T-49-19-16

NEC

NEC Electronics Inc.

μPD70320/322 (V25™) 16-BIT, SINGLE-CHIP CMOS MICROCOMPUTERS

PRELIMINARY INFORMATION

Description

The μPD70320 and μPD70322 (V25™) are high-performance, 16-bit, single-chip microcomputers with an 8-bit external data bus. They combine the instruction set of the μPD70108 (V20™) with many of the on-chip peripherals in NEC's 78000 series.

The μPD70320/322 processor has software compatibility with the V20 (and subsequently the 8086/8088), faster memory accessing, superior interrupt processing ability, and enhanced control of internal peripherals.

A variety of on-chip components, including 16K bytes of mask programmable ROM (μPD70322 only), 256 bytes of RAM, serial and parallel I/O, comparator port lines, timers, and a DMA controller make the μPD70320/322 a sophisticated microsystem.

Eight banks of registers are mapped into internal RAM below an additional 256-byte special function register (SFR) area that is used to control on-chip peripherals. Internal RAM and the SFR area are together relocatable to anywhere in the 1M-byte address space. This maintains compatibility with existing system memory maps.

The μPD70322 is the mask ROM version and the μPD70320 is the ROM-less version.

Features

- ☐ Complete single-chip microcomputer
  - — 16-bit ALU
  - — 16K bytes of ROM (μPD70322)
  - — 256 bytes of RAM
- ☐ Four-byte instruction prefetch queue
- ☐ 24 parallel I/O lines
- ☐ Eight analog comparator inputs with programmable threshold level
- ☐ Two independent DMA channels
- ☐ Two 16-bit timers
- ☐ Programmable time base counter
- ☐ Two full-duplex UARTs
- ☐ Programmable interrupt controller
  - — Eight priority levels
  - — Five external, 12 internal sources
  - — Register bank (eight) context switching
  - — Eight macro service function channels
- ☐ DRAM refresh pulse output
- ☐ Two standby modes
  - — HALT
  - — STOP
- ☐ Internal clock generator
  - — 5-MHz maximum frequency (0.4-μs instruction cycle time) (target specification: 8 MHz)
- ☐ Programmable wait state generation
- ☐ Separate address/data bus interface
- ☐ CMOS technology

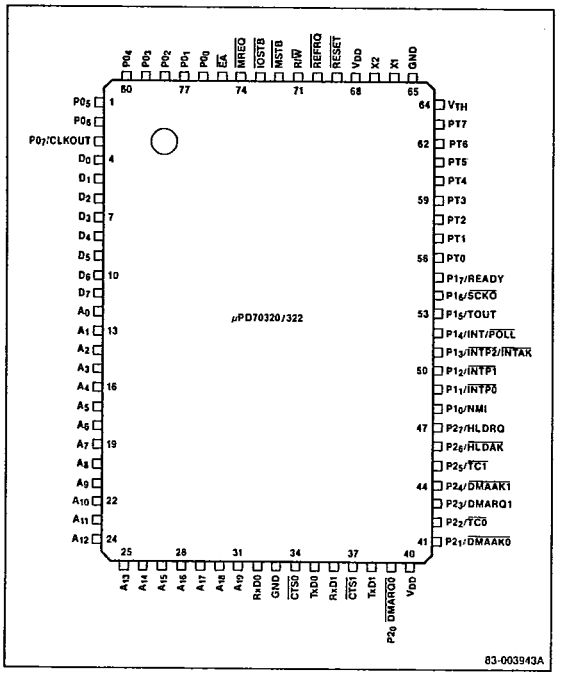
V20 and V25 are trademarks of NEC Corporation.

Ordering Information

Part Number	Package Type
μPD70320G-12	80-pin plastic miniflat
μPD70322G-12	80-pin plastic miniflat
μPD70320L	84-pin PLCC (plastic leadled chip carrier)
μPD70322L	84-pin PLCC (plastic leadled chip carrier)

Pin Configurations

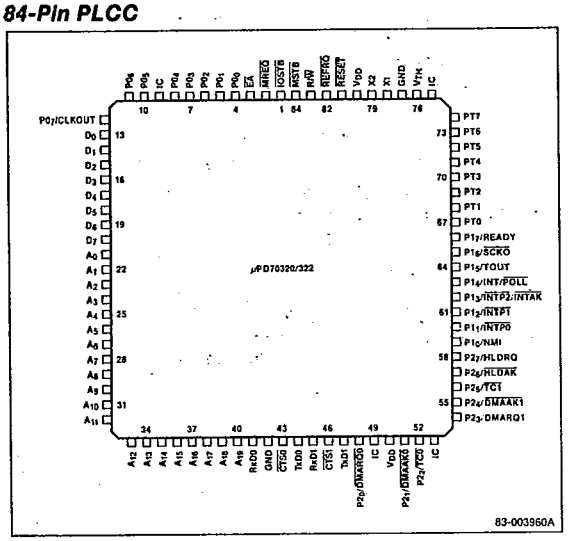
80-Pin Plastic Miniflat



5

Pin Configurations (cont)

84-Pin PLCC



Pin Identification

T-49-19-16

Symbol	Function
A_0-A_{19}	Address bus outputs
D_0-D_7	Bidirectional data bus
X1, X2	Crystal connection terminals
$\overline{RESET}$	Reset input
V_{DD}	Positive power supply voltage
V_{SS}	Ground
V_{TH}	Threshold voltage input
PT0-PT7	Comparator port input lines
$\overline{EA}$	External access
$\overline{MREQ}$	Memory request output
$P0_0$-$P0_7$	I/O port 0
CLKOUT	System clock output
NMI	Nonmaskable interrupt input
$P1_1$-$P1_2$/ $\overline{INTP0}$-$\overline{INTP1}$	Parallel input port lines/ External interrupt input lines
$P1_3$/$\overline{INTP2}$/$\overline{INTAK}$	Parallel input port line/ External interrupt input line/ Interrupt acknowledge output
$P1_4$/INT/$\overline{POLL}$	I/O port 1/Interrupt request input/ I/O poll input
$P1_5$/TOUT	I/O port 1 bit/Timer out
$P1_6$/$\overline{SCK0}$	I/O port 1 bit/Serial clock out
$P1_7$/READY	I/O port 1 bit/Ready input
$P2_0$/$\overline{DMARQ0}$	I/O port 2 bit/DMA request 0
$P2_1$/DMAAR0	I/O port 2 bit/DMA acknowledge 0
$P2_2$/$\overline{TC0}$	I/O port 2 bit/DMA terminal count 0
$P2_3$/DMARQ1	I/O port 2/DMA request 1
$P2_4$/$\overline{DMAAK1}$	I/O port 2/DMA acknowledge 1
$P2_5$/$\overline{TC1}$	I/O port 2/DMA terminal count 1
$P2_6$/$\overline{HLDAK}$	I/O port 2/Hold acknowledge output
$P2_7$/HLDRQ	I/O port 2/Hold request input
$\overline{IOSTB}$	I/O strobe output
$\overline{MSTB}$	Memory strobe output
R/$\overline{W}$	Read/Write output
$\overline{REFRQ}$	Refresh pulse output
RxD0	Serial receive data 0 input
$\overline{CTS0}$	Clear to send 0 input
TxD0	Serial transmit data 0 output
RxD1	Serial receive data 1 input
$\overline{CTS1}$	Clear to send 1 input
TxD1	Serial transmit data 1 output

Pin Functions

A_0-A_{19} [Address Bus]

A_0-A_{19} is the 20-bit address bus used to access all external devices.

D_0-D_7 [Data Bus]

D_0-D_7 is the 8-bit external data bus.

$\overline{\text{RESET}}$ [Reset]

A low on $\overline{\text{RESET}}$ resets the CPU and all on-chip peripherals. $\overline{\text{RESET}}$ can also release the standby modes. After $\overline{\text{RESET}}$ returns high, program execution begins from address FFFF0H.

X1, X2 [Crystal Connections]

The internal clock generator requires an external crystal across these terminals.

V_{DD} [Power Supply]

Two positive power supply pins (V_{DD}) reduce internal noise.

V_{SS} [Ground]

Two ground connections (V_{SS}) reduce internal noise.

V_{TH} [Threshold Voltage]

The comparator port uses this pin to determine the analog reference point. The actual threshold to each comparator line can be V_{TH} or V_{TH} x n/16, where n = 1 to 15.

$\overline{\text{EA}}$ [External Access]

If this pin is low on reset, the μPD70322 will execute program code from external memory instead of from internal ROM.

$\overline{\text{MREQ}}$ [Memory Request]

$\overline{\text{MREQ}}$ (active low) informs external memory that the current bus cycle is a memory access bus cycle.

PT0-PT7 [Comparator Port]

PT0-PT7 are inputs to the analog comparator port.

$P0_0$-$P0_7$ [Port 0]

$P0_0$-$P0_7$ are the lines of port 0, an 8-bit bidirectional parallel I/O port.

$P1_0$-$P1_7$ [Port 1]

$P1_1$-$P1_3$ are the input only lines of parallel port 1. $P1_0$ and $P1_4$-$P1_7$ are the remaining lines of parallel port 1, each line individually programmable as either an input or output.

$P2_0$-$P2_7$ [Port 2]

$P2_0$-$P2_7$ are the lines of port 2, an 8-bit bidirectional I/O port. The lines can also be used as control signals for the on-chip DMA controller.

CLKOUT [System Clock]

This is the internal system clock. It can be used to synchronize external devices to the CPU.

NMI [Nonmaskable Interrupt]

NMI cannot be masked through software and is typically used for emergency processing. Upon execution, the interrupt starting address is obtained from interrupt vector number 2. NMI can release the standby modes and can be programmed to be either rising or falling edge triggered.

$\overline{\text{INTP0}}$-$\overline{\text{INTP2}}$ [External Interrupt]

$\overline{\text{INTP0}}$-$\overline{\text{INTP2}}$ allow external devices to generate I/O requests (interrupts). Each can be programmed to be rising or falling edge triggered.

5

$\overline{\text{INTAK}}$ [Interrupt Acknowledge]

After INT is asserted, the CPU will respond with $\overline{\text{INTAK}}$ (active low) to inform external devices that the interrupt request has been granted.

INT [Interrupt Request]

INT is a maskable, active-low, vectored interrupt request input. After assertion, external hardware must provide the interrupt vector number.

$\overline{\text{POLL}}$ [Poll]

Upon execution of the $\overline{\text{POLL}}$ intruction, the CPU checks the status of this pin and, if low, program execution continues. If high, the CPU will check the level of the line every five clock cycles until it is low. $\overline{\text{POLL}}$ can be used to synchronize program execution to external conditions.

TOUT [Timer Out]

TOUT is the square-wave output signal from the internal timer.

SCKO, TxDn, CTSn, RxDn [Serial Clock Out, Serial Transmit Data, Clear to Send, Serial Receive Data]

The two on-chip serial ports use these lines for data transmission, receiving, and handshaking.

READY [Ready]

After READY is asserted (active low), the CPU will synchronize and insert at least two wait states into a read or write cycle to memory or I/O. This allows the processor to accommodate devices whose access times are longer than normal execution allows.

DMARQn, DMAAKn, TCn [DMA Request, DMA Acknowledge, Terminal Count]

These are the control signals to and from the on-chip DMA controller.

HLDRQ [Hold Request]

The HLDRQ input (active low) is used by external devices to request the CPU to release the system bus to an external bus master. The following lines go into a high-impedance state with internal 4.7-kΩ pull-up resistors: A_0-A_{19}, D_0-D_7, $\overline{MREQ}$, R/$\overline{W}$, and $\overline{MSTB}$.

HLDAK [Hold Acknowledge]

An $\overline{HLDAK}$ output (active low) informs external devices that the CPU has released the system bus.

IOSTB [I/O Strobe]

$\overline{IOSTB}$ is asserted during read and write operations to external I/O.

