. After releasing the respective MASK\_A/B-bit they will be indicated again in ISTA\_A/B and in the top level ISTA.

When writing register MASK\_A/B while ISTA\_A/B indicates a non masked interrupt the INT-pin is temporarily set into the inactive state. In this case the interrupt remains indicated in the ISTA\_A/B until these registers are read.

**5.1.1.6.5 Extended Interrupt Register (EXIR\_A/B)**

Access: read

Reset value: 00<sub>H</sub>

bit 7	XMR	XDU/EXE	EHC	RFO	0	RFS	0	0	bit 0
-------	-----	---------	-----	-----	---	-----	---	---	-------

**XMR** Transmit Message Repeat.

The transmission of a frame has to be repeated because:

- A frame consisting of more than 32 bytes is polled a second time in auto-mode.
- Collision has occurred after sending the 32nd data byte of a message in a bus configuration.
- CTS (transmission enable) has been withdrawn after sending the 32nd data byte of a message in point-to-point configuration.

**XDU/EXE** Transmission Data Underrun/Extended transmission End.

The actual frame has been aborted with IDLE, because the XFIFO holds no further data, but the frame is not yet complete according to registers XBCH/XBCL. In extended transparent mode, this bit indicates the transmission end condition.

It is not possible to transmit frames when a XMR- or XDU-interrupt is indicated.

**EHC** Extended HDLC-frame.

The SACCO has received a frame in auto-mode which is neither a RR- nor an I-frame. The control byte is stored temporarily in the RHCR-register but not in the RFIFO.

**RFO** Receive Frame Overflow.

A frame could not be stored due to the occupied RFIFO (i.e. whole frame has been lost). This interrupt can be used for statistical purposes and indicates, that the CPU does not respond quickly enough to an incoming RPF- or RME-interrupt.

Description of Registers

**RFS** Receive Frame Start.  
 This is an early receiver interrupt activated after the start of a valid frame has been detected, i.e. after a valid address check in operation modes providing address recognition, otherwise after the opening flag (transparent mode 0), delayed by two bytes.  
 After a RFS-interrupt the contents of

- RHCR
- RAL1
- RSTA bit3-0

are valid and can be read by the CPU.  
 The RFS-interrupt is maskable by programming bit CCR2:RIE.

**5.1.1.6.6 Command Register (CMDR)**

Access: write  
 Reset value: 00<sub>H</sub>

bit 7				bit 0			
RMC	RHR	AREP/ XREP	0	XPD/ XTF	XDD	XME	XRES

*Note: The maximum time between writing to the CMDR-register and the execution of the command is 2.5 HDC-clock cycles. Therefore, if the CPU operates with a very high clock speed in comparison to the SACCO-clock, it is recommended that the bit STAR:CEC is checked before writing to the CMDR-register to avoid losing of commands.*

**RMC** Receive Message Complete.  
 A '1' confirms, that the actual frame or data block has been fetched following a RPF- or RME-interrupt, thus the occupied space in the RFIFO can be released.

*Note: In DMA-mode this command is only issued once after a RME-interrupt. The SACCO does not generate further DMA requests prior to the reception of this command.*

**RHR** Reset HDLC-Receiver.  
 A '1' deletes all data in the RFIFO and in the HDLC-receiver.

**AREP/** Auto Repeat/Transmission Repeat.

**XREP**

- Auto-mode: AREP
- The frame (max. length 32 byte) stored in XFIFO can be polled repeatedly by the opposite station until the frame is acknowledged.
- Extended transparent mode 0,1: XREP  
 Together with XTF- and XME-set (CMDR = 2A<sub>H</sub>) the SACCO repeatedly

## Description of Registers

transmits the contents of the XFIFO (1...32 bytes) fully transparent without HDLC-framing, i.e. without flag, CRC-insertion, bit stuffing.

The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.

When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

*Note: MODE:CFT must be set to '0' when using cyclic transmission.*

**XPD/XTF** Transmit Prepared Data/Transmit Transparent Frame.

- Auto-mode: XPD  
Prepares the transmission of an I-frame ("prepared data") in auto-mode. The actual transmission starts, when the SACCO receives an I-frame with poll-bit set and AxH as the first data byte (PBC-command "transmit prepared data"). Upon the reception of a different poll frame a response is generated automatically (RR-poll ⇒RR-response, I-poll with first byte not AxH ⇒I-response).
- Non-auto-mode, transparent mode 0,1: XTF  
The transmission of the XFIFO contents is started, an opening flag sequence is automatically added.
- Extended transparent mode 0,1: XTF  
The transmission of the XFIFO contents is started, no opening flag sequence is added.

**XDD** Transmit Direct Data (auto-mode only!).

Prepares the transmission of an I-frame ("direct data") in auto-mode. The actual transmission starts, when the SACCO receives a RR-frame with poll-bit set. Upon the reception of an I-frame with poll-bit set, an I-response is issued.

**XME** Transmit Message End (interrupt mode only).

A '1' indicate that the data block written last to the XFIFO completes the actual frame. The SACCO can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data. XME is used only in combination with XPD/XTF or XDD.

*Note: When using the DMA-mode XME must not be used.*

**XRES** Transmit Reset.

The contents of the XFIFO is deleted and IDLE is transmitted. This command can be used by the CPU to abort a frame currently in transmission. After setting XRES a XPR-interrupt is generated in every case.

### 5.1.1.6.7 Mode Register (MODE)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
MDS1	MDS0	ADM	CFT	RAC	0	0	TLP

**MDS1...0** Mode Select.  
 The operating mode of the HDLC-controller is selected.  
 00...auto-mode  
 01...non-auto-mode  
 11...extended transparent mode

**ADM** Address Mode.  
 The meaning of this bit varies depending on the selected operating mode:

- Auto-mode / non-auto mode  
 Defines the length of the HDLC-address field.  
 0...8-bit address field,  
 1...16-bit address field.
- Transparent mode  
 0... no address recognition: transparent mode 0 (D-channel arbiter)  
 1... high byte address recognition: transparent mode 1
- Extended transparent mode  
 0... receive data in RAL1: extended transparent mode 0  
 1... receive data in RFIFO and RAL1: extended transparent mode 1

*Note: In extended transparent mode 0 and 1 the bit MODE:RAC must be reset to enable fully transparent reception.*

**CFT** Continuous Frame Transmission.  
 1...when CFT is set the XPR-interrupt is generated immediately after the CPU accessible part of XFIFO is copied into the transmitter section.  
 0...otherwise the XPR-interrupt is delayed until the transmission is completed (D-channel arbiter).

**RAC** Receiver Active.  
 Via RAC the HDLC-receiver can be activated/deactivated.  
 0...HDLC-receiver inactive  
 1...HDLC-receiver active  
 In extended transparent mode 0 and 1 RAC must be reset (HDLC-receiver disabled) to enable fully transparent reception.

**TLP** Test Loop.  
 When set input and output of the HDLC-channel are internally connected.  
 (transmitter channel A - receiver channel A)

transmitter channel B - receiver channel B)  
TXDA/B are active, RXDA/B are disabled.

### 5.1.1.6.8 Channel Configuration Register 1 (CCR1)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
PU	SC1	SC0	ODS	ITF	CM2	CM1	CM0

- PU** Power-Down Mode.  
0...power-down (standby), the internal clock is switched off.  
Nevertheless, register read/write access is possible.  
1...power-up (active).
- SC1...0** Serial Port Configuration.  
00...point to point configuration,  
01...bus configuration, timing mode 1, data is output with the rising edge of the data clock on pin TxDA/B and evaluated 1/2 clock period later with the falling clock edge at pin CxDA/B.  
11...bus configuration, timing mode 2, data is output with the falling edge of the data clock and evaluated with the next falling clock edge.  
Thus one complete clock period is available between data output and evaluation.
- ODS** Output Driver Select. (Only valid if configured in stand-alone mode. When connected to IOM or PCM sign.mux, CCR1:ODS of SIDE0 defines the output driver).  
Defines the function of the transmit data pin (TxDA/B).  
0...TxDA/B-pin open drain output  
1...TxDA/B-pin push-pull output

**Up to Version 1.2 when selecting a bus configuration only the open drain option must be selected.**

Compared to the Version 1.2 the **Version 1.3** provides new features:

Push-pull operation may be selected in bus configuration (up to Version 1.2 only open drain):

- When active TXDA / TXDB outputs serial data in push-pull-mode
- When inactive (interframe or inactive timeslots) TXDA / TXDB outputs '1'

*Note: When bus configuration with direct connection of multiple ELIC's is used open drain option is still recommended.*

*The push-pull option with bus configuration can only be used if an external tri-state buffer is placed between TXDA / TXDB and the bus.*

Description of Registers

Due to the delay of  $\overline{TSCA}$  /  $\overline{TSCB}$  in this mode (see description of bits SOC(0:1) in register CCR2 (**chapter 5.1.1.5.9**) these signals cannot directly be used to enable this buffer.

**ITF** Inter frame Time Fill.  
 Determines the “no data to send” state of the transmit data pin (TxDA/B).  
 0...continuous IDLE-sequences are output ('11111111' bit pattern).  
 In a bus configuration (CCR1:SC0 = 1) ITF is implicitly set to '0' (continuous '1's are transmitted).  
 1...continuous FLAG-sequences are output ('01111110' bit pattern). In a bus configuration (CCR1:SC0 = 1) ITF is implicitly set to '0' (continuous '1's are transmitted).

*Note: ITF has to be set 0 if clock mode 3 is used.*

**CM2** Clock rate.  
 0...single rate data clock  
 1...double rate data clock

**CM1...0** Clock Mode.  
 Determines the mode in which the data clock is forwarded toward the receiver/transmitter.  
 00...clock mode 0: external data clock, permanently enabled.  
 01...clock mode 1: external data clock, gated by an enable strobe forwarded via pin HFS.  
 10...clock mode 2: external data clock, programmable time-slot assignment, frame synchronization pulse forwarded via pin HFS.  
 11...not allowed

**5.1.1.6.9 Channel Configuration Register 2 (CCR2)**

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
SOC1	SOC0	XCS0	RCS0	<b>TXDE</b>	<b>RDS</b>	RIE	0

**SOC1,**  
**SOC0** The function of the TSCA/B-pin can be defined programming SOC1,SOC0.

- Bus configuration:
  - 00...the TSCA/B output is activated only during the transmission of a frame delayed by one clock period. **When transmission was stopped due to a collision TSCA/B remains inactive.**
  - 10...the TSCA/B-output is always high (disabled).



Description of Registers

11...the TSCA/B-output indicates the reception of a data frame (active low)

- Point-to-point configuration:  
 0x...the TSCA/B-output is activated during the transmission of a frame.  
 1x...the TSCA/B-output is activated during the transmission of a frame and of inter frame timefill.

**XCS0** Transmit/receive Clock Shift, bit 0 (only clock mode 2).

**RCS0** Together with the bits XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR) the clock shift relative to the frame synchronization signal of the transmit (receive) time-slot can be adjusted. A clock shift of 0...7 bits is programmable (clock mode 2 only!).

*Note: In the clock modes 0, 1 and 3 XCS0 and RCS0 has to be set to '0'.*

**TXDE** Transmit Data Enable.  
 0...the pin TxDA/B is disabled (in the state high impedance).  
 1...the pin TxDA/B is enabled. Depending on the programming of bit CCR1:ODS it has a push pull or open drain characteristic.

**RDS** Receive Data Sampling.  
 0 : serial data on RXDA/B is sampled at the falling edge of HDCA/B.  
 1 : serial data on RXDA/B is sampled at the rising edge of HDCA/B.

*Note: With RDS = 1 the sampling edge is shifted 1/2 clock phase forward. The data is internally still processed with the falling edge.*

**RIE** Receive frame start Enable.  
 When set, the RFS-interrupt in register EXIR\_A/B is enabled.

**5.1.1.6.10 Receive Length Check Register (RLCR)**

Access: write

Reset value: 0xxxxxxx<sub>H</sub>

bit 7								bit 0	
RC	RL6	RL5	RL4	RL3	RL2	RL1	RL0		

**RC** Receive Check enable.  
 A '1' enables, a '0' disables the receive frame length feature.

**RL6...0** Receive Length.  
 The maximum receive length after which data reception is suspended can be programmed in RL6...0. The maximum allowed receive frame length is  $(RL + 1) \times 32$  bytes. A frame exceeding this length is treated as if it was aborted by the opposite station (RME-interrupt, RAB-bit set (VFR in clock mode 3)).



**Description of Registers**

In this case the receive byte count (RBCH, RBCL) is greater than the programmed receive length.

**5.1.1.6.11 Status Register (STAR)**

Access: read

Reset value: 48<sub>H</sub>

bit 7

bit 0

XDOV	XFW	AREP/ XREP	RFR	RLI	CEC	XAC	AFI
------	-----	---------------	-----	-----	-----	-----	-----

**XDOV** Transmit Data Overflow.  
A '1' indicates, that more than 32 bytes have been written into the XFIFO.

**XFW** XFIFO Write enable.  
A '1' indicates, that data can be written into the XFIFO.

*Note: XFW is only valid when CEC = 0.*

**AREP/** Auto Repeat/Transmission Repeat.

**XREP** Read back value of the corresponding command bit CMDR:AREP/XREP.

**RFR** RFIFO Read enable.  
A '1' indicates, that valid data is in the RFIFO and read access is enabled.  
RFR is set with the RME- or RPF-interrupt and reset when executing the RMC-command.

**RLI** Receiver Line Inactive.  
Neither flags as inter frame time fill nor frames are received via the receive line.

*Note: Significant in point-to-point configurations!*

**CEC** Command Execution.  
When '0' no command is currently executed, the CMDR-register can be written to.  
When '1' a command (written previously to CMDR) is currently executed, no further command must temporarily be written to the CMDR-register.

**XAC** Transmitter Active.  
A '1' indicates, that the transmitter is currently active.  
In bus mode the transmitter is considered active also when it waits for bus access.

**AFI** Additional Frame Indication.  
A '1' indicates, that one or more completely received frames or the last part of a frame are in the CPU inaccessible part of the RFIFO.

In combination with the bit STAR:RFR multiple frames can be read out of the RFIFO without interrupt control.

### 5.1.1.6.12 Receive Status Register (RSTA)

Access: read

Reset value: xx<sub>H</sub>

bit 7				bit 0			
VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA

RSTA always displays the momentary state of the receiver. Because this state can differ from the last entry in the FIFO it is reasonable to always use the status bytes in the FIFO.

**VFR** Valid Frame.  
 Indicates whether the received frame is valid ('1') or not ('0' invalid).  
 A frame is invalid when

- its length is not an integer multiple of 8 bits ( $n \times 8$  bits), e.g. 25 bit,
- its is too short, depending on the selected operation mode:  
 auto-mode/non-auto mode (2-byte address field): 4 bytes  
 auto-mode/non-auto mode (1-byte address field): 3 bytes  
 transparent mode 1: 3 bytes  
 transparent mode 0: 2 bytes
- a frame was aborted (note: VFR can also be set when a frame was aborted)

*Note: Shorter frames are not reported.*

**RDO** Receive Data Overflow.  
 A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.

**CRC** CRC-Compare Check.  
 0: CRC check failed, received frame contains errors.  
 1: CRC check o.k., received frame is error free.

**RAB** Receive message Aborted.  
 When '1' the received frame was aborted from the transmitting station. According to the HDLC-protocol, this frame must be discarded by the CPU.

**HA1...0** High byte Address compare.  
 In operating modes which provide high byte address recognition, the SACCO compares the high byte of a 2-byte address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FEH and FCH (group address). Depending on the result of the comparison, the following bit combinations are possible:  
 10...RAH1 has been recognized.

---

## Description of Registers

00...RAH2 has been recognized.

01...group address has been recognized.

*Note: If RAH1, RAH2 contain the identical value, the combination 00 will be omitted.  
HA1...0 is significant only in 2-byte address modes.*

**C/R** Command/Response; significant only, if 2-byte address mode has been selected. Value of the C/R bit (bit of high address byte) in the received frame.

**LA** Low byte Address compare.  
The low byte address of a 2-byte address field or the single address byte of a 1-byte address field is compared with two programmable registers (RAL1, RAL2). Depending on the result of the comparison LA is set.

0...RAL2 has been recognized,

1...RAL1 has been recognized.

In non-auto mode, according to the X.25 LAP B-protocol, RAL1/RAL2 may be programmed to differ between COMMAND/RESPONSE frames.

*Note: The receive status byte is duplicated into the RFIFO (clock mode 0-2) following the last byte of the corresponding frame.*



### 5.1.1.6.15 Transmit Address Byte 2 (XAD2)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
XAD27	XAD26	XAD25	XAD24	XAD23	XAD22	XAD21	XAD20

#### XAD27...20 Transmit Address byte 2.

The value stored in XAD2 is included automatically as the low address byte of all frames transmitted in auto-mode (2-byte address field only).

### 5.1.1.6.16 Receive Address Byte Low Register 1 (RAL1)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAL17	RAL16	RAL15	RAL14	RAL13	RAL12	RAL11	RAL10

#### RAL17...10 Receive Address byte Low register 1.

The general function (read/write) and the meaning or contents of this register depends on the selected operating mode:

- Auto-mode, non-auto mode (address recognition) - write only: compare value 1, address recognition (low byte in case of 2-byte address field).
- Transparent mode 1 (high byte address recognition) - read only: RAL1 contains the byte following the high byte of the address in the received frame (i.e. the second byte after the opening flag).
- Transparent mode 0 (no address recognition) - read only: contains the first byte after the opening flag (first byte of the received frame).
- Extended transparent mode 0,1 - read only: RAL1 contains the actual data byte currently assembled at the RxD-pin by passing the HDLC-receiver (fully transparent reception without HDLC-framing).

*Note: In auto-mode and non-auto mode the read back of the programmed value is inverted.*

### 5.1.1.6.17 Receive Address Byte Low Register 2 (RAL2)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAL27	RAL26	RAL25	RAL24	RAL23	RAL22	RAL21	RAL20

**RAL27...20** Receive Address byte Low register 1.

- Auto-mode, non-auto mode (address recognition): compare value 2, address recognition (low byte in case of 2-byte address field).

*Note: Normally used for broadcast address.*

### 5.1.1.6.18 Receive Address Byte High Register 1 (RAH1)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAH17	RAH16	RAH15	RAH14	RAH13	RAH12	0	0

**RAH17...12** Receiver Address byte High register 1.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 1, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH1 must be set to 00<sub>H</sub>.*

### 5.1.1.6.19 Receive Address Byte High Register 2 (RAH2)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAH27	RAH26	RAH25	RAH24	RAH23	RAH22	0	0

**RAH27...22** Receiver Address byte High register 2.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 2, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH2 must be set to 00<sub>H</sub>.*

### 5.1.1.6.20 Receive Byte Count Low (RBCL)

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0
------	------	------	------	------	------	------	------

**RBC7...0** Receive Byte Count.  
 Together with RBCH (bits RBC11 - RBC8), the length of the actual received frame (0...4095 bytes) can be determined. These registers must be read by the CPU following a RME interrupt.

### 5.1.1.6.21 Receive Byte Count High (RBCH)

Access: read

Reset value: 000xxxxx<sub>H</sub>

bit 7

bit 0

DMA	0	0	OV	RBC11	RBC10	RBC9	RBC8
-----	---	---	----	-------	-------	------	------

**DMA** DMA-mode status indication.  
 Read back value representing the DMA-bit programmed in register XBCH.

**OV** Counter Overflow.  
 A '1' indicates that more than 4095 bytes were received.  
 The received frame exceeded the byte count in RBC11...RBC0.

**RBC11...8** Receive Byte Count high.  
 Together with RBCL (bits RBC7...RBC0) the length of the received frame can be determined.

### 5.1.1.6.22 Transmit Byte Count Low (XBCL)

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

XBC7	XBC6	XBC5	XBC4	XBC3	XBC2	XBC1	XBC0
------	------	------	------	------	------	------	------

**XBC7...0** Together with XBCH (bits XBC11...XBC8) this register is used in DMA-mode to program the length of the next frame to be transmitted (1...4096 bytes). The number of transmitted bytes is XBC + 1.  
 Consequently the SACCO can request the correct number of DMA-cycles after a XDD/XTF- or XDD-command.

### 5.1.1.6.23 Transmit Byte Count High (XBCH)

Access: write

Reset value: 0000xxxx<sub>H</sub>

bit 7				bit 0			
DMA	0	0	XC	XBC11	XBC10	XBC9	XBC8

**DMA** DMA-mode.  
 Selects the data transfer mode between the SACCO FIFOs and the system memory:  
 0...interrupt controlled data transfer (interrupt mode).  
 1...DMA controlled data transfer (DMA-mode).

**XC** Transmit Continuously.  
 When XC is set the SACCO continuously requests for transmit data ignoring the transmit byte count programmed in register XBCH and XBCL.

*Note: Only valid in DMA-mode.*

**XBC11...8** Transmit Byte Count high.  
 Together with XBC7...XBC0 the length of the next frame to be transmitted in DMA-mode is determined (1...4096 bytes).

### 5.1.1.6.24 Time-Slot Assignment Register Transmit (TSAX)

Access: write

Reset value: xx<sub>H</sub>

bit 7					bit 0		
TSNX5	TSNX4	TSNX3	TSNX2	TSNX1	TSNX0	XCS2	XCS1

**TSNX5...0** Time-Slot Number Transmit.  
 Selects one of up to 64 time-slots (00<sub>H</sub> - 3F<sub>H</sub>) in which data is transmitted in clock mode 2. The number of bits per time-slot is programmable in register XCCR.

**XCS2...1** Transmit Clock Shift bit2-1.  
 Together with XCS0 in register CCR2 the transmit clock shift can be adjusted in clock mode 2.



### 5.1.1.6.25 Time-Slot Assignment Register Receive (TSAR)

Access: write

Reset value:  $xx_H$

bit 7						bit 0	
TSNR5	TSNR4	TSNR3	TSNR2	TSNR1	TSNR0	RCS2	RCS1

**TSNR5...0** Time-Slot Number Receive.

Selects one of up to 64 time-slots ( $00_H - 3F_H$ ) in which data is received in clock mode 2. The number of bits per time-slot is programmable in register RCCR.

**RCS2...1** Receive Clock Shift bit2-1.

Together with RCS0 in register CCR2 the transmit clock shift can be adjusted in clock mode 2.

### 5.1.1.6.26 Transmit Channel Capacity Register (XCCR)

Access: write

Reset value:  $00_H$

bit 7						bit 0	
XBC7	XBC6	XBC5	XBC4	XBC3	XBC2	XBC1	XBC0

**XBC7...0** Transmit Bit Count.

Defines the number of bits to be transmitted in a time-slot in clock mode 2 (number of bits per time-slot =  $XBC + 1$  (1...256 bits/time-slot)).

*Note: In extended transparent mode the width of the time-slot has to be  $n \times 8$  bits.*

### 5.1.1.6.27 Receive Channel Capacity Register (RCCR)

Access: write

Reset value:  $00_H$

bit 7						bit 0	
RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0

**RBC7...0** Receive Bit Count.

Defines the number of bits to be received in a time-slot in clock mode 2. Number of bits per time-slot =  $RBC + 1$  (1...256 bits/time-slot).

*Note: In extended transparent mode the width of the time-slot has to be  $n \times 8$  bits.*

### 5.1.1.6.28 Version Status Register (VSTR)

Access: read

bit 7				bit 0			
1	0	0	0	VN3	VN2	VN1	VN0

**VN3...0** SACCO Version Number.  
 82<sub>H</sub>...SACCO-B in DOC V1.1.  
 83<sub>H</sub>...SACCO-B in DOC V2.1.

### 5.1.1.7 D-Channel Arbiter Registers

#### 5.1.1.7.1 Arbiter Mode Register (AMO)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7					bit 0		
FCC4	FCC3	FCC2	FCC1	FCC0	SCA	CCHH	CCHM

**FCC4...0** Full selection Counter.  
 The value (FCC4...0 +1) defines the number of IOM-frames before the arbiter state machine changes from the state “limited selection” to the state “full selection”, if the ASM does not detect any '0' on the remaining serial input lines (D-channels).  
 E.g. max. delay = 9 frames ⇒ AMO:FCC4...0 = 01000.

*Note: To avoid arbiter locking, either*

- a) the state limited selection can be skipped by setting FCC4...0 = 00<sub>H</sub>, or*
- b) the FCC4...0 value must be greater than the value described in **chapter 2.1.2.5.3**.*

**SCA** Suspend Counter Activation.  
 0...the suspend counter controls the arbiter state machine.  
 1...the suspend counter is disabled (e.g. for control by μP).

**CCHH** Control Channel Handling.  
 The control channel takes place:  
 0...in the C/I channel  
 1...in the MR bit (Monitor channel receive bit)

**CCHM** Control Channel Master activation.  
 0...disables the control channel master.  
 When disabled, all channels enabled in the DCE0-3 registers are sent the “available” information even when the SACCO-A is currently not

available.

1...enables the control channel master.

During reception of D-channel data from a channel which has been enabled in the DCE0-3 registers all other enabled channels are sent the “blocked” information from the Control Memory (CM).

*Note: The D-channel arbiter can only be operated with FSC framing control modes 3, 6 and 7.*

### 5.1.1.7.2 Arbiter State Register (ASTATE)

Access: read

Reset value: 00<sub>H</sub>

bit 7

bit 0

AS2	AS1	AS0	PAD1	PAD0	CHAD2	CHAD1	CHAD0
-----	-----	-----	------	------	-------	-------	-------

**AS2...0** Arbiter (receive channel selector) State:

- 000 : suspended
- 100 : full selection
- 011 : limited selection
- 001 : expect frame
- 010 : receive frame

**PAD1...0** Port Address.

The related frame was received on IOM-port PAD1...0

**CHAD2...0** Channel Address.

The related frame was received in IOM-channel CHAD2...0.

### 5.1.1.7.3 Suspend Counter Value Register (SCV)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7

bit 0

SCV7	SCV6	SCV5	SCV4	SCV3	SCV2	SCV1	SCV0
------	------	------	------	------	------	------	------

**SCV7...0** Suspend Counter Value.

The value  $(SCV7...0 + 1) \times 32$  defines the number of D-bits which are analyzed in the state “expect frame” before the arbiter enters the state suspended state and an interrupt is issued.

Min.:  $32 \times$  D-bits (16 frames), max: 8192 D-bits.

#### 5.1.1.7.4 D-Channel Enable Register IOM<sup>®</sup>-Port 0 (DCE0)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
DCE07	DCE06	DCE05	DCE04	DCE03	DCE02	DCE01	DCE00

#### 5.1.1.7.5 D-Channel Enable Register IOM<sup>®</sup>-Port 1 (DCE1)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
DCE17	DCE16	DCE15	DCE14	DCE13	DCE12	DCE11	DCE10

#### 5.1.1.7.6 D-Channel Enable Register IOM<sup>®</sup>-Port 2 (DCE2)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
DCE27	DCE26	DCE25	DCE24	DCE23	DCE22	DCE21	DCE20

#### 5.1.1.7.7 D-Channel Enable Register IOM<sup>®</sup>-Port 3 (DCE3)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
DCE37	DCE36	DCE35	DCE34	DCE33	DCE32	DCE31	DCE30

#### **DCE<sub>n</sub>7...0** D-Channel Enable bits channel 7-0, IOM-port n.

0...D-channel i on IOM-port n is disabled for data reception. The control channel of a disabled D-channel is not manipulated by the control channel master. It passes the value stored in the EPIC-1 control memory (C/I or MR must = "blocked"). The disabling of a D-channel has an immediate effect also when the channel is active. In this case the transmitter (HDLC-controller in the subscriber terminal) is forced to abort the current frame.

1...D-channel i on IOM-port n is enabled for data reception.

The control channel of an enabled D-channel is manipulated

- a) by the control channel master, if AMO:CCHM = 1,
- b) directly via DCE, if AMO:CCHM = 0.

Note: When ELIC1 is connected to IOM-2 port 4 to 7 the value for n is IOM port number minus 4.

### 5.1.1.7.8 Transmit D-Channel Address Register (XDC)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
0	0	BCT	PAD1	PAD0	CHAD2	CHAD1	CHAD0

**BCT** Broadcast Transmission, BCT = 1 enables broadcast transmission. The transmitted frame is send to all channels enabled in the registers BCG0-3.

**PAD1...0** Port Address, defines the transmit IOM-port when BCT = 0.

Note: When ELIC1 is connected to IOM-2 port 4 to 7 the value for n is IOM port number minus 4.

**CHAD2...0** Channel Address, defines the transmit IOM-channel when BCT = 0.

### 5.1.1.7.9 Broadcast Group IOM<sup>®</sup>-Port 0 (BCG0)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
BCE07	BCE06	BCE05	BCE04	BCE03	BCE02	BCE01	BCE00

### 5.1.1.7.10 Broadcast Group IOM<sup>®</sup>-Port 1 (BCG1)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
BCE17	BCE16	BCE15	BCE14	BCE13	BCE12	BCE11	BCE10

### 5.1.1.7.11 Broadcast Group IOM<sup>®</sup>-Port 2 (BCG2)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7				bit 0			
BCE27	BCE26	BCE25	BCE24	BCE23	BCE22	BCE21	BCE20

### 5.1.1.7.12 Broadcast Group IOM<sup>®</sup>-Port 3 (BCG3)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7						bit 0	
BCE37	BCE36	BCE35	BCE34	BCE33	BCE32	BCE31	BCE30

**BCEn7...0** Broadcast Enable bit channel 7-0, IOM-port n.

BCEni:

0...D-channel i, IOM-port n is disabled for broadcast transmission.

1...D-channel i, IOM-port n is enabled for broadcast transmission.

*Note: When ELIC1 is connected to IOM-2 port 4 to 7 the value for n is IOM port number minus 4.*

## 5.1.2 SIDE C Register Description

The SIDE C contains 4 identical HDLC Controllers: SACCO-n.

### 5.1.2.1 Receive FIFO (RFIFO)

Access: read

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

**RD7...0** Receive Data 7...0, data byte received on the serial interface.

#### Interrupt controlled data transfer

Up to 32 bytes of received data can be read from the RFIFO following an RPF or an RME interrupt.

RPF-interrupt: exactly 32 bytes to be read.

RME-interrupt: the number of bytes can be determined reading the registers RBCL, RBCH.

### 5.1.2.2 Transmit FIFO (XFIFO)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0

Description of Registers

**TD7...0** Transmit Data 7...0, data byte to be transmitted on the serial interface. Interrupt controlled data transfer. Up to 32 bytes of transmit data can be written to the XFIFO following an XPR-interrupt.

**5.1.2.3 Interrupt Status Register (ISTA\_A/B)**

Access: read

Reset value: 00<sub>H</sub>

bit 7				bit 0			
RME	RPF	0	XPR	0	0	0	0

**RME** Receive Message End.  
A message of up to 32 bytes or the last part of a message greater than 32 bytes has been received and is now available in the RFIFO. The message is complete! The actual message length can be determined by reading the registers RBCL, RBCH. RME is not generated when an extended HDLC- frame is recognized in auto-mode (EHC interrupt).

**RPF** Receive Pool Full.  
A data block of 32 bytes is stored in the RFIFO. The message is not yet completed!

**XPR** Transmit Pool Ready.  
A data block of up to 32 bytes can be written to the XFIFO.

**5.1.2.4 Mask Register (MASK\_A/B)**

Access: write

Reset value: 00<sub>H</sub> (all interrupts enabled)

bit 7				bit 0			
RME	RPF	0	XPR	0	0	0	0

**RME** enables(0)/disables(1) the Receive Message End interrupt.

**RPF** enables(0)/disables(1) the Receive Pool Full interrupts.

**XPR** enables(0)/disables(1) the Transmit Pool Ready interrupt.

Each interrupt source can be selectively masked by setting the respective bit in the MASK\_A/B-register (bit position corresponding to the ISTA\_A/B-register). Masked interrupts are internally stored but not indicated when reading ISTA\_A/B and also not flagged into the top level ISTA. After releasing the respective MASK\_A/B-bit they will be indicated again in ISTA\_A/B and in the top level ISTA.

**Description of Registers**

When writing register MASK\_A/B while ISTA\_A/B indicates a non masked interrupt the  $\overline{\text{INT}}$ -pin is temporarily set into the inactive state. In this case the interrupt remains indicated in the ISTA\_A/B until these registers are read.

**5.1.2.5 Extended Interrupt Register (EXIR\_A/B)**

Access: read

Reset value: 00<sub>H</sub>

bit 7							bit 0
XMR	XDU/EXE	EHC	RFO	0	RFS	0	0

**XMR** Transmit Message Repeat.  
 The transmission of a frame has to be repeated because:

- A frame consisting of more then 32 bytes is polled a second time in auto-mode.
- Collision has occurred after sending the 32nd data byte of a message in a bus configuration.
- CTS (transmission enable) has been withdrawn after sending the 32nd data byte of a message in point-to-point configuration.

**XDU/EXE** Transmission Data Underrun/Extended transmission End.  
 The actual frame has been aborted with IDLE, because the XFIFO holds no further data, but the frame is not yet complete according to registers XBCH/XBCL.  
 In extended transparent mode, this bit indicates the transmission end condition.

*Note: It is not possible to transmit frames when a XMR- or XDU-interrupt is indicated.*

**EHC** Extended HDLC-frame.  
 The SACCO has received a frame in auto-mode which is neither a RR- nor an I-frame. The control byte is stored temporarily in the RHCR-register but not in the RFIFO.

**RFO** Receive Frame Overflow.  
 A frame could not be stored due to the occupied RFIFO (i.e. whole frame has been lost). This interrupt can be used for statistical purposes and indicates, that the CPU does not respond quickly enough to an incoming RPF- or RME-interrupt.

**RFS** Receive Frame Start.  
 This is an early receiver interrupt activated after the start of a valid frame has been detected, i.e. after a valid address check in operation modes providing address recognition, otherwise after the opening flag (transparent mode 0), delayed by two bytes.



After a RFS-interrupt the contents of

- RHCR
- RAL1
- RSTA bit3-0

are valid and can be read by the CPU.

The RFS-interrupt is maskable by programming bit CCR2:RIE.

### 5.1.2.6 Command Register (CMDR)

Access: write

Reset value: 00<sub>H</sub>

bit 7

bit 0

RMC	RHR	AREP/ XREP	0	XPD/ XTF	XDD	XME	XRES
-----	-----	---------------	---	-------------	-----	-----	------

*Note: The maximum time between writing to the CMDR-register and the execution of the command is 2 bits clocks. Therefore, if the CPU operates with a very high clock speed in comparison to the SACCO-clock, it is recommended that the bit STAR:CEC is checked before writing to the CMDR-register to avoid losing of commands.*

- RMC** Receive Message Complete.  
A '1' confirms, that the actual frame or data block has been fetched following a RPF- or RME-interrupt, thus the occupied space in the RFIFO can be released.
- RHR** Reset HDLC-Receiver.  
A '1' deletes all data in the RFIFO and in the HDLC-receiver.
- AREP/** Auto Repeat/Transmission Repeat.
- XREP**
- Auto-mode: AREP  
The frame (max. length 32 byte) stored in XFIFO can be polled repeatedly by the opposite station until the frame is acknowledged.
  - Extended transparent mode 0,1: XREP  
Together with XTF- and XME-set (CMDR = 2A<sub>H</sub>) the SACCO repeatedly transmits the contents of the XFIFO (1...32 bytes) fully transparent without HDLC-framing, i.e. without flag, CRC-insertion, bit stuffing. The cyclical transmission continues until the command (CMDR:XRES) is executed or the bit XREP is reset. The inter frame timefill pattern is issued afterwards.  
When resetting XREP, data transmission is stopped after the next XFIFO-cycle is completed, the XRES-command terminates data transmission immediately.

---

**Description of Registers**

*Note: MODE:CFT must be set to '0' when using cyclic transmission.*

**XPD/XTF** Transmit Prepared Data/Transmit Transparent Frame.

- **Auto-mode: XPD**  
Prepares the transmission of an I-frame (“prepared data”) in auto-mode. The actual transmission starts, when the SACCO receives an I-frame with poll-bit set and AxH as the first data byte (PBC-command “transmit prepared data”). Upon the reception of a different poll frame a response is generated automatically (RR-poll ⇒RR-response, I-poll with first byte not AxH ⇒I-response).
- **Non-auto-mode, transparent mode 0,1: XTF**  
The transmission of the XFIFO contents is started, an opening flag sequence is automatically added.
- **Extended transparent mode 0,1: XTF**  
The transmission of the XFIFO contents is started, no opening flag sequence is added.

**XDD** Transmit Direct Data (auto-mode only!).

Prepares the transmission of an I-frame (“direct data”) in auto-mode. The actual transmission starts, when the SACCO receives a RR-frame with poll-bit set. Upon the reception of an I-frame with poll-bit set, an I-response is issued.

**XME** Transmit Message End .

A '1' indicate that the data block written last to the XFIFO completes the actual frame. The SACCO can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data. XME is used only in combination with XPD/XTF or XDD.

*Note: When using the DMA-mode XME must not be used.*

**XRES** Transmit Reset.

The contents of the XFIFO is deleted and IDLE is transmitted. This command can be used by the CPU to abort a frame currently in transmission. After setting XRES a XPR-interrupt is generated in every case.

