Pilot™ Motion Processor

MC3110 Single Chip
Technical Specifications

for Brushed Servo Motion Control

Performance Motion Devices, Inc.
55 Old Bedford Road
Lincoln, MA 01773

Revision 1.8, July 2003

NOTICE

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 DK3110 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 MC3110 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 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

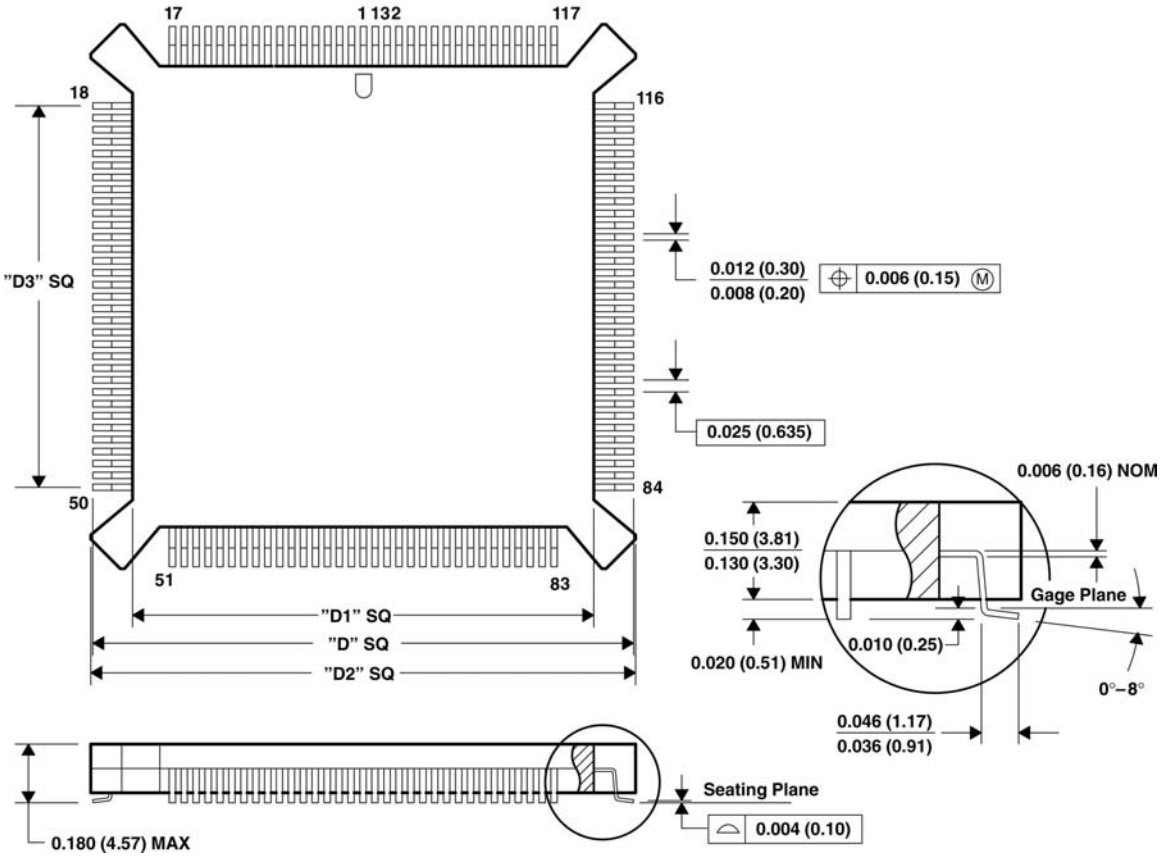
2.1 Configurations, parameters, and performance

Configuration	Single axis, single chip.
Operating modes	Closed loop (motor command is driven from output of servo filter) Open loop (motor command is driven from user-programmed register)
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)
Filter modes	Scalable PID + Velocity feedforward + Acceleration feedforward + Bias. Also includes integration limit, settable derivative sampling time, and output motor command limiting
Filter parameter resolution	16 bits
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)
Motor output modes	PWM (10-bit resolution at 20 kHz)
	DAC (16 bits)
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)
Servo loop timing range	102.4 μsec to 32.767 milliseconds
Minimum servo loop time	102.4 μsec
Multi-chip synchronization	<10μsec difference between master and slave servo cycle
	MC3113 chipset only
Limit switches	2 per axis: one for each direction of travel
Position-capture triggers	2 per axis: index and home signals
Other digital signals (per axis)	1 AxisIn signal per axis, 1 AxisOut signal per axis

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	28
Number of host instructions	132

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	

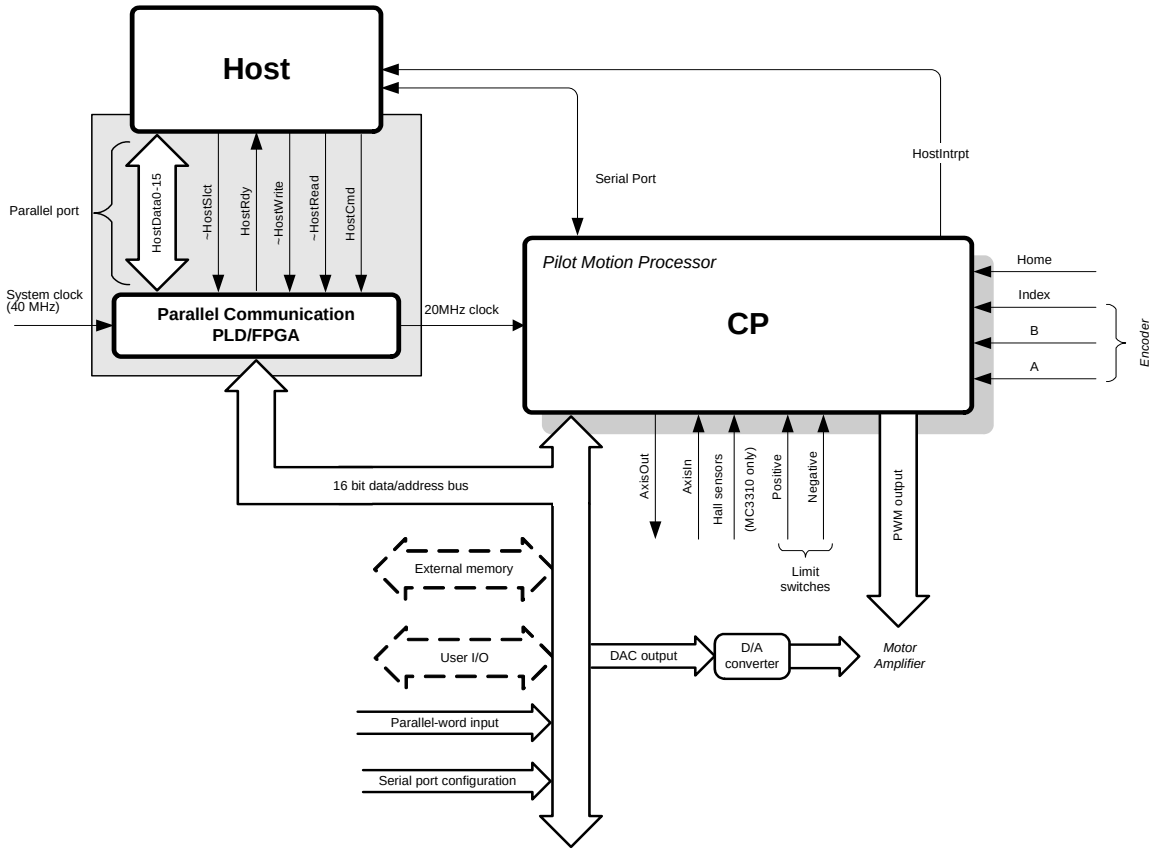
2.3 Environmental and electrical ratings

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 MC3110 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; and the digital servo filter, which stabilizes the motor output signal.

The filter produces one of two types of output:

- a Pulse-Width Modulated (PWM) signal output; or
- a DAC-compatible value routed via the data bus to the appropriate D/A converter.

Axis position information returns to the motion processor in the form of encoder feedback using either the incremental encoder input signals, or via the bus as parallel word input.

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			
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			
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			
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			
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
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.

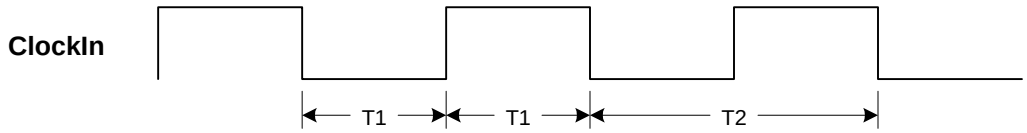
Note 2 The clock low/high split has an allowable range of 45-55%.

4 I/O Timing Diagrams

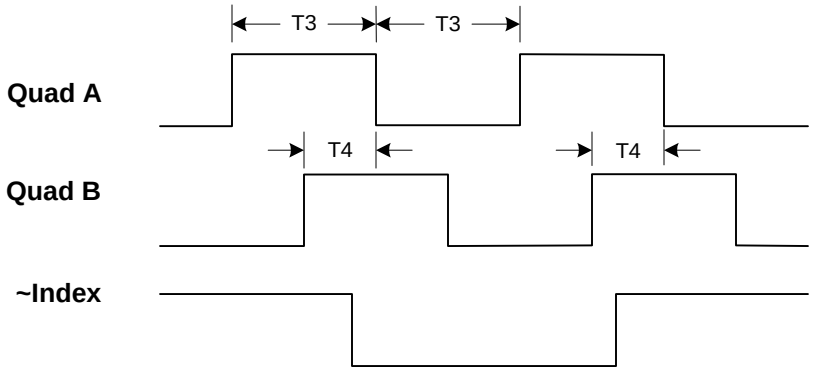
For the values of *Tn*, please refer to the table in Section 3.2.

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.

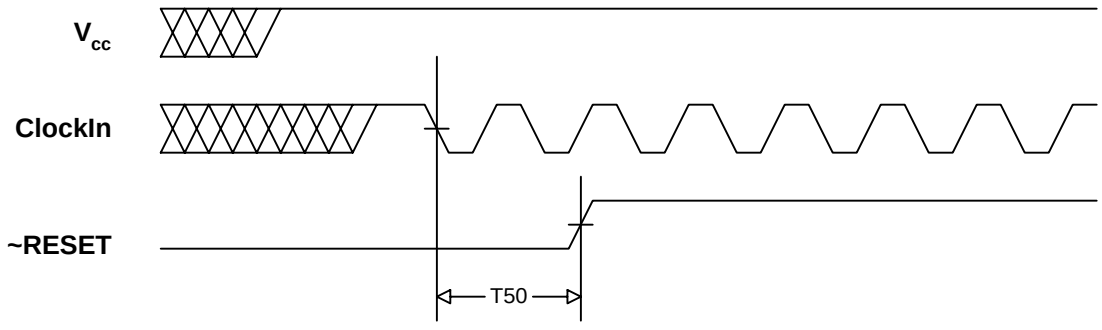
4.1 Clock



4.2 Quadrature encoder input

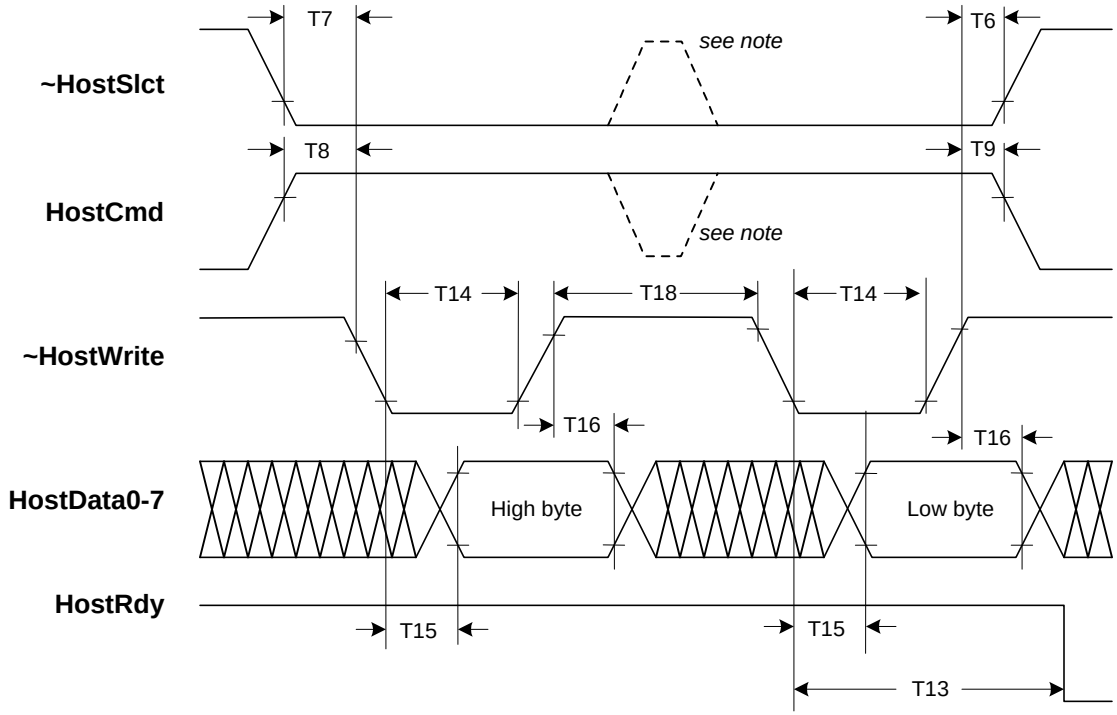


4.3 Reset



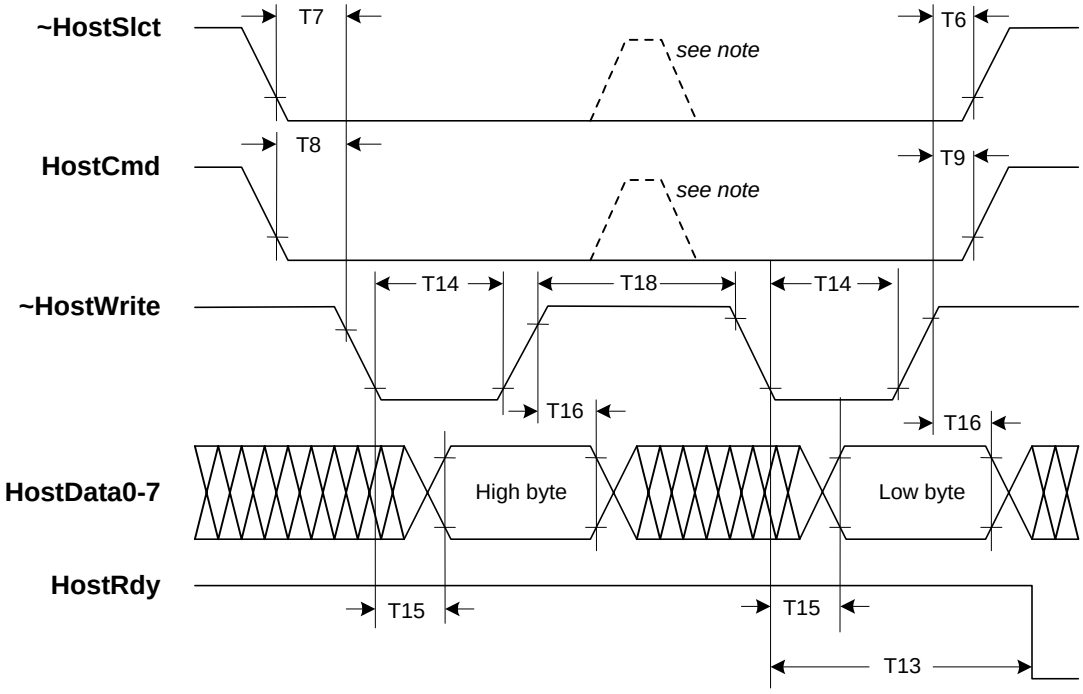
4.4 Host interface, 8/16 mode (requires external logic device)

4.4.1 Instruction write, 8/16 mode



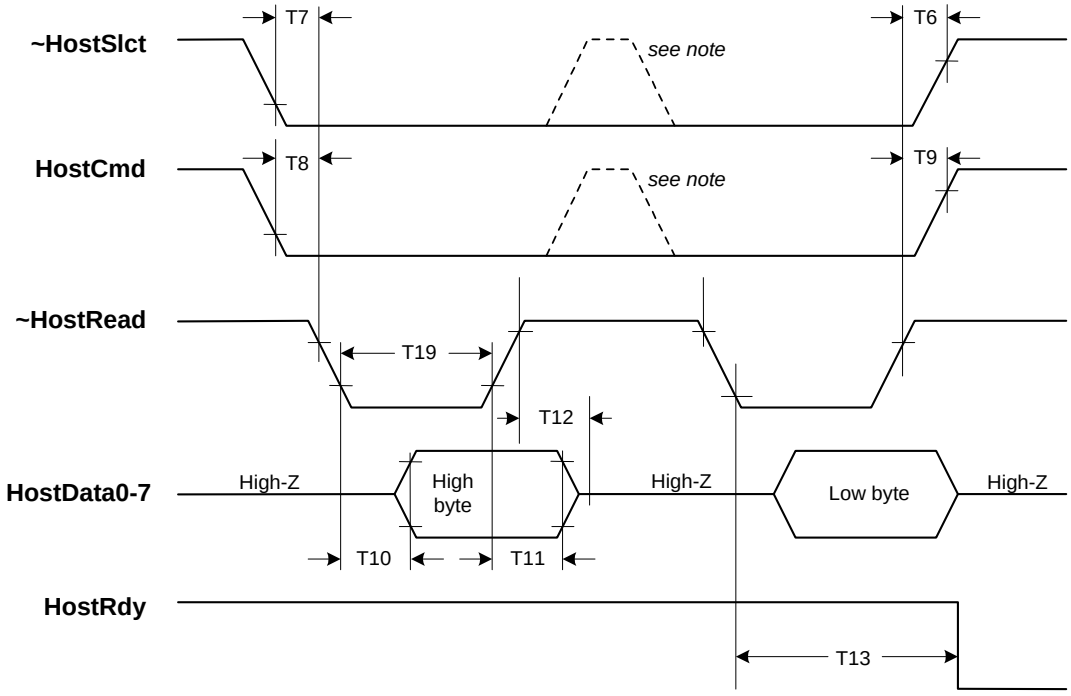
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.2 Data write, 8/16 mode



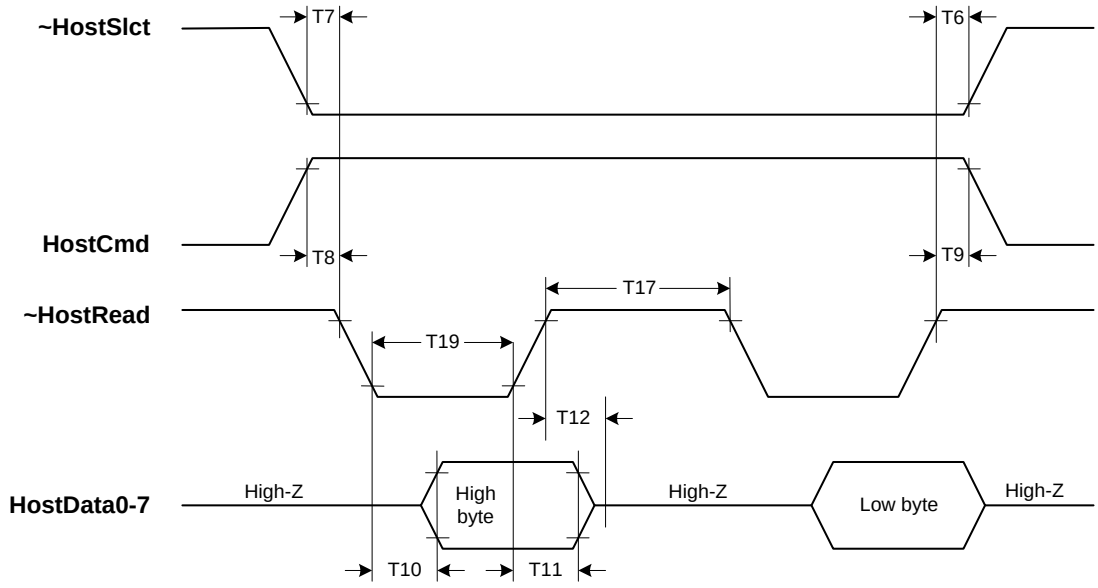
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.3 Data read, 8/16 mode



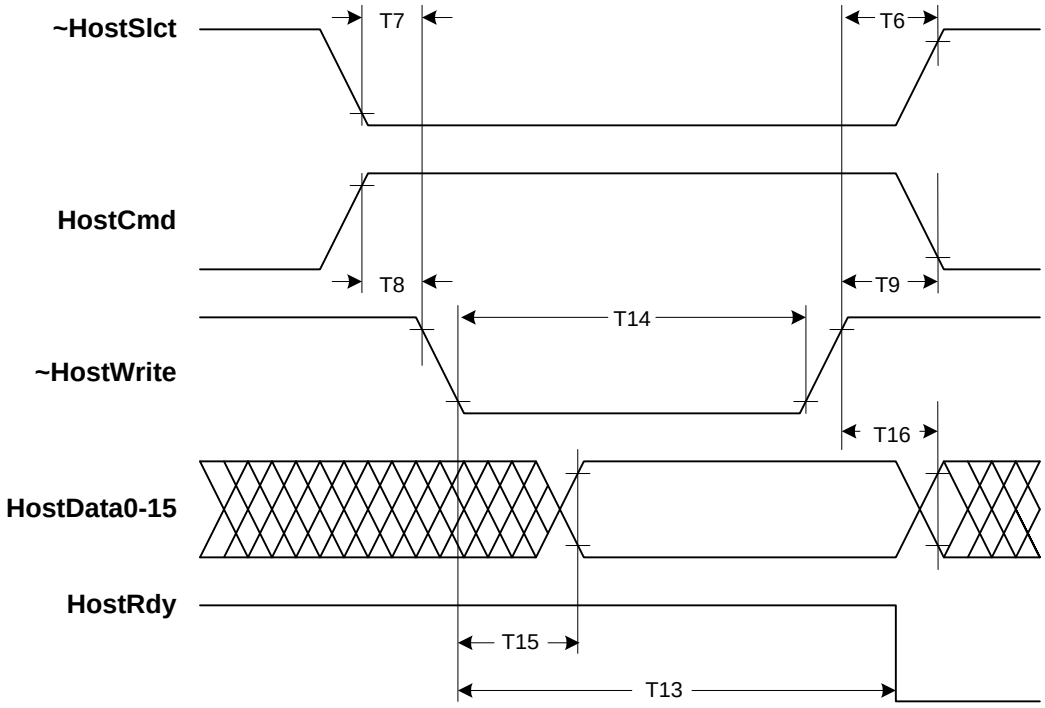
Note: If setup and hold times are met, ~HostSlct and HostCmd may be de-asserted at this point.

