National Semiconductor

PRELIMINARY
June 1994

HPC46100 High-Performance microController with DSP Capability

General Description

The HPC46100 is a member of the HPC™ family of High Performance microControllers. Each member of the family has a similar core CPU with unique memory, resources, and I/O configuration to suit specific applications. The HPC46100 is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, high speed computation, and low power consumption.

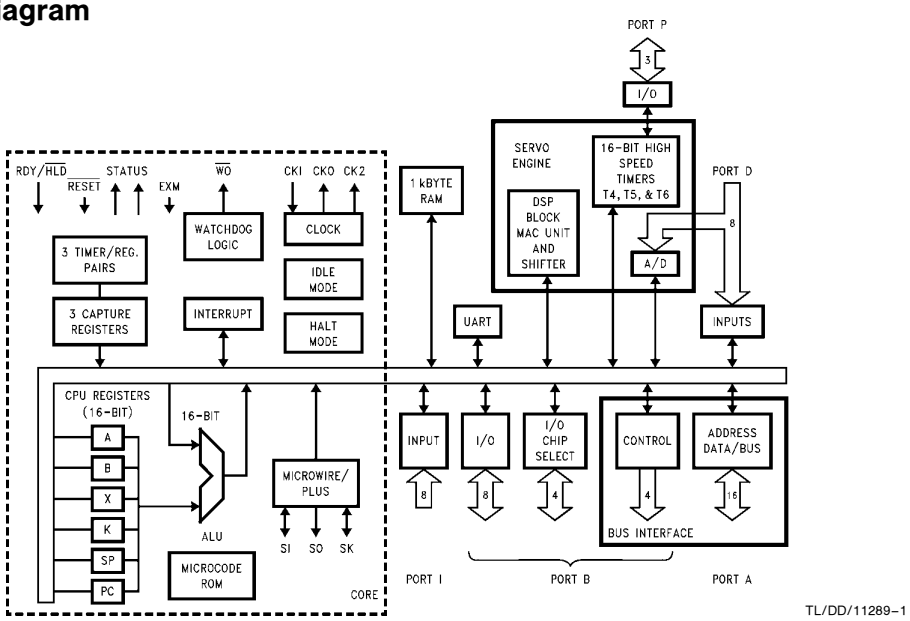
Throughput is enhanced by operating the HPC46100 at frequencies up to 40 MHz, by integrating a Multiply/Accumulate Unit (MAU) onto the chip, and by optimizing instructions to increase efficiency. These features increase performance in closed loop digital servo and filter applications.

The HPC devices are complete microcomputers on a single chip. All system timing, internal logic, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as an MAU unit, PWM outputs, Chip Select Signals, UART, up to seven 16-bit timers with input capture capability, WATCHDOG™ logic, vectored interrupts, and MICROWIRE/PLUS™ provide a high level of system integration. The ability to directly address up to 64 kbytes of memory enables the HPC to be used in powerful applications typically performed by microprocessors and peripheral chips. (Continued)

Features

- Multiply/Accumulate Unit for fast signed multiply or multiply-accumulate
- High speed 16 bit timers with PWM outputs or input capture logic
- 4 Chip select output logic with programmable control
- 8-channel, 8-bit A/D Converter
- 1024 bytes of on-chip 0 wait state RAM
- FAST 100 ns for fastest instruction when using 40.0 MHz clock
- Very low power with two power save modes: IDLE and HALT
- UART full duplex, with a programmable baud rate generator and parity checking/detection
- MICROWIRE/PLUS serial I/O interface
- 8 vectored interrupt sources
- Signed overflow flag for add and subtract instructions
- 16 x 16 multiply and 32 x 16 divide
- 16-bit architecture with byte and word operations
- 64 kbytes of direct memory addressing
- 8- or 16-bit wide external memory
- Program instructions can be executed from RAM
- Up to 31 general purpose I/O lines that are memory mapped
- WATCHDOG logic

Block Diagram



TL/DD/11289–1

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
HPC™, WATCHDOG™, MICROWIRE/PLUS™ and MICROWIRE™ are trademarks of National Semiconductor Corporation.

RRD-B30M105/Printed in U. S. A.

General Description (Continued)

The HPC46100 has, as an on-chip peripheral, an 8-channel 8-bit Analog-to-Digital Converter. This A/D converter can operate in a single-ended mode where the analog input voltage is applied across one of the eight input channels (D0–D7) and AGND. The A/D converter can also operate in a differential mode where the analog input voltage is applied across two adjacent input channels. The A/D converter will convert up to eight channels in the single-ended mode and up to four channel pairs in the differential mode.

A group of three high speed timers provide processor independent PWM signal generation. These timers and their support logic provide independent control of PWM frequency and PWM duty cycle with a minimum resolution of 50 ns when running at 40 MHz.

The HPC46100 is upwards source code compatible with the HPC family except for Decimal Add and Subtract.

The HPC46100 is available in an 80-pin QFP package.

40 MHz

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Total Allowable Source or Sink Current	100 mA
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
V_{CC} with Respect to GND	−0.5V to +6.5V
V_{REF} with Respect to GND	V_{CC}
All Other Pins	(V_{CC} +0.5V) to (GND − 0.5V)

DC Characteristics

V_{CC} = 5.0V ±10% unless otherwise specified. T_A = 0°C to +70°C unless otherwise specified.

Symbol	Parameter	Test Conditions	Min	Max	Units
SUPPLIES					
I_{CC1}	Supply Current	V_{CC} = 5.5V, f_{IN} = 40 MHz (Note 1) V_{CC} = 5.5V, f_{IN} = 20 MHz (Note 1)		65 40	mA mA
I_{CC2}	IDLE Mode Supply Current	V_{CC} = 5.5V, f_{IN} = 40 MHz (Note 1) V_{CC} = 5.5V, f_{IN} = 20 MHz (Note 1)		25 20	mA mA
I_{CC3}	HALT Mode Supply Current with $\overline{NMI}$ High	V_{CC} = 5.5V, f_{IN} = 0 (Note 1) V_{CC} = 2.5V, f_{IN} = 0 (Note 1)		300 100	µA µA
I_{CC4}	HALT Mode Supply Current with $\overline{NMI}$ Low	V_{CC} = 5.5V, f_{IN} = 40 MHz (Note 1)		25	mA
V_{RAM}	RAM Keep-Alive Voltage	(Note 2)	2.5		V
INPUT VOLTAGE LEVELS FOR SCHMITT TRIGGERED INPUTS $\overline{RESET}$, NMI AND $\overline{WO}$; AND ALSO CKI					
V_{IH1}	Logic High		0.9 V_{CC}		V
V_{IL1}	Logic Low			0.1 V_{CC}	V
INPUT VOLTAGE LEVELS FOR PORT A					
V_{IH2}	Logic High		2.0		V
V_{IL2}	Logic Low			0.8	V
INPUT VOLTAGE LEVELS FOR ALL OTHER INPUTS					
V_{IH3}	Logic High (except Port D)		0.7 V_{CC}		V
V_{IL3}	Logic Low (except Port D)			0.2 V_{CC}	V
V_{IH4}	Logic High (Port D only)	(Note 6 in AC Characteristics)	0.7 V_{CC}		V
V_{IL4}	Logic Low (Port D only)	(Note 6 in AC Characteristics)		0.2 V_{CC}	V
I_{LI1}	Input Leakage Current; all pins except below			10	µA
I_{LI2}	Input Leakage Current; pin RDY/$\overline{HLD}$ only		−3	−50	µA
I_{LI3}	Input Leakage Current; pin B12 ($\overline{HBE}$) only	RESET = GND, V_{IN} = V_{CC}	0.5	7.0	mA
C_I	Input Pin Capacitance	(Note 3)		10	pF
C_{IO}	Input/Output Pin Capacitance	(Note 3)		20	pF
OUTPUT VOLTAGE LEVELS					
CMOS USAGE: ALL OUTPUTS AND I/O PINS					
V_{OH1}	Logic High	I_{OH} = −10 µA (Note 3)	V_{CC} − 0.1		V
V_{OL1}	Logic Low	I_{OL} = 10 µA (Note 3)		0.1	V

40 MHz (Continued)

DC Characteristics

V_{CC} = 5.0V ±10% unless otherwise specified. T_A = 0°C to +70°C unless otherwise specified. (Continued)

Symbol	Parameter	Test Conditions	Min	Max	Units
BUS DRIVERS: PORT A AND PINS B8–B15, PALE, CK2, ST1 AND ST2					
V_{OH2}	Logic High	I_{OH} = −1 mA	2.4		V
V_{OL2}	Logic Low	I_{OL} = 3 mA		0.4	V
OTHER I/O PORT DRIVERS: B0–B7, $\overline{WO}$, P0–P2					
V_{OH3}	Logic High	I_{OH} = −1.6 mA	2.4		V
V_{OL3}	Logic Low	I_{OL} = 0.5 mA		0.4	V
I_{OZ}	TRI-STATE® Leakage Current			10	μA

Note 1: I_{CC1}, I_{CC2}, I_{CC3} and I_{CC4} are measured with no external drive (I_{OH} and I_{OL} = 0; I_{IH} and I_{IL} = 0). I_{CC1} is measured with $\overline{RESET}$ = GND, I_{CC3} is measured with NMI = V_{CC}, I_{CC4} is measured with NMI = GND, CKI driven to V_{IH1} and V_{IL1} with rise and fall times less than 10 ns.

Note 2: Test duration is 100 ms.

Note 3: This is guaranteed by design and not tested.

AC Electrical Characteristics

See Notes 1 and 4 and *Figure 1* thru *Figure 5*. V_{CC} = 5.0V ±10% unless otherwise specified, T_A = 25°C, one wait state.

Symbol	Parameter	Min	Max	Units
CLOCKS				
f_C	CKI Operating Frequency	2	40	MHz
$t_{C1} = 1/f_C$	CKI Period	25	500	ns
t_{C1H}	CKI High Time	11.25		ns
t_{C1L}	CKI Low Time	11.25		ns
$t_C = 2/f_C$	Bus Timing Cycle	50	1000	ns
t_{WAIT}	Wait State Period	50		ns
t_{DC1C2R}	CK2 Rising Edge after CKI Falling Edge (Note 2)	0	55	ns
t_{DC1C2F}	CK2 Falling Edge after CKI Falling Edge (Note 2)	0	55	ns
f_U	External UART Clock Input Frequency		5	MHz
f_{MW}	External MICROWIRE/PLUS Clock Input Frequency		2.5	MHz
TIMER T0–T3				
$f_{XIN} = f_C/22$	External Timer Input Frequency		1.82	MHz
$t_{XIN} = t_C$	Pulse Width for Timer Input	50		ns
TIMER T4–T6				
$f_{HSXIN} = f_C/5$	External Timer Input Frequency		8	MHz
$t_{HSXIN} = 1.5\ t_C$	Pulse Width for Timer Input	75		ns
MICROWIRE/PLUS				
t_{UWS}	MICROWIRE Setup Time			
Master		100		ns
Slave		20		
t_{UWH}	MICROWIRE Hold Time			
Master		20		ns
Slave		50		
t_{UWS}	MICROWIRE Output Valid Time			
Master			50	ns
Slave			150	

40 MHz (Continued)

AC Electrical Characteristics (Continued)

See Notes 1 and 4 and *Figure 1* thru *Figure 5*). V_{CC} = 5.0V ±10% unless otherwise specified, T_A = 25°C, one wait state.

Symbol	Parameter	Min	Max	Units
EXTERNAL HOLD				
$t_{SALE} = 0.75\,t_C + 40$	$\overline{HLD}$ Falling Edge before ALE Rising Edge	77.5		ns
$t_{HWP} = 0.75\,t_C + 35$	$\overline{HLD}$ Pulse Width	72.5		ns
$t_{HAE} = 0.75\,t_C + 100$	$\overline{HLDA}$ Falling Edge after $\overline{HLD}$ Falling Edge (Note 3)		137.5	ns
$t_{HAD} = 0.75\,t_C + 85$	$\overline{HLDA}$ Rising Edge after $\overline{HLD}$ Rising Edge		122.5	ns
t_{BF}	Bus TRI-Stated after $\overline{HLDA}$ Falling Edge (Note 5)		70	ns
$t_{BE} = 0.75\,t_C + 30$	Bus Enable after $\overline{HLDA}$ Rising Edge	67.5		ns
NATIVE BUS TIMING: ADDRESS CYCLE				
$t_{LL} = 0.5\,t_C - 10$	ALE Pulse Width	15		ns
t_{1ALR}	ALE Rising Edge after CK1 Rising Edge (Note 2)	0	35	ns
t_{1ALF}	ALE Falling Edge after CK1 Falling Edge (Note 2)	0	35	ns
$t_{2ALR} = 0.25\,t_C + 20$	ALE Rising Edge after CK2 Rising Edge		32.5	ns
$t_{2ALF} = 0.25\,t_C + 20$	ALE Falling Edge after CK2 Falling Edge		32.5	ns
$t_{ST} = 0.25\,t_C - 9$	Address Valid to ALE Falling Edge	3.5		ns
$t_{VP} = 0.5\,t_C - 10$	Address Hold after ALE Falling Edge	15		ns
READ CYCLE				
$t_{RW} = 0.25\,t_C + WS - 15$	$\overline{RD}$ Pulse Width	47.5		ns
$t_{ARD} = 0.75\,t_C - 20$	Address Valid to $\overline{RD}$ Falling Edge	17.5		ns
$t_{ARR} = 0.5\,t_C - 20$	ALE Falling Edge to $\overline{RD}$ Falling Edge	5		ns
$t_{RD} = 0.25\,t_C + WS - 20$	$\overline{RD}$ Falling Edge to Input Data Valid		42.5	ns
t_{DR}	Data Hold after $\overline{RD}$ Rising Edge	0	50	ns
$t_{ACC} = t_C + WS - 20$	Address Valid to Input Data Valid		80	ns
WRITE CYCLE				
$t_{WW} = 0.75\,t_C + WS - 15$	$\overline{WR}$ Pulse Width	72.5		ns
$t_V = 0.5\,t_C + WS - 20$	Data Valid before $\overline{WR}$ Rising Edge	55		ns
$t_{HW} = 0.5\,t_C - 10$	Data Hold after $\overline{WR}$ Rising Edge	15		ns
$t_{AWR} = 0.75\,t_C - 20$	Address Valid to $\overline{WR}$ Falling Edge	17.5		ns
READY INPUT				
t_{RDYS}	$\overline{RDY}$ Falling Edge before CK2 Falling Edge	45		ns
t_{RDYH}	$\overline{RDY}$ Rising Edge after CK2 Falling Edge	0		ns
$t_{RDYV} = WS + 0.25\,t_C - 30$	$\overline{RDY}$ Falling Edge after $\overline{RD}$ or $\overline{WR}$ Falling Edge		32.5	ns
CHIP SELECT NATIVE BUS TIMING				
$t_{CS30RD} = 0.75\,t_C - 30$	CS3, CS0 Valid to $\overline{RD}$ Falling Edge	7.5		ns
$t_{ACCS30} = t_C + WS - 30$	CS3, CS0 Valid to Input Data Valid		70	ns
$t_{CS21RD} = 0.75\,t_C - 35$	CS2, CS1 Valid to $\overline{RD}$ Falling Edge	2.5		ns
$t_{ACCS21} = t_C + WS - 35$	CS2, CS1 Valid to Input Data Valid		65	ns
$t_{CSHR} = t_C - 15$	Chip Select Hold after $\overline{RD}$ Rising Edge	35		ns
$t_{CS30WR} = 0.75\,t_C - 30$	CS3, CS0 Valid to $\overline{WR}$ Falling Edge	7.5		ns
$t_{CS21WR} = 0.75\,t_C - 35$	CS2, CS1 Valid to $\overline{WR}$ Falling Edge	2.5		ns
$t_{CSHW} = 0.5\,t_C - 15$	Chip Select Hold after $\overline{WR}$ Rising Edge	10		ns

40 MHz (Continued)

AC Electrical Characteristics (Continued)

See Notes 1 and 4 and *Figure 1* thru *Figure 5*. $V_{CC} = 5.0V \pm 10\%$ unless otherwise specified, $T_A = 25°C$, one wait state.