### 5.1.2.7 Mode Register (MODE)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7					bit 0		
MDS1	MDS0	ADM	CFT	RAC	0	0	TLP

**MDS1...0** Mode Select.  
 The operating mode of the HDLC-controller is selected.  
 00...auto-mode  
 01...non-auto-mode  
 10...transparent mode  
 11...extended transparent mode

**ADM** Address Mode.  
 The meaning of this bit varies depending on the selected operating mode:

- Auto-mode / non-auto mode  
 Defines the length of the HDLC-address field.  
 0...8-bit address field,  
 1...16-bit address field.
- Transparent mode  
 0...no address recognition: transparent mode 0  
 1...high byte address recognition: transparent mode 1
- Extended transparent mode  
 0...receive data in RAL1: extended transparent mode 0  
 1...receive data in RFIFO and RAL1: extended transparent mode 1

*Note: In extended transparent mode 0 and 1 the bit MODE:RAC must be reset to enable fully transparent reception.*

**CFT** Continuous Frame Transmission.  
 1...When CFT is set the XPR-interrupt is generated immediately after the CPU accessible part of XFIFO is copied into the transmitter section.  
 0...Otherwise the XPR-interrupt is delayed until the transmission is completed .

**RAC** Receiver Active.  
 Via RAC the HDLC-receiver can be activated/deactivated.  
 0...HDLC-receiver inactive  
 1...HDLC-receiver active  
 In extended transparent mode 0 and 1 RAC must be reset (HDLC-receiver disabled) to enable fully transparent reception.

**TLP** Test Loop.  
 When set input and output of the HDLC-channel are internally connected.

(transmitter channel A - receiver channel A  
transmitter channel B - receiver channel B)  
TXDA/B are active, RXDA/B are disabled.

### 5.1.2.8 Channel Configuration Register 1 (CCR1)

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
PU	0	0	ODS	ITF	CM2	1	0

**PU** Power-down mode.  
0...power-down (standby), the internal clock is switched off.  
Nevertheless, register read/write access is possible.  
1...power-up (active).

**ODS** Output Driver Select.  
Defines the function of the transmit data pin (TxDA/B).  
0...TxDA/B-pin open drain output  
1...TxDA/B-pin push-pull output

*Note: This bit has to be programmed for SIDE0 even if SIDE0 is not used. This bit is without function in SIDE0 1...3*

**ITF** Inter frame Time Fill.  
Determines the “no data to send” state of the transmit data pin (TxDA/B).  
0...continuous IDLE-sequences are output (‘11111111’ bit pattern).  
1...continuous FLAG-sequences are output (‘01111110’ bit pattern).

**CM2** Clock rate.  
0...single rate data clock  
1...double rate data clock (IOM-2)

*Note: This bit has to be programmed for SIDE0 even if SIDE0 is not used. This bit is without function in SIDE0 1...3*

**5.1.2.9 Channel Configuration Register 2 (CCR2)**

Access: read/write

Reset value: 00<sub>H</sub>

bit 7							bit 0
SOC1	0	XCS0	RCS0	0	<b>CTSEN</b>	RIE	0

**SOC1** The function of the TSCA/B-pin can be defined by programming SOC1.

- Point-to-point configuration:
  - 0...the TSCA/B-output is activated during the transmission of a frame.
  - 1...the TSCA/B-output is activated during the transmission of a frame and of inter frame timefill.

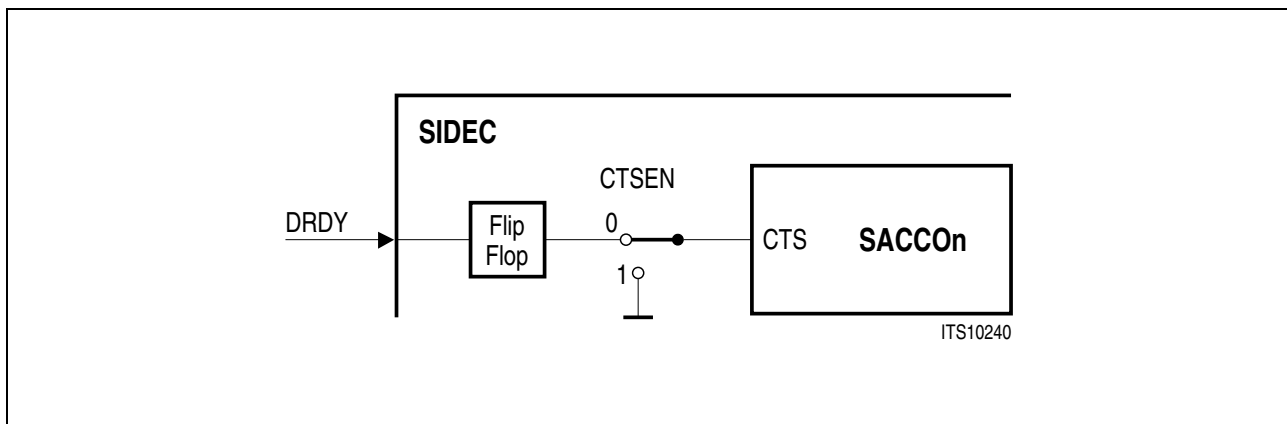
**XCS0,** Transmit/receive bit Shift, bit 0.

**RCS0** Together with the bits XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR) the bit shift relative to the frame synchronization signal of the transmit (receive) time-slot can be adjusted. A bit shift of 0...7 bits is programmable.

**CTSEN** Clear to Send Enable

**Table 5-47**

<b>CTSEN</b>	<b>Function</b>
0	SIDEC channel operation with DRDY signal according to the specification DRDY = 1 "Go" DRDY = 0 "Stop"
1	SIDEC is enabled; DRDY is disconnected. Collision detection is not applicable in this mode.



**Figure 5-5 Use of CTS Signal in SIDEc**

**RIE** Receive frame start Enable.  
When set, the RFS-interrupt in register EXIR\_A/B is enabled.

**5.1.2.10 Receive Length Check Register (RLCR)**

Access: write  
Reset value: 0xxxxxxx<sub>H</sub>

bit 7							bit 0
RC	RL6	RL5	RL4	RL3	RL2	RL1	RL0

**RC** Receive Check enable.  
A '1' enables, a '0' disables the receive frame length feature.

**RL6...0** Receive Length.  
The maximum receive length after which data reception is suspended can be programmed in RL6...0. The maximum allowed receive frame length is (RL + 1) × 32 bytes. A frame exceeding this length is treated as if it was aborted by the opposite station (RME-interrupt, RAB-bit set).  
In this case the receive byte count (RBCH, RBCL) is greater than the programmed receive length.

**5.1.2.11 Status Register (STAR)**

Access: read  
Reset value: 48<sub>H</sub>

bit 7							bit 0
XDOV	XFW	AREP/ XREP	RFR	RLI	CEC	XAC	AFI

Description of Registers

- XDOV** Transmit Data Overflow.  
A '1' indicates, that more than 32 bytes have been written into the XFIFO.
  - XFW** XFIFO Write enable.  
A '1' indicates, that data can be written into the XFIFO.
- Note: XFW is only valid when CEC = 0.*
- AREP/** Auto Repeat/Transmission Repeat.
  - XREP** Read back value of the corresponding command bit CMDR:AREP/XREP.
  - RFR** RFIFO Read enable.  
A '1' indicates, that valid data is in the RFIFO and read access is enabled.  
RFR is set with the RME- or RPF-interrupt and reset when executing the RMC-command.
  - RLI** Receiver Line Inactive.  
Neither flags as inter frame time fill nor frames are received via the receive line.
- Note: Significant in point-to-point configurations!*
- CEC** Command Execution.  
When '0' no command is currently executed, the CMDR-register can be written to.  
When '1' a command (written previously to CMDR) is currently executed, no further command must temporarily be written to the CMDR-register.
  - XAC** Transmitter Active.  
A '1' indicates, that the transmitter is currently active.
  - AFI** Additional Frame Indication.  
A '1' indicates, that one or more completely received frames or the last part of a frame are in the CPU inaccessible part of the RFIFO.  
In combination with the bit STAR:RFR multiple frames can be read out of the RFIFO without interrupt control.

**5.1.2.12 Receive Status Register (RSTA)**

Access: read  
Reset value: xx<sub>H</sub>

bit 7						bit 0	
VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA

RSTA always displays the momentary state of the receiver. Because this state can differ from the last entry in the FIFO it is reasonable to always use the status bytes in the FIFO.

- VFR** Valid Frame.  
Indicates whether the received frame is valid ('1') or not ('0' invalid).

Description of Registers

- A frame is invalid when
- its length is not an integer multiple of 8 bits ( $n \times 8$  bits), e.g. 25 bit,
  - its is too short, depending on the selected operation mode:
    - auto-mode/non-auto mode (2-byte address field): 4 bytes
    - auto-mode/non-auto mode (1-byte address field): 3 bytes
    - transparent mode 1: 3 bytes
    - transparent mode 0: 2 bytes
  - a frame was aborted (note: VFR can also be set when a frame was aborted)

*Note: Shorter frames are not reported.*

- RDO** Receive Data Overflow.  
A '1' indicates, that a RFIFO-overflow has occurred within the actual frame.
- CRC** CRC-Compare Check.  
0: CRC check failed, received frame contains errors.  
1: CRC check o.k., received frame is error free.
- RAB** Receive message Aborted.  
When '1' the received frame was aborted from the transmitting station. According to the HDLC-protocol, this frame must be discarded by the CPU.
- HA1...0** High byte Address compare.  
In operating modes which provide high byte address recognition, the SACCO compares the high byte of a 2-byte address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FEH and FCH (group address). Depending on the result of the comparison, the following bit combinations are possible:  
10...RAH1 has been recognized.  
00...RAH2 has been recognized.  
01...group address has been recognized.

*Note: If RAH1, RAH2 contain the identical value, the combination 00 will be omitted. HA1...0 is significant only in 2-byte address modes.*

- C/R** Command/Response; significant only, if 2-byte address mode has been selected. Value of the C/R bit (bit of high address byte) in the received frame.
- LA** Low byte Address compare.  
The low byte address of a 2-byte address field or the single address byte of a 1-byte address field is compared with two programmable registers (RAL1, RAL2). Depending on the result of the comparison LA is set.  
0...RAL2 has been recognized,  
1...RAL1 has been recognized.  
In non-auto mode, according to the X.25 LAP B-protocol, RAL1/RAL2 may be programmed to differ between COMMAND/RESPONSE frames.



Description of Registers

Note: The receive status byte is duplicated into the RFIFO (clock mode 0-2) following the last byte of the corresponding frame.

**5.1.2.13 Receive HDLC-Control Register (RHCR)**

Access: read

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RHCR7	RHCR6	RHCR5	RHCR4	RHCR3	RHCR2	RHCR1	RHCR0

**RHCR7...0** Receive HDLC-Control Register.

The contents of the RHCR depends on the selected operating mode.

- Auto-mode (1- or 2-byte address field):  
I-frame compressed control field  
    (bit 7-4: bit 7-4 of PBC-command,  
    bit 3-0: bit 3-0 of HDLC-control field)
- else HDLC-control field

Note: RR-frames and I-frames with the first byte = AxH (PBCcommand “transmit prepared data”) are handled automatically and are not transferred to the CPU (no interrupt is issued).

- Non-auto mode (1-byte address field): 2nd byte after flag
- Non-auto mode (2-byte address field): 3rd byte after flag
- Transparent mode 1: 3rd byte after flag
- Transparent mode 0: 2nd byte after flag

Note: The value in RHCR corresponds to the last received frame.

**5.1.2.14 Transmit Address Byte 1 (XAD1)**

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
XAD17	XAD16	XAD15	XAD14	XAD13	XAD12	XAD11	XAD10

**XAD17...10** Transmit Address byte 1.

The value stored in XAD1 is included automatically as the address byte (high address byte in case of 2-byte address field) of all frames transmitted in auto mode.

Using a 2 byte address field, XAD11 and XAD10 have to be set to ‘0’.

### 5.1.2.15 Transmit Address Byte 2 (XAD2)

Access: write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
XAD27	XAD26	XAD25	XAD24	XAD23	XAD22	XAD21	XAD20

#### XAD27...20 Transmit Address byte 2.

The value stored in XAD2 is included automatically as the low address byte of all frames transmitted in auto-mode (2-byte address field only).

### 5.1.2.16 Receive Address Byte Low Register 1 (RAL1)

Access: read/write

Reset value: xx<sub>H</sub>

bit 7						bit 0	
RAL17	RAL16	RAL15	RAL14	RAL13	RAL12	RAL11	RAL10

#### RAL17...10 Receive Address byte Low register 1.

The general function (read/write) and the meaning or contents of this register depends on the selected operating mode:

- Auto-mode, non-auto mode (address recognition) - write only: compare value 1, address recognition (low byte in case of 2-byte address field).
- Transparent mode 1 (high byte address recognition) - read only: RAL1 contains the byte following the high byte of the address in the received frame (i.e. the second byte after the opening flag).
- Transparent mode 0 (no address recognition) - read only: contains the first byte after the opening flag (first byte of the received frame).
- Extended transparent mode 0,1 - read only: RAL1 contains the actual data byte currently assembled at the RxD-pin by passing the HDLC-receiver (fully transparent reception without HDLC-framing).

*Note: In auto-mode and non-auto mode the read back of the programmed value is inverted.*

### 5.1.2.17 Receive Address Byte Low Register 2 (RAL2)

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RAL27	RAL26	RAL25	RAL24	RAL23	RAL22	RAL21	RAL20
-------	-------	-------	-------	-------	-------	-------	-------

**RAL27...20** Receive Address byte Low register 1.

- Auto-mode, non-auto mode (address recognition): compare value 2, address recognition (low byte in case of 2-byte address field).

*Note: Normally used for broadcast address.*

### 5.1.2.18 Receive Address Byte High Register 1 (RAH1)

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RAH17	RAH16	RAH15	RAH14	RAH13	RAH12	0	0
-------	-------	-------	-------	-------	-------	---	---

**RAH17...12** Receiver Address byte High register 1.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 1, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH1 must be set to 00<sub>H</sub>.*

### 5.1.2.19 Receive Address Byte High Register 2 (RAH2)

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

RAH27	RAH26	RAH25	RAH24	RAH23	RAH22	0	0
-------	-------	-------	-------	-------	-------	---	---

**RAH27...22** Receiver Address byte High register 2.

- Auto-mode, non-auto mode transparent mode 1, (2-byte address field). Compare value 2, high byte address recognition.

*Note: When a 1-byte address field is used in non-auto or auto-mode, RAH2 must be set to 00<sub>H</sub>.*

**5.1.2.20 Receive Byte Count Low (RBCL)**

Access: read

Reset value: 00<sub>H</sub>

bit 7				bit 0			
RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0

**RBC7...0** Receive Byte Count.  
 Together with RBCH (bits RBC11-RBC8), the length of the actual received frame (0...4095 bytes) can be determined. These registers must be read by the CPU following a RME interrupt.

**5.1.2.21 Receive Byte Count High (RBCH)**

Access: read

bit 7				bit 0			
0	0	0	OV	RBC11	RBC10	RBC9	RBC8

**OV** Counter Overflow.  
 A '1' indicates that more than 4095 bytes were received.  
 The received frame exceeded the byte count in RBC11...RBC0.

**RBC11...8** Receive Byte Count high.  
 Together with RBCL (bits RBC7...RBC0) the length of the received frame can be determined.

**5.1.2.22 Time-Slot Assignment Register Transmit (TSAX)**

Access: write

Reset value: xx<sub>H</sub>

bit 7				bit 0			
TSNX5	TSNX4	TSNX3	TSNX2	TSNX1	TSNX0	XCS2	XCS1

**TSNX5...0** Time-Slot Number Transmit.  
 Selects one of up to 64 time-slots (00<sub>H</sub>-3F<sub>H</sub>) in which data is transmitted. The number of bits per time-slot is programmable in register XCCR.

**XCS2...1** Transmit bit Shift bit2-1.  
 Together with XCS0 in register CCR2 the transmit bit shift can be adjusted.

**5.1.2.23 Time-Slot Assignment Register Receive (TSAR)**

Access: write

Reset value: xx<sub>H</sub>

bit 7

bit 0

TSNR5	TSNR4	TSNR3	TSNR2	TSNR1	TSNR0	RCS2	RCS1
-------	-------	-------	-------	-------	-------	------	------

**TSNR5...0** Time-Slot Number Receive.

Selects one of up to 64 time-slots (00<sub>H</sub> - 3F<sub>H</sub>) in which data is received. The number of bits per time-slot is programmable in register RCCR.

**RCS2...1** Receive bit Shift bit2-1.

Together with RCS0 in register CCR2 the transmit bit shift can be adjusted.

**5.1.2.24 Transmit Channel Capacity Register (XCCR)**

Access: write

Reset value: 00<sub>H</sub>

bit 7

bit 0

XBC7	XBC6	XBC5	XBC4	XBC3	XBC2	XBC1	XBC0
------	------	------	------	------	------	------	------

**XBC7...0** Transmit Bit Count.

Defines the number of bits to be transmitted in a time-slot in clock mode 2 (number of bits per time-slot = XBC + 1 (1...256 bits/time-slot)).

*Note: In extended transparent mode the width of the time-slot has to be n × 8 bits.*

**5.1.2.25 Receive Channel Capacity Register (RCCR)**

Access: write

Reset value: 00<sub>H</sub>

bit 7

bit 0

RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0
------	------	------	------	------	------	------	------

**RBC7...0** Receive Bit Count.

Defines the number of bits to be received in a time-slot.

Number of bits per time-slot = RBC + 1 (1...256 bits/time-slot).

*Note: In extended transparent mode the width of the time-slot has to be n × 8 bits.*

**5.1.2.26 Version Status Register (VSTR)**

Access: read

bit 7				bit 0			
1	0	0	0	VN3	VN2	VN1	VN0

**VN3...0** SACCO Version Number.  
 82<sub>H</sub>...SIDEK in DOC V1.1.  
 83<sub>H</sub>...SIDEK in DOC V2.1.

## 6 Applications

As the DOC is a powerful device it is possible to show only a few of the possible system configurations. Because the DSP is connected to one or to two CFI ports, which can be programmed to operate in IOM-2 or in PCM mode, and the user also can program the number of B-channels the DSP is supposed to use ( $n \times 4$  time-slots,  $n = 0$  to 16), it is difficult to count how many IOM-2 ports are fully or partly usable for layer-1 IC connection – this strongly depends on the specific application.

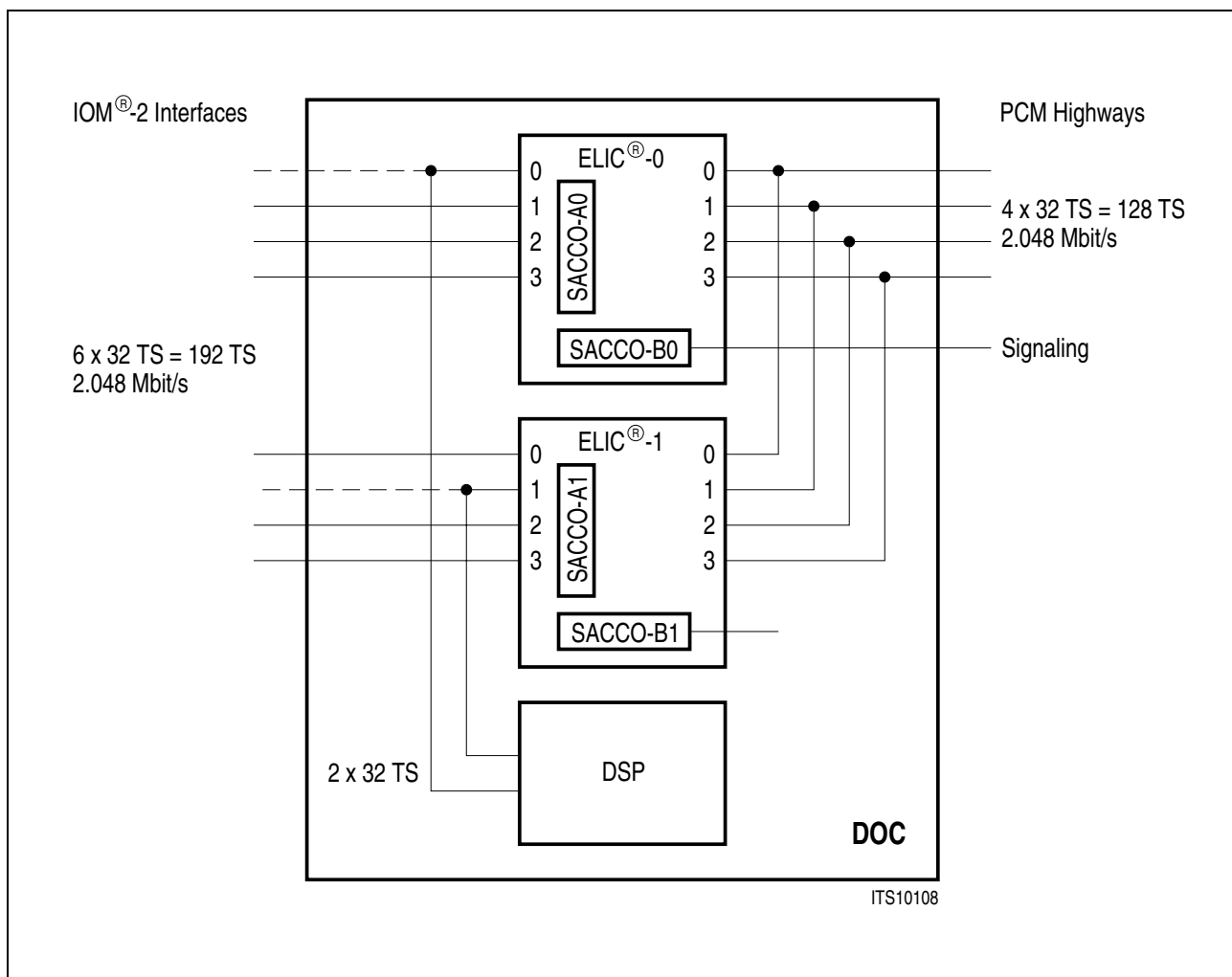
### 6.1 DOC in a Small PBX

The application is characterized by a high number of IOM-2 channels and a low number of time-slots at the PCM highway.

#### 6.1.1 Small PBX with 2.048 Mbit/s Data Rate

In this mode, the DOC provides:

- 6 fully usable IOM-2 (GCI) Interfaces with 48 IOM-2 subframes ( $6 \times 8$ ) and thus it can control up to 48 ISDN or 96 analogue subscribers.
- 2 partly usable IOM-2 Interfaces, as the two DSP ports are connected to them.
- 4 PCM highways with 128 timeslots, as the two ELICs must be connected together.



**Figure 6-1 DOC in a Small PBX with 2.048 Mbit/s Data Rate**

*Note: SACCO-B1 can be used as signaling controller but not as a stand alone controller.*



### 6.1.2 Small PBX with 4.096 Mbit/s Data Rate

In this mode, the DOC provides:

- 2 fully usable IOM-2 Interfaces with 32 IOM-2 subframes ( $2 \times 16$ ) and
- 2 partly usable IOM-2 Interfaces with 16 IOM-2 subframes ( $2 \times 8$ ) and thus
- control of up to 48 ISDN or 96 analogue subscribers.
- 2 PCM highways with 128 timeslots, as the two ELICs must be connected together.

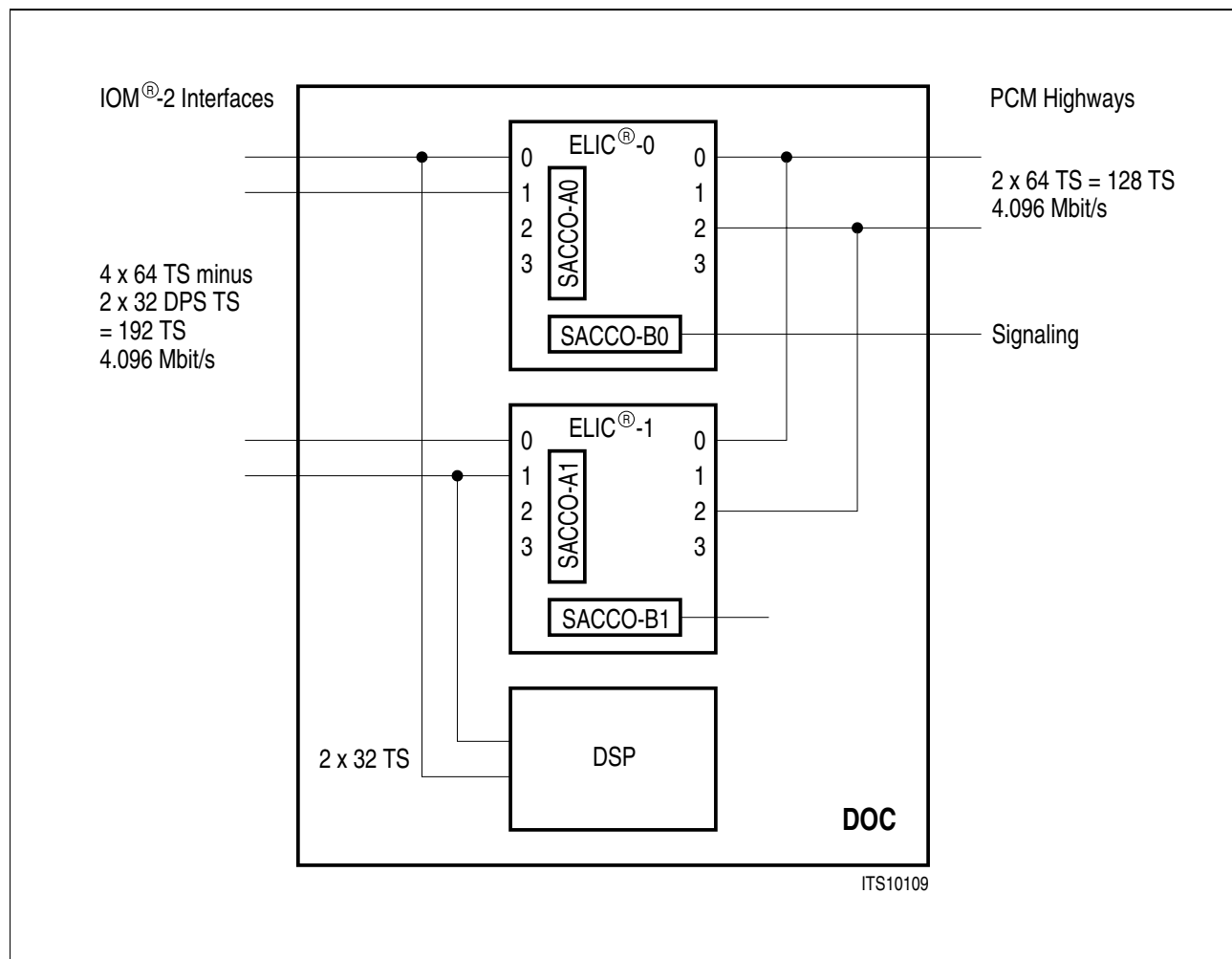


Figure 6-2 DOC in a Small PBX with 4.096 Mbit/s Data Rate

## 6.2 DOC on Line Card

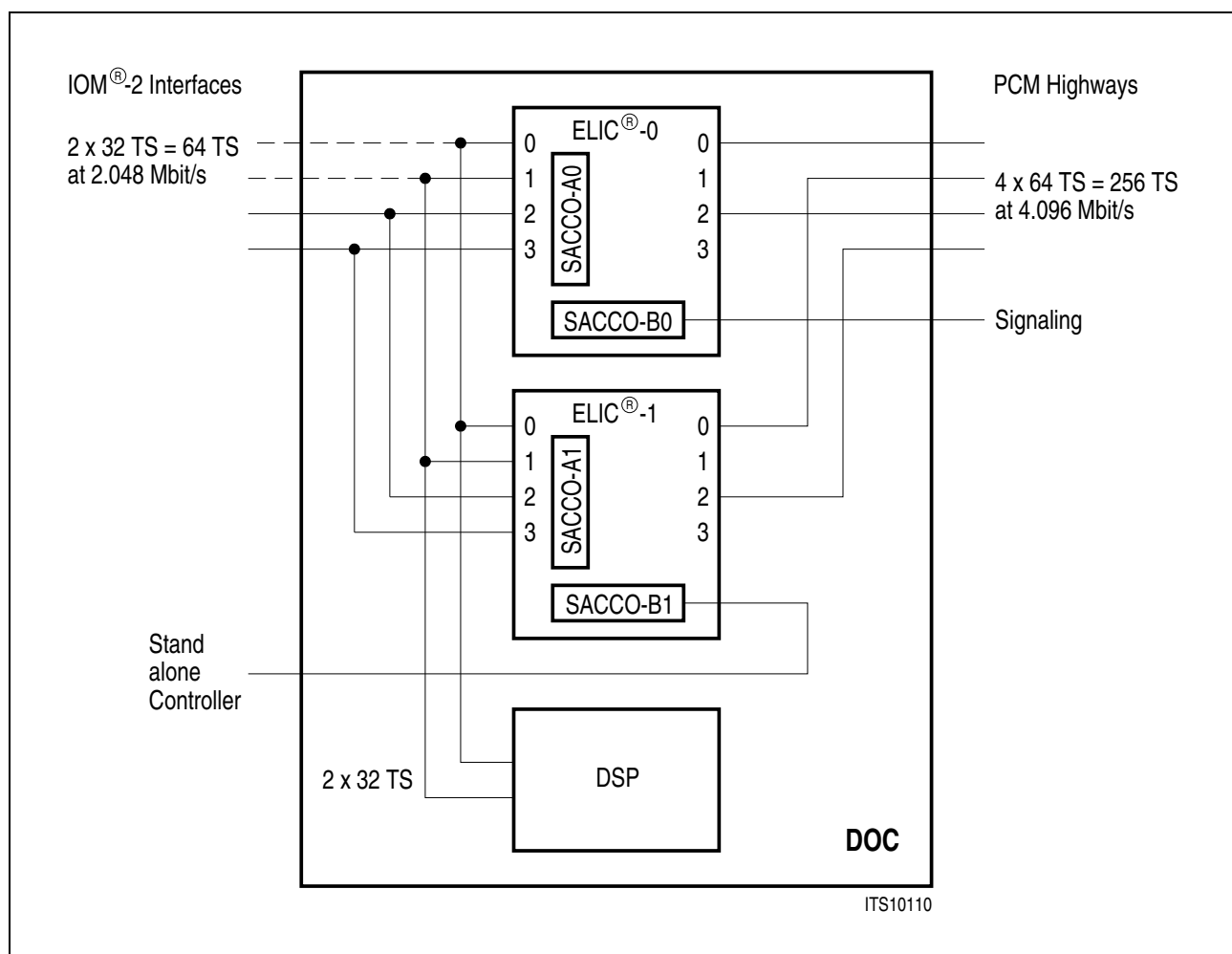
This mode is characterized by a low number of supported IOM-2 subframes and a high number of needed time-slots at the PCM highway.

As the DOC is complex it is possible to show only a few of all possible configurations.

### 6.2.1 Line Card with 2.048 and 4.096 Mbit/s Data Rates

In this mode, the DOC provides:

- 2 fully usable IOM-2 Interfaces with 16 IOM-2 subframes ( $2 \times 8$ ) and
- 2 limited IOM-2 Interfaces, as two DSP ports are connected.
- 4 PCM highways with 256 time-slots ( $4 \times 64$ ).



**Figure 6-3 DOC on Line Card with 2.048 and 4.096 Mbit/s Data Rates**

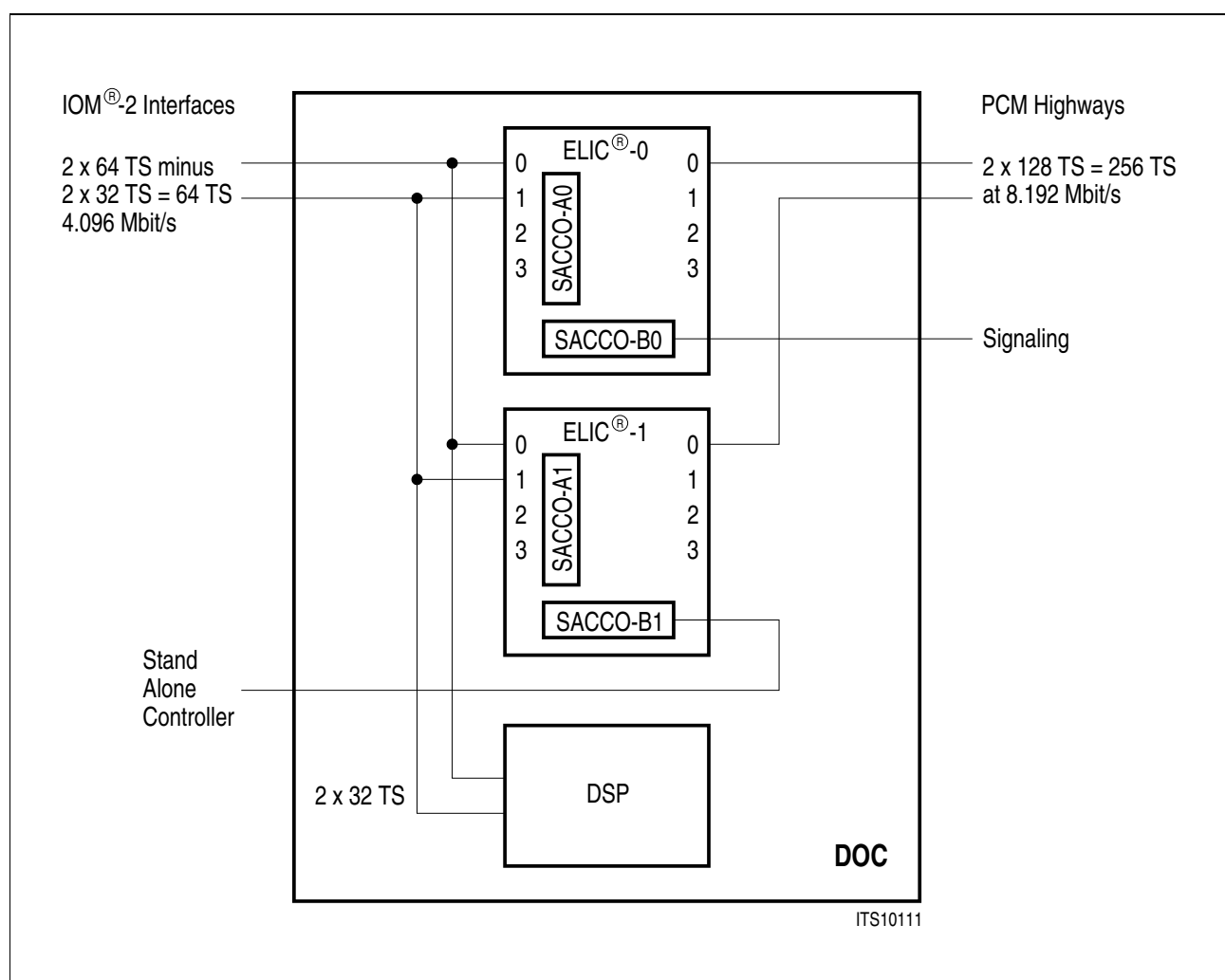
*Note: SACCO-B1 can be used also as a stand alone controller with all support lines as the ELIC1 IOM-2 ports are not used in this mode.*

### 6.2.2 Line Card with 4.096 and 8.192 Mbit/s Data Rates

In this mode, the DOC provides:

- 2 partly usable IOM-2 Interfaces with 16 IOM-2 subframes ( $2 \times 16$  minus  $2 \times 8$ ) as the two DSP ports are connected.
- 2 PCM highways with 256 time-slots ( $2 \times 128$ ).

*Note: SACCO-B1 can be used also as a stand alone controller with all support lines as the ELIC1 IOM-2 ports are not used in this mode.*

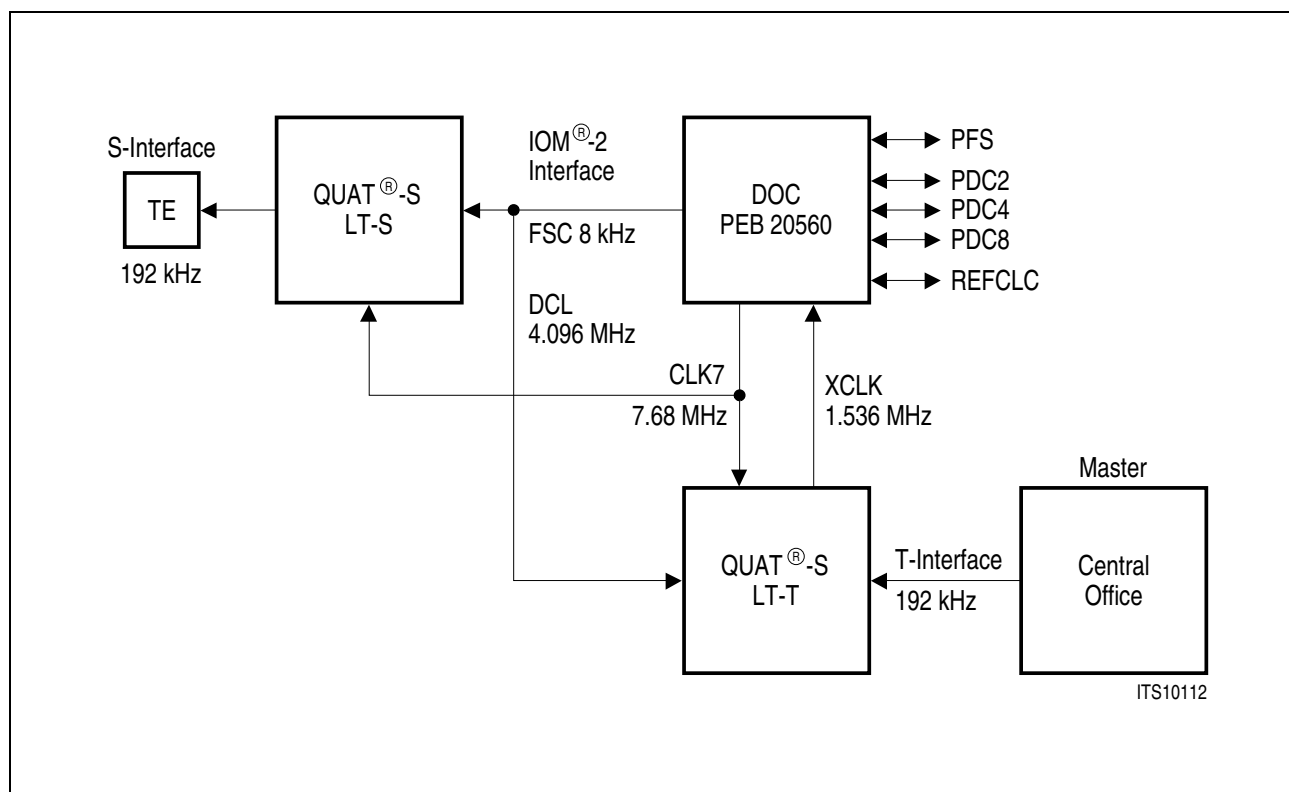


**Figure 6-4 DOC on Line Card with 4.096 and 8.192 Mbit/s Data Rates**

### 6.3 Clock Generation

#### 6.3.1 PBX with one DOC

An external reference clock is selected, e.g. 1.536 MHz from QUAT-S in LT-T mode.