4.4.4 Status read, 8/16 mode

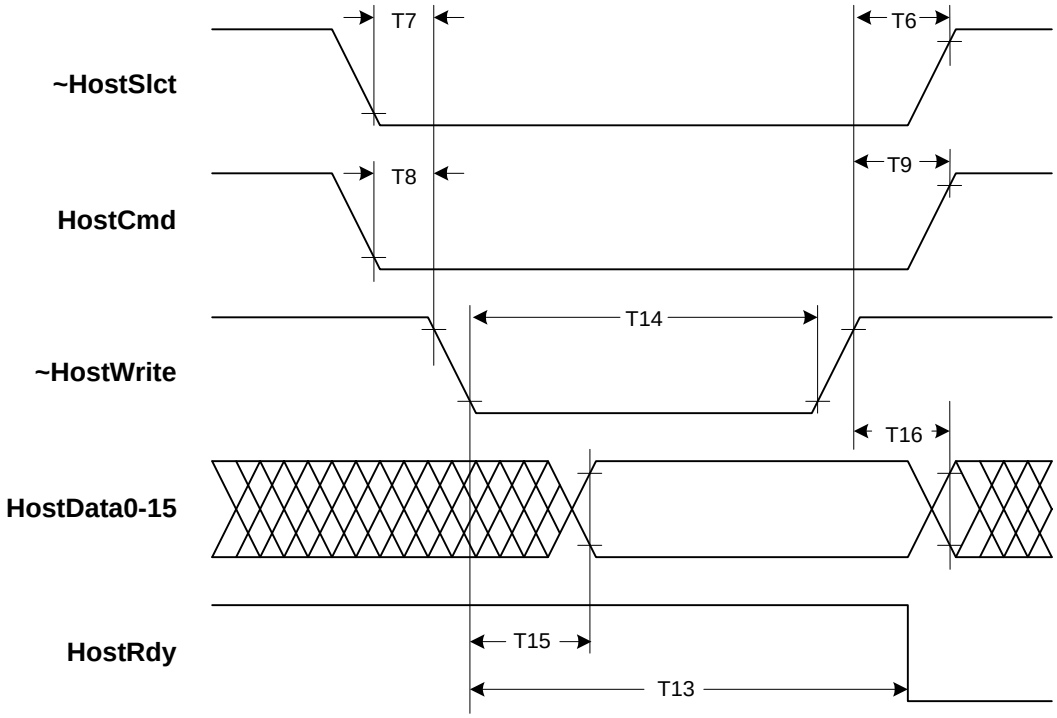


4.5 Host interface, 16/16 mode (requires external logic device)

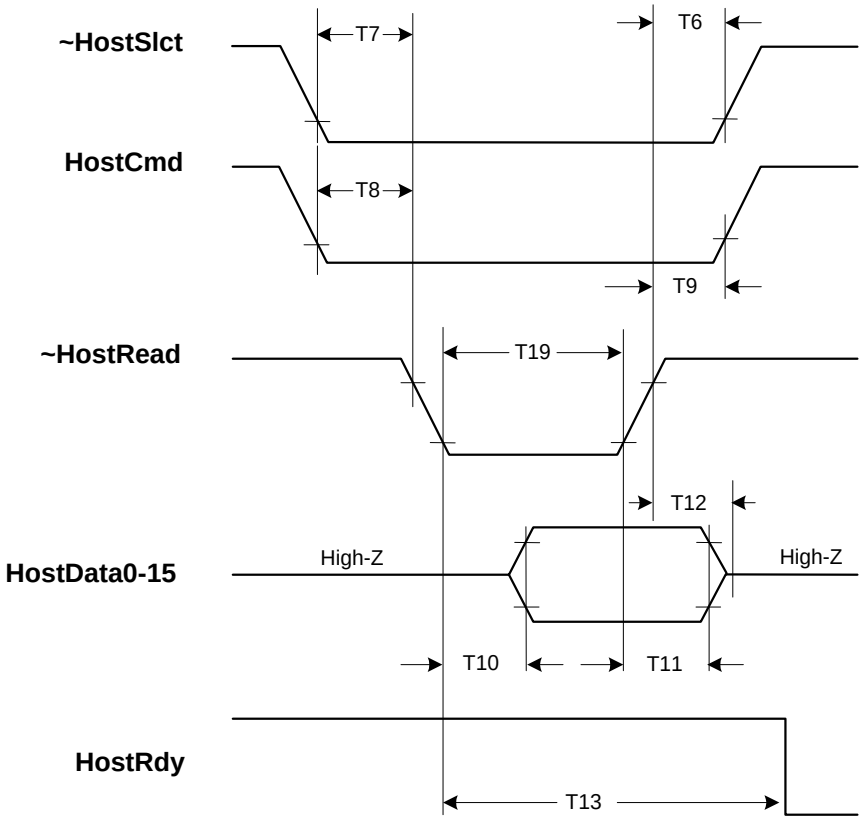
4.5.1 Instruction write, 16/16 mode



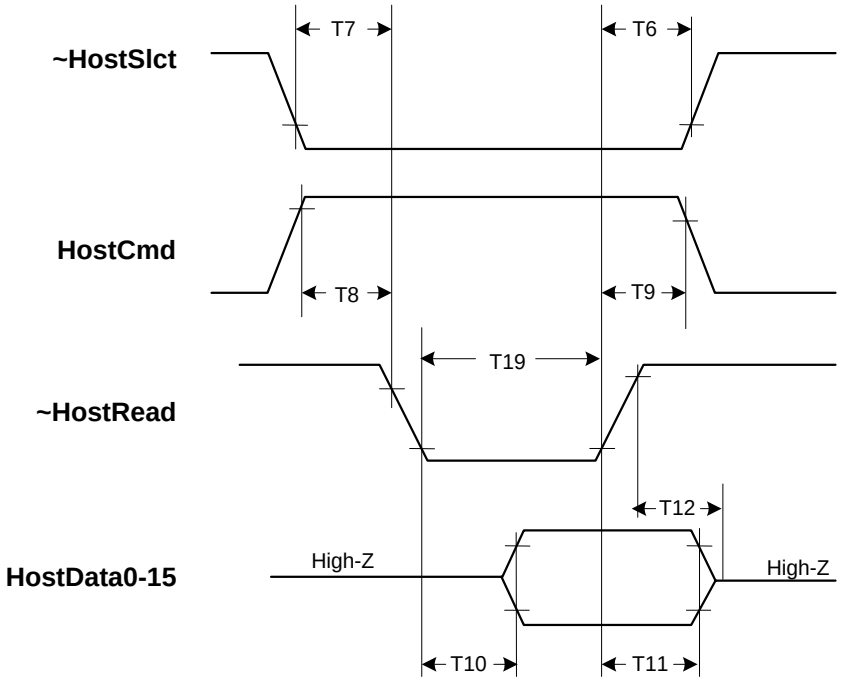
4.5.2 Data write, 16/16 mode



4.5.3 Data read, 16/16 mode



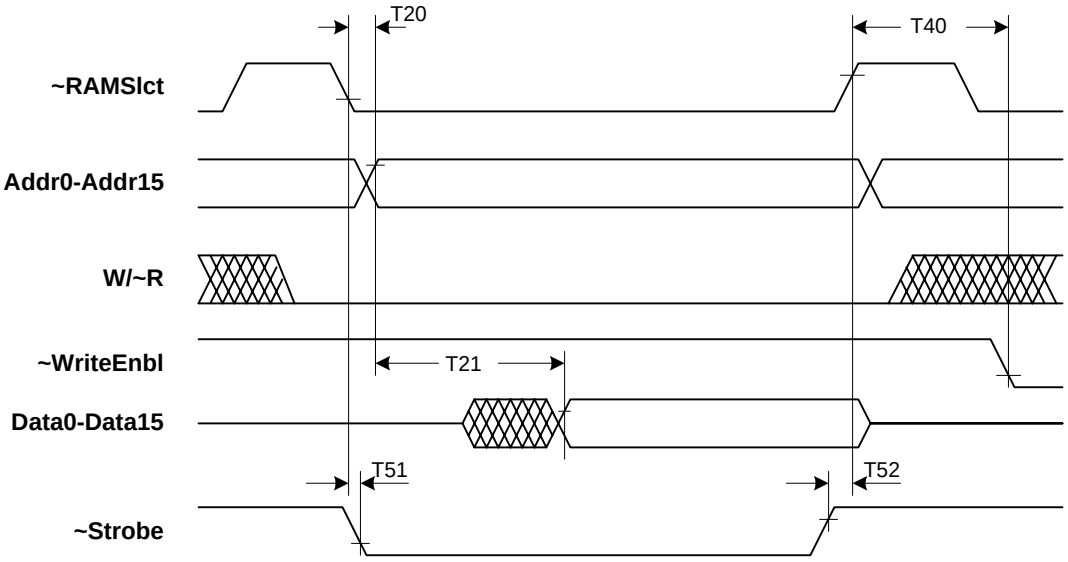
4.5.4 Status read, 16/16 mode



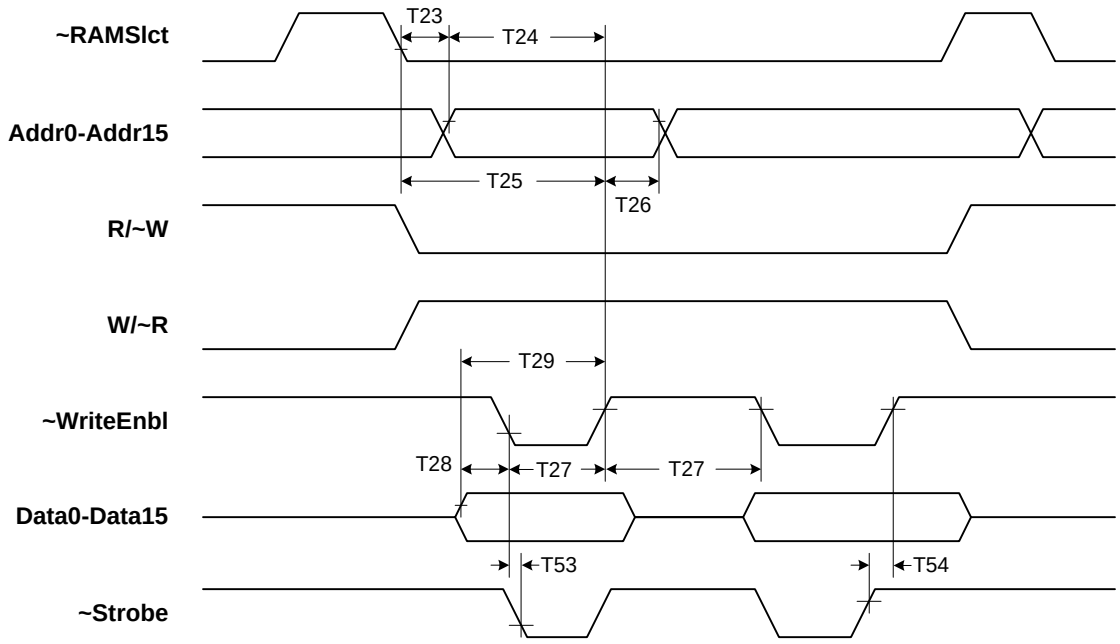
4.6 External memory timing

4.6.1 External memory read

Note: PMD recommends using memory with an access time no greater than 15 nsec.

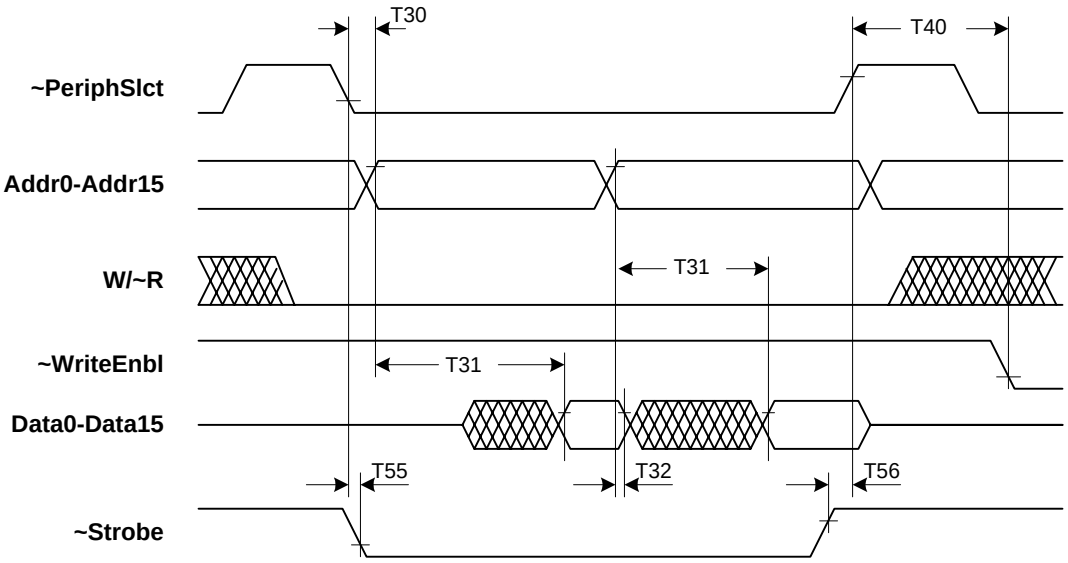


4.6.2 External memory write

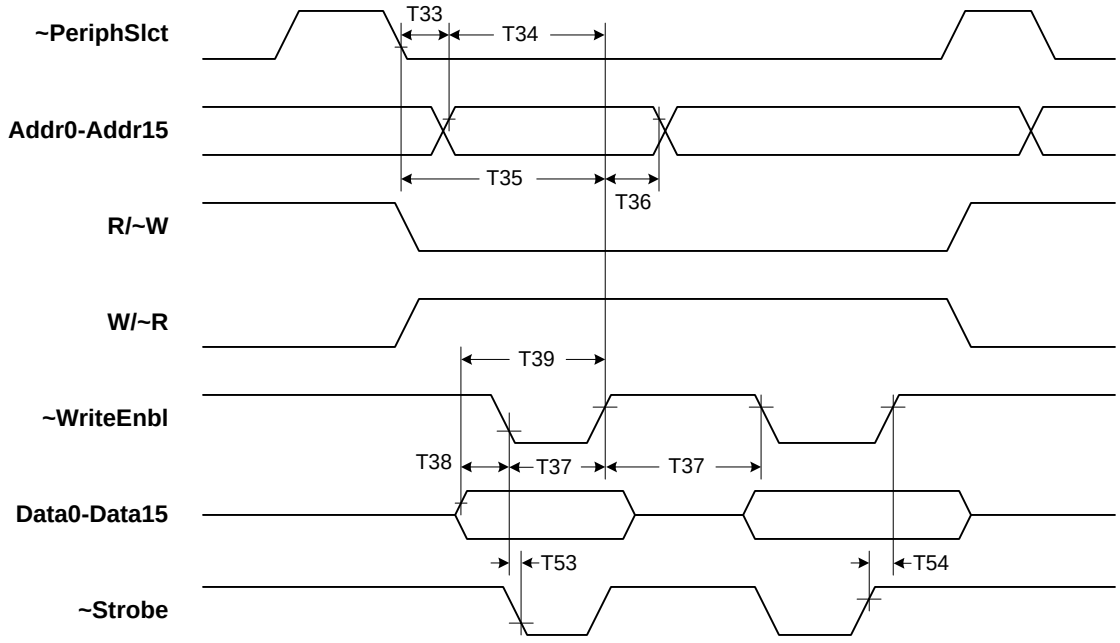


4.7 Peripheral device timing

4.7.1 Peripheral device read

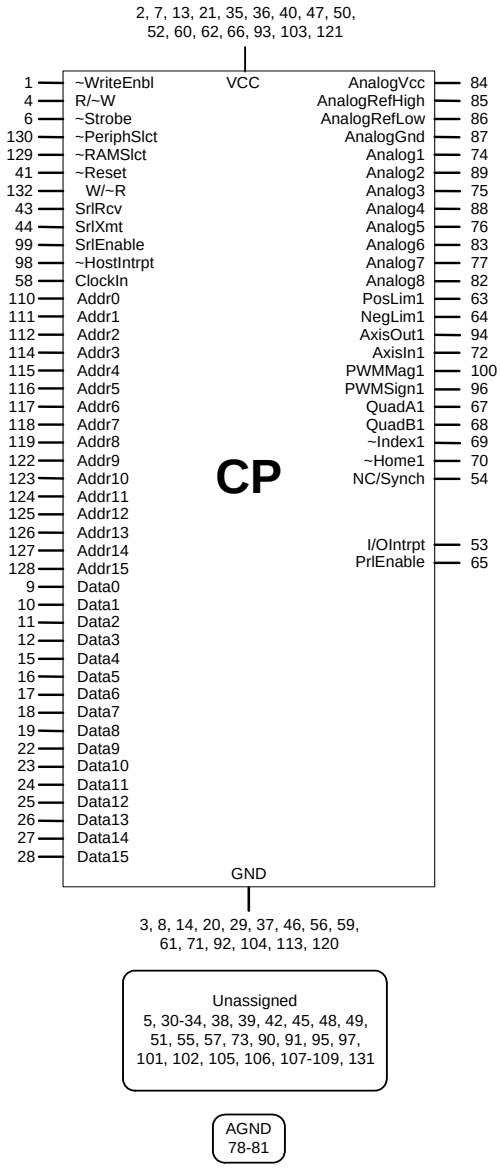


4.7.2 Peripheral device write



5 Pinouts and Pin Descriptions

5.1 Pinouts for MC3110



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 Addr1 Addr2 Addr3 Addr4 Addr5 Addr6 Addr7 Addr8 Addr9 Addr10 Addr11 Addr12 Addr13 Addr14 Addr15	110 111 112 114 115 116 117 118 119 122 123 124 125 126 127 128	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. 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.
ClockIn	58	input	This is the clock signal for the motion processor. It is driven at a nominal 20 MHz.
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 Analog2 Analog3 Analog4 Analog5 Analog6 Analog7 Analog8	74 89 75 88 76 83 77 82	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. The allowed range is AnalogRefLow to AnalogRefHigh. Any unused pins should be tied to AnalogGND. If the analog input circuitry is not used, these pins should be tied to GND.
PWMMag1	100	output	This pin provides the Pulse Width Modulated signal to the motor. This is the magnitude signal. The PWM resolution is 10 bits at a frequency of 20.0 KHz.
PWMSign1	96	output	This pin provides the sign (direction) of the PWM signal to the motor amplifier. This signal is *high* when the PWM output is positive, and *low* when it is negative.
QuadA1 QuadB1	67 68	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°. 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 processor 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 section 7.11 for an example schematic.**
~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 processor 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.
NC/Synch	54	input/output	On the MC3110 this pin is not used. On the MC3113, this pin is the synchronization signal. In the disabled mode, the pin is configured as an input and is not used. In the master mode, the pin outputs a synchronization pulse that can be used by slave nodes or other devices to synchronize with the internal chip cycle of the master node. In the slave mode, the pin is configured as an input and a pulse on the pin synchronizes the internal chip cycle.
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, 55, 73, 90, 91, 105, 106, 107, 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, 97, 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

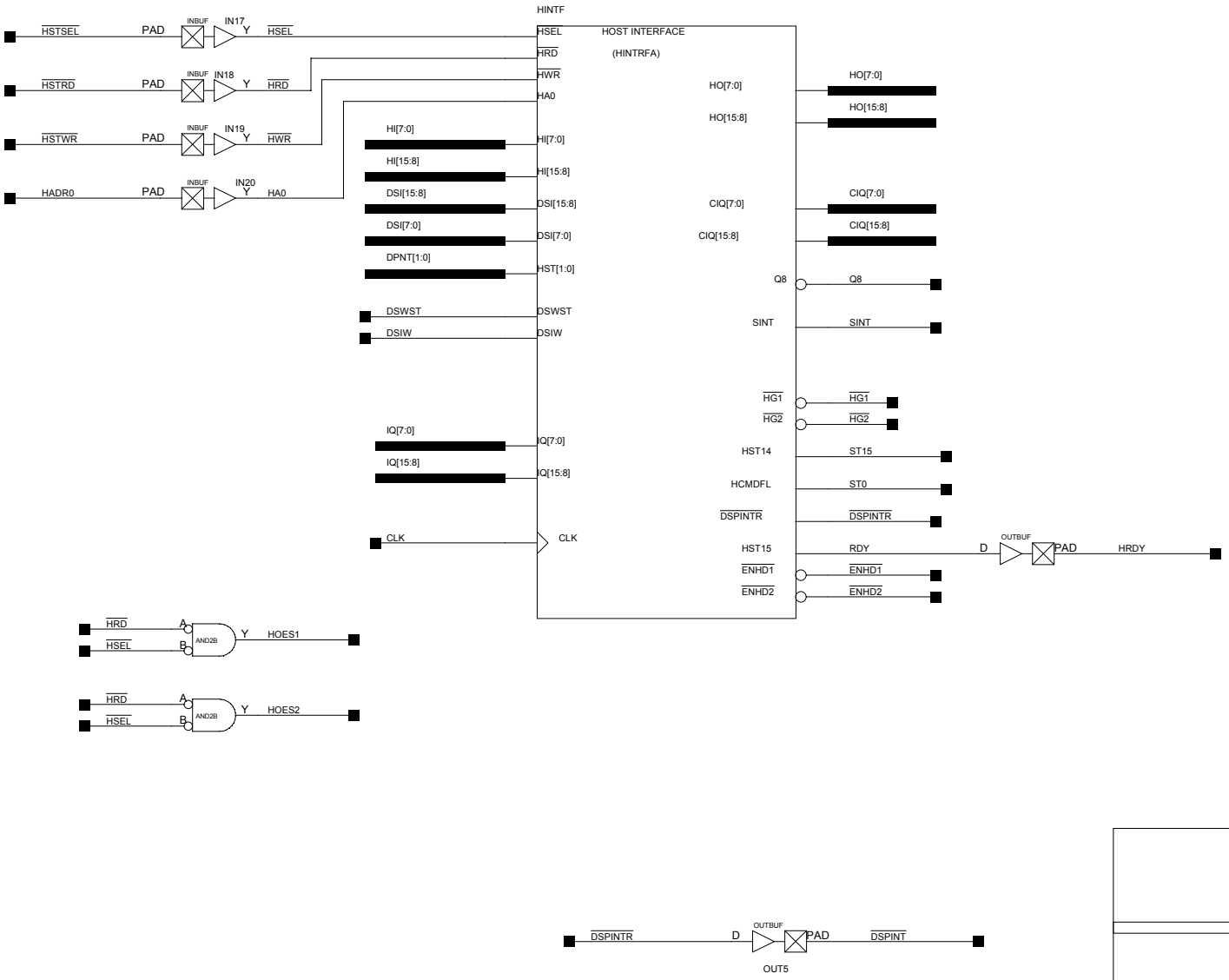
The CP write control is contained on schematic DSPWA. The CP interface uses page addressing to save I/O pins. F0, F1 and F2 make up the page register. In addition there are the 2 address bits, LA0 and LA1. A write to address 0 selects the page register with DSI[2:0] going to the page register and selecting 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.