Symbol	Parameter	Min	Max	Units
E SIGNAL TIMING PARAMETERS				
$t_{RWSE} = 0.25\,t_C - 7$	$\overline{WR}$ Falling Edge to E Rising Edge	5.5		ns
$t_{RWHE} = 0.5\,t_C - 7$	E Falling Edge to $\overline{WR}$ Rising Edge	18		ns
$t_{ASE} = t_C - 20$	Address Valid to E Rising Edge	30		ns
$t_{RDE} = WS - 20$	E Falling Edge to Data Input Valid	30		ns
SLOW PERIPHERAL TIMING PARAMETERS				
$t_{PLL} = t_C - 5$	PALE Pulse Width	45		ns
$t_{PST} = 0.75\,t_C - 10$	Address Valid to PALE Falling Edge	27.5		ns
$t_{PVL} = 0.75\,t_C - 15$	Address Hold from PALE Falling Edge	22.5		ns
$t_{PVP} = 0.75\,t_C - 10$	PALE Falling Edge to $\overline{RD}$ or $\overline{WR}$ Falling Edge	27.5		ns
$t_{PCSA} = 0.25\,t_C - 12.5$	Chip Select Setup to PALE Falling Edge	0		ns
$t_{PAS} = 1.5\,t_C - 20$	Address Setup to $\overline{RD}$ or $\overline{WR}$ Falling Edge	55		ns
$t_{PCSS} = t_C - 15$	Chip Select Setup to $\overline{RD}$ or $\overline{WR}$ Falling Edge	35		ns
$t_{PCSH} = 0.5\,t_C - 15$	Chip Select Hold from $\overline{RD}$ or $\overline{WR}$ Rising Edge	10		ns
$t_{PACC} = 5\,t_C - 25$	Address Valid to Input Data Valid		225	ns
$t_{PRD} = 3.5\,t_C - 25$	$\overline{RD}$ Falling Edge to Data In Valid		150	ns
$t_{PDR} = t_C$ (max)	Data Hold after $\overline{RD}$ Rising Edge	0	50	ns
$t_{PRW} = 3.5\,t_C - 15$	$\overline{RD}$ Strobe Width	160		ns
$t_{PSW} = 3.0\,t_C - 20$	Data Setup before $\overline{WR}$ Rising Edge	130		ns
$t_{PHW} = t_C - 20$	Data Hold after $\overline{WR}$ Rising Edge	30		ns
$t_{PWW} = 3.5\,t_C - 15$	$\overline{WR}$ Strobe Width	160		ns

Note: $C_L = 40$ pF.

Note 1: These AC characteristics are guaranteed with external clock drive on CKI having 50% duty cycle and with less than 15 pF load on CKO with rise and fall times (t_{CKIR} and t_{CKIL}) on CKI input less than 2.5 ns.

Note 2: Do not design with these parameters unless CKI is driven with an active signal. When using a passive crystal circuit, its stability is not guaranteed if either CKI or CKO is connected to any external logic other than the passive components of the crystal circuit.

Note 3: t_{HAE} is spec'd for case with $\overline{HLD}$ falling edge occurring at the latest time it can be accepted during the present CPU cycle begin executed. If $\overline{HLD}$ falling edge occurs later, t_{HAE} as long as ($3\,t_C + 4\,WS + 72\,t_C + 100$) may occur depending on the following CPU instruction cycles, its wait states and ready input.

Note 4: WS (t_{WAIT}) x (number of preprogrammed wait states). Minimum and maximum values are calculated at maximum operating frequency, $t_C = 40$ MHz, with one wait state programmed.

Note 5: Due to testing limitations—actual limits will be better.

A/D Converter Specifications

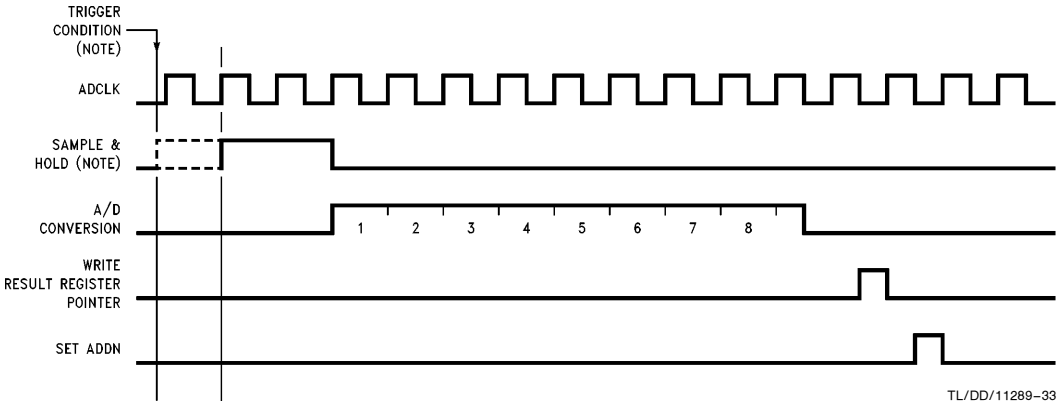
V_{CC} = 5V ±10% (V_{SS} − 0.050V) ≤ Any Input ≤ (V_{CC} + 0.050V), f_{IN} = 24 MHz

Parameter	Conditions	Min	Typ	Max	Units
Resolution				8	Bits
Reference Voltage Input	AGND = 0V	3		V_{CC}	V
Absolute Accuracy	$V_{REF} = V_{CC}$			±1	LSB
Non-Linearity	$V_{REF} = V_{CC}$ Deviation from the Best Straight Line			±½	LSB
Differential Non-Linearity	$V_{REF} = V_{CC}$			±½	LSB
Input Reference Resistance		1.6		4.8	kΩ
Common Mode Input Range (Note 6)		AGND		V_{REF}	V
DC Common Mode Error				±¼	LSB
Off Channel Leakage Current			1		μA
On Channel Leakage Current			1		μA
A/D Clock Frequency (Note 7)		0.1		2.0	MHz
Conversion Time (Note 8)			13.0		A/D Clock Cycle

Note 6: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input. The diodes will forward conduct for analog input voltages below ground or above the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 7: See Prescaler description.

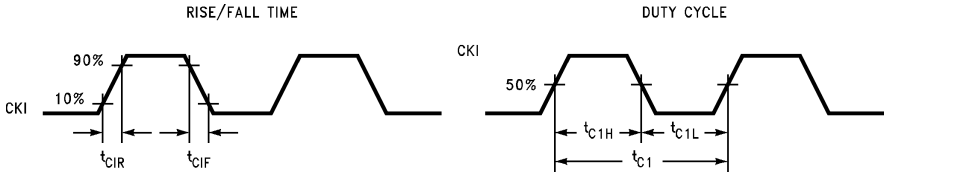
Note 8: Conversion Time includes sample and hold time. See following diagrams.



TL/DD/11289–33

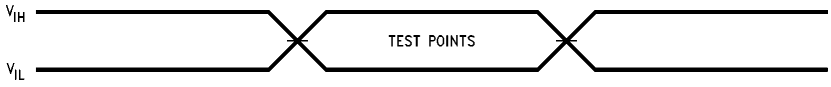
Note: The trigger condition generated by the start conversion method selected by the SC bits requires one CK2 to propagate through before the trigger condition is known. Once the trigger condition is known, the sample and hold will start at the next rising edge of ADCLK. The diagram shows worst case.

Timing Waveforms



TL/DD/11289–2

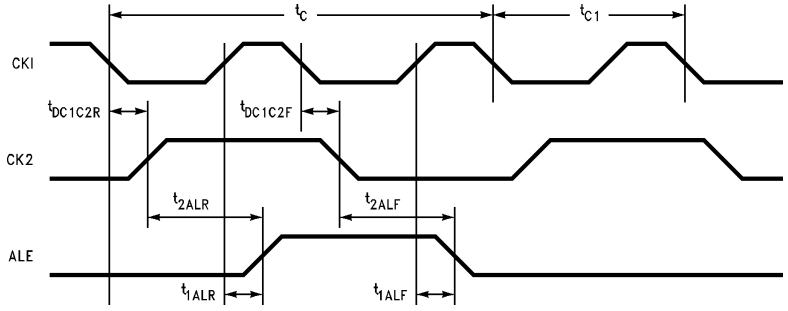
FIGURE 1. CKI Input Signal



TL/DD/11289–3

Note: AC testing inputs are driven at V_{IH} for a logic "1" and V_{IL} for a logic "0". Output timing measurements are made at $V_{CC}/2$ for both logic "1" and logic "0".

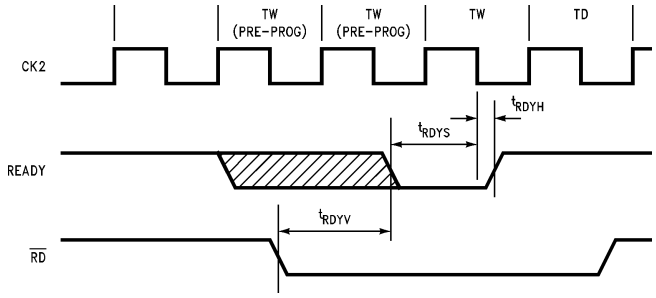
FIGURE 2. Input and Output for AC Tests



TL/DD/11289–4

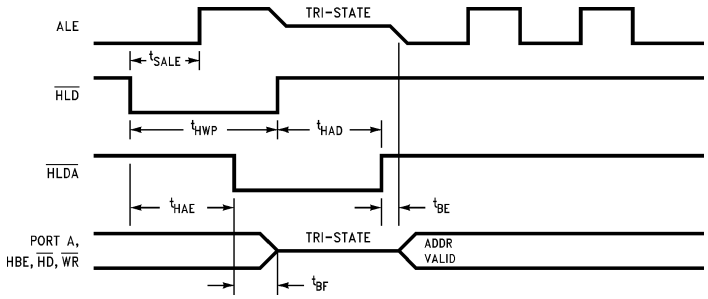
FIGURE 3. CKI, CK2 ALE Timing Diagram

Timing Waveforms (Continued)



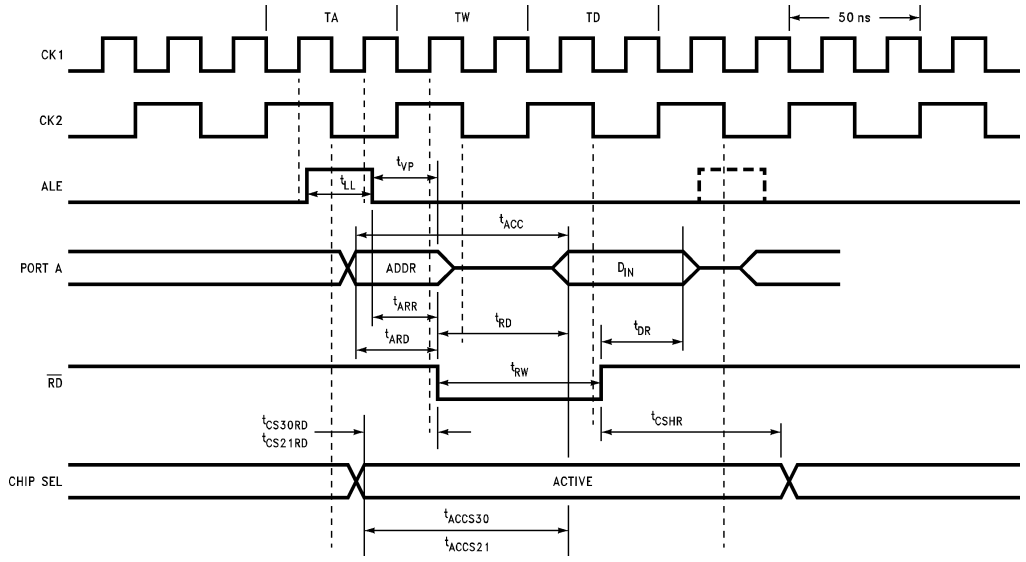
TL/DD/11289–5

FIGURE 4. Ready Mode Timing



TL/DD/11289–6

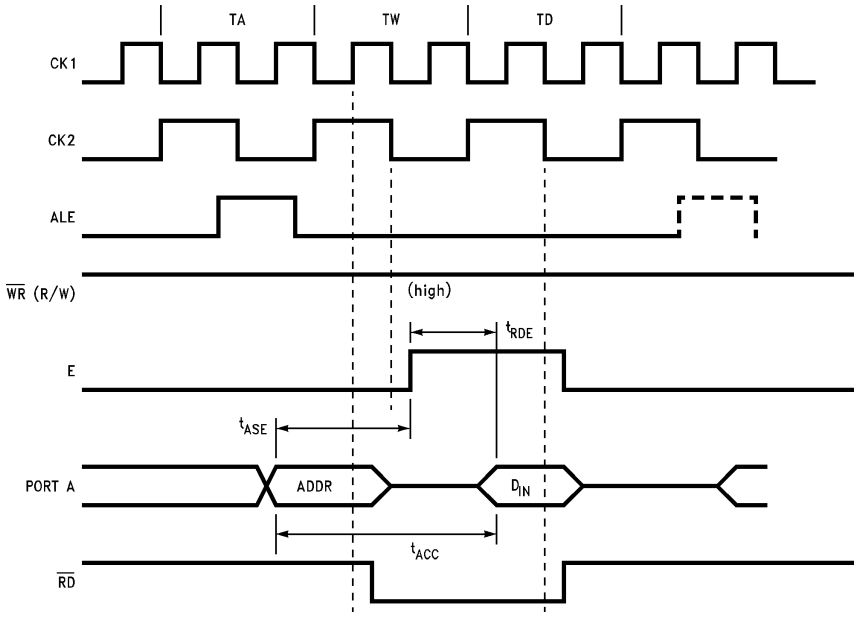
FIGURE 5. External Hold Timing



TL/DD/11289–7

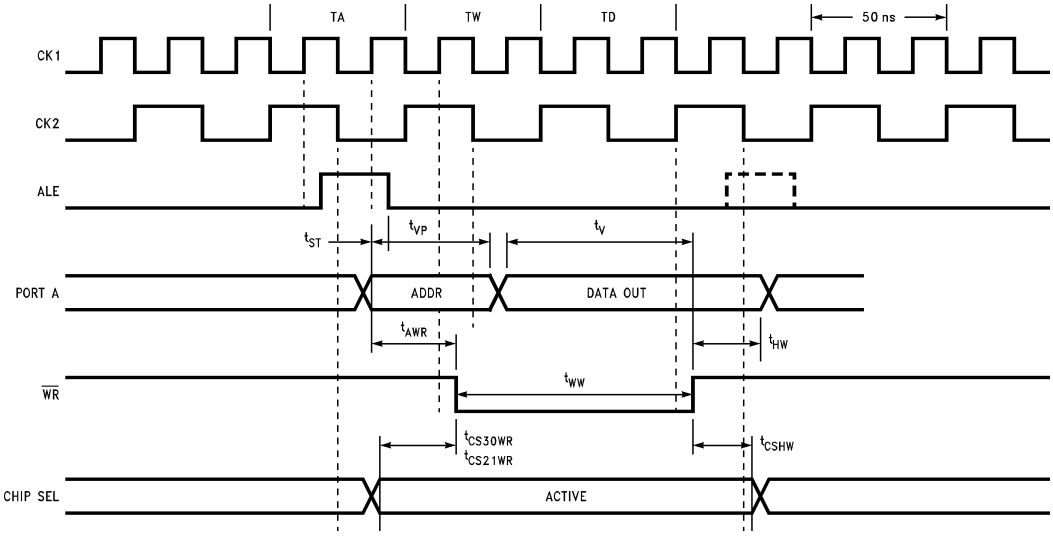
FIGURE 6. Native Bus Mode Read Cycle (1 Wait State)