MSTB [Memory Strobe]

$\overline{MSTB}$ (active low) is asserted during read and write operations to external memory.

R/W [Read/Write]

An R/$\overline{W}$ output allows external hardware to determine if the current operation is a read or write cycle. It can also control the direction of bidirectional buffers.

REFRQ [Refresh]

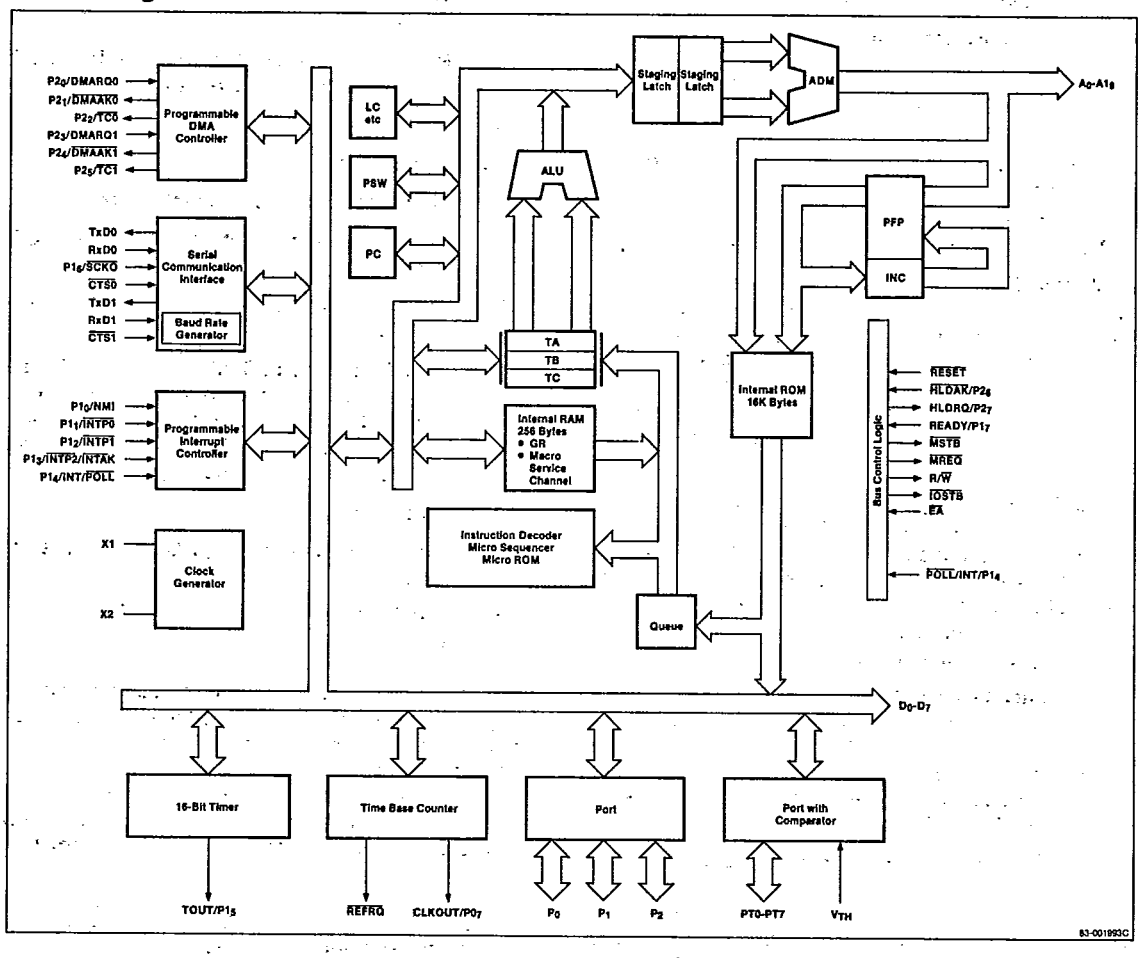
This active-low output pulse can refresh nonstatic RAM. It can be programmed to meet system specifications and is internally synchronized so that refresh cycles do not interfere with normal CPU operation.

NEC

μPD70320/322 (V25)

T-49-19-18

Block Diagram



5

Functional Description

Architectural Enhancements

The following features enable the μPD70320/322 to perform high-speed execution of instructions:

- Dual data bus
- 16-/32-bit temporary registers/shifters (TA, TB, TA + TB)
- 16-bit loop counter (LC)
- Program counter (PC) and prefetch pointer (PFP)

Dual Data Bus. The μPD70320/322 has two internal 16-bit data buses: the main data bus and a subdata bus. This reduces the processing time required for addition/subtraction, and logical comparison instructions by one third over single bus systems. The dual data bus method allows two operands to be fetched simultaneously from the general purpose registers and transferred to the ALU.

16-/32-Bit Temporary Registers/Shifters. The 16-bit temporary registers/shifters (TA, TB) allow high-speed execution of multiplication/ division and shift/rotation instructions. By using the temporary registers/shifters, the μPD70320/322 can execute multiplication/division instructions about four times faster than with the microprogramming method.

Loop Counter [LC]. The dedicated hardware loop counter counts the number of loops for string operations and the number of shifts performed for multiple bit shift/rotation instructions. The loop counter works with internal dedicated shifters to speed the processing of multiplication/division instructions.

Program Counter and Prefetch Pointer [PC and PFP]. The hardware PC addresses the memory location of the instruction to be executed next. The hardware PFP addresses the program memory location to be accessed next. Several clocks are saved for branch, call, return, and break instructions compared with processors having only one instruction pointer.

Register Set

Figure 1 shows the μPD70320/322 has eight banks of registers functionally mapped into internal RAM. Each bank contains general purpose registers, pointer and index registers, segment registers, and save areas.

General Purpose Registers [AW, BW, CW, DW]. There are four 16-bit general purpose registers that can each serve as individual 16-bit registers or two independent 8-bit registers (AH, AL, BH, BL, CH, CL, DH, DL). The following instructions use the general purpose registers for default:

AW	Word multiplication/division, word I/O, data conversion
AL	Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation
AH	Byte multiplication/division
BW	Translation
CW	Loop control branch, repeat prefix
CL	Shift instructions, rotation instructions, BCD operations
DW	Word multiplication/division, indirect addressing I/O

Pointers [SP, BP] and Index Registers [IX, IY]. These registers are used as 16-bit base pointers or index registers in based addressing, indexed addressing, and based indexed addressing. The registers are used as default registers under the following conditions:

SP	Stack operations
IX	Block transfer (source), BCD string operations
IY	Block transfer (destination), BCD string operations

Segment Registers. The segment registers divide the 1M-byte address space into 64K-byte blocks. Each segment register functions as a base address to a block; the effective address is an offset from that base. Physical addresses are generated by shifting the associated segment register left four binary digits and then adding the effective address. The segment registers are:

Segment Register	Default Offset
PS (Program segment)	PC
SS (Stack segment)	SP, Effective address
DS0 (Data segment-0)	IX, Effective address
DS1 (Data segment-1)	IY, Effective address

Save Registers. SAVE PC and SAVE PSW are used as save areas during register bank context switching. The VECTOR PC save location contains the effective address of the interrupt service routine when register bank switching is used to service interrupts.

Program Counter [PC]. The PC is a 16-bit binary counter that contains the offset address from the program segment of the next instruction to be executed. It is incremented every time an instruction is received from the queue. It is loaded with a new location whenever a branch, call, return, break, or interrupt is executed.

Processor Status Word [PSW]. The PSW contains the following status and control flags.

15			PSW				8
1	RB2	RB1	RB0	V	DIR	IE	BRK

7							0
S	Z	F1	AC	F0	P	BRKI	CY

Status Flags

V	Overflow bit
S	Sign
Z	Zero
AC	Auxiliary carry
P	Parity
CY	Carry

Control Flags

DIR	Direction of string processing
IE	Interrupt enable
BRK	Break (after every instruction)
RBn	Register bank select
BRKI	I/O trap enable (see software interrupts)
F0, F1	General-purpose user flags (accessed through the flag special function register)

Memory Map

The μPD70320/322 has a 20-bit address bus that can directly access 1M bytes of memory. Figure 2 shows that the 16K bytes of internal ROM (μPD70322 only) are located at the top of the address space from FC000H to FFFFFH.

Internal Data Area. Figure 2 shows the internal data area (IDA) is a 256-byte internal RAM area followed consecutively by a 256-byte special function register (SFR) area. All the data and control registers for on-chip peripherals and I/O are mapped into the SFR area and accessed as RAM. The IDA is dynamically relocatable in 4K-byte increments by changing the value in the internal data base (IDB) register. Whatever value is in this register will be assigned as the uppermost eight bits of the IDA address.

On reset, the internal data base register is set to FFH which maps the IDA into the internal ROM space. However, since the μPD70322 has a separate bus to internal ROM, this does not present a problem. When these address spaces overlap, program code cannot be executed from the IDA and internal ROM locations cannot be accessed as data. You can select any of the eight possible register banks which occupy the entire internal RAM space. Multiple register bank selection allows faster interrupt processing and facilitates multitasking.

In larger-scale systems where internal RAM is not required for data memory, the internal RAM can be removed completely from the address space and dedicated entirely to registers and control functions such as macro service and DMA channels. Clearing the RAMEN bit in the processor control register achieves this. When the RAMEN bit is cleared, internal RAM can only be accessed by register addressing or internal control processes.

5

Figure 1. Register Banks in Internal RAM

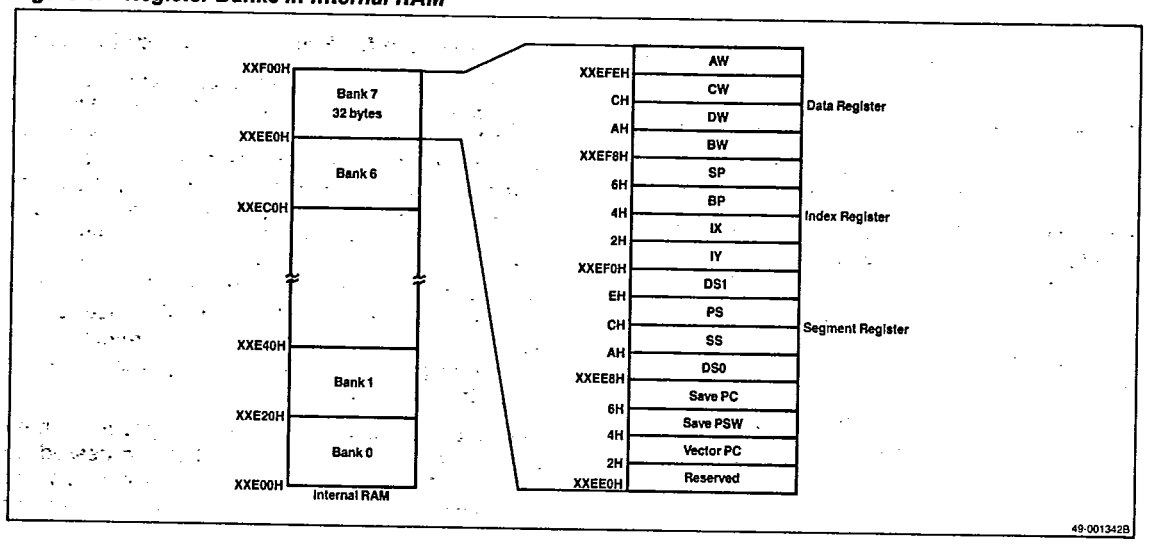
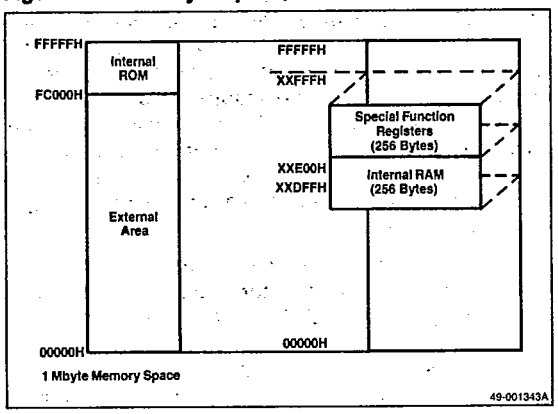


Figure 2. Memory Map



Instruction Set

The μPD70320/322 instruction set is fully compatible with the V20 native mode instruction set. The V20 instruction set is a superset of the μPD8086/8088 instruction set with different execution times and mnemonics.