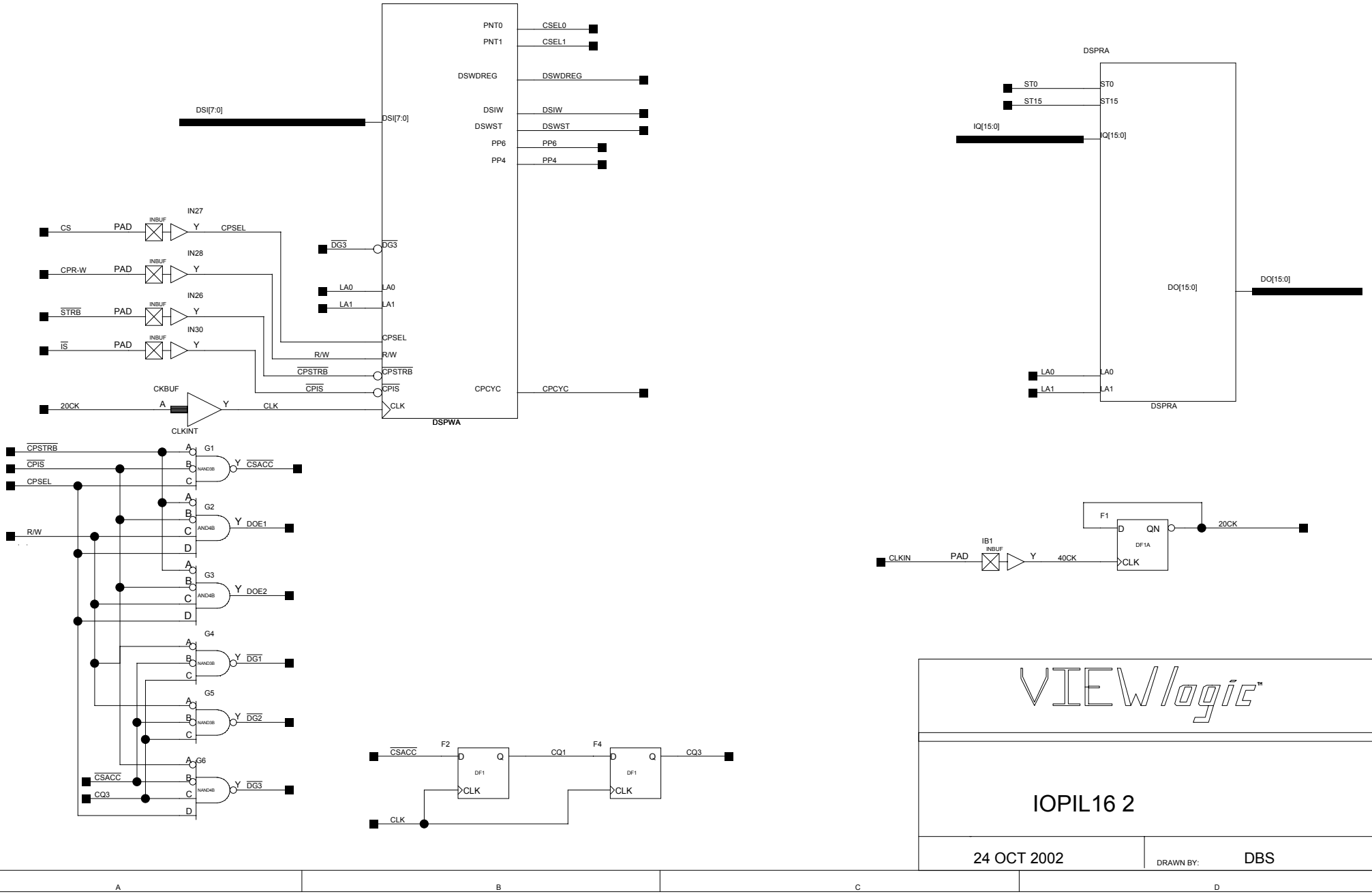


VIEWlogic™

IOPIL16 1

22 OCT 2002

DRAWN BY: DBS

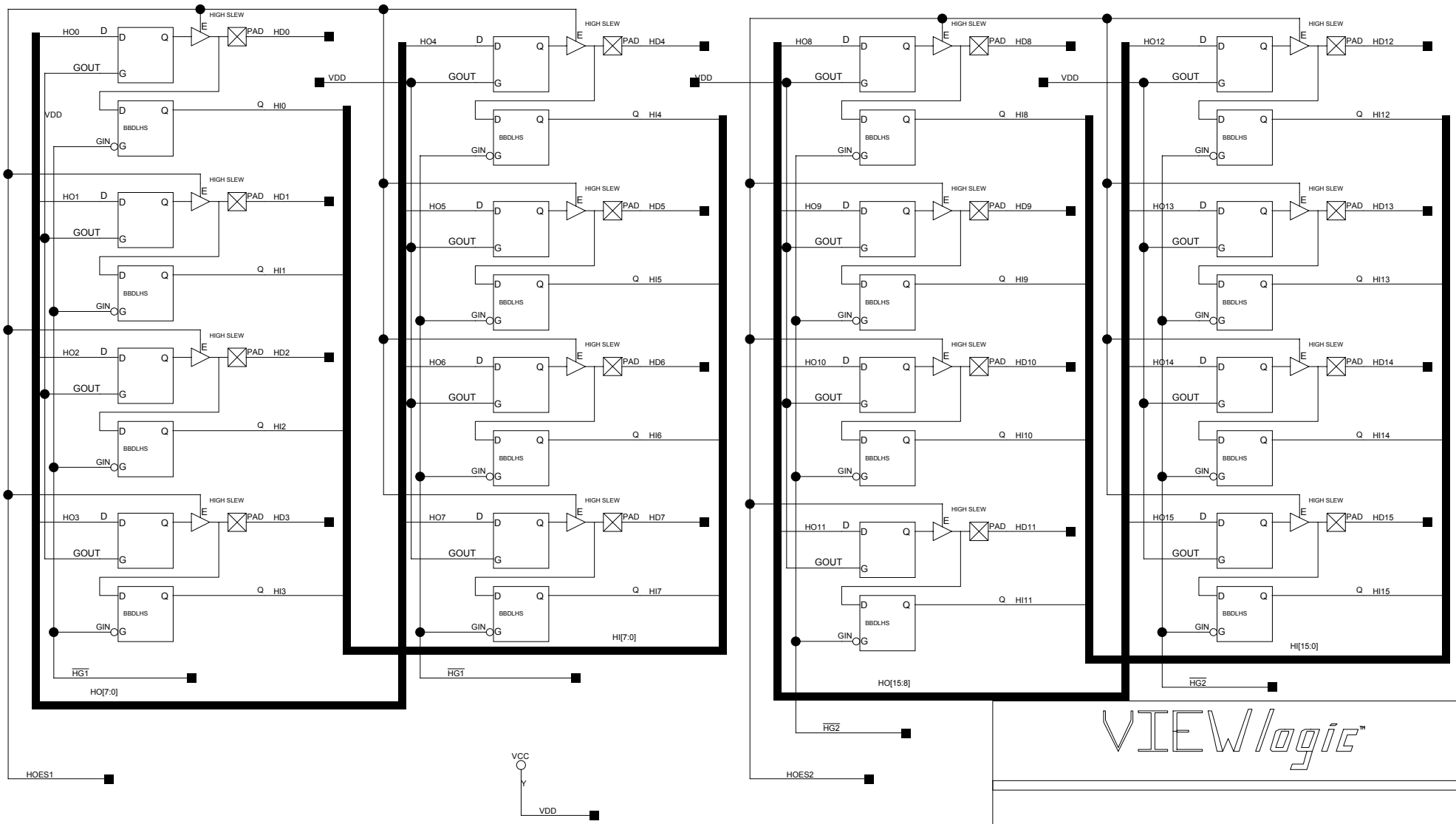


VIEWlogic™

IOPIL16 2

24 OCT 2002

DRAWN BY: DBS

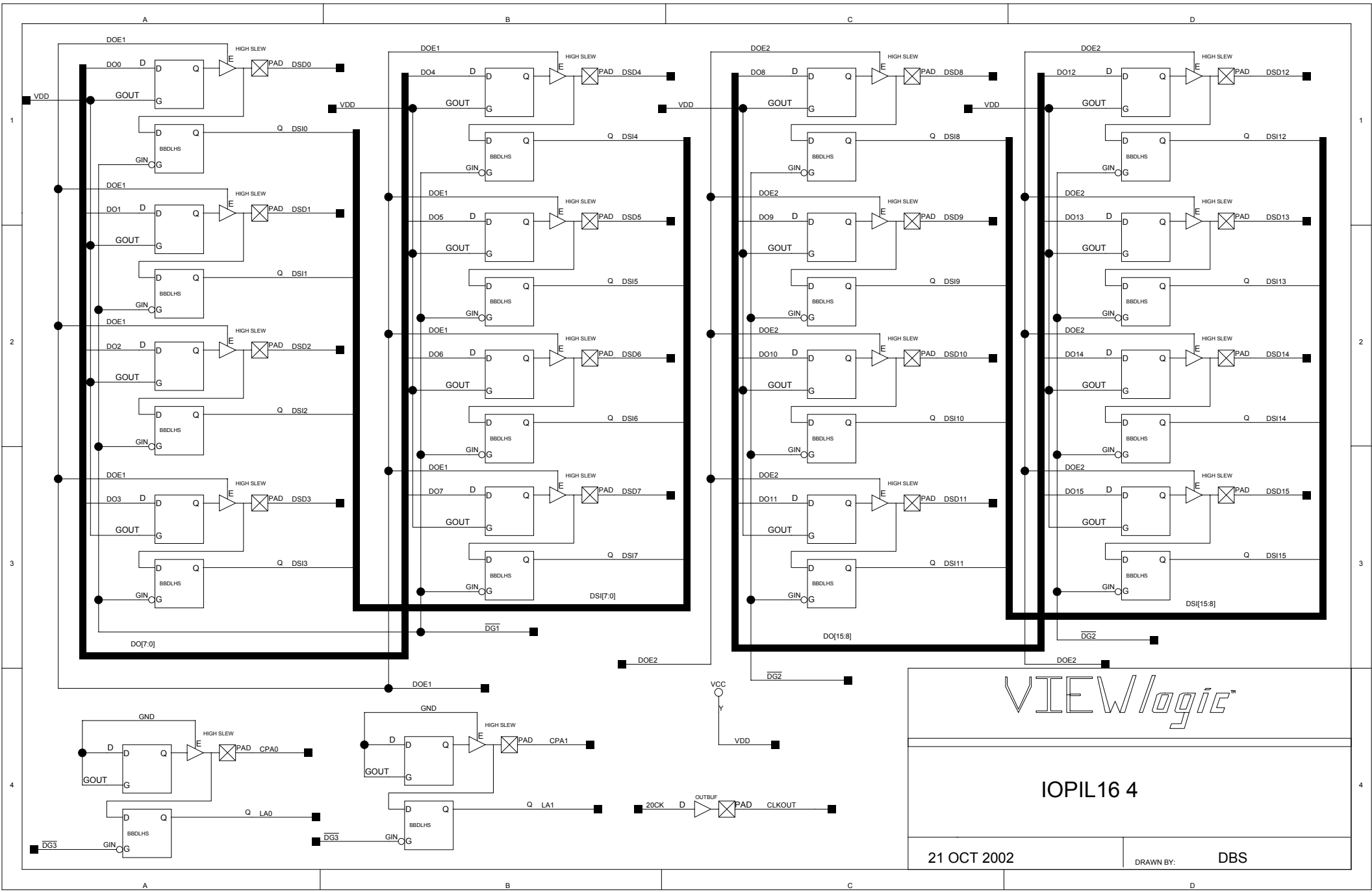


VIEWlogic™

IOPIL16 3

21 OCT 2002

DRAWN BY: DBS

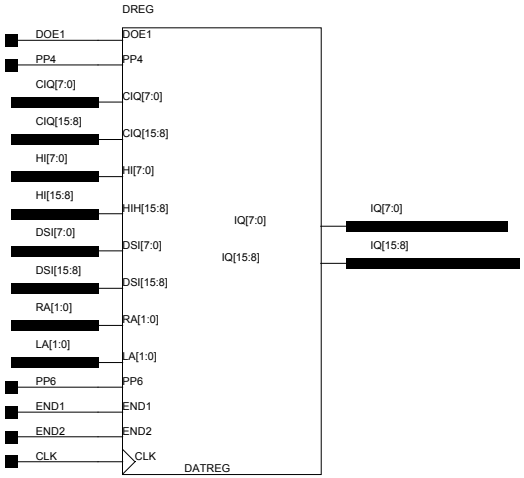


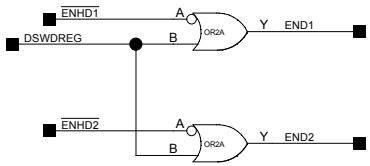
VIEWlogic™

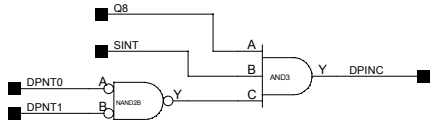
IOPIL16 4

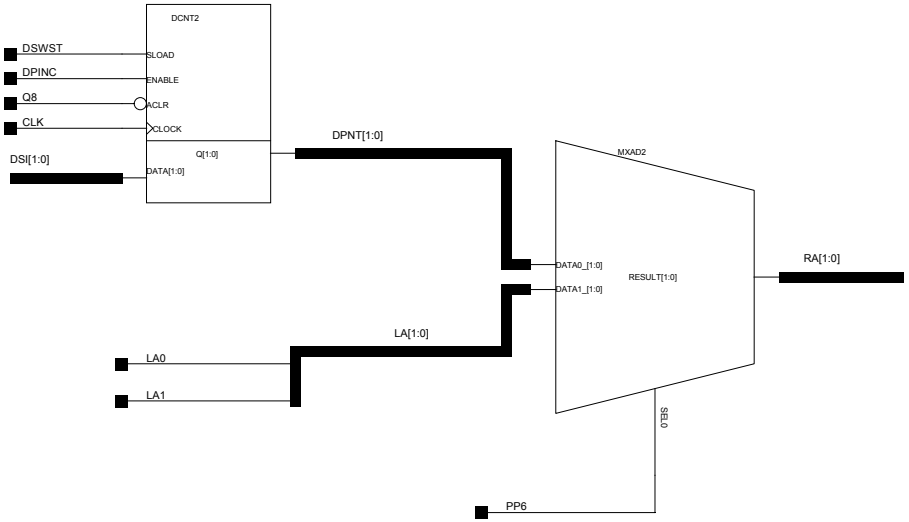
21 OCT 2002

DRAWN BY: DBS







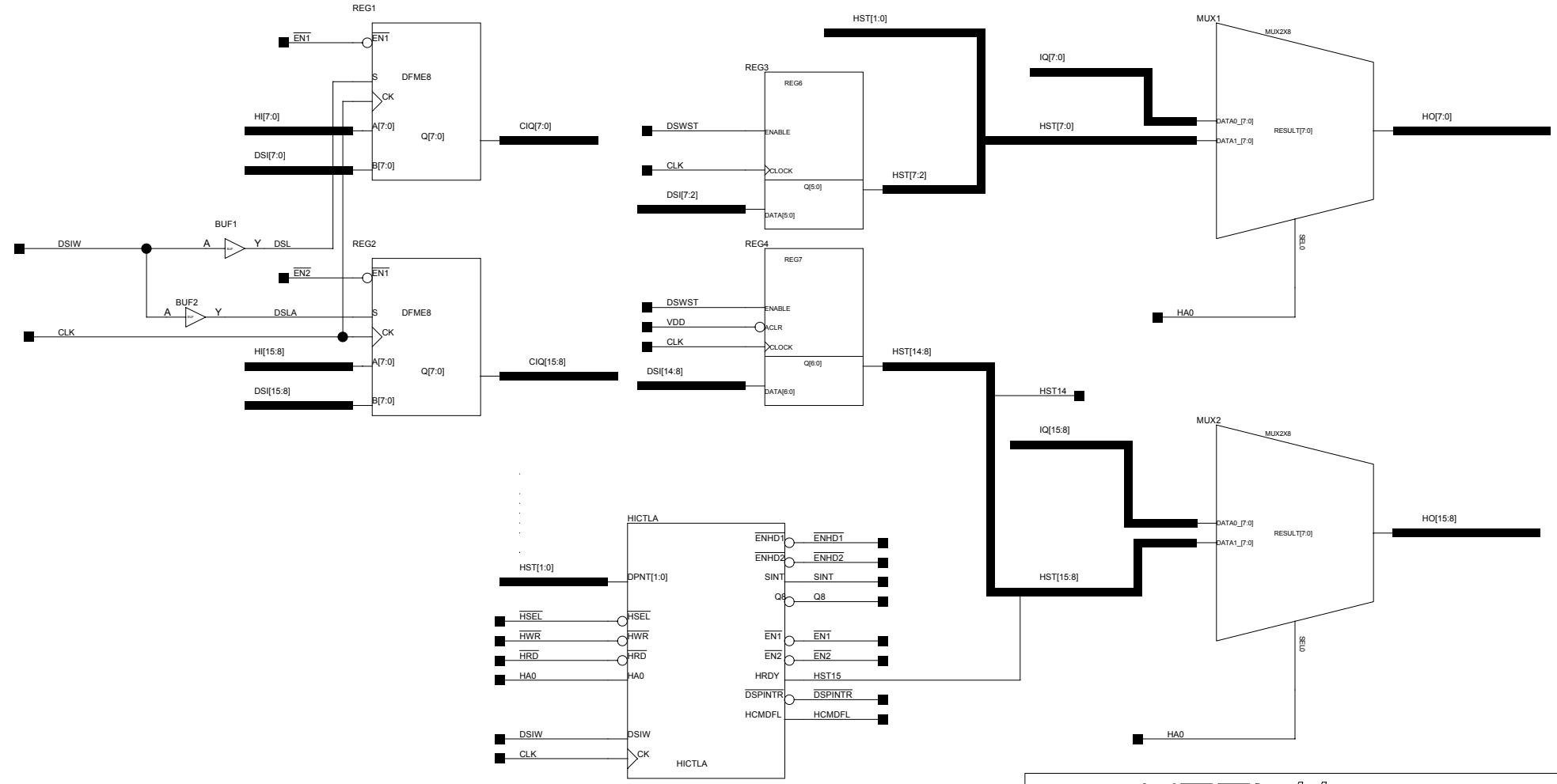


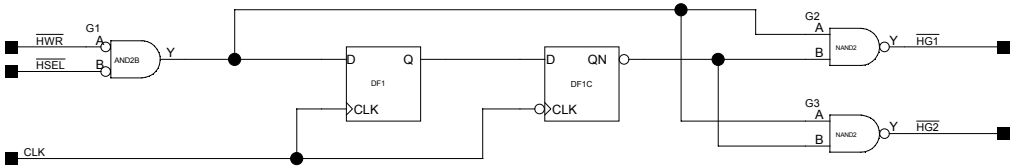
VIEWlogic™

IOPIL16 5

22 OCT 2002

DRAWN BY: DBS



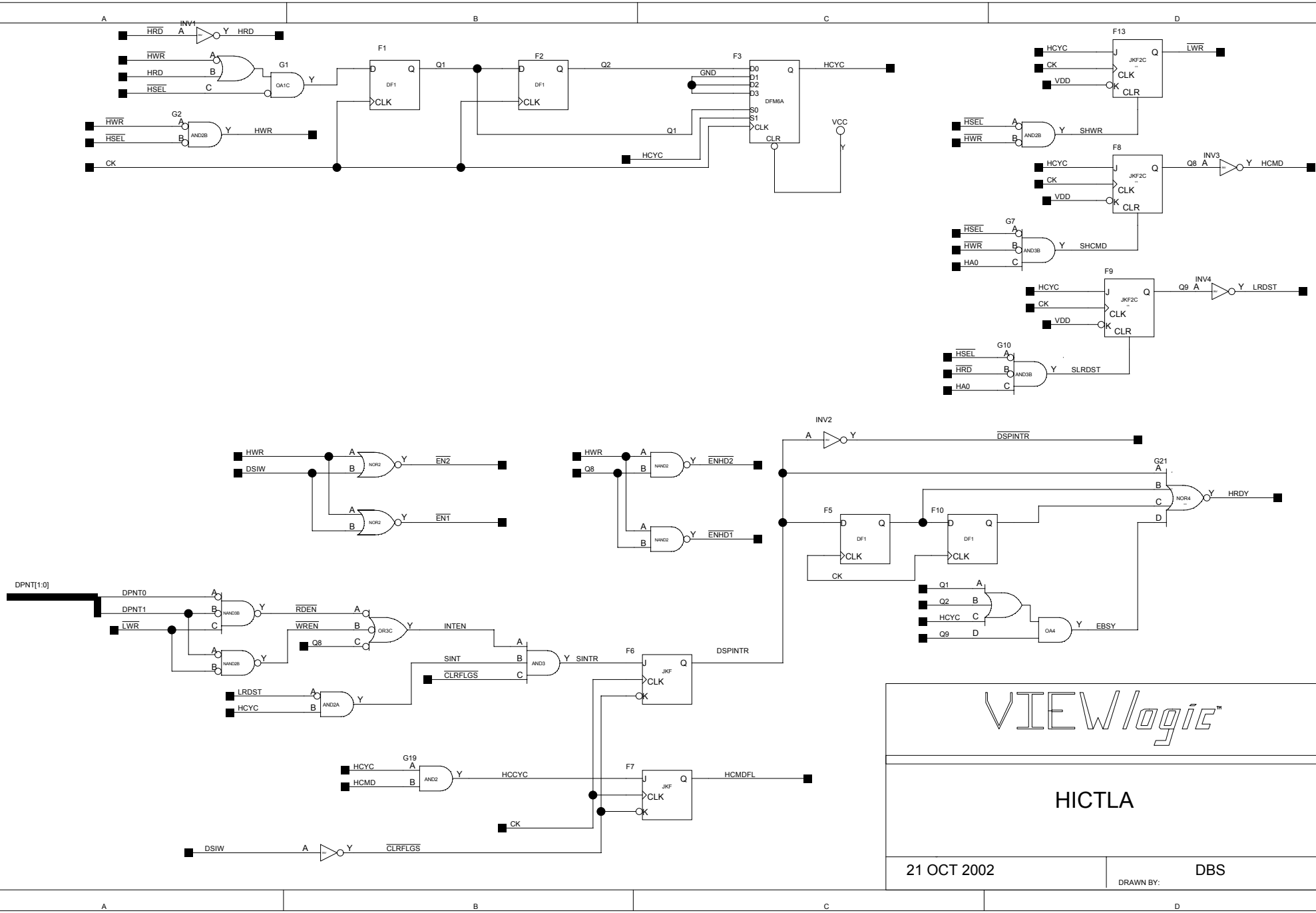


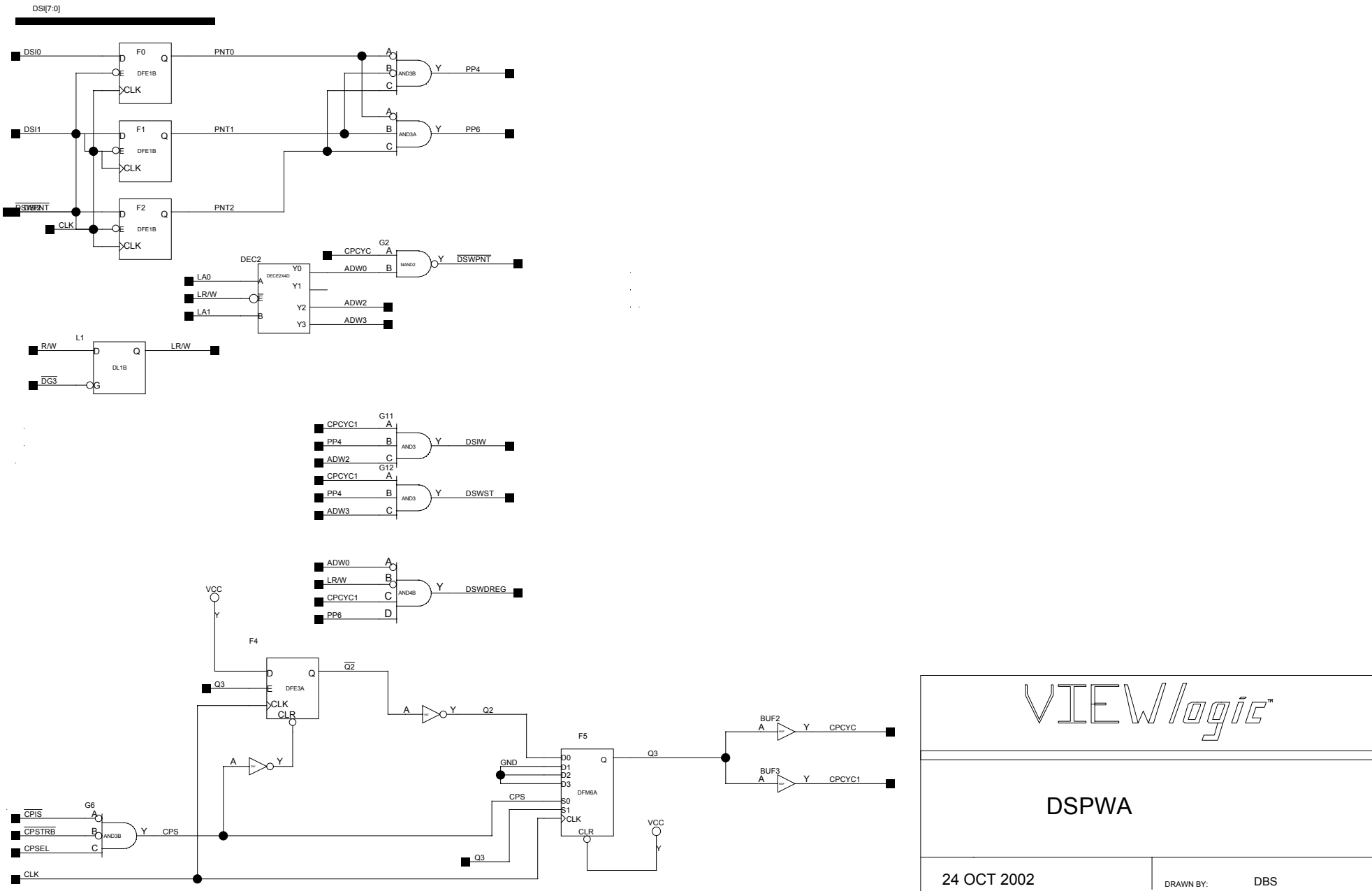
VIEWlogic™

HOST INTERFACE (HINTRFA)

24 OCT 2002

DRAWN BY: DBS


INV1
HRD
HRD
G1
HWR
HRD
HSEL
OA1C
F1
DF1
Q1
F2
DF1
Q2
F3
DFM8A
GND
HCYC
G2
HWR
HSEL
AND2B
HWR
CK
HCYC
VCC
F13
JKF2C
HCYC
CK
VDD
LWR
HSEL
