Pilot™ Motion Processor

MC3510 Single Chip
Technical Specifications
for Stepping Motion Control

Warranty

PMD warrants performance of its products to the specifications applicable at the time of sale in accordance with PMD's standard warranty. Testing and other quality control techniques are utilized to the extent PMD deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Performance Motion Devices, Inc. (PMD) reserves the right to make changes to its products or to discontinue any product or service without notice, and advises customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

Safety Notice

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage. Products are not designed, authorized, or warranted to be suitable for use in life support devices or systems or other critical applications. Inclusion of PMD products in such applications is understood to be fully at the customer's risk.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent procedural hazards.

Disclaimer

PMD assumes no liability for applications assistance or customer product design. PMD does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of PMD covering or relating to any combination, machine, or process in which such products or services might be or are used. PMD's publication of information regarding any third party's products or services does not constitute PMD's approval, warranty or endorsement thereof.

Related Documents

Pilot Motion Processor User's Guide (MC3000UG)

How to set up and use all members of the Pilot Motion Processor family.

Pilot Motion Processor Programmer's Reference (MC3000PR)

Descriptions of all Pilot Motion Processor commands, with coding syntax and examples, listed alphabetically for quick reference.

Pilot Motion Processor Technical Specifications

These booklets contain physical and electrical characteristics, timing diagrams, pinouts and pin descriptions of each:

MC3110, for brushed servo motion control (MC3110TS)

MC3310, for brushless servo motion control (MC3310TS)

MC3410, for microstepping motion control (MC3410TS)

MC3510, for stepper motion control (MC3510TS)

Pilot Motion Processor Developer's Kit Manual (DK3000M)

How to install and configure the DK3510 developer's kit PC board.

Table of Contents

1 The Pilot Family

	MC3110	MC3310	MC3410	MC3510
Number of axes	1	1	1	1
Motor type supported	Brushed servo	Brushless servo	Stepping	Stepping
Output format	Brushed servo (single phase)	Commutated (6-step or sinusoidal)	Microstepping	Pulse and Direction
Incremental encoder input	√	√	√	√
Parallel word device input	√	√	√	√
Parallel communication	√[1]	√[1]	√[1]	√[1]
Serial communication	√	√	√	√
S-curve profiling	√	√	√	√
On-the-fly changes	√	√	√	√
Directional limit switches	√	√	√	√
Programmable bit output	√	√	√	√
Software-invertable signals	√	√	√	√
PID servo control	√	√	−	−
Feedforward (accel & vel)	√	√	−	−
Derivative sampling time	√	√	−	−
Data trace/diagnostics	√	√	√	√
PWM output	√	√	√	−
Pulse & direction output	−	−	−	√
Index & Home signals	√	√	√	√
Motion error detection	√	√	√ (with encoder)	√ (with encoder)
Axis settled indicator	√	√	√ (with encoder)	√ (with encoder)
DAC-compatible output	√	√	√	−
Position capture	√	√	√	√
Analog input	√	√	√	√
User-defined I/O	√	√	√	√
External RAM support	√	√	√	√
Multi-chip synchronization	√ (MC3113)	√ (MC3313)	√ (MC3413)	−
Chip part numbers	MC3110	MC3310	MC3410	MC3510
Developer's Kit p/n's:	DK3110	DK3310	DK3410	DK3510

[1] Parallel communication is available via an additional logic device

Introduction

This manual describes the operational characteristics of the MC3510 Motion Processor from PMD. This device is a member of the MC3000 family of single-chip, single-axis motion processors.

Each device of the MC3000 family is a complete chip-based motion processor providing trajectory generation and related motion control functions for one axis including pulse and direction output or servo loop closure or on-board commutation where appropriate. This family of products provides a software-compatible selection of dedicated motion processors that can handle a large variety of system configurations.

The chip architecture not only makes it ideal for the task of motion control, it allows for similarities in software commands, so software written for one motor type can be re-used if the motor type is changed.

Pilot Family Summary

MC3110 – This single-chip, single-axis motion processor outputs motor commands in either Sign/Magnitude PWM or DAC-compatible format for use with brushed servo motors, or with brushless servo motors having external commutation.

MC3310 – This single-chip, single-axis motion processor outputs sinusoidally commutated motor signals appropriate for driving brushless motors. Depending on the motor type, the output is a two-phase or three-phase signal in either PWM or DAC-compatible format.

MC3410 – This single-chip, single-axis motion processor outputs microstepping signals for stepping motors. Two phased signals per axis are generated in either PWM or DAC-compatible format.

MC3510 – This single-chip, single-axis motion processor outputs pulse and direction signals for stepping motor systems.

2 Functional Characteristics

2.1   Configurations, parameters, and performance

Configuration	Single axis, single chip.
Operating modes	Open loop (pulse generator is driven by trajectory generator output) Stall detection (pulse generator is driven by trajectory generator output and encoder feedback is used for stall detection)
Communication modes	8/16 parallel (8 bit external parallel bus with 16 bit internal command word size)
	16/16 parallel (16 bit external parallel bus with 16 bit internal command word size)
	Point to point asynchronous serial
	Multi-drop asynchronous serial
Serial port baud rate range	1,200 baud to 416,667 baud
Position range	-2,147,483,648 to +2,147,483,647 counts
Velocity range	-32,768 to +32,767 counts/sample with a resolution of 1/65,536 counts/sample
Acceleration/ deceleration ranges	-32,768 to +32,767 counts/sample^2 with a resolution of 1/65,536 counts/sample^2
Jerk range	0 to ½ counts/sample^3, with a resolution of 1/4,294,967,296 counts/sample^3
Profile modes	S-curve point-to-point (Velocity, acceleration, jerk, and position parameters)
	Trapezoidal point-to-point (Velocity, acceleration, deceleration, and position parameters)
	Velocity-contouring (Velocity, acceleration, and deceleration parameters)
Position error tracking	Motion error window (allows axis to be stopped upon exceeding programmable window)
	Tracking window (allows flag to be set if axis exceeds a programmable position window)
	Axis settled (allows flag to be set if axis exceeds a programmable position window for a programmable amount of time after trajectory motion is compete)
Maximum pulse rate	50,000 pulses/sec
Maximum encoder rate	Incremental (up to 5 million counts/sec)
	Parallel-word (up to 160 million counts/sec)
Parallel encoder word size	16 bits
Parallel encoder read rate	20 kHz (reads all axes every 50 μsec)
Cycle loop timing range	102.4 μsec to 32.767 milliseconds
Minimum cycle loop time	102.4 μsec
Limit switches	2 per axis: one for each direction of travel
Position-capture triggers	2 per axis: index and home signals
Other digital signals	1xAxisIn, 1xAxisOut, 1xAtRest
Software-invertable signals	Index, Home, AxisIn, AxisOut, PositiveLimit, NegativeLimit (all individually programmable)
Analog input	8 10-bit analog inputs
User defined discrete I/O	256 16-bit wide user defined I/O
RAM/external memory support	65,536 blocks of 32,768 16-bit words per block. Total accessible memory is 2,147,483,648 16 bit words

Trace modes	one-time
	continuous
Max. number of trace variables	4
Number of traceable variables	20
Number of host instructions	112

2.2   Physical characteristics and mounting dimensions

All dimensions are in inches (with millimeters in brackets).
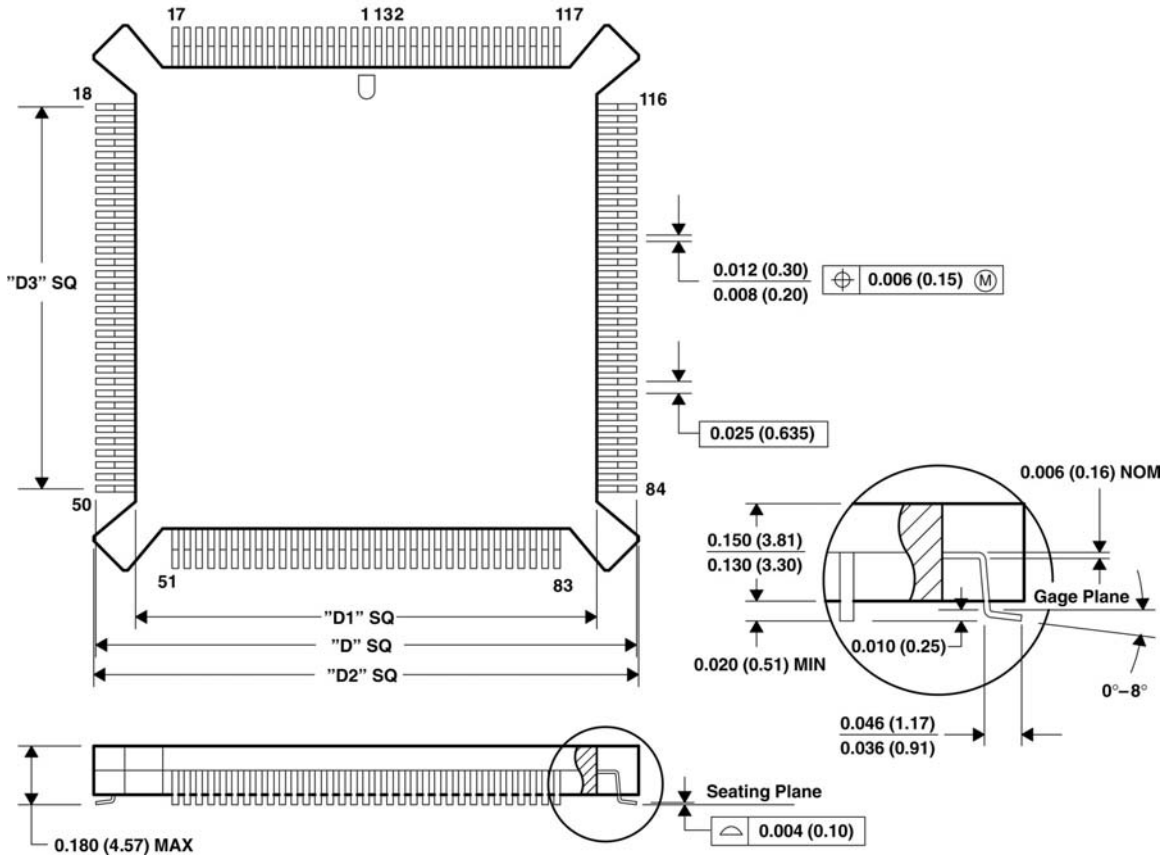
Dimension	Minimum (inches)	Maximum (inches)
D	1.070	1.090
D1	0.934	0.966
D2	1.088	1.112
D3	0.800 nominal	

2.3   Environmental and electrical ratings

Storage Temperature (T_s)	-55 °C to 150 °C
Operating Temperature (T_a)	0 °C to 70 °C*
Power Dissipation (P_d)	400 mW
Nominal Clock Frequency (F_{clk})	20.0 MHz
Supply Voltage limits (V_{cc})	-0.3V to +7.0V
Supply Voltage operating range (V_{cc})	4.75V to 5.25V

* An industrial version with an operating range of -40°C to 85°C is also available. Please contact PMD for more information.

2.4   System configuration

The following figure shows the principal control and data paths in an MC3510 system.
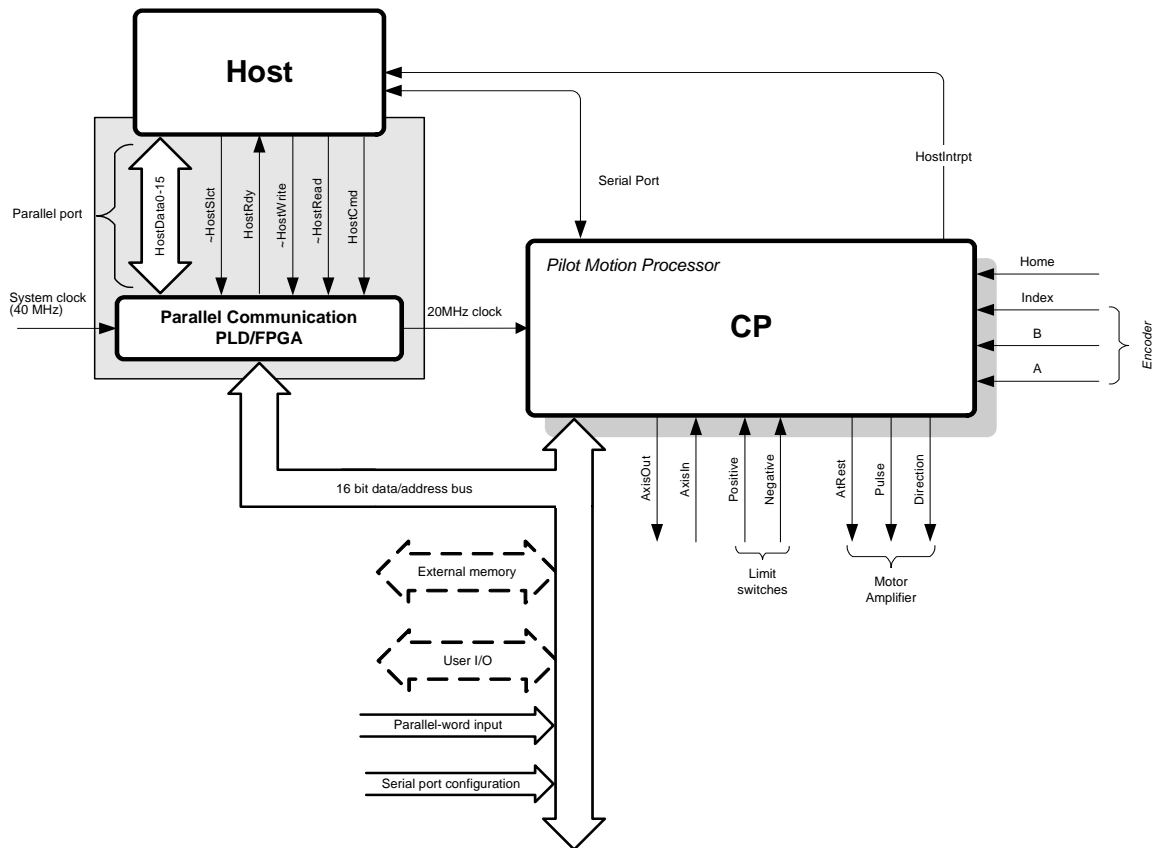
The shaded area shows the CPLD/FPGA that must be provided by the designer if parallel communication is required. A description and the necessary logic (in the form of schematics) of this device are detailed in section 6 of this manual. The CP chip contains the profile generator, which calculates velocity, acceleration, and position values for a trajectory. The output of the trajectory generator is used to produce pulse and direction signals that control motor position.

Optional axis position information returns to the motion in the form of incremental encoder feedback or in the form of parallel-word feedback. This position feedback may be used to detect motor stalling errors.

2.5   Peripheral device address mapping

Device addresses on the CP chip's data bus are memory-mapped to the following locations:

Address	Device	Description
0200h	Serial port data	Contains the configuration data (transmission rate, parity, stop bits, etc) for the asynchronous serial port
0800h	Parallel-word encoder	Base address for parallel-word feedback devices
1000h	User-defined	Base address for user-defined I/O devices
2000h	RAM page pointer	Page pointer to external memory
4000h	Motor-output DACs	Base address for motor-output D/A converters
8000h	Parallel interface	Base address for parallel interface communication

3 Electrical Characteristics

3.1 DC characteristics

(V_{cc} and T_a per operating ratings, F_{clk} = 20.0 MHz)

Symbol	Parameter	Minimum	Maximum	Conditions
V_{cc}	Supply Voltage	4.75 V	5.25 V	
I_{dd}	Supply Current		80 mA	open outputs

Input Voltages

Symbol	Parameter	Minimum	Maximum	Conditions
V_{ih}	Logic 1 input voltage	2.0 V	V_{cc} + 0.3 V	
V_{il}	Logic 0 input voltage	-0.3 V	0.8 V	
V_{ihclk}	Logic 1 voltage for clock pin (ClockIn)	3.0 V	V_{cc} + 0.3 V	
V_{oclk}	Logic 0 voltage for clock pin (ClockIn)	-0.3 V	0.7 V	
$V_{ihreset}$	Logic 1 voltage for reset pin (reset)	2.2 V	V_{cc} + 0.3 V	

Output Voltages

Symbol	Parameter	Minimum	Maximum	Conditions
V_{oh}	Logic 1 Output Voltage	2.4 V		@CP I_o = -23 mA
V_{ol}	Logic 0 Output Voltage		0.33 V	@CP I_o = 6 mA

Other

Symbol	Parameter	Minimum	Maximum	Conditions
I_{out}	Tri-State output leakage current	-5 μA	5 μA	@CP 0 < V_{out} < V_{cc}
I_{in}	Input current	-10 μA	10 μA	@CP 0 < V_i < V_{cc}
C_{io}	Input/Output capacitance	15 pF		@CP typical

Analog Input

Symbol	Parameter	Minimum	Maximum	Conditions
Z_{ai}	Analog input source impedance		9kΩ	
E_{dnl}	Differential nonlinearity error. Difference between the step width and the ideal value.	-1	1.5 LSB	
E_{inl}	Integral nonlinearity error. Maximum deviation from the best straight line through the ADC transfer characteristics, excluding the quantization error.		+/-1.5 LSB	

3.2 AC characteristics

See timing diagrams, Section 4, for *Tn* numbers. The symbol "~" indicates active low signal.

Timing Interval	*Tn*	Minimum	Maximum
Clock Frequency (F_{clk})		> 0 MHz	20 MHz *(note 1)*
Clock Pulse Width	T1	25 nsec	
Clock Period *(note 2)*	T2	50 nsec	
Encoder Pulse Width	T3	150 nsec	
Dwell Time Per State	T4	75 nsec	
~HostSlct Hold Time	T6	0 nsec	

Timing Interval	*Tn*	Minimum	Maximum
~HostSlct Setup Time	T7	0 nsec	
HostCmd Setup Time	T8	0 nsec	
HostCmd Hold Time	T9	0 nsec	
Read Data Access Time	T10		25 nsec
Read Data Hold Time	T11		10 nsec
~HostRead High to HI-Z Time	T12		20 nsec
HostRdy Delay Time	T13	100 nsec	150 nsec
~HostWrite Pulse Width	T14	70 nsec	
Write Data Delay Time	T15		35 nsec
Write Data Hold Time	T16	0 nsec	
Read Recovery Time *(note 2)*	T17	60 nsec	
Write Recovery Time *(note 2)*	T18	60 nsec	
Read Pulse Width	T19	70 nsec	
Address Setup Delay Time	T20		7 nsec
Data Access Time	T21		19 nsec
Data Hold Time	T22		2 nsec
Address Setup Delay Time	T23		7 nsec
Address Setup to WriteEnable High	T24	72 nsec	
RAMSlct Low to WriteEnable High	T25		79 nsec
Address Hold Time	T26	17 nsec	
WriteEnable Pulse Width	T27	39 nsec	
Data Setup Time	T28		3 nsec
Data Setup before Write High Time	T29		42 nsec
Address Setup Delay Time	T30		7 nsec
Data Access Time	T31		71 nsec
Data Hold Time	T32		2 nsec
Address Setup Delay Time	T33		7 nsec
Address Setup to WriteEnable High	T34	122 nsec	
PeriphSlct Low to WriteEnable High	T35		129 nsec
Address Hold Time	T36	17 nsec	
WriteEnable Pulse Width	T37	89 nsec	
Data Setup Time	T38		3 nsec
Data Setup before Write High Time	T39		92 nsec
Read to Write Delay Time	T40	50 nsec	
Reset Low Pulse Width	T50	5.0 μsec	
RAMSlct Low to Strobe Low	T51		1 nsec
Strobe High to RAMSlct High	T52		4 nsec
WriteEnable Low to Strobe Low	T53		1 nsec
Strobe High to WriteEnable High	T54		3 nsec
PeriphSlct Low to Strobe Low	T55		1 nsec
Strobe High to PeriphSlct High	T56		4 nsec

Note 1 Performance figures and timing information valid at F_{clk} = 20.0 MHz only. For timing information and performance parameters at F_{clk} < 20.0 MHz, refer to section 7.1.