The μPD70320/322 does not support the V20 8080 emulation mode. All of the instructions pertaining to this have been deleted from the μPD70320/322 instruction set.

Enhanced Instructions

In addition to the μPD8086/88 instructions, the μPD70320/322 has the following enhanced instructions.

Instruction	Function
PUSH imm	Pushes immediate data onto stack
PUSH R	Pushes eight general registers onto stack
POP R	Pops eight general registers from stack
MUL imm	Executes 16-bit multiply of register or memory contents by immediate data
SHL imm8 SHR imm8 SHRA imm8 ROL imm8 ROR imm8 ROLC imm8 RORC imm8	Shifts/rotates register or memory by immediate value
CHKIND	Checks array index against designated boundaries
INM	Moves a string from an I/O port to memory
OUTM	Moves a string from memory to an I/O port
PREPARE	Allocates an area for a stack frame and copies previous frame pointers
DISPOSE	Frees the current stack frame on a procedure exit

Unique Instructions

The μPD70320/322 has the following unique instructions.

Instruction	Function
INS	Inserts bit field
EXT	Extracts bit field
ADD4S	Performs packed BCD string addition
SUB4S	Performs packed BCD string subtraction
CMP4S	Performs packed BCD string comparison
ROL4	Rotates BCD digit left
ROR4	Rotates BCD digit right
TEST1	Tests bit
SET1	Sets bit
CLR1	Clears bit
NOT1	Complements bit
BTCLR	Tests bit; if true, clear and branch
REPC	Repeat while carry set
REPNC	Repeat while carry cleared

Variable Length Bit Field Operation Instructions

Bit fields are a variable length data structure that can range in length from 1 to 16 bits. The μPD70320/322 supports two separate operations on bit fields: insertion (INS) and extraction (EXT). There are no restrictions on the position of the bit field in memory. Separate segment, byte offset, and bit offset registers are used for insertion and extraction. Following the execution of these instructions, both the byte offset and bit offset are left pointing to the start of the next bit field, ready for the next operation. Bit field operation instructions are powerful and flexible and are therefore highly effective for graphics, high level languages, and packing/unpacking applications.

Insert bit field copies the bit field of specified length from the AW register to the bit field addressed by DS1:IY:reg8 (8-bit general purpose register). The bit field length can be located in any byte register or

supplied as immediate data. Following execution, both the IY and reg8 are updated to point to the start of the next bit field.

Bit field extraction copies the bit field of specified length from the bit field addressed by DS0:IX:reg8 to the AW register. If the length of the bit field is less than 16 bits, the bit field is right justified with a zero fill. The bit field length can be located in any byte register or supplied as immediate data. Following execution, both IX and reg8 are updated to point to the start of the next bit field.

Figures 3 and 4 show bit field insertion and bit field extraction.

Packed BCD

Packed BCD instructions process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte format operands (ROR4, ROL4). Packed BCD strings may be one to 254 digits in length. The two BCD rotation instructions perform rotation of a single BCD digit in the lower half of the AL register through the register or the memory operand.

Bit Manipulation Instructions

The μPD70320/322 has five unique bit manipulation instructions. The ability to test, set, clear, or complement a single bit in a register or memory operand increases code readability as well as performance over the logical operations traditionally used to manipulate bit data. This feature further enhances control over on-chip peripherals.

Repeat Prefixes

Two new repeat prefixes (REPC, REPNC) allow conditional block transfer instructions to use the state of the CY flag as the termination condition. This allows inequalities to be used when working on ordered data, thus increasing performance when searching and sorting algorithms.

Figure 3. Bit Field Insertion

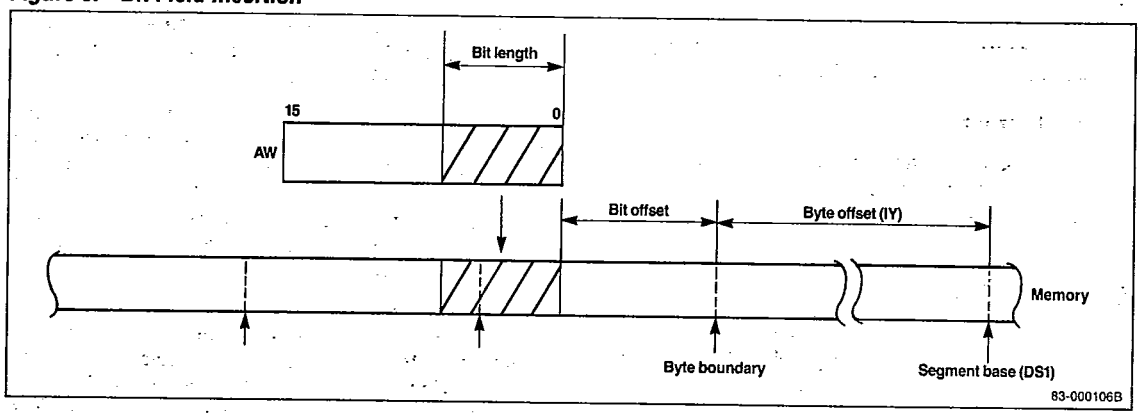
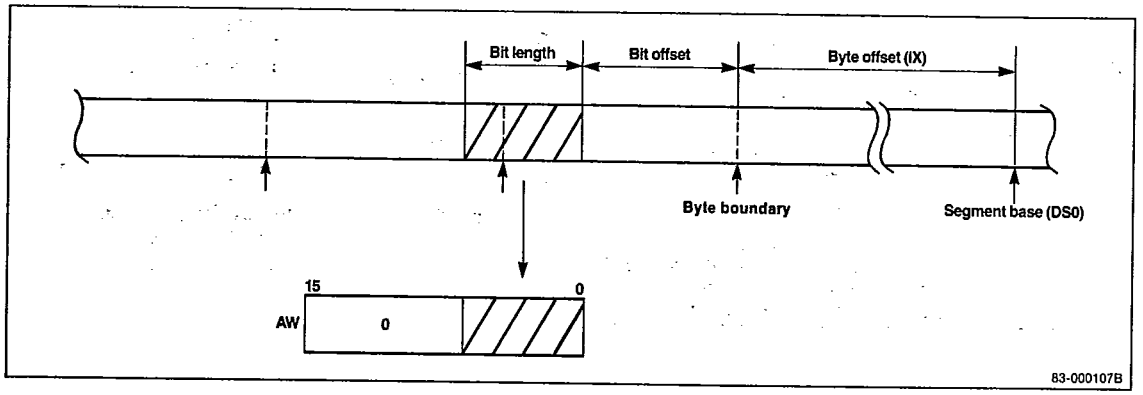


Figure 4. Bit Field Extraction



Besides the V20 instruction set, the μPD70320/322 has the four additional instructions described in table 1.

Table 1. Additional Instructions

Instruction	Function
BTCLR var,imm3, short label	Bit test and if true, clear and branch; otherwise, no operation
STOP (no operand)	Power down instruction, stops oscillator
RETRBI (no operand)	Return from register bank context switch interrupt
FINT (no operand)	Finished interrupt. After completion of a hardware interrupt or I/O request, this instruction must be used to reset the current priority bit in the in-service priority register (ISPR).

The ISPR is an 8-bit register; each of its bits, PR_0-PR_7, correspond to each of the eight possible I/O request priorities, respectively. The ISPR keeps track of the priority of the interrupt currently being serviced by setting the appropriate bit. The ISPR format is shown below.

PR_7	PR_6	PR_5	PR_4	PR_3	PR_2	PR_1	PR_0

Interrupt Structure

The μPD70320/322 can service interrupts generated through hardware and software. Table 2 shows the various software interrupts.

Table 2. Software Interrupts

Interrupt	Description
Divide error	The CPU will trap if a divide error occurs as the result of a DIV or DIVU instruction.
Single step	The interrupt is generated after every instruction if the BRK bit in the PSW is set.
Overflow	By using the BRKV instruction, an interrupt can be generated as the result of an overflow.
Interrupt instructions	The BRK 3 and BRK imm8 instructions can generate interrupts.
Array bounds	The CHKIND instruction will generate an interrupt if specified array bounds have been exceeded.
Escape trap	The CPU will trap on an FP01,2 instruction to allow software to emulate the floating point processor.
I/O trap	If the I/O trap bit in the PSW is set, a trap will be generated on every IN or OUT instruction. Software can then provide an updated peripheral address. This feature allows software interchangeability between different systems.

When executing software written for another system, it is better to implement I/O with on-chip peripherals to reduce external hardware requirements. However, since μPD70320/322 internal peripherals are memory mapped, software conversion could be difficult. The I/O trap feature allows easy conversion from external peripherals to on-chip peripherals.

Interrupt Vector Table. Table 3 shows the starting addresses of interrupt processing routines. The table begins at physical address 0H, which is outside the internal ROM space. Therefore, if utilizing an interrupt processing routine within the interrupt vector table, external memory will be required. By servicing interrupts via the macro service function or context switching, you can avoid the addition of external memory.

Each interrupt vector is four bytes. Upon execution of a vectored interrupt, the lower addressed word is transferred to the PC, and the upper word to the PS. However, the byte order within each word is reversed so that the low-order bytes of the vector address become the most significant bytes in the PC and PS.

Hardware Interrupt Configuration. There are two types of hardware interrupt requests: standard vectored interrupts and I/O requests.

After a vectored interrupt, the PC and PSW are saved on the stack and the program transfers to the location indicated by the interrupt vector contents. When an interrupt is triggered by NMI, the CPU automatically traps to vector number two. When an interrupt is triggered by INTR, external devices must provide the interrupt vector number.

I/O requests are a group of interrupts, generated externally or from on-chip peripherals. The internal interrupt controller controls I/O requests. I/O requests can be serviced (by the macro service function) without transferring program control to an interrupt routine. The following are the 14 possible I/O requests.

Group	Source
External interrupt request	INTP0, INTP1, INTP2
DMA controller	INTD0, INTD1
Timer	INTTU0, INTTU1, INTTU2
Serial interface	INTSER0, INTSR0, INTST0, INTSER1, INTSR1, INTST1

Table 3. Interrupt Vectors

Address (Hex)	Vector No.	Assigned Use
00	0	Divide error 04 1 Break flag
04	1	Break flag
08	2	NMI
0C	3	BRK3 instruction
10	4	BRKV instruction
14	5	CHKIND instruction
18	6	General purpose
1C	7	Escape trap
20	8	General purpose
24-3C	9-15	Reserved
40-4C	15-19	General purpose
50	20	I/O trap
54-5C	21-23	General purpose
60	24	Reserved
64-6C	25-27	General purpose
70	28	INTSER0
74	29	INTSR0
78	30	INTST0
7C	31	General purpose
80	32	INTSER1
84	33	INTSR1
88	34	INTST1
8C	35	General purpose
90	36	INTD0
94	37	INTD1
98-9C	38,39	General purpose
A0	40	$\overline{\text{INTP0}}$
A4	41	$\overline{\text{INTP1}}$
A8	42	$\overline{\text{INTP2}}$
AC	43	General purpose
B0	44	INTTU0
B4	45	INTTU1
B8	46	INTTU2
BC	47	INTTB
0C0-3FF	48-255	General purpose

Arbitration of I/O requests is resolved internally by the interrupt controller. The priority of each I/O request is individually programmable from 0 to 7 (0 is the highest priority). You can process these interrupts in one of three modes: standard vectored interrupt, register bank context switching, or macro service function. When standard vectored interrupt mode is selected, I/O requests are serviced as previously described vectored interrupts. The CPU automatically traps to the vector location shown in the interrupt vector table.