HWR
AND2B
SHWR
F8
JKF2C
HCYC
CK
VDD
Q8
INV3
HCMD
G7
HSEL
HWR
HA0
AND3B
SHCMD
F9
JKF2C
HCYC
CK
VDD
Q9
INV4
LRDST
G10
HSEL
HRD
HA0
AND3B
SLRDST
HWR
DSIW
NOR2
EN2
NOR2
EN1
HWR
Q8
NAND2
ENHD2
NAND2
ENHD1
INV2
DSPINTR
G21
NOR4
HRDY
F5
DF1
F10
DF1
CK
Q1
Q2
HCYC
Q9
OA4
EBSY
DPNT[1:0]
DPNT0
DPNT1
LWR
NAND3B
NAND2B
RDEN
WREN
Q8
OR3C
INTEN
SINT
CLRFLGS
AND3
SINTR
F6
JKF
DSPINTR
LRDST
HCYC
AND2A
G19
HCYC
HCMD
AND2
HCCYC
F7
JKF
HCMDFL
CK
DSIW
CLRFLGS
VIEWlogic™
HICTLA
21 OCT 2002
DRAWN BY:
DBS

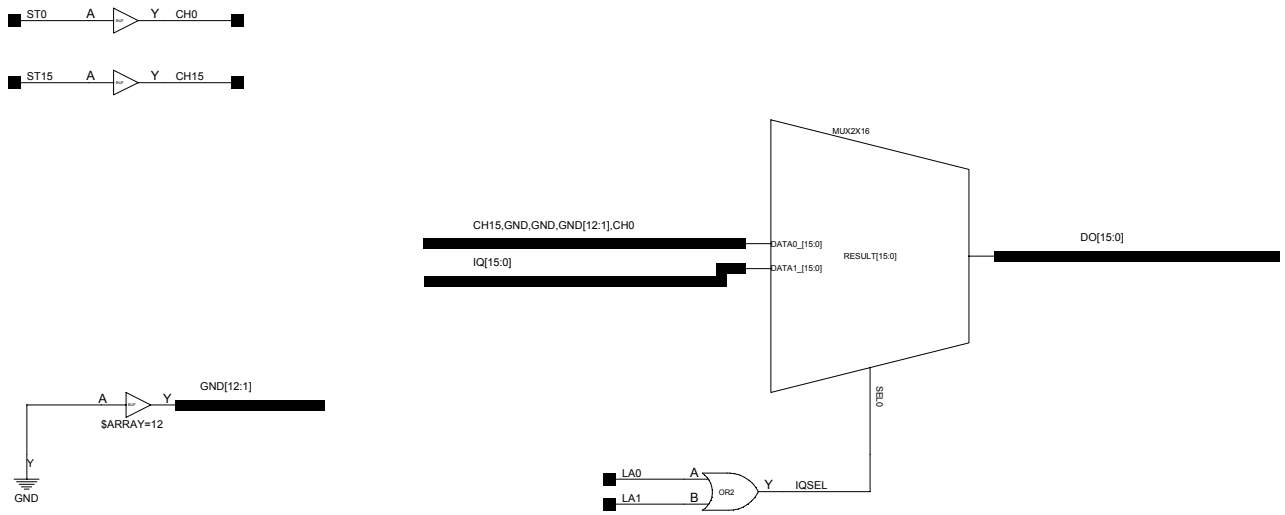


VIEWlogic™

DSPWA

24 OCT 2002

DRAWN BY: DBS

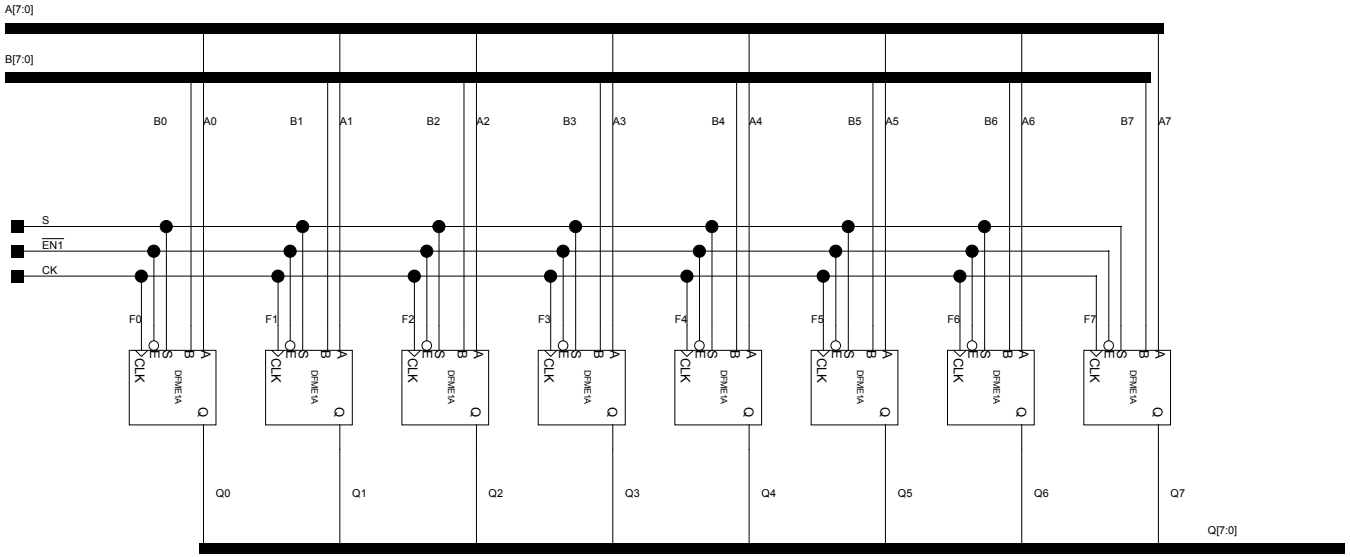


VIEWlogic™

DSPRA

24 OCT 2002

DRAWN BY: DBS

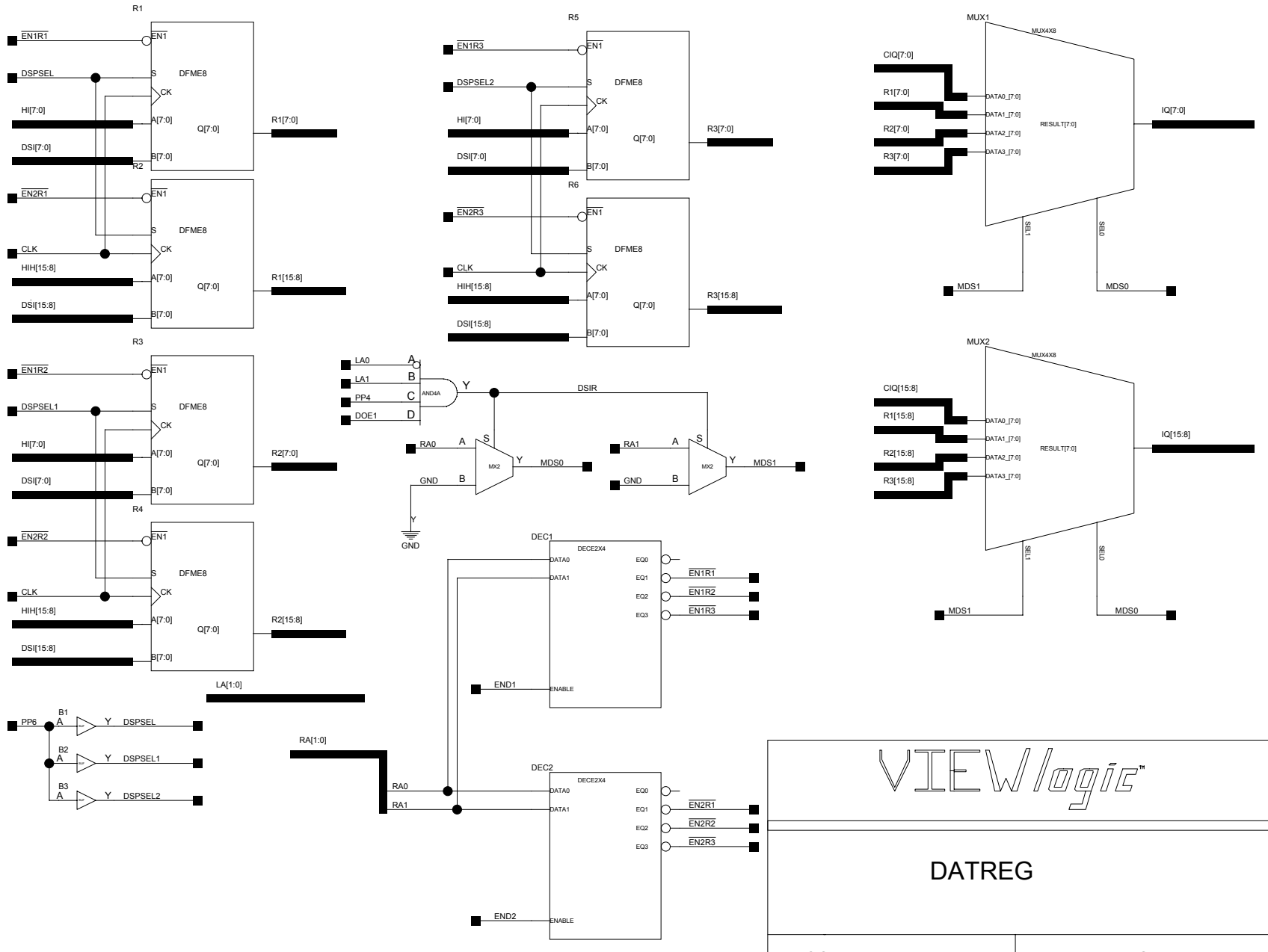


VIEWlogic™

DFME8

19 NOV. 2002

DRAWN BY: DBS



VIEWlogic™

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.

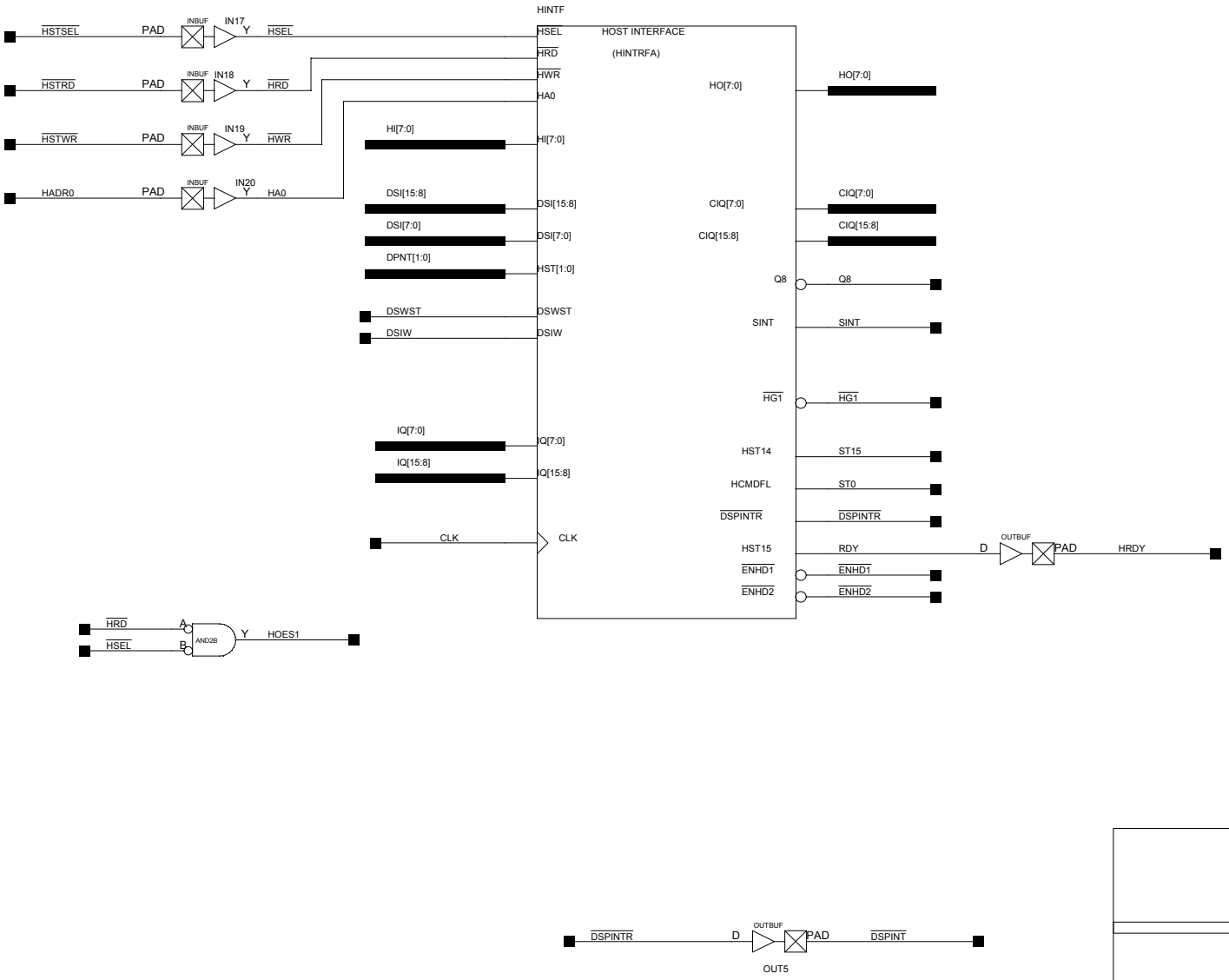
DSPRA

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.