Note 2 The clock low/high split has an allowable range of 45-55%.

4 I/O Timing Diagrams

For the values of *Tn*, please refer to the table in Section 3.2.

The host interface timing shown in diagrams 4.4 and 4.5 is only valid when an external logic device is used to provide a parallel communication interface. Refer to section 6 for more information.

4.1   Clock

ClockIn

T1   T1   T2
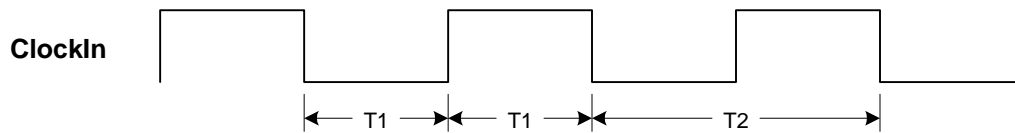
4.2   Quadrature encoder input

T3   T3

Quad A

T4   T4

Quad B

~Index

4.3   Reset

V_{cc}

ClockIn

~RESET

T50
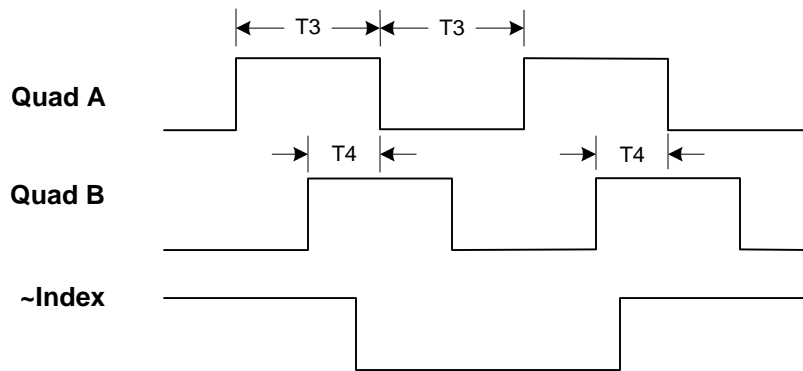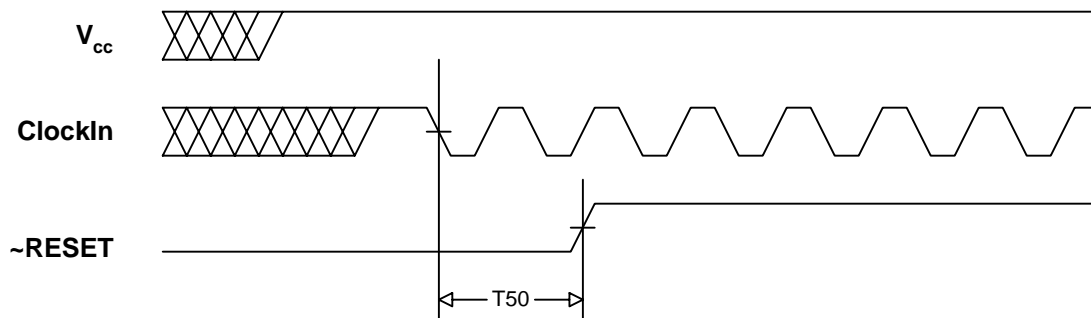
4.4   Host interface, 8/16 mode (requires external logic device)

4.4.1   Instruction write, 8/16 mode
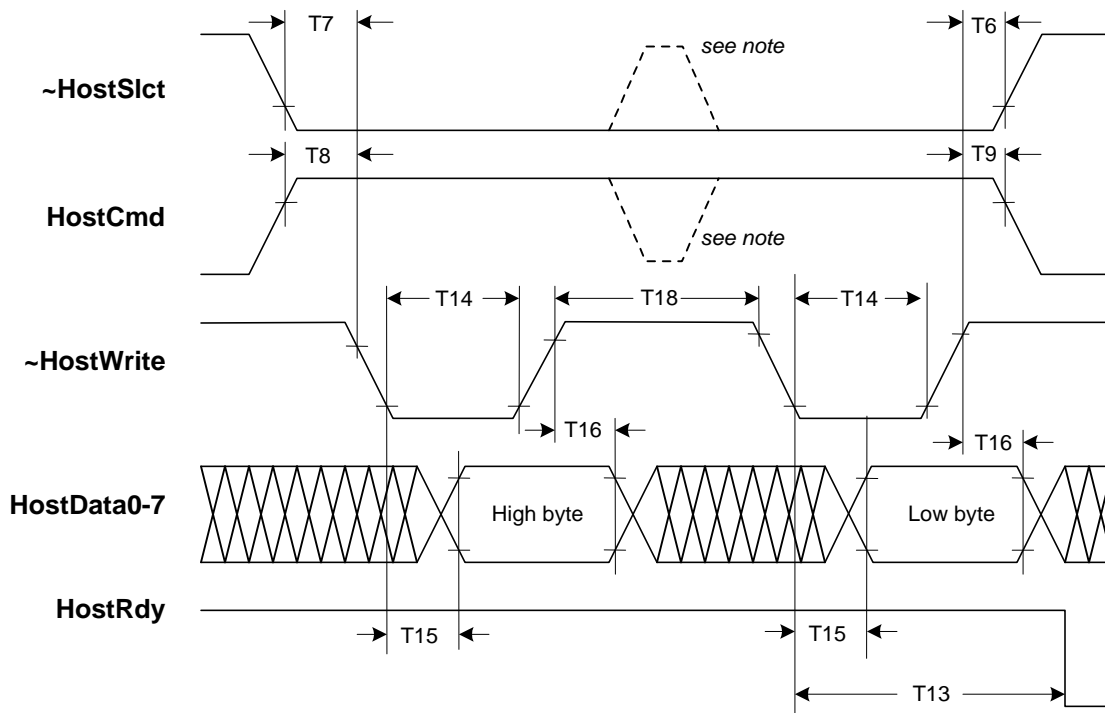
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.2   Data write, 8/16 mode
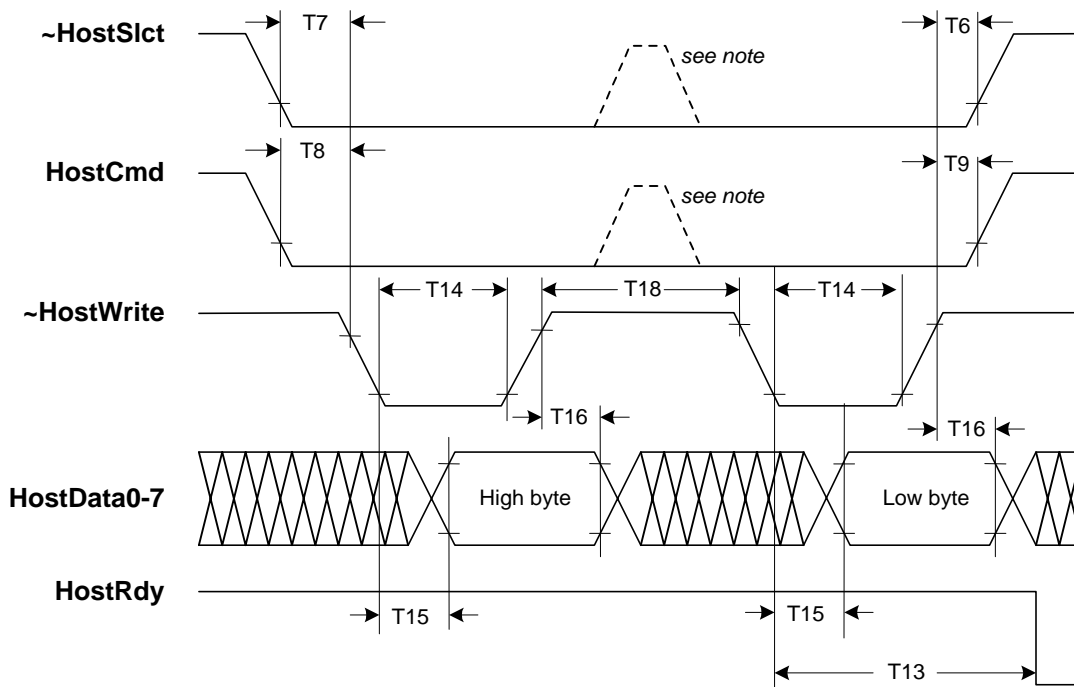
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.3   Data read, 8/16 mode
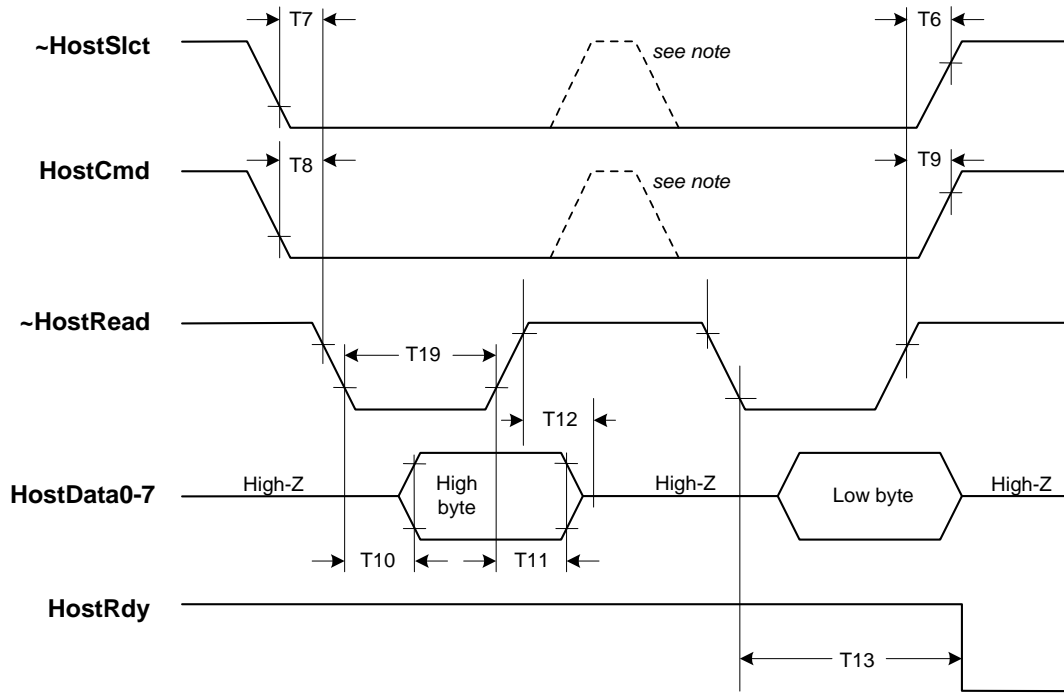
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.4   Status read, 8/16 mode
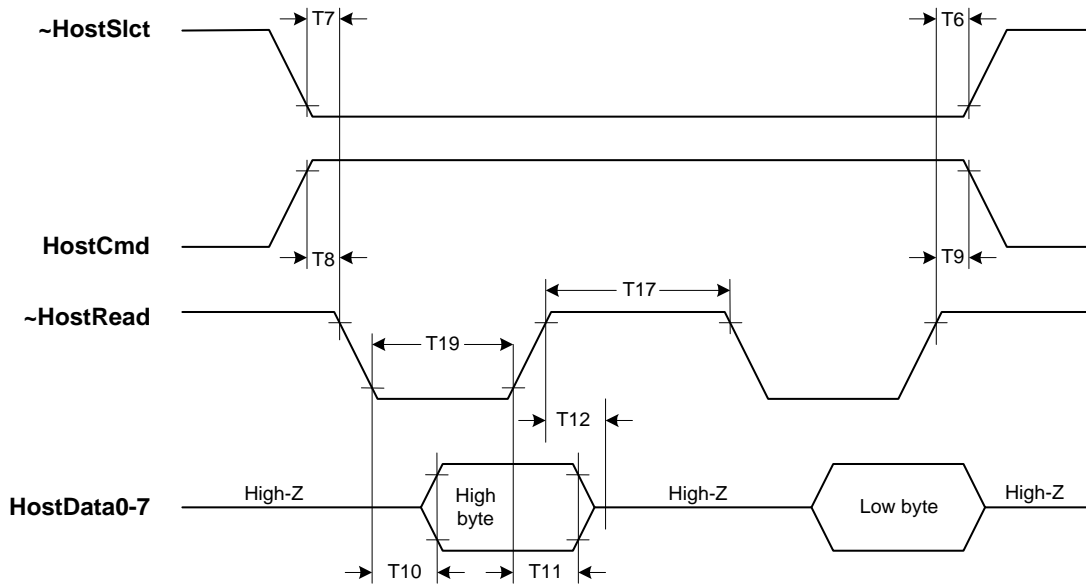
4.5   Host interface, 16/16 mode (requires external logic device)

4.5.1   Instruction write, 16/16 mode
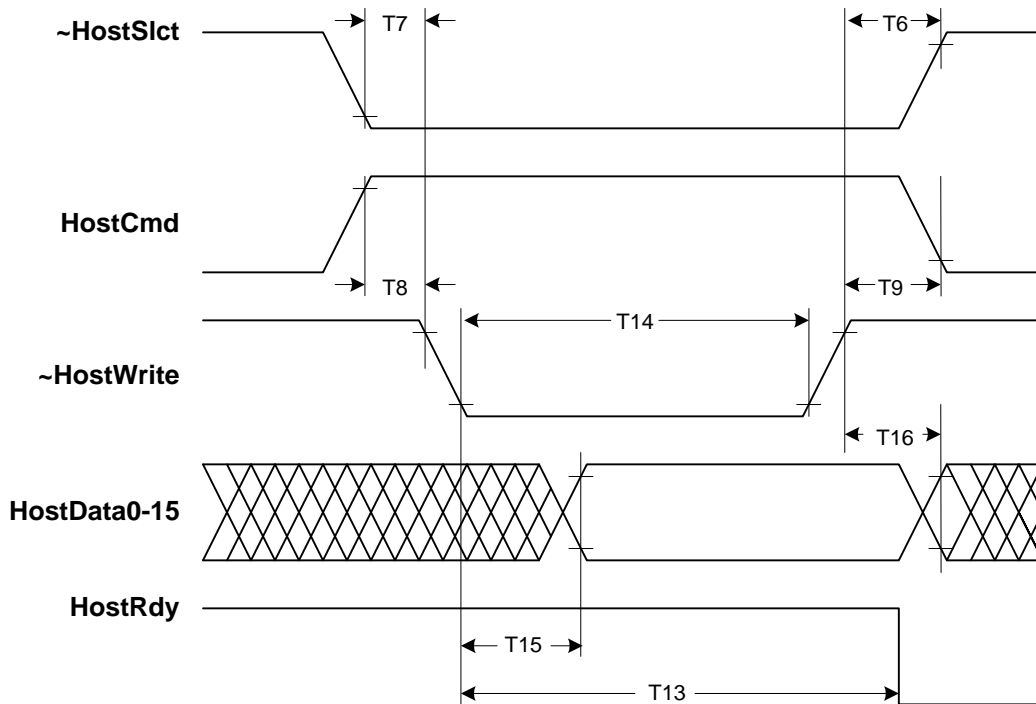
4.5.2   Data write, 16/16 mode
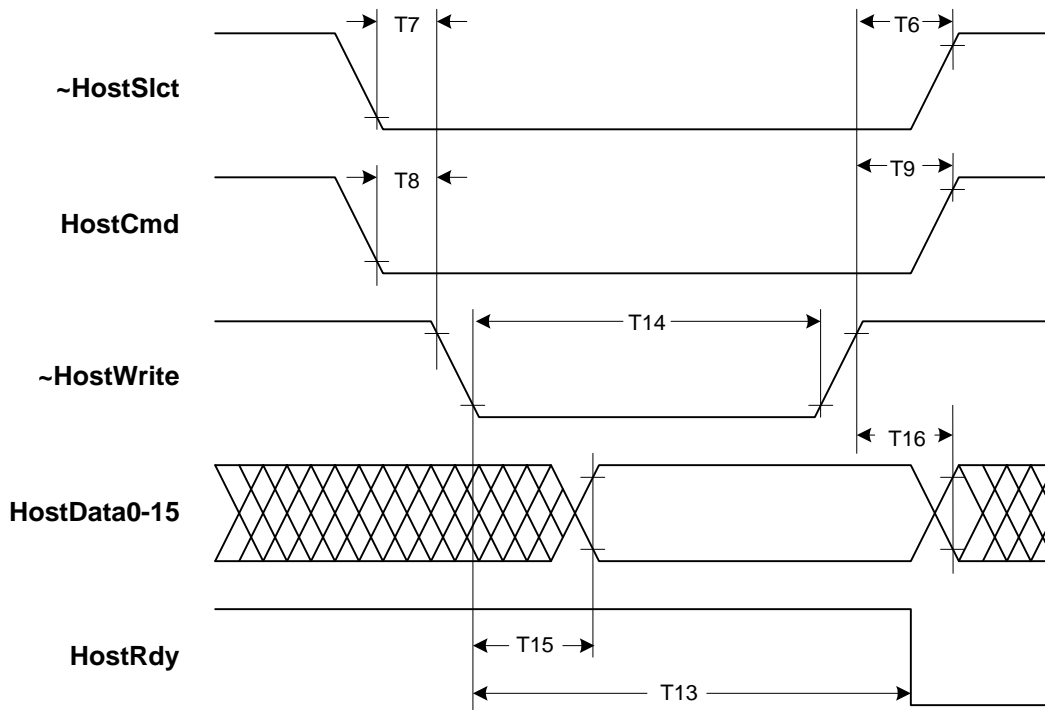
4.5.3   Data read, 16/16 mode
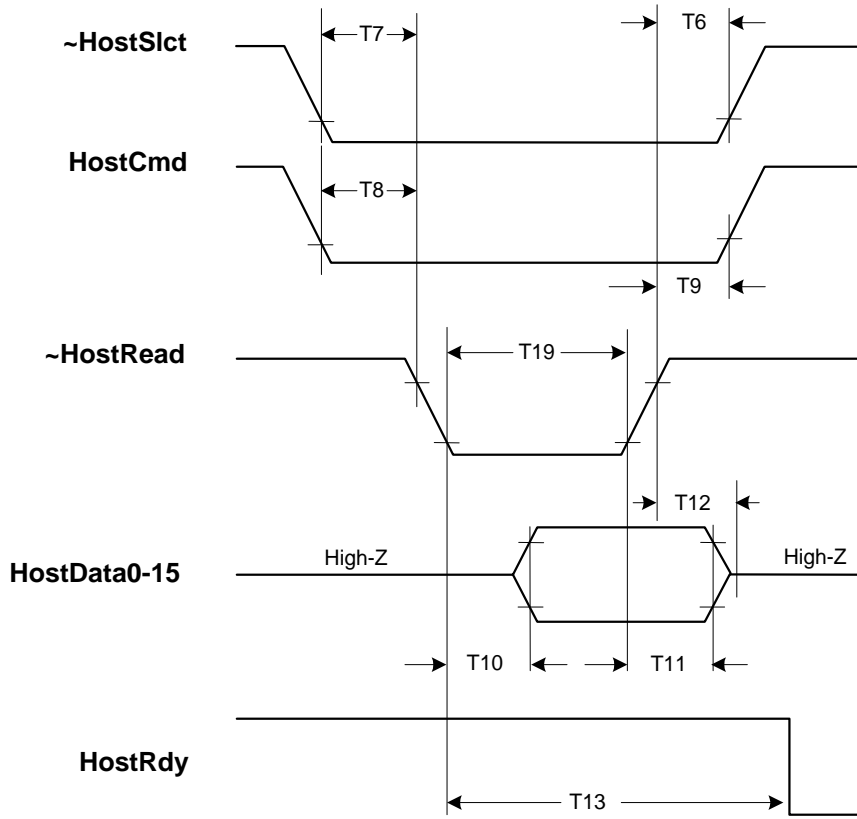
4.5.4   Status read, 16/16 mode
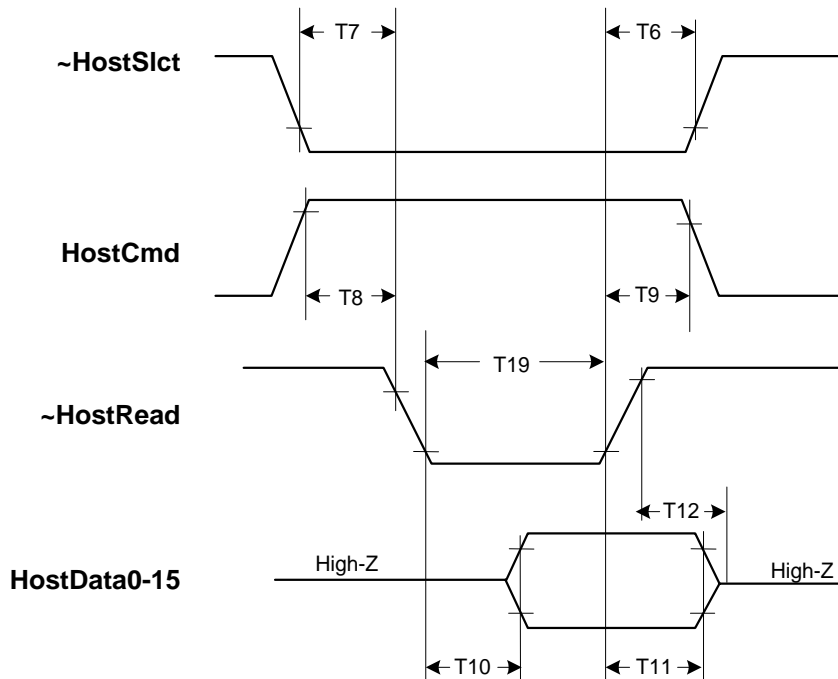
4.6 External memory timing