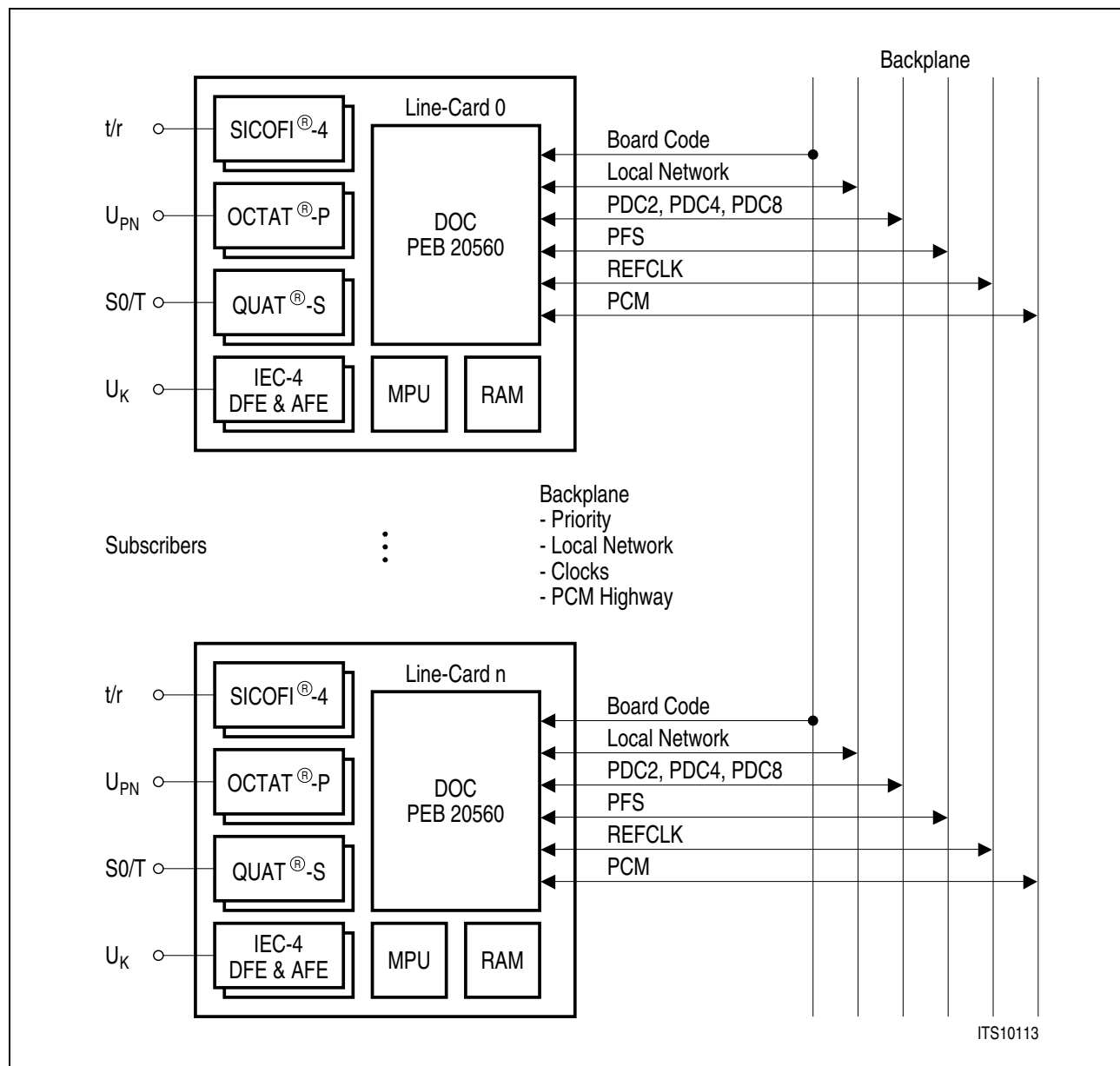
**Figure 6-5 Clock Generation in a PBX with one DOC**

The QUAT-S provides a synchronous 1.536 MHz clock (adaptive timing recovery). A synchronized DOC generates PFS, PDC2, PDC4, PDC8, FSC and DCL.

The PFS length must be at least 3 DCL period long when the PFS is used for IOM-2 synchronization (instead of FSC). A short pulse would be interpreted as a sync. pulse for Multiframe synchronization.

### 6.3.2 PBX with Multiple DOCs

PBXs may use multiple line cards with one DOC on each line card.



**Figure 6-6 Clock Synchronization in a PBX with Multiple DOCs**

The following initialization sequence is recommended after reset:

1. One DOC, selected by the  $\mu P$  as system master, generates free running master clocks PFS and PDC2/4/8 used for FSC and DCL generation on all DOCs/line cards. The system master is selected by the  $\mu P$  via a line card code (or DOC code).
2. One DOC, connected to the central office is selected by the  $\mu P$  as clock master. This DOC synchronizes to the central office clock via its trunk line (refer to the **Figure 6-5**) and generates a reference clock REFCLK for system master resynchronization. REFCLK = XCLK divided by 4 or 3 or it equals XCLK.

The clock master is selected by the  $\mu$ P via the integrated Local Network Controller (LNC).

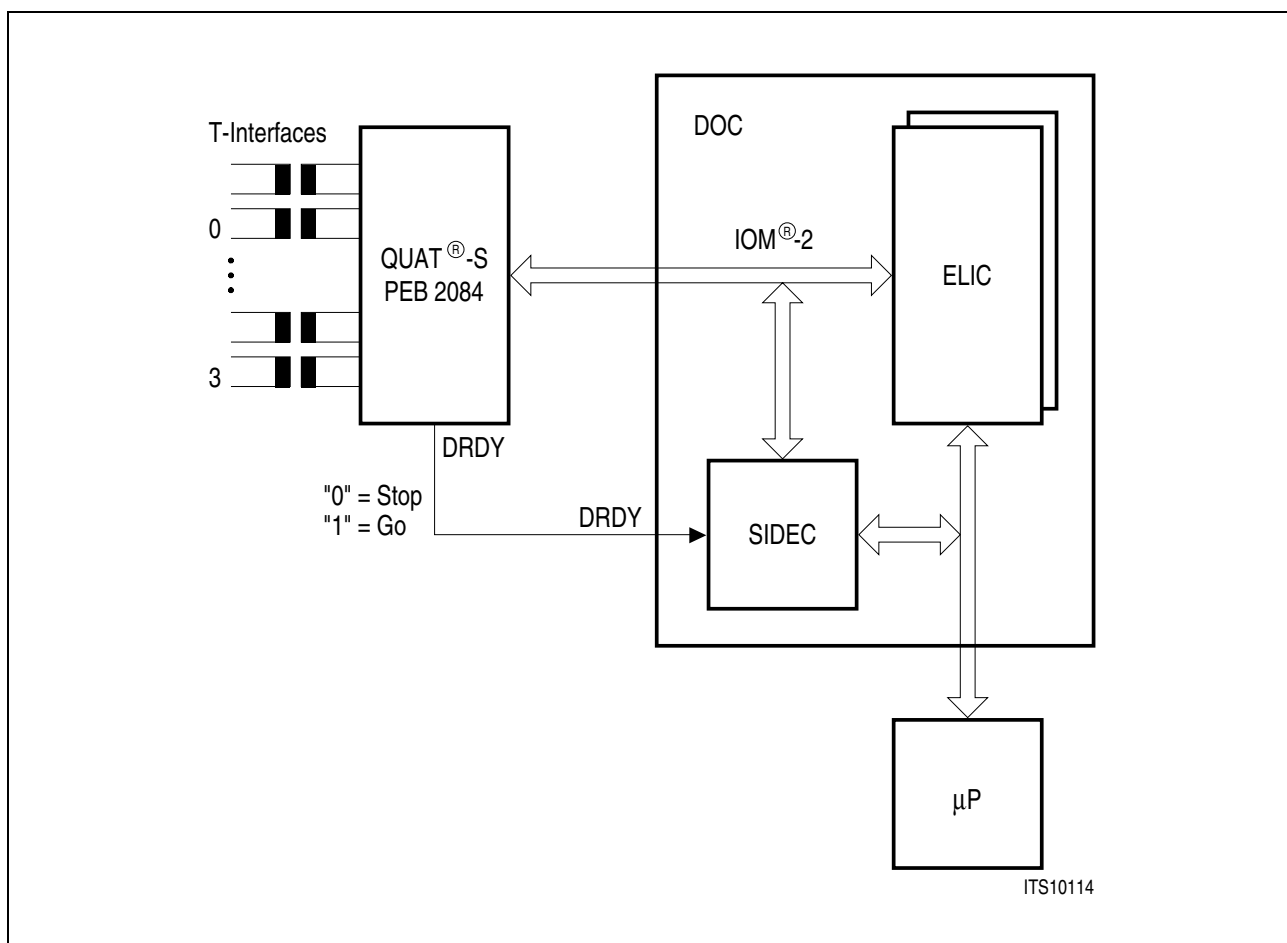
3. The system master synchronizes its master clocks PFS and PDC2/4/8 to the REFCLK.
4. All other DOCs “resynchronize” FSC and DCL (shift by a delta) to the system master. The DOC hardware will be optimized so that the phase shift will be minimal and constant for all DOCs.

### 6.4 Signaling with SIDEC

The four independent communication channels of the SIDEC can be used either for data communication with terminals (or PCs) or in connection with QUAT-S for communication to the central office.

If the QUAT-S is used for LT-T applications, the SIDEC must be assigned with all involved channels to one IOM-2 interface. See **Figure 6-7**.

The pin DRDY conveys control information synchronously to the D-channel time-slots to control the SIDEC. Refer also to **Figure 2-22** and **Figure 2-23**.



**Figure 6-7 QUAT-S in LT-T Mode with SIDEC for Four-Channel Trunk Applications**

## 6.5 Local Network Controller (SACCO-B1 as LNC)

The LNC can be used:

- For inter DOC communication when multiple DOCs are cascaded (i.e. for DOC selection to generate REFCLK)
- For signaling, if externally connected to IOM-2 or PCM interfaces
- As a general purpose HDLC Controller (up to 8.192 Mbit/s).  
The max. length of the transmit line may reach up to 2 m.

The high speed data rate may be handled by an external DMA controller.

## 6.6 UART Applications

The UART can be used for:

- System tests in the development phase
- Field tests
- Program download (i.e. via a V.24 interface)

## 6.7 IOM<sup>®</sup>-2 Channel Indication Signal (CHI)

The CHI signal in addition to the IOM-2 signals can be used for indication to any connected Layer-1 IC that the data sent downstream is not IOM-2 compatible data. By the use of CHI signal a proprietary data channel can be implemented.

## 6.8 Use of FSCD

In case of connecting 2 OCTAT-P or four QUAT-S to a 4.096 Mbit/s IOM-2 interface (extended IOM-2 specification) a second FSC, delayed by 62.5  $\mu$ s, is provided. It synchronizes the Layer-1 ICs connected to the time-slots 32 to 63. Layer-1 ICs connected to the time-slots 0 to 31 use the standard FSC signal.

## 6.9 Watch-Dog Activation

1. Activation (A disactivation by software is not possible after its activation)
2. Triggering both Watch-Dog Registers by software
3. Flag Reset Indication if software triggering fails; however DOC remains unaffected

## 6.10 Tone and Voice Processing

The integrated DSP (OAK) is supposed to execute routines for tone and voice processing such as:

- Tone generation and recognition
- DTMF transmitter and receiver
- Music on hold
- Conferencing
- Voice mail and voice recording

(a-/μ-Law coding and decoding is performed by firmware, independently of the DSP)

**Table 6-1 An Example for Required DSP Performance in a Comfort PBX with 30 Subscribers**

DSP Functionality	Performance		
	MIPS per Channel	Channels/ Operation	MIPS Total
DTMF Generator	0.37	4	1.48
DTMF Receiver	1.8	8	14.4
Tone Generator	0.2	4 <sup>1)</sup>	0.8
Tone Receiver (from CO)	0.4	4	1.6
Music on Hold	0.2	2 <sup>2)</sup>	0.4
Conferencing	0.7	4 × 5	2.8
Modem V.21 (300 Baud) <sup>3)</sup>	4.0	1	4.0
Operating System	2.0		2.0
			27 MIPS

<sup>1)</sup> Country specific number

<sup>2)</sup> User specific

<sup>3)</sup> Modem for exceptional use

As not all DSP routines are running at the same time and the integrated DSP provides 40 MIPS, additional DSP routines can be implemented by the user. For monitoring and optimizing the DSP load, the programmer can use the Statistics Register.

Siemens also provides a PC based expert system for fast and correct DOC initialization, refer to DOC Configurator, **chapter 9.6**.



### 6.11 DSP Frequency Recommendation

Depending on the speed (price) of the connected external memory (SRAM) the following maximal DSP frequency and thus DSP performance (number of MIPS) is possible; estimation:

SRAM Access Time <sup>1)</sup>	Max. DSP Frequency	Note
33.5 ns	20 MHz	DOC requires an ext. quartz <sup>2)</sup>
22 ns	26 MHz	DOC requires an ext. quartz
16.8 ns	30 MHz	DOC requires an ext. quartz <sup>3)</sup>
13.8 ns	33 MHz	<sup>3)</sup>
10.5 ns	37 MHz	DOC requires an ext. quartz
8.5 ns	40 MHz <sup>4)</sup>	DOC requires an ext. quartz

<sup>1)</sup> SRAM access time is the time from valid read address to the valid data.

<sup>2)</sup> The DOC provides 20.48 MHz

<sup>3)</sup> The DOC provides 30.72 MHz

<sup>4)</sup> The highest DSP frequency of 40 MHz (and all above recommended max. DSP frequencies can only be confirmed after DOC characterization).

Refer also to External Data Read Access Timing, **Table 7-24**.

## 7 Electrical Characteristics

### 7.1 Absolute Maximum Ratings

Table 7-1

Parameter	Symbol	Limit Values	Unit
Ambient temperature under bias	PEB $T_A$	0 to 70	°C
	PEF $T_A$	-40 to 85	°C
Storage temperature	$T_{stg}$	-65 to 125	°C
IC supply voltage	$V_{DD}$	-0.4 to 4.6	V
Protection supply voltage	$V_{DDP}$	-0.5 to 5.5	V
Voltage on any pin with respect to ground	$V_S$	-0.4 to $V_{DDP} + 0.4$	V
Maximum current on all lines connected to the backplane when the DOC is without power supply; at 5.5 V external signal level	$I_{max}$	2.3	mA

*Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

### 7.2 Operating Range

Table 7-2

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Ambient temperature	$T_A$	0	70	°C	
Supply voltage	$V_{DD}$	3.13	3.6	V	
Protection supply voltage	$V_{DDP}$	4.5	5.5	V	
	$V_{SS}$	0	0	V	

*Note: In the operating range, the functions given in the circuit description are fulfilled*

## 7.3 DC Characteristics

Table 7-3

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Input low voltage	$V_{IL}$	-0.4	0.8	V	
Input high voltage	$V_{IH}$	2.0	$V_{DDP} + 0.4$	V	
Output low voltage	$V_{OL}$		0.45	V	$I_{OL} = 7 \text{ mA}$ <sup>1)</sup> $I_{OL} = 2 \text{ mA}$ <sup>2)</sup>
Output high voltage	$V_{OH}$	2.4		V	$I_{OH} = -1.0 \text{ mA}$
Typical power supply current	$I_{CC} (AV)$		115	mA	$V_{DD} = 3.3 \text{ V}$ , $T_A = 25^\circ \text{ C}$ : PDC = 8 MHz HDC = 4 MHz DSP Clock = 30 MHz
Input leakage current	$I_{IL}$		1	$\mu\text{A}$	$V_{DD} = 3.3 \text{ V}$ , GND = 0 V; all other pins are floating; $V_{IN} = 0 \text{ V}$ , $V_{DDP} + 0.4$
Output leakage current	$I_{OZ}$		1	$\mu\text{A}$	$V_{DD} = 3.3 \text{ V}$ , GND = 0 V; $V_{OUT} = 0 \text{ V}$ , $V_{DDP} + 0.4$

<sup>1)</sup> Apply to the next pins: TxDB0, DD0, DD1, DD2, DD3, DD4/TxDB1, DD5/ $\overline{\text{TSCB1}}$ , DD6/DRQTB1, DD7/DRQRB1, TXD0, TXD1, TXD2, TXD3.

<sup>2)</sup> Apply to all the I/O and O pins that do not appear in the list in note 1), except CLK40-XO.

*Note: The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at  $T_A = 25^\circ \text{ C}$  and the given supply voltage.*

## 7.4 Capacitances

**Table 7-4**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock input capacitance	$C_{XIN}$			pF	$f_c = 1$ MHz The pins, which are not under test, are connected to GND
Clock output capacitance	$C_{XOUT}$			pF	
Input capacitance	$C_{IN}$			pF	
Output capacitance	$C_{OUT}$			pF	

## 7.5 Strap Pins Pull-up Resistors Specification

The strap input pins are sampled during reset, and determine some modes of the DOC. The strap inputs are: CDB0/BOOT, CDB1/DBG, CDB2/ROM, CDB4/URST and CDB12/SEIBDIS.

When a strap input pin is not driven externally during reset, it is driven internally by an internal pull-down. If a fixed external pull-up is applied on a strap, a pull-up resistor of 5 k $\Omega$  is required. The required precision of the resistor is 10%.

## 7.6 40 MHz External Crystal

**Table 7-5**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock input capacitance	$C_{LK40-XI}$		7	pF	
Clock output capacitance	$C_{LK40-XO}$		7	pF	
Motional capacitance	$C_1$	20		fF	
Shunt	$C_O$	5		pF	
Load	$C_L$	20		pF	
Resonance resistor	$R_1$	50		$\Omega$	

$$C_{LD} = 2 \times C_L - C_{CLK40-X(1 \text{ or } 0)}$$

*Note: Other external crystals may be used (up to 40 MHz). However, the external circuitry must be changed accordingly.*

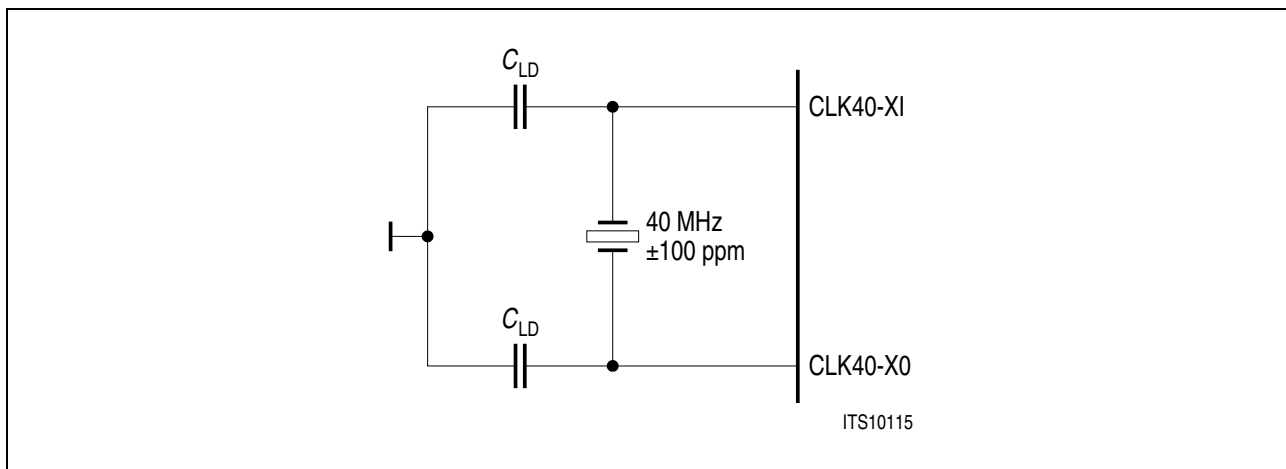


Figure 7-1

### 7.7 General Recommendations and Prohibitions

1. Any DOC input pin should not left disconnected, even if it's unused. CLK-40XI, especially, should be permanently driven by  $V_{SS}$ , if it is unused.

### 7.8 AC Characteristics

Ambient temperature under bias range,  $V_{DD} = 3.13\text{-}3.6\text{ V}$ .

Inputs are driven to 2.4 V for a logic '1' and to 0.4 V for logic '0'.

Timing measurements are made at 2.0 V for a logic '1' and at 0.8 V for a logic '0'.

The AC-testing input/output waveforms are shown below.

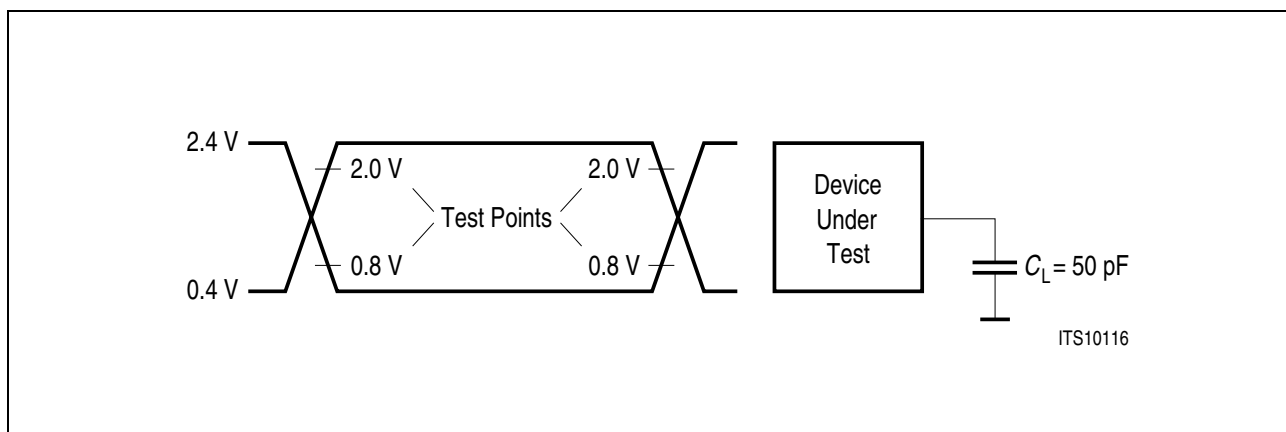
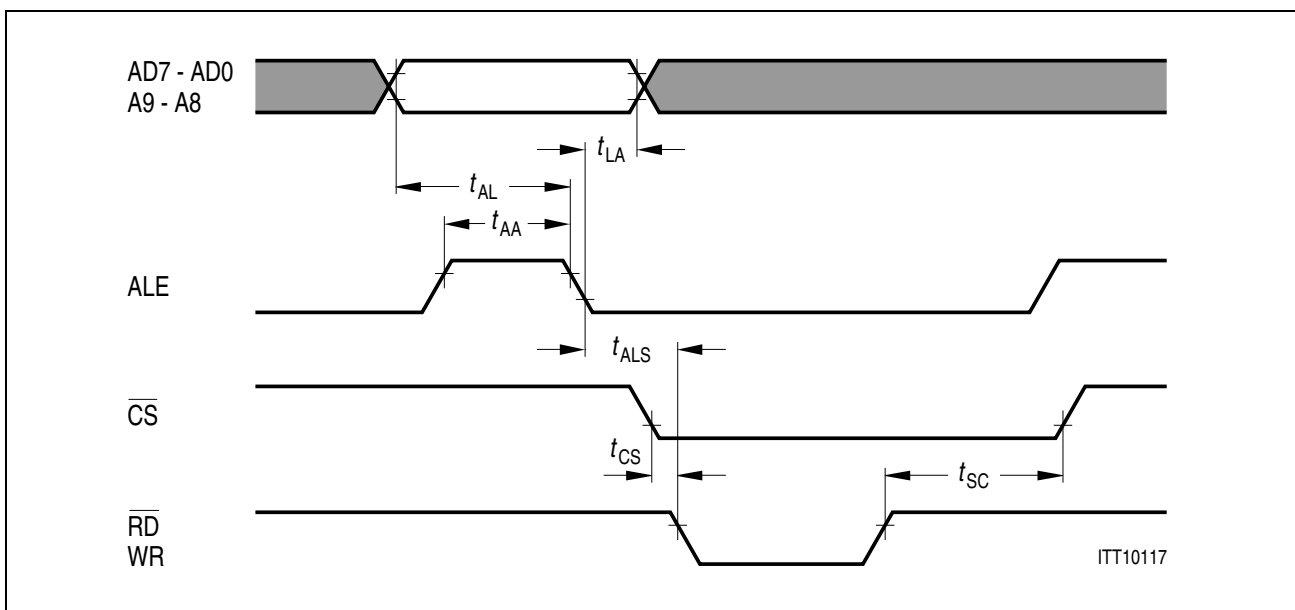


Figure 7-2 I/O-Wave Form for AC-test

### 7.9 Microprocessor Interface Timing

**Table 7-6 Bus Interface Timing**

Parameter	Symbol	Limit Values		Unit
		min.	max.	
$\overline{\text{RD}}$ -pulse width	$t_{\text{RR}}$	80		ns
$\overline{\text{RD}}$ -control interval	$t_{\text{RI}}$	50		ns
Data output delay from $\overline{\text{RD}}$	$t_{\text{RD}}$		65	ns
Data float delay from $\overline{\text{RD}}$	$t_{\text{DF}}$	5	25	ns
DMA-request delay	$t_{\text{DRH}}$		75	ns
$\overline{\text{WR}}$ -pulse width	$t_{\text{WW}}$	30		ns
$\overline{\text{WR}}$ -control interval	$t_{\text{WI}}$	40		ns
Data set-up time to $\overline{\text{WR}}$	$t_{\text{DW}}$	30		ns
Data hold time from $\overline{\text{WR}}$	$t_{\text{WD}}$	15		ns
ALE-pulse width	$t_{\text{AA}}$	15		ns
Address set-up time to ALE	$t_{\text{AL}}$	10		ns
Address hold time from ALE	$t_{\text{LA}}$	8		ns
ALE set-up time to $\overline{\text{WR}}$ , $\overline{\text{RD}}$	$t_{\text{ALS}}$	8		ns
$\overline{\text{CS}}$ set-up time to $\overline{\text{WR}}$ , $\overline{\text{RD}}$	$t_{\text{CS}}$	5		ns
$\overline{\text{CS}}$ hold-time from $\overline{\text{WR}}$ , $\overline{\text{RD}}$	$t_{\text{SC}}$	5		ns



**Figure 7-3 Address Timing**

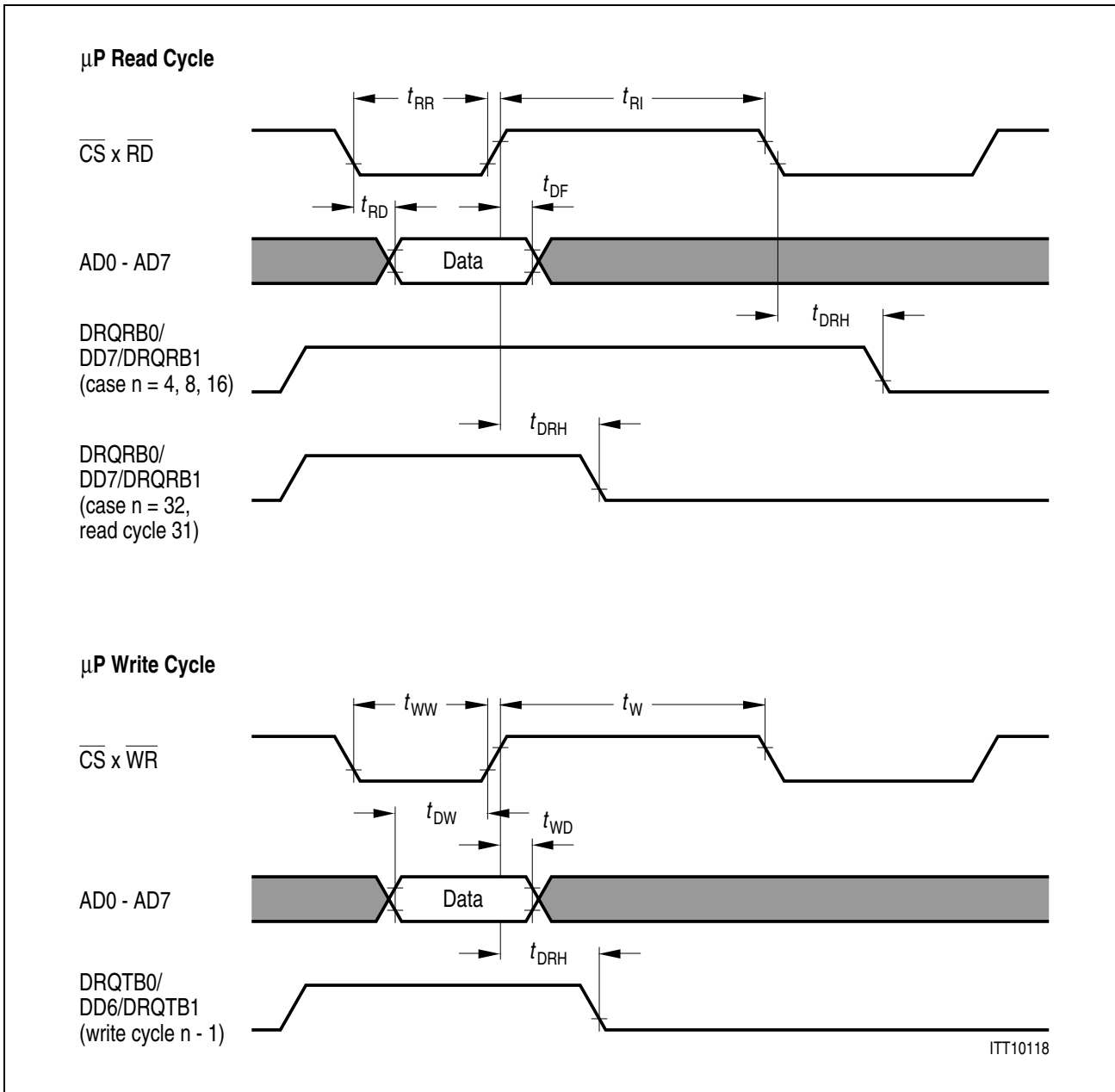
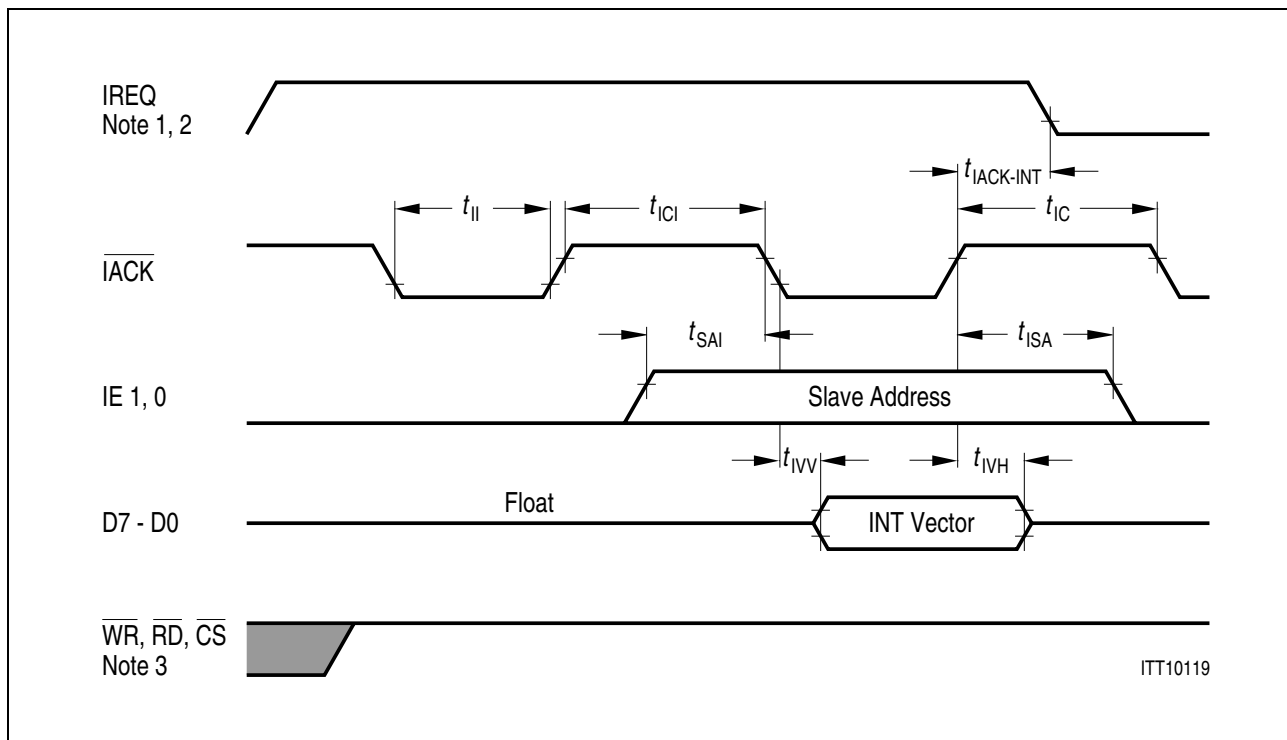


Figure 7-4 Data Timing

**Table 7-7 Siemens/Intel Interrupt Timing**

Parameter	Symbol	Limit Values		Unit
		min.	max.	
$\overline{\text{IACK}}$ pulse width	$t_{\text{II}}$	70		ns
$\overline{\text{IACK}}$ control interval	$t_{\text{ICI}}$	40		ns
IREQ reset after last $\overline{\text{IACK}}$ inactive	$t_{\text{IACK-INT}}$		200	ns
Slave address (IE0, IE1) setup time	$t_{\text{SAI}}$	5		ns
Slave address (IE0, IE1) hold time	$t_{\text{ISA}}$	0		ns
Interrupt vector (D7-D0) valid after $\overline{\text{IACK}}$ active	$t_{\text{IVV}}$		50	ns
Interrupt vector (D7-D0) valid after $\overline{\text{IACK}}$ inactive	$t_{\text{IVH}}$	5	40	ns
IE0 low after IE1 low	$t_{\text{IE10L}}$		20	ns
IE0 high after IE1 high	$t_{\text{IE10H}}$		20	ns
IE0 low after IREQ active	$t_{\text{IRIEOL}}$		10	ns
IREQ inactive after IE1 low	$t_{\text{DISINT}}$		25	ns
IREQ reactivated after IE1 high	$t_{\text{IE1H-INTV}}$		25	ns
IE0 high after IREQ reset	$t_{\text{INT-IE0H}}$		10	ns



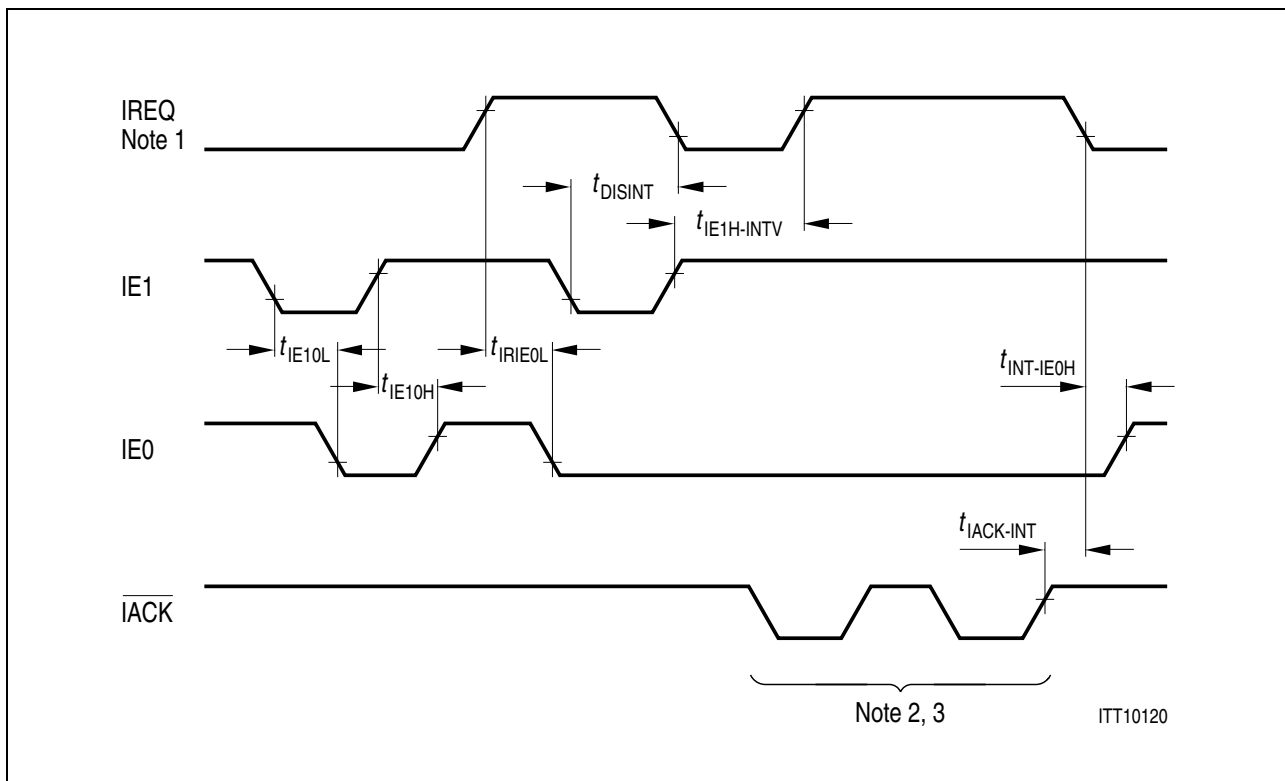


**Figure 7-5 Siemens/Intel Interrupt Timing (Slave mode)**

*Note: 1) The timing is valid for active-high push-pull signal. The timing for active-low push-pull signal is the same.*

*In the case of open drain output, reset time ( $t_{IACK-INT}$ ) depends on external devices.*

- 2)  $t_{IACK-INT}$  is valid only for FSC and RTC interrupts. The other interrupts are reset only by reading/writing from/to the appropriate register, in the interrupt source module (see **Figure 7-9**).*
- 3)  $\overline{WR}$ ,  $\overline{RD}$  and  $\overline{CS}$  must not be activated during  $\overline{IACK}$  activation.*



**Figure 7-6 Siemens/Intel Interrupt Timing (Daisy chaining)**

Note: 1) The timing is valid for active-high push-pull signal. The timing for active-low push-pull signal is the same.