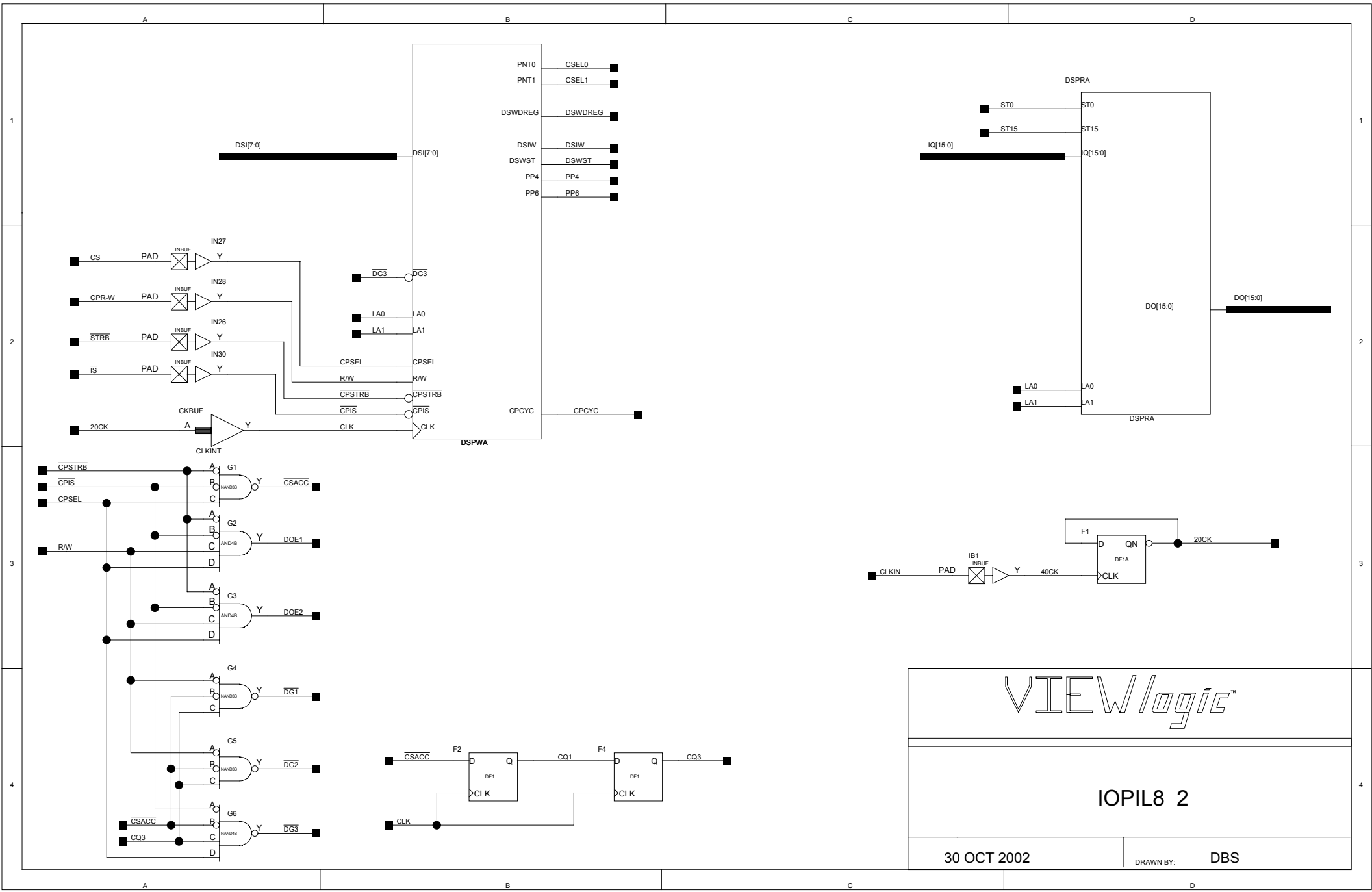


VIEWlogic™

IOPIL8 1

22 OCT 2002

DRAWN BY: DBS

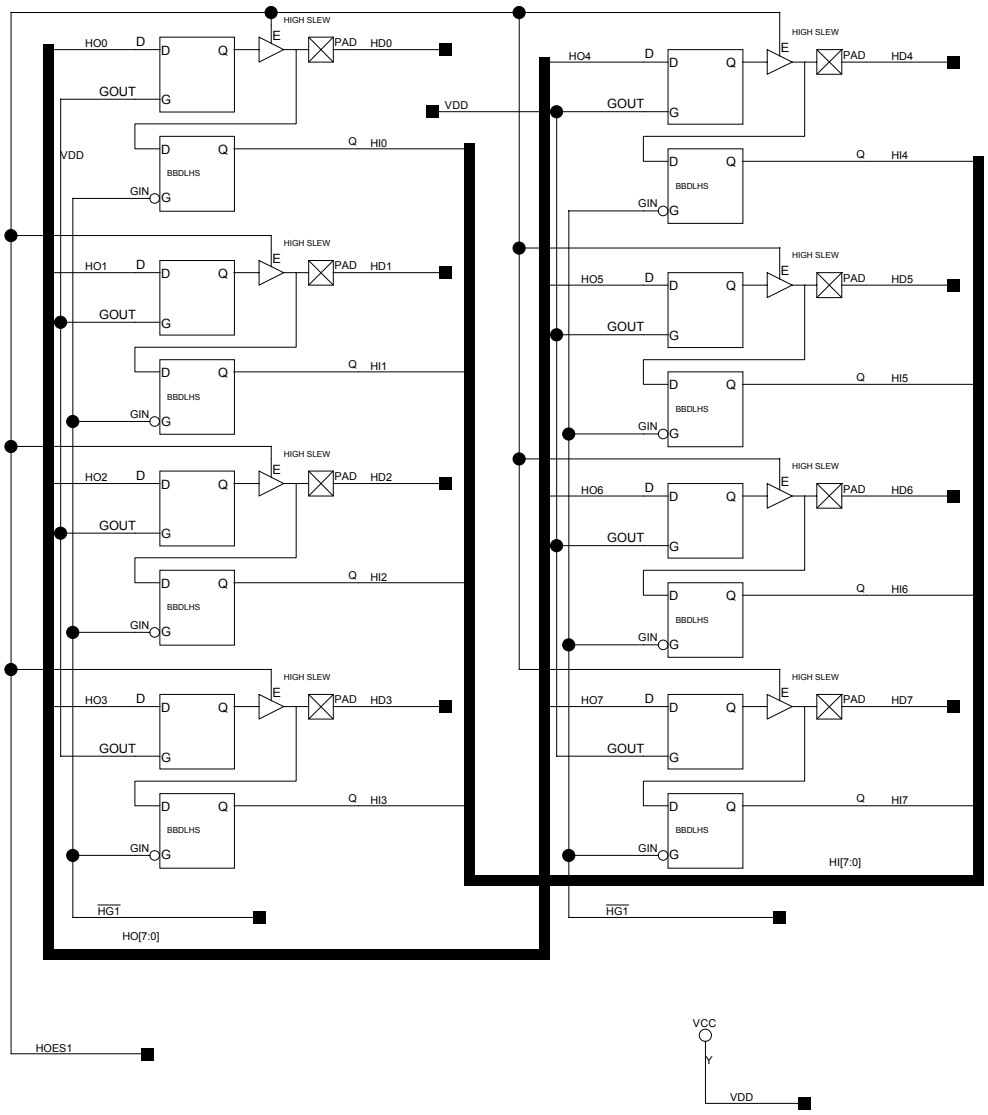


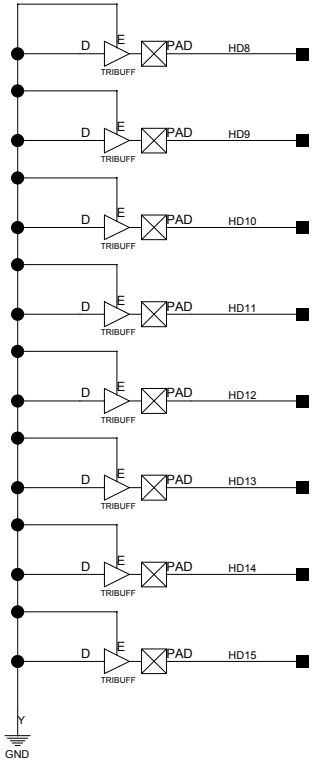
VIEWlogic™

IOPIL8 2

30 OCT 2002

DRAWN BY: DBS





HI BYTE TRISTATE TO

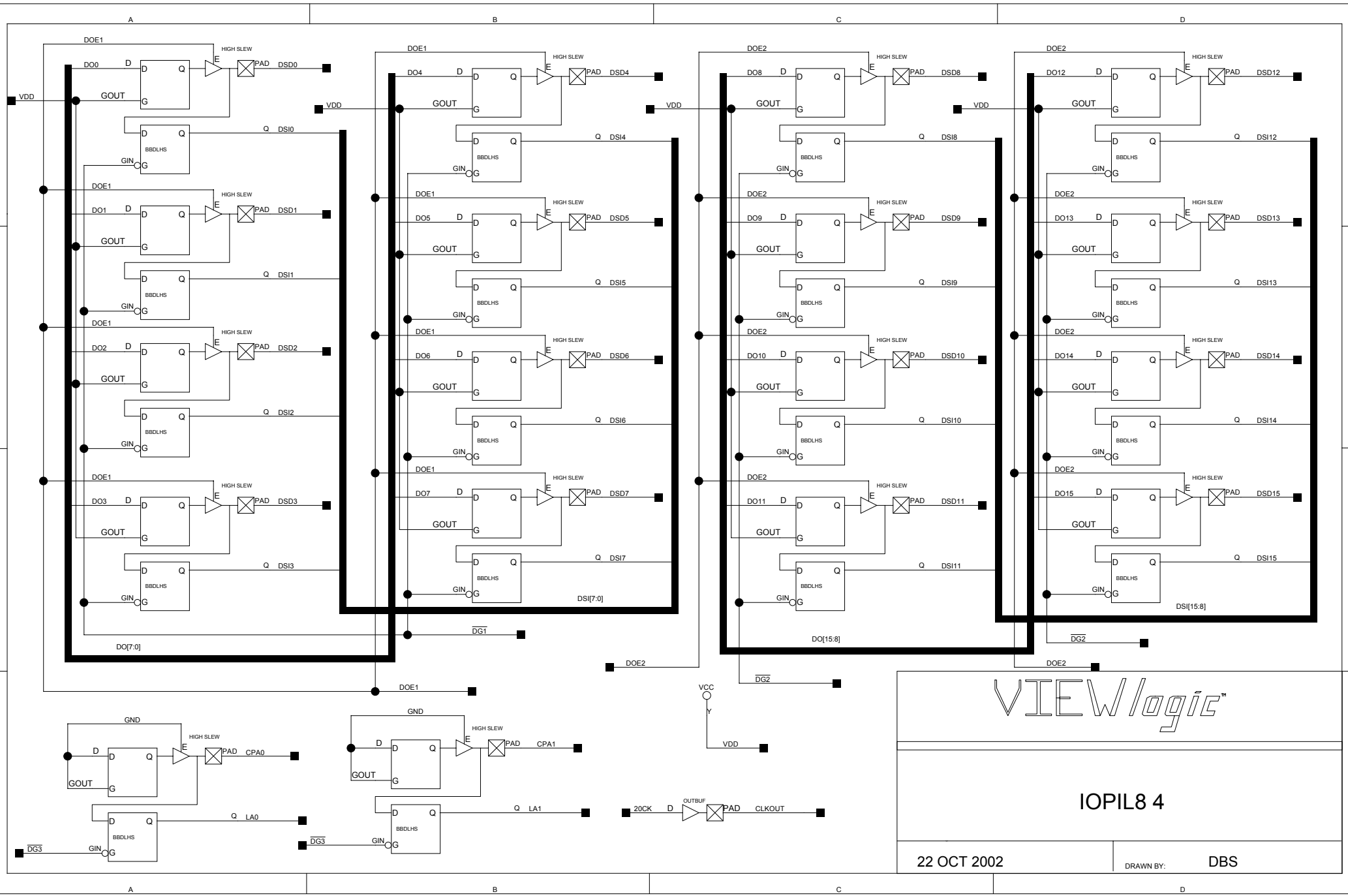
AVOID LOADING 16 BIT BUSSES

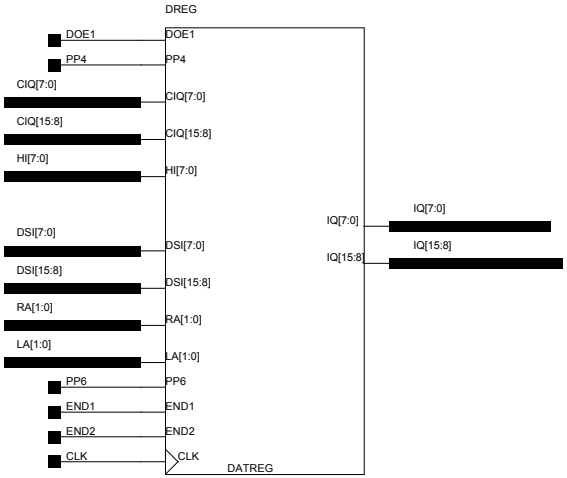
VIEWlogic™

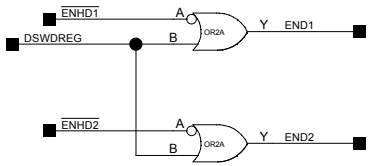
IOPIL8 3

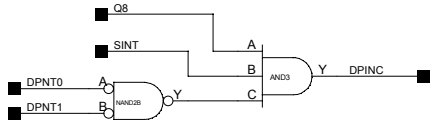
24 OCT 2002

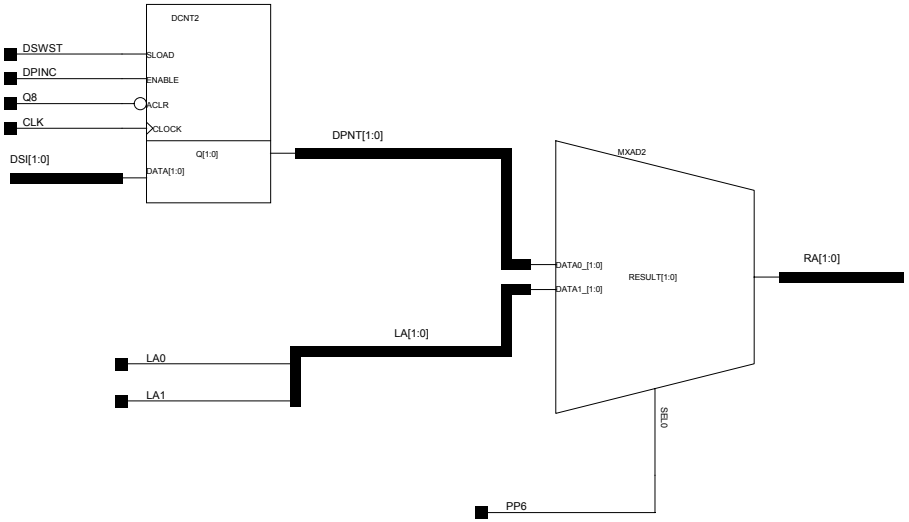
DRAWN BY: DBS


DOE1
DO0
D
Q
E
HIGH SLEW
PAD
DSD0
VDD
GOUT
G
Q
DSI0
BBDLHS
GIN
DO1
DSD1
DSI1
DO2
DSD2
DSI2
DO3
DSD3
DSI3
DO[7:0]
DO4
DSD4
DSI4
DO5
DSD5
DSI5
DO6
DSD6
DSI6
DO7
DSD7
DSI7
DSI[7:0]
DG1
DOE2
DO8
DSD8
DSI8
DO9
DSD9
DSI9
DO10
DSD10
DSI10
DO11
DSD11
DSI11
DO[15:8]
DO12
DSD12
DSI12
DO13
DSD13
DSI13
DO14
DSD14
DSI14
DO15
DSD15
DSI15
DSI[15:8]
DG2
VCC
Y
GND
CPA0
CPA1
LA0
LA1
DG3
20CK
OUTBUF
CLKOUT
VIEWlogic™
IOPIL8 4
22 OCT 2002
DRAWN BY:
DBS