4.6.1 External memory read

Note: PMD recommends using memory with an access time no greater than 15 nsec.
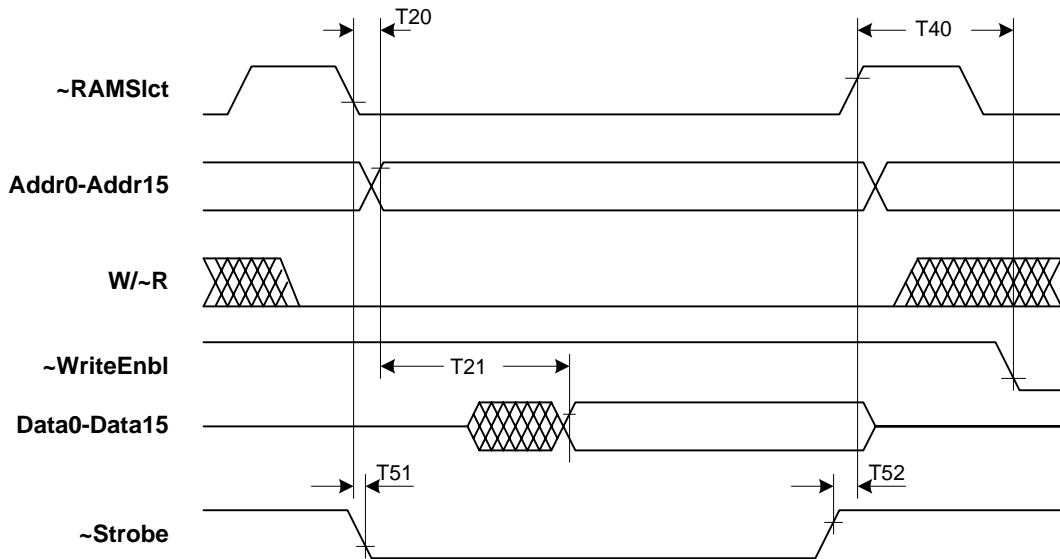
4.6.2 External memory write
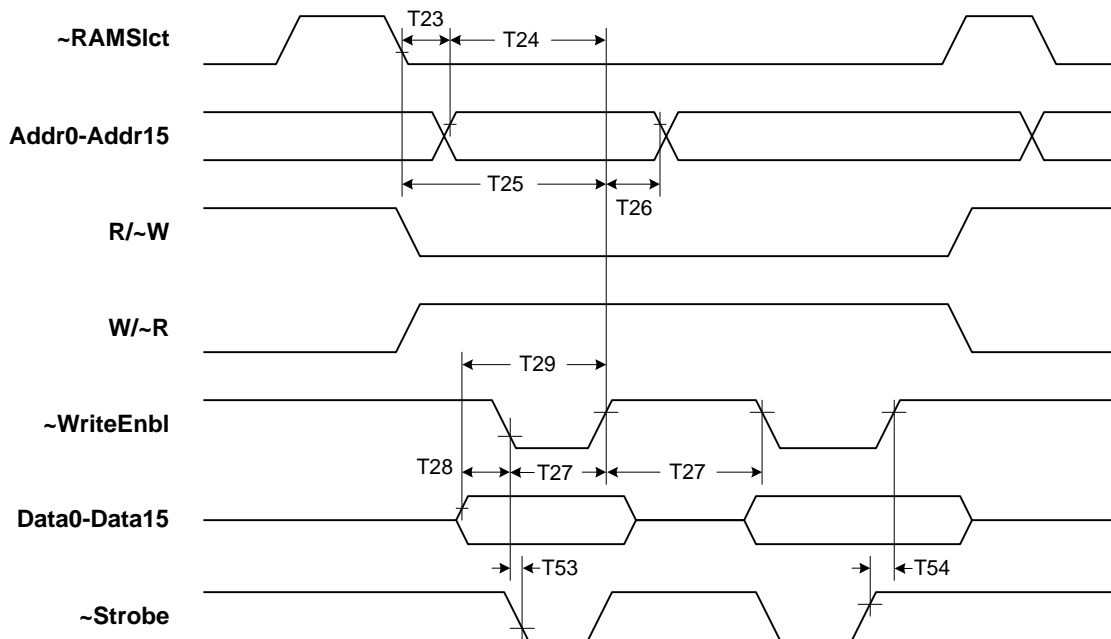
4.7   Peripheral device timing

4.7.1   Peripheral device read
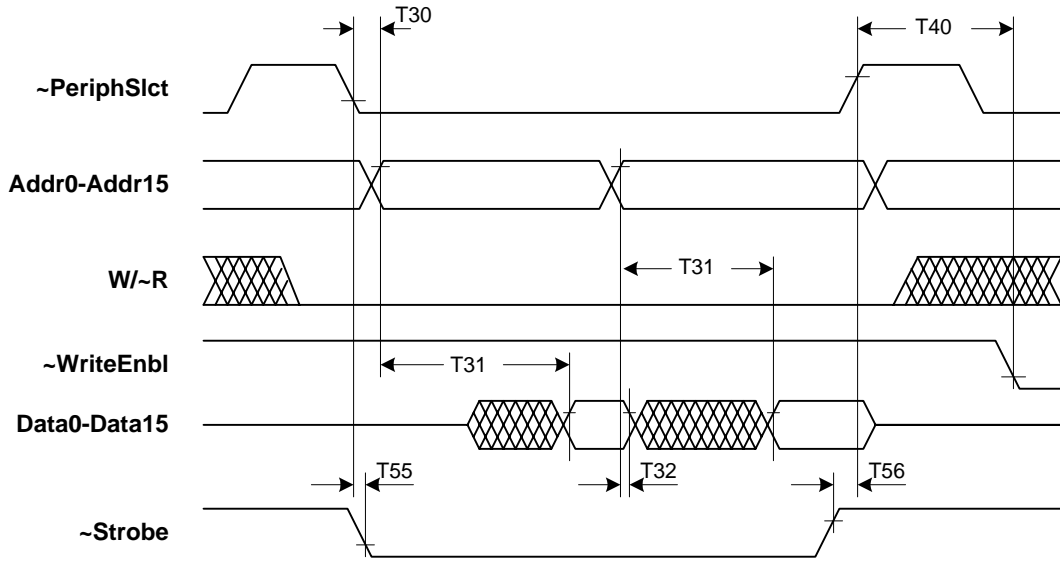
4.7.2   Peripheral device write
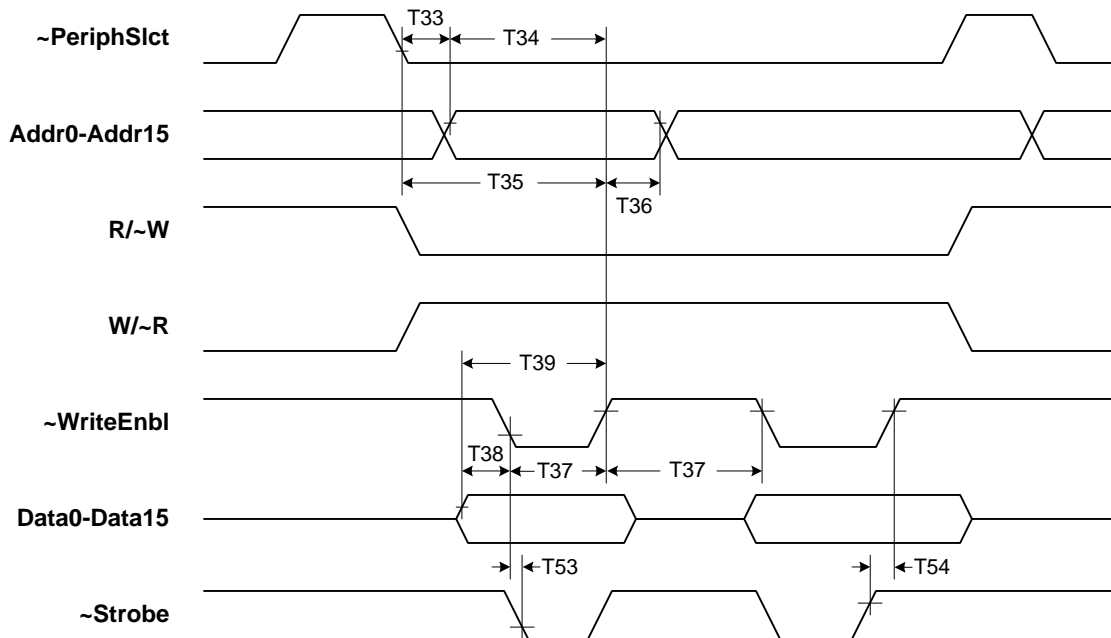
5 Pinouts and Pin Descriptions

5.1   Pinouts for MC3510

2, 7, 13, 21, 35, 36, 40, 47, 50,
52, 60, 62, 66, 93, 103, 121

VCC

Pin	Signal		Signal	Pin
1	~WriteEnbl		AnalogVcc	84
4	R/~W		AnalogRefHigh	85
6	~Strobe		AnalogRefLow	86
130	~PeriphSlct		AnalogGnd	87
129	~RAMSlct		Analog1	74
41	~Reset		Analog2	89
132	W/~R		Analog3	75
43	SrlRcv		Analog4	88
44	SrlXmt		Analog5	76
99	SrlEnable		Analog6	83
98	~HostIntrpt		Analog7	77
58	ClockIn		Analog8	82
110	Addr0		PosLim1	63
111	Addr1		NegLim1	64
112	Addr2		AxisOut1	94
114	Addr3		AxisIn1	72
115	Addr4		Direction1	105
116	Addr5		Pulse1	106
117	Addr6		AtRest1	107
118	Addr7		QuadA1	67
119	Addr8		QuadB1	68
122	Addr9		~Index1	69
123	Addr10	**CP**	~Home1	70
124	Addr11			
125	Addr12			
126	Addr13			
127	Addr14			
128	Addr15			
9	Data0			
10	Data1			
11	Data2			
12	Data3		I/OIntrpt	53
15	Data4		PrlEnable	65
16	Data5			
17	Data6			
18	Data7			
19	Data8			
22	Data9			
23	Data10			
24	Data11			
25	Data12			
26	Data13			
27	Data14			
28	Data15			

GND

3, 8, 14, 20, 29, 37, 46, 56, 59,
61, 71, 92, 104, 113, 120

Unassigned
5, 30-34, 38, 39, 42, 45, 48, 49,
51, 54, 55, 57, 73, 90, 91, 95-
97, 100-102, 108, 109, 131

AGND
78-81

5.2 CP chip pin description table

Pin Name and number		Direction	Description
~WriteEnbl	1	output	When *low*, this signal enables data to be written to the bus.
R/~W	4	output	This signal is *high* when the CP chip is performing a read, and *low* when it is performing a write.
~Strobe	6	output	This signal is *low* when the data and address are valid during CP communications.
~PeriphSlct	130	output	This signal is *low* when peripheral devices on the data bus are being addressed.
~RAMSlct	129	output	This signal is *low* when external memory is being accessed.
~Reset	41	input	This is the master reset signal. When brought *low*, this pin resets the processor to its initial conditions.
W/~R	132	output	This signal is the inverse of R/~W; it is *high* when R/~W is low, and vice versa. For some decode circuits, this is more convenient than R/~W.
SrlRcv	43	input	This pin receives serial data from the asynchronous serial port. If serial communication is not used, this pin should be tied to V_{cc}.
SrlXmt	44	output	This pin transmits serial data to the asynchronous serial port.
SrlEnable	99	output	This pin sets the serial port enable line. SrlEnable is always *high* for the point-to-point protocol and is *high* during transmission for the multi-drop protocol.
~HostIntrpt	98	output	When *low*, this signal causes an interrupt to be sent to the host processor.
I/OIntrpt	53	input	This signal interrupts the CP chip when a host I/O transfer is complete. It should be connected to CPIntrpt of the parallel interface chip. If the parallel interface is disabled (see below) this signal can be left unconnected or tied to V_{cc}.
PrlEnable	65	input	This signal enables/disables the parallel communication with the host. If this signal is tied *high*, the parallel interface is enabled. If this signal is tied *low* the parallel interface is disabled. See section 6 of this manual for more information on parallel communication. **WARNING! This signal should only be tied high if an external logic device that implements the parallel communication logic included in the design. This signal is an output during device reset and as such any connection to GND or V_{cc} must be via a series resistor.**
Data0 Data1 Data2 Data3 Data4 Data5 Data6 Data7 Data8 Data9 Data10 Data11 Data12 Data13 Data14 Data15	9 10 11 12 15 16 17 18 19 22 23 24 25 26 27 28	bi-directional	Multi-purpose data lines. These pins comprise the CP chip's external data bus, used for all communications with peripheral devices such as external memory or DACs. They may also be used for parallel-word input and for user-defined I/O operations.

Pin Name and number		Direction	Description
Addr0	110	output	Multi-purpose Address lines. These pins comprise the CP chip's external address bus, used to select devices for communication over the data bus.
Addr1	111		They may be used for DAC output, parallel word input, or user-defined I/O operations. See the *Pilot Motion Processor User's Guide* for a complete memory map.
Addr2	112		
Addr3	114		
Addr4	115		
Addr5	116		
Addr6	117		
Addr7	118		
Addr8	119		
Addr9	122		
Addr10	123		
Addr11	124		
Addr12	125		
Addr13	126		
Addr14	127		
Addr15	128		
ClockIn	58	input	This is the clock signal for the Motion Processor. It is driven at a nominal 20MHz.
AnalogVcc	84	input	CP chip analog power supply voltage. This pin must be connected to the analog input supply voltage, which must be in the range 4.5-5.5 V
			If the analog input circuitry is not used, this pin must be connected to V_{cc}.
AnalogRefHigh	85	input	CP chip analog high voltage reference for A/D input. The allowed range is AnalogRefLow to AnalogVcc.
			If the analog input circuitry is not used, this pin must be connected to V_{cc}.
AnalogRefLow	86	input	CP chip analog low voltage reference for A/D input. The allowed range is AnalogGND to AnalogRefHigh.
			If the analog input circuitry is not used, this pin must be connected to GND.
AnalogGND	87		CP chip analog input ground. This pin must be connected to the analog input power supply return.
			If the analog input circuitry is not used, this pin must be connected to GND.
Analog1	74	input	These signals provide general-purpose analog voltage levels, which are sampled by an internal A/D converter. The A/D resolution is 10 bits.
Analog2	89		The allowed range is AnalogRefLow to AnalogRefHigh.
Analog3	75		
Analog4	88		
Analog5	76		
Analog6	83		Any unused pins should be tied to AnalogGND.
Analog7	77		If the analog input circuitry is not used, these pins should be tied to GND.
Analog8	82		
Pulse1	106	output	This pins provides the pulse (also called step) signal to the motor amplifier. A "step" occurs when the signal transitions from a high state to a low state. This default operation can be changed using the **SetSignalSense** command. Refer to the *Pilot Programmer's Reference* for more information.
Direction1	105	output	This pin indicates the direction of motion and works in conjunction with the pulse signal. A high level on this signal indicates a positive direction move and a low level indicates a negative direction move.
AtRest1	107	output	The AtRest signal indicates that the axis is at rest and the step motor can be switched to low power or standby. A high level on this signal indicates the axis is at rest. A low signal indicates the axis is in motion.
QuadA1	67	input	These pins provide the A and B quadrature signals for the incremental encoder. When the axis is moving in the positive (forward) direction, signal A leads signal B by 90°.
QuadB1	68		The theoretical maximum encoder pulse rate is 5.1 MHz. Actual maximum rate will vary, depending on signal noise.
			NOTE: Many encoders require a pull-up resistor on each signal to establish a proper high signal. Check your encoder's electrical specification.

Pin Name and number		Direction	Description
~Index1	69	input	This pin provides the Index signal for the incremental encoder. A valid index pulse is recognized by the chip when this signal transitions from *high* to *low*.
			There is no internal gating of the index signal with the encoder A and B inputs. This must be performed externally if desired. Refer to the Application Notes section at the end of this manual for an example.
~Home1	70	input	This pin provides the Home signal, general-purpose inputs to the position-capture mechanism. A valid Home signal is recognized by the chip when ~Home goes *low*.
			WARNING! If this pin is not used, its signal should be tied high.
PosLim1	63	input	This signal provides input from the positive-side (forward) travel limit switch. On power-up or Reset this signal defaults to active *low* interpretation, but the interpretation can be set explicitly using the SetSignalSense instruction.
			WARNING! If this pin is not used, its signal should be tied high.
NegLim1	64	input	This signal provides input from the negative-side (reverse) travel limit switch. On power-up or Reset this signal defaults to active *low* interpretation, but the interpretation can be set explicitly using the SetSignalSense instruction.
			WARNING! If this pin is not used, its signal should be tied high. This signal is an output during device reset and as such any connection to GND or V_{cc} must be via a series resistor.
AxisOut1	94	output	This pin can be programmed to track the state of any bit in the status registers. If this pin is not used it may be left unconnected.
AxisIn1	72	input	This is a general-purpose or programmable input. It can be used as a breakpoint input, to stop a motion axis, or to cause an Update to occur. If this pin is not used it may be left unconnected.
V_{cc}	2, 7, 13, 21, 35, 36, 40, 47, 50, 52, 60, 62, 66, 93, 103, 121		CP digital supply voltage. All of these pins must be connected to the supply voltage. V_{cc} must be in the range 4.75 - 5.25 V
			WARNING! Pin 35 must be tied HIGH with a pull-up resistor. A nominal value of 22K Ohms is suggested.
GND	3, 8, 14, 20, 29, 37, 46, 56, 59, 61, 71, 92, 104, 113, 120		CP ground. All of these pins must be connected to the power supply return.
AGND	78-81		These signals must be tied to AnalogGND. If the analog input circuitry is not used, these pins must be tied to GND.
unassigned	45, 48, 49, 51, 54, 55, 73, 90, 91, 108, 109		These signals may be connected to GND for better noise immunity and reduced power consumption or they can be left unconnected (floating).
unassigned	5, 30-34, 38, 39, 42, 57, 95, 96, 97, 100, 101, 102, 131		These signals must be left unconnected (floating).