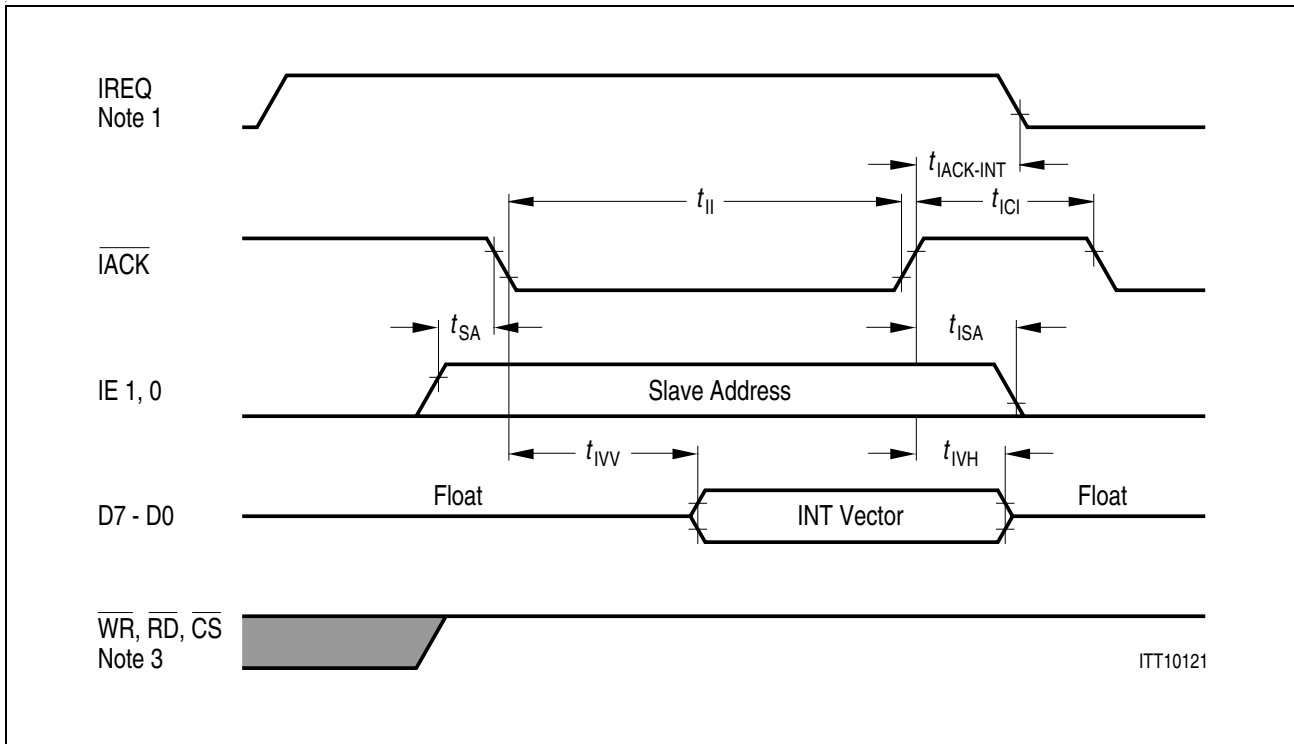
In the case of an open drain output, reset times ( $t_{IACK-INT}$ ,  $t_{DISINT}$ ) depend on external devices.

2) The timing for IREQ,  $\overline{IACK}$  and D7-D0 is similar to slave mode.

3)  $t_{IACK-INT}$  is valid only for FSC and RTC interrupts. The other interrupts are reset only by reading/writing from/to the appropriate register, in the interrupt source module (see **Figure 7-9**).

**Table 7-8 Motorola Interrupt Timing**

Parameter	Symbol	Limit Values		Unit
		min.	max.	
$\overline{\text{IACK}}$ pulse width	$t_{\text{II}}$	85		ns
$\overline{\text{IACK}}$ control interval	$t_{\text{ICI}}$	40		ns
IREQ reset after last $\overline{\text{IACK}}$ inactive	$t_{\text{IACK-INT}}$		200	ns
Slave address (IE0, IE1) setup time	$t_{\text{SAI}}$	5		ns
Slave address (IE0, IE1) hold time	$t_{\text{ISA}}$	0		ns
Interrupt vector (D7-D0) valid after $\overline{\text{IACK}}$ active	$t_{\text{IVV}}$		75	ns
Interrupt vector (D7-D0) valid after $\overline{\text{IACK}}$ inactive	$t_{\text{IVH}}$	5	40	ns
IE0 low after IE1 low	$t_{\text{IE10L}}$		20	ns
IE0 high after IE1 high	$t_{\text{IE10H}}$		20	ns
IE0 low after IREQ active	$t_{\text{IRIEOL}}$		10	ns
IREQ inactive after IE1 low	$t_{\text{DISINT}}$		25	ns
IREQ reactivated after IE1 high	$t_{\text{IE1H-INTV}}$		25	ns
IE0 high after IREQ reset	$t_{\text{INT-IE0H}}$		10	ns

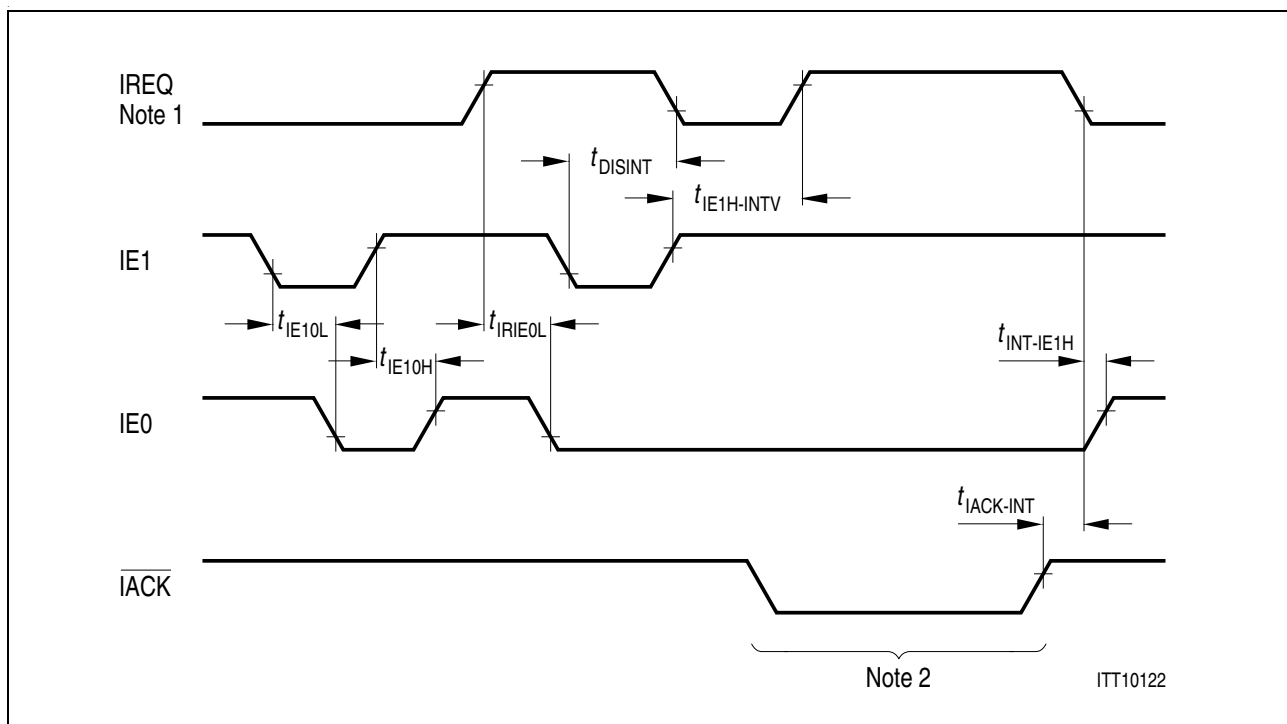


**Figure 7-7 Motorola Interrupt Timing (Slave mode)**

Note: 1) The timing is valid for active-high push-pull signal. Timing for active-low push-pull signal is the same.

In the case of open drain output, reset time ( $t_{IACK-INT}$ ) depends on external devices.

- 2)  $t_{IACK-INT}$  is valid only for FSC and RTC interrupts. The other interrupts are reset only by reading/writing from/to the appropriate register, in the interrupt source module (see **Figure 7-9**).
- 3)  $\overline{WR}$ ,  $\overline{RD}$  and  $\overline{CS}$  must not be activated during  $\overline{IACK}$  activation.



**Figure 7-8 Motorola Interrupt Timing (Daisy chaining)**

Note: 1) The timing is valid for active-high push-pull signal. The timing for active-low push-pull signal is the same.

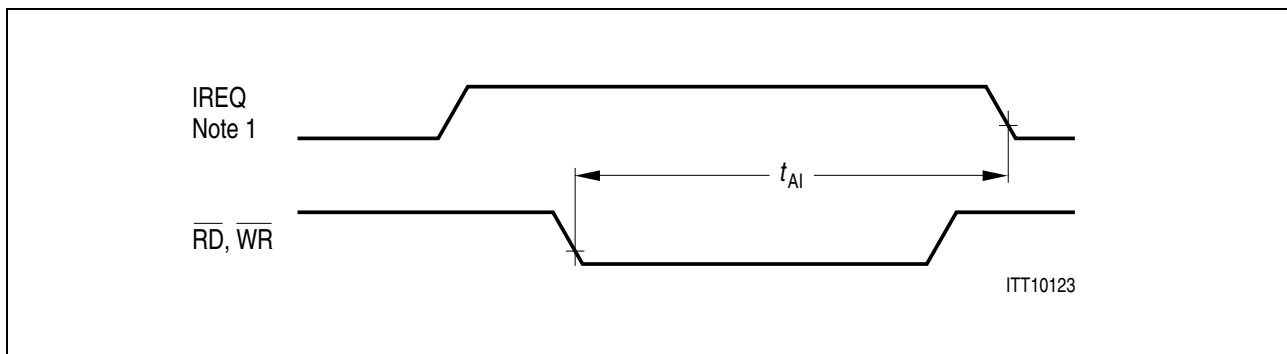
In the case of an open drain output, reset times ( $t_{IACK-INT}$ ,  $t_{DISINT}$ ) depend on external devices.

2) The timing for IREQ,  $\overline{IACK}$  and D7-D0 is similar to slave mode.

3)  $t_{IACK-INT}$  is valid only for FSC and RTC interrupts. The other interrupts are reset only by reading/writing from/to the appropriate register, in the interrupt source module (see **Figure 7-9**).

**Table 7-9 Interrupt Timing**

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Interrupt inactivation delay	$t_{AI}$	200		ns



**Figure 7-9 Interrupt inactivation from  $\overline{WR}$ ,  $\overline{RD}$**

- Note: 1) Timing valid for active-high push-pull signal. Timing for active-low push-pull signal is the same. In the case of open-drain output,  $t_{AI}$  depends on external devices.
- 2) IREQ output signal may be inactivate by read instruction from the appropriate registers in ELIC0/ELIC1, SIDEC 0-4 and the GPIO or write instruction to the appropriate registers in the UART and in the OAK Mail-Box.

### 7.9.1 PCM Interface Timing

**Table 7-10 PDC and PFS Timing In Master Mode**

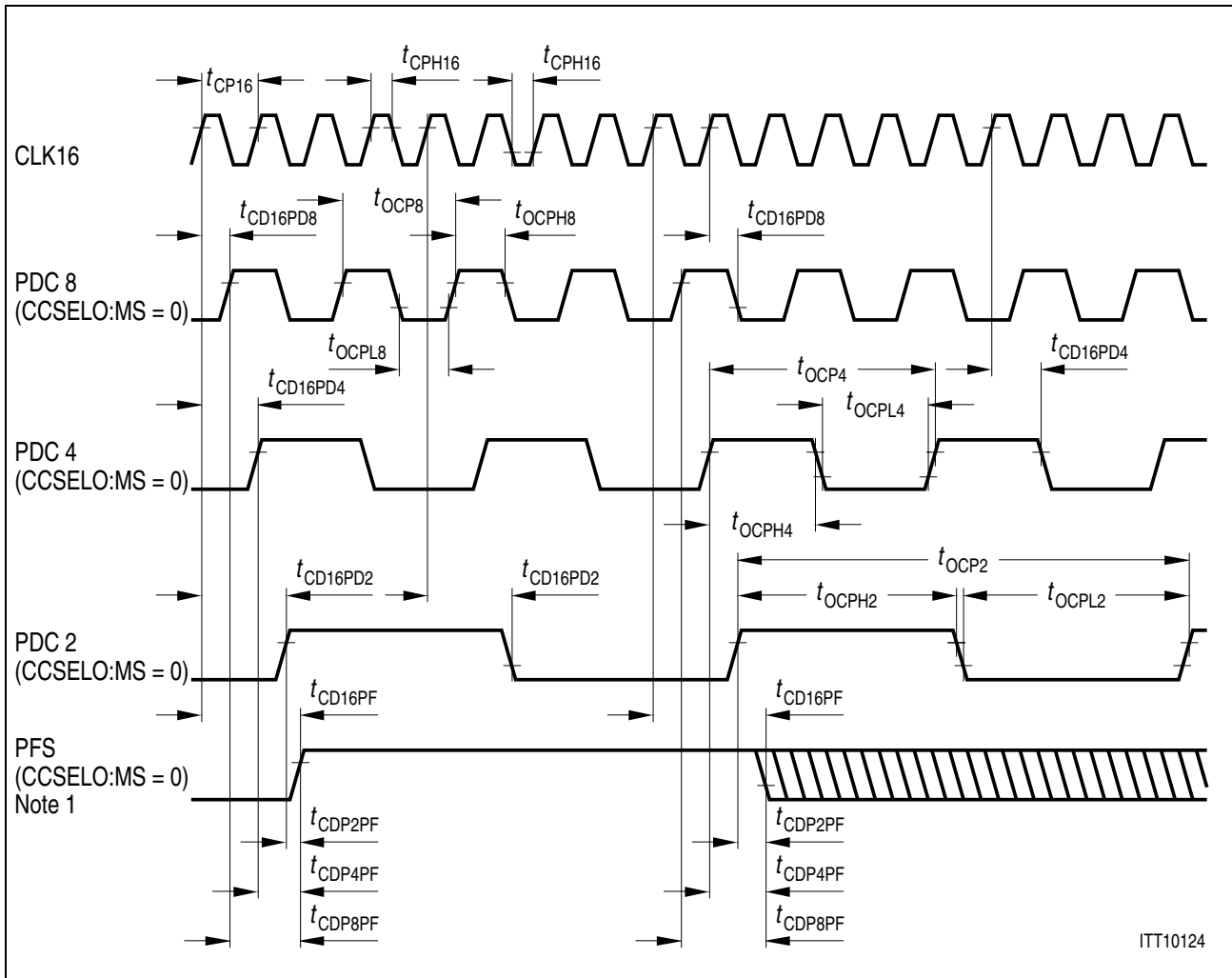
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK16 Clock period	$t_{CP16}$	See <b>Table 7-27</b>			
CLK16 Clock period high	$t_{CPH16}$				
CLK16 Clock period low	$t_{CPL16}$				

Electrical Characteristics

Table 7-10 PDC and PFS Timing In Master Mode

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
PDC8 Output Clock Period	$t_{OCP8}$	120		ns	CLK16 = 16.384 MHz, $C_{LOAD} = 50 \text{ pF}^{1)}$
PDC8 Output Clock Period Low	$t_{OCPL8}$	45		ns	
PDC8 Output Clock Period High	$t_{OCPH8}$	45		ns	
PDC4 Output Clock Period	$t_{OCP4}$	240		ns	
PDC4 Output Clock Period Low	$t_{OCPL4}$	105		ns	
PDC4 Output Clock Period High	$t_{OCPH4}$	105		ns	
PDC2 Output Clock Period	$t_{OCP2}$	480		ns	
PDC2 Output Clock Period low	$t_{OCPL2}$	225		ns	
PDC2 Output Clock Period High	$t_{OCPH2}$	225		ns	
Clock delay CLK16-PDC8	$t_{CD16PD8}$	6	23	ns	
Clock delay CLK16-PDC4	$t_{CD16PD4}$	7	26	ns	
Clock delay CLK16-PDC2	$t_{CD16PD2}$	8	29	ns	
Clock delay CLK16-PFS	$t_{CD16PF}$	13	43	ns	
Clock delay PDC8-PFS	$t_{CDP8PF}$		20	ns	
Clock delay PDC4-PFS	$t_{CDP4PF}$		18	ns	
Clock delay PDC2-PFS	$t_{CDP2PF}$		15	ns	

<sup>1)</sup> When using one DOC as a master of other slave DOC, The  $C_{LOAD}$  of PFS must not be greater then the  $C_{LOAD}$  of PDC 2/4/8 (especially PDC8) in more then 25%



**Figure 7-10 PDC and PFS Timing In Master Mode (PDC & PFS are outputs)**

*Note: When in Master Mode, the pulse width of PFS (PFS period high ) is  
4 PDC2 cycles = 8 PDC4 cycles = 16 PDC8 cycles*

**Table 7-11 PCM-Interface Timing in Master Mode**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock Period	$t_{CP}^{1)}$	<b>See Table 7-10</b>			
Clock Period Low	$t_{CPL}^{2)}$				
Clock Period High	$t_{CPH}^{3)}$				
Frame Delay From PDC	$t_{FPD}^{4)}$				
Serial Data Input Set-Up Time	$t_S$	37		ns	PCM data frequency > 4096 kbit/s
Serial Data Input Hold Time	$t_H$	25		ns	



**Table 7-11 PCM-Interface Timing in Master Mode (cont'd)**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Serial Data Input Set-Up Time	$t_S$	50		ns	PCM data frequency $\leq 4096$ kbit/s
Serial Data Input Hold Time	$t_H$	45		ns	
PCM-Serial Data Output Delay	$t_D$	15	51	ns	
Tri-state Control Delay	$t_T$	5	50	ns	

- 1) For more details about  $t_{CP}$ , see  $t_{OCP8}$ ,  $t_{OCP4}$  and  $t_{OCP2}$ , in **Table 7-10**.
- 2) For more details about  $t_{CPI}$ , see  $t_{OCPI8}$ ,  $t_{OCPI4}$  and  $t_{OCPI2}$ , in **Table 7-10**.
- 3) For more details about  $t_{CPH}$ , see  $t_{OCPH8}$ ,  $t_{OCPH4}$  and  $t_{OCPH2}$ , in **Table 7-10**.
- 4) The  $C_{LOAD}$  of PFS must not be greater than the  $C_{LOAD}$  of PDC2/4/8 (especially PDC8) in more than 25%. For more details about  $t_{FPD}$ , see  $t_{CDP8PF}$ ,  $t_{CDP4PF}$  and  $t_{CDP2PF}$ , in **Table 7-10**

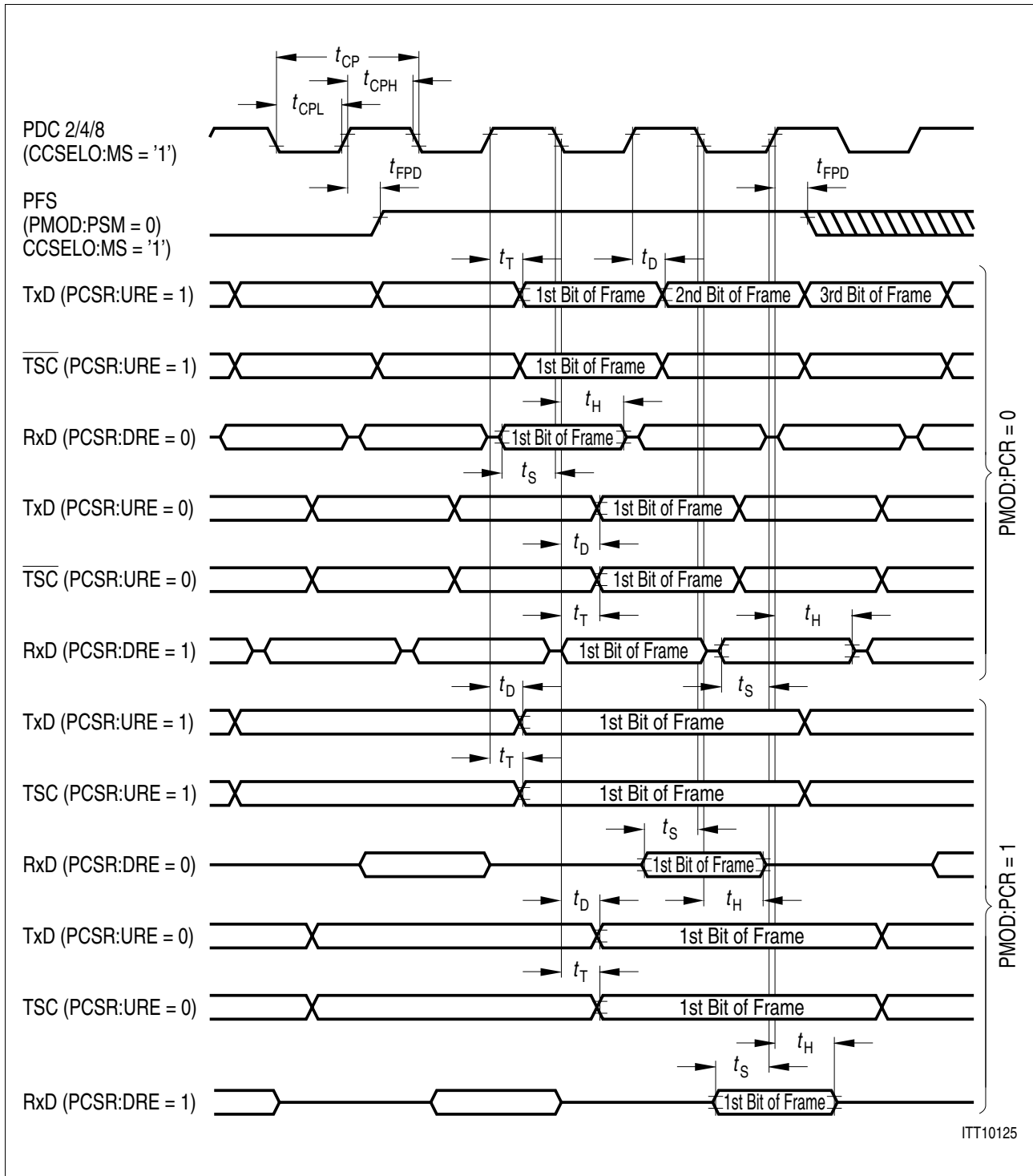


Figure 7-11 PCM-Interface Timing in Master Mode

**Table 7-12 PCM Interface Timing In Slave Mode**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock period	$t_{CP}$	240		ns	clock frequency $\leq 4096$ kHz
Clock period low	$t_{CPL}$	50		ns	
Clock period high	$t_{CPH}$	60		ns	
Clock period	$t_{CP}$	120		ns	clock frequency $> 4096$ kHz
Clock period low	$t_{CPL}$	30		ns	
Clock period high	$t_{CPH}$	30		ns	
Frame set-up time to clock	$t_{FS}$	36		ns	
Frame hold time from clock	$t_{FH}$	58		ns	
Serial data input set-up time	$t_S$	16		ns	PCM data frequency $> 4096$ kbit/s
Serial data hold time	$t_H$	44		ns	
Serial data input set-up time	$t_S$	29		ns	PCM data frequency $\leq 4096$ kbit/s
Serial data hold time	$t_H$	64		ns	
Tri-state control delay	$t_T$	15	60	ns	
PCM - serial data output delay	$t_D$	18	56	ns	PCM data frequency $> 4096$ kbit/s
PCM - serial data output delay	$t_D$	22	66	ns	PCM data frequency $\leq 4096$ kbit/s

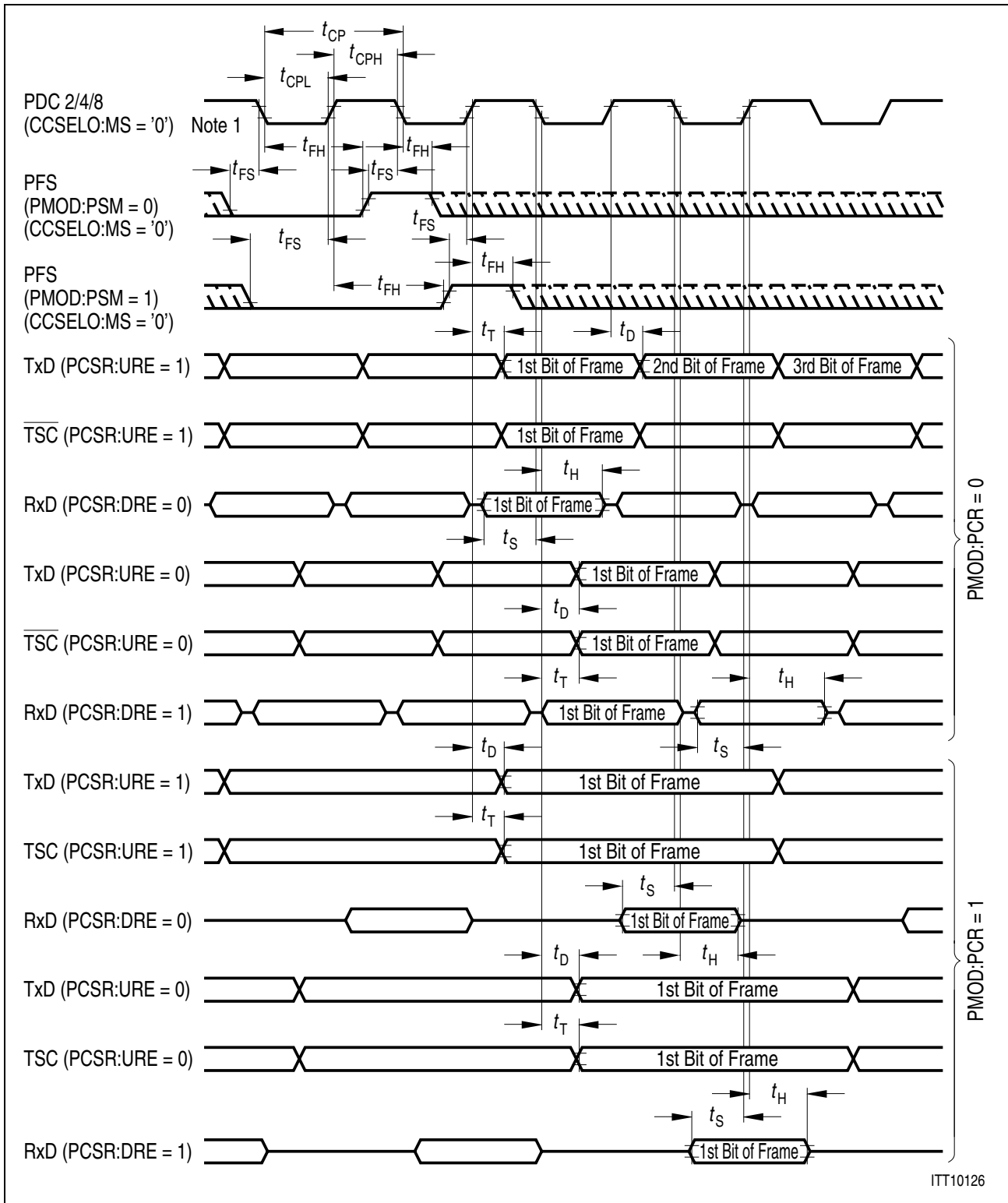
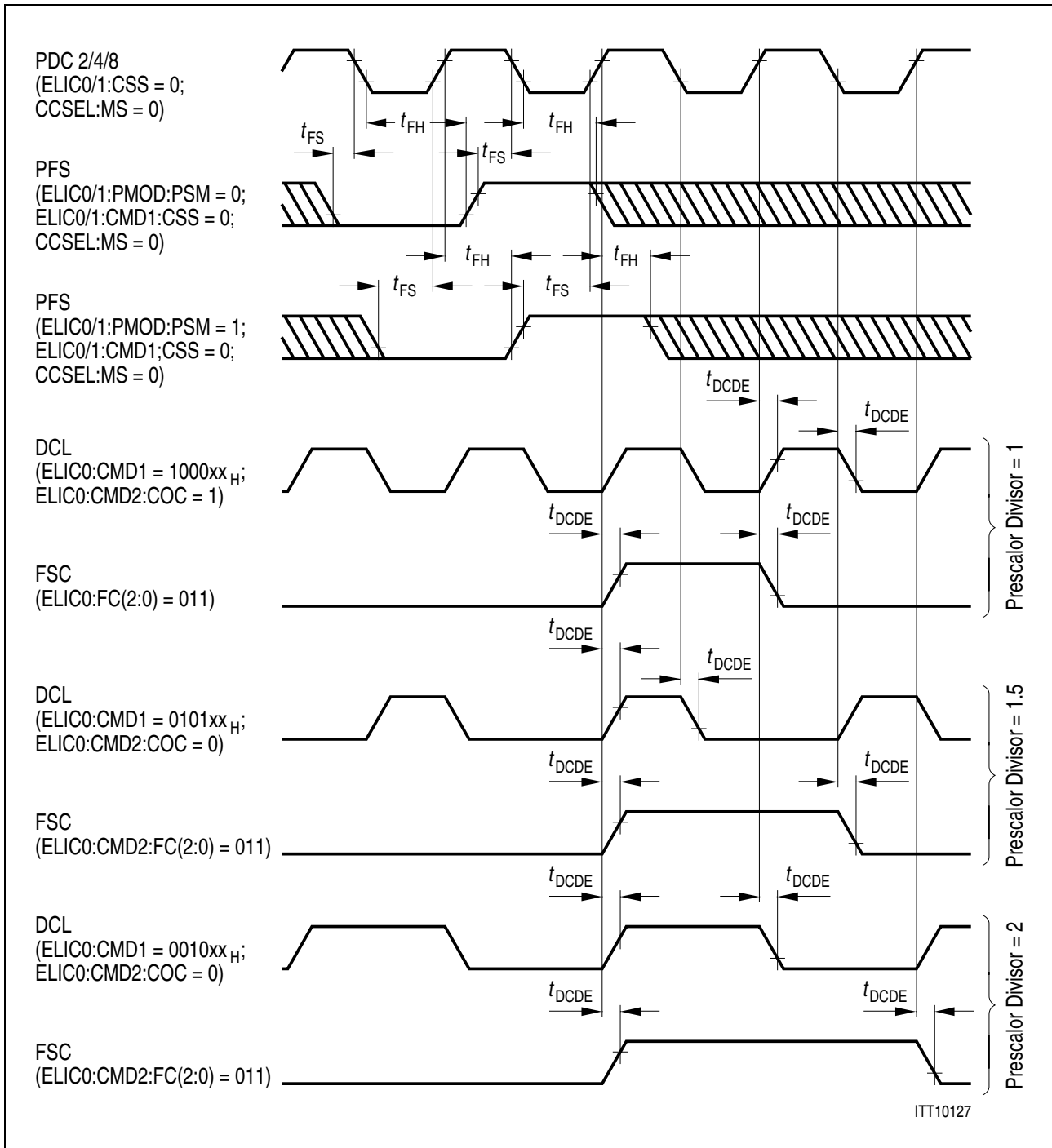


Figure 7-12 PCM-Interface Timing in Slave Mode

### 7.9.2 IOM<sup>®</sup>-2 Interface Timing

**Table 7-13 IOM<sup>®</sup>-2 Interface Clocks Timing when FSC and DCL are Driven by ELIC0 and the DOC is in Slave Mode**

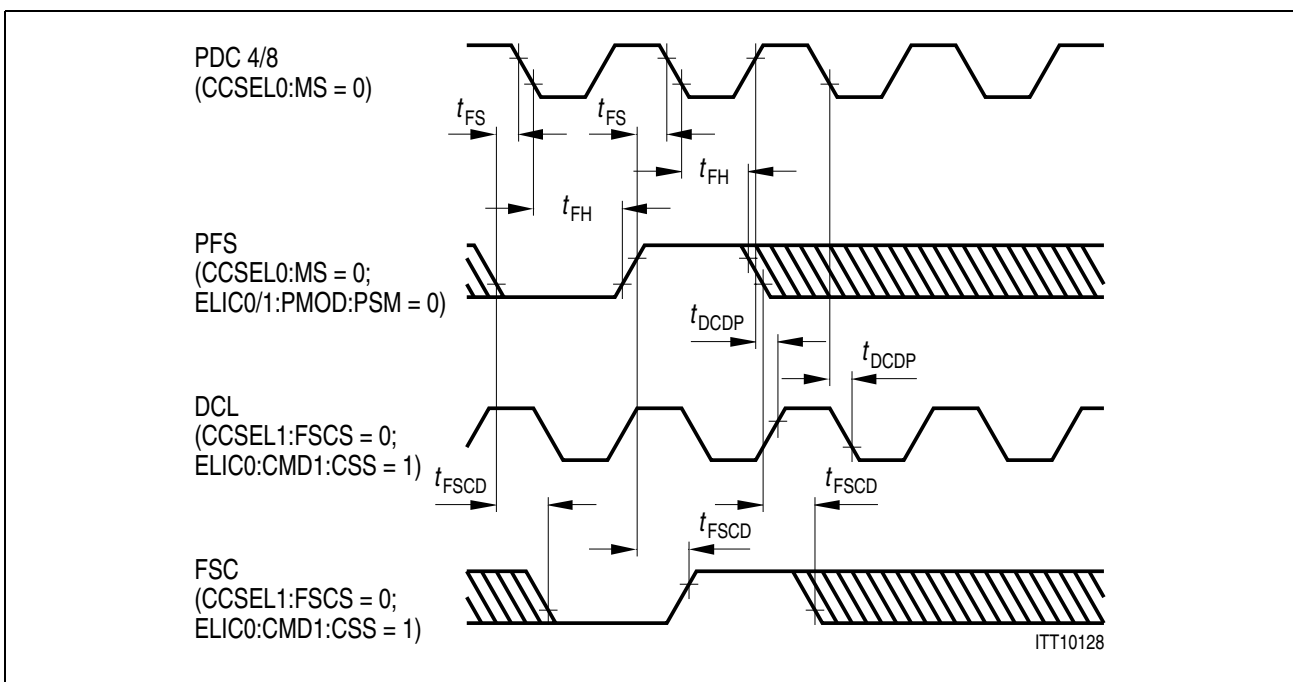
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Frame set-up time to clock	$t_{FS}$			ns	See <b>Table 7-12</b>
Frame hold time from clock	$t_{FH}$			ns	
Data clock delay when driven by ELIC0	$t_{DCDE}$			ns	



**Figure 7-13 IOM<sup>®</sup>-2 Interface Clocks Timing When FSC and DCL are Driven by ELIC0 and the DOC is in Slave Mode**