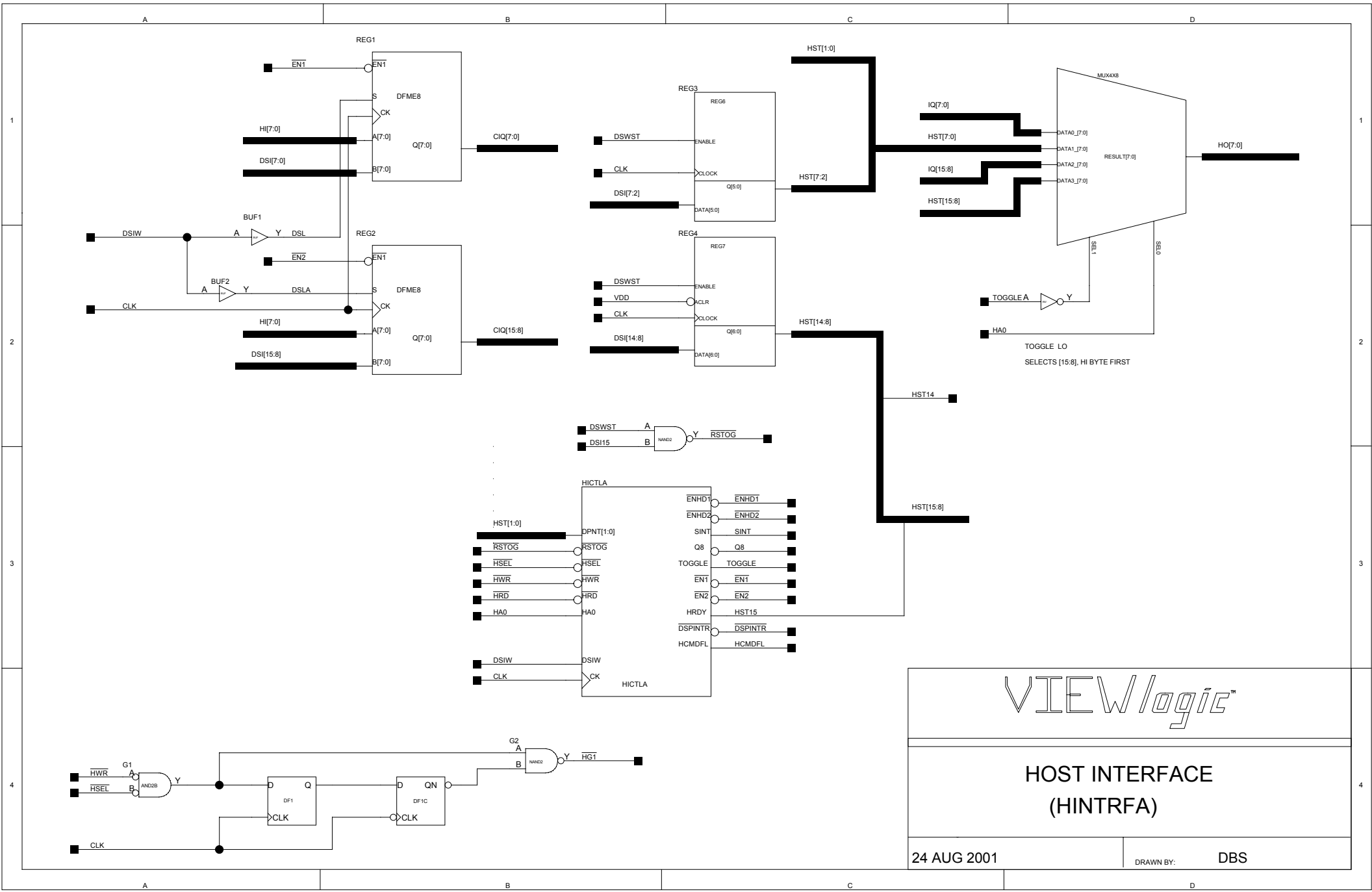


VIEWlogic™

IOPIL8 5

22 OCT 2002

DRAWN BY: DBS

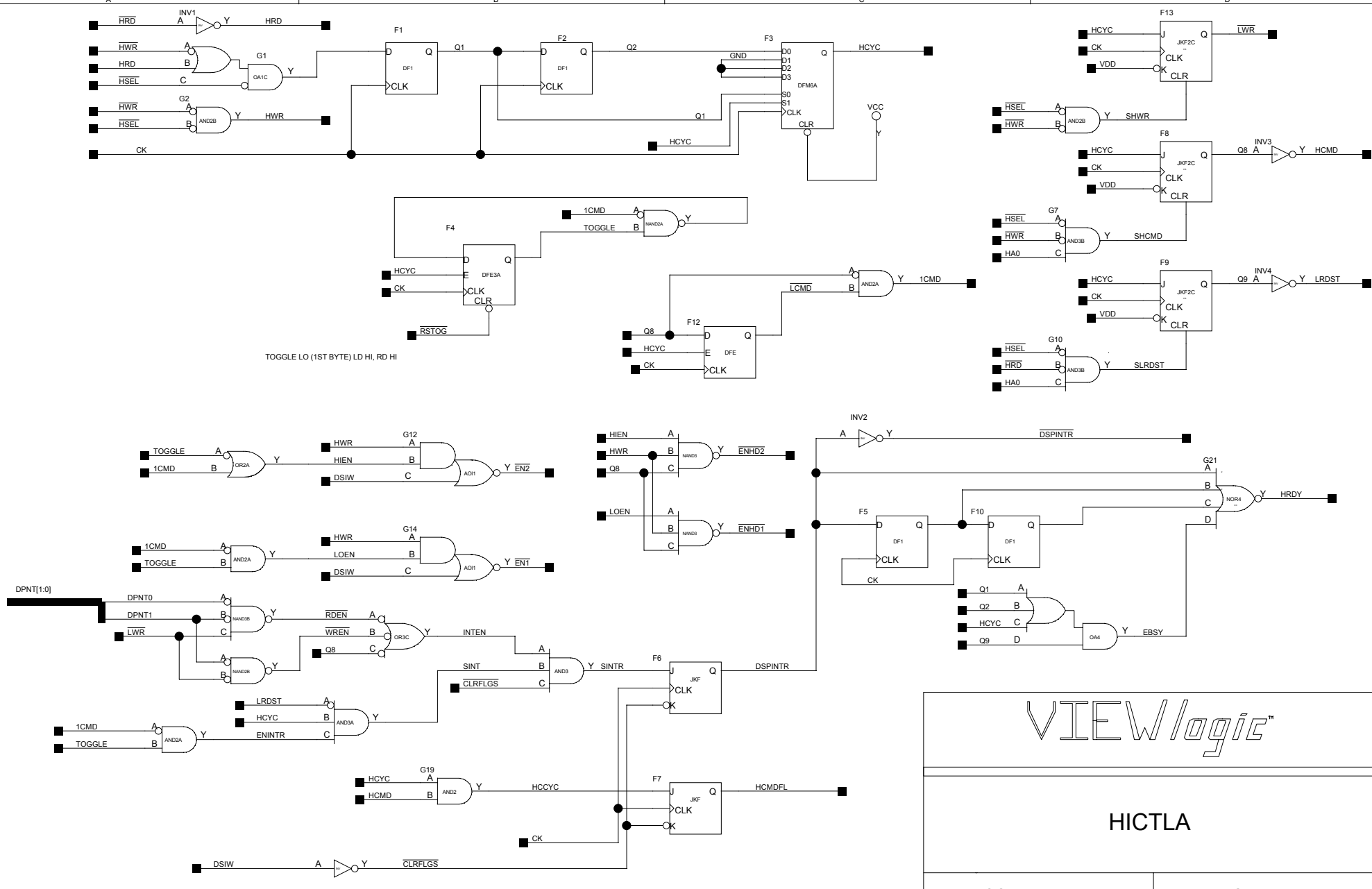


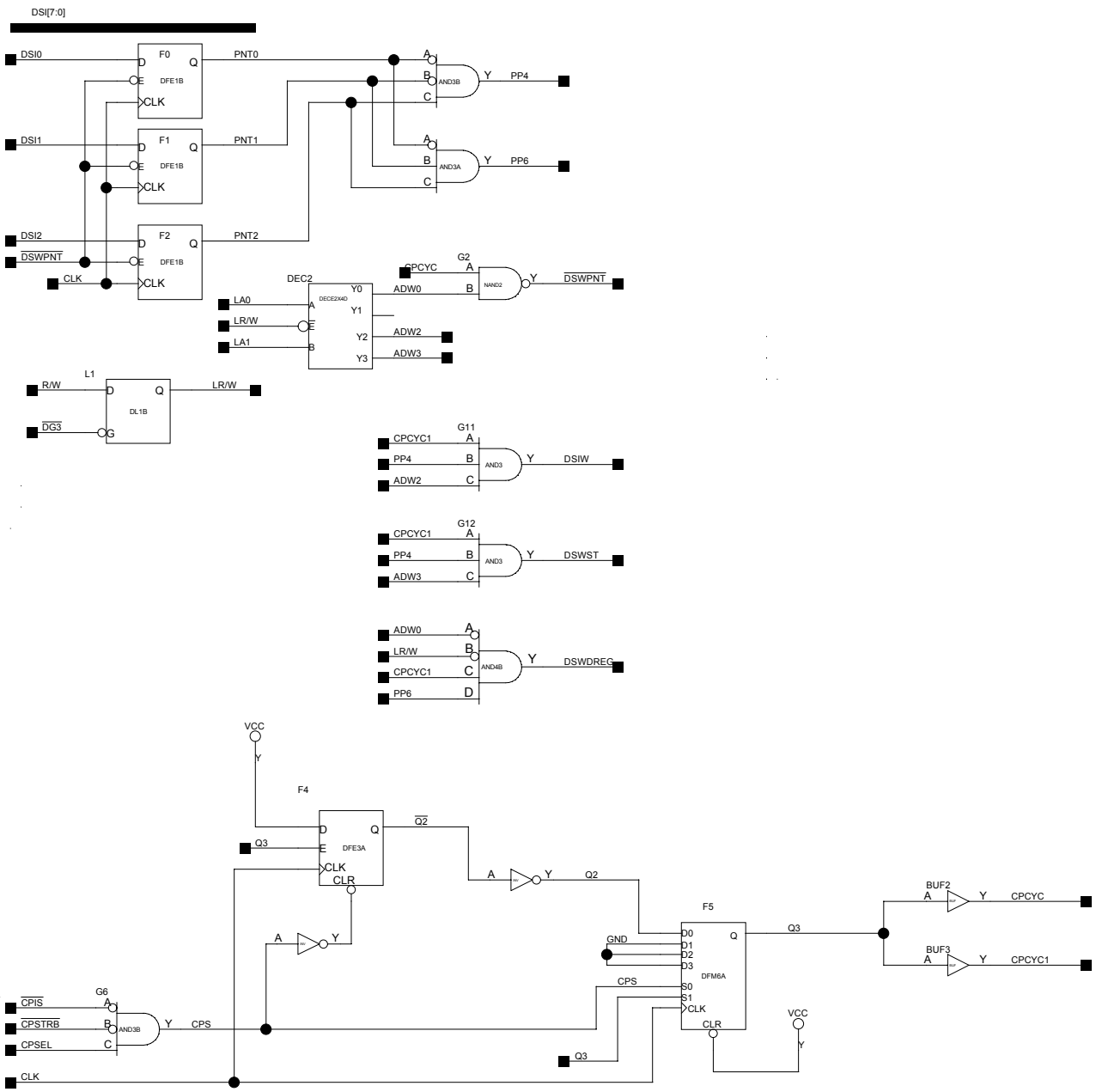
VIEWlogic™

HOST INTERFACE (HINTRFA)

24 AUG 2001

DRAWN BY: DBS


INV1
HRD
HRD
HWR
HRD
HSEL
G1
OA1C
HWR
HSEL
G2
AND2B
HWR
CK
F1
DF1
Q1
F2
DF1
Q2
F3
DFM8A
GND
D0
D1
D2
D3
S0
S1
CLK
CLR
Q
HCYC
VCC
Y
Q1
HCYC
F13
JKF2C
HCYC
CK
VDD
LWR
HSEL
HWR
AND2B
SHWR
F8
JKF2C
HCYC
CK
VDD
Q8
INV3
HCMD
HSEL
HWR
HA0
G7
AND3B
SHCMD
F9
JKF2C
HCYC
CK
VDD
Q9
INV4
LRDST
HSEL
HRD
HA0
G10
AND3B
SLRDST
F4
DFE3A
1CMD
TOGGLE
NAND2A
HCYC
CK
RSTOG
TOGGLE LO (1ST BYTE) LD HI, RD HI
F12
DFE
Q8
HCYC
CK
LCMD
AND2A
1CMD
TOGGLE
LO (1ST BYTE) LD HI, RD HI
INV2
DSPINTR
HIEN
HWR
Q8
NAND3
ENHD2
LOEN
NAND3
ENHD1
TOGGLE
1CMD
OR2A
G12
HWR
HIEN
DSIW
AOI1
EN2
1CMD
TOGGLE
AND2A
G14
HWR
LOEN
DSIW
AOI1
ENT
DPNT[1:0]
DPNT0
DPNT1
LWR
NAND3B
NAND2B
RDEN
WREN
Q8
OR3C
INTEN
SINT
CLRFLGS
AND3
SINTR
F6
JKF
DSPINTR
LRDST
HCYC
ENINTR
AND3A
1CMD
TOGGLE
AND2A
F5
DF1
F10
DF1
CK
G21
NOR4
HRDY
Q1
Q2
HCYC
Q9
OA4
EBSY
HCYC
HCMD
G19
AND2
HCCYC
F7
JKF
HCMDFL
CK
DSIW
CLRFLGS
VIEWlogic™
HICTLA
22 OCT 2002
DRAWN BY: DBS

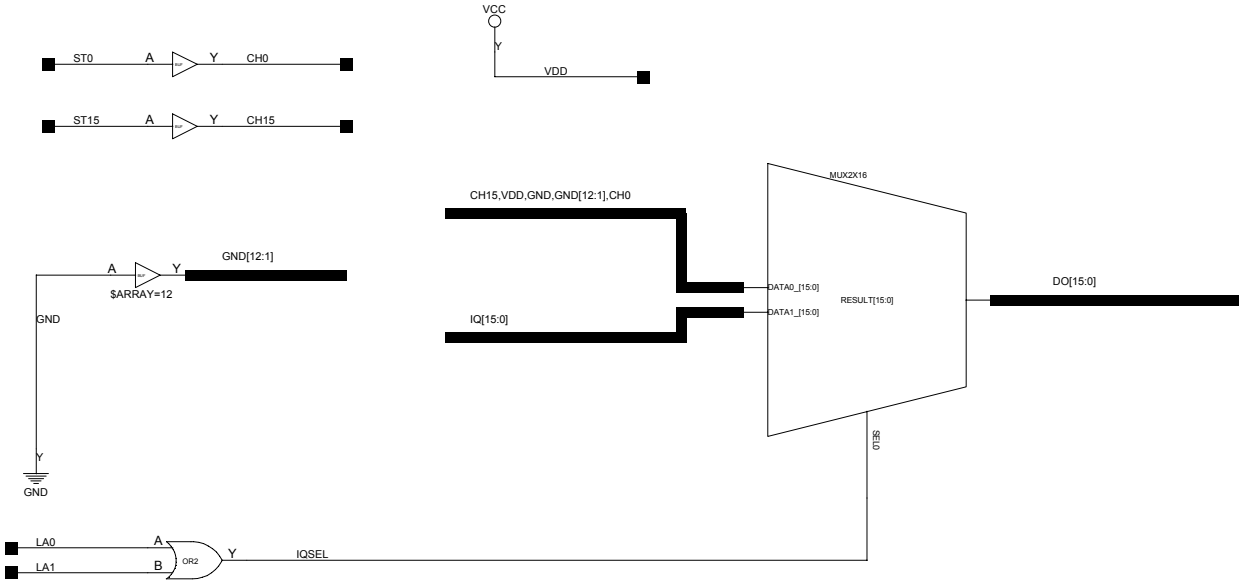


VIEWlogic™

DSPWA

24 OCT 2002

DRAWN BY: DBS

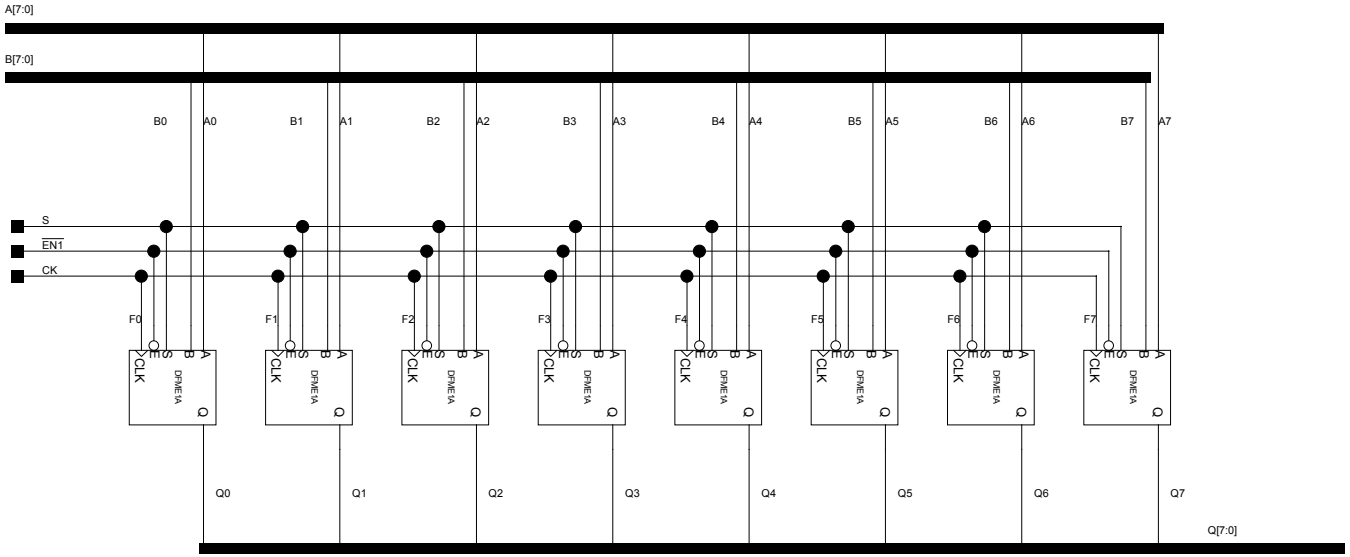


VIEWlogic™

DSPRA

30 OCT 2002

DRAWN BY: DBS

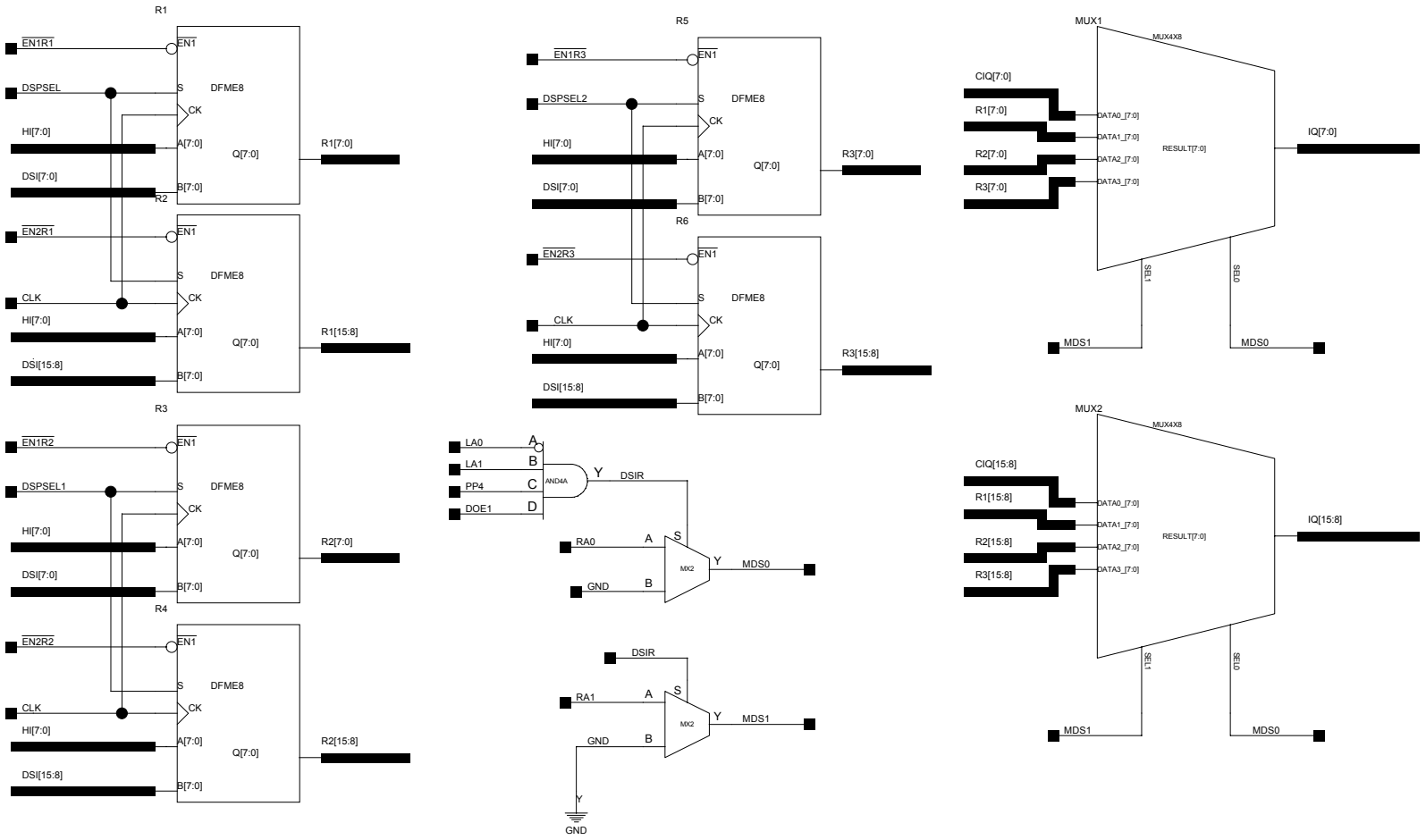


VIEWlogic™

DFME8

19 NOV. 2002

DRAWN BY: DBS



VIEWlogic™

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 PWM 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 PWM 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
- PWM carrier frequency
- 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:-

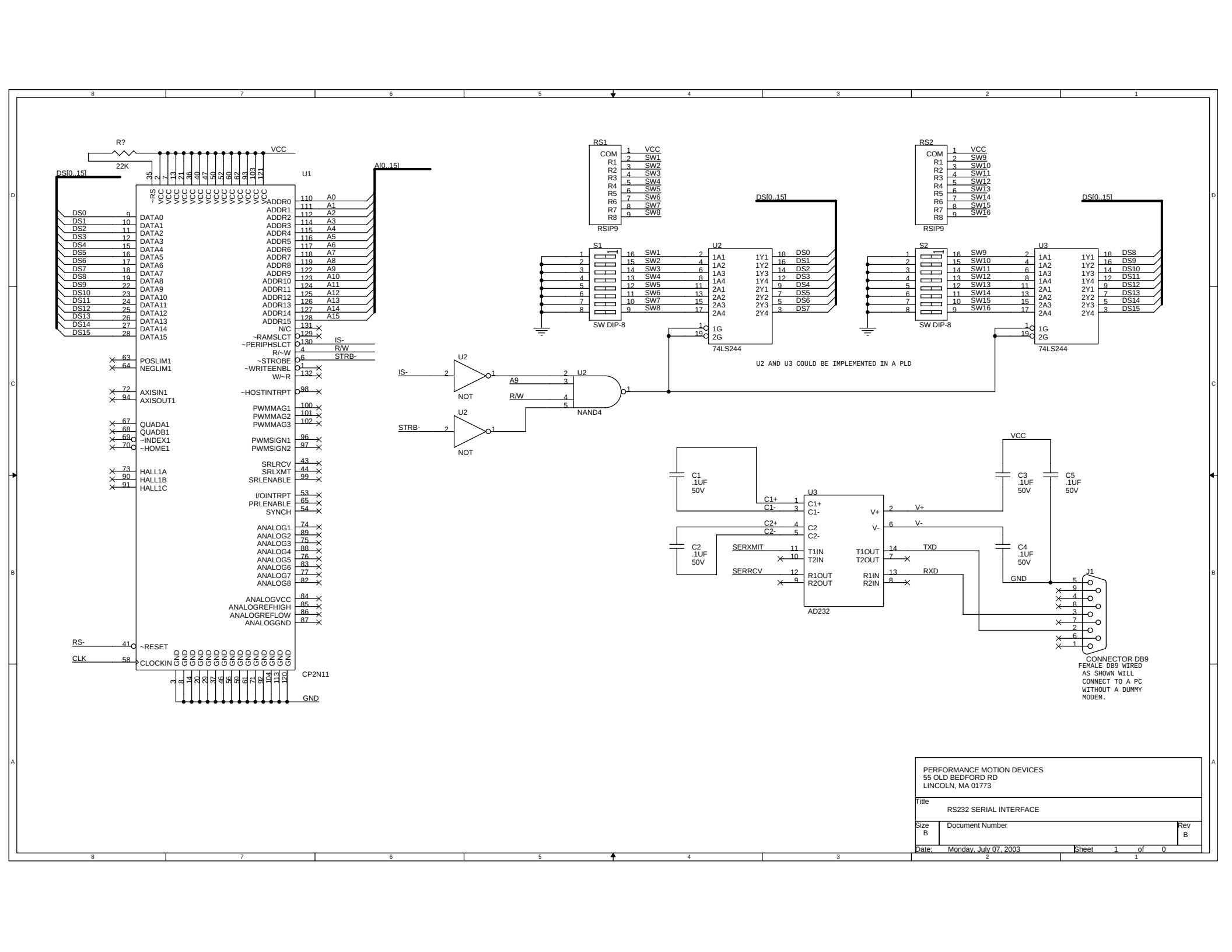
- Serial baud rate decreases to 9600 bps *17/20 = 8160 bps
- PWM frequency decreases to 20 KHz *17/20 = 17 KHz
- Cycle time increases to 102.4 μsec *20/17 = 120.48 μsec

7.2 RS-232 Serial Interface

The interface between the MC3110 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.



PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title: RS232 SERIAL INTERFACE

Size: B | Document Number | Rev: B

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

7.3 RS 422/485 Serial Interface

The interface between the MC3110 chip and an RS-422/485 serial port is shown in the following figure.