Register bank context switching allows I/O requests to be processed rapidly by switching register banks. After an interrupt, the new register bank selected is that which has the same register bank number (0-7) as the priority of the interrupt to be serviced. The PC and PSW are automatically stored in the save areas of the new register bank and the address of the interrupt routine is loaded from the vector PC storage location in the new register bank. After interrupt processing, execution of the RETRBI (return from register bank interrupt) returns control to the former register bank and restores the former PC and PSW. Figures 5 and 6 show register bank context switching and register bank return.

The macro service function (MSF) acts as an internal DMA controller between on-chip peripherals (special function registers) and memory. The MSF greatly reduces the software overhead and CPU time that other processors would require for register save processing, register returns, and other handling associated with interrupt processing.

If the MSF is selected for a particular I/O request, each time the request is received, a byte or word of data will be transferred between the SFR and memory without interrupting the CPU. Each time a request occurs, the macro service counter is decremented. When the counter reaches zero, an interrupt is generated. The MSF also has a character search option. When selected, every byte transferred will be compared to an 8-bit search character and an interrupt will be generated if a match occurs or if the macro service counter counts out.

There are eight eight-byte macro service channels mapped into internal RAM from XXE00H to XXE3FH. Figure 7 shows the components of each channel.

Figure 5. Register Bank Context Switching

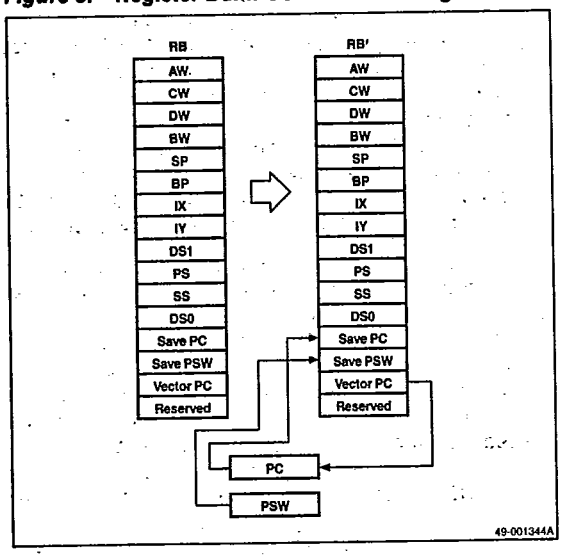


Figure 6. Register Bank Return

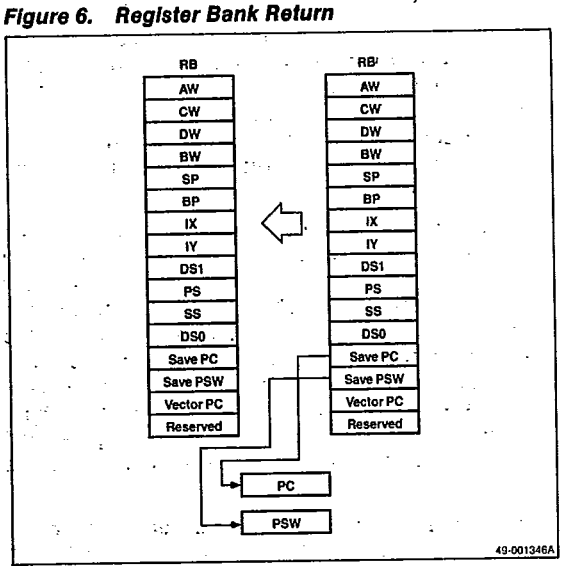
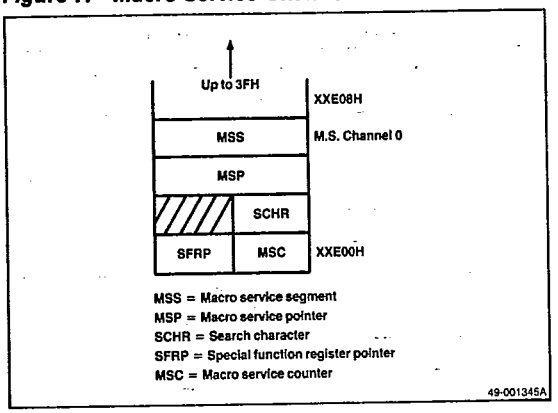


Figure 7. Macro Service Channels



On-Chip Peripherals

Timer Unit

The μPD70320/322 (figure 8) has two programmable 16-bit interval timers (TM0,TM1) with variable input clock frequencies on-chip. Each of the two 16-bit timer registers has an associated 16-bit modulus register (MD0, MD1). The timer operates in interval timer mode or one-shot mode.

Interval Timer Mode. In this mode, TM0/TM1 are decremented by the selected input clock and, after counting out, the registers are automatically reloaded from the modulus registers and counting continues. Each time TM1 counts out, I/O requests are generated through TF1 and TF2 (Timer Flags 1,2). When TM0 counts out, an I/O request is generated through TF0. The timer out signal can be used as a square wave output whose half-cycle is equal to the count time. There are two selectable input clocks (SCLK: system clock = fosc/2, fosc = 10 MHz).

Clock	Timer Resolution	Full Count
SCLK/6	1.2 μs	78.643 ms
SCLK/128	25.6 μs	1.678 s

One-Shot Mode. In the one-shot mode, TM0 and MD0 operate as independent one-shot timers. Starting with a preset value, each is decremented to zero. At zero, counting ceases and an I/O request is generated by TF0 or TF1. One-shot mode allows two selectable input clocks (fosc = 10 MHz).

Clock	Timer Resolution	Full Count
SCLK/12	2.4 μs	157.283 ms
SCLK/128	25.6 μs	1.678 s

Time Base Counter

The μPD70320/70322 has a free-running long base counter that can be used to generate periodic interrupts at lengthy intervals. The counter has three selectable input clocks: SCLK, SCLK/2, and SCLK/4. You can select one of the following four taps (outputs) from the counter as an interrupt source: i/1024, i/8192, i/64K, or i/1M ("i" is the selected input clock).

The TBC interrupt is unlike the other on-chip peripheral I/O requests in that it is preset as a level seven vectored interrupt. Macro service and register bank switching cannot be used to service this interrupt. Figure 9 is the time base counter block diagram.

Refresh Controller

The μPD70320/322 has an on-chip refresh controller for dynamic and pseudostatic RAM mass storage memories. The refresh controller generates refresh addresses and refresh pulses. It inserts refresh cycles between the normal CPU bus cycles according to refresh specifications.

The refresh controller outputs a 9-bit refresh address on address bits A_0-A_8 during the refresh bus cycle. Address bits A_8-A_{19} are all 1's. The 9-bit refresh address is automatically incremented at every refresh timing for 512 row addresses. The 8-bit refresh mode (RFM) register specifies the refresh operation and allows refresh during both CPU HALT and HOLD modes. Refresh cycles are automatically timed to $\overline{\text{REFRQ}}$ following read/write cycles to minimize the effect on system throughput.

The following shows the $\overline{\text{REFRQ}}$ pin level in relation to bits 4 (RFEN) and 7 (RELV) of the refresh mode register.

RFEN	RELV	$\overline{\text{REFRQ}}$ Level
0	0	0
0	1	1
1	0	0
1	1	Refresh pulse output

Figure 9. Time Base Counter Block Diagram

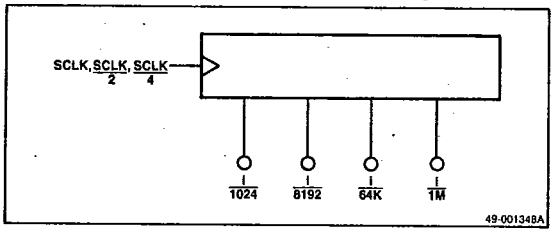
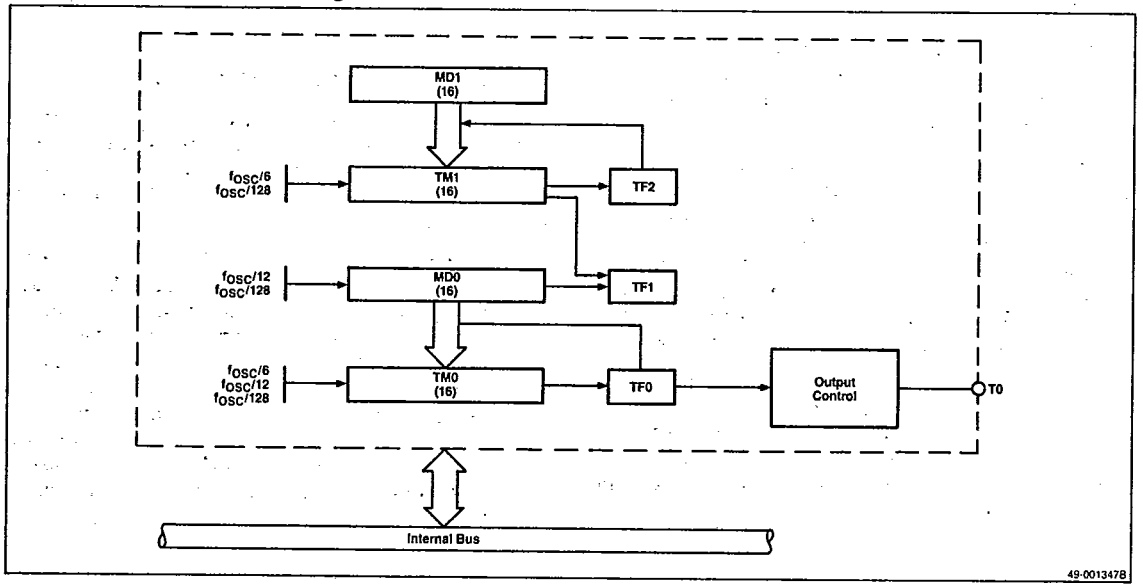


Figure 8. Timer Unit Block Diagram



Serial Interface

The μPD70320/322 has two full-duplex UARTs, channel 0 and channel 1. Each serial port channel has a transmit line (TxDn), a receive line (RxDn), and a clear to send (CTSn) input line for handshaking. Communication is synchronized by a start bit, and you can program the ports for even, odd, or no parity, character lengths of seven or eight bits, and one or two stop bits.

The μPD70320/322 has dedicated baud rate generators for each serial channel. This eliminates the need to obligate the on-chip timers. The baud rate generator allows a wide range of data transfer rates (up to 1 Mbps). This includes all of the standard baud rates without being restricted by the value of the particular external crystal. Each baud rate generator has an 8-bit baud rate generator (BRGn) data register which functions as a prescaler to a programmable input clock selected by the serial communication control (SCCn) register. Together these must be set to generate a frequency that is equivalent to the desired baud rate.

In addition to the asynchronous mode, channel 0 has a synchronous I/O interface mode. In this mode, each bit of data transferred is synchronized to a serial clock ($\overline{SCKO}$). This is the same as the NEC μCOM75 and μCOM87 series, and allows easy interfacing to these devices. Figure 10 shows the serial interface block diagram.

DMA Controller

The μPD70320/322 has a two-channel, on-chip DMA controller. This allows rapid data transfer between memory and auxiliary storage devices. The DMA controller supports four modes of operation, two for memory-to-memory transfers and two for transfers between I/O and memory.

Memory-to-Memory Transfers. In single-step mode, the falling edge of $\overline{DMARQ}$ causes DMA transfer cycles and CPU bus cycles to alternate as long as $\overline{DMARQ}$ is low or until the prescribed number of DMA transfers has occurred. Interrupts can be accepted while in this mode. In burst mode, DMA transfer cycles continue until the DMA terminal counter decrements to zero. Software can also initiate memory-to-memory transfers.

