National Semiconductor Corporation

PRELIMINARY

HPC16040/HPC26040/HPC36040/HPC46040/HPC16030/ HPC36030/HPC46030 High-Performance Microcontrollers

General Description

The HPC16040 is a member of the HPC™ family of High Performance microControllers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

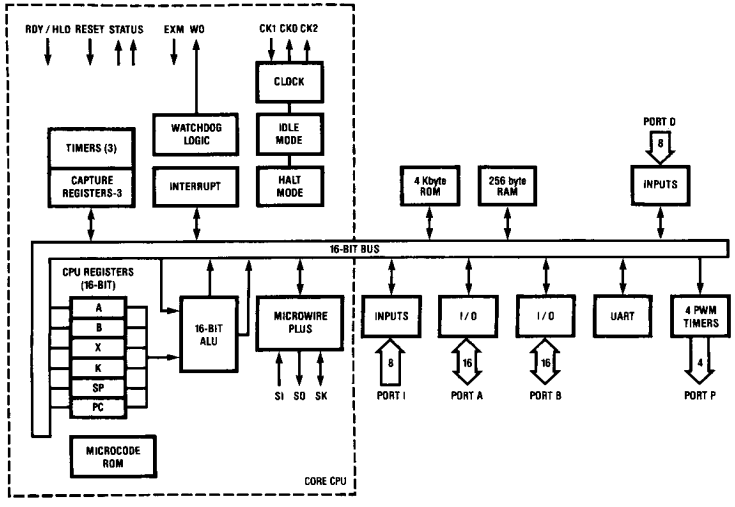
The HPC16040 is a complete microcomputer on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, eight 16-bit timers, vectored interrupts, WATCHDOG™ logic and MICROWIRE/PLUS™ provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC16040 to be used in powerful applications typically performed by microprocessors and expensive peripheral chips.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC16040 is available in 68-pin PCC, LCC and PGA packages.

Features

- HPC family—core features:
  - — 16-bit architecture, both byte and word
  - — 16-bit data bus, ALU, and registers
  - — 64k bytes of external memory addressing
  - — FAST!—240 ns for register instructions when using 17.0 MHz clock
  - — High code efficiency—most instructions are single byte
  - — 16 x 16 multiply and 32 x 16 divide
  - — Eight vectored interrupt sources
  - — Four 16-bit timer/counters with 3 input capture registers and 4 synchronous outputs
  - — WATCHDOG logic monitors processor
  - — MICROWIRE/PLUS serial I/O interface
  - — CMOS—very low power with two power save modes: IDLE and HALT (2 mA, 250 μA—typ.)
- UART—full duplex, programmable baud rate
- Four additional 16-bit timer/counters with pulse width modulated outputs
- 52 general purpose I/O lines (memory mapped)
- 4k bytes of ROM, 256 bytes of RAM on chip
- ROMless versions available
- Wide voltage supply range: 3V to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8340–1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

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 7.0V
All Other Pins	(V_{CC} + 0.5)V to (GND − 0.5)V

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

V_{CC} = 5.0V ±10% unless otherwise specified, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040, −40°C to +105°C for HPC26040, −55°C to +125°C for HPC16040

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
I_{CC_1}	Supply Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz*		20		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz		2.4		mA
I_{CC_2}	IDLE Mode Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz, T_A = 25°C*		2		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz, T_A = 25°C		0.2		mA
I_{CC_3}	HALT Mode Current	V_{CC} = 5.0V, f_{in} = 0 kHz, T_A = 25°C*		250		μA
		V_{CC} = 2.5V, f_{in} = 0 kHz, T_A = 25°C		150		μA
INPUT VOLTAGE LEVELS RESET, NMI, CKI AND WO (SCHMITT TRIGGERED)						
V_{IH_1}	Logic High		0.9 V_{CC}			V
V_{IL_1}	Logic Low				0.1 V_{CC}	V
ALL OTHER INPUTS						
V_{IH_2}	Logic High		0.7 V_{CC}			V
V_{IL_2}	Logic Low				0.2 V_{CC}	V
I_{LI}	Input Leakage Current				±1	μA
C_I	Input Capacitance			10		pF
C_{IO}	I/O Capacitance			20		pF
OUTPUT VOLTAGE LEVELS CMOS OPERATION						
V_{OH_1}	Logic High	I_{OH} = −10 μA	V_{CC} − 0.1			V
V_{OL_1}	Logic Low	I_{OH} = 10 μA			0.1	V
V_{OH_2}	Port A/B Drive, CK2 (A_0–A_{15}, B_{10}, B_{11}, B_{12}, B_{15})	I_{OH} = −7 mA, V_{CC} = 5.0V	2.4			V
V_{OL_2}		I_{OL} = 3 mA			0.4	V
V_{OH_3}	Other Port Pin Drive, WO (open drain) (B_0–B_9, B_{13}, B_{14}, P_0–P_3)	I_{OH} = −1.6 mA, V_{CC} = 5.0V	2.4			V
V_{OL_3}		I_{OL} = 0.5 mA			0.4	V
V_{OH4}	ST1 and ST2 Drive	I_{OH} = −6 mA, V_{CC} = 5.0V	2.4			V
V_{OL_4}		I_{OL} = 1.6 mA			0.4	V
V_{RAM}	RAM Keep-Alive Voltage			2.5		V
I_{OZ}	TRI-STATE Leakage Current			±5		μA