**Table 7-14 IOM<sup>®</sup>-2 Interface Clocks Timing When FSC and DCL are Driven directly by PDC4/8 and PFS, and the DOC is in Slave Mode**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Frame set-up time to clock	$t_{FS}$			ns	See <b>Table 7-12</b>
Frame hold time from clock	$t_{FH}$			ns	
Data clock delay when driven directly by PDC 4/8	$t_{DCDP}$		31	ns	
FSC Delay from PFS	$t_{FSCD}$		30	ns	

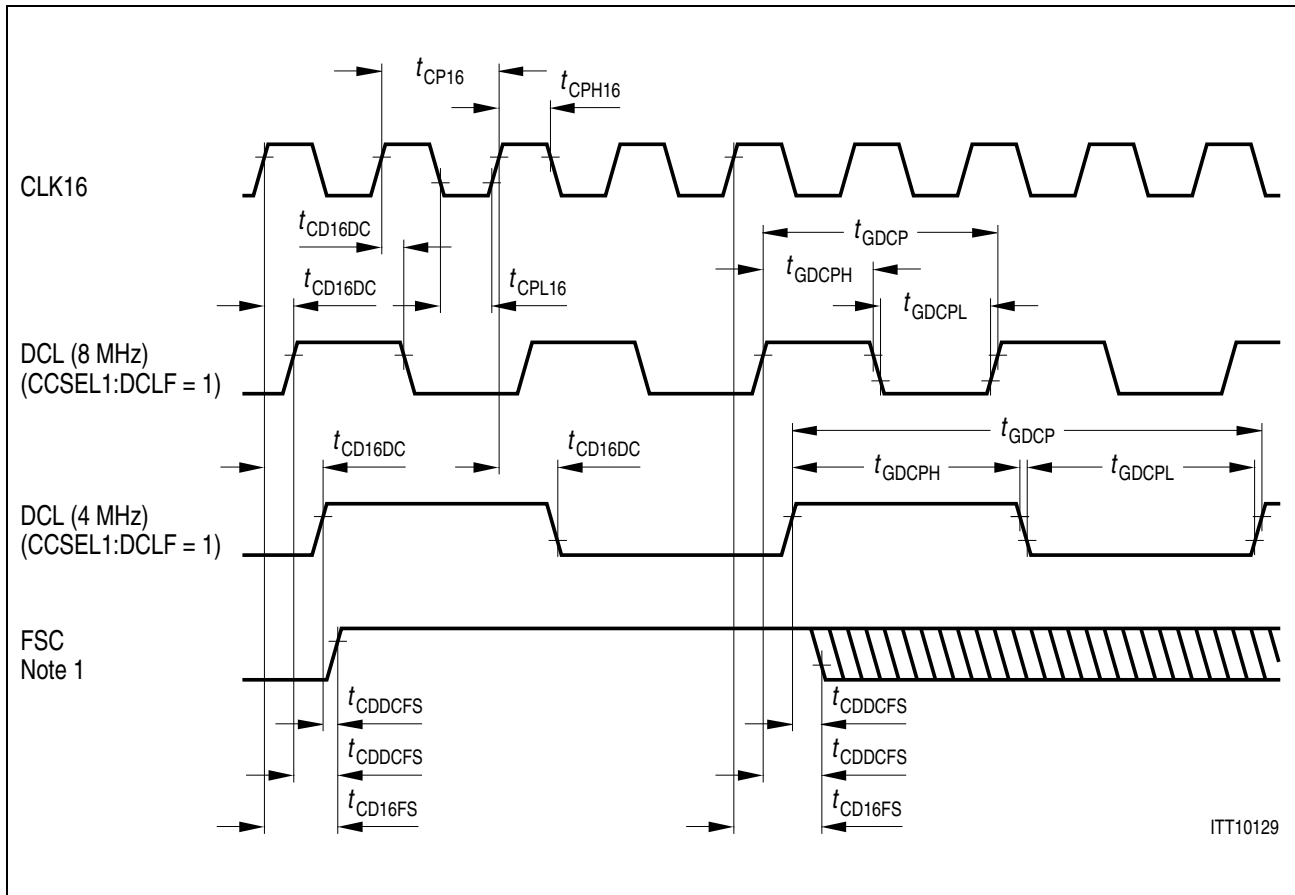


**Figure 7-14 IOM<sup>®</sup>-2 Interface Clocks Timing When FSC and DCL are Driven directly by PDC4/8 and PFS, and the DOC is in Slave Mode**

**Table 7-15 IOM<sup>®</sup>-2 Interface Clocks Timing when FSC and DCL are Driven Directly by PDC4/8 and PFS, and the DOC is in Master Mode (PDC and PFS are generated by internal clocks generator)**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK16 Clock period	$t_{CP}$	See Table 7-27			
CLK16 Clock period high	$t_{CPH16}$				
CLK16 Clock period low	$t_{CPL16}$				
Clock Delay CLK16 to FSC	$t_{CD16FS}$	13	46	ns	
Clock Delay CLK16 to DCL	$t_{CD16DC}$	8	34		
Clock Delay DCLto FSC	$t_{CDDCFs}$	7	33		
<b>When DCL (output) is driven internally by PDC8</b>					
Generated DCL Clock period	$t_{GDcP}$		122	ns	CCSEL1 = xxxxx000; CCSEL0: MS = 1; ELIC0:CMD1: CSS = 1
Generated Clock period high	$t_{GDcPH}$	50		ns	
Generated Clock period low	$t_{GDcPL}$	50		ns	
<b>When DCL (output) is driven internally by PDC4</b>					
Generated DCL Clock period	$t_{GDcP}$		244	ns	CCSEL1 = xxxxx010; CCSEL0: MS = 1; ELIC0:CMD1: CSS = 1
Generated Clock period high	$t_{GDcPH}$	105		ns	
Generated Clock period low	$t_{GDcPL}$	105		ns	





ITT10129

**Figure 7-15 IOM<sup>®</sup>-2 Interface Clocks Timing when FSC and DCL are Driven Directly by PDC4/8 and PFS, and the DOC is in Master Mode (PDC and PFS are generated by internal clocks generator)**

*Note: When FSC and DCL are driven by PFS and PDC4/8, which are generated by internal clocks generator, the pulse width of FSC (FSC period high) is 8 DCL cycles, when DCL is driven by PDC4, or 16 DCL cycles, when DCL is driven by PDC8.*

**Table 7-16 IOM<sup>®</sup>-2 Interface Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When FSC and DCL are inputs of the DOC (and of the ELIC) (ELIC0:CMD1:CSS = '1' AND CCSEL1[0,2] = "11")</b>					
Clock period	$t_{CP}$	240		ns	clock frequency $\leq 4096$ kHz <sup>1))</sup>
Clock period low	$t_{CPL}$	50		ns	
Clock period high	$t_{CPH}$	60		ns	

**Table 7-16 IOM<sup>®</sup>-2 Interface Timing (cont'd)**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock period	$t_{CP}$	120		ns	clock frequency > 4096 kHz <sup>1)</sup>
Clock period low	$t_{CPL}$	30		ns	
Clock period high	$t_{CPH}$	30		ns	
Frame set-up time to clock	$t_{FS}$	35		ns	1)
Frame hold time from clock	$t_{FH}$	54		ns	1)
Serial data input set-up time	$t_S$	29		ns	CFI data frequency > 4096 kbit/s
Serial data hold time	$t_H$	55		ns	
Serial data input set-up time	$t_S$	29		ns	CFI data frequency ≤ 4096 kbit/s
Serial data hold time	$t_H$	80		ns	
CFI-Serial output delay	$t_{CDR}$	19	70	ns	

**When FSC and DCL are outputs of the DOC but inputs of ELIC0 and ELIC1  
(ELIC0:CMD1:CSS = '1' AND CCSEL1[0,2] = "00")**

Serial data input set-up time	$t_S$	50		ns	CFI data frequency > 4096 kbit/s
Serial data hold time	$t_H$	25		ns	
Serial data input set-up time	$t_S$	50		ns	CFI data frequency ≤ 4096 kbit/s
Serial data hold time	$t_H$	45		ns	
CFI-Serial output delay	$t_{CDR}$	7	59	ns	

**When FSC and DCL are driven by ELIC0 (and outputs of the DOC)  
(ELIC0:CMD1:CSS = '0')**

Serial data input set-up time	$t_S$	TBD		ns	CFI data frequency > 4096 kbit/s
Serial data hold time	$t_H$	TBD		ns	
Serial data input set-up time	$t_S$	TBD		ns	CFI data frequency ≤ 4096 kbit/s
Serial data hold time	$t_H$	TBD		ns	
CFI-Serial output delay	$t_{CDR}$	TBD	TBD	ns	

<sup>1)</sup> These parameters are relevant only when FSC and DCL are inputs of the DOC.

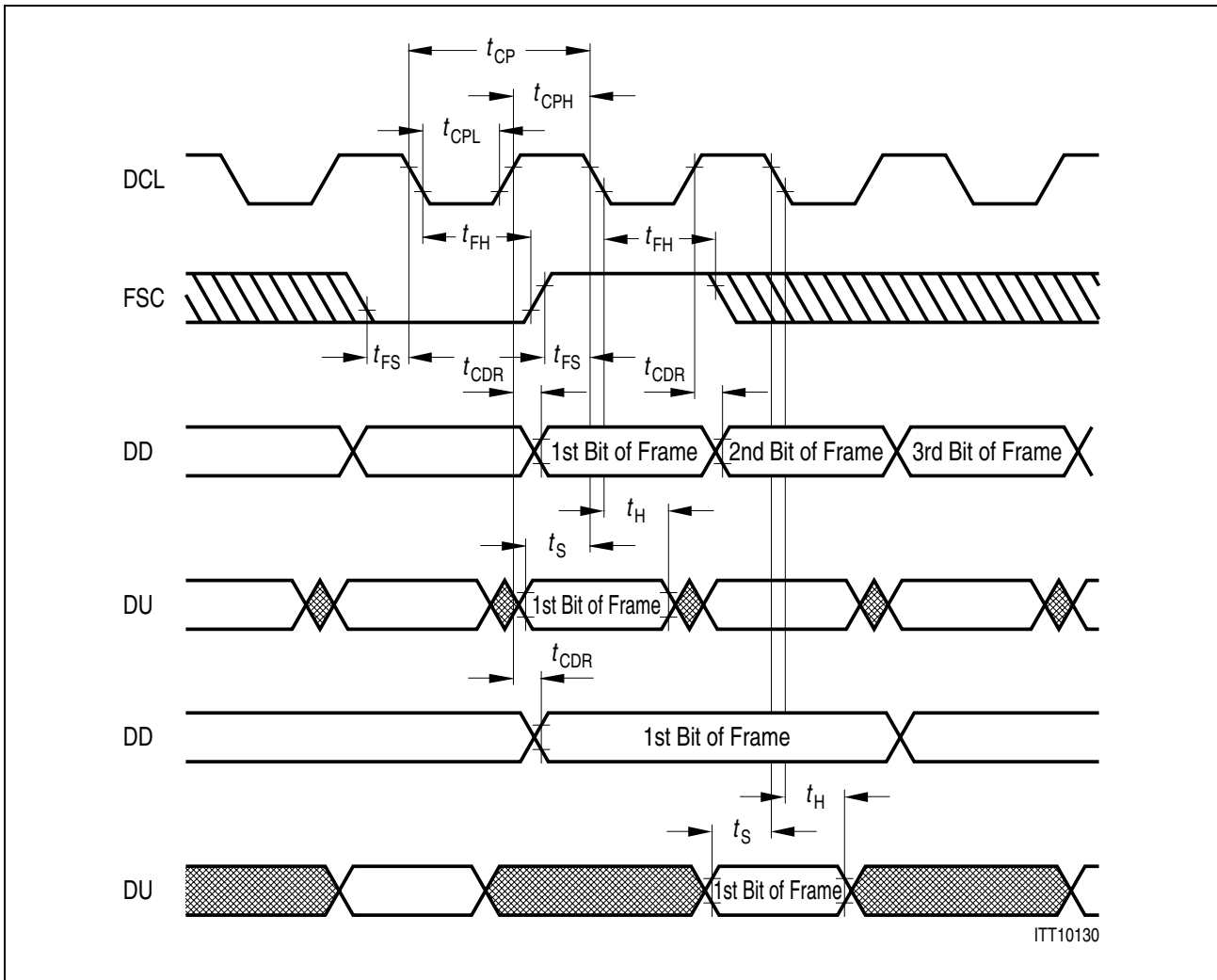
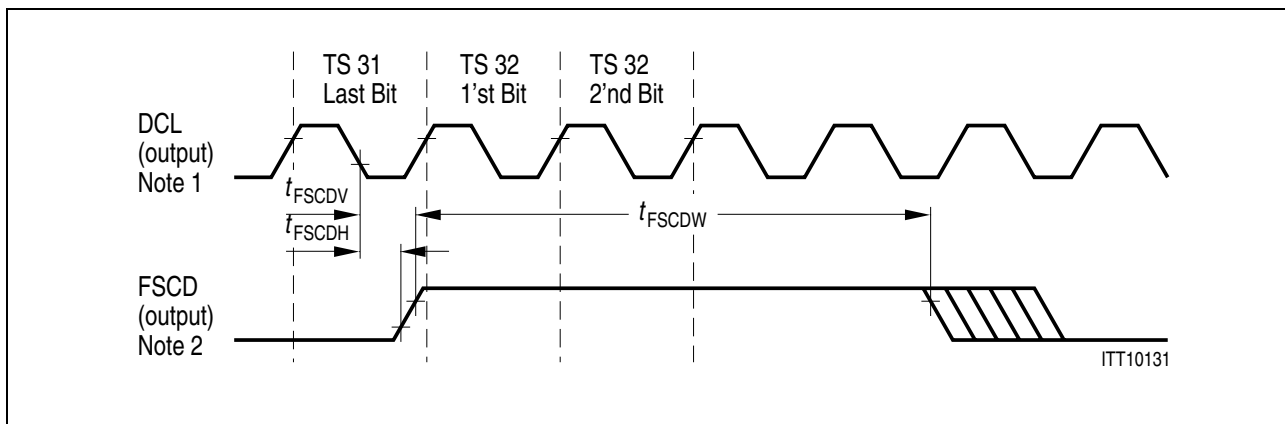


Figure 7-16 IOM<sup>®</sup>-2 Interface Timing

Table 7-17 FSCD (Delayed FSC) Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
FSCD hold time after output DCL falling edge	$t_{FSCDH}$	20		ns	
FSCD valid time after output DCL falling edge	$t_{FSCDV}$		80	ns	When DCL rate is 8 MHz and DSP clock rate is 40 MHz. <sup>1)</sup>
			150	ns	When DCL rate is 4 MHz
FSCD width	$t_{FSCDW}$	900		ns	

<sup>1)</sup> When 8 MHz DCL is selected, the DSP clock rate is restricted to 40 MHz, to ensure proper work of the PEDIU.

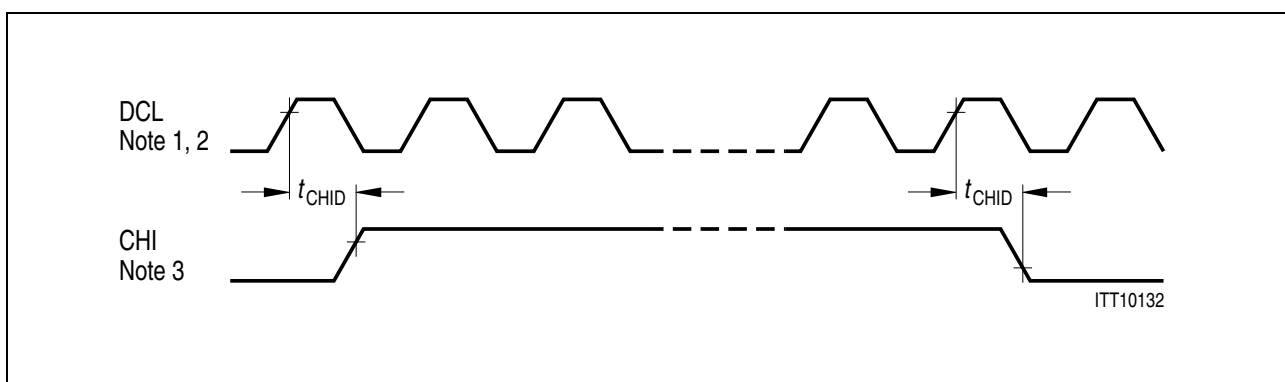


**Figure 7-17 FSCD Timing**

- Note: 1) The only way to use FSCD is when DCL and FSC are both outputs.  
 For DCL parameters when it is used as an output see tables: , and **Table 7-15**.
- 2) FSCD is activated only when FSC is an output (CCSEL1:FSCS = 0 or ELIC0:CMD1:CSS = 0), and when the PEDIU is in mode 2, 3 or 4 (4 Mbit/s or 8 Mbit/s) and in active mode (refer to **Table 7-13**, **Table 7-14**, **Table 7-15** and **Table 7-16**). When FSC is configured as input, FSCD is in tri-state. If FSC is configured as output and the PEDIU is in idle mode or is not in work mode 2, 3 or 4, FSCD will be driven by constant '0'

**Table 7-18 Channel Indication (CHI) Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CHI delay from DCL rising edge	$t_{CHID}$	-5	15	ns	When DCL is an output
		5	30	ns	When DCL is an input



**Figure 7-18 Channel Indication (CHI) Timing**

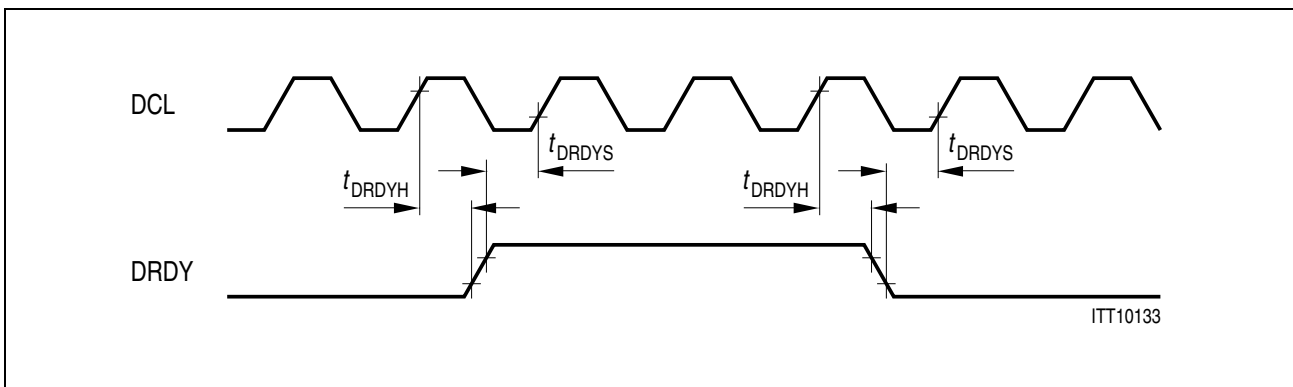
- Note: 1) For DCL parameters when it is used as an output see **Table 7-15**.
- 2) For DCL parameters when it is used as an input see **Table 7-16**.

**Electrical Characteristics**

3) CHI signal width is 8 DCL cycles when it is operated in a single clock mode (VMODR:MOD[1:0] = 00, 01, 10), and 16 DCL cycles when it is operated in a double clock mode (VMODR:MOD[1:0] = 11). For more details see **Table 7-13, Table 7-14, Table 7-15 and Table 7-16.**

**Table 7-19 DRDY Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
DRDY setup prior to DCL rising edge	$t_{DRDYS}$	TBD		ns	When DCL is an output
DRDY hold time after DCL rising edge	$t_{DRDYH}$	TBD		ns	
DRDY setup prior to DCL rising edge	$t_{DRDYS}$	TBD		ns	When DCL is an input
DRDY hold time after DCL rising edge	$t_{DRDYH}$	TBD		ns	



**Figure 7-19 DRDY Timing**

**7.9.3 Serial Interface Timing**

**Table 7-20 Serial Interface Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When SACCO-B0/1 HDC is Driven by an Externally Generated Clock <sup>1)</sup></b>					
Receive data set-up	$t_{RDS}$	24		ns	
Receive data hold	$t_{RDH}$	22		ns	
Collision data set-up	$t_{CDS}$	9		ns	

**Table 7-20 Serial Interface Timing (cont'd)**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Collision data hold	$t_{CDH}$	37		ns	
Transmit data delay	$t_{XDD}$	19	60	ns	
Tri-state control delay	$t_{RTD}$	19	70	ns	
Clock period	$t_{CP}$	See <b>chapter 7.9.1</b> for the timing of inputs PDC2/4/8 and <b>Table 7-16</b> for the timing of input DCL.			When the source of SACCO-B0/1 is one of the next inputs: PDC8,PDC4,PDC2 or DCL.
Clock period low	$t_{CPL}$				
Clock period high	$t_{CPH}$				
Strobe set-up time to clock	$t_{XSS}$	90	$t_{CP-41}$	ns	
Strobe set-up time to clock (extended transparent mode)	$t_{XSX}$	40	$t_{CP-41}$	ns	
Strobe hold time from clock	$t_{XSH}$	41		ns	
Transmit data delay from strobe	$t_{SDD}$		66	ns	
Transmit data high impedance from clock	$t_{XCZ}$		65	ns	
Transmit data high impedance from strobe	$t_{XSZ}$		55	ns	
Tri-stste control delay from strobe	$t_{STD}$		66	ns	
Sync pulse set-up time to clock	$t_{SS}$	40	$t_{CP-41}$	ns	
Sync pulse width	$t_{SW}$	40		ns	

**When SACCO-B0/1 HDC is Driven by an Internally Generated Clock <sup>2)</sup>**

Receive data set-up	$t_{RDS}$	49		ns	
Receive data hold	$t_{RDH}$	13		ns	
Collision data set-up	$t_{CDS}$	35		ns	
Collision data hold	$t_{CDH}$	33		ns	
Transmit data delay	$t_{XDD}$	0	42	ns	
Tri-state control delay	$t_{RTD}$	0	48	ns	

**Table 7-20 Serial Interface Timing (cont'd)**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Clock period	$t_{CP}$	See <b>Table 7-13</b> , <b>Table 7-14</b> and <b>Table 7-15</b> for the timing of outputs PDC2/4/8 and DCL.			When the source of SACCO-B0/1 is one of the next internally generated clocks: PDC8,PDC4,PDC2 or DCL.
Clock period low	$t_{CPL}$				
Clock period high	$t_{CPH}$				
Strobe set-up time to clock	$t_{XSS}$	112	$t_{CP-25}$	ns	
Strobe set-up time to clock (extended transparent mode)	$t_{XSX}$	62	$t_{CP-25}$	ns	
Strobe hold time from clock	$t_{XSH}$	25		ns	
Transmit data delay from strobe	$t_{SDD}$		46	ns	
Transmit data high impedance from clock	$t_{XCZ}$		44	ns	
Transmit data high impedance from strobe	$t_{XSZ}$		51	ns	
Tri-stste control delay from strobe	$t_{STD}$		46	ns	
Sync pulse set-up time to clock	$t_{SS}$	61	$t_{CP-30}$	ns	
Sync pulse width	$t_{SW}$	70		ns	

- 1) The source which drives SACCO-B0/1 HDC can be PDC8, PDC4, PDC2 OR DCL. Anyone of these optional sources may be generated internally or may be input to the DOC. The first part of the above table, is valid during the latter case.
- 2) The source which drives SACCO-B0/1 HDC can be PDC8, PDC4, PDC2 OR DCL. Anyone of these optional sources may be generated internally or may be input to the DOC. The second part of the above table, is valid during the former case.

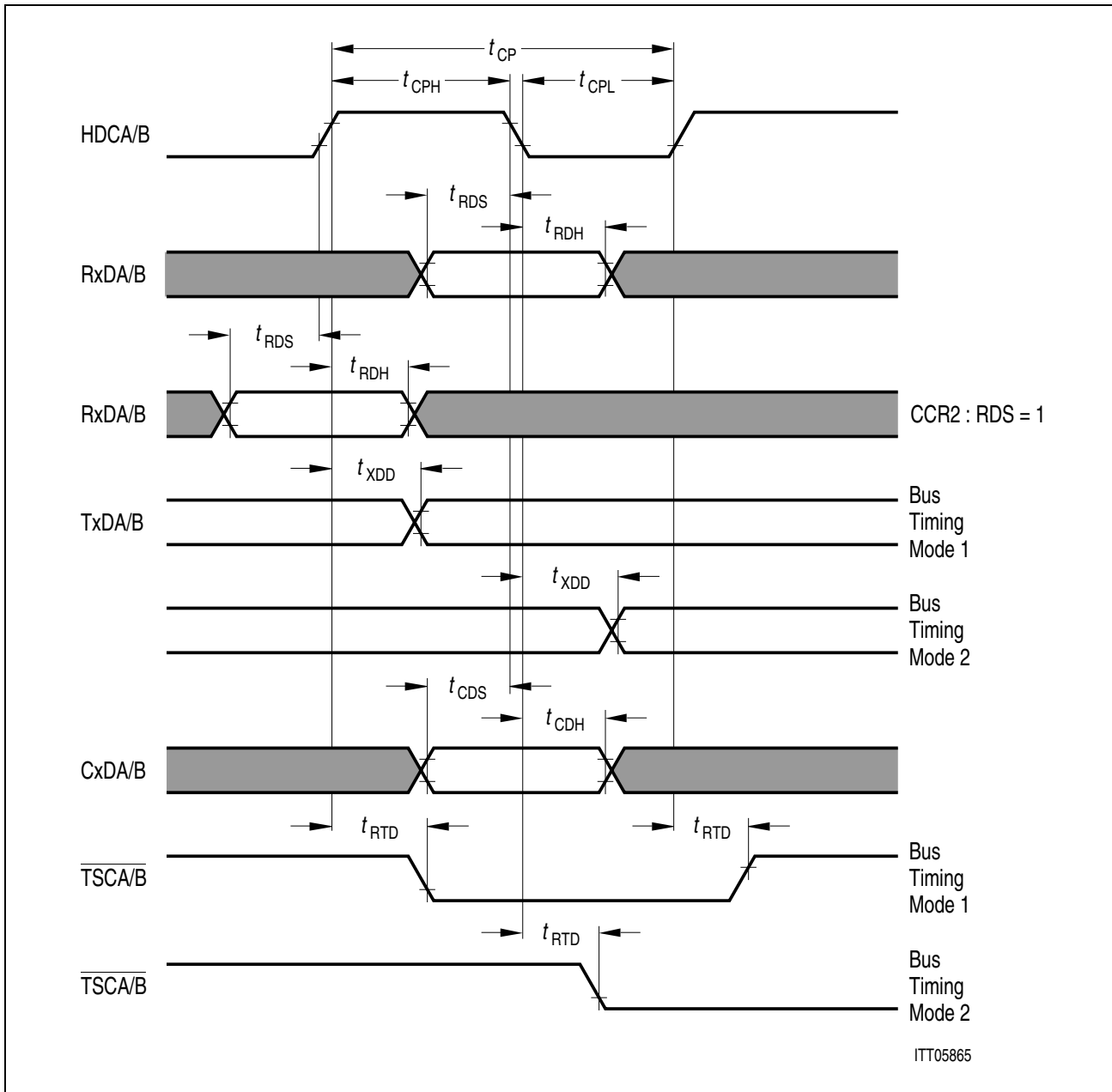
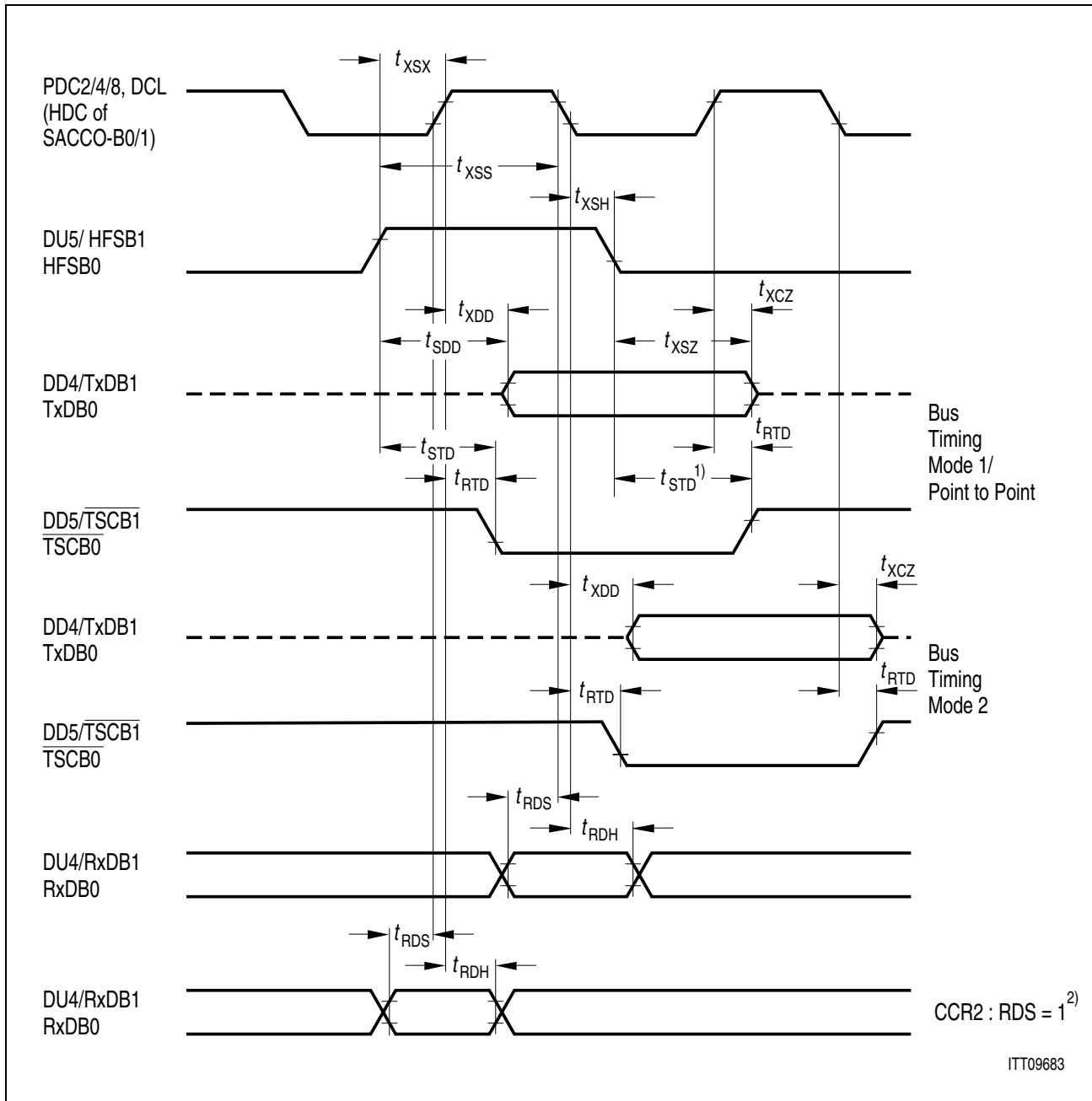


Figure 7-20 Serial Interface Timing





**Figure 7-21 Serial Interface Strobe Timing (clock mode 1)**

Note: 1) Only applicable in Point-to-Point configuration if  $t_{XSH} < 0$ . (HFS is turned off before HDC falling edge). In this case TxDB becomes invalid again,  $\overline{TSCB}$  is set to inactive '1', and the internal counters are not incremented, so the same valid data will be shifted out again with the next HDC/HFS rising edge.

2) With  $RDS = 1$  the sampling edge is shifted 1/2 clock phase forward. The data is internally still processed with the falling edge. Therefore the strobe timing is still related to the next falling edge in that case.

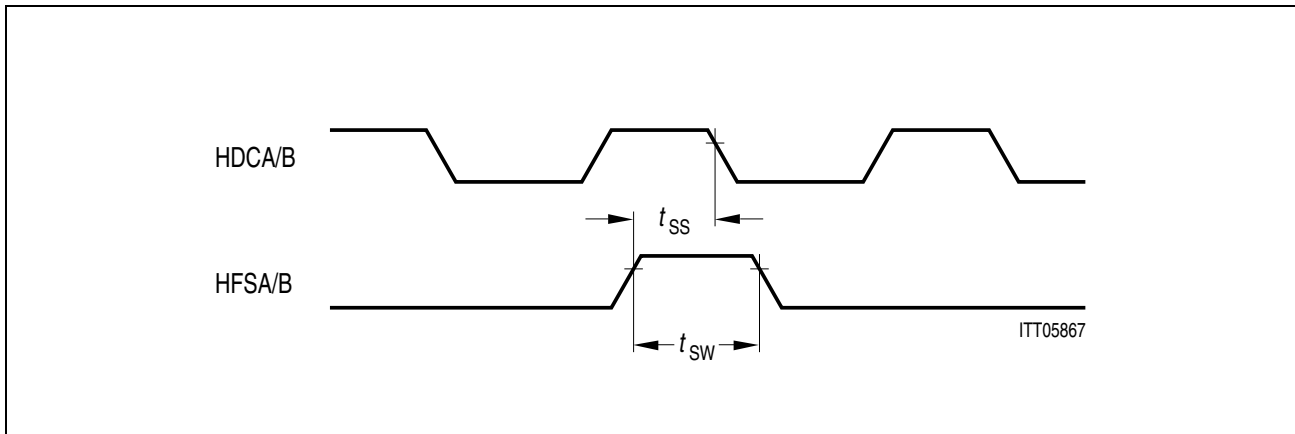


Figure 7-22 Serial Interface Synchronization Timing (clock mode 2)

### 7.9.4 Reset Timing

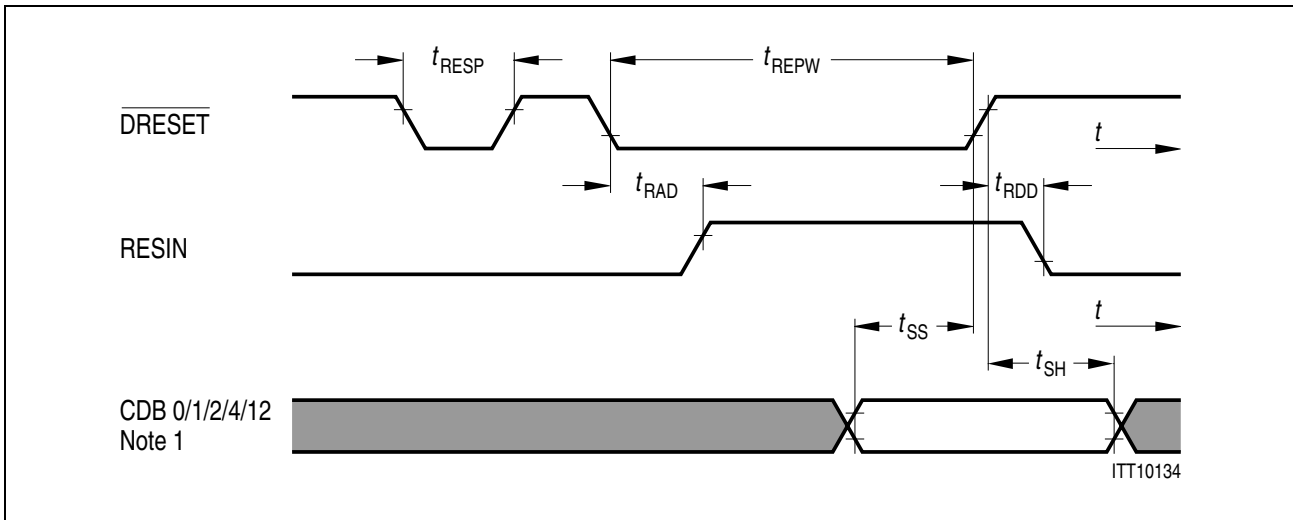
Table 7-21 Reset Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
DRESET $\bar$ -spike pulse width	$t_{RESP}$		5	ns	
DRESET-pulse width	$t_{REPW}$	1250		ns	Warm reset <sup>1)</sup>
		10		ms	Cold reset <sup>2)</sup>
RESIN-activation delay	$t_{RAD}$		30	ns	
RESIN-deactivation delay	$t_{RDD}$		300	ns	
Strap set-up time	$t_{SS}$	10		ns	<sup>3)</sup>
Strap hold time	$t_{SH}$	10		ns	

<sup>1)</sup> Warm reset - when the 40 MHz external crystal is already in steady state.

<sup>2)</sup> Cold reset - when the 40 MHz external crystal is not yet in steady state.

<sup>3)</sup> The strap inputs are: CDB0/BOOT, CDB1/DBG, CDB2/ROM, CDB4/URST and CDB12/SEIBDIS. When a strap is not driven externally, it is driven internally by an internal pull-down. If a fixed external pull-up is applied on a strap, a pull-up resistor of 5 k $\Omega$  is required.



**Figure 7-23**  $\overline{\text{DRESET}}$  and RESIN timing

Note: CDB0/BOOT, CDB1/DBG, CDB2/ROM, CDB4/URST and CDB12/SEIBDIS are used as straps during reset.