6 Parallel Communication

With the addition of an external logic device, the Pilot motion processor can communicate with a host processor using a parallel data stream. This offers a higher communication rate than a serial interface and may be used in configurations where a serial connection is not available or not convenient. This section details the required logic that must be implemented in the external device as well as the necessary connections to the CP chip.

The reference design files for the parallel interface chip, in Actel/ViewLogic format, are available from PMD. There are two versions of the design, one for interfacing with host processors that have an 8-bit data bus and one for host processors that have a 16-bit data bus. The designs are called IOPIL8 and IOPIL16 respectively. The interface to the CP chip is essentially identical in both.

The function of the I/O chip is to provide a shared-memory style interface between the host and CP chip, comprised of four 16-bit wide locations. These are used for transferring commands and data between the host and Pilot motion processor. The CP chip accesses the command/data registers using its 16-bit external data bus while the host accesses the registers via a parallel interface with chip select, read, write and command/data signals. If necessary, the host side interface can be modified by the designer to match specific requirements of the host processor.

6.1   Host interface pin description table

Pin Name	Direction	Description
HostCmd	input	This signal is asserted *high* to write a host instruction to the motion processor, or to read the status of the HostRdy and HostIntrpt signals. It is asserted *low* to read or write a data word.

HostRdy	output	This signal is used to synchronize communication between the motion processor and the host. HostRdy will go *low* (indicating host port busy) at the end of a read or write operation according to the interface mode in use, as follows:
		
		Interface Mode HostRdy goes low
		8/16 after the second byte of the instruction word
		                      after the second byte of each data word is transferred
		16/16 after the 16-bit instruction word
		                      after each 16-bit data word
		serial *n/a*
		
		HostRdy will go *high,* indicating that the host port is ready to transmit, when the last transmission has been processed. All host port communications must be made with HostRdy *high* (ready).
		A typical busy-to-ready cycle is 12.5 microseconds, but can be substantially longer, up to 100 microseconds.

~HostRead	input	When ~HostRead is *low*, a data word is read from the motion processor.
~HostWrite	input	When ~HostWrite is *low*, a data word is written to the motion processor.
~HostSlct	input	When ~HostSlct is *low*, the host port is selected for reading or writing operations.
CPIntrpt	output	I/O chip to CP chip interrupt. This signal sends an interrupt to the CP chip whenever a host–chipset transmission occurs. It should be connected to CP chip pin 53, I/OIntrpt.
CPR/~W	input	This signal is *high* when the I/O chip is reading data from the I/O chip, and *low* when it is writing data. It should be connected to CP chip pin 4, R/W.
CPStrobe	input	This signal goes *low* when the data and address become valid during Motion processor communication with peripheral devices on the data bus, such as external memory or a DAC. It should be connected to CP chip pin 6, Strobe.

Pin Name	Direction	Description
CPPeriphSlct	input	This signal goes *low* when a peripheral device on the data bus is being addressed. It should be connected to CP chip pin 130, PeriphSlct.
CPAddr0 CPAddr1 CPAddr15	input	These signals are *high* when the CP chip is communicating with the I/O chip (as distinguished from any other device on the data bus). They should be connected to CP chip pins 110 (Addr0), 111 (Addr1), and 128 (Addr15).
MasterClkIn	input	This is the master clock signal for the motion processor. It is driven at a nominal 40 MHz
CPClk	output	This signal provides the clock pulse for the CP chip. Its frequency is half that of MasterClkIn (pin 89), or 20 MHz nominal. It is connected directly to the CP chip I/Oclk signal (pin 58).
HostData0 HostData1 HostData2 HostData3 HostData4 HostData5 HostData6 HostData7 HostData8 HostData9 HostData10 HostData11 HostData12 HostData13 HostData14 HostData15	bi-directional, tri-state	These signals transmit data between the host and the Motion processor through the parallel port. Transmission is mediated by the control signals ~HostSlct, ~HostWrite, ~HostRead and HostCmd. In 16-bit mode, all 16 bits are used (HostData0-15). In 8-bit mode, only the low-order 8 bits of data are used (HostData0-7).
CPData0 CPData1 CPData2 CPData3 CPData4 CPData5 CPData6 CPData7 CPData8 CPData9 CPData10 CPData11 CPData12 CPData13 CPData14 CPData15	bi-directional	These signals transmit data between the I/O chip and pins Data0-15 of the CP chip, via the motion processor data bus.

6.2   16-bit Host Interface (IOPIL16)

This design implements a parallel interface with a host processor utilizing a 16-bit data bus. An understanding of the underlying operation of the design is only necessary if the designer intends to make modifications. In most cases this design can be implemented without changes. The following notes should be read while referencing the schematics. IOPIL16 1 is the top level schematic. The timing for the host to I/O chip communication can be found in section 4.5 and the timing for the CP to I/O chip communication can be found in section 4.7.

The description below identifies the key elements of each schematic starting with the host side signals. The paragraph title identifies the key schematic(s) being described in the text.

IOPIL16 3

The host interface is shown in sheet IOPIL16 3. The incoming data HD[15:0] is latched in the transparent latches when ~HG1 and ~HG2 go high. This would be the result of a write from the host to the CP. The latched data HI[15:8] and HI[7:0] go to schematic IOPIL16 1 and IOPIL16 5. Data from the interface to the host, HO[15:8] and HO[7:0] is enabled onto the host bus, HD[15:0], by HOES2 and HOES1 respectively. The output latches, which present the data during a host read, are always transparent because GOUT is connected to VDD. The latched I/O is an I/O option on the Actel part used and could be omitted in the host interface if a different CPLD or FPGA does not have this feature.

IOPIL16 1

The control for the host interface starts on IOPIL16 1. HOES1 and HOES2 are the AND of ~HSEL and ~HRD and enable read data onto the host bus, as previously described. HRDY is a handshaking signal to the host to allow asynchronous communication between the host and the CP. The host must wait until HRDY is true before attempting to communicate with the CP. This signal is copied as a bit in the host status register. The host status register may be read at any time to determine the state of HRDY, or the HRDY output may be used as an interrupt to the host. ~HSEL, ~HRD, ~HWR, and HA0 are the buffered inputs of the host control signals.

HOST INTERFACE/IOPIL16 5

Data from the host HI[15:8] and HI[7:0] is written into REG1 and REG2 on the schematic HOST INTERFACE by ~EN1 and ~EN2. These registers have a 2 to 1 multiplexed input with both the host data and the CP data being written to these registers. This is convenient for diagnostic purposes and is very efficient in the Actel A42MX FPGA's, which are multiplexer based but if the configuration of the logic device used demands it, separate registers could be used for the host and CP data. The schematic for this register is shown as DFME8. Only commands and checksums are written to registers REG1 and REG2 while data is written and read from the set of data registers, DATREG shown on IOPIL16 5. These 3 data registers buffer data sent to and from the CP, reducing the number of interrupts the CP must handle. The output from REG1 and REG2, CIQ[15:8] and CIQ[7:0] go to IOPIL16 5, where they are multiplexed with the other data registers. The multiplexed result, IQ[15:8] and IQ[7:0], is multiplexed with HST[15:8] and HST[7:0] - the output of the host status registers REG3 and REG4. As previously mentioned, HRDY becomes HST15 so it can be read by the host. The rest of the status register is written by the CP to provide information to the host. HA0 acts as an address bit, and usually is an address bit on the bus. When the host is writing, HA0 low indicates data and HA0 high indicates a command. When the host is reading, HAO low indicates data and HA0 high indicates status. Read status is the only transaction

allowed while HRDY is low. During a host write the AND gate (G1:HOST INTERFACE) and two flops latch the incoming data in the interface latches by driving ~HG1, and ~HG2 low from the start of the write transaction until the first negative clock transition after the first positive transition following the start of the write cycle. This tail-biting circuit removes the requirement for hold time on the data bus.

HICTLA

Most of the control logic for the host interface is shown on schematic HICTLA. The sequencer at the top generates HCYC one clock interval after the interface has been accessed and the host has finished the transaction. The nature of the transaction, rd/wr, command/data, and read status is preserved in the three flops F13, F8, and F9. A host write or a CP write, DSIW, enable REG1 and REG2 on the HOST INTERFACE schematic discussed previously. A host data write generates ~ENHD1 and ~ENHD2 for the data registers on the DATREG schematic. The logic at the bottom of the page generates the CP interrupt, the HRDY and the HCMDFL. The HCMDFL is used in the CP status to indicate a command. DSIW, the CP writing to REG1 and REG2 on the HOST INTERFACE schematic clears the interrupt and reasserts HRDY. HRDY is de-asserted during all host transactions except read status, and stays de-asserted until the CP has completed the DSIW cycle that clears the interrupt and reasserts HRDY. As mentioned previously data transfers to and from the host use the data registers and do not interrupt the CP. The CP knows the number of data transfers that must take place after decoding the command. It places this number, 0-3, in the 2 least significant bits of the host status register, HST[1:0]. These become DPNT[1:0] on this page of the schematic and enable an interrupt at 0 for a read and 1 or 0 for a write. The CP always leaves theses bit set to 0 unless setting up a multiple word data transfer. If INTEN is true and LRDST, latched read status, is false, HCYC will generate an interrupt to the CP. This will also hold HRDY false until after the CP writes to the interface register, DSIW, thereby generating ~CLRFLGS.

IOPIL16 4

The CP interface is shown in sheet IOPIL16 4. The incoming data DSD[15:0] is latched in the transparent latches when ~DG1 and ~DG2 go high. This occurs at the completion of a write from the CP to the I/O chip. The latched data DSI[15:8] and DSI[7:0] go to schematic IOPIL16 1 and IOPIL16 5. DSI[7:0] also goes to IOPIL16 2. Data from the interface to the CP, DO[15:8] and DO[7:0] is enabled onto the CP bus, DSD[15:0], by DOE2 and DOE1 respectively. The output latches, which present the data during a CP read, are always transparent because GOUT is connected to VDD. The latched I/O in the Actel part contains both input and output latches. The output latches could be omitted in the CP interface if a different CPLD or FPGA does not have this feature. The two incoming CP address bits CPA0 and CPA1 are also latched using ~DG3. The 20CK signal is the clock for the CP. This is a 20 MHz clock derived from a 40 MHz clock input.

IOPIL16 2

The CP control starts on IOPIL16 2. The I/O control is generated from ~CPSTRB, ~CPIS, CPSEL and R/W. ~DG1, ~DG2, and ~DG3 latch the incoming data and DOE1 and DOE2 out-enable the data from this chip to the CP. F2 and F4 tail-bite the write to avoid having to specify hold times on the data. Flop F1 divides the 40MHz clock down to 20 MHz. A 20 MHz clock could be used for this interface and the CP.

DSPWA

The CP write control is contained on schematic DSPWA. The CP interface uses page addressing to save I/O pins. F0, F1 and F2 make up the page register. In addition there are the 2 address bits, LA0 and LA1. A write to address 0 selects the page register with DSI[2:0] going to the page register and selecting the page for the successive transfers. A read from address 0 reads the status register on all pages. Pages 4 and 6 are the only ones implemented in this device. L1 latches the r/w level. The write decoding generates DSIW which enables writes to the DFME8 registers reg1 and reg2 shown on the HOST INTERFACE schematic. DSIW also clears the CP interrupt and restores HRDY. DSWST writes to the host status register also shown on the HOST INTERFACE schematic. DSWDREG implements writing to the data registers shown on IOPIL16 5 and DATREG. Finally the logic at the bottom of the page generates CPCYC, a 1-clock interval after the CP cycle is over that implements the actual writes to the registers. The use of the data bus latches and the post bus cycle transfers keeps as much of the logic synchronous as possible given two asynchronous devices, without requiring clocking at several times the bus speed.

DSPRA

The CP read control is contained on schematic DSPRA. The 2 by 16 bit mux selects CP status if the CP latched address is 0 and IQ[15:0] if the address is not 0. The only significant status bits are bits 15 (indicating the CP is interrupting the host), bits 13 and 14 (both 0 indicating a 16 bit host interface) and bit 0 (set to 1 during a host command transfer and 0 during data transfer).

HOST INTERFACE

Both the CP and the host use a special mode to transfer data to avoid unnecessary CP interrupts. This special mode is under the control of the CP and is transparent to the host. When the CP receives a command from the host it initializes the transfer by setting the number of transfers expected (0,1,2 or 3) in the 2 LSB's of the host status register, REG3 and REG4 on HOST INTERFACE. This write (DSWST) also loads these bits into the 2 bit down counter DCNT2 on IOPIL16 5. Note that a Q8 low, which indicates a host command, asynchronously clears this register enabling interrupts on schematic HICTLA. If DPNT[1:0] is not 0 and Q8 is high, indicating a host data transfer, and SINT goes high indicating the end of a host cycle the counter is decremented. MXAD2 selects address RA from the CP latched address bits if the page register contains 6, or the counter contents DPNT[1:0] if not. This allows the CP to have direct access to registers 1, 2, and 3, using addresses 1,2,and 3 on page 6. The host on the other hand can only read or write to the data register, HA0 low and the counter will auto decrement from 3 down to 0 allowing the host to access the registers on DATREG where REG1=R1 and R2, REG2=R3 and R4, and REG3=R5 and R6. The writes are enabled by the two decoders DECE2X4, while the reads are selected by the two 4x8 muxes, MUX1 and MUX2 controlled by the two 2x1 muxes MDS1 and MDS0. The output data IQ[15:0] goes to HOST INTERFACE schematic below IOPIL16 1 and to DSPRA below IOPIL16 2. The write data is HI[15:8], HI[7:0] from the host and DSI[15:8] and DSI[7:0] from the CP.

HINTF

HSTSEL — PAD — INBUF — IN17 — Y — HSEL → HSEL   HOST INTERFACE
HRD   (HINTRFA)
HSTRD — PAD — INBUF — IN18 — Y — HRD → HWR
HSTWR — PAD — INBUF — IN19 — Y — HWR → HA0
HADR0 — PAD — INBUF — IN20 — Y — HA0

HO[7:0] → HO[7:0]
HO[15:8] → HO[15:8]

HI[7:0] → HI[7:0]
HI[15:8] → HI[15:8]
DSI[15:8] → DSI[15:8]
DSI[7:0] → DSI[7:0]
DPNT[1:0] → HST[1:0]

CIQ[7:0] → CIQ[7:0]
CIQ[15:8] → CIQ[15:8]

Q8 → Q8

DSWST → DSWST
DSIW → DSIW

SINT → SINT

HG1 → HG1
HG2 → HG2

IQ[7:0] → IQ[7:0]
IQ[15:8] → IQ[15:8]

HST14 → ST15
HCMDFL → ST0
DSPINTR → DSPINTR

CLK → CLK

HST15 → RDY — D — OUTBUF — PAD — HRDY
ENHD1 → ENHD1
ENHD2 → ENHD2

HRD — A
HSEL — B — AND2B — Y — HOES1

HRD — A
HSEL — B — AND2B — Y — HOES2

DSPINTR — D — OUTBUF — PAD — DSPINT
OUT5

VIEWlogic™

IOPIL16 1

22 OCT 2002   DRAWN BY: DBS

DSPWA

PNT0	CSEL0
PNT1	CSEL1
DSWDREG	DSWDREG
DSIW	DSIW
DSWST	DSWST
PP6	PP6
PP4	PP4

DSI[7:0] — DSI[7:0]

DSPRA

ST0 — ST0
ST15 — ST15

IQ[15:0] — IQ[15:0]

DO[15:0] — DO[15:0]

LA0 — LA0
LA1 — LA1

DSPRA

CS PAD INBUF IN27 Y CPSEL

CPR-W PAD INBUF IN28 Y

STRB PAD INBUF IN26 Y

IS PAD INBUF IN30 Y

DG3 — DG3
LA0 — LA0
LA1 — LA1

R/W — R/W
CPSTRB — CPSTRB
CPIS — CPIS

20CK CKBUF A Y CLK
CLKINT
CLK

CPCYC — CPCYC

CLKIN PAD IB1 INBUF Y 40CK CLK

F1 D QN 20CK
DF1A

CPSTRB A G1 Y CSACC
CPIS B NAND3B
CPSEL C

A G2 Y DOE1
B AND4B
R/W C
D

A G3 Y DOE2
B AND4B
C
D

A G4 Y DG1
B NAND3B
C

A G5 Y DG2
B NAND3B
C

CSACC A G6 Y DG3
B NAND4B
CQ3 C
D

CSACC F2 D Q CQ1 F4 D Q CQ3
DF1 DF1
CLK CLK
CLK

IOPIL16 3

21 OCT 2002

DRAWN BY: DBS

Grid references: A | B | C | D (top and bottom); 1, 2, 3, 4 (left and right margins)

Column A:
DOE1
DO0 D D Q E HIGH SLEW PAD DSD0
VDD GOUT G
Q DSI0 BBDLHS GIN G
DOE1
DO1 D D Q E HIGH SLEW PAD DSD1
GOUT G
Q DSI1 BBDLHS GIN G
DOE1
DO2 D D Q E HIGH SLEW PAD DSD2
GOUT G
Q DSI2 BBDLHS GIN G
DOE1
DO3 D D Q E HIGH SLEW PAD DSD3
GOUT G
Q DSI3 BBDLHS GIN G
DO[7:0]
DOE1

Column B:
DOE1
DO4 D D Q E HIGH SLEW PAD DSD4
VDD GOUT G
Q DSI4 BBDLHS GIN G
DOE1
DO5 D D Q E HIGH SLEW PAD DSD5
GOUT G
Q DSI5 BBDLHS GIN G
DOE1
DO6 D D Q E HIGH SLEW PAD DSD6
GOUT G
Q DSI6 BBDLHS GIN G
DOE1
DO7 D D Q E HIGH SLEW PAD DSD7
GOUT G
Q DSI7 BBDLHS GIN G
DSI[7:0]
DG1
DOE2

Column C:
DOE2
DO8 D D Q E HIGH SLEW PAD DSD8
VDD GOUT G
Q DSI8 BBDLHS GIN G
DOE2
DO9 D D Q E HIGH SLEW PAD DSD9
GOUT G
Q DSI9 BBDLHS GIN G
DOE2
DO10 D D Q E HIGH SLEW PAD DSD10
GOUT G
Q DSI10 BBDLHS GIN G
DOE2
DO11 D D Q E HIGH SLEW PAD DSD11
GOUT G
Q DSI11 BBDLHS GIN G
DO[15:8]
DG2

Column D:
DOE2
DO12 D D Q E HIGH SLEW PAD DSD12
VDD GOUT G
Q DSI12 BBDLHS GIN G
DOE2
DO13 D D Q E HIGH SLEW PAD DSD13
GOUT G
Q DSI13 BBDLHS GIN G
DOE2
DO14 D D Q E HIGH SLEW PAD DSD14
GOUT G
Q DSI14 BBDLHS GIN G
DOE2
DO15 D D Q E HIGH SLEW PAD DSD15
GOUT G
Q DSI15 BBDLHS GIN G
DSI[15:8]
DG2
DOE2