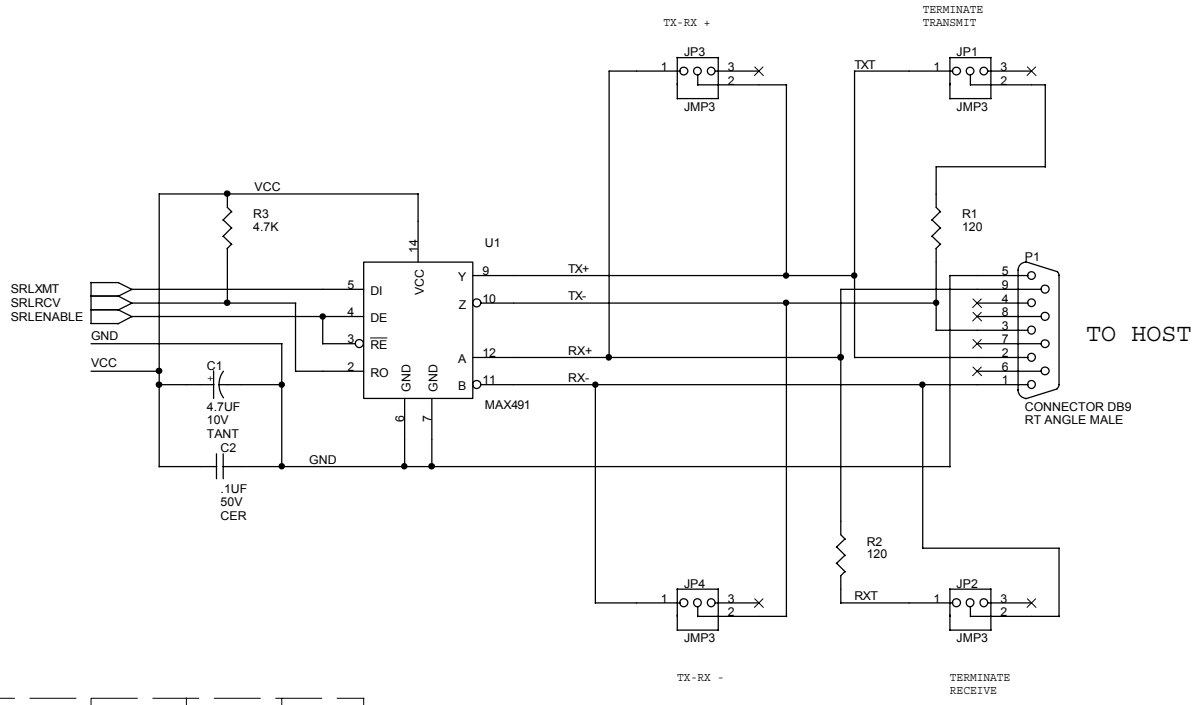
Comments on Schematic

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.



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

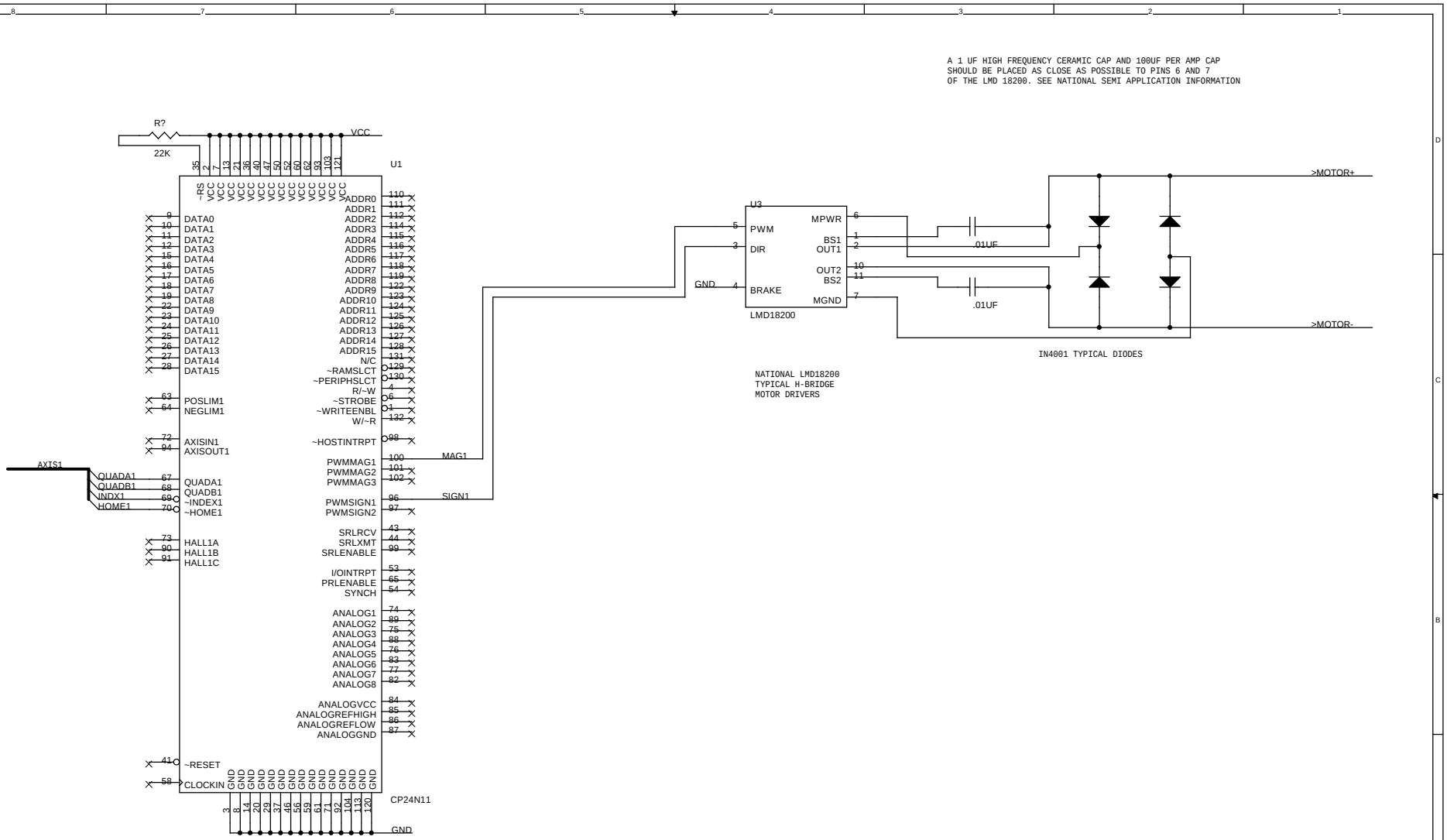
Date: Thursday, April 11, 2002 | Sheet 1 of 1

7.4 PWM Motor Interface

The following schematic shows a typical interface circuit between the MC3110 and an amplifier in PWM output mode.

Comments on Schematic

The LMD18200 H-bridge driver is used.

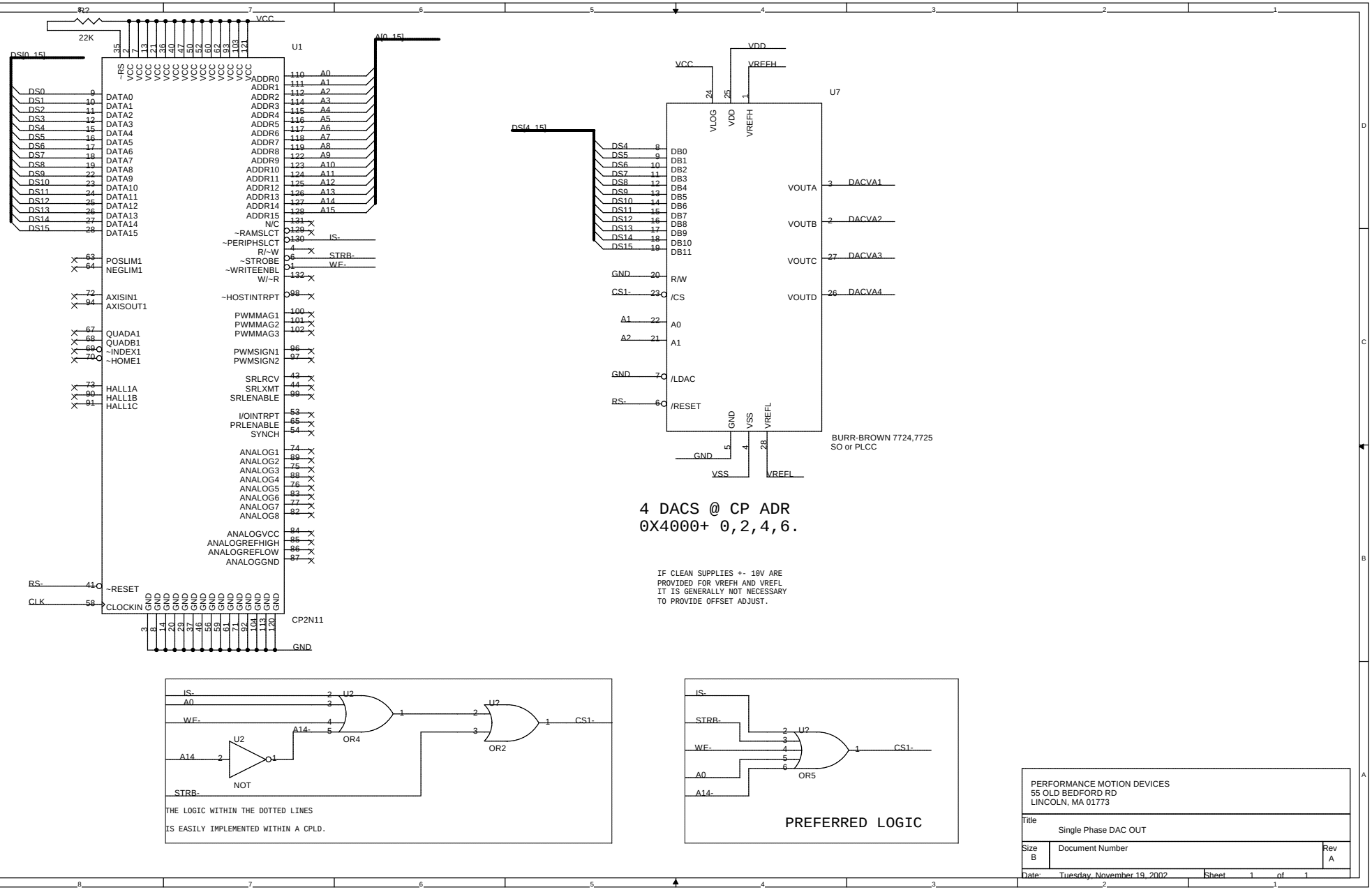

A 1 UF HIGH FREQUENCY CERAMIC CAP AND 100UF PER AMP CAP
SHOULD BE PLACED AS CLOSE AS POSSIBLE TO PINS 6 AND 7
OF THE LMD 18200. SEE NATIONAL SEMI APPLICATION INFORMATION
R?
22K
VCC
U1
~RS
VCC
GND
ADDR0
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
N/C
~RAMSLCT
~PERIPHSLCT
R/~W
~STROBE
~WRITEENBL
W/~R
~HOSTINTRPT
PWMMAG1
PWMMAG2
PWMMAG3
PWMSIGN1
PWMSIGN2
SRLRCV
SRLXMT
SRLENABLE
I/OINTRPT
PRLENABLE
SYNCH
ANALOG1
ANALOG2
ANALOG3
ANALOG4
ANALOG5
ANALOG6
ANALOG7
ANALOG8
ANALOGVCC
ANALOGREFHIGH
ANALOGREFLOW
ANALOGGND
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA14
DATA15
POSLIM1
NEGLIM1
AXISIN1
AXISOUT1
QUADA1
QUADB1
INDX1
HOME1
~INDEX1
~HOME1
HALL1A
HALL1B
HALL1C
~RESET
CLOCKIN
CP24N11
MAG1
SIGN1
AXIS1
U3
PWM
DIR
BRAKE
MPWR
BS1
OUT1
OUT2
BS2
MGND
LMD18200
.01UF
.01UF
>MOTOR+
>MOTOR-
IN4001 TYPICAL DIODES
NATIONAL LMD18200
TYPICAL H-BRIDGE
MOTOR DRIVERS
PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773
Title
Single Phase SIGN MAGNITUDE PWM
Size
B
Document Number
Rev
A
Date: Tuesday, November 19, 2002
Sheet 1 of 0

7.5 12-bit Parallel DAC Interface

The interface between the MC3110 chip and a quad 12 bit DAC is shown in the following figure. Any single channel A/D can also be used provided it meets the interface timing requirements.

Comments on Schematic

The 12 data bits are written to the DAC addressed by address bits A1 and A2, when A0 is 0. In this fashion CP address 4000 is used for axis 1. The odd address is reserved for chips with 2 drives per axis.

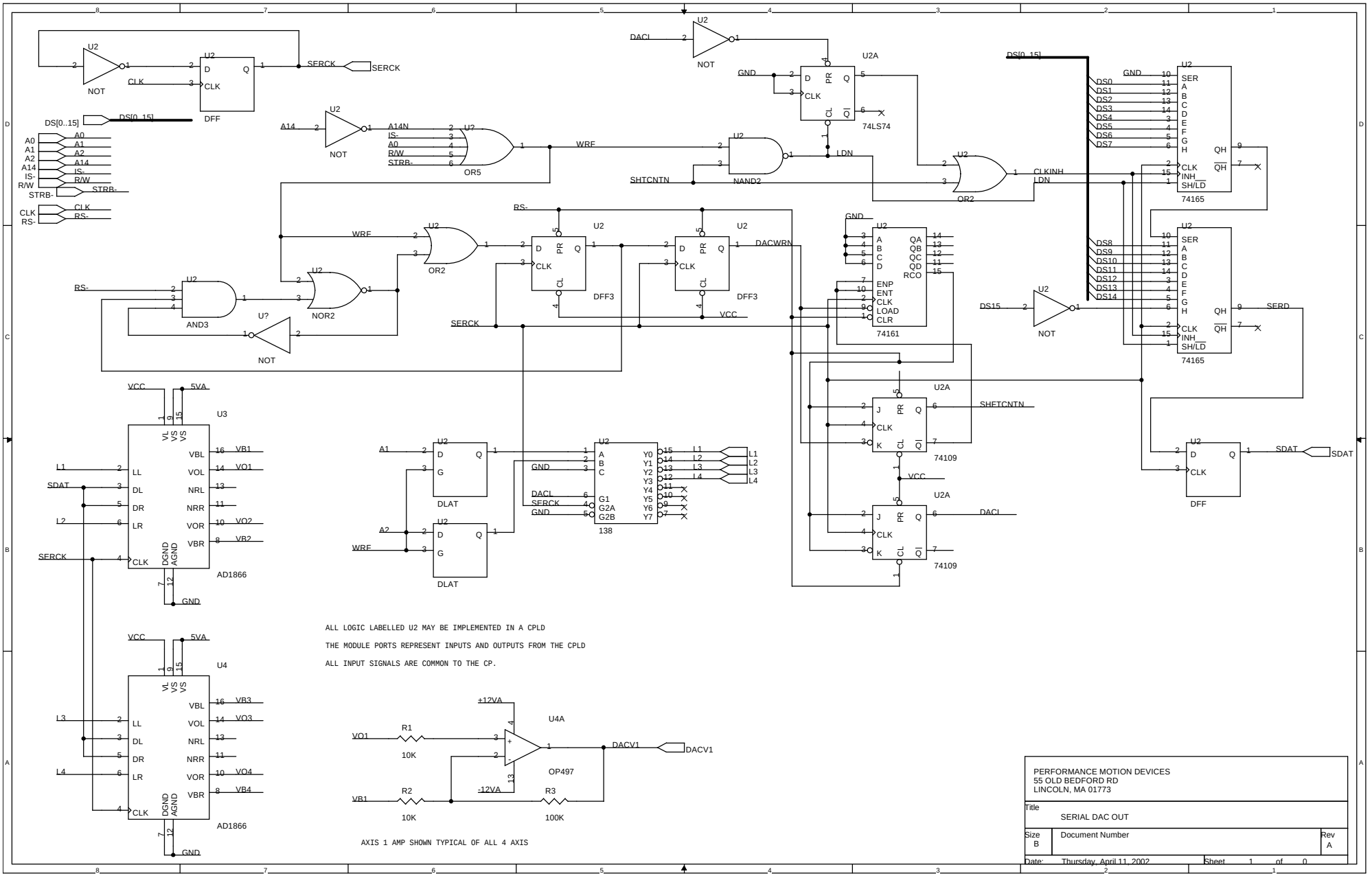

R2
22K
VCC
U1
DS[0..15]
A[0..15]
~RS
DS0
DS1
DS2
DS3
DS4
DS5
DS6
DS7
DS8
DS9
DS10
DS11
DS12
DS13
DS14
DS15
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA14
DATA15
ADDR0
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
N/C
~RAMSLCT
~PERIPHSLCT
R/~W
~STROBE
~WRITEENBL
W/~R
~HOSTINTRPT
PWMMAG1
PWMMAG2
PWMMAG3
PWMSIGN1
PWMSIGN2
SRLRCV
SRLXMT
SRLENABLE
I/OINTRPT
PRLENABLE
SYNCH
ANALOG1
ANALOG2
ANALOG3
ANALOG4
ANALOG5
ANALOG6
ANALOG7
ANALOG8
ANALOGVCC
ANALOGREFHIGH
ANALOGREFLOW
ANALOGGND
IS-
STRB-
WF-
POSLIM1
NEGLIM1
AXISIN1
AXISOUT1
QUADA1
QUADB1
~INDEX1
~HOME1
HALL1A
HALL1B
HALL1C
RS-
~RESET
CLK
CLOCKIN
GND
CP2N11
U7
VDD
VREFH
VLOG
VOUTA
VOUTB
VOUTC
VOUTD
DACVA1
DACVA2
DACVA3
DACVA4
R/W
/CS
/LDAC
/RESET
VREFL
BURR-BROWN 7724,7725
SO or PLCC
4 DACS @ CP ADR
0X4000+ 0,2,4,6.
IF CLEAN SUPPLIES +- 10V ARE
PROVIDED FOR VREFH AND VREFL
IT IS GENERALLY NOT NECESSARY
TO PROVIDE OFFSET ADJUST.
U2
OR4
OR2
NOT
A14-
THE LOGIC WITHIN THE DOTTED LINES
IS EASILY IMPLEMENTED WITHIN A CPLD.
OR5
PREFERRED LOGIC
PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773
Title
Single Phase DAC OUT
Size
B
Document Number
Rev
A
Date: Tuesday, November 19, 2002
Sheet 1 of 1

7.6 16-bit Serial DAC Interface

The following schematic shows an interface circuit between the MC3110 and a dual 16-bit serial DAC.

Comments on Schematic

The 16 data bits from the CP chip are latched in the two 74H165 shift registers when the CP writes to address 400x hex, and the address bits A1 and A2 are latched in the 2 DLAT latches and decoded by the 138 CPU cycle. The fed-back and-or gate latches, the decoded WRF, and the next clock will clear the 1st sequencer flop DFF3. This will disable the WRF latch and the second clock will clear the second DFF3 flop, forcing DACWRN low, and setting the first flop since WRF will have gone high. DACWRN low will clear the 74109, SHFTCNTN. The 4 bit counter, 74161, is also parallel loaded to 0, and the counter is enabled by ENP going high. The counter will not start counting nor the shift register start shifting until the clock after the DACWRN flop sets since the load overrides the count enable. When the DACWR flop is set the shift register will start shifting and the counter will count the shifts. After 15 shifts CNT15 from the counter will go high and the next clock will set the DACLAT flop and set the SHFTCNTN flop. This will stop the shift after 16 shifts and assert L1 through L4 depending on the address stored in the latch. The 16th clock also was counted causing the counter to roll over to 0 and CNT15 to go low. The next clock will therefore clear the DACLAT flop causing the DAC latch signal L1 through L4 to terminate and the 16 bits of data to be latched in the addressed DAC. The control logic is now back in its original state waiting for the next write to the DACs by the CP. SERCK is a 10MHz clock, the 20MHz CP clock divided by 2, since the AD1866 DACs will not run at 20MHz.