Timing Waveforms (Continued)



TL/DD/11289–8

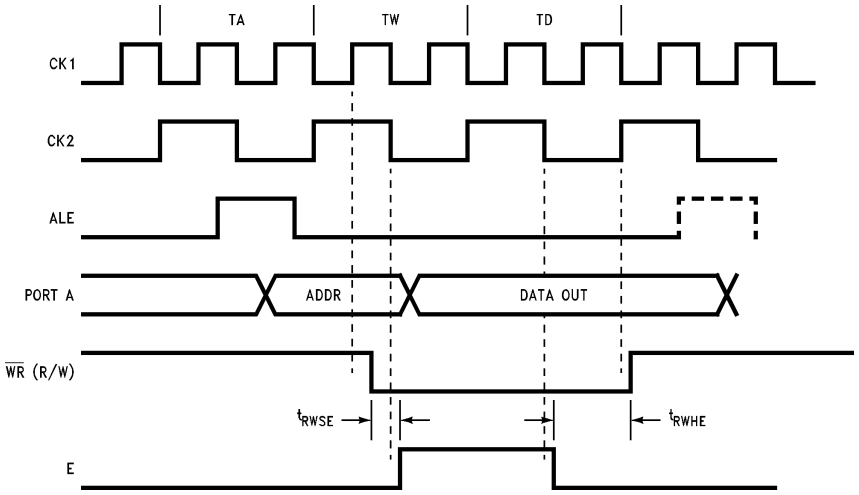
FIGURE 7. E Signal Read Cycle (1 Wait State)



TL/DD/11289–9

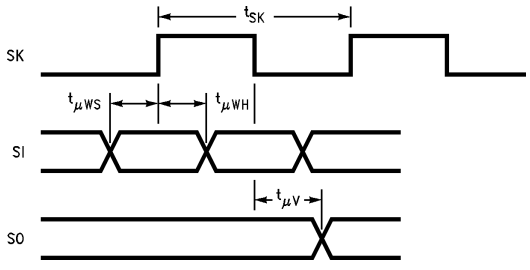
FIGURE 8. Native Bus Mode Write Cycle (1 Wait State)

Timing Waveforms (Continued)



TL/DD/11289–10

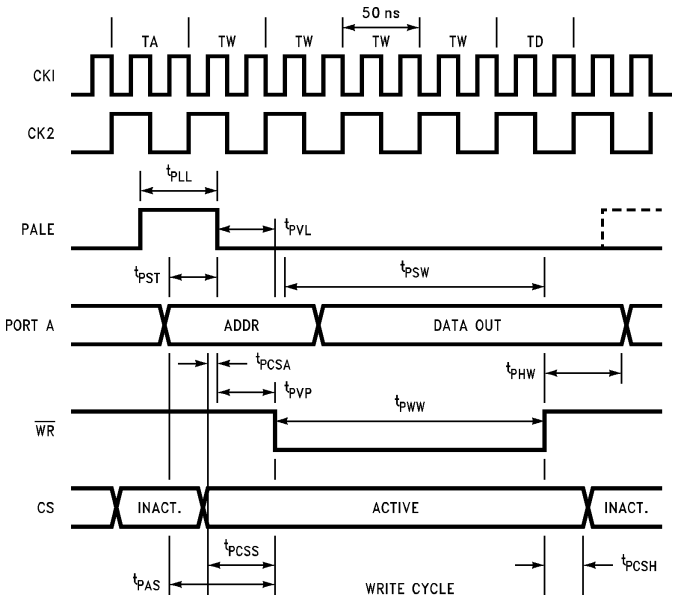
FIGURE 9. E Signal Write Cycle (1 Wait State)



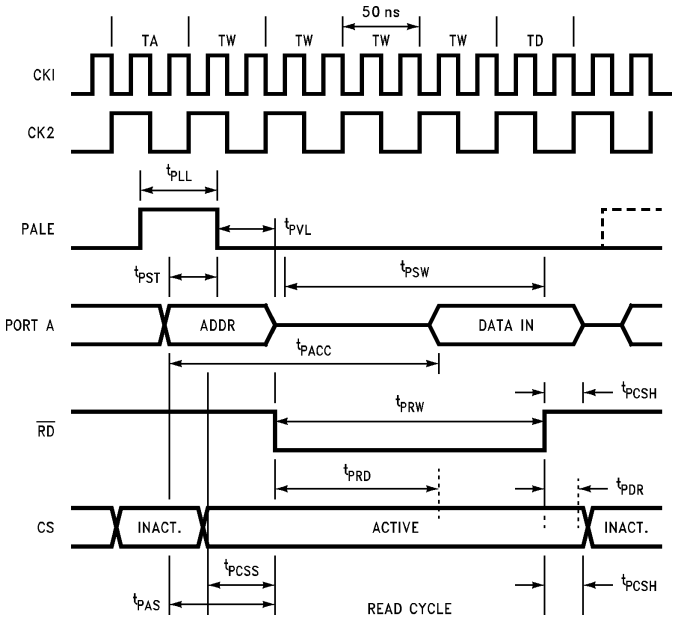
TL/DD/11289–11

FIGURE 10. MICROWIRE™ Setup/Hold Timing

Timing Waveforms (Continued)



TL/DD/11289–12



TL/DD/11289–13

FIGURE 11. Slow Peripheral Bus Timing

I/O Ports

PORT A

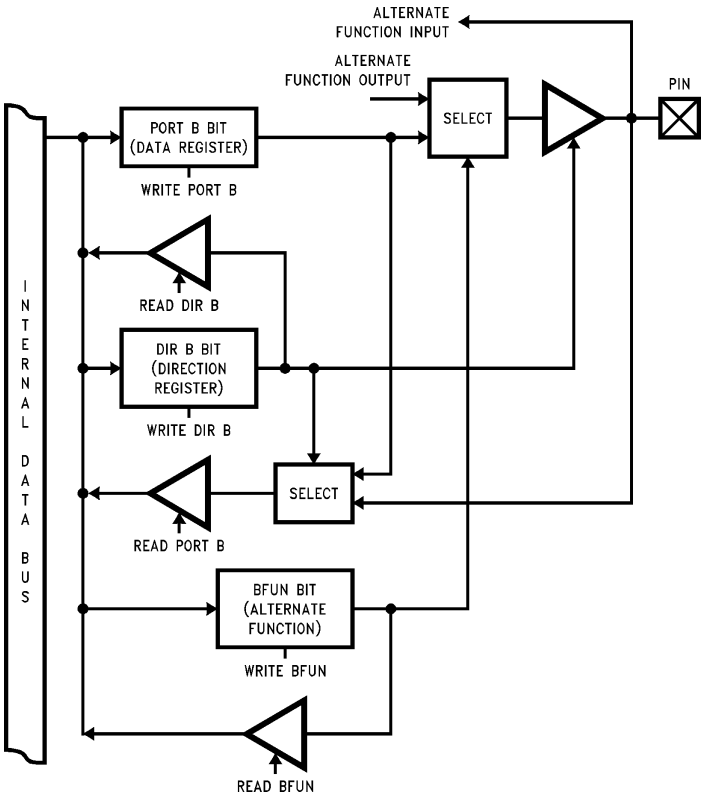
Port A is a 16-bit multiplexed address/data bus used for accessing external program and data memory. Four associated bus control signals are available on port B. The Address Latch Enable (ALE) signal is used to provide timing to demultiplex the bus. Reading from and writing to external memory are signalled by $\overline{RD}$ and $\overline{WR}$ respectively. External memory can be addressed as either bytes or words with the decoding controlled by two lines. Bus High Bus Enable ($\overline{HBE}$) and Address/Data line 0 (A0).

PORT B

Port B is a 16-bit port with 12 bits of bidirectional I/O. B10, B11, B12 and B15 are the control bus signals for the address/data bus. Port B may also be configured via a function register BFUN to individually allow each bidirectional I/O pin to have an alternate function. The direction of port B is determined by the direction register (DIRB). This register is used to set up each pin to be individually defined as an input or output. A specific I/O bit is selected as a high impedance input by clearing the corresponding bit in the direction register. It is selected as an output by setting this bit. The data register (PORTB) is used to hold data to be output on the B port. A write operation to a port pin configured as an input causes the value to be written into the data register, a read operation returns the value of the pin. Writing to port pins configured as outputs causes the pins to have the same value, reading the pins returns the value of the data register. *Figure 12* through *Figure 14* show the structure of Port B.

Port B may also be configured via a 16-bit alternate function register BFUN, to individually allow each pin to have an alternate function. The alternate functions are enabled by setting the corresponding bits in the BFUN register. The alternate B port functions are as follows:

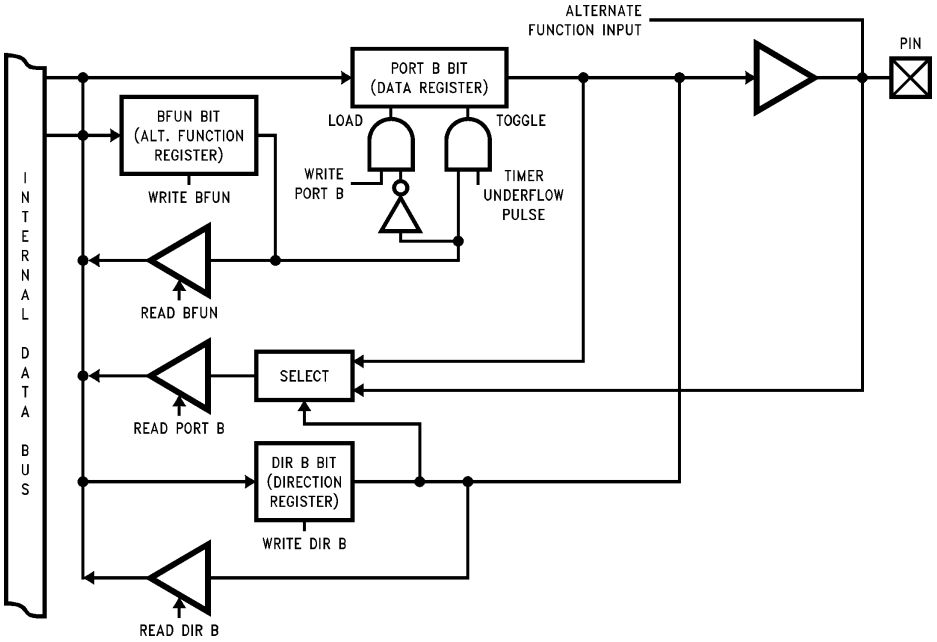
Pin	Alternate	Function
B0	TDX	UART Data Output
B1	E	E signal output
B2	CKX	UART Clock
B3	T2IO	Timer2 I/O Pin
B4	T3IO	Timer3 I/O Pin
B5	SO	MICROWIRE/PLUS Output (data)
B6	SK	MICROWIRE/PLUS Clock
B7	$\overline{HLDA}$	Hold Acknowledge Output
B8	TS0/CS0	Timer Synchronous or Chip Select Output
B9	TS1/CS1	Timer Synchronous or Chip Select Output
B10	ALE	Address/data bus Address Latch Enable
B11	$\overline{WR}$	Address/data bus Write Output
B12	$\overline{HBE}$	Address/data bus High Byte Enable
B13	TS2/CS2	Timer Synchronous or Chip Select Output
B14	TS3/CS3	Timer Synchronous or Chip Select Output
B15	$\overline{RD}$	Address/data bus Read Output



TL/DD/11289–14

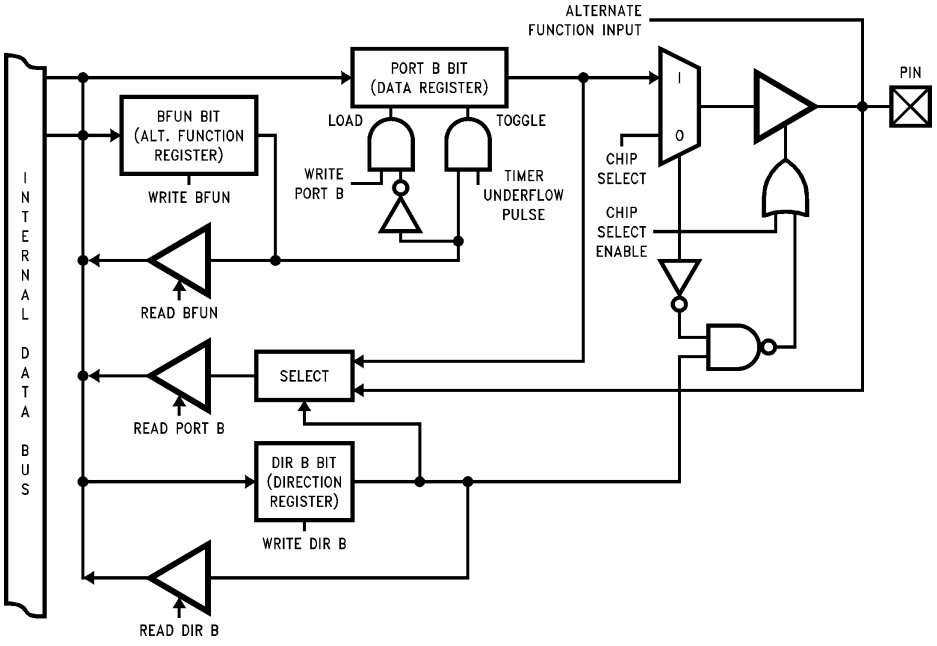
FIGURE 12. Structure of Port B Pins B0, B1, B2, B5, B6 and B7 (Typical Pins)

I/O Ports (Continued)



TL/DD/11289–15

FIGURE 13. Structure of Port B Pins B3 and B4 (Timer Synchronous)



TL/DD/11289–16

FIGURE 14. Structure of Port B Pins B8, B9, B13 and B14 (Timer Synchronous)

I/O Ports (Continued)

PORT I

Port I is an 8-bit input port that can be read as general purpose inputs and can also be used for the following functions:

Pin	Alternate	Function
I0	R5B	Timer T5 R5B Input
I1	NMI	Nonmaskable Interrupt Input
I2	INT2	Maskable Interrupt/Input Capture
I3	INT3	Maskable Interrupt/Input Capture
I4	INT4	Maskable Interrupt/Input Capture
I5	SI	MICROWIRE/PLUS Data Input
I6	RDX	UART Data Input
I7	R6B	Timer T6 R6B Input and A/D Trigger Input

PORT D

Port D is an 8-bit input port that can be used as general purpose digital inputs and as analog inputs for the A/D converter.