Transfers Between I/O and Memory. In single-transfer mode, one DMA transfer occurs after each falling edge of $\overline{DMARQ}$. After the transfer, the bus is returned to the CPU. In demand release mode, the falling edge of $\overline{DMARQ}$ enables DMA cycles, which continue as long as $\overline{DMARQ}$ is low.

In all modes, the $\overline{TC}$ (terminal count) output pin will pulse low and a DMA completion I/O request will be generated after the predetermined number of DMA cycles has been completed. Figure 11 shows the DMA channel area in memory.

The bottom of internal RAM contains all of the necessary address information for the designated DMA channels. The DMA channel mnemonics are as follows:

TC	Terminal counter
SAR	Source address register
SARH	Source address register high
DAR	Destination address register
DARH	Destination address register high

The DMA controller generates physical source addresses by offsetting SARH 12 bits to the left and then adding the SAR. The same procedure is also used to generate physical destination addresses. You can program the controller to increment or decrement source and/or destination addresses independently during DMA transfers.

Parallel Ports

The μPD70320/322 has three 8-bit parallel I/O ports: P0, P1, and P2. SFR locations can access these ports. The port lines are individually programmable as inputs or outputs. Many of the port lines have dual functions as port or control lines.

The analog comparator port (PT) compares each input line to a reference voltage. The reference voltage is programmable to be the V_{REF} input or V_{REF} x n/16, where n = 1 to 15.

Programmable Wait State Generation

You can generate wait states internally to further reduce the necessity for external hardware. Insertion of these wait states allows direct interface to devices whose access times cannot meet the CPU read/write timing requirements.

When using this function, the entire 1M-byte memory address space is divided into 128K-blocks. Each block, with the exception of the uppermost block, can be programmed for zero, one, or two wait states, or for external control (READY signal). The appropriate bits in the wait control word (WTC) control wait state generation. Programming the bits corresponding to the top 128K-byte block of memory, will actually set the wait state conditions for the entire I/O address space. Figure 12 shows the memory map for programmable wait state generation.

Figure 10. Serial Interface Block Diagram

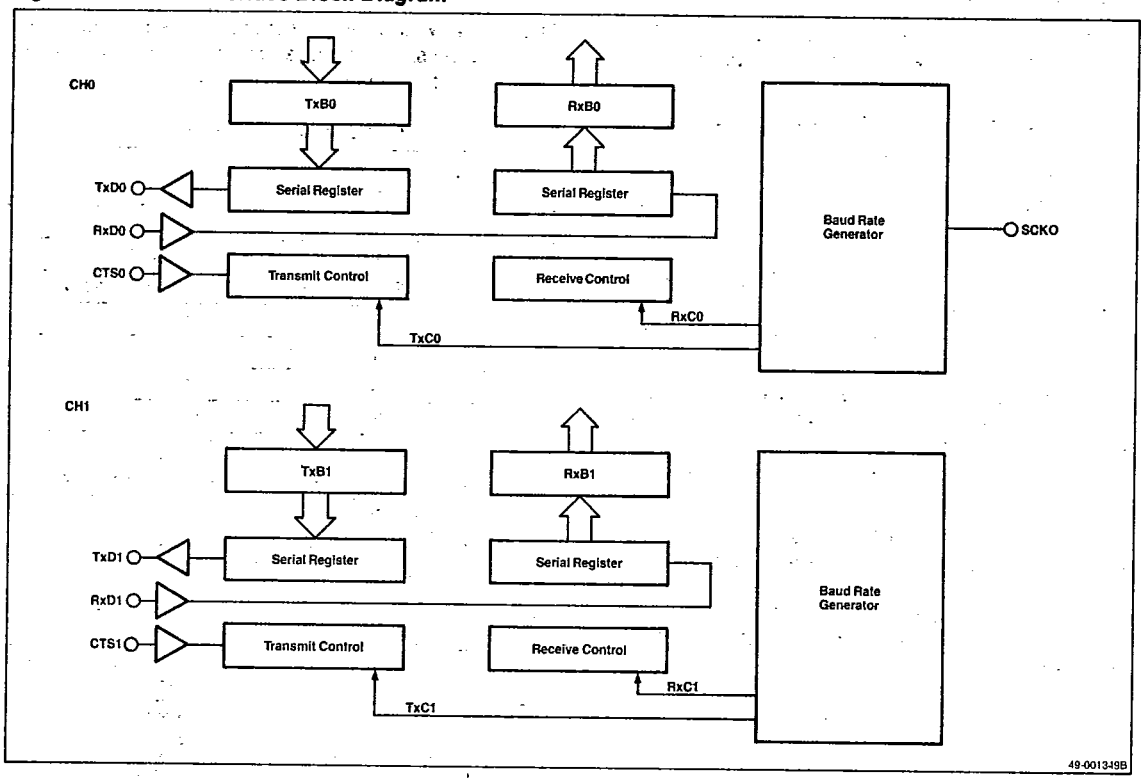


Figure 11. DMA Channels

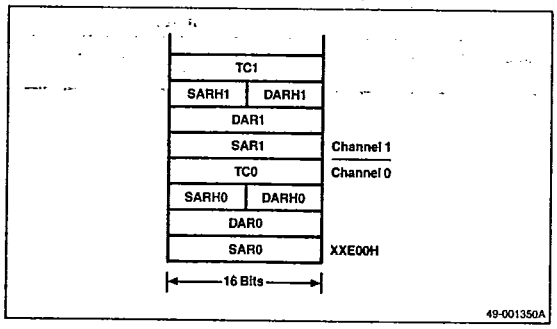
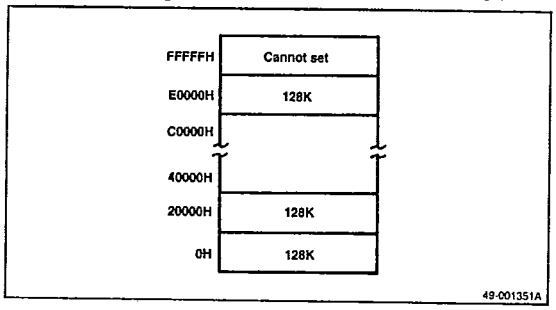


Figure 12. Programmable Wait State Generation



5

Low-Power Standby

There are two low-power standby modes: HALT and STOP. Software causes the processor to enter either mode.

HALT Mode. In the HALT mode, the processor is inactive and the chip consumes much less power than when operational. The external oscillator remains functional and all peripherals are active. Internal status and output port line conditions are maintained. Any unmasked interrupt or I/O request can release this mode. In the EI state, I/O requests subsequently will be processed as vectored interrupts. In the DI state, program execution is restarted with the instruction following the HALT instruction.

STOP Mode. The STOP mode allows the largest power reduction while maintaining RAM. The oscillator is stopped, halting all internal peripherals. All internal status is maintained. Only a reset or NMI can release this mode.

A standby flag in the SFR area is set by rises in the supply voltage. The flag is reset when its status is read. Its status is maintained during normal operation and standby. Use the standby flag to determine whether program execution is returning from standby or from a cold start.

Special Function Registers

Table 4 shows the special function register mnemonic, type, address, reset value, and function. Figures 13 through 32 show the register formats.

Table 4. Special Function Registers

Name	Byte/ Word	Address	Reset Value	Function
P0	B	xxF00H		Port 0
PM0	B	xxF01H	FFH	Port mode 0
PMC0	B	xxF02H	00H	Port mode control 0
P1	B	xxF08H		Port 1
PM1	B	xxF09H	FFH	Port mode 1
PMC1	B	xxF0AH	00H	Port mode control 1
P2	B	xxF10H		Port 2
PM2	B	xxF11H	FFH	Port mode 2
PMC2	B	xxF12H	00H	Port mode control 2
PT	B	xxF38H		Port T
PMT	B	xxF3BH	00H	Port mode T
INTM	B	xxF40H	00H	Interrupt mode
EMS0	B	xxF44H		External interrupt macro service 0
EMS1	B	xxF45H		External interrupt macro service 1
EMS2	B	xxF46H		External interrupt macro service 2
EXIC0	B	xxF4CH	47H	External I/O request control 0
EXIC1	B	xxF4DH	47H	External I/O request control 1
EXIC2	B	xxF4EH	47H	External I/O request control 2

Table 4. Special Function Registers (cont)

Name	Byte/ Word	Address	Reset Value	Function
RXB0	B	xxF60H		Receive buffer 0
TXB0	B	xxF62H		Transfer buffer 0
SRMS0	B	xxF65H		Serial receive macro service 0
STMS1	B	xxF66H		Serial transmit macro service 1
SCM0	B	xxF68H	00H	Serial communication mode 0
SCC0	B	xxF69H	00H	Serial communication control 0
BRG0	B	xxF6AH	00H	Baud rate generator 0
SCE0	B	xxF6BH	00H	Serial communication error 0
SEIC0	B	xxF6CH	47H	Serial error I/O request control 0
SRIC0	B	xxF6DH	47H	Serial receive I/O request control 0
STIC0	B	xxF6EH	47H	Serial transmit I/O request control 0
RXB1	B	xxF70H		Receive buffer 1
TXB1	B	xxF72H		Transmit buffer 1
SRMS1	B	xxF75H		Serial receive macro service 1
STMS1	B	xxF76H		Serial transmit macro service 1
SCM1	B	xxF78H	00H	Serial communication mode 1
SCC1	B	xxF79H	00H	Serial communication control 1
BRG1	B	xxF7AH	00H	Baud rate generator register 1
SCE1	B	xxF7BH	00H	Serial communication error 0
SEIC1	B	xxF7CH	47H	Serial error I/O request control 1
SRIC1	B	xxF7DH	47H	Serial receive I/O request control 1
STIC1	B	xxF7EH	47H	Serial transmit I/O request control 1
TM0	W	xxF80H		Timer register 0
TM0L	B	xxF80H		Timer register 0 low
TM0H	B	xxF81H		Timer register 0 high
MD0	W	xxF82H		Modulo register 0
MD0L	B	xxF82H		Modulo register 0 low
MD0H	B	xxF83H		Modulo register 0 high
TM1	W	xxF88H		Timer register 1
TM1L	B	xxF88H		Timer register 1 low
TM1H	B	xxF89H		Timer register 1 high
MD1	W	xxF8AH		Modulo register 1
MD1L	B	xxF8AH		Modulo register 1 low
MD1H	B	xxF8BH		Modulo register 1 high
TMC0	B	xxF90H	00H	Timer control 0
TMC1	B	xxF91H	00H	Timer control 1
TMMS0	B	xxF94H		Timer macro service 0
TMMS1	B	xxF95H		Timer macro service 1
TMMS2	B	xxF96H		Timer macro service 2
TMIC0	B	xxF9CH	47H	Timer I/O request control 0
TMIC1	B	xxF9DH	47H	Timer I/O request control 1
TMIC2	B	xxF9EH	47H	Timer I/O request control 2
DMAC0	B	xxFA0H		DMA control 0
DMAM0	B	xxFA1H	00H	DMA mode 0
DMAC1	B	xxFA2H		DMA control 1
DMAM1	B	xxFA3H	00H	DMA mode 1
DIC0	B	xxFACH	47H	DMA I/O request control 0
DIC1	B	xxFADH	47H	DMA I/O request control 1
RFM	B	xxFE1H	10H	Refresh mode
TBIC	B	xxFECH	47H	Time base I/O request control
WTC	W	xxFE8H	FFH	Wait control
WTCL	B	xxFE8H	FFH	Wait control low
WTCH	B	xxFE9H	FFH	Wait control high
PSWL	B	xxFEAH	00H	Flag register
PRC	B	xxFEBH	4EH	Processor control
SB	B	xxFE0H		Standby control
IDB	B	FFFFFH	FFH	Internal data area base

5

Figure 13. Port Mode Registers 0, 1, and 2.