### 7.9.5 Boundary Scan Timing

**Table 7-22** Boundary Scan Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Test Clock Period	$t_{TCP}$	160		ns	
Test Clock Period Low	$t_{TCPL}$	80		ns	
Test Clock Period High	$t_{TCPH}$	80		ns	
TMS-set-up time to TCK	$t_{MSS}$	30		ns	
TMS-hold time from TCK	$t_{MSH}$	30		ns	
TDI-set-up time to TCK	$t_{DIS}$	30		ns	
TDI-hold time to TCK	$t_{DIH}$	30		ns	
TDO-valid delay from TCK	$t_{DOD}$		60	ns	

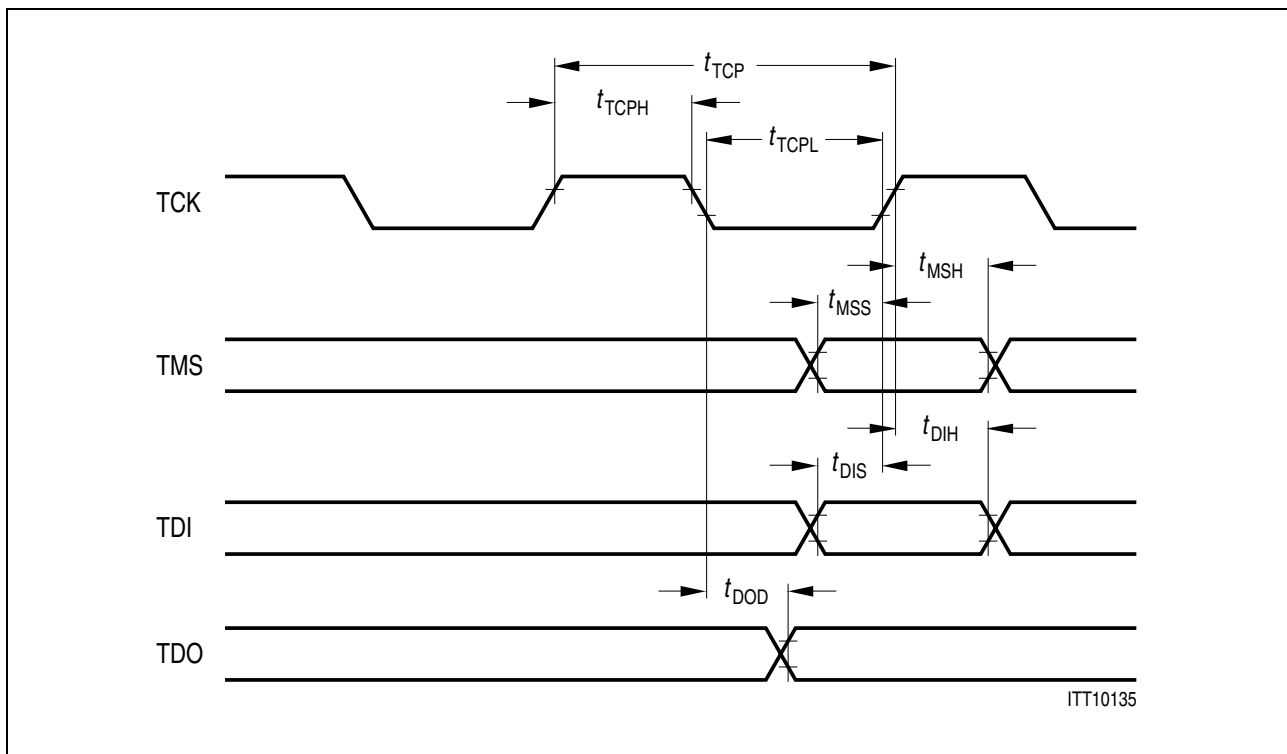


Figure 7-24 Boundary Scan Timing

### 7.9.6 DSP External Memory Interface Timing

Table 7-23 Program Read Access Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When <math>C_{LOAD} = 50 \text{ pF}</math></b>					
Program access time from $\overline{CDPR}$ and CAB	$t_{PACC}$		32.5	ns	Max. DSP Frequency = 20 MHz
			21	ns	Max. DSP Frequency = 2 MHz
			15.8	ns	Max. DSP Frequency = 30 MHz
			12.8	ns	Max. DSP Frequency = 33 MHz
			9.5	ns	Max. DSP Frequency = 37 MHz
			7.5	ns	Max. DSP Frequency = 40 MHz

Electrical Characteristics

Table 7-23 Program Read Access Timing (cont'd)

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		

When  $C_{LOAD} = 30 \text{ pF}$

Program access time from $\overline{\text{CDPR}}$ and CAB	$t_{PACC}$		33.5	ns	Max. DSP Frequency = 20 MHz
			22	ns	Max. DSP Frequency = 26 MHz
			16.8	ns	Max. DSP Frequency = 30 MHz
			13.8	ns	Max. DSP Frequency = 33 MHz
			10.5	ns	Max. DSP Frequency = 37 MHz
			8.5	ns	Max. DSP Frequency = 40 MHz

Parameters for production test, only

$\overline{\text{CDPR}}$ delay from DTCLK	$t_{CDPRD}$		6	ns	
CAB delay from DTCLK	$t_{CABD}$		6	ns	
CDB set-up time prior to DTCLK	$t_{CDBS}$	11.5		ns	

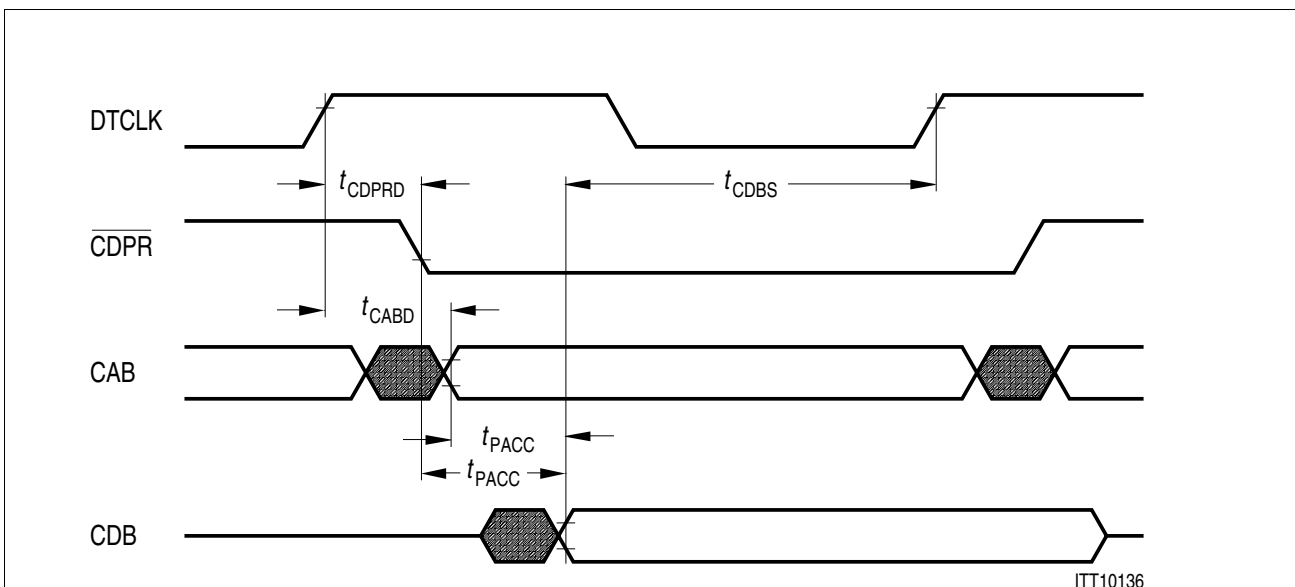


Figure 7-25 Program Read Access Timing

Table 7-24 External Data Read Access Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When <math>C_{LOAD} = 50</math> pF</b>					
Data access time from $\overline{CDPR}$ , $\overline{CMBR}$ or $\overline{CBR1}$ )	$t_{DACCR}$ 1)		32.5	ns	Max. DSP Frequency = 20 MHz
			21	ns	Max. DSP Frequency = 26 MHz
			15.8	ns	Max. DSP Frequency = 30 MHz
			12.8	ns	Max. DSP Frequency = 33 MHz
			9.5	ns	Max. DSP Frequency = 37 MHz
			7.5	ns	Max. DSP Frequency = 40 MHz
Data access time from $\overline{CAB}$ )	$t_{DACCA}$ 1)		82.5	ns	Max. DSP Frequency = 20 MHz
			59	ns	Max. DSP Frequency = 26 MHz
			48.8	ns	Max. DSP Frequency = 30 MHz
			36.8	ns	Max. DSP Frequency = 33 MHz
			31.5	ns	Max. DSP Frequency = 37 MHz
			27.5	ns	Max. DSP Frequency = 40 MHz

Table 7-24 External Data Read Access Timing (cont'd)

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When <math>C_{LOAD} = 30</math> pF</b>					
Data access time from $\overline{CDPR}$ , $\overline{CMBR}$ or $\overline{CBR1}$ <sup>1)</sup>	$t_{DACCR}$ 1)		33.5	ns	Max. DSP Frequency = 20 MHz
			22	ns	Max. DSP Frequency = 26 MHz
			16.8	ns	Max. DSP Frequency = 30 MHz
			13.8	ns	Max. DSP Frequency = 33 MHz
			10.5	ns	Max. DSP Frequency = 37 MHz
			8.5	ns	Max. DSP Frequency = 40 MHz
Data access time from $\overline{CAB}$ <sup>1)</sup>	$t_{DACCA}$ 1)		83.5	ns	Max. DSP Frequency = 20 MHz
			60	ns	Max. DSP Frequency = 26 MHz
			49.8	ns	Max. DSP Frequency = 30 MHz
			37.8	ns	Max. DSP Frequency = 33 MHz
			32.5	ns	Max. DSP Frequency = 37 MHz
			28.5	ns	Max. DSP Frequency = 40 MHz

<sup>1)</sup> The values of  $t_{DACCR}$  and  $t_{DACCA}$ , which are presented in the table, above, are valid when the number of wait-states on the external data space is 0 (MEMCONFR:DWS = 0, see **section 2.7.3.1**). When DWS > 0 (DWS = number of data wait-states), the time which is taken by the wait-states, should be added to the appropriate TV (TV = Table Value = the appropriate value from the table, above).

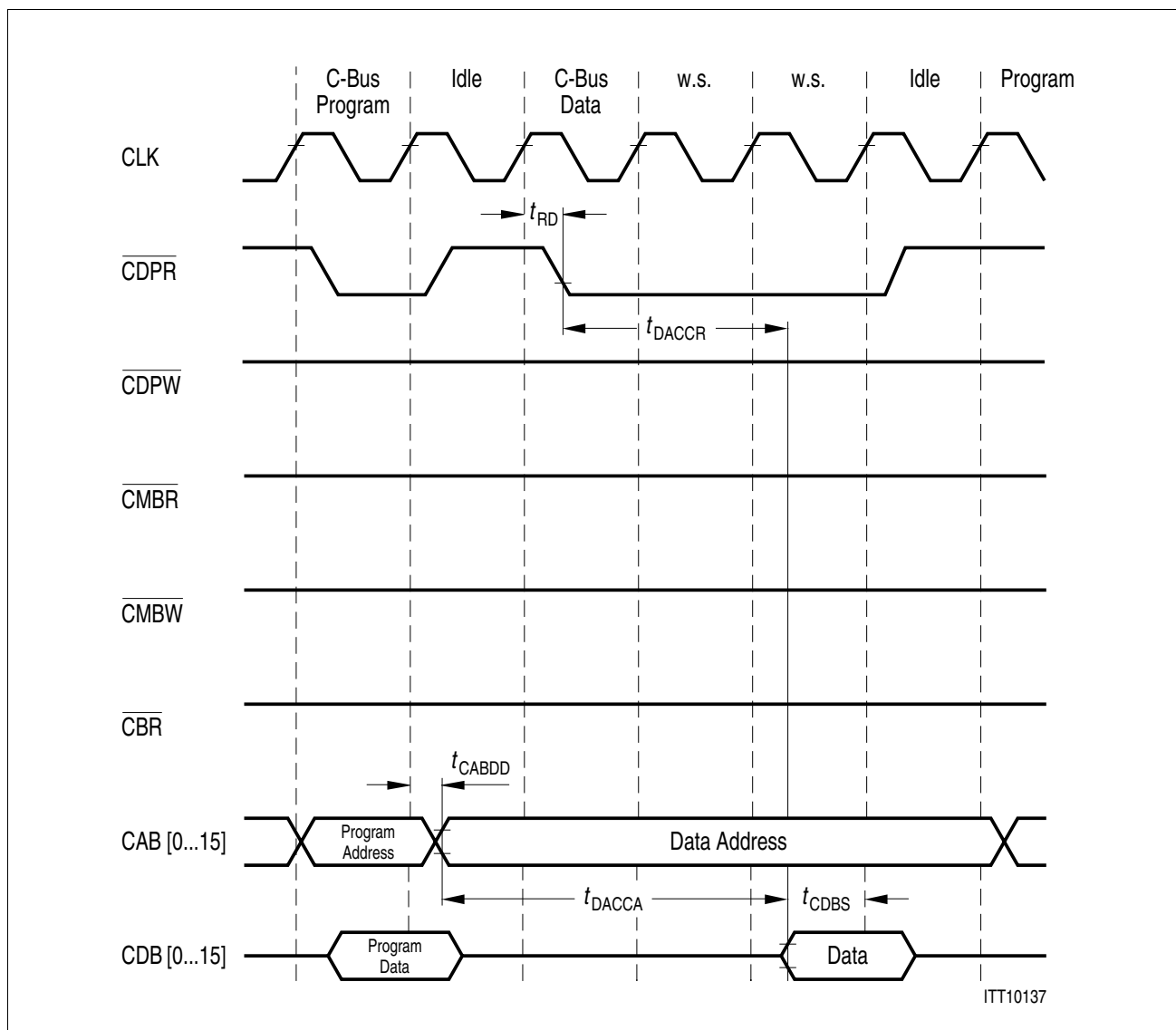
For example, when  $C_{LOAD} = 50$  pF, Max. DSP Frequency = 40 MHz and DWS = 3:

$$t_{DACCR} = TV + DWS \times \text{clock-period} = 7.5 + 3 \times 25 = 82.5 \text{ ns}$$

$$t_{DACCA} = TV + DWS \times \text{clock-period} = 27.5 + 3 \times 25 = 102.5 \text{ ns}$$

**Table 7-24 External Data Read Access Timing (cont'd)**

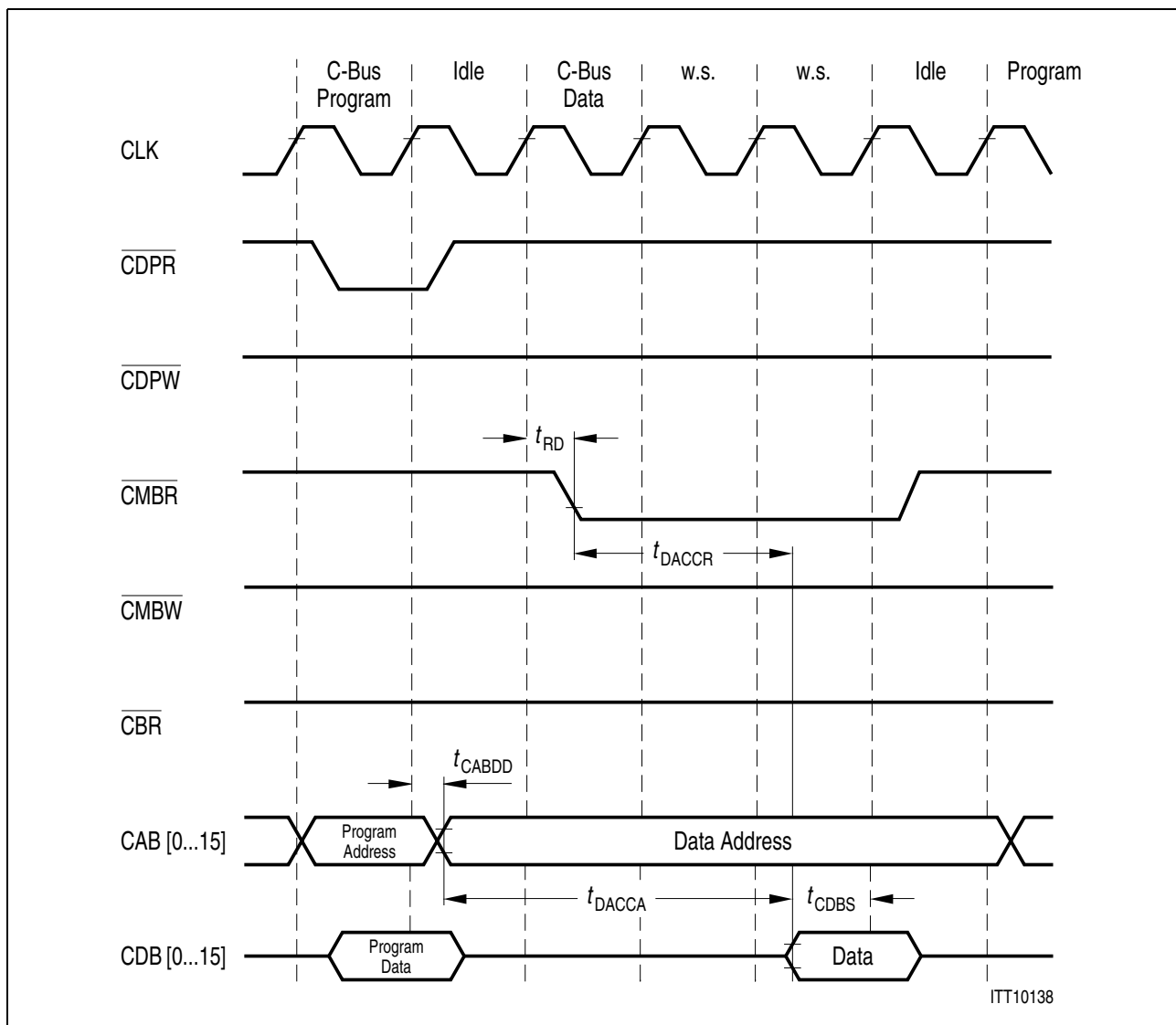
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>Parameters for production test, only</b>					
$\overline{\text{CDPR}}$ , $\overline{\text{CMBR}}$ or $\overline{\text{CBR}}$ delay from DTCLK	$t_{\text{RD}}$		3.5	ns	
CAB delay from DTCLK	$t_{\text{CABD}}$		4.5	ns	
CDB set-up time prior to DTCLK	$t_{\text{CDBS}}$	11.5		ns	



**Figure 7-26 External RAM Data Read Access Timing Diagram**

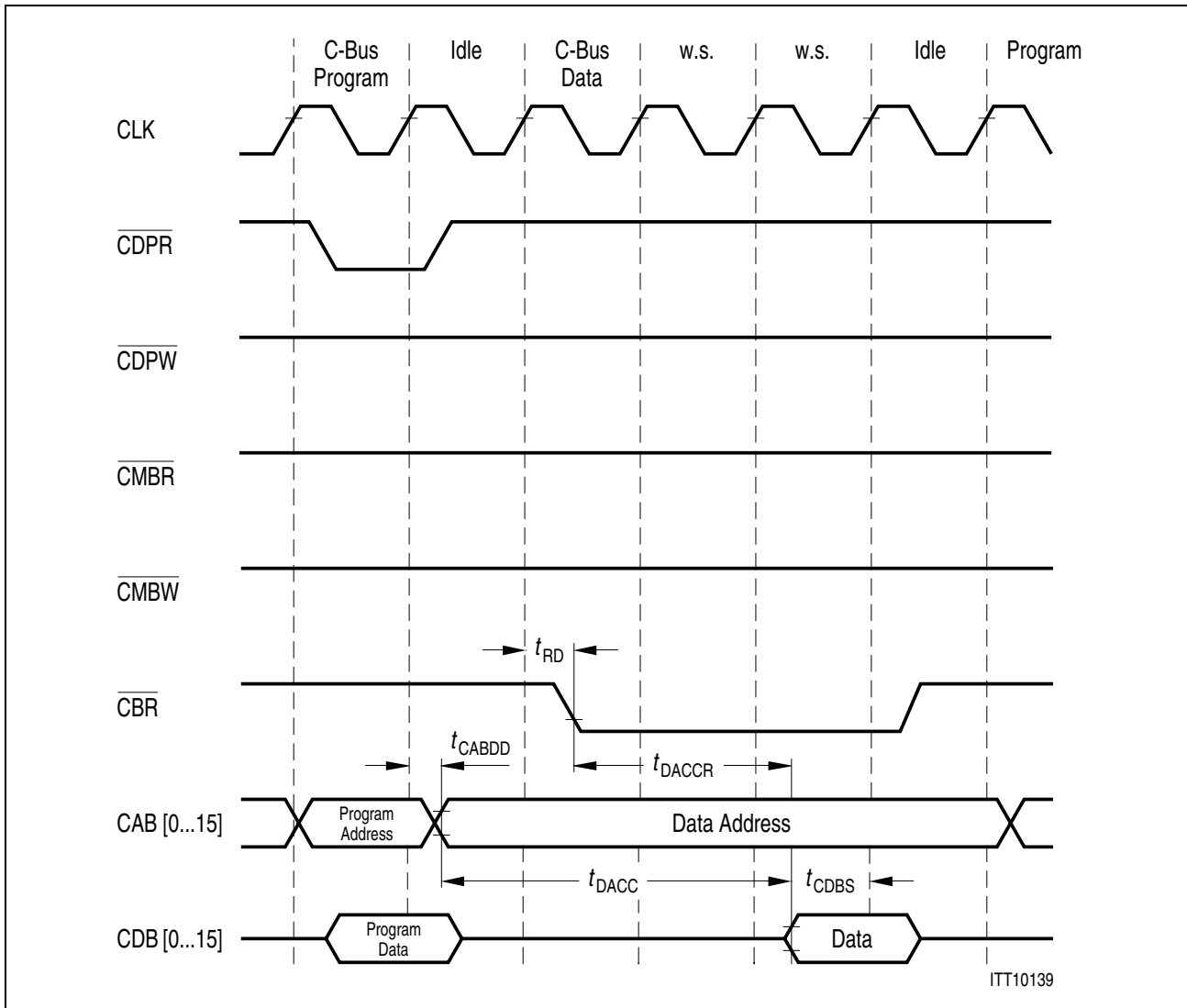


Note: The figure, above, describes a situation in which the number of wait states is 2 (MEMCONF:DWS = 2, see **section 2.7.3.1**). DWS can be programmed to any value from 0 to 7, and the timing will be changed accordingly.



**Figure 7-27 Emulation Mail-Box Read Access Timing Diagram**

Note: The figure, above, describes a situation in which the number of wait states is 2 (MEMCONF:DWS = 2, see **section 2.7.3.1**). DWS can be programmed to any value from 0 to 7, and the timing will be changed accordingly.



**Figure 7-28 Boot ROM Read Access Timing Diagram**

Note: The figure, above, describes a situation in which the number of wait states is 2 (MEMCONF:DWS = 2, see **section 2.7.3.1**). DWS can be programmed to any value from 0 to 7, and the timing will be changed accordingly.

**Table 7-25 External Program/Data Write Access**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CDPW or CMBW width <sup>1)</sup>	$t_{DWW}^{1)}$	8		ns	
CAB set-up time prior to CDPW or CMBW <sup>1)</sup>	$t_{DAS}^{1)}$	30		ns	
CDB set-up time prior to CDPW or CMBW <sup>1)</sup>	$t_{DDS}^{1)}$	20		ns	

**Table 7-25 External Program/Data Write Access**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CAB hold time from $\overline{\text{CDPW}}$ or $\overline{\text{CMBW}}$	$t_{\text{DAH}}$	5		ns	
CDB hold time from $\overline{\text{CDPW}}$ or $\overline{\text{CMBW}}$	$t_{\text{DDH}}$	4		ns	
Parameters for production test, only					
CDB delay from DTCLK	$t_{\text{DDD}}$		20	ns	
CAB delay from DTCLK	$t_{\text{DAD}}$		10	ns	
$\overline{\text{CDPW}}$ or $\overline{\text{CMBW}}$ delay from DTCLK	$t_{\text{DWD}}$		6	ns	
CAB hold time from DTCLK	$t_{\text{DAHC}}$	0		ns	
CDB hold time from DTCLK	$t_{\text{DDHC}}$	10		ns	

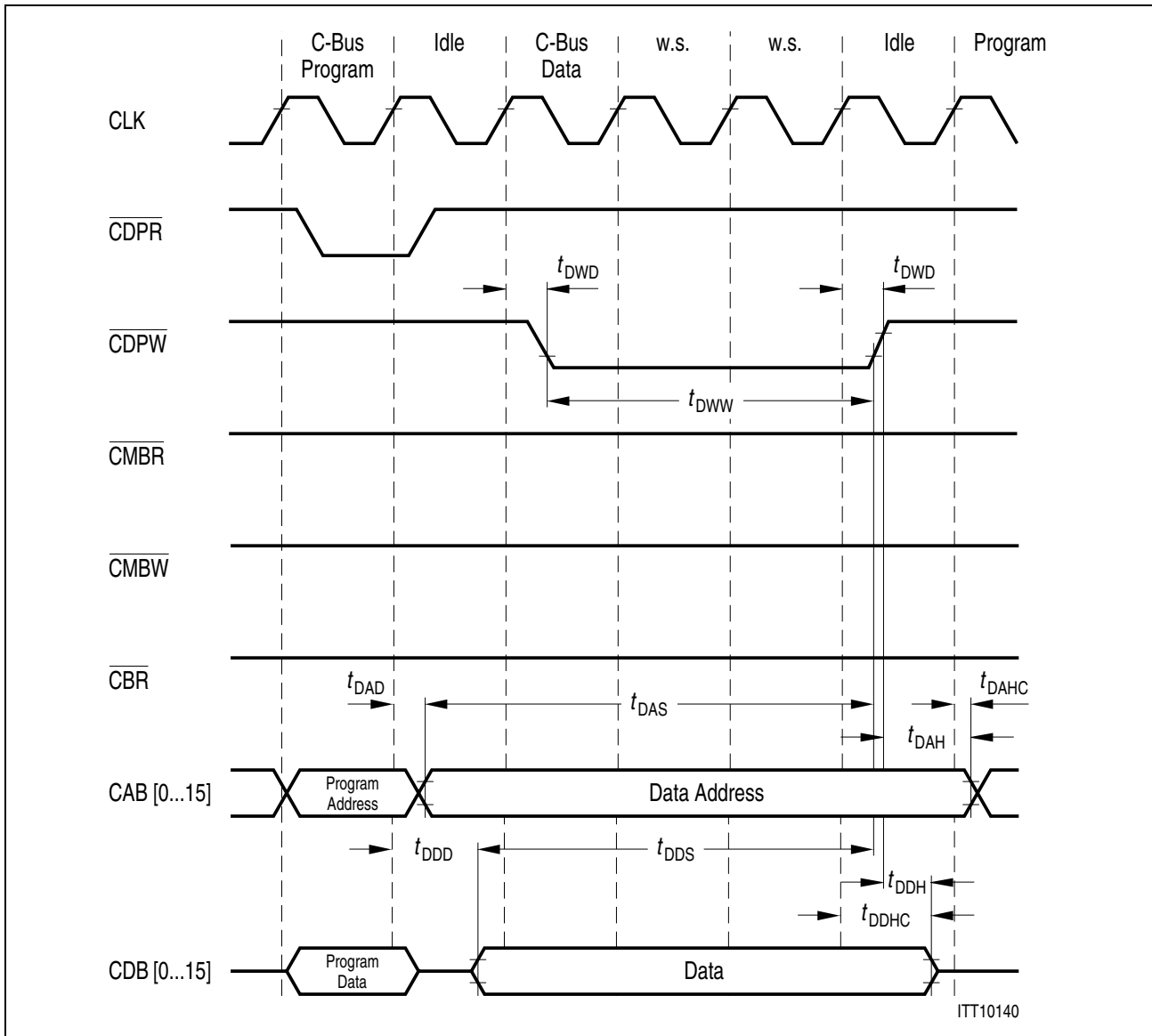
1) The values of  $t_{\text{DWW}}$ ,  $t_{\text{DAS}}$  and  $t_{\text{DDS}}$  which are presented in the table, above, are valid only during Program Write Access, or during Data Write Access, when the number of wait-states on the external data space is 0 (MEMCONFR:DWS = 0, see **section 2.7.3.1**). For Data Write Access Timing, When DWS > 0 (DWS = number of data wait-states), the time which is taken by the wait-states, should be added to the appropriate TV (TV = Table Value = the appropriate value from the table, above).

For example, when DWS = 3:

$$t_{\text{DWW}} = \text{TV} + \text{DWS} \times \text{clock-period} = 8 + 3 \times 25 = 83 \text{ ns}$$

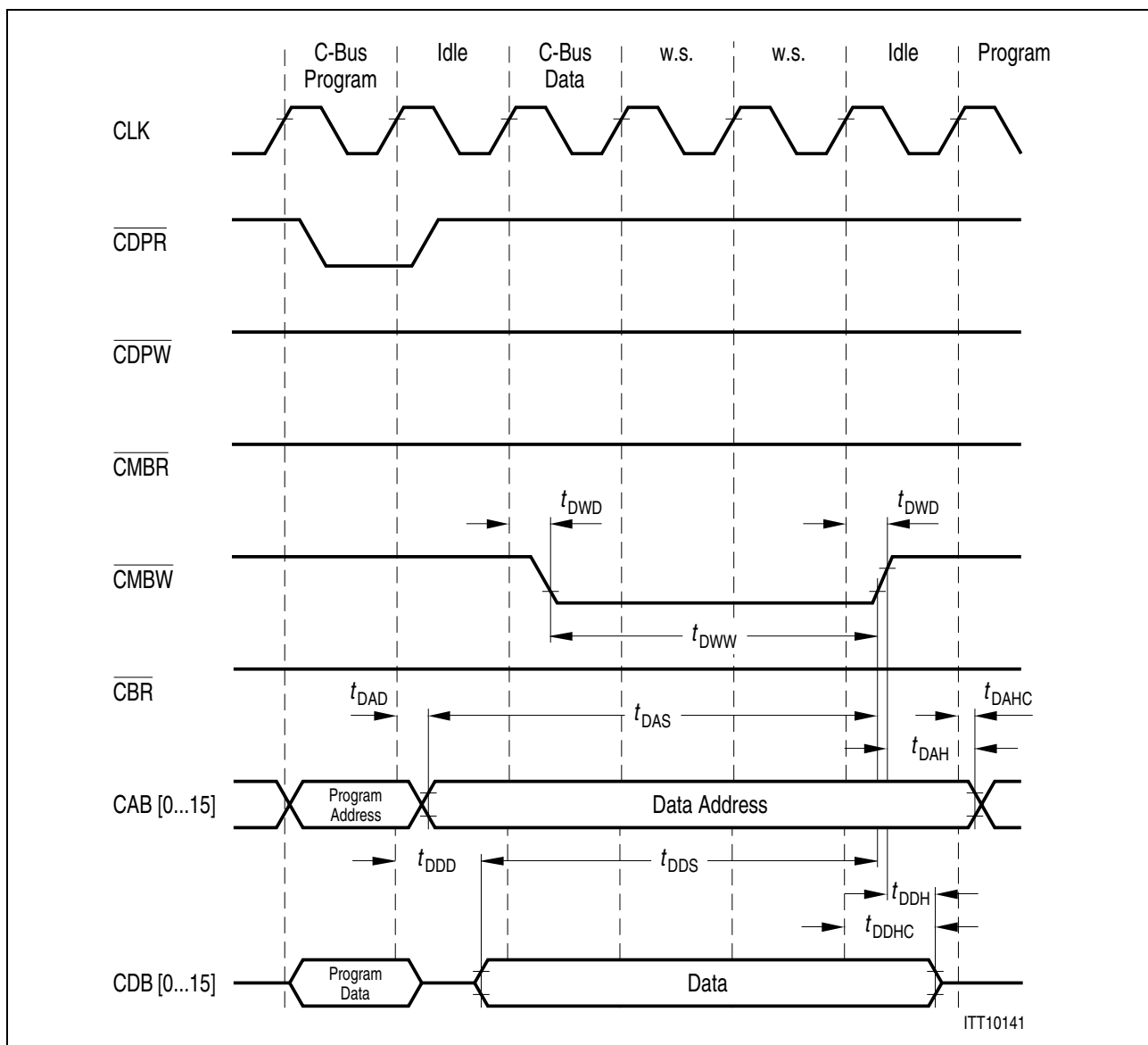
$$t_{\text{DAS}} = \text{TV} + \text{DWS} \times \text{clock-period} = 30 + 3 \times 25 = 105 \text{ ns}$$

$$t_{\text{DDS}} = \text{TV} + \text{DWS} \times \text{clock-period} = 20 + 3 \times 25 = 95 \text{ ns}$$



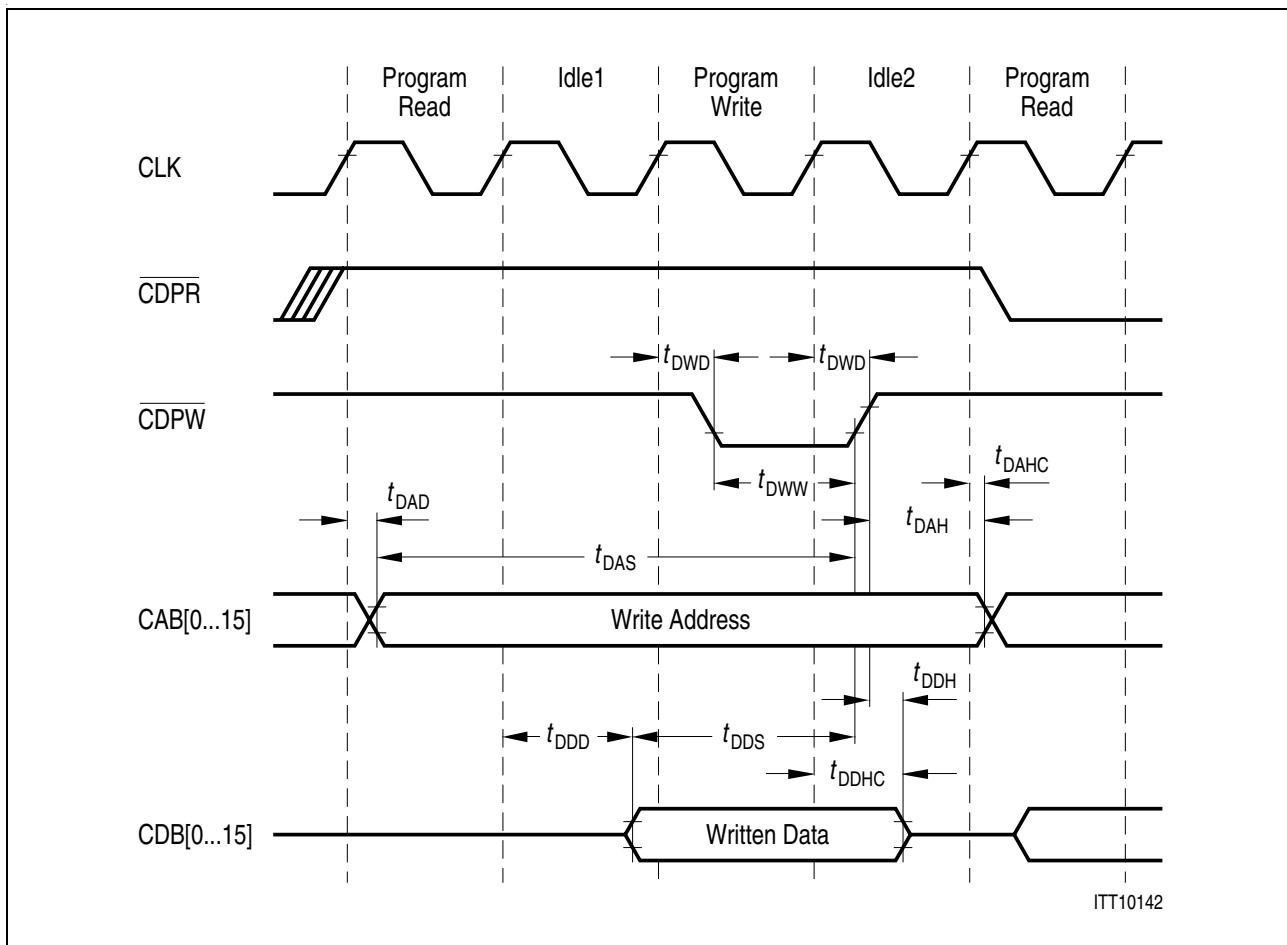
**Figure 7-29 External Data Write Access**

*Note: The figure, above, describes a situation in which the number of wait states is 2 (MEMCONF:DWS = 2, see **section 2.7.3.1**). DWS can be programmed to any value from 0 to 7, and the timing will be changed accordingly.*



**Figure 7-30 Emulation Mail-Box Write Access**

*Note: The above figure describes a situation in which the number of wait states is 2 (MEMCONF:DWS = 2, see **section 2.7.3.1**). DWS can be programmed to any value from 0 to 7, and the timing will be changed accordingly.*



**Figure 7-31 External Program Write Access Due to MOVD**

*Note: MOVD write cycle does not include any wait states, at all.*

**7.9.7 Clocks Signals Timing (additional to the IOM<sup>®</sup>-2 and PCM clocks)**

**Table 7-26 CLK61 (input) Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK61 Clock Period	$t_{C61CP}$	16		ns	61.44 MHz
CLK61 Clock Period Low	$t_{C61CPL}$	5		ns	
CLK61 Clock Period High	$t_{C61CPH}$	5		ns	