PORT P

Port P is a 3-bit input/output port that is used as general purpose outputs, or I/O that is controlled by timers T4, T5 and T6. These pins can be configured as Pulse Width Modulated Outputs (PWM), capture inputs or event counter inputs.

POWER SUPPLY PINS

Four pairs of power supply pins are provided to minimize cross talk between the analog, digital, and output driver sections of the chip.

Pin	Function
V_{CC}	Power for Digital Logic
GND	Ground for Digital Logic
DV_{CC}	Power for Output Drivers
DGND	Ground for Output Drivers
AV_{CC}	Power for Analog Logic
AV_{SS}	Ground for Analog Logic
V_{REF}	A/D Converter Reference Voltage Input
AGND	Ground Reference for Analog Logic

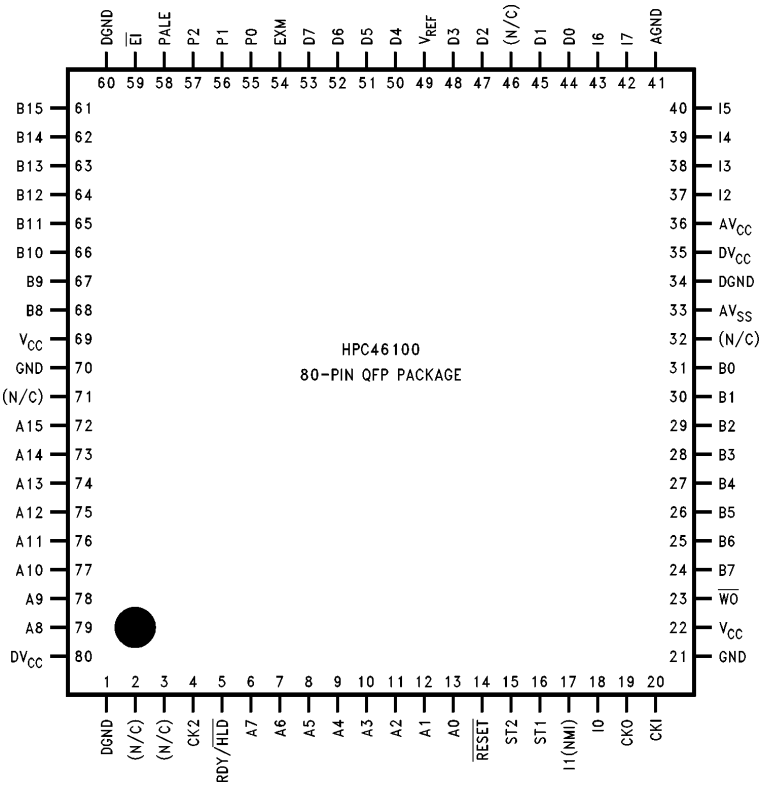
CLOCK PINS

Pin	Function
CKI	System Oscillator Input/External Clock Input
CKO	System Oscillator Output (Inversion of CKI)
CK2	Clock Output (CKI divided by 2)

OTHER PINS

Pin	Function
$\overline{\text{RESET}}$	System reset input, active low.
PALE	Slow peripheral address latch enable.
RDY/$\overline{\text{HLD}}$	Has two uses, selected by a software bit. It's either a READY input to extend the bus cycle for slower memories, or HOLD request input to put the bus in a high impedance state for DMA purposes.
$\overline{\text{WO}}$	This is an active low open drain output that signals an illegal situation has been detected by the WATCHDOG logic.
ST1	Bus Cycle Status Output: indicates first opcode fetch.
ST2	Bus Cycle Status Output: indicates machine states (skip, interrupt and first instruction cycle).
EXM	External memory enable (active high) disables internal ROM and maps it to external memory.
$\overline{\text{EI}}$	Has two uses, it's either an active low level external interrupt with vector address FFF3:FFF2 which is shared with the UART, or Timer T4 R4B input.

Connection Diagram



TL/DD/11289–17

Operating Modes

The HPC46100 does not have any internal ROM, and has only one mode of operation, the Expanded ROMless Mode. The EXM pin must be pulled high (logic "1"). The EA bit in the PSW register of the HPC46100 is hard wired to a logic "1". The use of this bit is reserved. Currently, the HPC46100 is intended for use with external memory. The external memory may be any combination of ROM, RAM, or peripherals and may be configured with an 8-bit or 16-bit external address/data bus (see *Figure 16* and *Figure 17*). Up to 62k bytes of external memory may be accessed.

Wait States

The HPC46100 provides four selectable Wait States that allow access to slower memories. The Wait States are selectable by the state of two bits in the PSW register or by two bits in the Chip Select Control registers. Additionally, the RDY input may be used to extend the instruction or memory access cycle, allowing the user to interface with slow memories and peripherals. There also is a slow peripheral bus mode when using the Chip Select logic.

Power Save Modes

Two power saving modes are available on the HPC46100: HALT and IDLE. In the HALT mode, all processor activities are stopped. In the IDLE mode, the internal oscillator and timer T0 are active but all other processor activities are stopped. In either mode, all internal RAM, registers and I/O are unaffected.

HALT MODE

The HPC46100 is placed in the HALT mode under software control by setting bits in the PSW. All processor activities,

Power Save Modes (Continued)

including the clock and timers, are stopped when the HALT mode is entered with the NMI input high. When the HALT mode is entered with the NMI input low, the high speed timers (T4, T5 and T6), remain active. In the HALT mode, power requirements for the HPC46100 are minimal and the applied voltage (V_{CC}) may be decreased without altering the state of the machine. There are two ways of exiting the HALT mode: via the $\overline{\text{RESET}}$ or the NMI. The $\overline{\text{RESET}}$ input re-initializes the processor. Use of the NMI input will generate a vectored interrupt and resume operation from that point with no initialization. The HALT mode can be enabled or disabled by means of a control register HALT enable. To prevent accidental use of the HALT mode the HALT enable register can be modified only once.

IDLE MODE

THE HPC46100 is placed in the IDLE mode through the PSW. In this mode, all processor activity, except the internal oscillator, the high speed timers (T4, T5, and T6), and Timer T0, is stopped. As with the HALT mode, the processor is returned to full operation by the $\overline{\text{RESET}}$ or NMI inputs, but without waiting for oscillator stabilization. A timer T0 overflow will also cause the HPC46100 to resume normal operation.

HPC46100 Interrupts

Complex interrupt handling is easily accomplished by the HPC46100's vectored interrupt scheme. There are eight possible interrupt sources as shown in Table I.

TABLE I. Interrupts

Vector Address	Interrupt Source	Arbitration Ranking
FFFF:FFFE	$\overline{\text{RESET}}$	0
FFFD:FFFC	Non-maskable external on rising edge of I1 pin	1
FFFB:FFFA	External interrupt on I2 pin	2
FFF9:FFF8	External interrupt on I3 pin	3
FFF7:FFF6	External interrupt on I4 pin	4
FFF5:FFF4	Overflow on internal timers	5
FFF3:FFF2	Internal by UART or external on $\overline{\text{EI}}$ pin	6
FFF1:FFF0	A/D converter	7

INTERRUPT ARBITRATION

The HPC46100 contains arbitration logic to determine which interrupt will be serviced first if two or more interrupts occur simultaneously. The arbitration ranking is given in Table I. The interrupt on $\overline{\text{RESET}}$ has the highest rank and is serviced first.

INTERRUPT PROCESSING

Interrupts are serviced after the current instruction is completed and except for the $\overline{\text{RESET}}$, which is serviced immediately. $\overline{\text{RESET}}$ and $\overline{\text{EI}}$ are level-LOW-sensitive interrupts. All other external interrupts are edge-sensitive. NMI is positive-edge sensitive. The external interrupts on I2, I3 and I4 can be software selected to be rising or falling edge.

INTERRUPT CONTROL REGISTERS

The HPC46100 allows the various interrupt sources and conditions to be programmed. This is done through the various control registers. A brief description of the different control registers is given below.

INTERRUPT ENABLE REGISTER (ENIR)

$\overline{\text{RESET}}$ and the External Interrupt on I1 are non-maskable interrupts. The other interrupts can be individually enabled or disabled. Additionally, a Global Interrupt Enable Bit in the ENIR Register allows the Maskable interrupts to be collectively enabled or disabled. Thus, in order for a particular interrupt to request service, both the individual enable bit and the Global Interrupt bit (GIE) have to be set.

INTERRUPT PENDING REGISTER (IRPD)

The IRPD register contains a bit allocated for each interrupt vector excluding the $\overline{\text{EI}}$ interrupt which has a dedicated register containing an interrupt enable and pending bit. The occurance of specified interrupt trigger conditions causes the appropriate bit to be set. There is no indication of the order in which the interrupts have been received. The bits are set independently of the fact that the interrupts may be disabled. IRPD is a Read/Write register. The bits corresponding to the maskable, external interrupts are normally cleared by the HPC46100 after servicing the interrupts. For the interrupts from the on-board peripherals, the user has the responsibility of resetting the interrupt pending flags through software. The NMI bit is read only and I2, I3, and I4 are designed as to only allow a zero to be written to the pending bit (writing a one has no effect). A LOAD IMMEDIATE instruction is to be the only instruction used to clear a bit or bits in the IRPD register. This allows a mask to be used, thus ensuring that the other pending bits are not affected.

INTERRUPT CONDITION REGISTER (IRCD)

Three bits of the register select the input polarity of the external interrupt on I2, I3 and I4.

$\overline{\text{EI}}$ INTERRUPT CONFIGURATION REGISTER (EICON)

The $\overline{\text{EI}}$ pin is an active low level sensitive interrupt. Interrupts from the $\overline{\text{EI}}$ pin are enabled by using the EICON register.

SERVICING THE INTERRUPTS

The Interrupt, once acknowledged, pushes the program counter (PC) onto the stack then incrementing the stack pointer (SP) by two. The Global Interrupt Enable bit (GIE) is copied into the CGIE bit of the PSW register; it is then reset, thus disabling further interrupts. The program counter is loaded with the contents of the memory at the vector address and the processor resumes operation at this point. At the end of the interrupt service routine, the user does a RETI instruction to pop the stack and re-enable interrupts if the CGIE bit is set, or RET to just pop the stack if the CGIE bit is clear, and then returns to the main program. The GIE bit can be set in the interrupt service routine to nest interrupts if desired. *Figure 15* shows the interrupt enable logic.

RESET

$\overline{\text{RESET}}$ is an active-low Schmitt trigger input that initializes the processor and sets all pins in a TRI-STATE condition except for CK0, CKI, ST1, ST2, $\overline{\text{HBE}}$ and $\overline{\text{WO}}$ when held low. When rising edge is detected on $\overline{\text{RESET}}$, the processor vectors to FFFF:FFFE and resumes operation at the address contained at that memory location.

HPC46100 Interrupts (Continued)

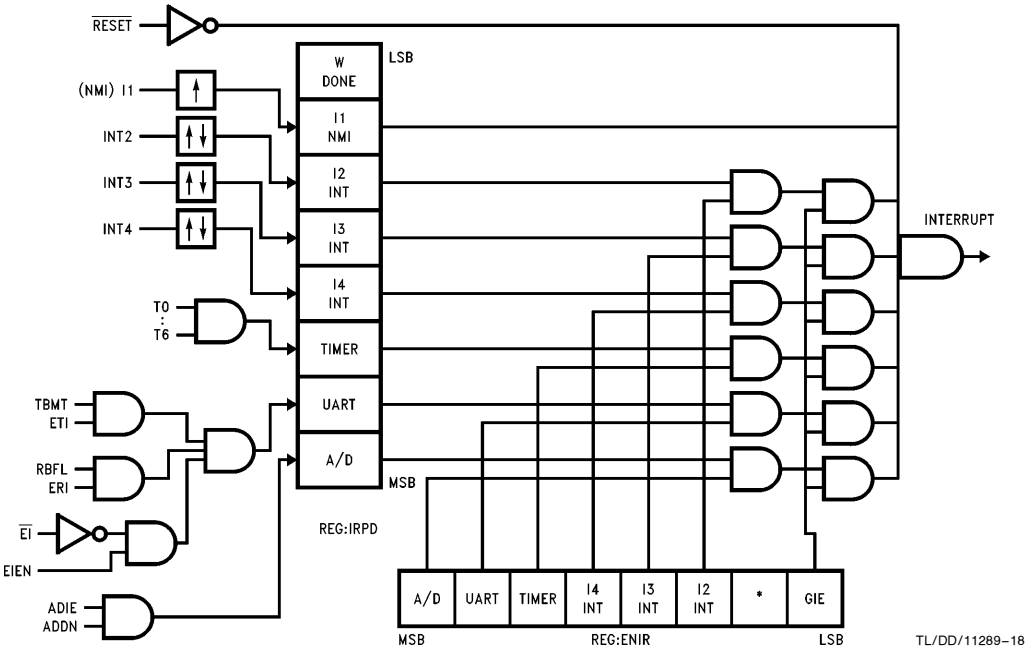


FIGURE 15. Interrupt Enable Logic

Multiply/Accumulate Unit (MAU)

This is dedicated hardware that is supported by instructions which perform basic multiply-accumulate DSP steps for FIR and IIR filter calculations, an arithmetic right shift of the Math unit Result Register (MRR), and a signed multiply of two 16-bit values producing a 32-bit result.