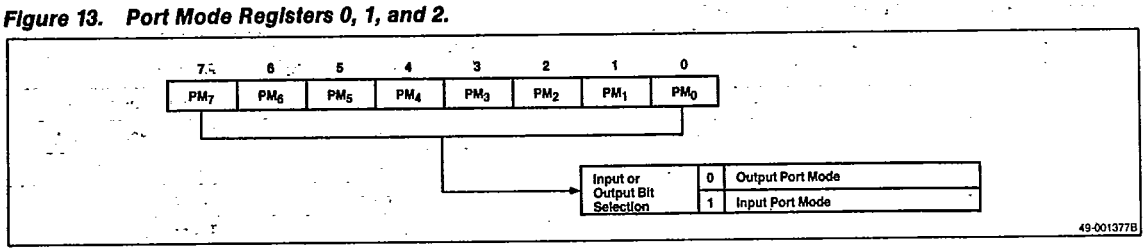


Figure 14. Port Mode Control 0 Register

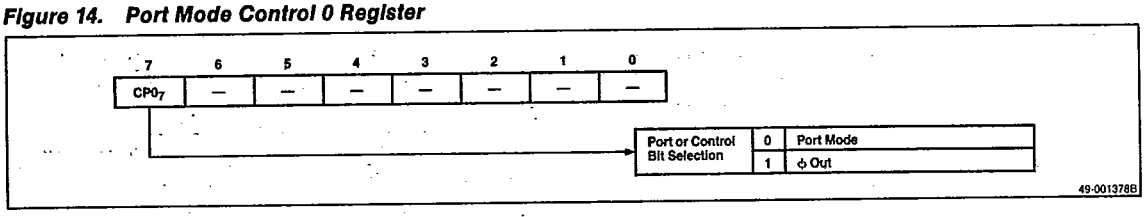
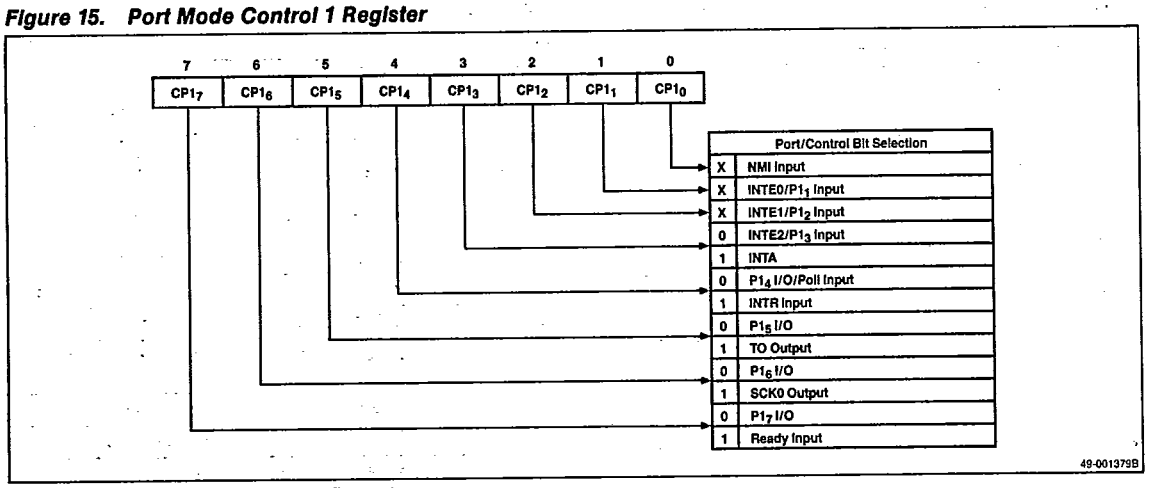


Figure 15. Port Mode Control 1 Register



T-49-19-16

Figure 16. Port Mode Control 2 Register

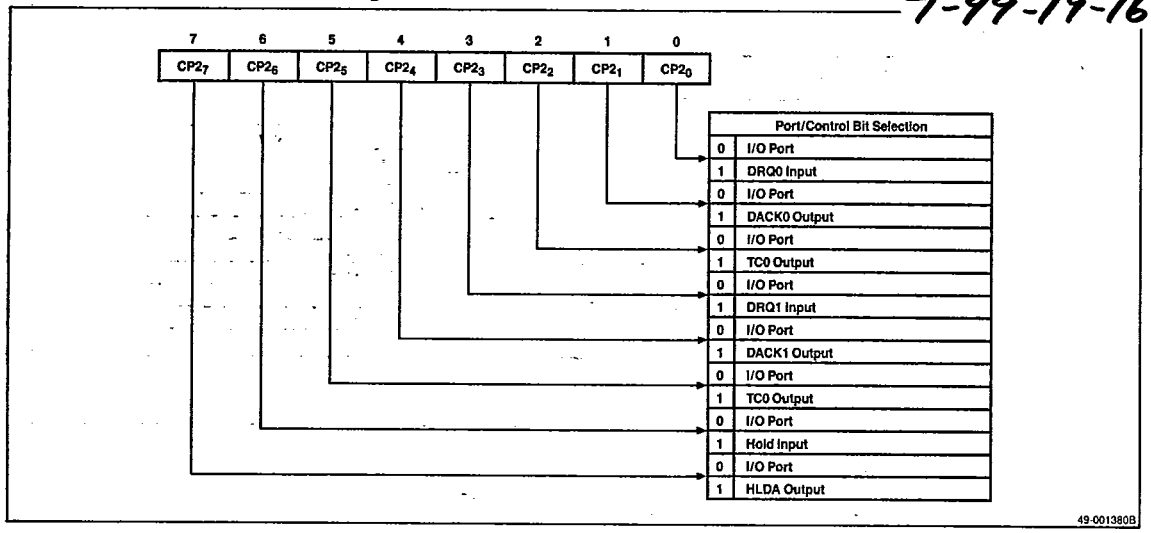
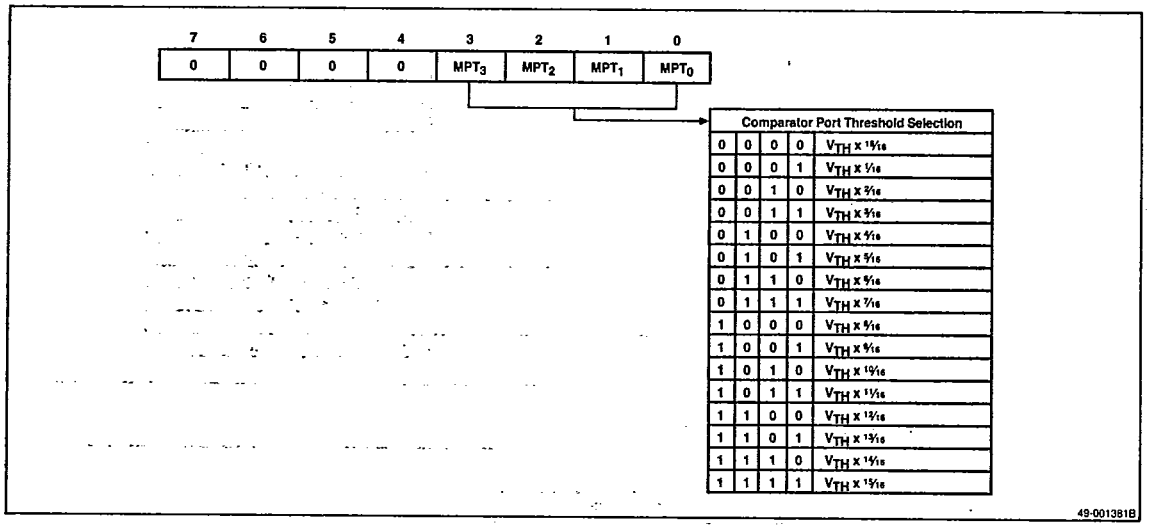


Figure 17. Port Mode T Register



5

Figure 18. Interrupt Mode Register

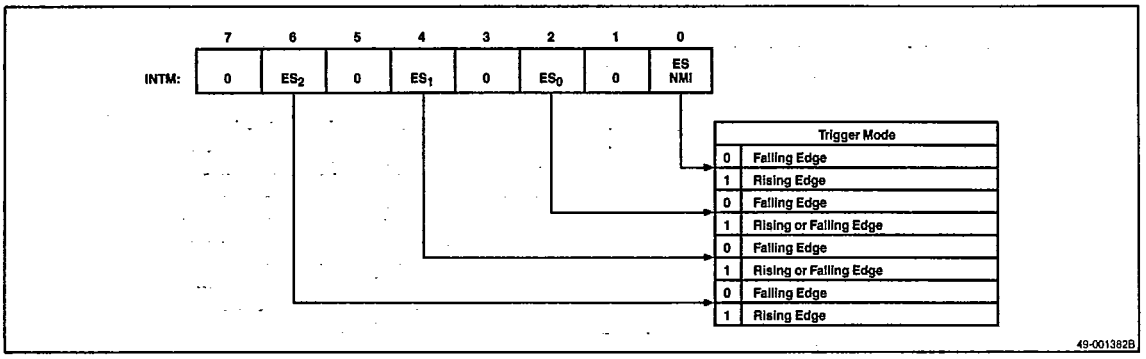


Figure 19. I/O Request Control Registers

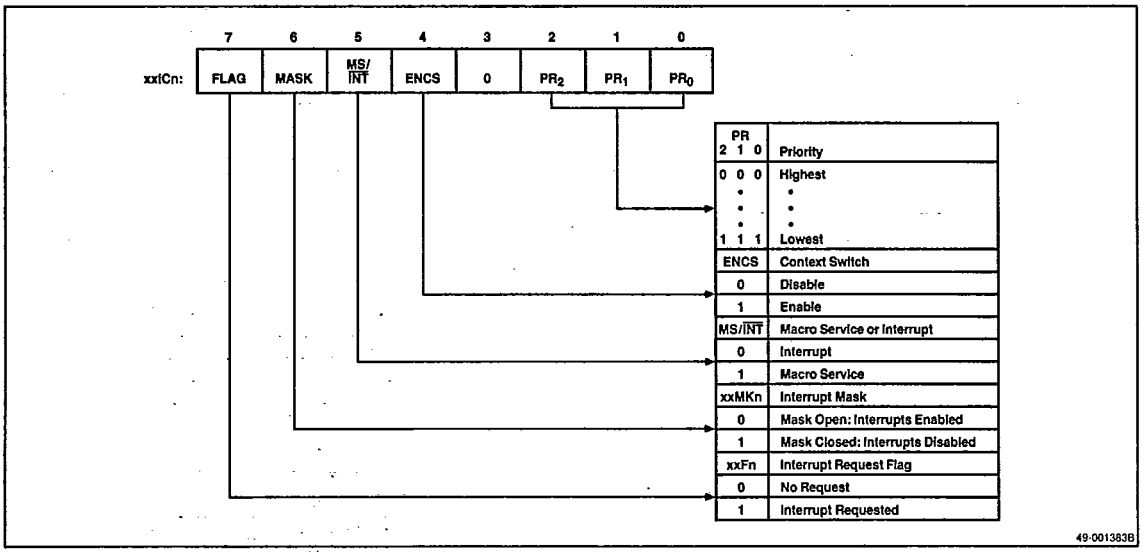
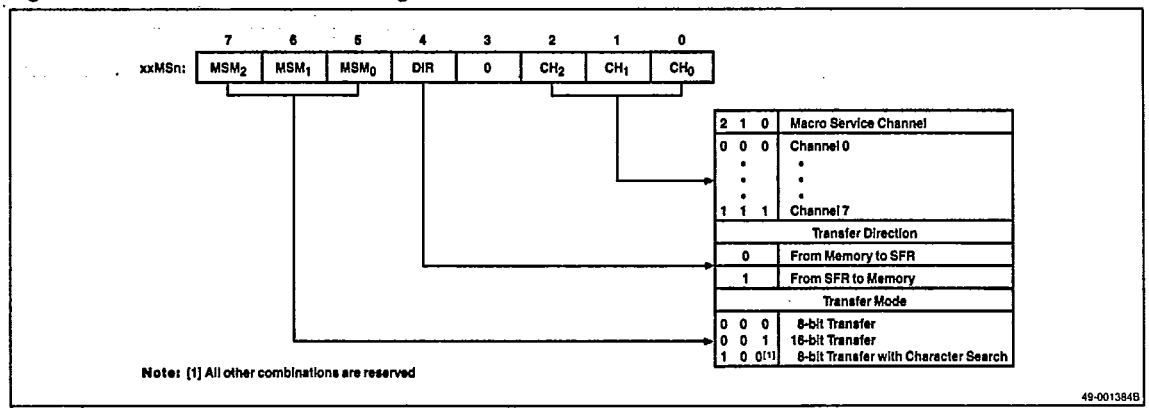