Bottom section:
GND
D D Q E HIGH SLEW PAD CPA0
GOUT G
D Q Q LA0 BBDLHS
DG3 GIN G

GND
D D Q E HIGH SLEW PAD CPA1
GOUT G
D Q Q LA1 BBDLHS
DG3 GIN G

VCC Y VDD
DG2

20CK D OUTBUF PAD CLKOUT

VIEWlogic™

IOPIL16 4

21 OCT 2002   DRAWN BY: DBS

DREG

DOE1 — DOE1
PP4 — PP4
CIQ[7:0] — CIQ[7:0]
CIQ[15:8] — CIQ[15:8]
HI[7:0] — HI[7:0]
HI[15:8] — HIH[15:8]
DSI[7:0] — DSI[7:0]
DSI[15:8] — DSI[15:8]
RA[1:0] — RA[1:0]
LA[1:0] — LA[1:0]
PP6 — PP6
END1 — END1
END2 — END2
CLK — CLK

IQ[7:0]
IQ[15:8]

IQ[7:0]
IQ[15:8]

DATREG

ENHD1 — A OR2A Y — END1
DSWDREG — B
ENHD2 — A OR2A Y — END2
B

Q8
SINT — A
— B AND3 Y — DPINC
DPNT0 — A NAND2B Y — C
DPNT1 — B

DCNT2

DSWST — SLOAD
DPINC — ENABLE
Q8 — ACLR
CLK — CLOCK
DSI[1:0] — DATA[1:0] Q[1:0]

DPNT[1:0]

MXAD2

DATA0_[1:0]
RESULT[1:0] — RA[1:0]
DATA1_[1:0]

LA[1:0]

LA0
LA1

SEL0

PP6

VIEWlogic™

IOPIL16 5

22 OCT 2002   DRAWN BY: DBS

HOST INTERFACE
(HINTRFA)

24 OCT 2002

DRAWN BY: DBS

VIEWlogic™

HICTLA

21 OCT 2002   DBS

DRAWN BY:

DSI[7:0]

DSI0 — F0 Q — PNT0 — A / AND3B / Y — PP4
F0 / DFE1B / CE / CLK — B / C

DSI1 — F1 Q — PNT1 — A / AND3A / Y — PP6
F1 / DFE1B / CE / CLK — B / C

DSWPNT — F2 Q — PNT2
CLK — F2 / DFE1B / CE / CLK

DEC2
LA0 — A
LR/W — E
LA1 — B
DECE2X4D
Y0 — CPCYC — A / G2 / NAND2 / Y — DSWPNT
Y1 — ADW0 — B
Y2 — ADW2
Y3 — ADW3

L1
R/W — D Q — LR/W
DG3 — G
DL1B

G11
CPCYC1 — A
PP4 — B / AND3 / Y — DSIW
ADW2 — C
G12
CPCYC1 — A
PP4 — B / AND3 / Y — DSWST
ADW3 — C

ADW0 — A
LR/W — B
CPCYC1 — C / AND4B / Y — DSWDREG
PP6 — D

VCC — Y

F4
Q3 — E
D Q — Q2
DFE3A / CLK / CLR

A — Y — Q2

A — Y

F5
GND — D0 Q — Q3
D1
D2
D3
CPS — S0
S1
DFM6A / CLK
CLR
VCC — Y

BUF2
A — Y — CPCYC

BUF3
A — Y — CPCYC1

G6
CPIS — A
CPSTRB — B / AND3B / Y — CPS
CPSEL — C
CLK

Q3

VIEWlogic™

DSPWA

24 OCT 2002   DRAWN BY: DBS

ST0　A　Y　CH0

ST15　A　Y　CH15

MUX2X16

CH15,GND,GND,GND[12:1],CH0

DATA0_[15:0]

IQ[15:0]

DATA1_[15:0]

RESULT[15:0]

DO[15:0]

A　Y　GND[12:1]

$ARRAY=12

Y
GND

SEL0

LA0　A
LA1　B　OR2　Y　IQSEL

VIEW/logic™

DSPRA

24 OCT 2002

DRAWN BY:　DBS

A[7:0]

B[7:0]

B0 A0   B1 A1   B2 A2   B3 A3   B4 A4   B5 A5   B6 A6   B7 A7

S

EN1

CK

F0   F1   F2   F3   F4   F5   F6   F7

CLK S B A DFME16A Q   (×8)

Q0   Q1   Q2   Q3   Q4   Q5   Q6   Q7

Q[7:0]

VIEWlogic™

DFME8

19 NOV. 2002   DRAWN BY: DBS

R1

EN1R1 — EN1
DSPSEL — S — DFME8
HI[7:0] — CK
DSI[7:0] — A[7:0] — Q[7:0] — R1[7:0]
B[7:0]

R2

EN2R1 — EN1
S — DFME8
CLK — CK
HIH[15:8] — A[7:0] — Q[7:0] — R1[15:8]
DSI[15:8] — B[7:0]

R3

EN1R2 — EN1
DSPSEL1 — S — DFME8
HI[7:0] — CK
DSI[7:0] — A[7:0] — Q[7:0] — R2[7:0]
B[7:0]

R4

EN2R2 — EN1
S — DFME8
CLK — CK
HIH[15:8] — A[7:0] — Q[7:0] — R2[15:8]
DSI[15:8] — B[7:0]

R5

EN1R3 — EN1
DSPSEL2 — S — DFME8
HI[7:0] — CK
DSI[7:0] — A[7:0] — Q[7:0] — R3[7:0]
B[7:0]

R6

EN2R3 — EN1
S — DFME8
CLK — CK
HIH[15:8] — A[7:0] — Q[7:0] — R3[15:8]
DSI[15:8] — B[7:0]

MUX1 — MUX4X8

CIQ[7:0]
R1[7:0] — DATA0_[7:0]
R2[7:0] — DATA1_[7:0] — RESULT[7:0] — IQ[7:0]
R3[7:0] — DATA2_[7:0]
DATA3_[7:0]
SEL1   SEL0
MDS1   MDS0

MUX2 — MUX4X8

CIQ[15:8]
R1[15:8] — DATA0_[7:0]
R2[15:8] — DATA1_[7:0] — RESULT[7:0] — IQ[15:8]
R3[15:8] — DATA2_[7:0]
DATA3_[7:0]
SEL1   SEL0
MDS1   MDS0

LA0 — A
LA1 — B
PP4 — C — AND4A — Y — DSIR
DOE1 — D

RA0 — A — S
MX2 — Y — MDS0
GND — B

RA1 — A — S
MX2 — Y — MDS1
GND — B

Y
GND

DEC1 — DECE2X4

DATA0 — EQ0
DATA1 — EQ1 — EN1R1
EQ2 — EN1R2
EQ3 — EN1R3
END1 — ENABLE

DEC2 — DECE2X4

DATA0 — EQ0
DATA1 — EQ1 — EN2R1
EQ2 — EN2R2
EQ3 — EN2R3
END2 — ENABLE

LA[1:0]

PP6 — A — B1 — Y — DSPSEL
A — B2 — Y — DSPSEL1
A — B3 — Y — DSPSEL2

RA[1:0]
RA0
RA1

VIEW*logic*™

DATREG

24 OCT 2002   DRAWN BY: DBS

6.3   8-bit Host Interface (IOPIL8)

This design implements a parallel interface with a host processor utilizing an 8-bit data bus. An understanding of the underlying operation of the design is only necessary if the designer intends to make modifications. In most cases this design can be implemented without changes. The following notes should be read while referencing the schematics. IOPIL16 1 is the top level schematic. The timing for the host to I/O chip communication can be found in section 4.4 and the timing for the CP to I/O chip communication can be found in section 4.7.

The description below identifies the key elements of each schematic starting with the host side signals. The paragraph title identifies the key schematic(s) being described in the text.

IOPIL8 3

The host interface for IOPIL8 is shown in sheet IOPIL8 3. The incoming data HD[7:0] is latched in the transparent latches when ~HG1 goes high. This would be a write from the host to the CP. The latched data HI[7:0] goes to IOPIL8 1 and IOPIL8 5. Data from the interface to the host, HO[7:0] is enabled onto the host bus, HD[7:0], by HOES1. The output latches, which present the data during a host read, are always transparent because GOUT is connected to VDD. The latched I/O is an I/O option on the Actel part used and could be omitted in the host interface if a different CPLD or FPGA does not have this feature. HD[15:8] are tri-stated outputs because Actel grounds unused I/O pins and this would interfere with using existing PMD test equipment. These reserved I/O's can be ommitted in a different implementation with an 8 bit bus.

IOPIL8 1

The control for the host interface starts on IOPIL8 1. HOES1 is the AND of ~HSEL and ~HRD, and enable read data onto the host bus, as previously described. HRDY is a handshaking signal to the host to allow asynchronous communication between the host and the CP. The host must wait until HRDY is true before attempting to communicate with the CP. This signal is copied as a bit in the host status register. The host status register may be read at any time to determine the state of HRDY, or the HRDY output may be used as an interrupt to the host. ~HSEL, ~HRD, ~HWR, and HA0 are the buffered inputs of the host control signals.

HOST INTERFACE/IOPIL8 5

Data from the host HI[7:0] is written into REG1 and REG2 on the schematic HOST INTERFACE by ~EN1 and ~EN2. All transfers are 16 bits and take two writes or reads on the 8-bit bus. These registers have a 2 to 1 multiplexed input with both the host data and the CP data being written to this register.

This is convenient for diagnostic purposes and is very efficient in the Actel A42MX FPGA's, which are multiplexer based but if the configuration of the logic device used demands it, separate registers could be used for the host and CP data. The schematic for this register is shown as DFME8. Only commands and checksums are written to registers REG1 and REG2 while data is written and read from the set of data registers, DATREG shown on IOPIL8 5. These 3 data registers buffer data sent to and from the CP, reducing the number of interrupts the CP must handle. The output from REG1 and REG2, CIQ[15:8] and CIQ[7:0] go to IOPIL8 5, where they are multiplexed with the other data registers. The multiplexed result, IQ[15:8] and IQ[7:0], is multiplexed with HST[15:8] and HST[7:0] - the output of the host status registers REG3 and REG4. This four input mux, MUX4X8, also muxes the 16 bit data onto the 8-bit bus. As previously mentioned HRDY becomes HST15 so it can be read by the host. The rest of the status register is written by the CP to provide information to the

host. HA0 acts as an address bit, and usually is an address bit on the bus. When the host is writing, HA0 low indicates data and HA0 high indicates a command. When the host is reading, HAO low indicates data and HA0 high indicates status. Read status is the only transaction allowed while HRDY is low. During a host write the AND gate (G1:HOST INTERFACE) and two flops latch the incoming data in the interface latches by driving ~HG1 low from the start of the write transaction until the first negative clock transition after the first positive transition following the start of the write cycle. This tail-biting circuit removes the requirement for hold time on the data bus.

HICTLA

Most of the control logic for the host interface is shown on schematic HICTLA. The sequencer at the top generates HCYC one clock interval after the interface has been accessed and the host has finished the transaction. The nature of the transaction, rd/wr, command/data, and read status is preserved in the three flops F13, F8, and F9. Since 16 bit transfers must take place over an 8 bit bus two transfers are required. The toggle flop is used to determine whether a cycle is the first or second of the 2 required. The toggle flop may be initialized to the 0 state, which indicates that this is the first transfer (high byte), by the CP writing a one to host status bit 15. This status bit is read by the host as the HRDY bit and is not writable by the CP. In addition flop F12 and the associated gating determine if the present command transaction is the first or second byte of a command. If the toggle flop gets into the wrong state due to a missed or aborted transfer the next command will set it back to the correct state. A host write or a CP write, DSIW, enable REG1 and REG2 on the HOST INTERFACE schematic discussed previously. A host data write generates ~ENHD1 and ~ENHD2 for the data registers on the DATAREG schematic. For host writes ~EN2, ~EN1, ~ENHD2, and ~ENHD1 are also determined by the state of the toggle flop using HIEN and LOEN. 1CMD is used in this logic to ensure correct behavior when the command is correcting the state of the toggle. The logic at the bottom of the page generates the CP interrupt, the HRDY and the HCMDFL. The HCMDFL is used in the CP status to indicate a command. DSIW, the CP writing to REG1 and REG2 on the HOST INTERFACE schematic clears the interrupt and reasserts HRDY. HRDY is de-asserted during all host transactions except read status, and stays de-asserted until the CP has completed the DSIW cycle that clears the interrupt and reasserts HRDY. As mentioned previously data transfers to and from the host use the data registers and do not interrupt the CP. The CP knows the number of data transfers that must take place after decoding the command. It places this number, 0-3, in the 2 least significant bits of the host status register, HST[1:0]. These become DPNT[1:0] on this page of the schematic and enable an interrupt at 0 for a read and 1 or 0 for a write. The CP always leaves these bits at 0 unless setting up a multiple word data transfer. If INTEN is true and LRDST, latched read status, is false, HCYC will generate an interrupt to the CP. This will also hold HRDY false until after the CP writes to the interface register, DSIW, thereby generating ~CLRFLGS.

IOPIL8 4

The CP interface is shown in sheet IOPIL8 4. The incoming data DSD[15:0] is latched in the transparent latches when ~DG1 and ~DG2 go high. This occurs at the completion of a write from the CP to the I/O chip. The latched data DSI[15:8] and DSI[7:0] go to schematic IOPIL8 1 and IOPIL16 5. DSI[7:0] also goes to IOPIL16 2. Data from the interface to the CP, DO[15:8] and DO[7:0] is enabled onto the CP bus, DSD[15:0], by DOE2 and DOE1 respectively. The output latches, which present the data during a CP read, are always transparent because GOUT is connected to VDD. The latched I/O in the Actel part contains both input and output latches. The output latches could be omitted in the CP interface if a different CPLD or FPGA does not have this feature. The two incoming CP address bits CPA0 and CPA1 are also latched using ~DG3. The 20CK signal is the clock for the CP. This is a 20 MHz clock derived from a 40 MHz clock input.

IOPIL8 2

The CP control starts on IOPIL8 2. The I/O control is generated from ~CPSTRB, ~CPIS, CPSEL and R/W. ~DG1, ~DG2, and ~DG3 latch the incoming data and DOE1 and DOE2 out-enable the data from this chip to the CP. F2 and F4 tail-bite the write to avoid having to specify hold times on the data. Flop F1 divides the 40MHz clock down to 20 MHz. A 20 MHz clock could be used for this interface and the CP.
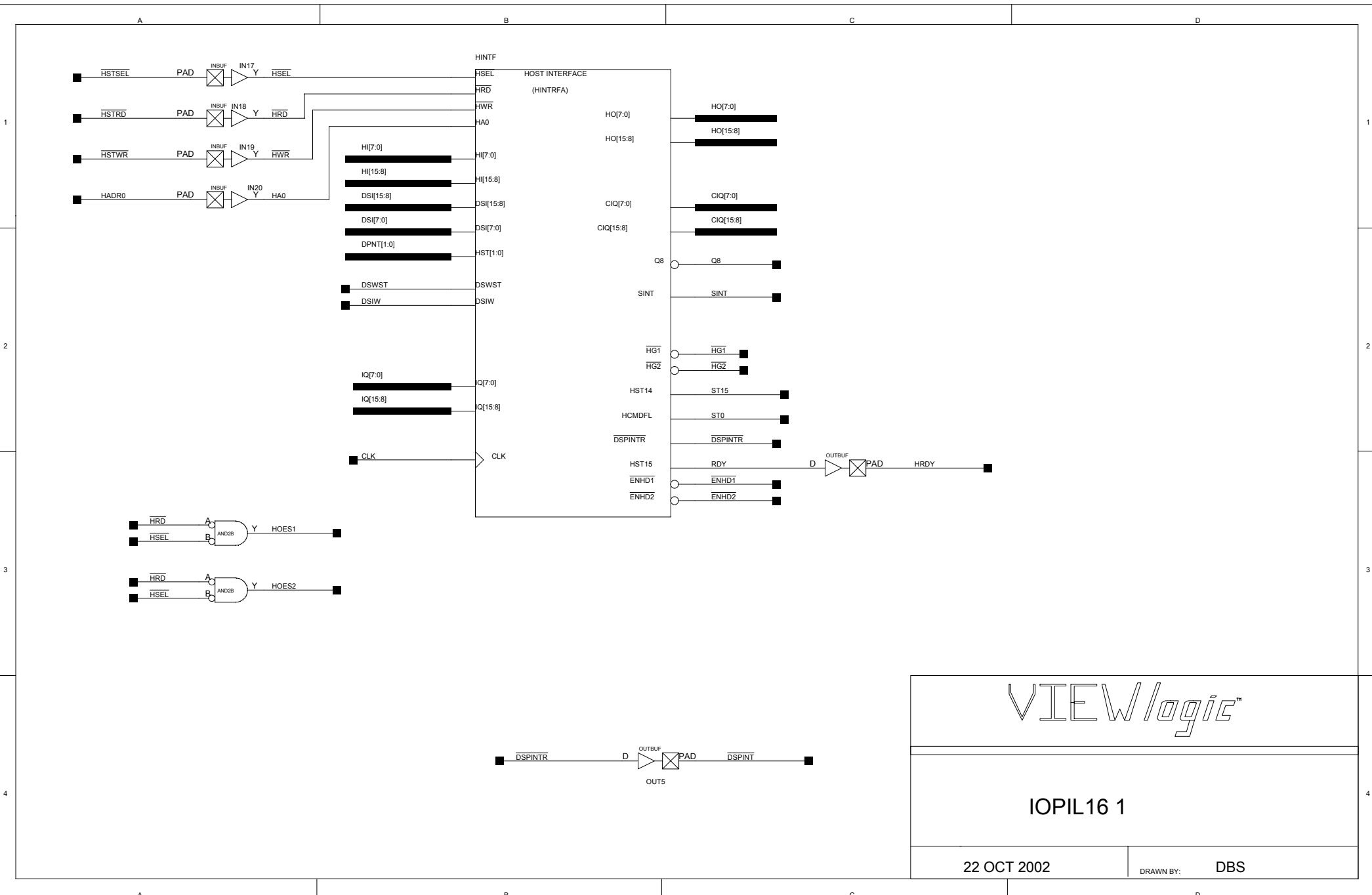
DSPWA

The CP write control is contained on schematic DSPWA. The CP interface uses page addressing to save I/O pins. F0, F1 and F2 make up the page register. In addition there are the 2 address bits, LA0 and LA1. A write to address 0 selects the page register with DSI[2:0] going to the page register and selecting the page for the successive transfers. A read from address 0 reads the status register on all pages. Pages 4 and 6 are the only ones implemented in this device. L1 latches the r/w level. The write decoding generates DSIW which enables writes to the DFME8 registers reg1 and reg2 shown on the HOST INTERFACE schematic. DSIW also clears the CP interrupt and restores HRDY. DSWST writes to the host status register also shown on the HOST INTERFACE schematic. DSWDREG implements writing to the data registers shown on IOPIL8 5 and DATREG. Finally the logic at the bottom of the page generates CPCYC, a 1-clock interval after the CP cycle is over that implements the actual writes to the registers. The use of the data bus latches and the post bus cycle transfers keeps as much of the logic synchronous as possible given two asynchronous devices, without requiring clocking at several times the bus speed.

DSPRA

The CP read control is contained on schematic DSPRA. The 2 by 16 bit mux selects CP status if the CP latched address is 0 and IQ[15:0] if the address is not 0. The only significant status bits are bits 15 (indicating the CP is interrupting the host), bit 14 (1 indicating an 8-bit host interface) and bit 0 (set to 1 during a host command transfer and 0 during data transfer).