The MACZ and MAC instructions support the MAU by fetching data and performing circular buffer management in parallel with the multiplication. The MACZ instruction is used initially, and it is followed by a string of additional MAC instructions, one instruction per filter tap (including the MACZ). The source values are taken as 16-bit values; there is not a form that operates on byte-wide inputs. The MACZ instruction clears the result register before completing the first multiply operation.

The MAC instruction is opcode 38 hex. The MACZ instruction is opcode 39 hex. Both instructions are one byte in length, and neither allows use of an Addressing Directive prefix.

The specific function performed is as follows:

- Clear MRR to zero (MACZ instruction only).
- Fetch 16-bit data pointed to by B, and issue it to the MAU as the first operand.
- Increment B by two (bytes), compare B with K: if B > K then load B from A.
- Fetch 16-bit data pointed to by X, increment X by two (bytes).
- Issue data to the MAU to start multiply-accumulate operation. The MAU multiplies the two operands issued to it, and adds the 32-bit result to the current 32-bit contents of the MRR register.

On completion, the result goes to the MRR register. The MVP bit in the PSW is set to a ''1'' if a signed (2's complement) overflow occurred in the positive direction as a result of the accumulation substep (overflow from the multiplication substep is impossible). If the overflow occurred in the negative direction, the MVN bit is set instead. Neither of these bits is affected by the MAU if the other is already set.

By stringing together a sequence of MAC instructions, the HPC46100 can do a multiply-accumulate every 9 cycles (assuming a 1 wait state instruction fetch). At 40 MHz, this gives a 450 ns multiply-accumulate (including data fetching and circular buffer management). This can be reduced to 400 ns if executed from internal RAM.

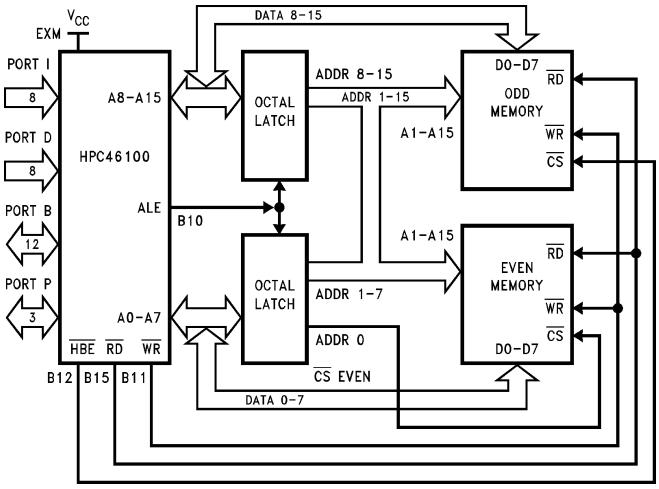
Chip Select Signals

CHIP SELECT LOGIC

The chip select logic can produce up to four chip select signals without any off-chip logic. The chip select logic supports two bus timing modes. The first bus timing mode is native bus mode, which is the standard bus mode of all HPC family members. The second bus mode is the slow peripheral bus mode, which allows the HPC46100 to interface with slow peripherals without external chip select logic. There is an additional data strobe provided for auxiliary bus timing. This auxiliary strobe is called the E signal.

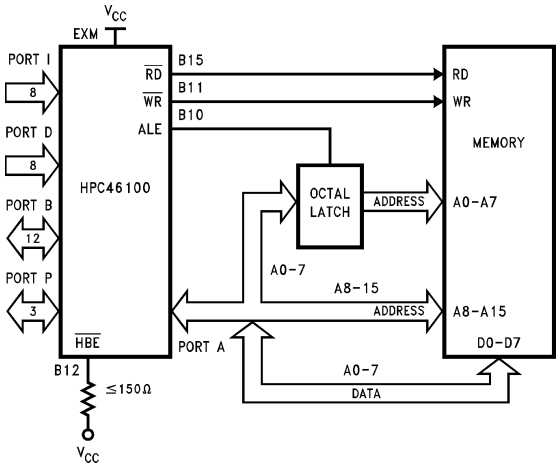
Each of chip select signals is controlled by a dedicated chip select control register (CSC0–CSC3). The control registers contain one bit to enable/disable the chip select signals, one bit to select the polarity of the chip select signal, two bits program the wait states of the data accesses, four bits that select the address range which the chip select signals are active in and one bit to enable or disable the E signal.

Chip Select Signals (Continued)



TL/DD/11289–19

FIGURE 16. 16-Bit External Address/Data Bus



TL/DD/11289–20

FIGURE 17. 8-Bit External Address/Data Bus

Chip Select Signals (Continued)

The Chip Select Address Range Selection (SEL) defines the address range the chip select is valid over. For Chip Select 0 (CSC0), the address range starts at location 0800 hex and extends to S_0FFF hex, where S_0 is the 4-bit contents of the SEL field. For both Chip Select 1 (CSC1) and Chip Select 2 (CSC2) the SEL field defines a single 4 kbyte range: $S_1$000 through S_1FFF for CSC1, and $S_2$000 through S_2FFF for CSC2, where S_1 and S_2 are their respective SEL field contents. These ranges can be used to control peripherals by using the ''Slow Peripheral'' bus timing mode. Chip Select 3 (CSC3) defines a range beginning at $S_3$000 hex, and continuing through FFFF hex. This range will typically define the off-chip ROM space. See *Figure 18*.

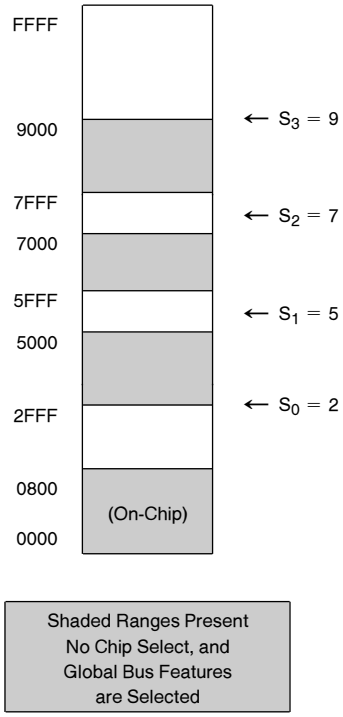
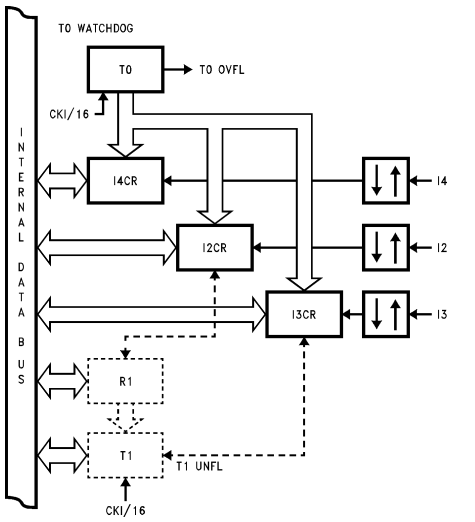


FIGURE 18. Chip Select Address Ranges

Chip Select 0 and Chip Select 3 can be programmed independently to operate with 1, 2 or 4 wait states. Chip Select 1 and 2 can be programmed independently to operate in native bus mode with 1, 2 or 4 wait states or slow peripheral bus mode.

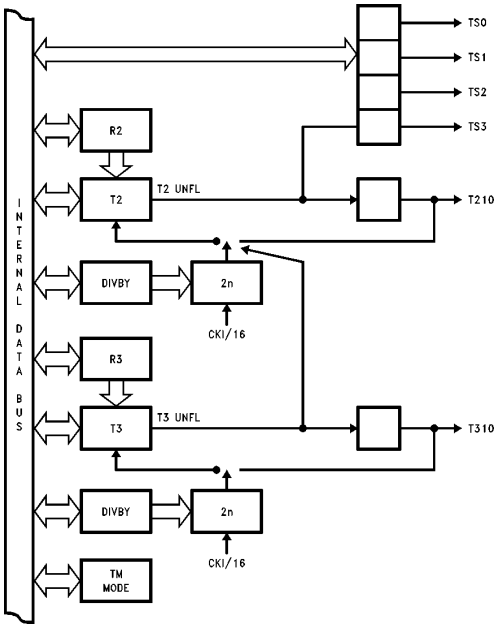
Timer Overview

The HPC46100 contains seven 16-bit timers, four core timers and three high speed timers. Timers T0–T3 are the standard core timers and are fully compatible with the core timers on other HPC family members. See *Figure 19* and *Figure 20*.



TL/DD/11289–21

FIGURE 19. Timers T0–T1 Block



TL/DD/11289–22

FIGURE 20. Timers T2–T3 Block

Timer Overview (Continued)

CORE TIMERS

Timer T0 is a free-running timer, counting up at a fixed CKI/16 (Clock Input/16) rate. It is used for Watchdog logic, high speed event capture, and to exit from the IDLE mode. Consequently, it cannot be stopped or written to under software control. Timer T0 permits precise measurements by means of the capture registers I2CR, I3CR, and I4CR. Registers I2CR and I3CR have the alternate function of being R1 and T1 respectively. The function of these registers (I2CR/R1 and I3CR/T1) are mutually exclusive and under the control of software. The capture registers I2CR, I3CR, and I4CR respectively, record the value of timer T0 when specific events occur on the interrupt pins I2, I3, and I4. The control register IRCD programs the capture registers to trigger on either a rising edge or a falling edge of its respective input. The specified edge can also be programmed to generate an interrupt.

The timers T2 and T3 have a clock rate which is selectable. The clock input to these two timers may be selected from the following two sources: an external pin, or derived internally by dividing the clock input. Timer T2 has additional capability of being clocked by the timer T3 underflow. This allows the user to cascade timers T3 and T2 into a 32-bit timer/counter.

The timers T1 through T3 in conjunction with their registers form Timer-Register pairs. All the Timer-Register pairs can be read from or written to. Each timer can be started or stopped under software control. Once enabled, the timers count down, and upon underflow, the contents of its associated register are automatically loaded into the timer.

Synchronous Outputs

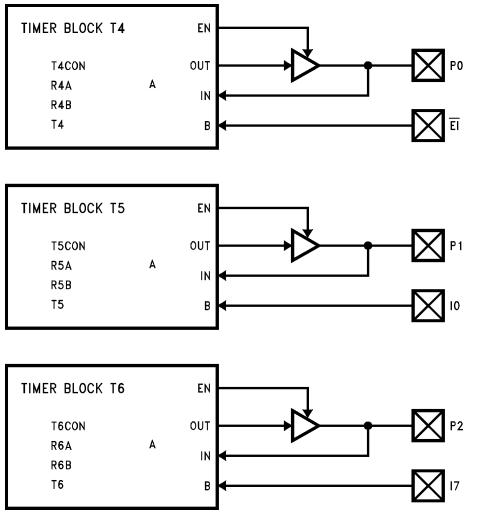
There are four synchronous timer outputs (TS0 through TS3) that work in conjunction with the timer T2. The synchronous timer outputs can be used either as regular outputs or individually programmed to toggle on timer T2 underflow.

Note: These outputs are shared with the chip select outputs. The use of these two functions are mutually exclusive.

TIMERS T4, T5 AND T6

The HPC46100 has a set of three powerful timers/counters, T4, T5 and T6. Since the three timers, T4, T5 and T6 are identical, all comments are equally applicable to any of the three timer blocks.

These timers are designed to allow the device to easily perform all timer functions with minimal software overhead. All timers are synchronized on the first overflow of timer T0. Each timer has four 16-bit registers dedicated to it, a control register (TnCON), timer register (Tn), and two auto-load/capture registers (RnA, RnB). *Figure 21* shows a block diagram of the three high speed timers.



TL/DD/11289–23

FIGURE 21. High Speed Timers Block

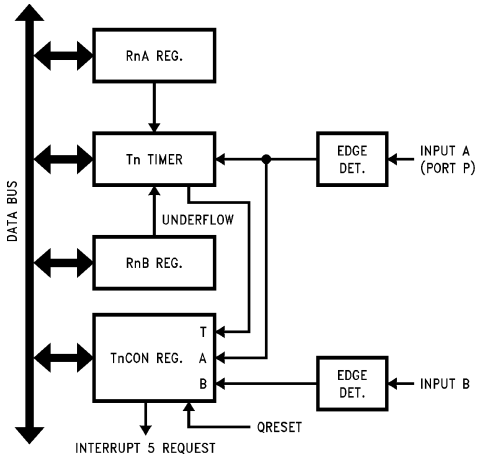
TIMER CONTROL REGISTERS

There are three timer control registers (T4CON, T5CON and T6CON). These control registers have bits which set the clock input rate, mode of operation and interrupt control structure. Each timer control register has interrupt pending and interrupt acknowledge bits for Tn, RnA and RnB, and a global interrupt pending bit for that specific timer. There are bits to enable/disable the interrupts from Tn, RnA and RnB. The clock input rate can be selected to be CKI/2, CKI/4, CKI/8 or CKI/16. The Four modes of operation are: External Event Counter mode, Input Capture mode, Processor Independent PWM mode and externally triggered PWM mode.

MODE 0. EXTERNAL EVENT COUNTER MODE

This mode is the default after $\overline{\text{RESET}}$. In this mode the timer register Tn is decremented each time there is an active edge on the A input. The active edge is determined by the value of a bit in the control register. Upon every underflow of the timer register (Tn), the timer (Tn) is aternately reloaded with the contents of the supporting registers RnA and RnB. The first underflow of the timer after entry into this mode will cause the timer to reload from register RnA. All following underflows will alternate which reload is used beginning with RnB. Every underflow from the timer will set a Tn global interrupt pending bit in the control register. The selected edge on Input A and Input B will also set corresponding pending bits in the control register. *Figure 22* shows a block diagram of the high speed timers in Mode 0.

Timer Overview (Continued)

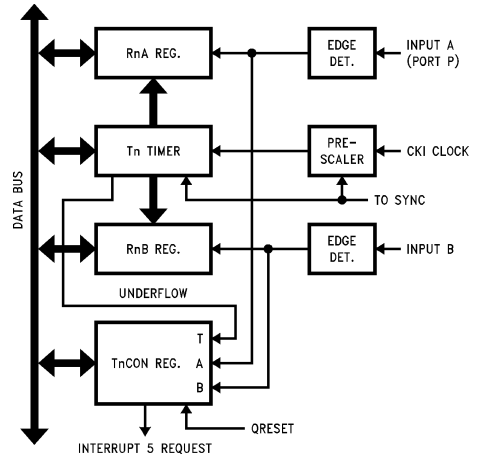


TL/DD/11289–24

FIGURE 22. External Event Counter

MODE 1. DUAL INPUT CAPTURE MODE

The device can precisely measure external frequencies or time external events by placing the timer in the input capture mode. In this mode, the timer Tn is constantly running at a fixed rate as selected in the timer control register. The two registers, RnA and RnB, act as capture registers. Each register is loaded with the contents of the timer register Tn when an active edge on it's associated pin is detected. The active edge is determined by the value of two bits in the control register. The active edge for each input pin can be specified independently, and can be programmed to generate an interrupt. The interrupt can be generated on an input from the A or B input along with an underflow of the timer register (Tn). *Figure 23* shows a block diagram of the timer in Input Capture mode.