Figure 20. Macro Service Control Registers



T-49-19-16

Figure 21. Serial Communication Mode Register

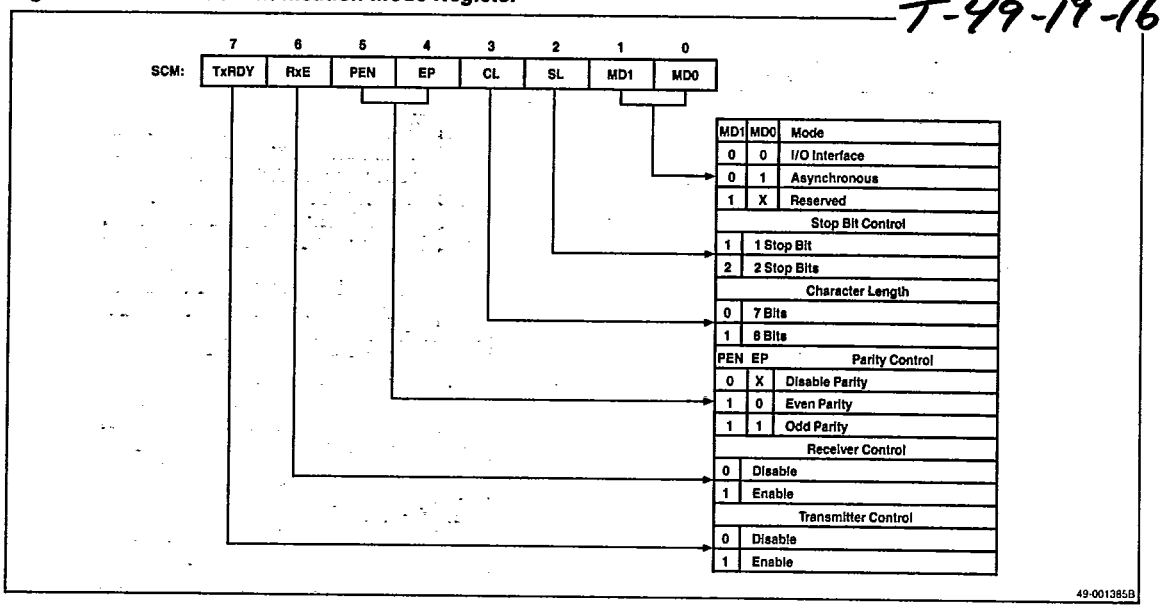
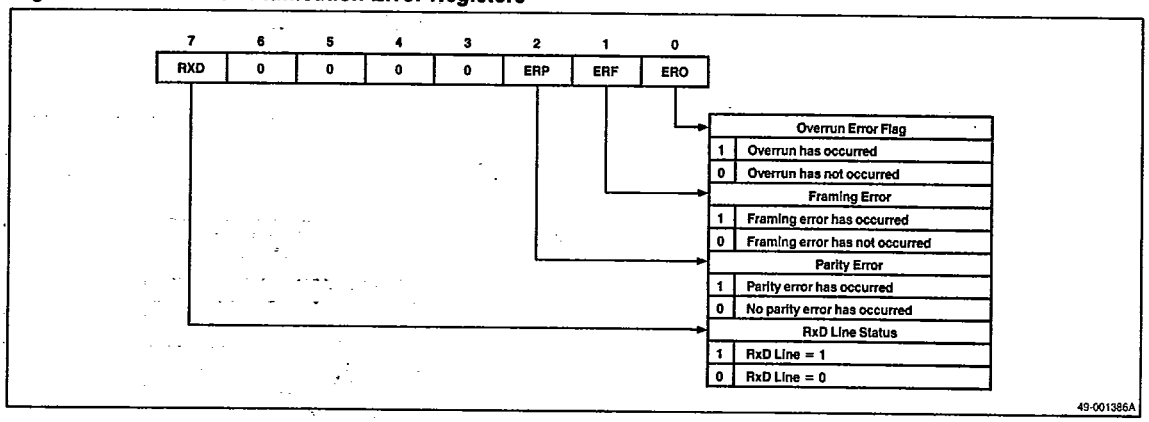