HOST INTERFACE

Both the CP and the host use a special mode to transfer data to avoid unnecessary CP interrupts. This special mode is under the control of the CP and is transparent to the host. When the CP receives a command from the host it initializes the transfer by setting the number of transfers expected (0,1,2 or 3) in the 2 LSB's of the host status register, REG3 and REG4 on HOST INTERFACE. This write (DSWST) also loads these bits into the 2 bit down counter DCNT2 on IOPIL8 5. Note that a Q8 low, which indicates a host command, asynchronously clears this register enabling interrupts on schematic HICTLA. If DPNT[1:0] is not 0 and Q8 is high, indicating a host data transfer, and SINT goes high indicating the end of a host cycle the counter is decremented. MXAD2 selects address RA from the CP latched address bits if the page register contains 6, or the counter contents DPNT[1:0] if not. This allows the CP to have direct access to registers 1, 2, and 3, using address 1,2,and 3 on page 6. The host on the other hand can only read or write to the data register, HA0 low and the counter will auto decrement from 3 down to 0 allowing the host to access the registers on DATAREG where REG1=R1 and R2, REG2=R3 and R4, and REG3=R5 and R6. The writes are enabled by the two decoders DECE2X4 while the reads are selected by the two 4x8 muxes, MUX1 and MUX2 controlled by the two 2x1 muxes MDS1 and MDS0. The output data IQ[15:0] goes to HOST INTERFACE schematic below IOPIL8 1 and to DSPRA below IOPIL8 2. The write data is HI[7:0] from the host and DSI[15:8] and DSI[7:0] from the CP. Note that END1
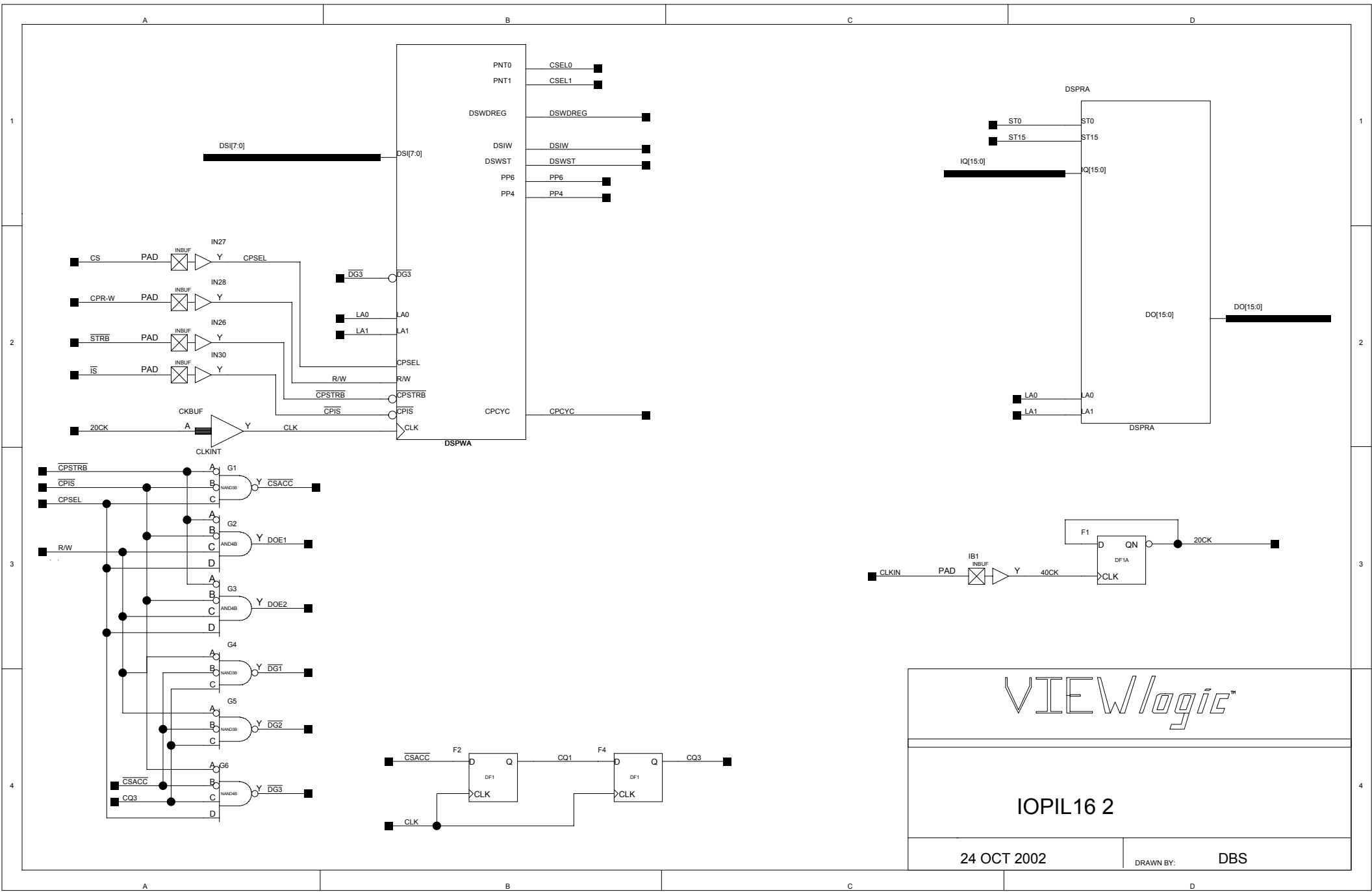
and END2, the write enables, are both high for DSWDREG, while they are high one at a time for host writes controlled by the toggle flop. SINT enables DPINC only when the toggle is high after the second transfer.

HINTF

HOST INTERFACE
(HINTRFA)

Input Pin	PAD	Buffer	Gate	Y	Signal
HSTSEL	PAD	INBUF	IN17	Y	HSEL
HSTRD	PAD	INBUF	IN18	Y	HRD
HSTWR	PAD	INBUF	IN19	Y	HWR
HADR0	PAD	INBUF	IN20	Y	HA0

Left side of HINTF block inputs:
- HSEL
- HRD
- HWR
- HA0
- HI[7:0]
- DSI[15:8]
- DSI[7:0]
- HST[1:0] (from DPNT[1:0])
- DSWST
- DSIW (DSIW)
- IQ[7:0]
- IQ[15:8]
- CLK

Right side of HINTF block outputs:
- HO[7:0]
- CIQ[7:0]
- CIQ[15:8]
- Q8
- SINT
- HG1
- HST14 → ST15
- HCMDFL → ST0
- DSPINTR → DSPINTR
- HST15 → RDY → OUTBUF → PAD → HRDY
- ENHD1
- ENHD2

AND2B gate:
- A: HRD
- B: HSEL
- Y: HOES1

DSPINTR → D → OUTBUF (OUT5) → PAD → DSPINT

VIEWlogic™

IOPIL8 1

22 OCT 2002

DRAWN BY: DBS

A | B | C | D

PNT0 — CSEL0
PNT1 — CSEL1
DSWDREG — DSWDREG
DSIW — DSIW
DSWST — DSWST
PP4 — PP4
PP6 — PP6

DSPRA

ST0 — ST0
ST15 — ST15
IQ[15:0] — IQ[15:0]

DSI[7:0] — DSI[7:0]

DO[15:0] — DO[15:0]

CS — PAD — INBUF — IN27 — Y
CPR-W — PAD — INBUF — IN28 — Y
STRB — PAD — INBUF — IN26 — Y
IS — PAD — INBUF — IN30 — Y

DG3 — DG3
LA0 — LA0
LA1 — LA1
CPSEL — CPSEL
R/W — R/W
CPSTRB — CPSTRB
CPIS — CPIS

LA0 — LA0
LA1 — LA1

DSPRA

20CK — A — CKBUF — Y — CLK — CLK
CLKINT
DSPWA

CPCYC — CPCYC

CPSTRB — A — G1 — NAND3B — Y — CSACC
CPIS — B
CPSEL — C

A — G2 — AND4B — Y — DOE1
R/W — B
C
D

A — G3 — AND4B — Y — DOE2
B
C
D

CLKIN — PAD — IB1 — INBUF — Y — 40CK — CLK — DF1A — D — QN — 20CK
F1

A — G4 — NAND3B — Y — DG1
B
C

A — G5 — NAND3B — Y — DG2
B
C

CSACC — F2 — DF1 — D — Q — CQ1 — F4 — DF1 — D — Q — CQ3
CLK — CLK — CLK

CSACC — A — G6 — NAND4B — Y — DG3
CQ3 — B
C
D

VIEWlogic™

IOPIL8 2

30 OCT 2002 | DRAWN BY: DBS

A

1

HO0   D
HIGH SLEW
D   Q
E
PAD   HD0
GOUT
G

VDD

Q   HI0
BBDLHS
GIN
G

VDD

HO1   D
HIGH SLEW
D   Q
E
PAD   HD1
GOUT
G

Q   HI1
BBDLHS
GIN
G

2

HO2   D
HIGH SLEW
D   Q
E
PAD   HD2
GOUT
G

Q   HI2
BBDLHS
GIN
G

HO3   D
HIGH SLEW
D   Q
E
PAD   HD3
GOUT
G

3

Q   HI3
BBDLHS
GIN
G

$\overline{HG1}$

HO[7:0]

HOES1

B

HO4   D
HIGH SLEW
D   Q
E
PAD   HD4
GOUT
G

VDD

Q   HI4
BBDLHS
GIN
G

HO5   D
HIGH SLEW
D   Q
E
PAD   HD5
GOUT
G

Q   HI5
BBDLHS
GIN
G

HO6   D
HIGH SLEW
D   Q
E
PAD   HD6
GOUT
G

Q   HI6
BBDLHS
GIN
G

HO7   D
HIGH SLEW
D   Q
E
PAD   HD7
GOUT
G

Q   HI7
BBDLHS
GIN
G

HI[7:0]

$\overline{HG1}$

VCC
Y
VDD

C

D   E
PAD   HD8
TRIBUFF

D   E
PAD   HD9
TRIBUFF

D   E
PAD   HD10
TRIBUFF

D   E
PAD   HD11
TRIBUFF

D   E
PAD   HD12
TRIBUFF

D   E
PAD   HD13
TRIBUFF

D   E
PAD   HD14
TRIBUFF

D   E
PAD   HD15
TRIBUFF

Y
GND

HI BYTE TRISTATE TO

AVOID LOADING 16 BIT BUSSES

D

VIEWlogic™

IOPIL8 3

24 OCT 2002   DRAWN BY: DBS

A | B | C | D

DOE1
DO0 D — D Q — HIGH SLEW — E — PAD — DSD0
VDD — GOUT — G
Q — DSI0
GIN — G — BBDLHS

DOE1
DO1 D — D Q — HIGH SLEW — E — PAD — DSD1
GOUT — G
Q — DSI1
GIN — G — BBDLHS

DOE1
DO2 D — D Q — HIGH SLEW — E — PAD — DSD2
GOUT — G
Q — DSI2
GIN — G — BBDLHS

DOE1
DO3 D — D Q — HIGH SLEW — E — PAD — DSD3
GOUT — G
Q — DSI3
GIN — G — BBDLHS

DO[7:0]
DOE1

DOE1
DO4 D — D Q — HIGH SLEW — E — PAD — DSD4
VDD — GOUT — G
Q — DSI4
GIN — G — BBDLHS

DOE1
DO5 D — D Q — HIGH SLEW — E — PAD — DSD5
GOUT — G
Q — DSI5
GIN — G — BBDLHS

DOE1
DO6 D — D Q — HIGH SLEW — E — PAD — DSD6
GOUT — G
Q — DSI6
GIN — G — BBDLHS

DOE1
DO7 D — D Q — HIGH SLEW — E — PAD — DSD7
GOUT — G
Q — DSI7
GIN — G — BBDLHS
DSI[7:0]

$\overline{DG1}$
DOE2

DOE2
DO8 D — D Q — HIGH SLEW — E — PAD — DSD8
VDD — GOUT — G
Q — DSI8
GIN — G — BBDLHS

DOE2
DO9 D — D Q — HIGH SLEW — E — PAD — DSD9
GOUT — G
Q — DSI9
GIN — G — BBDLHS

DOE2
DO10 D — D Q — HIGH SLEW — E — PAD — DSD10
GOUT — G
Q — DSI10
GIN — G — BBDLHS

DOE2
DO11 D — D Q — HIGH SLEW — E — PAD — DSD11
GOUT — G
Q — DSI11
GIN — G — BBDLHS

DO[15:8]
$\overline{DG2}$

DOE2
DO12 D — D Q — HIGH SLEW — E — PAD — DSD12
VDD — GOUT — G
Q — DSI12
GIN — G — BBDLHS

DOE2
DO13 D — D Q — HIGH SLEW — E — PAD — DSD13
GOUT — G
Q — DSI13
GIN — G — BBDLHS

DOE2
DO14 D — D Q — HIGH SLEW — E — PAD — DSD14
GOUT — G
Q — DSI14
GIN — G — BBDLHS

DOE2
DO15 D — D Q — HIGH SLEW — E — PAD — DSD15
GOUT — G
Q — DSI15
GIN — G — BBDLHS
DSI[15:8]

$\overline{DG2}$
DOE2

GND
D — D Q — HIGH SLEW — E — PAD — CPA0
GOUT — G
Q — LA0
$\overline{DG3}$ — GIN — G — BBDLHS

GND
D — D Q — HIGH SLEW — E — PAD — CPA1
GOUT — G
D Q — Q — LA1
$\overline{DG3}$ — GIN — G — BBDLHS

VCC
Y
VDD

20CK — D — OUTBUF — PAD — CLKOUT

VIEWlogic™

IOPIL8 4

22 OCT 2002 | DRAWN BY: | DBS

ENHD1 A
DSWDREG B OR2A Y END1

ENHD2 A
B OR2A Y END2

DREG
DOE1 DOE1
PP4 PP4
CIQ[7:0] CIQ[7:0]
CIQ[15:8] CIQ[15:8]
HI[7:0] HI[7:0]
IQ[7:0] IQ[7:0]
IQ[15:8] IQ[15:8]
DSI[7:0] DSI[7:0]
DSI[15:8] DSI[15:8]
RA[1:0] RA[1:0]
LA[1:0] LA[1:0]
PP6 PP6
END1 END1
END2 END2
CLK CLK DATREG

Q8
SINT A
B AND3 Y DPINC
DPNT0 A
DPNT1 B NAND2B Y C

DCNT2
DSWST SLOAD
DPINC ENABLE
Q8 ACLR
CLK CLOCK
DPNT[1:0]
MXAD2
DSI[1:0] DATA[1:0] Q[1:0]
DATA0_[1:0]
RESULT[1:0] RA[1:0]
DATA1_[1:0]
LA[1:0]
LA0
LA1
SEL0
PP6

VIEWlogic™

IOPIL8 5

22 OCT 2002 DRAWN BY: DBS

REG1

$\overline{EN1}$

EN1

S

CK

DFME8

A[7:0]

B[7:0]

Q[7:0]

CIQ[7:0]

HI[7:0]

DSI[7:0]

BUF1

A   Y   DSL

DSIW

BUF2

A   Y   DSLA

CLK

REG2

$\overline{EN2}$

EN1

S

CK

DFME8

A[7:0]

B[7:0]

Q[7:0]

CIQ[15:8]

HI[7:0]

DSI[15:8]

REG3

REG6

DSWST

CLK

DSI[7:2]

ENABLE

CLOCK

DATA[5:0]

Q[5:0]

REG4

REG7

DSWST

VDD

CLK

DSI[14:8]

ENABLE

ACLR

CLOCK

DATA[6:0]

Q[6:0]

HST[1:0]

IQ[7:0]

HST[7:0]

IQ[15:8]

HST[15:8]

HST[7:2]

HST[14:8]

HST14

MUX4X8

DATA0_[7:0]

DATA1_[7:0]

DATA2_[7:0]

DATA3_[7:0]

RESULT[7:0]

HO[7:0]

SEL1

SEL0

TOGGLE A   Y

HA0

TOGGLE LO

SELECTS [15:8], HI BYTE FIRST

DSWST   A

DSI15   B

NAND2

Y $\overline{RSTOG}$

HICTLA

HST[1:0]

$\overline{RSTOG}$

$\overline{HSEL}$

$\overline{HWR}$

$\overline{HRD}$

HA0

DSIW

CLK

DPNT[1:0]

RSTOG

HSEL

HWR

HRD

HA0

DSIW

CK

$\overline{ENHD1}$

$\overline{ENHD2}$

SINT

Q8

TOGGLE

$\overline{EN1}$

$\overline{EN2}$

HRDY

$\overline{DSPINTR}$

HCMDFL

$\overline{ENHD1}$

$\overline{ENHD2}$

SINT

Q8

TOGGLE

$\overline{EN1}$

$\overline{EN2}$

HST15

$\overline{DSPINTR}$

HCMDFL

HST[15:8]

HICTLA

G2

A

B

NAND2

Y   HG1

$\overline{HWR}$

$\overline{HSEL}$

G1

A

B

AND2B

Y

D   Q

DF1

CLK

D   QN

DF1C

CLK

CLK

VIEWlogic™

HOST INTERFACE
(HINTRFA)

24 AUG 2001

DRAWN BY: DBS

A B C D

HRD̄ INV1 A Y HRD
HWR̄ A
HRD̄ B G1 Y
HSEL̄ C OA1C
HWR̄ A G2
HSEL̄ B AND2B Y HWR
CK

F1 DF1 D Q Q1
CLK

F2 DF1 D Q Q2
CLK

F3 GND D0 D1 D2 D3 Q HCYC
DFM6A S0 S1 CLK CLR
Q1
HCYC
VCC Y

F13 HCYC CK VDD JKF2C J Q CLK K CLR LWR̄

HSEL̄ A HWR̄ B AND2B Y SHWR

F8 HCYC CK VDD JKF2C J Q8 A INV3 Y HCMD CLK K CLR

HSEL̄ A G7 HWR̄ B AND3B Y SHCMD HA0 C

F9 HCYC CK VDD JKF2C J Q9 A INV4 Y LRDST CLK K CLR

HSEL̄ A G10 HRD̄ B AND3B Y SLRDST HA0 C

F4 HCYC E DFE3A D Q CK CLK CLR RSTOG
1CMD A TOGGLE B NAND2A Y

TOGGLE LO (1ST BYTE) LD HI, RD HI

LCMD A AND2A Y 1CMD B

F12 Q8 D Q HCYC E DFE CK CLK

INV2 A Y DSPINTR̄

TOGGLE A OR2A Y 1CMD B
HWR A G12 HIEN B AOI2 Y EN2̄ DSIW C

1CMD A AND2A Y TOGGLE B
HWR A G14 LOEN B AOI1 Y EN1̄ DSIW C

HIEN A HWR B NAND3 Y ENHD2̄ Q8 C
LOEN A B NAND3 Y ENHD1̄ C

F5 DF1 D Q CLK
F10 DF1 D Q CLK
CK

DPNT[1:0]
DPNT0 A DPNT1 B NAND3B Y LWR̄ C
A NAND2B Y B
RDEN̄ A WREN̄ B OR3C Y INTEN Q8 C

SINT A SINTR̄ B AND3 Y CLRFLGS̄ C

F6 JKF J Q DSPINTR CLK K

LRDST A HCYC B AND3A Y ENINTR C

1CMD A AND2A Y TOGGLE B

Q1 A Q2 B HCYC C OA4 Y EBSY Q9 D

G21 A B C NOR4 Y HRDY D

HCYC A G19 HCMD B AND2 Y HCCYC

F7 JKF J Q HCMDFL CLK K
CK

DSIW A Y CLRFLGS̄

VIEWlogic™

22 OCT 2002   DRAWN BY: DBS

HICTLA

DSI[7:0]

DSI0 — F0 — Q — PNT0 — A — AND3B — Y — PP4
D
E DFE1B
CLK

DSI1 — F1 — Q — PNT1 — A — AND3A — Y — PP6
D
E DFE1B
CLK
B
C

DSI2 — F2 — Q — PNT2
D
DSWPNT E DFE1B
CLK B
CLK

DEC2
LA0 — A — Y0 — ADW0 — CPCYC — G2 — A — NAND2 — Y — DSWPNT
LR/W — E DECE2X4D — Y1 — B
LA1 — B — Y2 — ADW2
Y3 — ADW3

L1
R/W — D — Q — LR/W
DL1B
DG3 — G

CPCYC1 — G11 — A
PP4 — B — AND3 — Y — DSIW
ADW2 — C

CPCYC1 — G12 — A
PP4 — B — AND3 — Y — DSWST
ADW3 — C

ADW0 — A
LR/W — B — AND4B — Y — DSWDREG
CPCYC1 — C
PP6 — D

VCC
Y

F4
D — Q — Q2
Q3 — E DFE3A
CLK
CLR
A — Y — Q2

A — Y

F5
GND — D0 — Q — Q3 — BUF2 — A — Y — CPCYC
D1
D2
D3 DFM6A
CPS — S0 BUF3
S1 A — Y — CPCYC1
CLK
CLR
Q3 — VCC
Y

CPIS — G6 — A
CPSTRB — B — AND3B — Y — CPS
CPSEL — C
CLK

ST0　A　　Y　CH0

ST15　A　　Y　CH15

VCC
○
Y
VDD

CH15,VDD,GND,GND[12:1],CH0

MUX2X16

A　Y　GND[12:1]
$ARRAY=12

GND

DATA0_[15:0]
DATA1_[15:0]

IQ[15:0]

RESULT[15:0]

DO[15:0]

Y
GND

LA0　A
LA1　B　OR2　Y　IQSEL

SELD

VIEWlogic™

DSPRA

30 OCT 2002　　DRAWN BY:　DBS

A[7:0]

B[7:0]

B0 A0   B1 A1   B2 A2   B3 A3   B4 A4   B5 A5   B6 A6   B7 A7

S

EN1

CK

F0   F1   F2   F3   F4   F5   F6   F7

CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A   CLK DFME1/4A

Q0   Q1   Q2   Q3   Q4   Q5   Q6   Q7

Q[7:0]

VIEW/logic™

DFME8

19 NOV. 2002   DRAWN BY: DBS

DATREG

30 OCT 2002

DRAWN BY: DBS

7 Application Notes

7.1   Design Tips

The following are recommendations for the design of circuits that utilize a PMD Motion Processor.