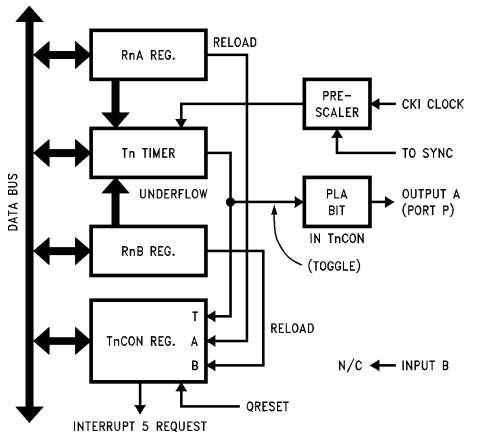


TL/DD/11289–25

FIGURE 23. Dual Input Capture

MODE 2. PROCESSOR INDEPENDENT PWM MODE

As the name suggests, this mode allows the device to generate a PWM signal with very minimal user intervention. The user only has to define the parameters of the PWM signal (ON time and OFF time). Once begun, the timer block will continuously generate the PWM signal completely independent of the microcontroller. The user software services the timer block only when the PWM parameters require updating. In this mode the timer Tn counts down at a fixed rate as programmed in the control register. Upon every underflow the timer is alternately reloaded with the contents of its supporting registers, RnA and RnB. The very first underflow of the timer after entry into this mode causes the timer to reload from the register RnA. Subsequent underflows cause the timer to be reloaded from the registers alternately beginning with the register RnB. *Figure 24* shows a block diagram of the timer in PWM mode.



TL/DD/11289–26

FIGURE 24. PWM

The underflow can be programmed to toggle the A output pin (Port P). The underflow can also be programmed to generate interrupts. These interrupts can occur on a reload from RnA or RnB. This gives the user the flexibility of interrupting once per PWM period on either the rising or falling edge of the PWM output. Alternatively, the user may choose to interrupt on both edges of the PWM output.

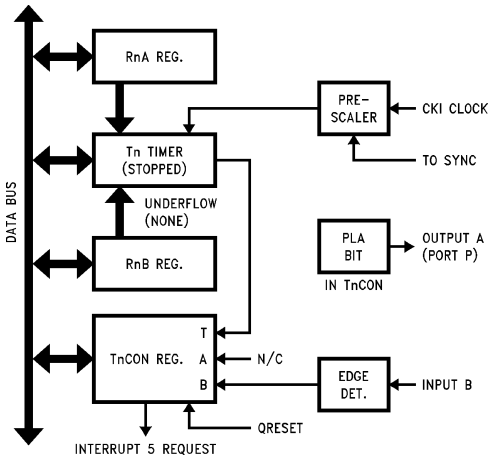
MODE 3. EXTERNALLY TRIGGERED PWM/PORT OUTPUT

In this mode the timer is stopped and the corresponding port P pin can be programmed as a general purpose output pin. The timer block can be programmed to remain halted using its A pin for use as a general purpose output, or it can be programmed to exit mode 3 and enter mode 2 (Processor Independent PWM) when a rising edge is detected on the B input.

The external triggering of this feature provides the capability of generating a delayed pulse triggered by an external event as follows: From the rising edge of the B input the timer enters PWM mode. When the timer register underflows the output toggles and the value of RnA is loaded to the timer. The next underflow will cause the RnB register to load into

Timer Overview (Continued)

the timer and operation will continue as in the PWM mode. Alternately, an interrupt from the RnB load can be used to branch to a routine that would set the timer into mode 3 waiting for the trigger for another pulse. *Figure 25* shows a block diagram of the timer in this mode.



TL/DD/11289–27

FIGURE 25. Port Output/Externally Triggered PWM

WATCHDOG LOGIC

The WATCHDOG Logic monitors the operations taking place and signals upon the occurrence of any illegal activity. The illegal conditions that trigger the WATCHDOG logic are potentially infinite loops. Should the WATCHDOG register not be written to before Timer T0 overflows twice, or more often than once every 4096 counts, an infinite loop condition is assumed to have occurred. Any illegal condition forces the WATCHDOG Output ($\overline{WO}$) pin low. The $\overline{WO}$ pin is an open drain output and can be connected to the $\overline{RESET}$ or NMI inputs or to the users external logic.

MICROWIRE/PLUS

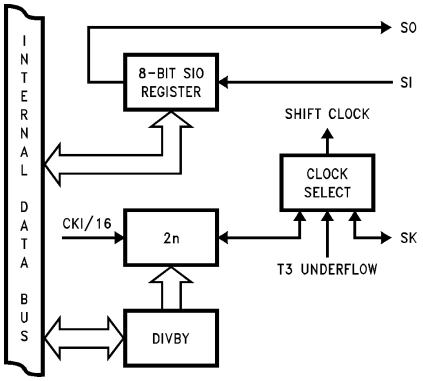
MICROWIRE/PLUS is used for synchronous serial data communications and has an 8-bit parallel-loaded, serial shift register using SI as the input and SO as the output. SK is the clock for the serial shift register (SIO). The SK clock signal can be provided by an internal or external source. The internal clock rate is programmable by the DIVBY register. A DONE flag indicates when the data shift is completed.

The MICROWIRE/PLUS capability enables it to interface with any of National Semiconductor's MICROWIRE peripherals (i.e., A/D converters, display drivers, EEPROMs).

MICROWIRE/PLUS OPERATION

The HPC46100 can enter the MICROWIRE/PLUS mode as the master or a slave. A control bit in the IRCD register determines whether the HPC46100 is the master or slave. The shift clock is generated when the HPC46100 is configured as a master. An externally generated shift clock on the SK pin is used when the HPC46100 is configured as a slave.

When the HPC46100 is a master, the DIVBY register programs the frequency of the SK clock. The DIVBY register allows the SK clock frequency to be programmed in 15 selectable steps from 64 Hz to 1 MHz with CKI at 16.0 MHz. The contents of the SIO register may be accessed through any of the memory access instructions. Data waiting to be transmitted in the SIO register is clocked out on the falling edge of the SK clock. Serial data on the SI pin is clocked in on the rising edge of the SK clock.



TL/DD/11289–28

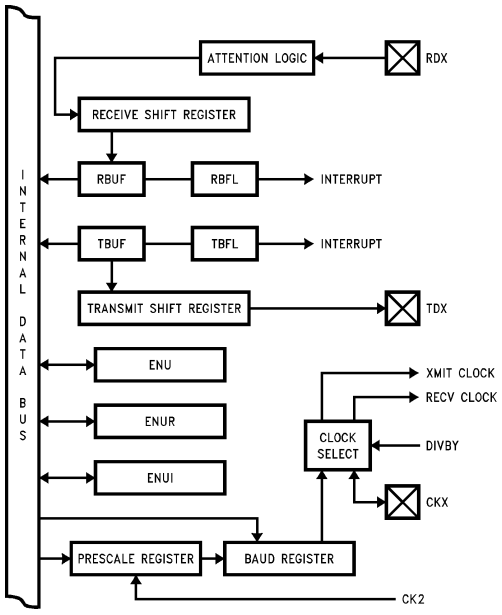
FIGURE 26. MICROWIRE/PLUS

HPC46100 UART

The HPC46100 contains a software programmable UART. The UART (see *Figure 27*) consists of a transmit shift register, a receiver shift register and five addressable registers, as follows: a transmit buffer register (TBUF), a receiver buffer register (RBUF), a UART control and status register (ENU), a UART receive control and status register (ENUR) and a UART interrupt and clock source register (ENUI). The ENU register contains flags for transmit and receive functions; this register also determines the length of the data frame (7, 8 or 9 bits) and the value of the ninth bit in transmission. The ENUR register flags framing, parity, and data overrun errors while the UART is receiving. Other functions of the ENUR register include saving the ninth bit received in the data frame, reporting receiving and transmitting status, and enabling or disabling the UART's Attention Mode of operation.

The determination of an internal or external clock source is done by the ENUI register, as well as selecting the number of stop bits ($\frac{7}{8}$, 1, $1\frac{7}{8}$ or 2 stop bits), selecting between the synchronous or asynchronous mode and enabling or disabling transmit and receive interrupts. The clock inputs to the Transmitter and Receiver sections of the UART can be individually selected to come from either an off-chip source on the CKX pin or one of the two on-chip sources. The Divide-By (DIVBY) Register provides upward compatibility from earlier HPC family members, and the most flexible and accurate on-chip clocking is provided by the Baud Rate Generator (BRG).

HPC46100 UART (Continued)



TL/DD/11289–29

FIGURE 27. UART Block Diagram

The Baud Rate Generator is controlled by the register pair PSR and BAUD. The Prescaler factor is selected by the upper 5 bits of the PSR register (the PRESCALE field), in units of the CK2 clock from 1 to 16 in ½ step increments. The lower 3 bits of the PSR register, in conjunction with the 8 bits of the baud register, form the 11-bit BAUDRATE field, which defines a baud rate divisor ranging from 1 to 2048, in units of the prescaled clock selected by the PRESCALE field.

In Asynchronous Mode, the resulting baud rate is $\frac{1}{16}$ of the clocking rate selected through the BRG circuit. The maximum baud rate generated using the BRG is 625 kbaud.

In the Synchronous Mode data is transmitted on the rising edge and received on the falling edge of the external clock. Although the data is transmitted and received synchronously, it is still contained within an asynchronous frame; i.e., a start bit, parity bit (if selected) and stop bit(s) are still present.

UART ATTENTION MODE

The HPC46100 UART features an Attention Mode of operation. This mode of operation enables the HPC46100 to be networked with other processors. Typically in such environments, the messages consist of addresses and actual data. Addresses are specified by having the ninth bit in the data frame set to 1. Data in the message is specified by having the ninth bit in the data frame reset to 0. The UART monitors the communication stream looking for addresses. When the data word with the ninth bit set is received, the UART signals the HPC46100 with an interrupt. The processor then examines the content of the receiver buffer to decide whether it has been addressed and whether to accept subsequent data.

A/D Converter

The HPC46100 has an on-board eight-channel 8-bit Analog to Digital converter. Conversion is performed using a successive approximation technique. The A/D converter cell can operate in single-ended mode where the input voltage is applied across one of the eight input channels (D0–D7) and AGND or in differential mode where the input voltage is applied across two adjacent input channels. The A/D converter will convert up to eight channels in single-ended mode and up to four channel-pairs in differential mode.

OPERATING MODES

The operating modes of the converter are selected by 4 bits called ADMODE (ADCR2.4–7) see Table II. Associated with the eight input channels in single-ended mode are eight result registers, one for each channel. The A/D converter can be programmed by software to convert on any specific channel storing the result in the result register associated with that channel. It can also be programmed to stop after one conversion or to convert continuously. If a brief history of the signal on any specific input channel is required, the converter can be programmed to convert on that channel and store the consecutive results in each of the result registers before stopping. As a final configuration in single-ended mode, the converter can be programmed to convert the signal on each input channel and store the result in its associated result register continuously.

Associated with each even-odd pair of input channels in differential mode of operation are four result register-pairs. The A/D converter performs two conversions on the selected pair of input channels. One conversion is performed assuming the positive connection is made to the even channel and the negative connection is made to the following odd channel. This result is stored in the result register associated with the even channel. Another conversion is performed assuming the positive connection is made to the odd channel and the negative connection is made to the preceding even channel. This result is stored in the result register associated with the odd channel. This technique does not require that the programmer know the polarity of the input signal. If the even channel result register is non-zero (meaning the odd channel result register is zero), then the input signal is positive with respect to the odd channel. If the odd channel result register is non-zero (meaning the even channel result register is zero), then the input signal is positive with respect to the even channel.

A/D Converter (Continued)

TABLE II. Operating Modes

Mode 0	single-ended, single channel, single result register, one-shot (default value on power-up)
Mode 1	single-ended, single channel, single result register, continuous
Mode 2	single-ended, single channel, multiple result registers, stop after 8
Mode 3	single-ended, multiple channel, multiple result register, continuous
Mode 4	differential, single channel-pair, single result register-pair, one-shot
Mode 5	differential, single channel-pair, single result register-pair, continuous
Mode 6	differential, single channel-pair, multiple result register-pairs, stop after 4 pairs
Mode 7	differential, multiple channel-pair, multiple result register-pairs, continuous
Mode 8	single-ended, single channel, single result register, one-shot (default value on power-up), quiet address/data bus
Mode C	differential, single channel-pair, single result register-pair, one-shot, quiet address/data bus

The same operating modes for single-ended operation also apply when the inputs are taken from channel-pairs in differential mode. The programmer can configure the A/D to convert on any selected channel-pair and store the result in its associated result register-pair then stop. The A/D can also be programmed to do this continuously. Conversion can also be done on any channel-pair storing the result into four result register-pairs for a history of the differential input. Finally, all input channel-pairs can be converted continuously.

The final mode of operation suppresses the external address/data bus activity during the single conversion modes. These quiet modes of operation utilize the RDY function of the HPC Core to insert wait states in the instruction being executed in order to limit digital noise in the environment due to external bus activity when addressing external memory.

CONTROL

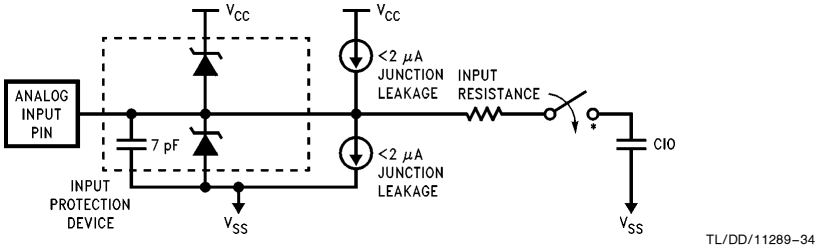
The conversion clock supplied to the A/D converter can be selected by three bits in ADCR1. These bits are used as a prescaler on CKI, and can provide a clock rate from CKI/4 to CKI/32. These bits can be used to ensure that the A/D is clocked as fast as possible when different external crystal frequencies are used. Controlling the starting of conversion cycles in each of the operating modes can be done by four different methods. The method is selected by two bits called SC (ADCR3.0–1). Conversion cycles can be initiated through software by resetting a bit in a control register, through hardware by an underflow of Timer T2, or externally by a rising or falling edge of a signal input on I7.