Figure 22. Serial Communication Error Registers



5

Figure 23. Timer Control 0 Register

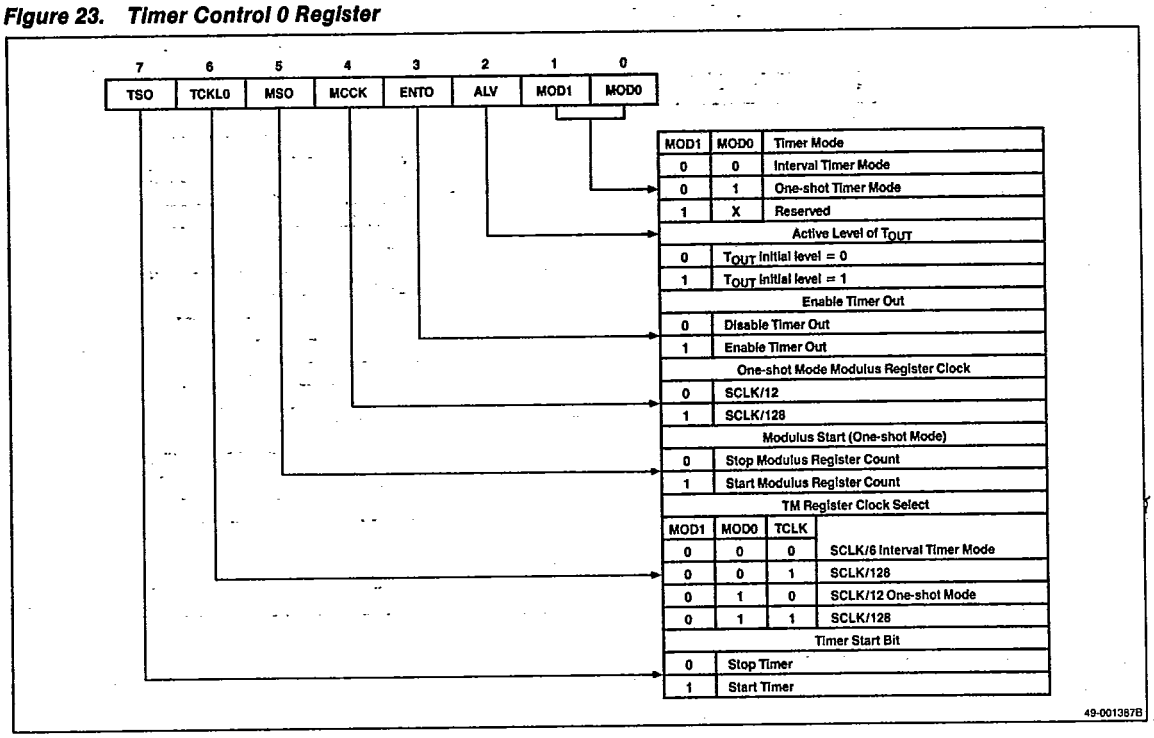


Figure 24. Serial Communication Control Register

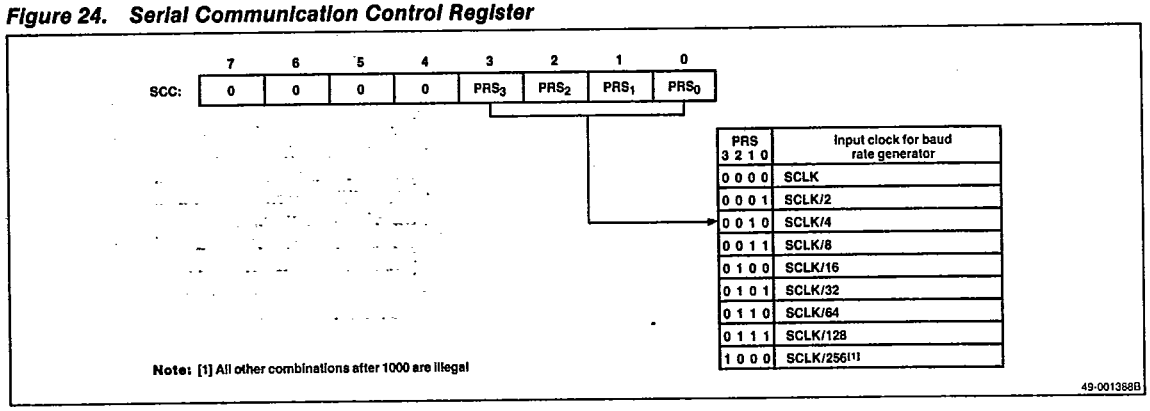


Figure 25. Timer Control 1 Register

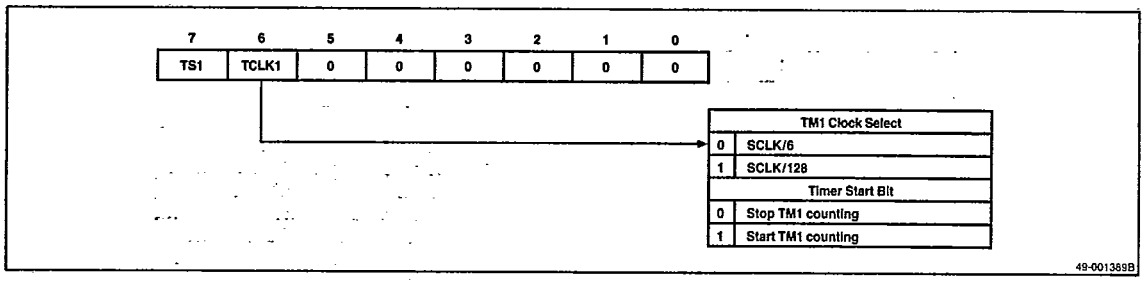


Figure 26. DMA Mode Registers

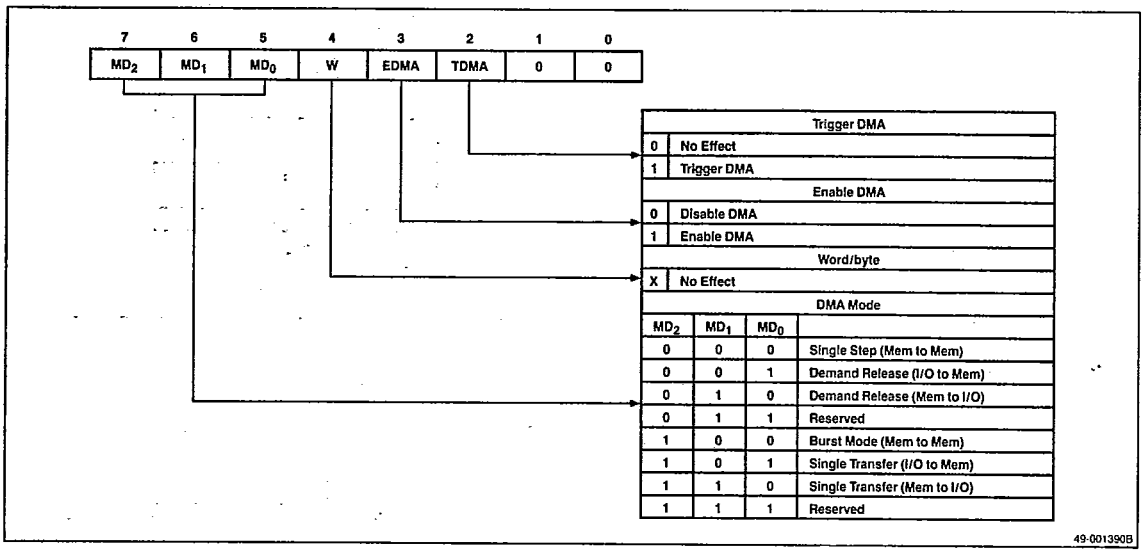
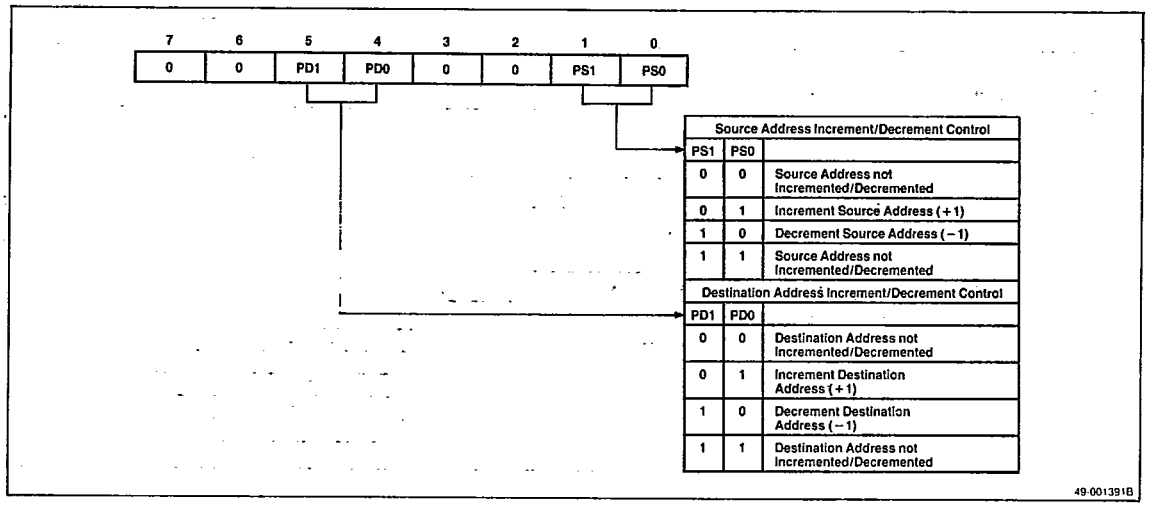


Figure 27. DMA Control Register



5

Figure 28. Refresh Mode Register

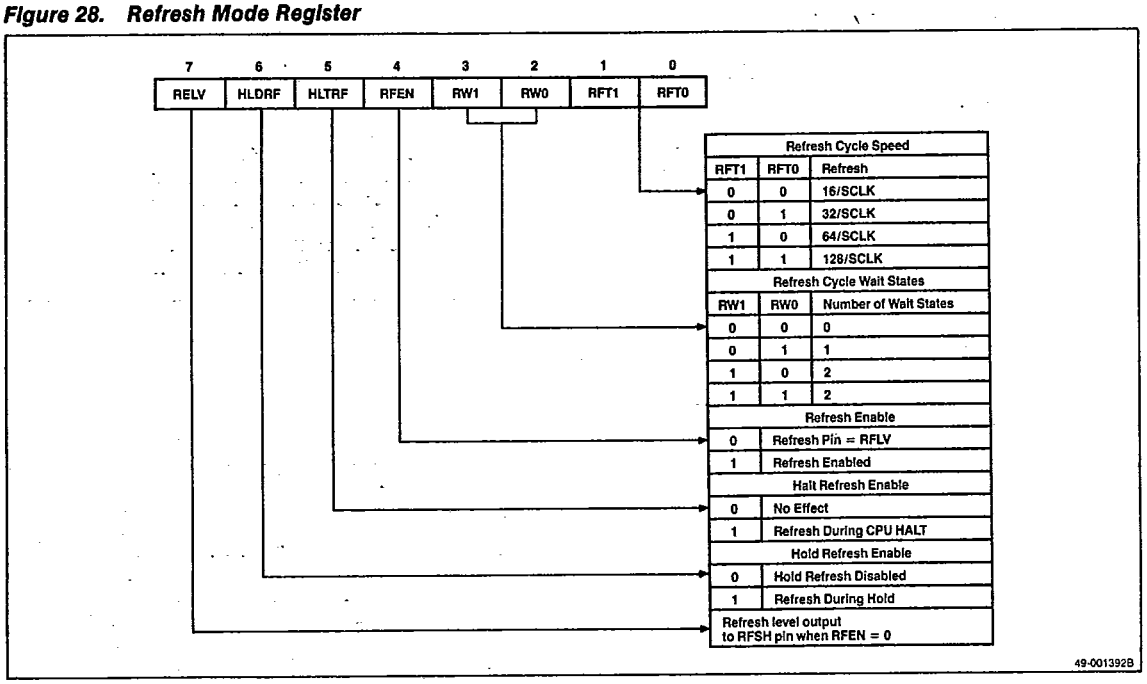


Figure 29. Time Base I/O Request Control Register

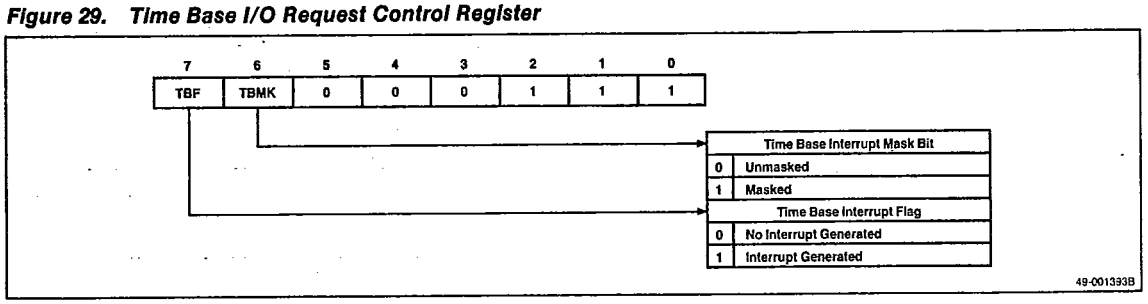
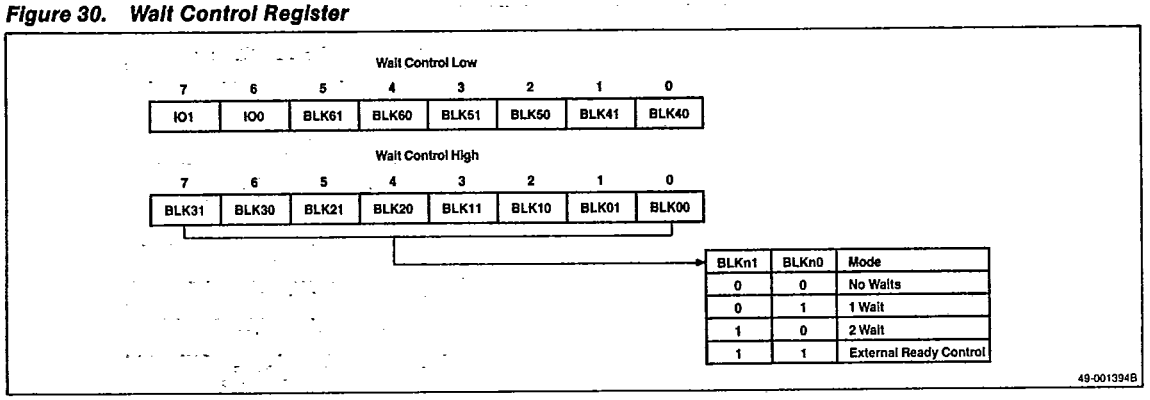


Figure 30. Wait Control Register



T-49-19-16

Figure 31. Processor Control Register

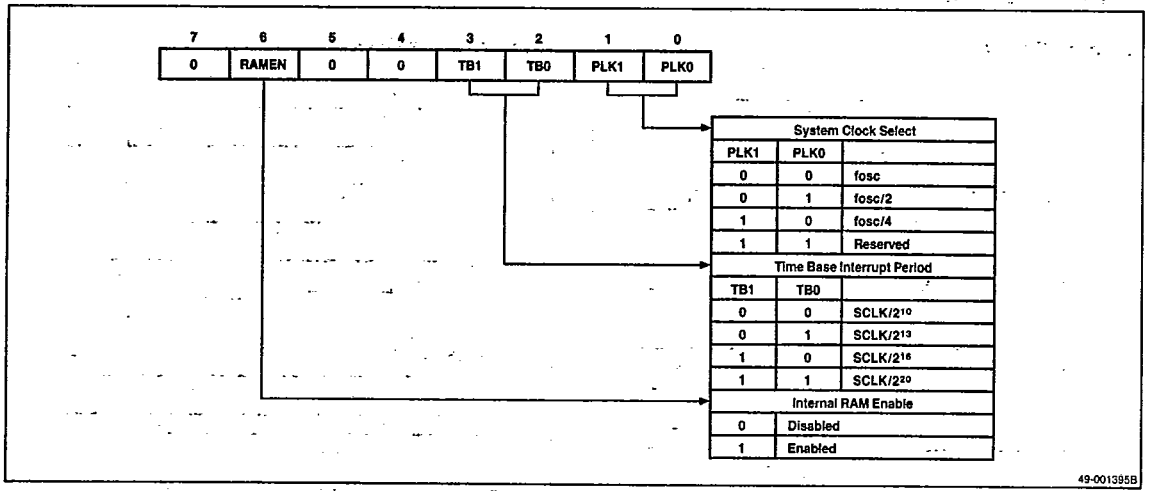
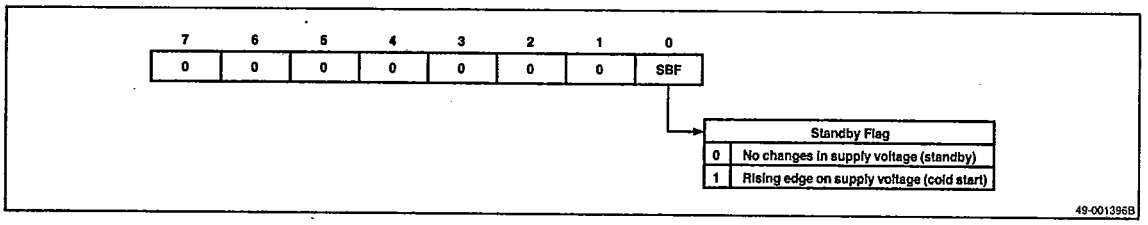


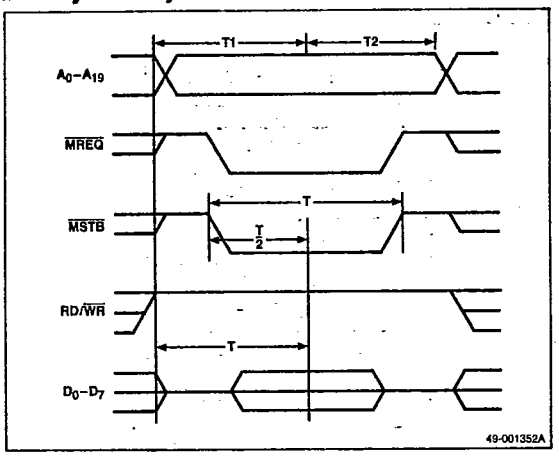
Figure 32. Standby Register



5

Timing Waveforms

Memory Read Cycle



Memory Write Cycle