*Note: I_{CC_1}, I_{CC_2}, I_{CC_3} measured with no external drive (I_{OH} and I_{OL} = 0, I_{IH} and I_{IL} = 0).

5

AC Electrical Characteristics

V_{CC} = 5.0V ±10%, f_C = 17.0 MHz, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040

Symbol	Parameter	Min	Typ	Max	Units
f_C = CKI freq.	Operating Frequency	2		17.0	MHz
$t_{C1} = 1/f_C$	Clock Period	59			ns
$t_C = 2/f_C$	Timing Cycle	118	120		ns
$t_{LL} = \frac{1}{2} t_C - 9$	ALE Pulse Width	50	60		ns
$t_{ST} = \frac{1}{4} t_C - 6$	Address Valid to ALE Trailing Edge	23	30		ns
$t_{WAIT} = t_C = WS$	Wait State Period	118	120		ns
$f_{XIN} = \frac{1}{19\, t_{C1}}$	External Timer Input Frequency		877		kHz
$t_{XIN} = 3\, t_{C1}$	Pulse Width for Timer Inputs		180		ns
$f_{XOUT} = \frac{1}{16\, t_{C1}}$	Timer Output Frequency		1.04		MHz
$f_{MW} = \frac{1}{19\, t_{C1}}$	External MICROWIRE/PLUS Clock Input Frequency		877		kHz
$f_U = \frac{1}{19\, t_{C1}}$	External UART Clock Input Frequency		877		kHz
t_{DC1C2}	CK2 Delay from CK1			55	ns

Read Cycle Timing with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARR} = \frac{1}{4} t_C - 5$	ALE Trailing Edge to $\overline{RD}$ Falling Edge	24			ns
$t_{RW} = \frac{1}{2} t_C + WS - 10$	$\overline{RD}$ Pulse Width	167			ns
$t_{DR} = \frac{3}{4} t_C - 15$	Data Hold after Rising Edge of $\overline{RD}$	0		75	ns
$t_{ACC} = t_C + WS - 55$	Address Valid to Input Data Valid			181	ns
$t_{RD} = \frac{1}{2} t_C + WS - 65$	$\overline{RD}$ Falling Edge to Data in Valid			112	ns
$t_{RDA} = t_C - 5$	$\overline{RD}$ Rising Edge to Address Valid	111			ns
$t_{VPR} = \frac{1}{4} t_C - 5$	Address Valid from ALE Trailing Edge Prior to $\overline{RD}$	24	35		ns

Write Cycle Timing with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARW} = \frac{1}{2} t_C - 5$	ALE Trailing Edge to $\overline{WR}$ Falling Edge	54			ns
$t_{WW} = \frac{3}{4} t_c + WS - 15$	$\overline{WR}$ Pulse Width	192			ns
$t_{HW} = \frac{1}{4} t_C - 5$	Data Hold after Trailing Edge of $\overline{WR}$	24			ns
$t_V = \frac{1}{2} t_C + WS - 5$	Data Valid before Trailing Edge of $\overline{WR}$	172			ns
$t_{VPW} = \frac{1}{4} t_C + 20$	Address Valid from Trailing Edge Prior to $\overline{WR}$	50			ns

Note: Bus Output (Port A) C_L = 100 pF, CK2 Output C_L = 50 pF, other Outputs C_L = 80 pF.