INTERRUPTS

The A/D converter can interrupt the HPC when it completes a conversion cycle if one of the non-continuous modes has been selected. If one of the cycle modes was selected, then the converter will request an interrupt after eight conversions. If one of the one-shot modes was selected, then the converter will request an interrupt after every conversion. When this interrupt is generated, the HPC vectors to the A/D converter interrupt vector location at address FFF0.

Analog Input and Source Resistance Considerations

Figure 28 shows the A/D pin model for the HPC46100 in single ended mode. The differential mode has similiar A/D pin model. The leads to the analog inputs should be kept as short as possible. Both noise and digital clock coupling to an A/D input can cause conversion errors. The clock lead should be kept away from the analog input line to reduce coupling. The A/D channel input pins do not have any internal output driver circuitry connected to them because this circuitry would load the analog input signals due to output buffer leakage current.



*The analog switch is closed only during the sample time.

FIGURE 28. Port D Input Structure

A/D Converter (Continued)

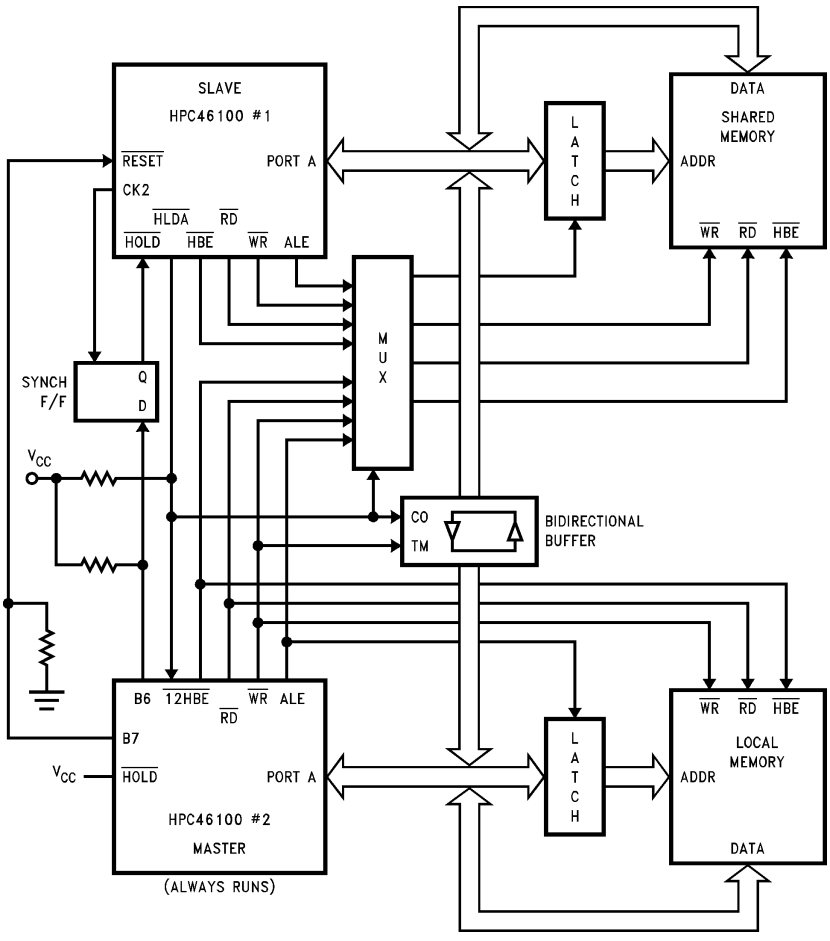
Source impedances greater than 1 kΩ on the analog input lines will adversely affect internal RC charging time during input sampling. As shown in *Figure 28,* the analog switch to the DAC array is closed only during the 2 A/D cycle sample time. Large source impedances on the analog inputs may result in the DAC array not being charged to the correct voltage levels, causing scale errors.

If large source resistance is necessary, the recommended solution is to slow down the A/D clock speed in proportion to the source resistance. The A/D converter may be operated at the maximum speed for R_S less than 1 kΩ. For R_S greater than 1 kΩ, A/D clock speed needs to be reduced. For example, with $R_S = 2$ kΩ, the A/D converter may be operated at half the maximum speed. A/D converter clock speed may be slowed down by either increasing the A/D prescaler divide-by or decreasing the CKI clock frequency. The A/D clock speed may be reduced to its minimum frequency of 100 kHz.

Shared Memory Support

Shared memory access provides a rapid technique to exchange data. It is effective when data is moved from a peripheral to memory or when data is moved between blocks of memory. A related area where shared memory access proves effective is in multiprocessing applications where two CPUs share a common memory block (see *Figure 29*). The HPC46100 supports shared memory access with two pins. The pins are the RDY/$\overline{HLD}$ input pin and the HLDA output pin. The user can software select either the Hold or Ready function by the state of a control bit. The HLDA output is multiplexed onto port B.

The host uses DMA to interface with the HPC46100. The host initiates a data transfer by activating the HLD input of the HPC46100. In response, the HPC46100 places its system bus in a TRI-STATE Mode, freeing it for use by the host.



TL/DD/11289–31

FIGURE 29. Shared Memory Application, Using $\overline{HOLD}$

Shared Memory Support (Continued)

The host waits for the acknowledge signal (HLDA) from the HPC46100 indicating that the system bus is free. On receiving the acknowledge, the host can rapidly transfer data into, or out of, the shared memory by using a conventional DMA controller. Upon completion of the message transfer, the host removes the HOLD request and the HPC46100 resumes normal operations.

Memory

The HPC46100 has been designed to offer flexibility in memory usage. A total address space of 64 kbytes can be directly addressed including 1024 bytes of RAM available on the chip itself.

Program memory addressing is accomplished by the 16-bit program counter on a byte basis. Memory can be addressed directly by instructions or indirectly through registers or any memory word in the first 256 bytes of memory (On-Chip Basepage RAM). Memory can be addressed as words or bytes. Words are always addressed on even-byte boundaries. The HPC46100 uses memory-mapped organization to support registers, I/O and on-chip peripheral functions. The HPC46100 memory address space extends to 64 kbytes and registers and I/O are mapped as shown in Table III.

TABLE III. HPC46100 Memory Map

FFFF:FFF0 FFEF:FFD0 FFCF:0800	Interrupt Vectors JSRP Vectors External Memory	User Memory
07FF:04C0	On-Chip RAM	User RAM
04BF:0196	RESERVED	
0195:0194	Watchdog Register	Watchdog Logic
0192 0191:0190 018F:018E 018D:018C 018B:018A 0189:0188 0187:0186 0185:0184 0183:0182 0181:0180	T0CON Register TMMODE Register DIVBY Register T3 Timer R3 Register T2 Timer R2 Register I2CR Register/R1 I3CR Register/T1 I4CR Register	Timer Block T0:T3
017F:0168	RESERVED	
0167:0166 0165:0164 0163:0162 0161:0160	CSC3 Register CSC2 Register CSC1 Register CSC0 Register	Chip Select Control
015F:0158	RESERVED	
0157:0156 0155:0154 0153:0152 0151:0150	T6 Timer T6CON Register R6A Register R6B Register	Timer T6
014F:012D	RESERVED	
012C 012A 0128 0126 0124 0122 0120	Baud Register PSR Register ENUR Register TBUF Register RBUF Register ENUI Register ENU Register	UART
011E 011C 011A 0118 0116 0114 0112 0110 010F:0109	A/D Result Register 7 A/D Result Register 6 A/D Result Register 5 A/D Result Register 4 A/D Result Register 3 A/D Result Register 2 A/D Result Register 1 A/D Result Register 0 RESERVED	A/D Converter
0108	EICON Register	EI Pin Control
0106 0104 0102 0100	ADCR3 Register PORTD Register ADCR2 Register ADCR1 Register	A/D Control
00FF:00FE 00FD:00FC 00FB:00FA 00F9:00F8	T5 TIMER T5CON Register R5A Register R5B Register	Timer 5
00F7:00F6	RESERVED	
00F5:00F4 00F3:00F2 00F1:00F0	BFUN Register DIR B Register RESERVED FOR DIRA	Ports A&B Control
00EF:00EE 00ED:00EC 00EB:00EA 00E9:00E8 00E7:00E6	T4 TIMER T4CON Register R4A Register R4B Register RESERVED FOR UPIC	Timer 4
00E5:00E6	RESERVED	
00E3:00E2 00E1:00E0	PORTB Register RESERVED FOR PORTA	Ports A&B
00DF:00DE 00DD:00DC 00DA	MRU (MRR upper) MRL (MRR lower) MIR	Math Unit
00D8 00D6 00D4 00D2 00D0	PORTI Register SIO Register IRCD Register IRPD Register ENIR Register	Interrupt Control Registers
00CF:00CE 00CD:00CC 00CB:00CA 00C9:00C8 00C7:00C6 00C5:00C4 00C3:00C2 00C1:00C0	X Register B Register K Register A Register PC Register SP Register HALTEN Register PSW Register	HPC Core Registers
00BF:0000	On-Chip RAM	Basepage RAM

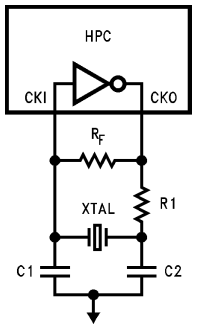
Design Considerations

Designs using the HPC family of 16-bit high speed CMOS microcontrollers need to follow some general guidelines on usage and board layout.

Floating inputs are a frequently overlooked problem. CMOS inputs have extremely high impedance and, if left open, can float to any voltage. You should thus tie unused inputs to V_{CC} or ground, either through a resistor or directly. Unlike the inputs, unused outputs should be left floating to allow the output to switch without drawing any DC current.

To reduce voltage transients, keep the supply line's parasitic inductances as low as possible by reducing trace lengths, using wide traces, ground planes, and by decoupling the supply with bypass capacitors. In order to prevent additional voltage spiking, this local bypass capacitor must exhibit low inductive reactance. You should therefore use high frequency ceramic capacitors and place them very near the IC to minimize wiring inductance.

- Keep V_{CC} bus routing short. When using double sided or multilayer circuit boards, use ground plane techniques.
- Keep ground lines short, and on PC boards make them as wide as possible, even if trace width varies. Use separate ground traces to supply high current devices such as relay and transmission line drivers.
- In systems mixing linear and logic functions and where supply noise is critical to the analog components' performance, provide separate supply buses or even separate supplies.
- If you use local regulators, bypass their inputs with a tantalum capacitor of at least 1 μF and bypass their outputs with a 10 μF to 50 μF tantalum or aluminum electrolytic capacitor.
- If the system uses a centralized regulated power supply, use a 10 μF to 20 μF tantalum electrolytic capacitor or a 50 μF to 100 μF aluminum electrolytic capacitor to decouple the V_{CC} bus connected to the circuit board.
- Provide localized decoupling. For random logic, a rule of thumb dictates approximately 10 nF (spaced within 12 cm) per every two to five packages, and 100 nF for every 10 packages. You can group these capacitances, but it's more effective to distribute them among the ICs. If the design has a fair amount of synchronous logic with outputs that tend to switch simultaneously, additional decoupling might be advisable. Octal flip-flop and buffers in bus-oriented circuits might also require more decoupling. Note that wire-wrapped circuits can require more decoupling than ground plane or multilayer PC boards.



TL/DD/11289–32

FIGURE 30. Recommended Fundamental Crystal Circuit

A recommended crystal oscillator circuit to be used with the HPC is shown in *Figure 30*. See table for recommended component values. The recommended values given in the table have yielded consistent results and are made to match a crystal with a 18 pF load capacitance, with some small allowance for layout capacitance.

For frequencies between 26 MHz and 40 MHz a third overtone frequency ''AT'' cut crystal or fundamental frequency ''BT'' cut crystal may be used. Tite ''AT'' crystal has a tighter frequency tolerance over temperature than the ''BT'' cut. The ''BT'' crystal network is easier to design due to its fundamental nature. For the ''BT'' crystal:

R_F = 1–2 MΩ, R_1 = 0Ω–100Ω

C_1 = 22 pF–27 pF, C_2 = 27 pF–53 pF.

The ''AT'' crystal can be configured in one of two ways. One circuit is shown in *Figure 31*.

where:

R_F = 1 MΩ–2 MΩ, R_1 = 0Ω–100Ω,

C_I = 22 pF–27 pF, C_2 = 27 pF–33 pF

C_3 = 15 pF–30 pF, and L_1 is determined by the equation:

$$f = \frac{1}{2\pi\sqrt{L_1 C_3}}$$

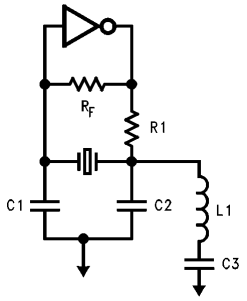
f is the frequency which is ½ the crystal frequency, from this, the value of L_1 can be calculated. The second circuit is similar to *Figure 30* where R_F = 2 kΩ, R_1 = 0Ω, C_1 = 12 pF–15 pF, C_2 = 15 pF–22 pF. The lower C_1 and C_2 values allow a greater influence to stray capacitance and EMI. The lower R_F resistance decreases gain while increasing bandwidth. The oscillator networks and component values are supplied for reference only. Actual networks and component values should be obtained from the crystal manufacturer.

XTAL Frequency (MHz)	R1 (Ω)
$\leq$2	1500
4	1200
6	910
8	750
10	600
12	470
14	390
16	300
18	220
20	180
22	150
24	120

R_F = 3.3 MΩ, C1 = 27 pF, C2 = 33 pF

XTAL Specifications: The crystal used was an M-TRON Industries MP-1 Series crystal. Fundamental frequency ''AT'' cut parallel resonant with a loading capacitance C_L = 18 pF, and a series resistance of 25Ω @ 25 MHz, 40Ω @ 10 MHz, or 600Ω @ 2 MHz.

Design Considerations (Continued)



TL/DD/11289–35

FIGURE 31. Recommended Overtone Crystal Circuit

A recommended layout for the oscillator network should be as close to the processor as physically possible, entirely within ''1'' distance. This is to reduce lead inductance from long PC traces, as well as interference from other components, and reduce trace capacitance. The layout contains a large ground plane either on the top or bottom surface of the board to provide signal shielding, and a convenient location to ground both the HPC, and the case of the crystal. It is very critical to have an extremely clean power supply for the HPC crystal oscillator. Ideally one would like a V_{CC} and ground plane that provide low inductance power lines to the chip. The power planes in the PC board should be decoupled with three decoupling capacitors as close to the chip as possible. A 1.0 μF, a 0.1 μF, and a 0.001 μF capacitor dipped mica or ceramic cap mounted as close to the HPC as physically possible on the board using short leads, or surface mount components. This should provide a stable power supply, and noiseless ground plane which will vastly improve the performance of the crystal oscillator network.

HPC46100 CPU

The HPC46100 CPU has a 16-bit ALU and six 16-bit registers.

ARITHMETIC LOGIC UNIT (ALU)

The ALU is 16 bits wide and can do 16-bit add, subtract and shift or logic AND, OR and exclusive OR in one timing cycle.