Serial Interface

If the serial configuration decode logic is not implemented (see section 7.2) the CP data bus should be tied high. This places the serial interface in a default configuration of 9600,n,8,1 after power on or reset.

Controlling pulse output during reset

When the motion processor is in a reset state (when the reset line is held low) or immediately after a power on, the pulse outputs can be in an unknown state, causing undesirable motor movement. It is recommended that the enable line of any motor amplifier be held in a disabled state by the host processor or some logic circuitry until communication to the motion processor is established. This can be in the form of a delay circuit on the amplifier enable line after power up, or the enable line can be ANDed with the CP reset line.

Parallel word encoder input

When using parallel word input for motor position, it is useful to also decode this information into the User I/O space. This allows the current input value to be read using the chip instruction ReadIO for diagnostic purposes.

Using a non standard system clock frequency

It is often desirable to share a common clock among several components in a design. In the case of the PMD Motion Processors it is possible to use a clock below the standard value of 20MHz. In this case all system frequencies will be reduced as a fraction of the input clock verses the standard 20MHz clock. The list below shows the affected system parameters:-

- Serial baud rate
- Maximum pulse rate
- Timing characteristics as shown in section 3.2
- Cycle time

For example, if an input clock of 17MHz is used with a serial baud rate of 9600 the following timing changes will result:-

- Serial baud rate decreases to 9600 bps $*17/20 = 8160$ bps
- Maximum step rate decreases to 50K pulses $*17/20 = 42.5$K pulses
- Cycle time per axis increases to 102.4 μsec $*20/17 = 120.48$ μsec

7.2 RS-232 Serial Interface

The interface between the MC3510 chip and an RS-232 serial port is shown in the following figure.

Comments on Schematic

S1 and S2 encode the characteristics of the serial port such as baud rate, number of stop bits, parity, etc. The CP will read these switches during initialization, but these parameters may also be set or changed using the **SetSerialPort** chipset command. The DB9 connector wired as shown can be connected directly to the serial port of a PC without requiring a null modem cable.

R?
22K
VCC
DS[0..15]
A[0..15]

U1
CP2N11

-RS
~RS
VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC
35 2 7 13 21 36 40 47 50 60 62 93 103 121

DS0 9   DATA0   ADDR0 110 A0
DS1 10 DATA1   ADDR1 111 A1
DS2 11 DATA2   ADDR2 112 A2
DS3 12 DATA3   ADDR3 114 A3
DS4 15 DATA4   ADDR4 115 A4
DS5 16 DATA5   ADDR5 116 A5
DS6 17 DATA6   ADDR6 117 A6
DS7 18 DATA7   ADDR7 118 A7
DS8 19 DATA8   ADDR8 119 A8
DS9 22 DATA9   ADDR9 122 A9
DS10 23 DATA10   ADDR10 123 A10
DS11 24 DATA11   ADDR11 124 A11
DS12 25 DATA12   ADDR12 125 A12
DS13 26 DATA13   ADDR13 126 A13
DS14 27 DATA14   ADDR14 127 A14
DS15 28 DATA15   ADDR15 128 A15

                      131 N/C
                 129 ~RAMSLCT
                 130 ~PERIPHSLCT
                 4   R/~W   IS-
                 6   ~STROBE   R/W
                 132 ~WRITEENBL STRB-
                      W/~R

63 POSLIM1
64 NEGLIM1   ~HOSTINTRPT 98

72 AXISIN1   DIRECTION1   105
94 AXISOUT1   PULSE1   106
                      ATREST1   107
67 QUADA1
68 QUADB1
69 ~INDEX1
70 ~HOME1

                      SRLRCV   43
                      SRLXMT   44
                      SRLENABLE 99

                      I/OINTRPT 53
                      PRLENABLE 65

                      ANALOG1   74
                      ANALOG2   89
                      ANALOG3   75
                      ANALOG4   88
                      ANALOG5   76
                      ANALOG6   83
                      ANALOG7   77
                      ANALOG8   82

                      ANALOGVCC   84
                      ANALOGREFHIGH 85
                      ANALOGREFLOW   86
                      ANALOGGND   87

RS- 41 ~RESET
CLK 58 CLOCKIN

GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND
3 8 14 20 29 37 46 56 59 61 71 92 104 113 120

GND

RS1
RSIP9
1 VCC
COM
2 SW1
R1
3 SW2
R2
4 SW3
R3
5 SW4
R4
6 SW5
R5
7 SW6
R6
8 SW7
R7
9 SW8
R8

RS2
RSIP9
1 VCC
COM
2 SW9
R1
3 SW10
R2
4 SW11
R3
5 SW12
R4
6 SW13
R5
7 SW14
R6
8 SW15
R7
9 SW16
R8

DS[0..15]
DS[0..15]

S1
SW DIP-8
1   16 SW1
2   15 SW2
3   14 SW3
4   13 SW4
5   12 SW5
6   11 SW6
7   10 SW7
8   9   SW8

U2
74LS244
2   SW1 1A1   18 1Y1 DS0
4   SW2 1A2   16 1Y2 DS1
6   SW3 1A3   14 1Y3 DS2
8   SW4 1A4   12 1Y4 DS3
11 SW5 2A1   9   2Y1 DS4
13 SW6 2A2   7   2Y2 DS5
15 SW7 2A3   5   2Y3 DS6
17 SW8 2A4   3   2Y4 DS7
1   1G
19 2G

S2
SW DIP-8
1   16 SW9
2   15 SW10
3   14 SW11
4   13 SW12
5   12 SW13
6   11 SW14
7   10 SW15
8   9   SW16

U3
74LS244
2   SW9   1A1 18 1Y1 DS8
4   SW10 1A2 16 1Y2 DS9
6   SW11 1A3 14 1Y3 DS10
8   SW12 1A4 12 1Y4 DS11
11 SW13 2A1 9   2Y1 DS12
13 SW14 2A2 7   2Y2 DS13
15 SW15 2A3 5   2Y3 DS14
17 SW16 2A4 3   2Y4 DS15
1   1G
19 2G

U2 AND U3 COULD BE IMPLEMENTED IN A PLD

IS- 2   1   A9
NOT   3
R/W   4
          5   U2
               NAND4 1
STRB- 2   1
NOT

VCC

C1   .1UF 50V
C2   .1UF 50V
C3   .1UF 50V
C5   .1UF 50V
C4   .1UF 50V

U3
AD232
C1+ 1 C1+
        C1- 3 C1-
C2+ 4 C2+
        C2- 5 C2-
SERXMIT 11 T1IN
         10 T2IN
SERRCV   12 R1OUT
         9   R2OUT

2   V+   V+
6   V-   V-
14 T1OUT TXD
7   T2OUT
13 R1IN   RXD
8   R2IN

GND

J1
CONNECTOR DB9
5
9
4
8
3
7
2
6
1

FEMALE DB9 WIRED
AS SHOWN WILL
CONNECT TO A PC
WITHOUT A DUMMY
MODEM.

PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title
RS232 SERIAL INTERFACE

Size Document Number   Rev
B   B

Date: Monday, July 07, 2003   Sheet 1 of 0

7.3   RS 422/485 Serial Interface

The interface between the MC3510 chip and an RS-422/485 serial port is shown in the following figure.

Comments on Schematic

Use the included table to determine the jumper setup that matches the chosen configuration. If using RS485, the last CP must have its jumpers set to RS485 LAST. The DB9 connector wiring is for example only. The DB9 should be wired according to the specification that accompanies the connector to which it is attached.

For correct operation, logic should be provided that contains the start up serial configuration for the motion processor. Refer to the RS232 Serial Interface schematic for an example of the required logic.

Note that the RS485 interface cannot be used in point to point mode. It can only be used in a multi-drop configuration where the chip SrlEnable line is used to control transmit/receive operation of the serial transceiver.

Chips in a multi-drop environment should not be operated at different baud rates. This will result in communication problems.

TX-RX +

TERMINATE
TRANSMIT

JP3
1 3
2
JMP3

TXT

JP1
1 3
2
JMP3

VCC

R3
4.7K

R1
120

U1

P1
5
9
4
8
3
7
2
6
1

TO HOST

VCC
14

SRLXMT
SRLRCV
SRLENABLE

5   DI   Y   9   TX+
4   DE   Z   10   TX-
3   RE
2   RO   A   12   RX+
          B   11   RX-

GND

VCC

C1
4.7UF
10V
TANT

GND GND
6   7

MAX491

C2
.1UF
50V
CER

GND

CONNECTOR DB9
RT ANGLE MALE

R2
120

JP4
1 3
2
JMP3

RXT

JP2
1 3
2
JMP3

TX-RX -

TERMINATE
RECEIVE
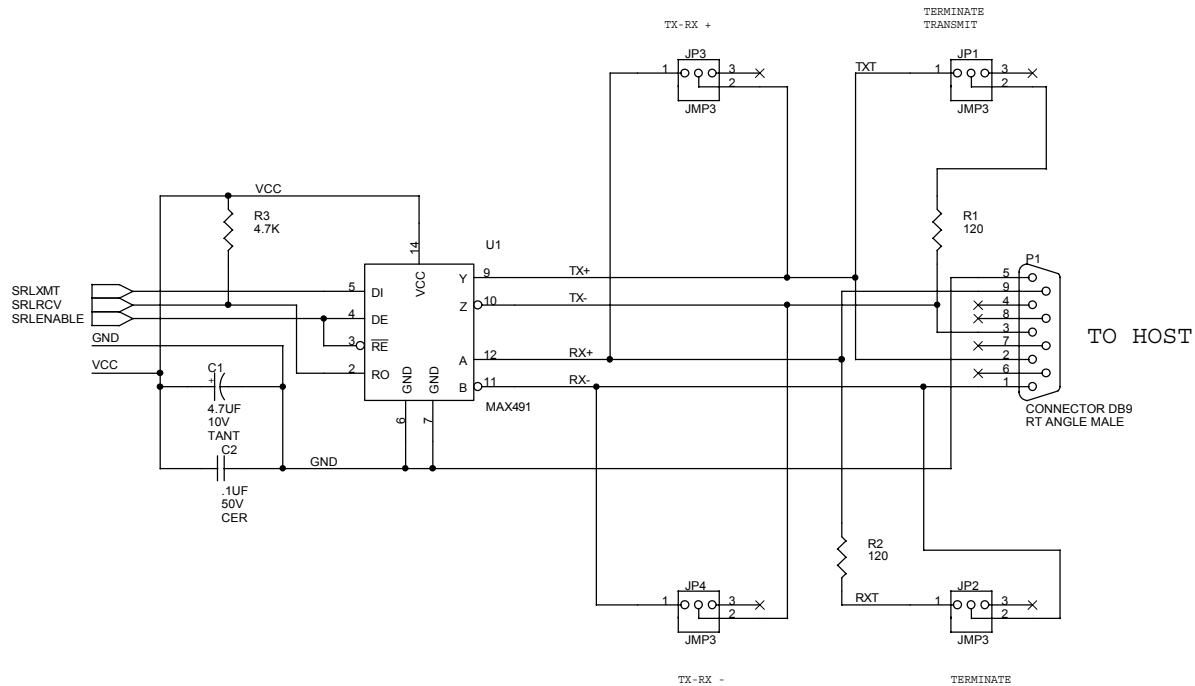
COM TYPE	JP1	JP2	JP3	JP4
RS422	1-2	1-2	2-3	2-3
RS485	2-3	2-3	1-2	1-2
RS485 LAST	1-2	2-3	1-2	1-2

NOTE:RS422 IS CAPABLE OF FULL DUPLEX AND USES 2 PAIRS.

RS485 IS HALF-DUPLEX ON 1 PAIR AND MAY BE DAISY CHAINED

THE CP USES RS485. A SINGLE CP MAY COMMUNICATE WITH AN

RS422 HOST AS SHOWN IN THE TABLE.

A SINGLE PAIR MAY BE WIRED TO EITHER P1-1,9 OR P1-2,3

FOR RS485.

PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title
RS422/485 Interface

Size
B

Document Number

Rev
A

Date: Thursday, April 11, 2002   Sheet 1 of 1

7.4   RAM Interface

The following schematic shows an interface circuit between the MC3510 and external ram.

Comments on Schematic

The CP is capable of directly addressing 32K words of 16-bit memory. It will also use a 16 bit paging register to address up to 32K word pages. The schematic shows the paging and addressing for 128KB RAM chips, i.e. 4 pages per RAM chip. The page address decoding is shown for only 6 of the 16 possible paging bits. The decoding time from W/R and DS- to the memory output must not exceed 18 ns. for a read with no wait states. The writes provide 25 extra ns access time for W/R and DS- to reverse the CP data bus.

R? VCC
22K

VCC
VCC

U?

~RS
VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC

D0 9   DATA0   ADDR0 110 A0
D1 10   DATA1   ADDR1 111 A1
D2 11   DATA2   ADDR2 112 A2
D3 12   DATA3   ADDR3 113 A3
D4 15   DATA4   ADDR4 114 A4
D5 16   DATA5   ADDR5 115 A5
D6 17   DATA6   ADDR6 116 A6
D7 18   DATA7   ADDR7 117 A7
D8 19   DATA8   ADDR8 118 A8
D9 22   DATA9   ADDR9 122 A9
D10 23   DATA10   ADDR10 123 A10
D11 24   DATA11   ADDR11 124 A11
D12 25   DATA12   ADDR12 125 A12
D13 26   DATA13   ADDR13 126 A13
D14 27   DATA14   ADDR14 127 A14
D15 28   DATA15   ADDR15 128
                      N/C   131
         ~RAMSLCT   129 DS-
         ~PERIPHSLCT 130 IS-
63 POSLIM1   R/~W   4   R/W
64 NEGLIM1   ~STROBE 6   WE-
                      ~WRITEENBL 119 W/R
                                 132
72 AXISIN1   ~HOSTINTRPT 98
94 AXISOUT1
                      DIRECTION1 105
67 QUADA1   PULSE1   106
68 QUADB1   ATREST1   107
69 ~INDEX1
70 ~HOME1

                      SRLRCV   43
                      SRLXMT   99
                      SRLENABLE

                      I/OINTRPT 53
                      PRLENABLE 65

                      ANALOG1   74
                      ANALOG2   89
                      ANALOG3   75
                      ANALOG4   88
                      ANALOG5   76
                      ANALOG6   83
                      ANALOG7   77
                      ANALOG8   82

                      ANALOGVCC   84
                      ANALOGREFHIGH 85
                      ANALOGREFLOW 86
                      ANALOGGND   87

RS- 41 ~RESET
CLK 58 CLOCKIN GND GND GND GND GND GND GND GND GND GND GND GND GND GND
         3 8 14 20 29 37 46 56 61 71 92 104 113 120
         CP2N11
                                GND

D[0..15]
A[0..14]

NOTE: POS139 IS A STANDARD 139 WITH INVERTED OUTPUTS

U2 74LS377
D0 3   D1 Q1 2   MPG0
D1 4   D2 Q2 5   MPG1
D2 7   D3 Q3 6
D3 8   D4 Q4 9
D4 13 D5 Q5 12
D5 14 D6 Q6 15
D6 17 D7 Q7 16 X
D7 18 D8 Q8 19 X
WE- 11 CLK
PGR- 1   G

U2A POS139
2 A Y0 4 CS1
3 B Y1 5 CS2
          Y2 6 CS3
1 G Y3 7 CS4

U2B POS139
14 A Y0 12 CS5
13 B Y1 11 CS6
           Y2 10 CS7
GND 15 G Y3 9   CS8

U2 74LS377
D8 3   D1 Q1 2
D9 4   D2 Q2 5
D10 7   D3 Q3 6
D11 8   D4 Q4 9
D12 13 D5 Q5 12
D13 14 D6 Q6 15
D14 17 D7 Q7 16
D15 18 D8 Q8 19
WE- 11 CLK
PGR- 1   G

PAGE REGISTER UP TO 16 BITS

NOTE: THE CRITICAL DECODE AND MEMORY
ACCESS TIME IS DURING READ,
THE REQUIRED ACCESS TIME IS
18 NS. FROM DS- LOW.
AS ILLUSTRATED THERE IS ~ 100NS.
TO ACCOMPLISH THE DECODING
FROM PAGE REG WRITE TO
MEMORY READ OR WRITE.
DECODING WILL HAVE TO BE
CAREFULLY DONE ON MEMORIES
WITH A SINGLE CHIP SELECT.