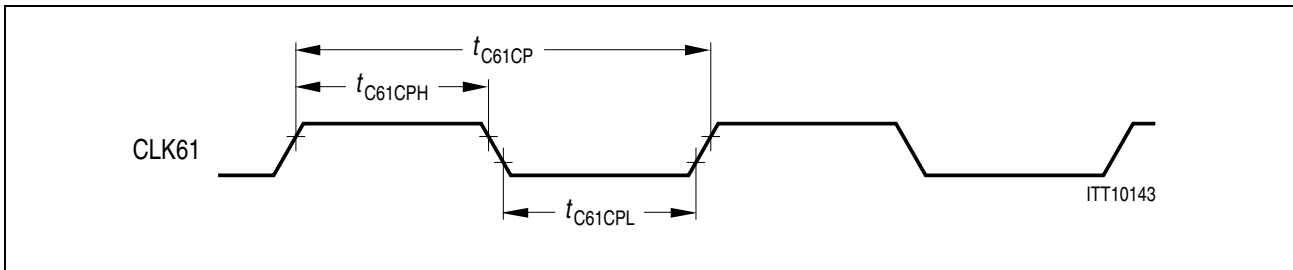


Figure 7-32 CLK61 (input) Timing

Table 7-27 CLK16 (input) Timing

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK16 Clock Period	$t_{C16CP}$	61		ns	16.384 MHz
CLK16 Clock Period Low	$t_{C16CPL}$	20		ns	
CLK16 Clock Period High	$t_{C16CPH}$	20		ns	

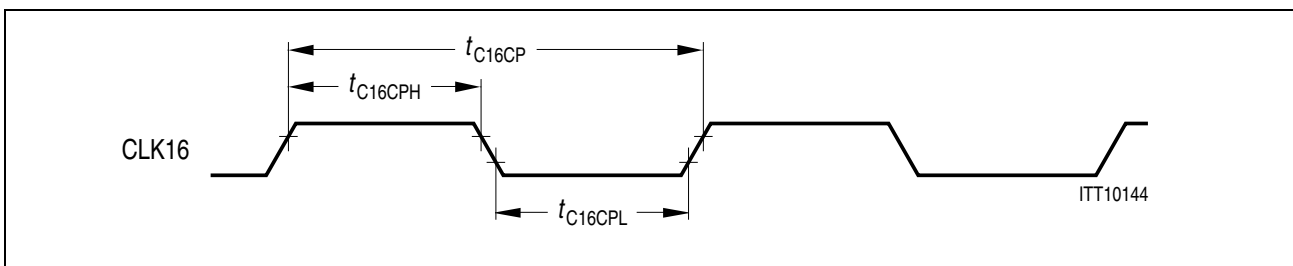


Figure 7-33 CLK16 (input) Timing

Table 7-28 REFCLK Timing

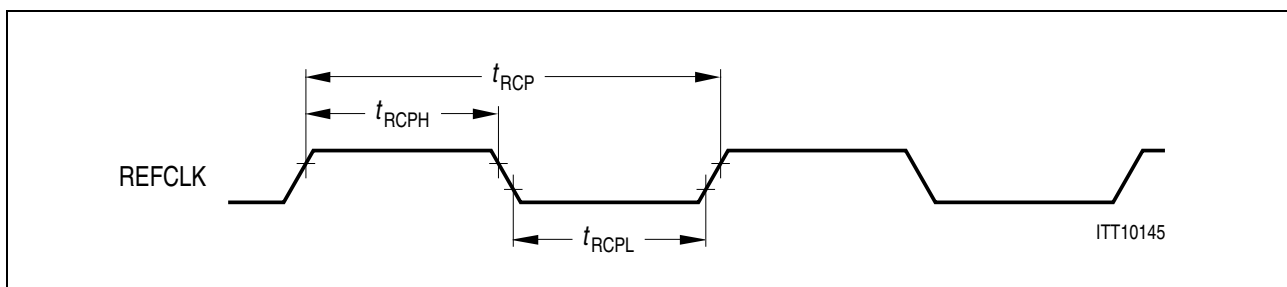
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		

When REFCLK is used as an input

REFCLK Clock Period	$t_{RCP}$	125		$\mu$ S	8 kHz
		1953		ns	512 kHz
		651		ns	1536 kHz
		488		ns	2048 kHz
REFCLK Clock Period Low	$t_{RCPL}$	150		ns	
REFCLK Clock Period High	$t_{RCPH}$	150		ns	

**Table 7-28 REFCLK Timing**

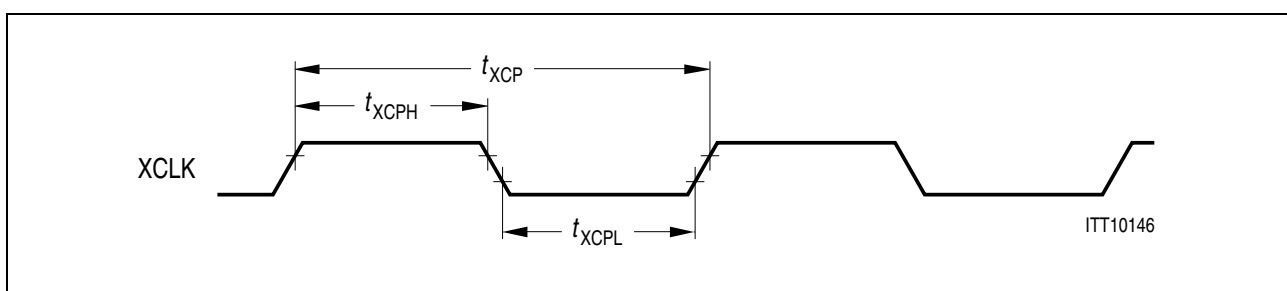
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
<b>When REFCLK is used as an output</b>					
REFCLK Clock Period	$t_{RCP}$	125		$\mu\text{s}$	8 kHz
		1953		ns	512 kHz
REFCLK Clock Period Low	$t_{RCPL}$	100		ns	
REFCLK Clock Period High	$t_{RCPH}$	100		ns	



**Figure 7-34 REFCLK Timing**

**Table 7-29 XCLK (input) Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
XCLK Clock Period	$t_{XCP}$	125		$\mu\text{s}$	8 kHz
		1953		ns	512 kHz
		651		ns	1536 kHz
		488		ns	2048 kHz
XCLK Clock Period Low	$t_{XCPL}$	150		ns	
XCLK Clock Period High	$t_{XCPH}$	150		ns	

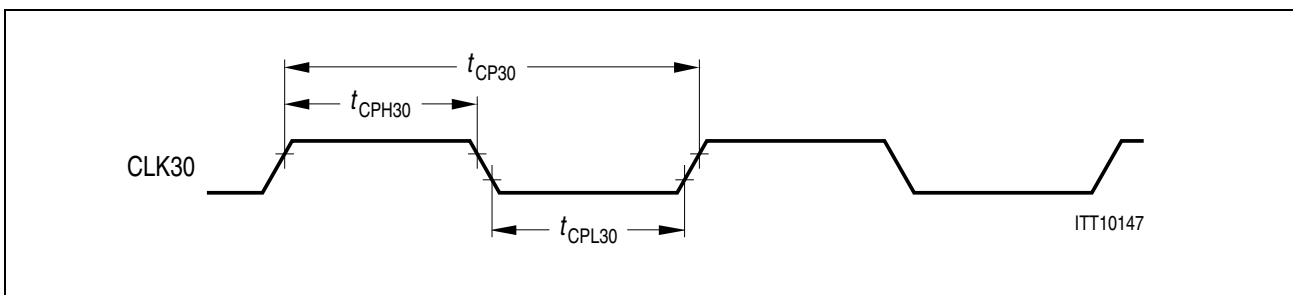


**Figure 7-35 XCLK (input) Timing**



**Table 7-30 CLK30 (output) Timing**

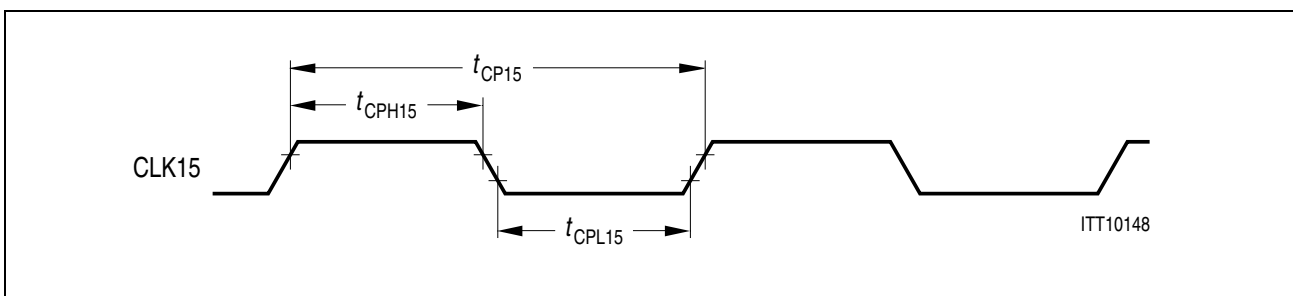
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK30 Clock Period	$t_{CP30}$	32		ns	30.72 MHz when CLK61 = 61.44 MHz
CLK30 Clock Period Low	$t_{CPL30}$	10		ns	
CLK30 Clock Period High	$t_{CPH30}$	10		ns	



**Figure 7-36 CLK30 (output) Timing**

**Table 7-31 CLK15 (output) Timing**

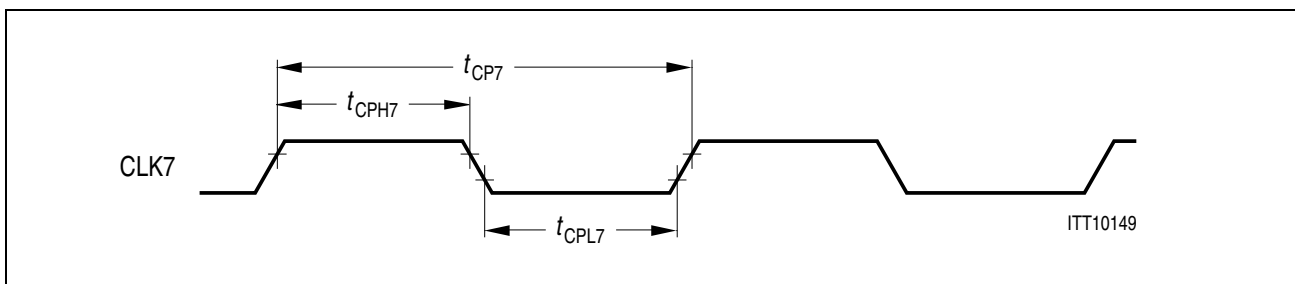
Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK15 Clock Period	$t_{CP15}$	65		ns	15.36 MHz when CLK61 = 61.44 MHz
CLK15 Clock Period Low	$t_{CPL15}$	20		ns	
CLK15 Clock Period High	$t_{CPH15}$	20		ns	



**Figure 7-37 CLK15 (output) Timing**

**Table 7-32 CLK7 (output) Timing**

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
CLK7 Clock Period	$t_{CP7}$	130		ns	7.68 MHz when CLK61 = 61.44 MHz
CLK7 Clock Period Low	$t_{CPL7}$	40		ns	
CLK7 Clock Period High	$t_{CPH7}$	40		ns	



**Figure 7-38 CLK7 (output) Timing**

## 8 Ordering Information and Mechanical Data

### 8.1 Package Outlines

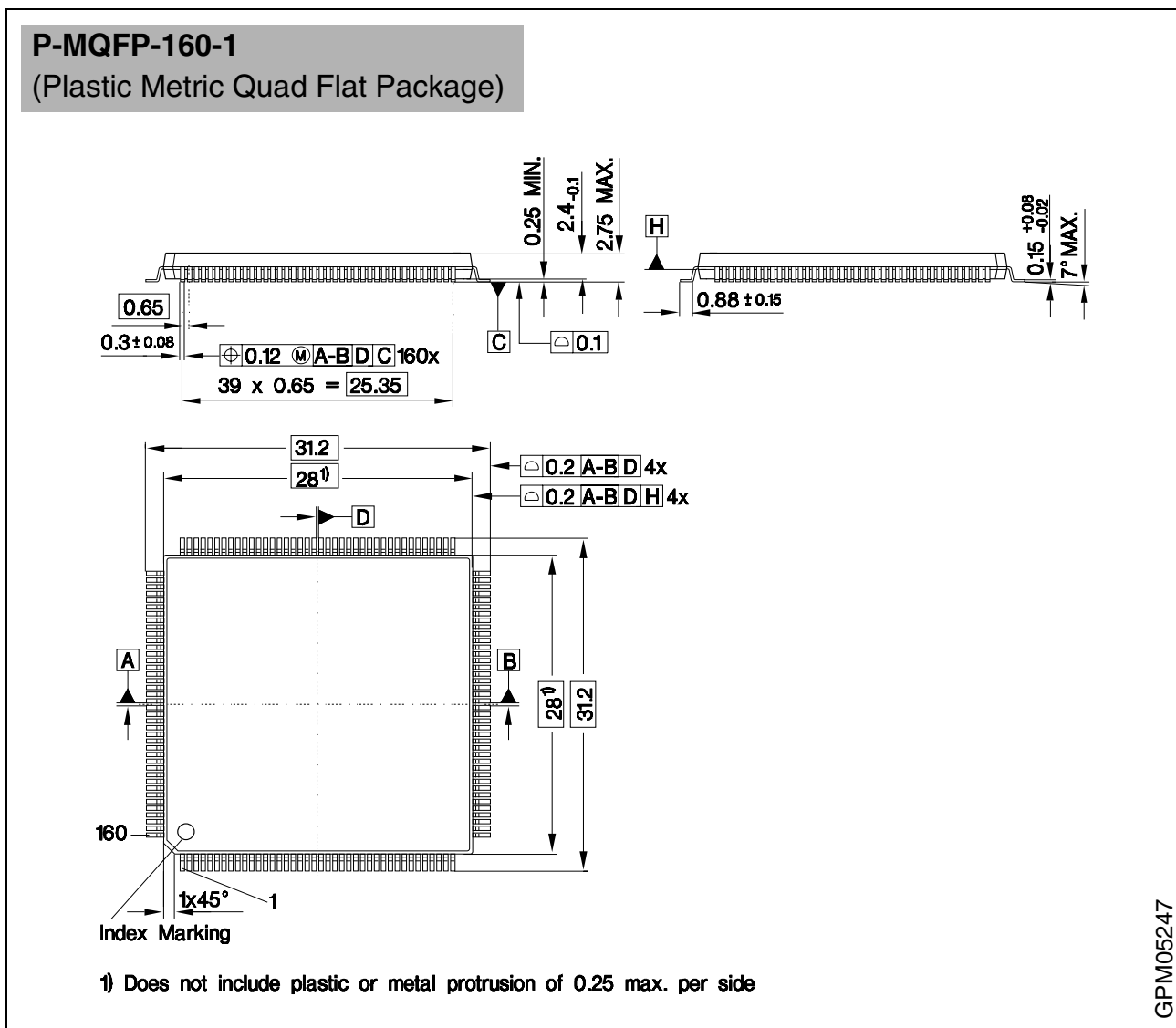


Figure 8-1 Package Outline

#### Sorts of Packing

Package outlines for tubes, trays etc. are contained in our Data Book "Package Information"

SMD = Surface Mounted Device

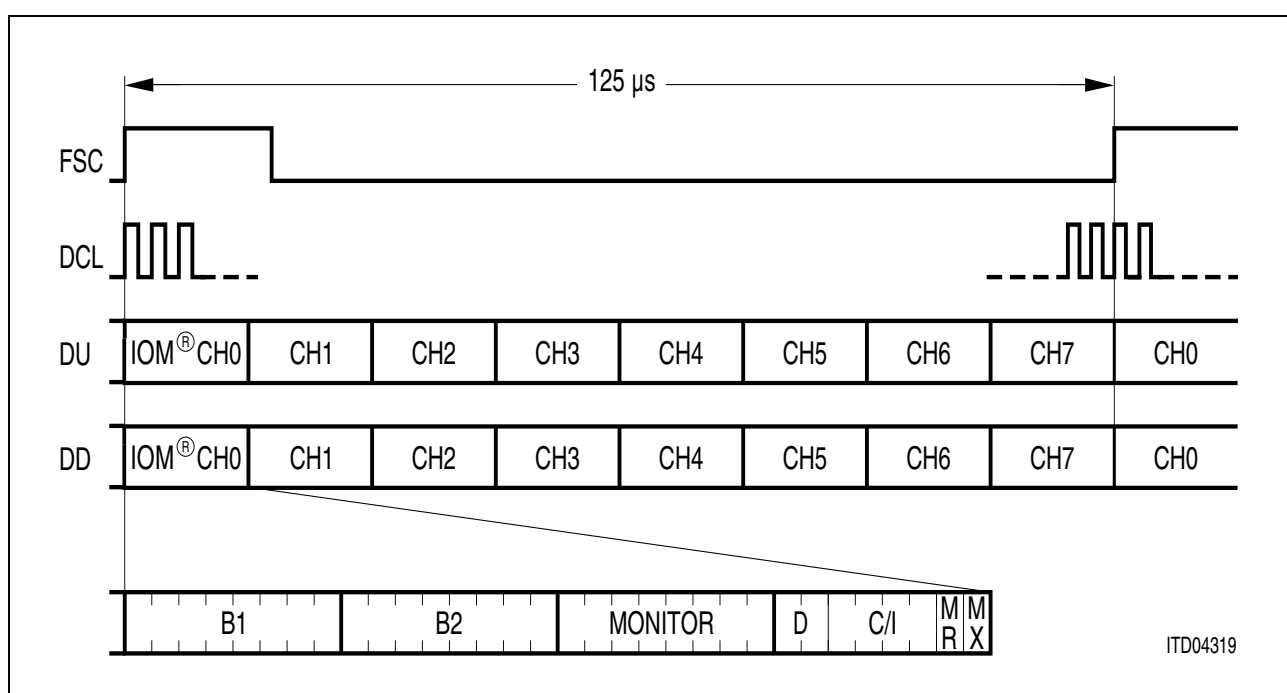
Dimensions in mm

## 9 Appendix

### 9.1 IOM<sup>®</sup>-2 Interface

This standardized interface for interchip communication in ISDN line cards for digital exchange systems was developed by the “Group of Four” (ALCATEL, Siemens, Plessey and ITALTEL systems companies).

The IOM-2 interface is a four-wire interface with a bit clock, a frame clock and one data line per direction. It has a flexible data clock. This way, data transmission requirements are optimized for different applications.



ITD04319

**Figure 9-1 IOM<sup>®</sup>-2 Interface with 4-bit C/I Channel**

#### 9.1.1 Signals / Channels

FSC	Frame Synchronization Clock, 8 kHz
DCL	Data Clock, 2.048 or 4.096 MHz
DD	Data Downstream, 2.048 Mbit/s (4.096 Mbit/s)
DU	Data Upstream, 2.048 Mbit/s (4.096 Mbit/s)
MONITOR	Monitor Channel
D	Signaling Channel, 16 kbit/s
C/I	Command/Indication Channel
MR	Monitor Receive handshake signal
MX	Monitor Transmit handshake signal

### 9.1.2 Monitor and C/I Handlers

The EPIC also supports the Monitor and Command/Indication (C/I) channels in accordance with the IOM-2 interface protocol. The monitor handler controls the data flow on the monitor channel either with or without an active handshake protocol. To reduce the dynamic load of the  $\mu\text{P}$ , a 16-byte transmit/receive FIFO is provided.

The C/I handler supports different schemes:

In downstream direction the relevant content of the control memory (= Command) is transmitted in the appropriate time-slot.

In upstream direction the C/I handler monitors the received data (= Indication). Upon a change it generates an interrupt, stores the channel address in the 9-byte deep C/I-FIFO and the actual C/I value in the control memory. Double last-look check is provided.

A 7-bit hardware timer can be used to interrupt the  $\mu\text{P}$  or DSP cyclically to determine the double last-look period.

For more details please refer to IOM-2 Interface Reference Guide 3.91

### 9.1.3 IOM<sup>®</sup>-2 Interface Timing

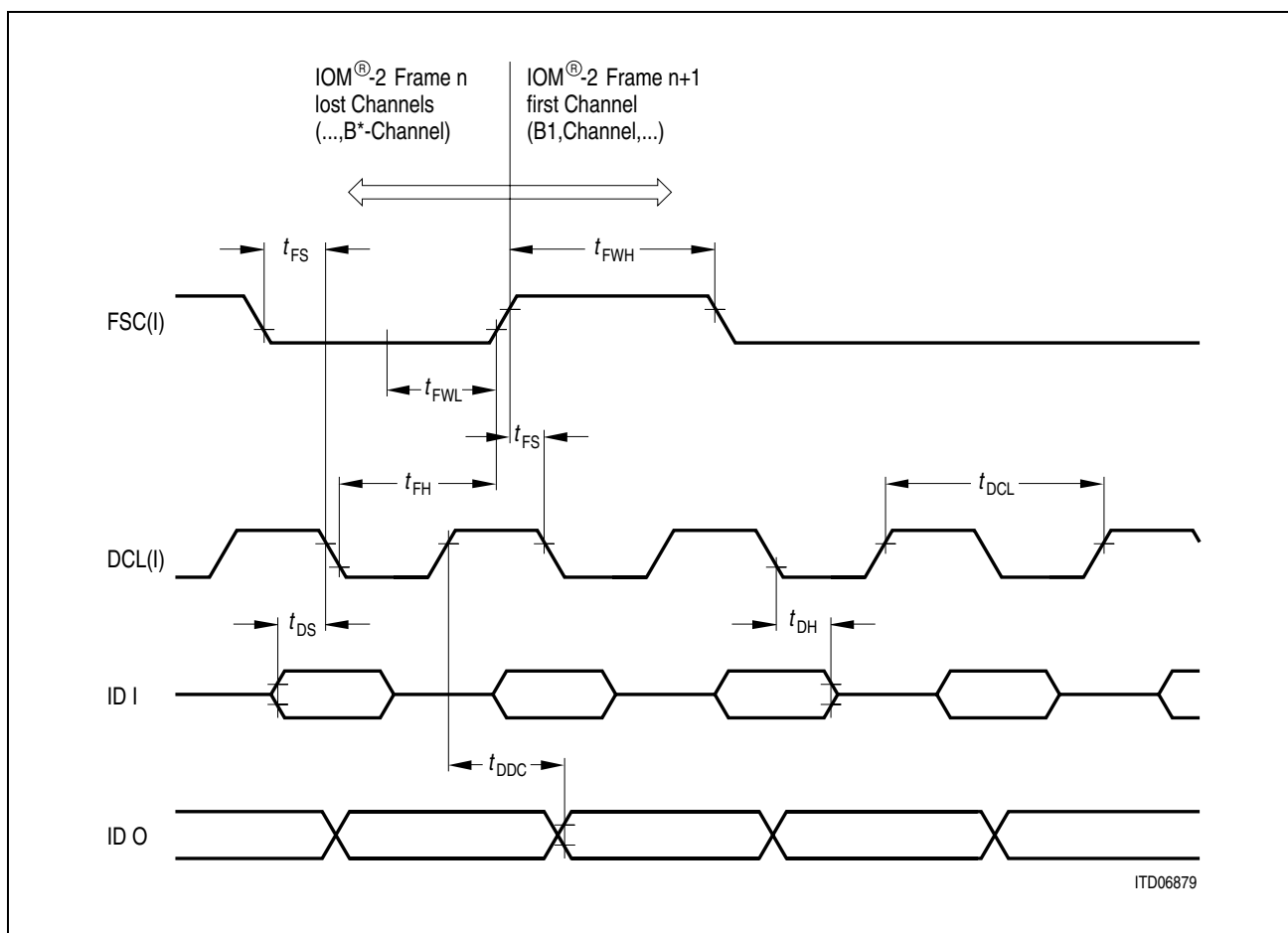
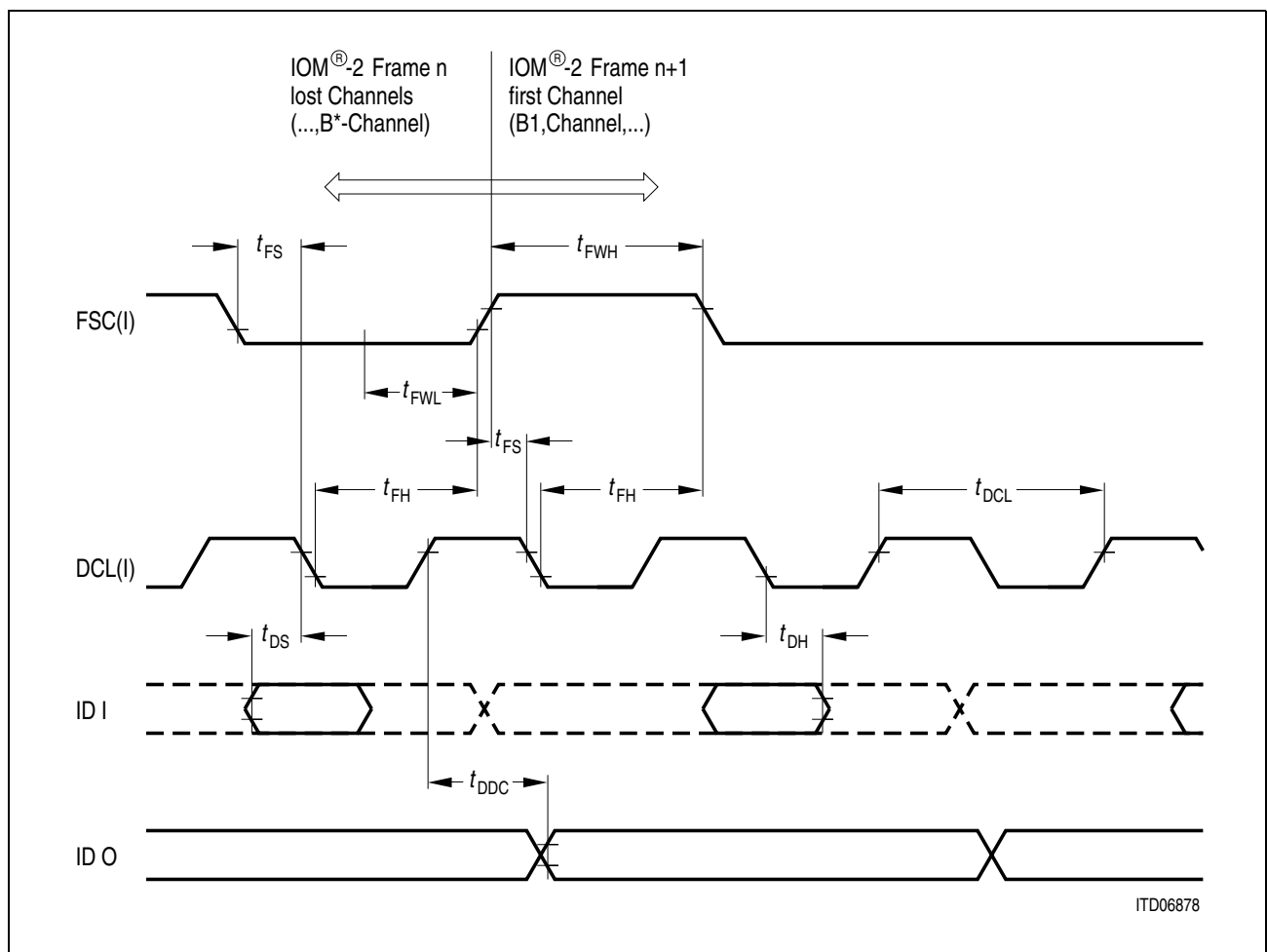


Figure 9-2 IOM<sup>®</sup>-2 Interface Timing with Single Data Rate DCL

**Table 9-1 Timing Characteristics of the IOM<sup>®</sup>-2**

Parameter	Symbol	Limit Values			Unit	Condition
		min.	typ.	max.		
Frame sync. hold	$t_{FH}$	30			ns	
Frame sync. setup	$t_{FS}$	70			ns	
Frame sync. high	$t_{FWH}$	130			ns	
Frame sync. low	$t_{FWL}$	$t_{DCL}$				
Data delay to clock	$t_{DDC}$			100	ns	
Data delay to frame <sup>1)</sup>	$t_{DDF}$			150	ns	
Data setup	$t_{DS}$	20			ns	
Data hold	$t_{DH}$	50			ns	

1)  $t_{DDF} = 0.5 t_{DCL} + t_{DDC} - t_{FH}$



ITD06878

**Figure 9-3 Timing of the IOM<sup>®</sup>-2 Interface with Double Data Rate DCL**

**Table 9-2 Timing Characteristics of the IOM<sup>®</sup>-2**

Parameter	Symbol	Limit Values			Unit	Condition
		min.	typ.	max.		
Frame sync hold	$t_{FH}$	30			ns	
Frame sync setup	$t_{FS}$	70			ns	
Frame sync high	$t_{FWH}$	130			ns	
Frame sync low	$t_{FWL}$	$t_{DCL}$				
Data delay to clock	$t_{DDC}$			100	ns	
Data delay to frame <sup>1)</sup>	$t_{DDF}$			150	ns	
Data setup	$t_{DS}$	20			ns	
Data hold	$t_{DH}$	50			ns	

<sup>1)</sup>  $t_{DDF} = 0.5 t_{DCL} + t_{DDC} - t_{FH}$

## 9.2 Working Sheets for Multiplexers Programming

### 9.3 Working Sheets

The following pages contain working sheets to facilitate the programming of the DOC incl. EPIC-1. For several tasks (i.e. initialization, time-slot switching, ...) the corresponding registers are summarized in a way the programmer gets a quick overview on the registers he has to use.

- Register Summary for EPIC Initialisation
- Switching of PCM Time-Slots to the CFI Interface (Data Downstream)
- Switching of CFI Time-Slots to the PCM Interface (Data Upstream)
- Preparing EPICs C/I Channels
- Receiving and Transmitting IOM-2 C/I Codes
- DOC Port and Signaling Multiplexers
- DOC Clocking Multiplexers





**POFD** PCM Offset Downstream Register RW, 24<sub>H</sub> (2<sub>H</sub> + RBS = 1), reset-val. = 0

OFD9..2
---------

OFD2...9 = Offset Downstream (see PCSR for OFD0...1)

Mode 0:  $(BND - 17 + BPF) \bmod BPF \rightarrow OFD2...9$

Mode 1:  $(BND - 33 + BPF) \bmod BPF \rightarrow OFD1...9$

Mode 2:  $(BND - 65 + BPF) \bmod BPF \rightarrow OFD0...9$

BND = number of bits + 1 that the downstream frame start is left shifted relative to the frame sync

BPF = number of bits per frame

Unused bits must be set to 0 !

**POFU** PCM Offset Upstream Register RW, 26<sub>H</sub> (3<sub>H</sub> + RBS = 1), reset-val. = 0

OFU9..2
---------

OFU2...9 = Offset Upstream (see PCSR for OFU0...1)

Mode 0:  $(BND + 23 + BPF) \bmod BPF \rightarrow OFU2...9$

Mode 1:  $(BND + 47 + BPF) \bmod BPF \rightarrow OFU1...9$

Mode 2:  $(BND + 95 + BPF) \bmod BPF \rightarrow OFU0...9$

BND = number of bits + 1 that the upstream frame is left shifted relative to the frame start

BPF = number of bits per frame

Unused bits must be set to 0 !

**PCSR** PCM Clock Shift Register RW, 28<sub>H</sub> (4<sub>H</sub> + RBS = 1), reset-val. = 0

0	OFD1...0	DRE	0	OFU1...0	URE
---	----------	-----	---	----------	-----

OFD0...1 = Offset Downstream (see POFD)

DRE = Downstream Rising Edge,

0 = receive data on falling edge,

1 = receive data on rising edge

OFU0...1 = Offset Upstream (see POFU)

URE = Upstream Rising Edge,

0 = send data on falling edge,

1 = send data on rising edge

**Figure 9-4b EPIC® Initialization Register Summary (working sheet)**

**CFI Interface**

**CMD1** CFI Mode Register 1 RW, 2C<sub>H</sub> (6<sub>H</sub> + RBS = 1), reset-val. = 00

CSS	CSM	CSP1...0	CMD1...0	CIS1...0
-----	-----	----------	----------	----------

CSS = Clock Source Select,

0 = PDC/PFS used for CFI,

1 = DCL/FSC are inputs

CSM = CFI Synchronization Mode:

1 = frame syncr. with rising edge,

0 = falling edge of DCL

if CSS = 0 ==> CMD1:CSM = PMOD:PSM !

CSP0...1 = Clock Source Prescaler: 00 = 1/2, 01 = 1/1.5, 10 = 1/1

CMD0...1 = CFI Mode: 00 = 0, 01 = 1, 10 = 2, 11 = 3

CIS0...1 = CFI Alternative Input Section

Mode 0, 3: CIS0...1 = 0

Mode 1, 2: CIS0: 0 = IN0 = DU0, 1 = IN0 = DU2

Mode 1: CIS1: 0 = IN1 = DU1, 1 = IN1 = DU3

**CMD2** CFI Mode Register 2 RW, 2E<sub>H</sub> (7<sub>H</sub> + RBS = 1), reset-val. = 00

FC2...0	COC	CXF	CRR	CBN9...8
---------	-----	-----	-----	----------

**For IOM<sup>®</sup>-2 CMD2 can be set to D0<sub>H</sub>**

FC0...2 = Framing Signal Output Control (CMD1:CSS = 0)

= 010 suitable for PBC, = 011 for IOM-2, = 110 IOM-2 and SLD

COC = Clock Output Control (CMD1:CSS = 0)

= 0 DCL = data rate,

= 1 DCL 2 × data rate (only mode 0 and 3 !)