The ALU has two carry bits; one for signed overflow (V bit) and one for unsigned overflow (C bit).

ACCUMULATOR (A) REGISTER

The 16-bit A register is the source and destination register for most I/O, arithmetic, logic and data memory access operations.

ADDRESS (B AND X) REGISTERS

The 16-bit B and X registers can be used for indirect addressing. They can automatically count up or down to sequence through data memory.

BOUNDARY (K) REGISTER

The 16-bit K register is used to set limits in repetitive loops of code as register B sequences through data memory. The K register can also be used as a pointer register.

STACK POINTER (SP) REGISTER

The 16-bit SP register is the pointer that addresses the stack. The SP register is incremented by two for each push or call and decremented by two for each pop or return. The stack can be placed anywhere in user memory and be as deep as the available memory permits. The SP register can also be used as a pointer register.

PROGRAM (PC) REGISTER

The 16-bit PC register addresses program memory.

MAU RESULT REGISTER (MRR)

The 32-bit MAU Result Register holds the results from MAC (Multiply/Accumulate) instructions. In addition, it receives the result of the MULS (Multiply Signed) instruction, and can be shifted in place by the ASHR (Arithmetic Shift Right) operation.

Addressing Modes

ADDRESSING MODES WITH THE ACCUMULATOR AS DESTINATION

Register Indirect

The operand is the memory addressed by the A, B, X or K register. This mode of addressing for the HPC46100 produces single byte instructions when using the B or X register (depending on the instruction).

Direct

The instruction contains an 8-bit or 16-bit address field that directly points to the memory for the operand.

Indirect

The instruction contains an 8-bit address field. The contents of the WORD addressed points to the memory for the operand.

Indexed

The instruction contains an 8-bit address field and an 8- or 16-bit displacement field. The contents of the WORD addressed is added to the displacement to get the address of the operand.

Immediate

The instruction contains an 8-bit or 16-bit immediate field that is used as the operand.

Register Indirect (Auto Increment and Decrement)

The operand is the memory addressed by the X register. This mode automatically increments or decrements the X register (by 1 for bytes and by 2 for words).

Register Indirect Auto Increment and Decrement) with Conditional Skip

The operand is the memory addressed by the B register. This mode automatically increments or decrements the B register (by 1 for bytes and by 2 for words). The B register is then compared with the K register. A skip condition is generated if B goes past K.

ADDRESSING MODES WITH DIRECT MEMORY AS DESTINATION

Direct Memory to Direct Memory

The instruction contains two 8- or 16-bit address fields. One field directly points to the source operand and the other field directly points to the destination operand.

Immediate to Direct Memory

The instruction contains an 8- or 16-bit address field and an 8- or 16-bit immediate field. The immediate field is the operand and the direct field is the destination.

Addressing Modes (Continued)

Double Register Indirect Using the B and X Registers

Used only with Reset, Set, IF and IF NOT bit instructions; a specific bit within the 64 kbyte address range is addressed using the B and X registers. The address of a byte of memory is formed by adding the contents of the B register to the most significant 13 bits of the X register. The specific bit to be modified or tested within the byte of memory is selected using the least significant 3 bits of register X.

Code Efficiency

One of the most important criteria of a single chip microcontroller is code efficiency. The more efficient the code, the more features that can be put on a chip. The memory size on a chip is fixed so if code is not efficient, features may have to be sacrificed or the programmer may have to buy a larger, more expensive version of the chip.

The HPC46100 has been designed to be extremely code-efficient. The HPC46100 looks very good in all the standard coding benchmarks; however, it is not realistic to rely only on benchmarks. Many large jobs have been programmed onto the HPC46100, and the code savings over other popular microcontrollers has been considerable. Reasons for this saving of code include the following:

SINGLE BYTE INSTRUCTIONS

The majority of instructions on the HPC46100 are single-byte. There are two especially code-saving instructions: JP is a 1-byte jump. True, it can only jump within a range of plus or minus 32, but many loops and decisions are often within a small range of program memory.

JSRP is a 1-byte call subroutine. The user makes a table of his 16 most frequently called subroutines and these calls will only take one byte. Most other micros require two and even three bytes to call a subroutine. The user does not have to decide which subroutine addresses to put into his table; the assembler can give him this information.

EFFICIENT SUBROUTINE CALLS

The 2-byte JSR instructions can call any subroutine within plus or minus 1k of program memory.

MULTIFUNCTION INSTRUCTION FOR DATA MOVEMENT AND PROGRAM LOOPING

The HPC46100 has single-byte instructions that perform multiple tasks. For example, the XS instruction will do the following.

1. Exchange A and memory pointed to by the B register
2. Increment or decrement the B register
3. Compare the B register to the K register
4. Generate a conditional skip if B has passed K

The value of this multipurpose instruction becomes evident when looping through sequential areas of memory and exiting when the loop is finished.

BIT MANIPULATION INSTRUCTIONS

Any bit of memory, I/O or registers can be set, reset or tested by the single byte bit instructions. The bits can be addressed directly or indirectly. Since all registers and I/O are mapped into the memory, it is very easy to manipulate specific bits to do efficient control.

MULTIPLY AND DIVIDE INSTRUCTIONS

The HPC46100 has 16-bit multiply, 16-bit by 16-bit divide, and 32-bit by 16-bit divide instructions. This saves both code and time. Multiply and divide can use immediate data or data from memory. The ability to multiply and divide by immediate data saves code since this function is often needed for scaling, base conversion, computing indexes of arrays, etc.

HPC Instruction Set Description

Mnemonic	Description	Action
ARITHMETIC INSTRUCTIONS		
ADD	Add	MA + MemI ⟶ MA, carry ⟶ C
ADC	Add with carry	MA + MemI + C ⟶ MA, carry ⟶ C
SUB	Subtract without carry	MA − MemI ⟶ MA
SUBC	Subtract with carry	MA − MemI + C ⟶ MA, carry ⟶ C
MULT	Multiply (unsigned)	MA*MemI ⟶ MA & X, 0 ⟶ K, 0 ⟶ C
DIV	Divide (unsigned)	MA/MemIMA, rem. ⟶ X, 0 ⟶ K, 0 ⟶ C
DIVD	Divide Double Word (unsigned)	X & MA/MemIMA, rem. ⟶ X, 0K, Carry ⟶ C
IFEQ	If equal	Compare MA & MemI, Do next if equal
IFGT	If greater than	Compare MA & MemI, Do next if MA > MemI
IFGE	If greater than or equal	Compare MA & MemI, Do next if MA = or > MemI
IFGES	If greater than or equal signed	Signed compare MA & MemI, Do next if MA = or > MemI
IFGTS	If greater than signed	Signed compare MA & MemI, Do next if MA > MemI
AND	Logical and	MA and MemI ⟶ MA
OR	Logical or	MA or MemI ⟶ MA
XOR	Logical exclusive-or	MA xor MemI ⟶ MA
MACZ	Multiply-accumulate From	0 ⟶ MRR, [B+]*[X+] + MRR ⟶ MRR
	Zero signed word	OVERFLOW ⟶ MVP or MVN, IF B>K THEN A ⟶ B
MAC	Multiple-accumulate signed word	[B+]*[X+] + MRR ⟶ MRR, OVERFLOW ⟶ MVP or MVN, IF B>K then A ⟶ B
MULS	Multiply, signed	MA*MemI ⟶ MRR
ASHR	Arithmetic right	MRR/2^IMM ⟶ MRR, or MRR/2^A ⟶ MRR
MEMORY MODIFY INSTRUCTIONS		
INC	Increment	Mem + 1 ⟶ Mem
DECSZ	Decrement, skip if 0	Mem − 1 ⟶ Mem, Skip next if Mem = 0
BIT INSTRUCTIONS		
SBIT	Set bit	1 ⟶ Mem.bit
RBIT	Reset bit	0 ⟶ Mem.bit
IFBIT	If bit	If Mem.bit is = 1, do next instruction
IFNBIT	If not bit	If Mem.bit is = 0, do next instruction
MEMORY TRANSFER INSTRUCTIONS		
LD	Load	MemI ⟶ MA
	Load, incr/decr X	Mem(X) ⟶ A, X ± 1 (or 2) ⟶ X
LD B, mode	Load B	MemI ⟶ B
LD X, mode	Load X	MemI ⟶ X
LD K, mode	Load K	MemI ⟶ K
ST	Store to Memory	A ⟶ Mem
X	Exchange	A ⟶ Mem
	Exchange, incr/decr X	A ⟶ Mem(X), X ± 1 (or 2) ⟶ X
PUSH	Push Memory to Stack	W ⟶ W(SP), SP + 2 ⟶ SP
FETCH	Dummy read	Gen. addressed byte is read and discarded
POP	Pop Stack to Memory	SP − 2 ⟶ SP, W(SP) ⟶ W
LDS	Load A, incr/decr B,	Mem(B) ⟶ A, B ± 1 (or 2) ⟶ B,
	Skip on condition	Skip next if B greater/less than K
XS	Exchange, incr/decr B,	MEM(B) ⟵ ⟶ A, B ± 1 (or 2) ⟶ B,
	Skip on condition	Skip next if B greater/less than K

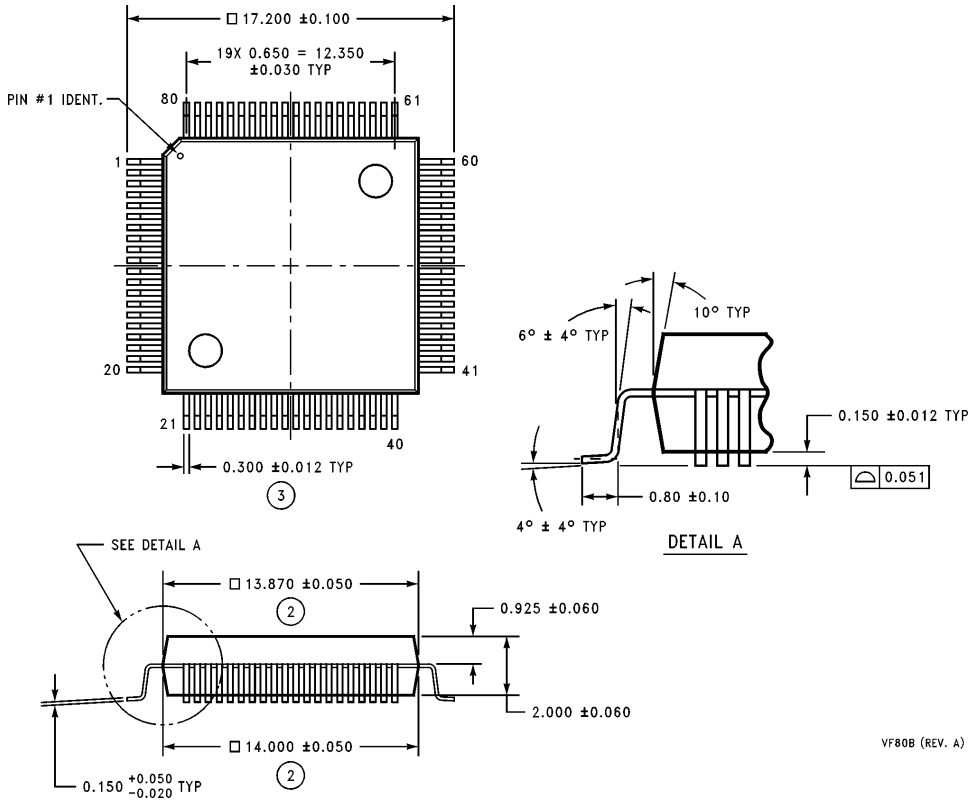
Notes: W is 16-bit word of memory
MA is Accumulator A or direct memory (8- or 16-bit)
Mem is 8-bit byte or 16-bit word of memory
MemI is 8- or 16-bit memory or 8- or 16-bit immediate data
imm is 8-bit or 16-bit immediate data
imm8 is 8-bit immediate data only

HPC Instruction Set Description (Continued)

Mnemonic	Description	Action
REGISTER LOAD IMMEDIATE INSTRUCTIONS		
LD B	Load B	Meml ⟶ B
LD K	Load B	Meml ⟶ K
LD X	Load X	Meml ⟶ X
LD BK	Load B and K immediate	imm ⟶ B, imm ⟶ K
ACCUMULATOR AND C INSTRUCTIONS		
CLR A	Clear A	0 ⟶ A
INC A	Increment A	A + 1 ⟶ A
DEC A	Decrement A	A + 1 ⟶ A
COMP A	Complement A	1's complement A ⟶ A
SWAP A	Swap nibbles of A	A15:12 ⟵ A11:8 ⟵ A7:4 ⟵ ⟶ A3:0
RRC A	Rotate A right thru C	C ⟶ A15 ⟶ .. ⟶ A0 ⟶ C
RLC A	Rotate A left thru C	C ⟵ A15 ⟵ .. ⟵ A0 ⟵ C
SHR A	Shift A right	0 ⟶ A15 ⟶ .. ⟶ A0 ⟶ C
SHL A	Shift A left	C ⟵ A15 ⟵ ⟵ A0 ⟵ 0
SC	Set C	1 ⟶ C
RC	Reset C	0 ⟶ C
IFC	IF C	Do next if C = 1
IFNC	IF not C	Do next if C = 0
TRANSFER OF CONTROL INSTRUCTIONS		
JSRP	Jump Subroutine from table	PC ⟶ W(SP),SP+2 ⟶ SP,W(table#) ⟶ PC
JSR	Jump Subroutine relative	PC ⟶ W(SP),SP+2 ⟶ SP,PC+ # ⟶ PC (# is +1025 to −1023)
JSRL	Jump Subroutine long	PC ⟶ W(SP),SP+2 ⟶ SP,PC+ # ⟶ PC
JP	Jump relative short	PC+ # ⟶ PC(# is +32 to −31)
JMP	Jump relative	PC+ # ⟶ PC(# is +257 to −255)
JMPL	Jump relative long	PC+ # ⟶ PC
JID	Jump indirect at PC+A	PC+A+1 ⟶ PC
JIDW		then Mem(PC)+PC ⟶ PC
NOP	No Operation	PC+1 ⟶ PC
RET	Return	SP−2 ⟶ SP,W(SP) ⟶ PC
RETSK	Return then skip next	SP−2 ⟶ SP,W(SP) ⟶ PC, & skip
RETI	Return from interrupt	SP−2 ⟶ SP,W(SP) ⟶ PC, interrupt re-enabled

Notes: W is 16-bit word of memory
MA is Accumulator A or direct memory (8- or 16-bit)
Mem is 8-bit byte or 16-bit word of memory
Meml is 8- or 16-bit memory or 8- or 16-bit immediate data
imm is 8-bit or 16-bit immediate data
imm8 is 8-bit immediate data only

Physical Dimensions inches (millimeters)



Plastic Quad Flatpack (VF)
Order Number HPC46100VF40
NS Package Number VF80B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408