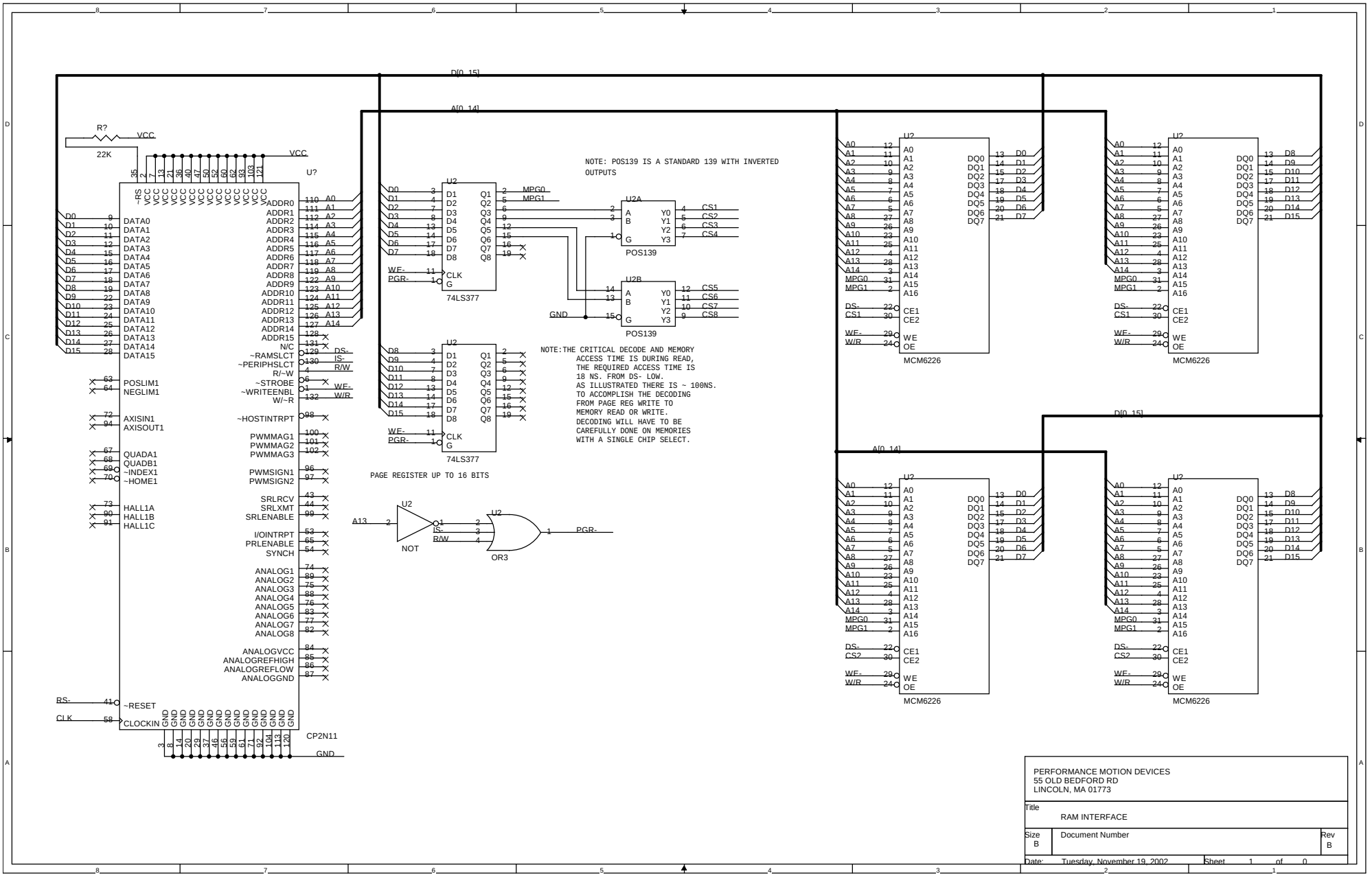

U2
NOT
SERCK
DFF
DS[0..15]
DS[0..15]
A0
A1
A2
A14
IS-
R/W
STRB-
CLK
RS-
A14
A14N
IS-
A0
R/W
STRB-
OR5
WRF
DACL
GND
U2A
74LS74
LDN
SHTCNTN
NAND2
OR2
CLKINH
LDN
74165
DS0
DS1
DS2
DS3
DS4
DS5
DS6
DS7
DS8
DS9
DS10
DS11
DS12
DS13
DS14
DS15
SERD
SDAT
RS-
AND3
NOR2
DFF3
DACWRN
74161
VCC
74109
SHFTCNTN
DACI
U3
U4
AD1866
VCC
5VA
GND
L1
L2
L3
L4
VB1
VB2
VB3
VB4
VO1
VO2
VO3
VO4
DLAT
138
A1
A2
ALL LOGIC LABELLED U2 MAY BE IMPLEMENTED IN A CPLD
THE MODULE PORTS REPRESENT INPUTS AND OUTPUTS FROM THE CPLD
ALL INPUT SIGNALS ARE COMMON TO THE CP.
+12VA
-12VA
U4A
OP497
R1
R2
R3
10K
100K
VO1
VB1
DACV1
AXIS 1 AMP SHOWN TYPICAL OF ALL 4 AXIS
PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773
Title
SERIAL DAC OUT
Size B
Document Number
Rev A
Date: Thursday, April 11, 2002
Sheet 1 of 0

7.7 RAM Interface

The following schematic shows an interface circuit between the MC3110 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.

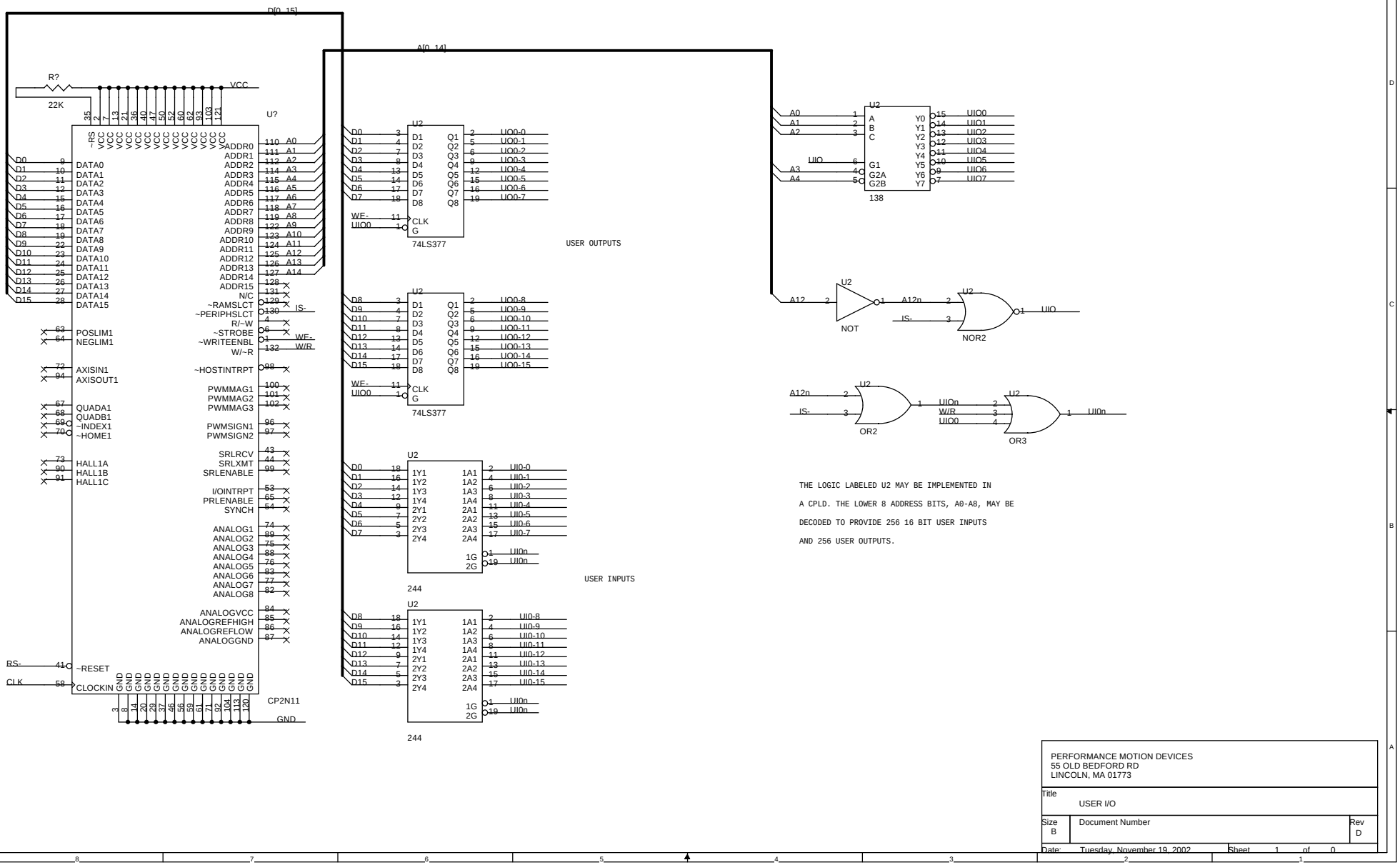

D[0..15]
A[0..14]
R?
22K
VCC
U?
~RS
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA14
DATA15
POSLIM1
NEGLIM1
AXISIN1
AXISOUT1
QUADA1
QUADB1
~INDEX1
~HOME1
HALL1A
HALL1B
HALL1C
ADDR0
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
N/C
~RAMSLCT
~PERIPHSLCT
R/~W
~STROBE
~WRITEENBL
W/~R
~HOSTINTRPT
PWMMAG1
PWMMAG2
PWMMAG3
PWMSIGN1
PWMSIGN2
SRLRCV
SRLXMT
SRLENABLE
I/OINTRPT
PRLENABLE
SYNCH
ANALOG1
ANALOG2
ANALOG3
ANALOG4
ANALOG5
ANALOG6
ANALOG7
ANALOG8
ANALOGVCC
ANALOGREFHIGH
ANALOGREFLOW
ANALOGGND
~RESET
CLOCKIN
CP2N11
GND
DS-
IS-
R/W
WE-
W/R
RS-
CLK
74LS377
MPG0
MPG1
CS1
CS2
CS3
CS4
CS5
CS6
CS7
CS8
POS139
U2A
U2B
NOTE: POS139 IS A STANDARD 139 WITH INVERTED OUTPUTS
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.
PAGE REGISTER UP TO 16 BITS
A13
NOT
OR3
PGR-
MCM6226
DQ0
DQ1
DQ2
DQ3
DQ4
DQ5
DQ6
DQ7
CE1
CE2
WE
OE
PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773
Title
RAM INTERFACE
Size
B
Document Number
Rev
B
Date: Tuesday, November 19, 2002
Sheet 1 of 0

7.8 User-defined I/O

The interface between the MC3110 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.



PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

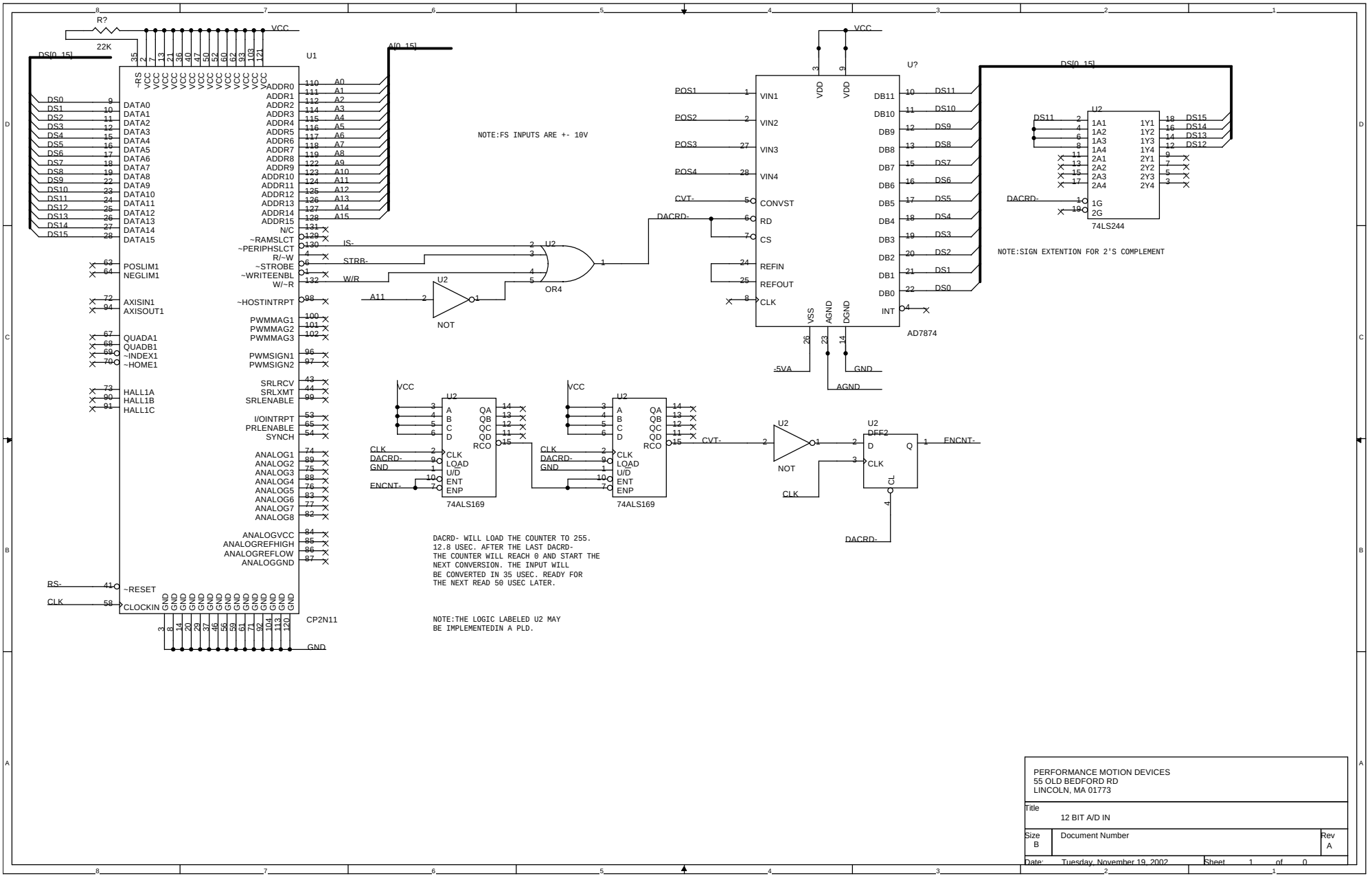
Title: USER I/O		
Size: B	Document Number	Rev: D
Date: Tuesday, November 19, 2002	Sheet 1 of 0	

7.9 12-bit A/D Interface

The following schematic shows a typical interface circuit between the MC3110 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.

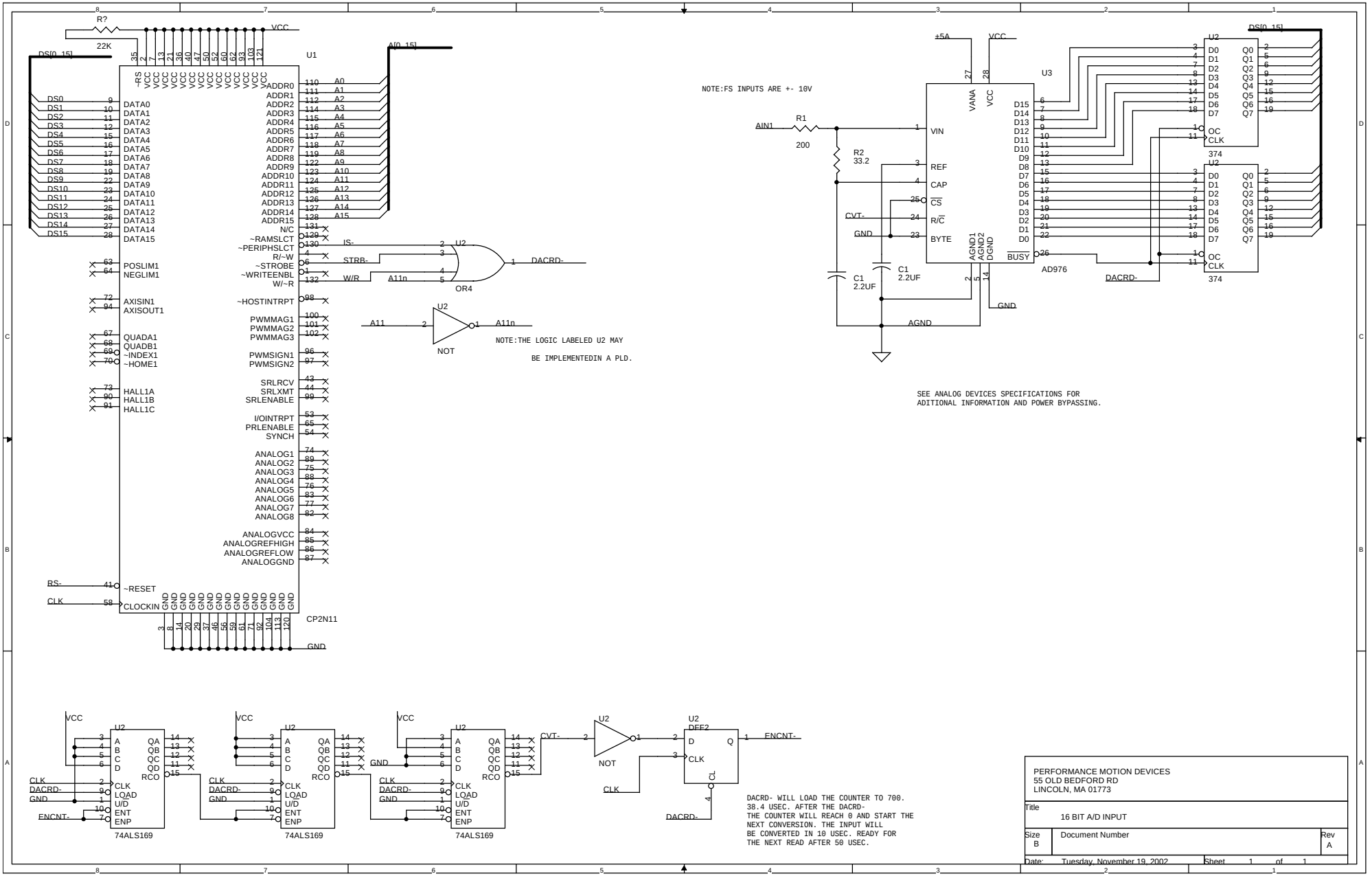

R?
22K
VCC
DS[0..15]
U1
CP2N11
A[0..15]
NOTE:FS INPUTS ARE +- 10V
U?
AD7874
NOTE:SIGN EXTENTION FOR 2'S COMPLEMENT
74LS244
OR4
NOT
74ALS169
DACRD- WILL LOAD THE COUNTER TO 255.
12.8 USEC. AFTER THE LAST DACRD-
THE COUNTER WILL REACH 0 AND START THE
NEXT CONVERSION. THE INPUT WILL
BE CONVERTED IN 35 USEC. 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: Tuesday, November 19, 2002
Sheet 1 of 0

7.10 16-bit A/D Input

The interface between the MC3110 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. 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.

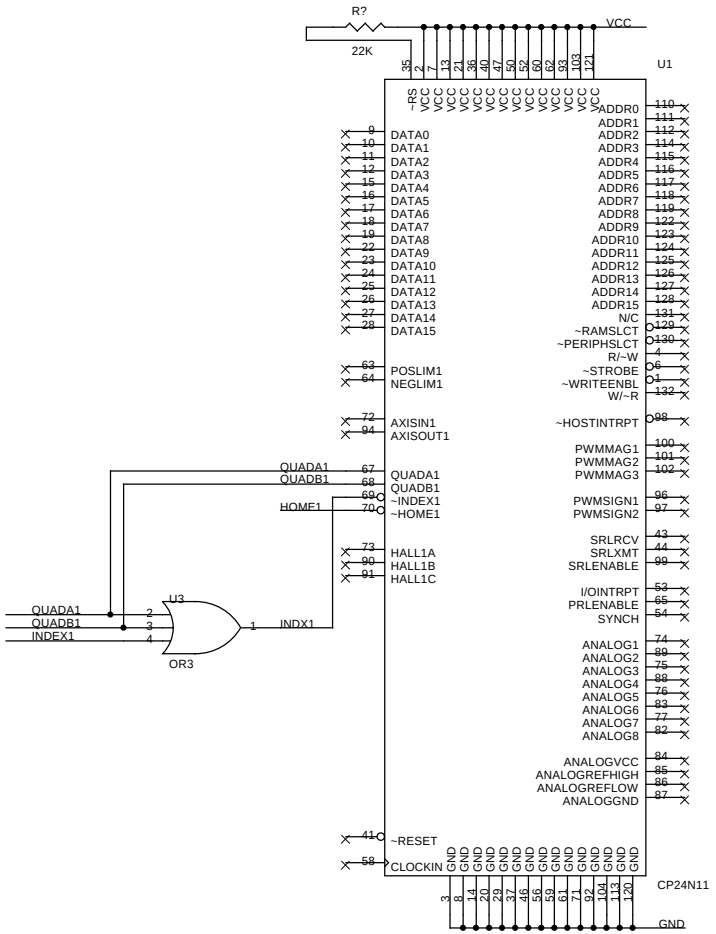

R?
22K
VCC
DS[0..15]
U1
A[0..15]
~RS
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA14
DATA15
POSLIM1
NEGLIM1
AXISIN1
AXISOUT1
QUADA1
QUADB1
~INDEX1
~HOME1
HALL1A
HALL1B
HALL1C
ADDR0
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
N/C
~RAMSLCT
~PERIPHSLCT
R/~W
~STROBE
~WRITEENBL
W/~R
~HOSTINTRPT
PWMMAG1
PWMMAG2
PWMMAG3
PWMSIGN1
PWMSIGN2
SRLRCV
SRLXMT
SRLENABLE
I/OINTRPT
PRLENABLE
SYNCH
ANALOG1
ANALOG2
ANALOG3
ANALOG4
ANALOG5
ANALOG6
ANALOG7
ANALOG8
ANALOGVCC
ANALOGREFHIGH
ANALOGREFLOW
ANALOGGND
~RESET
CLOCKIN
GND
CP2N11
IS-
STRB-
W/R
A11n
OR4
A11
NOT
NOTE:THE LOGIC LABELED U2 MAY
BE IMPLEMENTEDIN A PLD.
NOTE:FS INPUTS ARE +- 10V
+5A
AIN1
R1
200
R2
33.2
C1
2.2UF
VIN
REF
CAP
CS
R/C
BYTE
VANA
AGND1
AGND2
DGND
BUSY
AD976
U3
D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
OC
CLK
374
DACRD-
AGND
SEE ANALOG DEVICES SPECIFICATIONS FOR
ADITIONAL INFORMATION AND POWER BYPASSING.
RS-
CLK
RS:
CLK
DACRD-
GND
ENCNT-
A
B
C
D
QA
QB
QC
QD
RCO
LOAD
U/D
ENT
ENP
74ALS169
CVT-
DEF2
D
Q
CL
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.
PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773
Title
16 BIT A/D INPUT
Size
B
Document Number
Rev
A
Date:
Tuesday, November 19, 2002
Sheet
1
of
1

7.11 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, 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.



PERFORMANCE MOTION DEVICES
55 OLD BEDFORD RD
LINCOLN, MA 01773

Title: EXTERNAL GATING LOGIC INDEX

Size: B | Document Number | Rev: A

Date: Tuesday, November 19, 2002 | Sheet 1 of 1