U2 NOT
A13 2   1
         IS- 3
         R/~W 4
U2 OR3
         1 PGR-

MCM6226 (U2)
A0 12 A0   DQ0 13 D0
A1 11 A1   DQ1 14 D1
A2 10 A2   DQ2 15 D2
A3 9   A3   DQ3 17 D3
A4 8   A4   DQ4 18 D4
A5 7   A5   DQ5 19 D5
A6 6   A6   DQ6 20 D6
A7 5   A7   DQ7 21 D7
A8 27 A8
A9 26 A9
A10 23 A10
A11 25 A11
A12 4   A12
A13 28 A13
A14 3   A14
MPG0 31 A15
MPG1 2   A16
DS- 22 CE1
CS1 30 CE2
WE- 29 WE
W/R 24 OE

MCM6226 (U2)
A0 12 A0   DQ0 13 D8
A1 11 A1   DQ1 14 D9
A2 10 A2   DQ2 15 D10
A3 9   A3   DQ3 17 D11
A4 8   A4   DQ4 18 D12
A5 7   A5   DQ5 19 D13
A6 6   A6   DQ6 20 D14
A7 5   A7   DQ7 21 D15
A8 27 A8
A9 26 A9
A10 23 A10
A11 25 A11
A12 4   A12
A13 28 A13
A14 3   A14
MPG0 31 A15
MPG1 2   A16
DS- 22 CE1
CS1 30 CE2
WE- 29 WE
W/R 24 OE

D[0..15]
A[0..14]

MCM6226 (U2)
A0 12 A0   DQ0 13 D0
A1 11 A1   DQ1 14 D1
A2 10 A2   DQ2 15 D2
A3 9   A3   DQ3 17 D3
A4 8   A4   DQ4 18 D4
A5 7   A5   DQ5 19 D5
A6 6   A6   DQ6 20 D6
A7 5   A7   DQ7 21 D7
A8 27 A8
A9 26 A9
A10 23 A10
A11 25 A11
A12 4   A12
A13 28 A13
A14 3   A14
MPG0 31 A15
MPG1 2   A16
DS- 22 CE1
CS2 30 CE2
WE- 29 WE
W/R 24 OE

MCM6226 (U2)
A0 12 A0   DQ0 13 D8
A1 11 A1   DQ1 14 D9
A2 10 A2   DQ2 15 D10
A3 9   A3   DQ3 17 D11
A4 8   A4   DQ4 18 D12
A5 7   A5   DQ5 19 D13
A6 6   A6   DQ6 20 D14
A7 5   A7   DQ7 21 D15
A8 27 A8
A9 26 A9
A10 23 A10
A11 25 A11
A12 4   A12
A13 28 A13
A14 3   A14
MPG0 31 A15
MPG1 2   A16
DS- 22 CE1
CS2 30 CE2
WE- 29 WE
W/R 24 OE

7.5   User-defined I/O

The interface between the MC3510 chip and 16 bits of user output and 16 bits of user input is shown in the following figure.

Comments on Schematic

The schematic implements 1 word of user output registered in the 74LS377's and 1 word of user inputs read via the 244's. The schematic decodes the low 3 bits of the address to 8 possible UIO addresses UIO0 through UIO7. Registers and buffers are shown for only UIO0, but the implementation shown may be easily extended. The lower 8 address bits may be decoded to provide up to 256 user output words and 256 user input words of 16 bits.

D[0..15]

A[0..14]

R?
22K
VCC

U?

35 ~RS
2 VCC
7 VCC
13 VCC
21 VCC
36 VCC
40 VCC
47 VCC
50 VCC
52 VCC
60 VCC
62 VCC
93 VCC
100 VCC
121 VCC

D0 9 DATA0   ~ADDR0 110 A0
D1 10 DATA1   ADDR1 111 A1
D2 11 DATA2   ADDR2 112 A2
D3 12 DATA3   ADDR3 114 A3
D4 15 DATA4   ADDR4 115 A4
D5 16 DATA5   ADDR5 116 A5
D6 17 DATA6   ADDR6 117 A6
D7 18 DATA7   ADDR7 118 A7
D8 19 DATA8   ADDR8 119 A8
D9 22 DATA9   ADDR9 122 A9
D10 23 DATA10   ADDR10 123 A10
D11 24 DATA11   ADDR11 124 A11
D12 25 DATA12   ADDR12 125 A12
D13 26 DATA13   ADDR13 126 A13
D14 27 DATA14   ADDR14 127 A14
D15 28 DATA15   ADDR15 128
                   N/C 131
             ~RAMSLCT 129   IS-
            ~PERIPHSLCT 130
63 POSLIM1   R/~W 4
64 NEGLIM1   ~STROBE 6   WF-
            ~WRITEENBL 5
                 W/~R 132   W/R
72 AXISIN1   ~HOSTINTRPT 98
94 AXISOUT1
             DIRECTION1 105
67 QUADA1   PULSE1 106
68 QUADB1   ATREST1 107
69 ~INDEX1
70 ~HOME1
                SRLRCV 43
                SRLXMT 44
             SRLENABLE 99
             I/OINTRPT 53
             PRLENABLE 65
               ANALOG1 74
               ANALOG2 89
               ANALOG3 75
               ANALOG4 88
               ANALOG5 76
               ANALOG6 83
               ANALOG7 77
               ANALOG8 82
            ANALOGVCC 84
         ANALOGREFHIGH 85
         ANALOGREFLOW 86
            ANALOGGND 87
RS- 41 ~RESET
CLK 58 ~CLOCKIN   GND
3 8 14 20 29 37 46 56 59 61 71 92 103 120
CP2N11
GND

U2   74LS377
D0 3 D1   Q1 2 UIO0-0
D1 4 D2   Q2 5 UIO0-1
D2 7 D3   Q3 6 UIO0-2
D3 8 D4   Q4 9 UIO0-3
D4 13 D5   Q5 12 UIO0-4
D5 14 D6   Q6 15 UIO0-5
D6 17 D7   Q7 16 UIO0-6
D7 18 D8   Q8 19 UIO0-7
WF- 11 CLK
UIO0 1 G

USER OUTPUTS

U2   74LS377
D8 3 D1   Q1 2 UIO0-8
D9 4 D2   Q2 5 UIO0-9
D10 7 D3   Q3 6 UIO0-10
D11 8 D4   Q4 9 UIO0-11
D12 13 D5 Q5 12 UIO0-12
D13 14 D6 Q6 15 UIO0-13
D14 17 D7 Q7 16 UIO0-14
D15 18 D8 Q8 19 UIO0-15
WF- 11 CLK
UIO0 1 G

U2   244
D0 18 1Y1   1A1 2 UI0-0
D1 16 1Y2   1A2 4 UI0-1
D2 14 1Y3   1A3 6 UI0-2
D3 12 1Y4   1A4 8 UI0-3
D4 9 2Y1   2A1 11 UI0-4
D5 7 2Y2   2A2 13 UI0-5
D6 5 2Y3   2A3 15 UI0-6
D7 3 2Y4   2A4 17 UI0-7
           1G 1 UIOn
           2G 19 UIOn

USER INPUTS

U2   244
D8 18 1Y1   1A1 2 UI0-8
D9 16 1Y2   1A2 4 UI0-9
D10 14 1Y3 1A3 6 UI0-10
D11 12 1Y4 1A4 8 UI0-11
D12 9 2Y1   2A1 11 UI0-12
D13 7 2Y2   2A2 13 UI0-13
D14 5 2Y3   2A3 15 UI0-14
D15 3 2Y4   2A4 17 UI0-15
           1G 1 UIOn
           2G 19 UIOn

U2   138
A0 1 A   Y0 15 UIO0
A1 2 B   Y1 14 UIO1
A2 3 C   Y2 13 UIO2
          Y3 12 UIO3
UIO 6 G1 Y4 11 UIO4
A3 4 G2A Y5 10 UIO5
A4 5 G2B Y6 9 UIO6
          Y7 7 UIO7

U2
A12 1 NOT 2
A12n 1   U2
IS- 3   NOR2   UIO

U2
A12n 2 OR2 1 UIOn
IS- 3   W/R 3 U2 UIOn
             UIO0 4 OR3

THE LOGIC LABELED U2 MAY BE IMPLEMENTED IN
A CPLD. THE LOWER 8 ADDRESS BITS, A0-A8, MAY BE
DECODED TO PROVIDE 256 16 BIT USER INPUTS
AND 256 USER OUTPUTS.

PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title
USER I/O

Size B   Document Number   Rev D

Date: Saturday, December 07, 2002   Sheet 1 of 0

7.6   12-bit A/D Interface

The following schematic shows a typical interface circuit between the MC3510 and a quad 12 bit 2's complement A/D converter used as a position input device. Any single channel A/D can also be used provided it meets the interface timing requirements.

Comments on Schematic

The A/D converter samples the 2's complement digital words. DACRD- is used to perform the read and is also used to load the counter to FFh. The counter will be reloaded for each read and will not count significantly between reads. The counter will therefore start counting down after the last read and will generate the cvt- pulse after 12.75 μsec. The conversions will take approximately 35 μsec, and the data will be available for the next set of reads after 50 μsec. The 12 bit words from the A/D are extended to 16 bits with the 74LS244.

DS[0..15]

R?
22K
VCC

U1

~RS VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC

A[0..15]

NOTE:FS INPUTS ARE +- 10V

DS0	9	DATA0
DS1	10	DATA1
DS2	11	DATA2
DS3	12	DATA3
DS4	15	DATA4
DS5	16	DATA5
DS6	17	DATA6
DS7	18	DATA7
DS8	19	DATA8
DS9	22	DATA9
DS10	23	DATA10
DS11	24	DATA11
DS12	25	DATA12
DS13	26	DATA13
DS14	27	DATA14
DS15	28	DATA15

ADDR0 110 A0
ADDR1 111 A1
ADDR2 112 A2
ADDR3 114 A3
ADDR4 116 A4
ADDR5 117 A5
ADDR6 118 A6
ADDR7 119 A7
ADDR8 122 A8
ADDR9 123 A9
ADDR10 124 A10
ADDR11 125 A11
ADDR12 126 A12
ADDR13 127 A13
ADDR14 128 A14
ADDR15 128 A15
N/C 131

~RAMSLCT 129 IS-
~PERIPHSLCT 130
R/~W 4 STRB-
~STROBE 6
~WRITEENBL 5 W/R
W/~R 132

U2
OR4

A11 2 1
NOT

~HOSTINTRPT 98

DIRECTION1 105
PULSE1 106
ATREST1 107

SRLRCV 43
SRLXMT 44
SRLENABLE 99

I/OINTRPT 53
PRLENABLE 65

ANALOG1 74
ANALOG2 89
ANALOG3 75
ANALOG4 88
ANALOG5 76
ANALOG6 83
ANALOG7 77
ANALOG8 82

ANALOGVCC 84
ANALOGREFHIGH 85
ANALOGREFLOW 86
ANALOGGND 87

POSLIM1 63
NEGLIM1 64

AXISIN1 72
AXISOUT1 94

QUADA1 67
QUADB1 68
~INDEX1 69
~HOME1 70

RS- 41 ~RESET
CLK 58 CLOCKIN

GND GND GND GND GND GND GND GND GND GND GND GND GND GND GND

CP2N11

GND

U?

POS1 1 VIN1
POS2 2 VIN2
POS3 27 VIN3
POS4 28 VIN4

CVT- 5 CONVST
DACRD- 6 RD
7 CS

24 REFIN
25 REFOUT

8 CLK

VDD VDD

DB11 10 DS11
DB10 11 DS10
DB9 12 DS9
DB8 13 DS8
DB7 15 DS7
DB6 16 DS6
DB5 17 DS5
DB4 18 DS4
DB3 19 DS3
DB2 20 DS2
DB1 21 DS1
DB0 22 DS0

INT 4

VSS AGND DGND
26 23 14

-5VA GND

AGND

AD7874

DS[0..15]

U2

DS11 2 1A1 1Y1 18 DS15
6 1A2 1Y2 16 DS14
8 1A3 1Y3 14 DS13
11 1A4 1Y4 12 DS12
13 2A1 2Y1 9
15 2A2 2Y2 7
17 2A3 2Y3 5
2A4 2Y4 3

DACRD- 1 1G
19 2G

74LS244

NOTE:SIGN EXTENTION FOR 2'S COMPLEMENT

VCC

U2
3 A QA 14
4 B QB 13
5 C QC 12
6 D QD 11
CLK 2 CLK RCO 15
DACRD- 9 LOAD
GND 1 U/D
10 ENT
ENCNT- 7 ENP

74ALS169

VCC

U2
3 A QA 14
4 B QB 13
5 C QC 12
6 D QD 11
CLK 2 CLK RCO 15
DACRD- 9 LOAD
GND 1 U/D
10 ENT
7 ENP

74ALS169

CVT- 2 1
NOT

U2
DFF2
D Q 1 ENCNT-
CLK 3 CLK
CL 4

DACRD-

DACRD- WILL LOAD THE COUNTER TO 255.
12.8 USEC. AFTER THE LAST DACRD-
THE COUNTER WILL REACH 0 AND START THE
NEXT CONVERSION. THE INPUT WILL
BE CONVERTED IN 35 USEC. READY FOR
THE NEXT READ 50 USEC LATER.

NOTE:THE LOGIC LABELED U2 MAY
BE IMPLEMENTEDIN A PLD.

PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title
12 BIT A/D IN

Size
B

Document Number

Rev
A

Date: Saturday, December 07, 2002   Sheet 1 of 0

7.7   16-bit A/D Input

The interface between the MC3510 chip and a 16-bit A/D converter as a parallel input position device is shown in the following figure.

Comments on Schematic

The schematic shows a 16 bit A/D used to provide parallel position input to axis 1. The 374 registers are required on the output of the A/D converters to make the 68-nanosecond access time of the CP. The worst-case timing of the A/D's specify 83 nanoseconds for data on the bus and 83 nanoseconds from data to tri-state on the bus. Each time the data is read the 169 counter is set to 703 decimal. This provides a 35.2-microsecond delay before the next conversion. With a 10-microsecond conversion time the data will be available for the next set of reads after 50 microseconds. The delay is used to provide a position sample close to the actual position.

VCC

R?
22K

DS[0_15]

A[0_15]

U1

~RS
VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC

DS0 9	DATA0
DS1 10	DATA1
DS2 11	DATA2
DS3 12	DATA3
DS4 15	DATA4
DS5 16	DATA5
DS6 17	DATA6
DS7 18	DATA7
DS8 19	DATA8
DS9 22	DATA9
DS10 23	DATA10
DS11 24	DATA11
DS12 25	DATA12
DS13 26	DATA13
DS14 27	DATA14
DS15 28	DATA15

ADDR0	110	A0
ADDR1	111	A1
ADDR2	112	A2
ADDR3	114	A3
ADDR4	115	A4
ADDR5	116	A5
ADDR6	117	A6
ADDR7	118	A7
ADDR8	119	A8
ADDR9	122	A9
ADDR10	123	A10
ADDR11	124	A11
ADDR12	125	A12
ADDR13	126	A13
ADDR14	127	A14
ADDR15	128	A15
N/C	131	
~RAMSLCT	129	
~PERIPHSLCT	130	IS-
R/~W	4	
~STROBE	6	STRB-
~WRITEENBL	5	
W/~R	132	W/R

63 POSLIM1
64 NEGLIM1

72 AXISIN1
94 AXISOUT1

67 QUADA1
68 QUADB1
69 ~INDEX1
70 ~HOME1

98 ~HOSTINTRPT
105 DIRECTION1
106 PULSE1
107 ATREST1

43 SRLRCV
44 SRLXMT
99 SRLENABLE

53 I/OINTRPT
65 PRLENABLE

74 ANALOG1
89 ANALOG2
75 ANALOG3
88 ANALOG4
76 ANALOG5
83 ANALOG6
77 ANALOG7
82 ANALOG8

84 ANALOGVCC
85 ANALOGREFHIGH
86 ANALOGREFLOW
87 ANALOGGND

RS- 41 ~RESET
CLK 58 CLOCKIN

CP2N11

GND

U2 OR4 — DACRD-

A11n

U2 NOT
A11 2 1 A11n

NOTE:THE LOGIC LABELED U2 MAY
BE IMPLEMENTED IN A PLD.

NOTE:FS INPUTS ARE +- 10V

AIN1 R1 200

R2 33.2

CVT-

GND

C1 2.2UF

C1 2.2UF

+5A VCC

U3

VANA VCC

1 VIN
3 REF
4 CAP
25 ~CS
24 R/~C
23 BYTE

AGND1 AGND2 DGND

D15 6	
D14 7	
D13 8	
D12 9	
D11 10	
D10 11	
D9 12	
D8 13	
D7 15	
D6 16	
D5 17	
D4 18	
D3 19	
D2 20	
D1 21	
D0 22	
BUSY 26	

AD976

AGND

GND

DACRD-

DS[0_15]

U2
374

D0 3		Q0 2	
D1 4		Q1 5	
D2 7		Q2 6	
D3 8		Q3 9	
D4 13		Q4 12	
D5 14		Q5 15	
D6 17		Q6 16	
D7 18		Q7 19	
OC 1			
CLK 11			

U2
374

D0 3		Q0 2	
D1 4		Q1 5	
D2 7		Q2 6	
D3 8		Q3 9	
D4 13		Q4 12	
D5 14		Q5 15	
D6 17		Q6 16	
D7 18		Q7 19	
OC 1			
CLK 11			

SEE ANALOG DEVICES SPECIFICATIONS FOR
ADITIONAL INFORMATION AND POWER BYPASSING.

VCC
U2
74ALS169

3 A QA 14
4 B QB 13
5 C QC 12
6 D QD 11
RCO 15
2 CLK
9 LOAD
1 U/D
10 ENT
7 ENP

CLK
DACRD-
GND

ENCNT-

VCC
U2
74ALS169

3 A QA 14
4 B QB 13
5 C QC 12
6 D QD 11
RCO 15
2 CLK
9 LOAD
1 U/D
10 ENT
7 ENP

CLK
DACRD-
GND

GND

VCC
U2
74ALS169

3 A QA 14
4 B QB 13
5 C QC 12
6 D QD 11
RCO 15
2 CLK
9 LOAD
1 U/D
10 ENT
7 ENP

CLK
DACRD-
GND

CVT-

U2 NOT
2 1

CLK

U2 DFF2
3 D Q 1 ENCNT-
CLK
CL

DACRD-

DACRD- WILL LOAD THE COUNTER TO 700.
38.4 USEC. AFTER THE DACRD-
THE COUNTER WILL REACH 0 AND START THE
NEXT CONVERSION. THE INPUT WILL
BE CONVERTED IN 10 USEC. READY FOR
THE NEXT READ AFTER 50 USEC.

7.8   External Gating Logic Index

A typical circuit for gating the Index signal with the encoder A & B channels is shown in the following schematic.

Comments on Schematic

In order for proper operation of the Index signal when used for position capture or phase correction, the signal must be gated with the A & B encoder channels to ensure that this signal is only active when all three signals are LOW. The motion processor does not perform this gating internally.

R?
22K
VCC

U1

35 ~RS
2 VCC
7 VCC
13 VCC
21 VCC
36 VCC
40 VCC
47 VCC
50 VCC
52 VCC
60 VCC
62 VCC
93 VCC
103 VCC
121 VCC

9 DATA0
10 DATA1
11 DATA2
12 DATA3
15 DATA4
16 DATA5
17 DATA6
18 DATA7
19 DATA8
22 DATA9
23 DATA10
24 DATA11
25 DATA12
26 DATA13
27 DATA14
28 DATA15

63 POSLIM1
64 NEGLIM1

72 AXISIN1
94 AXISOUT1

QUADA1 67 QUADA1
QUADB1 68 QUADB1
69 ~INDEX1
HOME1 70 ~HOME1

ADDR0 110
ADDR1 111
ADDR2 112
ADDR3 114
ADDR4 115
ADDR5 116
ADDR6 117
ADDR7 118
ADDR8 119
ADDR9 122
ADDR10 123
ADDR11 124
ADDR12 125
ADDR13 126
ADDR14 127
ADDR15 128
N/C 131
~RAMSLCT 129
~PERIPHSLCT 130
R/~W 4
~STROBE 6
~WRITEENBL 1
W/~R 132

~HOSTINTRPT 98

DIRECTION1 105
PULSE1 106
ATREST1 107

SRLRCV 43
SRLXMT 44
SRLENABLE 99

I/OINTRPT 53
PRLENABLE 65

ANALOG1 74
ANALOG2 89
ANALOG3 75
ANALOG4 88
ANALOG5 76
ANALOG6 83
ANALOG7 77
ANALOG8 82

ANALOGVCC 84
ANALOGREFHIGH 85
ANALOGREFLOW 86
ANALOGGND 87

41 ~RESET
58 CLOCKIN

U3
QUADA1 2
QUADB1 3
INDEX1 4
1 INDX1
OR3

GND
3 8 14 20 29 37 46 56 59 61 71 92 104 113 120
GND

CP24N11

PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title
EXTERNAL GATING LOGIC INDEX

Size B
Document Number
Rev A

Date: Saturday, December 07, 2002   Sheet 1 of 1

5   4   3   2   1
D   C   B   A