CXF = CFI Transmit on Falling Edge: 0 = send on rising edge, 1 = send on falling DCL edge

CRR = CFI Receive on Rising Edge: 0 = receive on falling edge, 1 = send on rising DCL edge

CBN8...9 = CFI Bit Number (see CBNR)

**CBNR** CFI Bit Number Register RW, 30<sub>H</sub> (8<sub>H</sub> + RBS = 1), reset-val. = FF

CBN
-----

CBN0...7 = CFI Bit Number per Frame – 1 (see CMD2:CBN8...9)

**Figure 9-4c EPIC<sup>®</sup> Initialization Register Summary (working sheet)**

**CTAR** CFI Time-Slot Adjustment Register RW, 32<sub>H</sub> (9<sub>H</sub> + RBS = 1), reset-val. = 00

0	TSN
---	-----

TSN0...6 = (number of time-slots + 2) the DU and DD frame is left shifted relative to frame start (see also CBSR)

**CBSR** CFI Bit Shift Register RW, 34<sub>H</sub> (A<sub>H</sub> + RBS = 1), reset-val. = 00

0	CDS2...0	CUS3...0
---	----------	----------

CDS2...0: CFI Downstream/Upstream Bit Shift

Shift DU and DD frame:

- 000 = 2 bits right
- 001 = 1 bit right
- 010 = 6 bits left
- 011 = 5 bits left
- 100 = 4 bits left
- 101 = 3 bits left
- 110 = 2 bits left
- 111 = 1 bit left

Relative to PFS (if CMD1:CSS = 0)

Relative to FSC (if CMD1:CSS = 1)

**CSCR** CFI Subchannel Register RW, 36<sub>H</sub> (A<sub>H</sub> + RBS = 1), reset-val. = 00

CS3	CS2	CS1	CS0
-----	-----	-----	-----

SC3 0...1 control port 3 (+ port 7 for CFI mode 3 (SLD))

SC2 0...1 control port 2 (+ port 6 for CFI mode 3 (SLD))

SC1 0...1 control port 1 (+ port 5 for CFI mode 3 (SLD))

SC0 0...1 control port 0 (+ port 4 for CFI mode 3 (SLD))

for 64 kBit/s channel: 00/01/10/11 = bits 7...0

for 32 kBit/s channel: 00/10 = bits 7...4,

01/11 = bits 3...0

**Figure 9-4d EPIC® Initialization Register Summary (working sheet)**

### 9.3.2 Switching of PCM Time-Slots to the CFI Interface (Data Downstream)

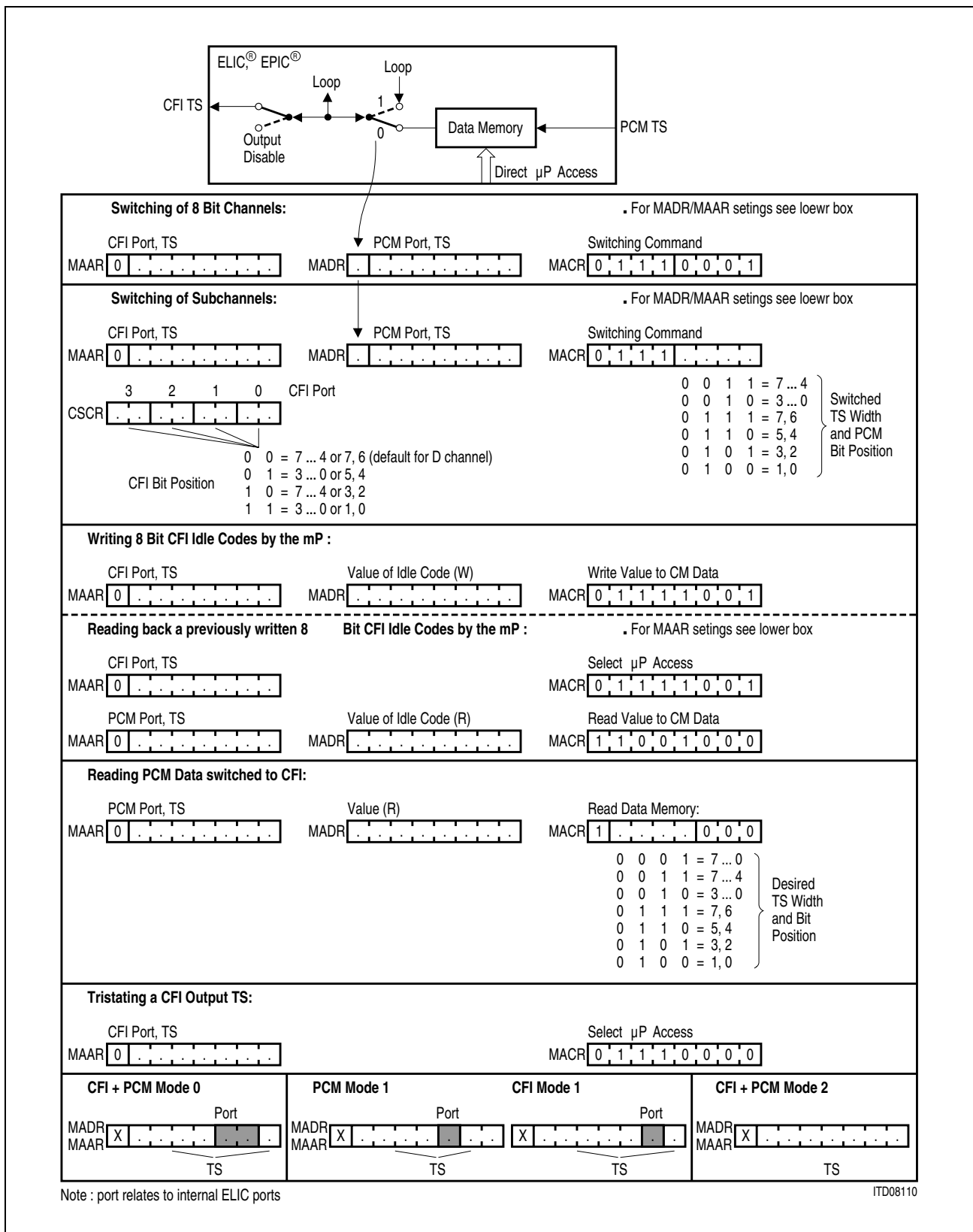


Figure 9-5 Switching of PCM Time-Slots to the CFI Interface (working sheet)

### 9.3.3 Switching of CFI Time-Slots to the PCM Interface (Data Upstream)

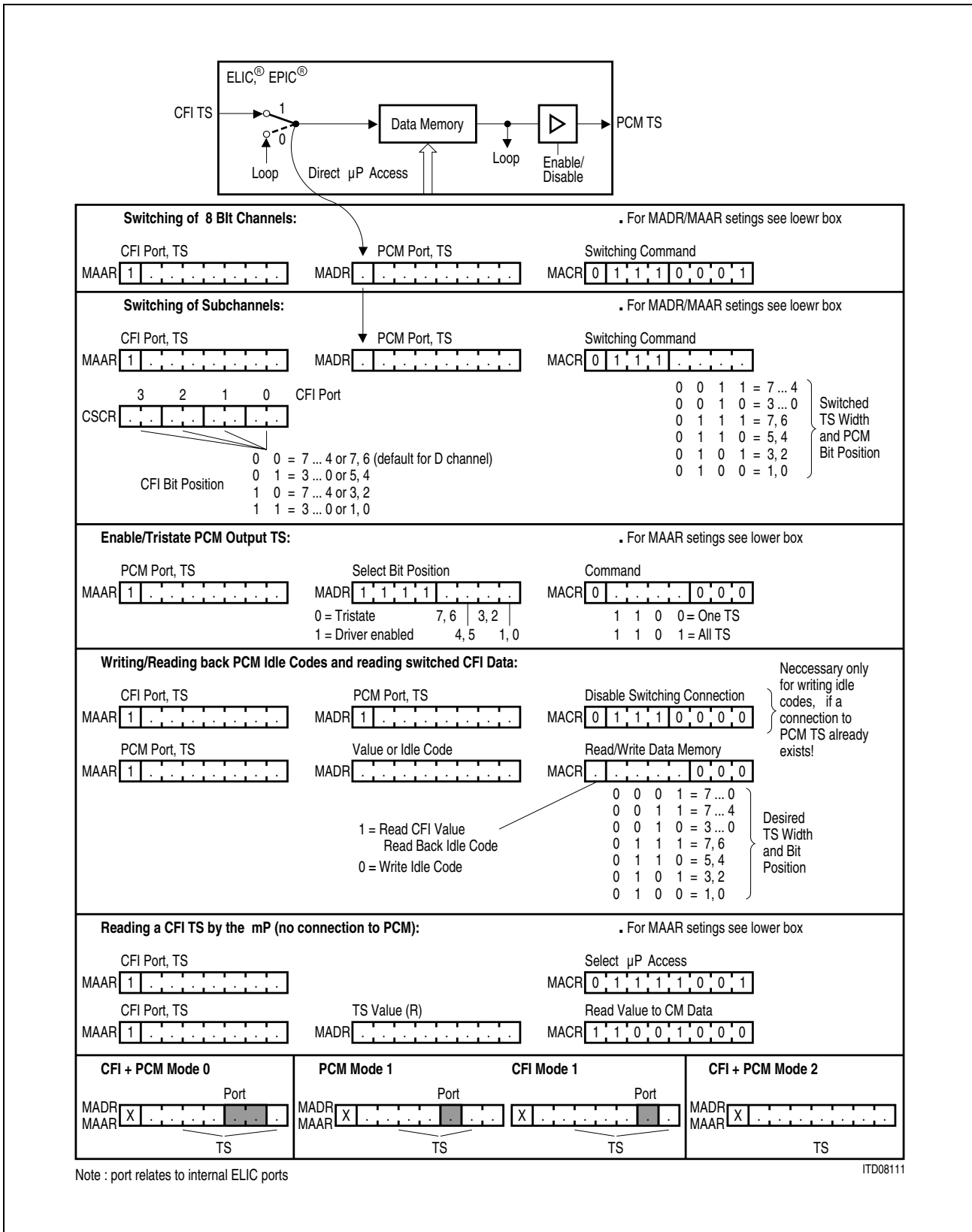
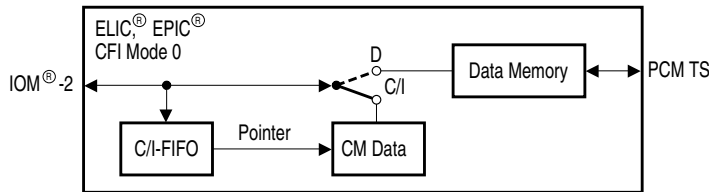


Figure 9-6 Switching of CFI Time-Slots to the PCM Interface (working sheet)

### 9.3.4 Preparing EPICs C/I Channels



**Initialization of the C/I Channels Data Downstream:**

IOM <sup>®</sup> -2 Channel MAAR <b>0</b> . . . . . <b>1</b> . . . . . <b>0</b> Port	C/I Idle Code MADR . . . . . <b>1</b> <b>1</b> 1 1 4 Bit C/I 6 Bit C/I	Mode Selection MACR <b>0</b> <b>1</b> <b>1</b> <b>1</b> . . . . . 1 0 0 0 = Decentral D Channel Handling 1 0 1 0 = Central D Channel Handling 1 0 1 0 = 6 Bits Analog C/I 1 0 1 0 = ELIC <sup>®</sup> SACCO-A D Channel Handling
IOM <sup>®</sup> -2 Channel MAAR <b>0</b> . . . . . <b>1</b> . . . . . <b>1</b> Port	PCM Port, TS MADR <b>0</b> . . . . . Only for central D Channel Handling !	Mode Selection MACR <b>0</b> <b>1</b> <b>1</b> <b>1</b> . . . . . 1 0 1 1 = Decentral D Channel Handling 0 1 X X = Central D Channel Handling 1 1 = PCM TS Bit 7, 6 1 0 = PCM TS Bit 5, 4 0 1 = PCM TS Bit 3, 2 0 0 = PCM TS Bit 1, 0 1 0 1 1 = Analog C/I 1 0 1 1 = ELIC <sup>®</sup> SACCO-A

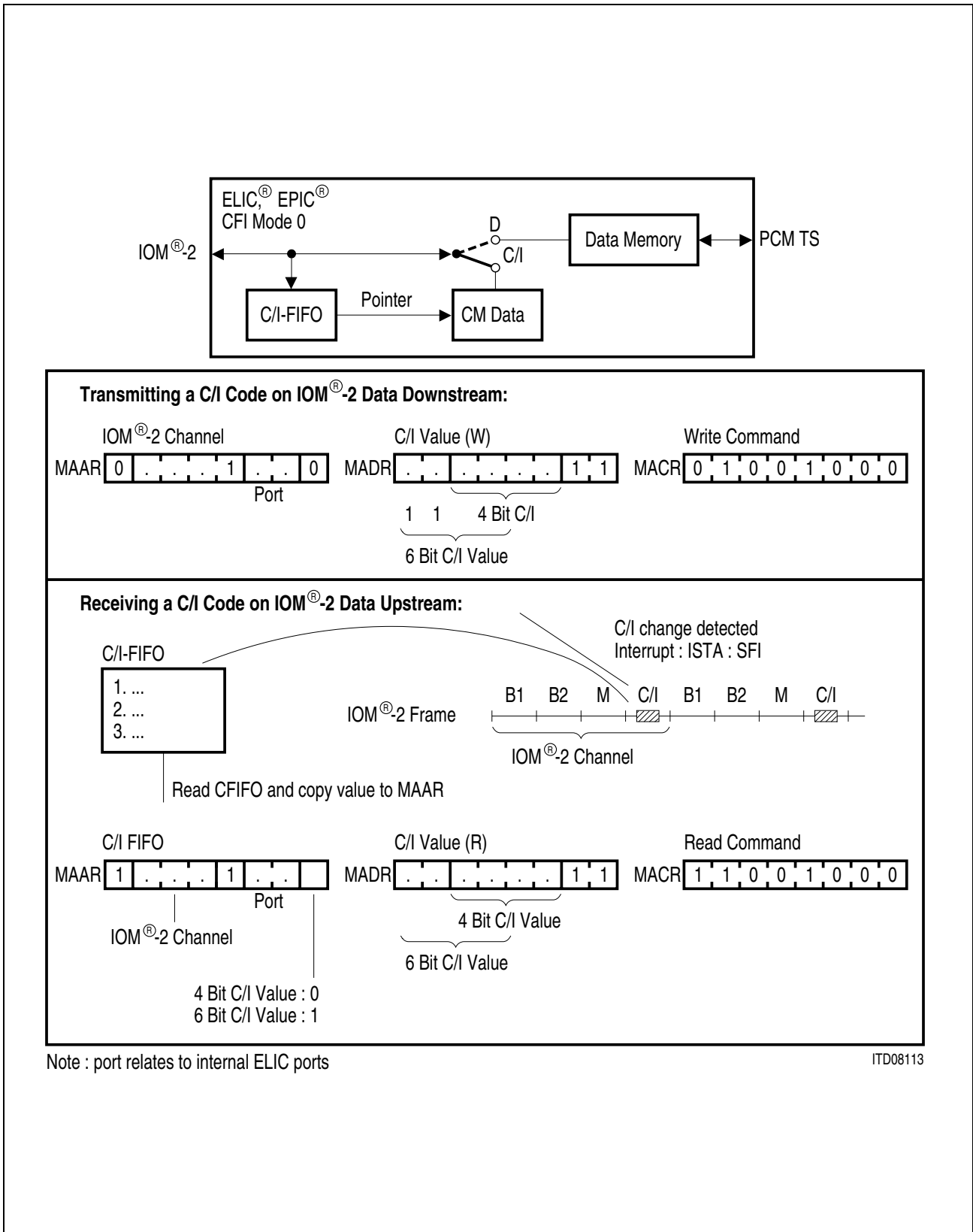
**Initialization of the C/I Channels Data Upstream:**

IOM <sup>®</sup> -2 Channel MAAR <b>1</b> . . . . . <b>1</b> . . . . . <b>0</b> Port	C/I Idle Code MADR . . . . . <b>1</b> <b>1</b> Expected C/I-Value	Mode Selection MACR <b>0</b> <b>1</b> <b>1</b> <b>1</b> . . . . . 1 0 0 0 = Decentral D Channel Handling 1 0 0 0 = Central D Channel Handling 1 0 1 0 = 6 Bit Analog C/I 1 0 0 0 = ELIC <sup>®</sup> SACCO-A D Channel Handling
IOM <sup>®</sup> -2 Channel MAAR <b>1</b> . . . . . <b>1</b> . . . . . <b>1</b> Port	PCM Port, TS MADR <b>1</b> . . . . . Only for central D Channel Handling !	Mode Selection MACR <b>0</b> <b>1</b> <b>1</b> <b>1</b> . . . . . 0 0 0 0 = Decentral D Channel Handling 0 1 X X = Central D Channel Handling 1 1 = PCM TS Bit 7, 6 1 0 = PCM TS Bit 5, 4 0 1 = PCM TS Bit 3, 2 0 0 = PCM TS Bit 1, 0 1 0 1 0 = Analog C/I 0 0 0 0 = ELIC <sup>®</sup> SACCO-A D Channel Handling
	Expectend C/I Value MADR . . . . . <b>1</b> <b>1</b> Only analog 6 Bit C/I Handling !	

ITD08112

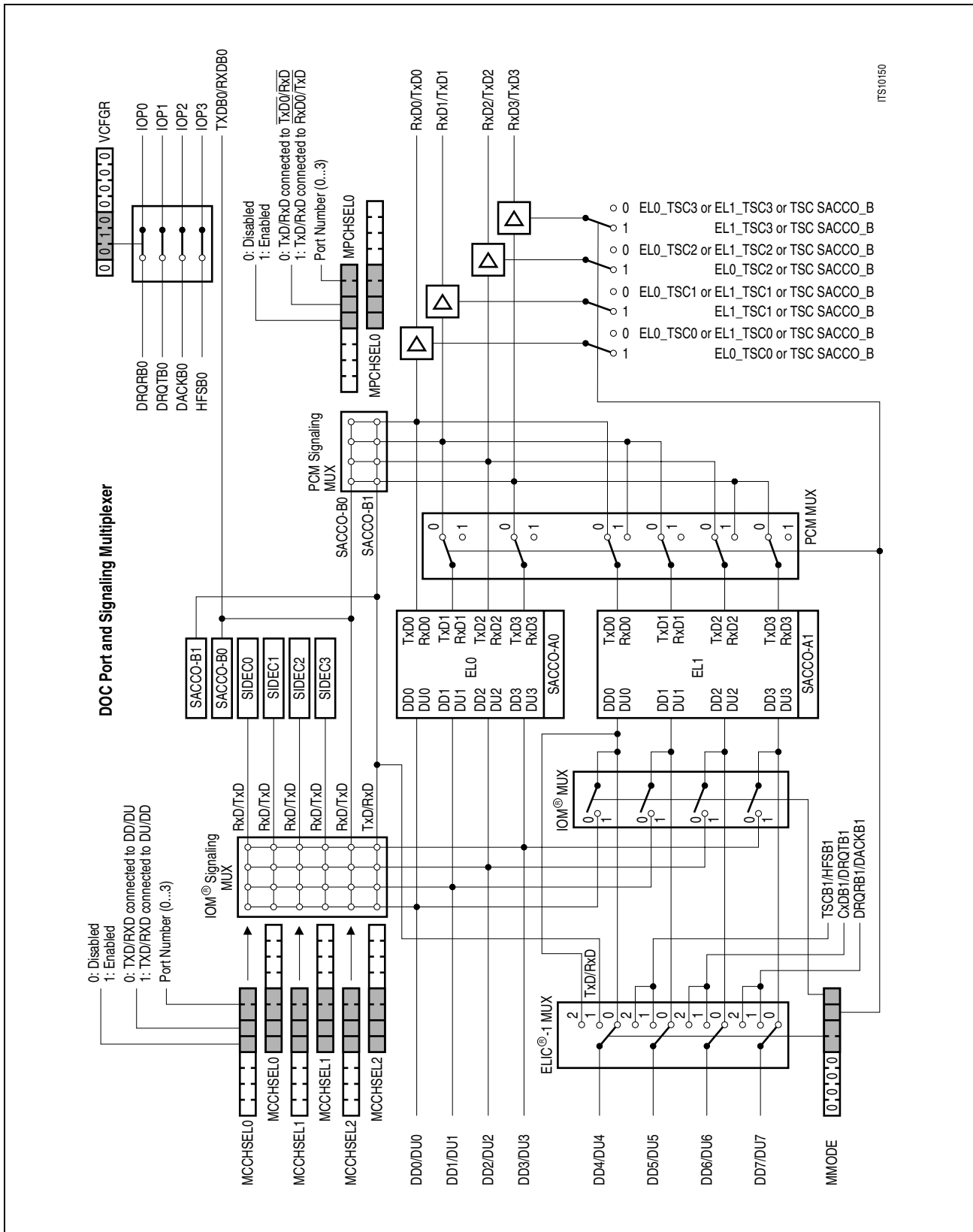
Figure 9-7 Preparing EPIC<sup>®</sup>s C/I Channels (working sheet)

### 9.3.5 Receiving and Transmitting IOM<sup>®</sup>-2 C/I-Codes



**Figure 9-8 Receiving and Transmitting IOM<sup>®</sup>-2 C/I-Codes (working sheet)**

### 9.3.6 DOC Port and Signaling Multiplexers



ITS10150

Figure 9-9



### 9.3.7 DOC Clcking Multiplexers

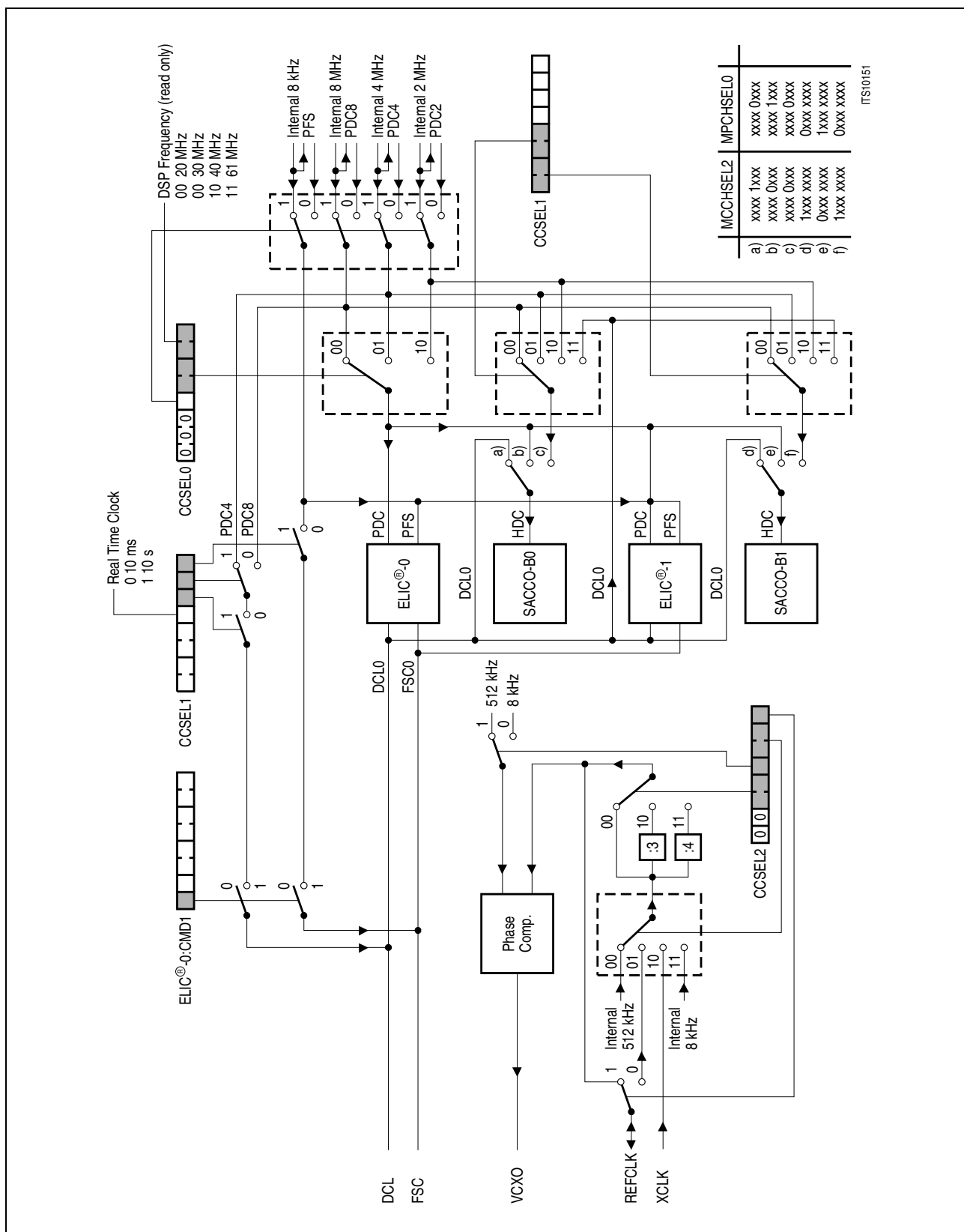


Figure 9-10

## **9.4 Development Tools and Software Support**

### **9.4.1 Software**

The software development tools help to minimize the time to market and development costs. The software consists of: Macro Assembler, Linker/Locator, Object Format Converter, ANSI C Compiler, Simulator with a Debugger for OCEM.

All DSP Software Development Tools can operate on Microsoft.

### **9.4.2 Macro Assembler**

The Macro Assembler translates DSP assembly language source files into DSP machine language object files. It consists of a macro preprocessor which checks DSP programming restrictions and prepares the object for full symbolic debugging. It contains C-like operators and conventions that allow easy development of code and data structures. The object files generated are compatible to the Common Object File Format (COFF).

### **9.4.3 Linker/Locator**

The Linker/Locator combines object files generated by the COFF Macro Assembler into a single executable COFF object file. As it creates the executable object file, it performs relocation which means map them to the target systems memory map. It also supports user defined memory classes and enables to locate segments to absolute locations or relative to other segments and to overlay segments. Its linking capability is very flexible and modular.

### **9.4.4 C Compiler**

The C cross compiler is a state-of-the-art compiler providing 2 options for generating efficient DSP object code:

1. to mix object files generated by the compiler with those generated by the assembler directly from efficient hand coded assembly instructions,
2. to use DSP specific C language extensions.

### **9.4.5 Object Format Converter**

Most EPROM programmers do not accept executable COFF object files as input. Therefore the Object Format Converter translates the COFF file into Intel hex file format that can be downloaded to any ordinary EPROM programmer.

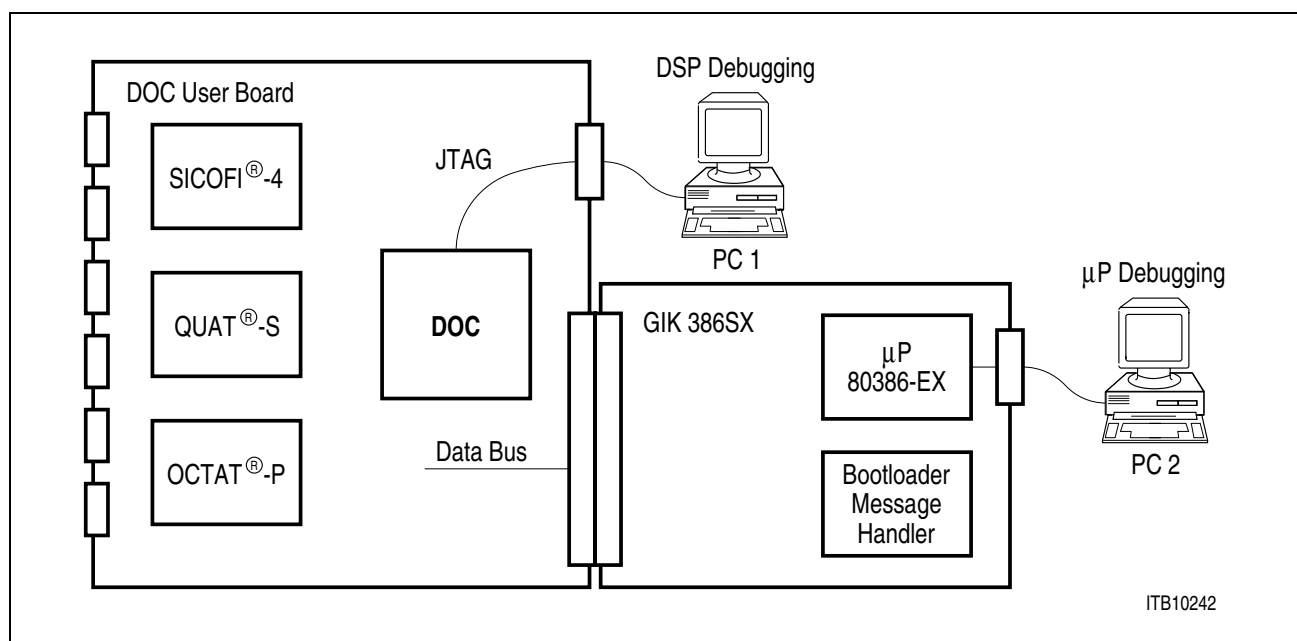
### **9.4.6 Simulator**

The Simulator simulates the operation of the DSP for program verification and debugging purposes. It simulates the entire DSP instruction set and accepts executable

COFF object code generated from the Linker/Locator. The simulator allows verification and monitoring of the DSP states without the requirement of the DSP hardware. Besides a windowed mouse driven interface which can be user-customized it also contains a high level language debug interface. To simulate external signals or hardware logic, it is possible to connect DOS files and integrate C functions using the Dynamic Link Library (DLL) mechanism of WINDOWS. During program execution, the internal registers and memory of the simulated DSP are modified as each instruction is interpreted by the host. Execution is suspended when either a breakpoint or an error is encountered or when the user halts execution. Then the DSP internal registers and both program and data memory can be inspected and modified.

### 9.5 OAK Development / Evaluation Board

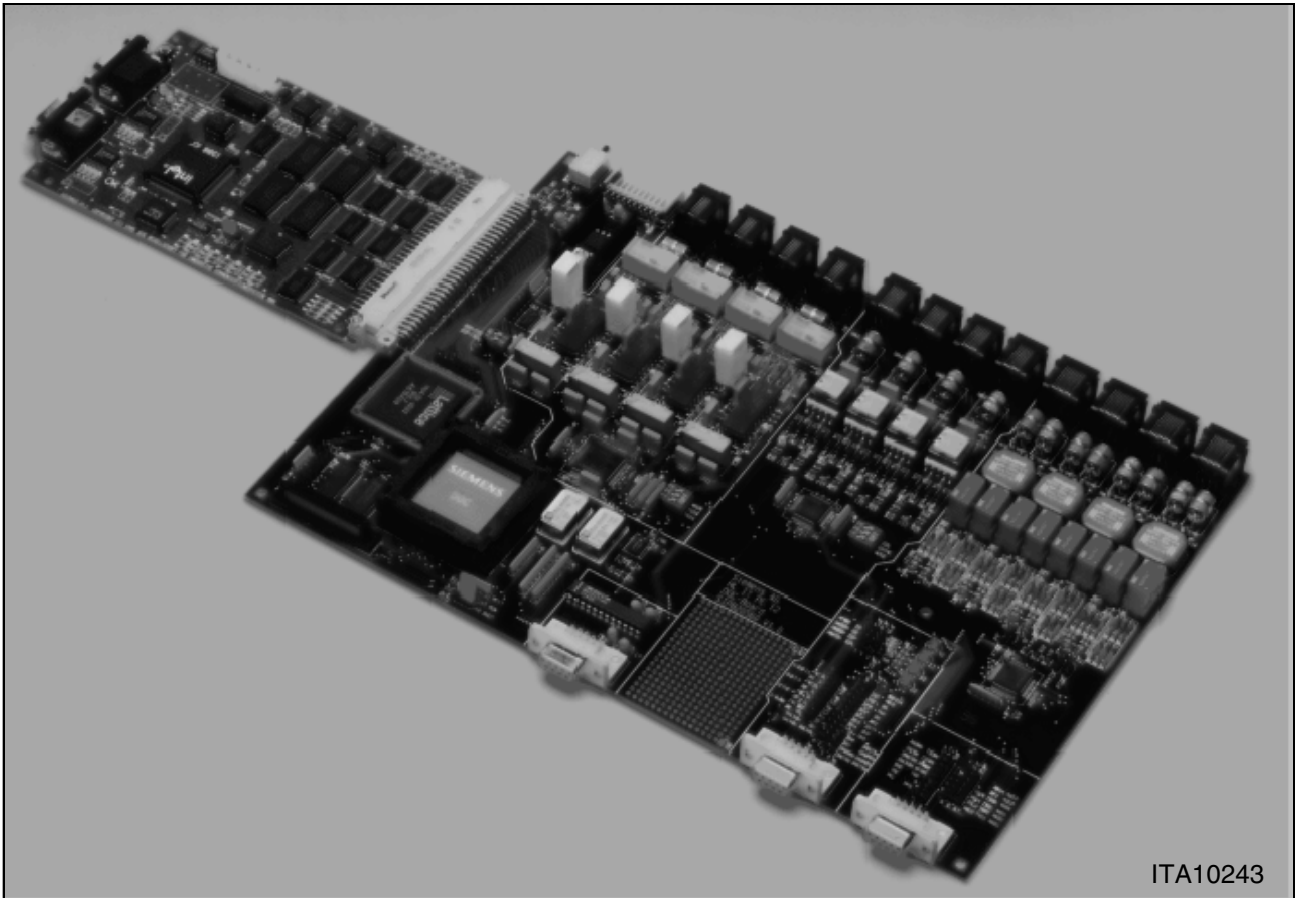
Siemens provides a development board which can be used for a quick start for a new project; to develop software, test interfaces, memory configuration, critical ICs, etc. Both, the Microprocessor and the DSP can be controlled at a time via two different serial interfaces, **Figure 9-11**.



**Figure 9-11 DOC Evaluation Board - Block Diagram**

Thus the software development engineers can interactively read from and write to all DOC registers.

The DOC evaluation board, **Figure 9-12**, implements the full hardware functionality of a small PBX and supports many features of the DOC within an INTEL 386EX based microprocessor system environment. With the DOC board Siemens will also offer a PBX oriented and basic software modules.



**Figure 9-12 DOC Evaluation System**

As the development board differs from the target application the user software must be ported to the final hardware.

Different DSP Software and Development Tools can be provided for the following PC operating systems: Windows 3.1, Windows 95 and Windows NT.



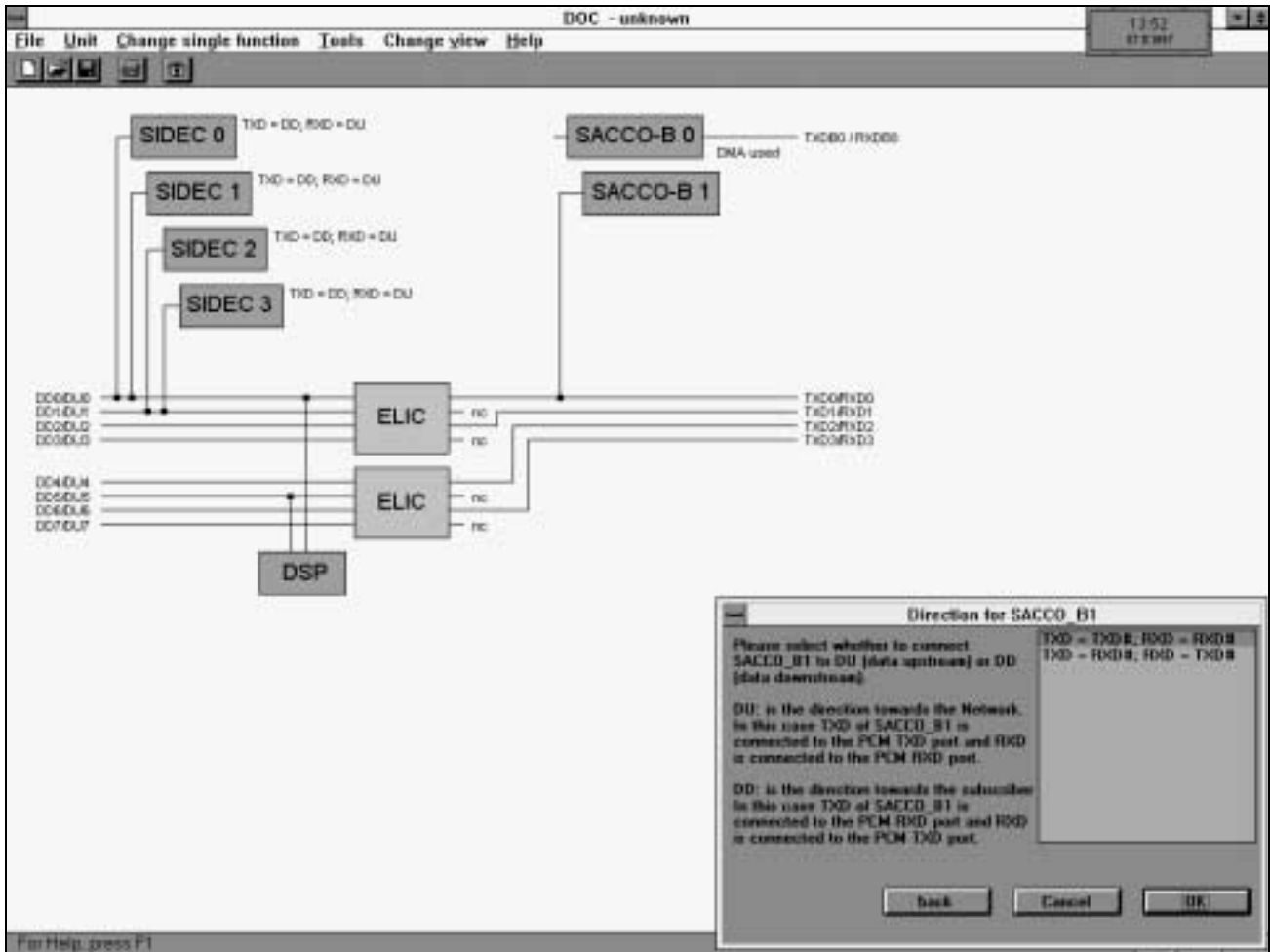


Figure 9-14 Example for Selection of ELIC Clocks

Note: The DOC Configurator runs at any standard PC.

**10****Acronym****A**

ALE Address Latch Enable

ASM Arbiter State Machine

**B**

B-channel 64 kbit/s voice and data channel

**C**

C/I Command/Indication

CCITT Comité Consultatif International Télégraph et Téléphone

CFI ConFIGurable Interface

CO Central Office

CODEC COder/DECoder

CPU Central Processing Limit

CRC Cyclic Redundancy Check

CV Code Violation

**D**

D-channel 16 kbit/s packetized (P) Data and control (S)

DCL Data Clock

DD Data Downstream

DMA Direct Memory Access

DS Data Strobe signal

DSP Digital Signal Processor

DTE Data Terminal equipment

DU Data Upstream

**E**

ELIC Extended Line Card Interface Controller

EOM End Of Transmission

EPIC Extended PCM Interface Controller

ET Exchange Termination

ETSI European Telecommunications Standardization Institute

**F**

FCS Frame Check Sequence

FIFO First In First Out

FSC Frame Synchronization Clock

**G**

GCI General Circuit Interface

**H**

HDLC High level Data Link Control procedure

HSCX High level Serial Communications controller eXtended

**I**

IBC ISDN Burst Controller

IDEC ISDN D channel Exchange Controller

ISDN Integrated Service Digital Network

IEC-Q ISDN Echo Cancellation circuit conforming to 2B/1Q line code

IEPC ISDN Exchange Power Controller

IOM ISDN Oriented Modular Interface

ISAC-S ISDN Subscriber Access Controller for S/T bus

ISO International Standard Organization

**L**

LAP-B Link Access Procedure on B-channel

LAP-D Link Access Procedure on D-channel

LT Line Termination

LT-S Line Termination on S-interface

LT-T Line Termination on T-interface

**M**

MD Monitor Data

MR Monitor Receive handshake signal

MUX Multiplexer

MX Monitor Transmit handshake signal

**N**

NT Network Termination

**P**

PBX Private Branch Exchange

PBC Peripheral Board Controller

PCM Pulse Code Modulation

PLL Phase Locked Loop

**Q**

QUAT-S QUAdruple Transceiver for SIT-interfaces



**R**

RES	Reset
RD/WR	Read/Write
RxD	Receive Data

**S**

SCC	Serial Communication Controller
S-interface	CCITT I.430, 4-wire int. for up to 8 ISDN terminals
S/T-interface	Subscriber/Trunk interface
SACCO	Special Application-Communication COntroller
SBCX	S/T Bus interface Circuit eXtended
SICOFI	Signal processing COdec/FIlter
SLIC	Subscriber Line Interface Controller

**T**

TE	Terminal Equipment
TSA	Time-Slot Assignment
T × D	Transmit Data

**U**

U-Interface	2-wire ping-pong connection
-------------	-----------------------------