Ready/Hold Timing f_C = 16.78 MHz with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{DAR} = 1/4\ t_C + WS - 50$	Falling Edge of ALE to Falling Edge of $\overline{RDY}$		100		ns
$t_{RWP} = t_C$	$\overline{RDY}$ Pulse Width		120		ns
$t_{SALE} = 1/4\ t_C + 40$	Falling Edge of $\overline{HLD}$ to Rising Edge of ALE		70		ns
$t_{HWP} = t_C + 10$	$\overline{HLD}$ Pulse Width		130		ns
t_{HAD}	Rising Edge on $\overline{HLD}$ to Rising Edge on $\overline{HLDA}$		120		ns
$t_{HAE} = t_C + 100$	Falling Edge on $\overline{HLD}$ to Falling Edge on $\overline{HLDA}$		220	*	ns
$t_{BF} = t_C + 30$	Bus Float before Falling Edge on $\overline{HLDA}$		150		ns
$t_{BE} = 2\ t_C + 50$	Bus Enable from Rising Edge of $\overline{HLD}$		290		ns

***Note:** t_{HAE} may be as long as ($3t_C$ + 4ws + $72t_C$ + 90) depending on which instruction is being executed, the addressing mode and number of wait states.

Status Timing f_C = 16.78 MHz

Symbol	Parameter	Min	Typ	Max	Units
$t_{SRS2} = 40 - (1/4\ t_C + 25)$	Setup Time for ST2 on Rising Edge of ALE		−15		ns
$t_{HRS2} = 3/4\ t_C - 15$	Hold Time for ST2 on Rising Edge of ALE		75		ns
$t_{SFS2} = 40 - (1/4\ t_C + 25)$	Setup Time for ST2 on Falling Edge of ALE		−15		ns
$t_{HFS2} = 3/4\ t_C - 15$	Hold Time for ST2 on Falling Edge of ALE		75		ns
t_{SFS1}	Setup Time for ST1 on Falling Edge of $\overline{RD}$		20		ns
$t_{HRS1} = 1/2\ t_C - 15$	Hold Time for ST1 on Rising Edge of $\overline{RD}$		45		ns

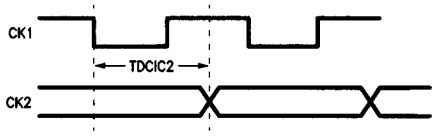
UPI Read/Write Timing

Symbol	Parameter	Min	Typ	Max	Units
t_{UAS}	Address Setup Time to Falling Edge of $\overline{UPIRD}$		5		ns
t_{UAH}	Address Hold Time from Rising Edge of $\overline{UPIRD}$		5		ns
t_{RPW}	$\overline{UPIRD}$ Pulse Width		100		ns
t_{OE}	$\overline{UPIRD}$ Falling Edge to Data Out Valid		60		ns
t_{OD}	End of $\overline{UPIRD}$ to Data Out Valid		35		ns
t_{DRDY}	$\overline{RDRDY}$ Delay from Trailing Edge of $\overline{UPIRD}$		70		ns
t_{WDW}	$\overline{UPIWR}$ Pulse Width		40		ns
t_{UDS}	Data in Valid before Trailing Edge of $\overline{UPIWR}$		10		ns
t_{UDH}	Data in Hold after Trailing Edge of $\overline{UPIWR}$		15		ns
t_A	$\overline{WRRDY}$ Delay from Trailing Edge of $\overline{UPIWR}$		70		ns

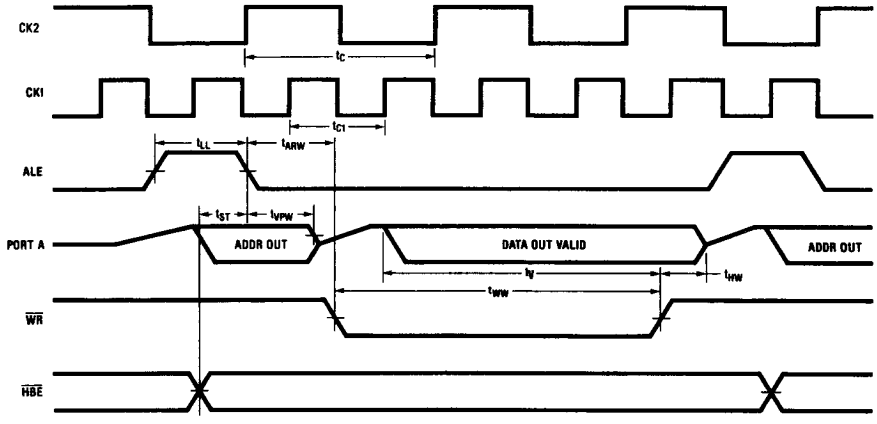
Note: Bus Output (Port A) C_L = 100 pF, CK2 Output C_L = 50 F, other Outputs C_L = 80 pF.

Timing Waveforms

CK2 Delay Timing Diagram

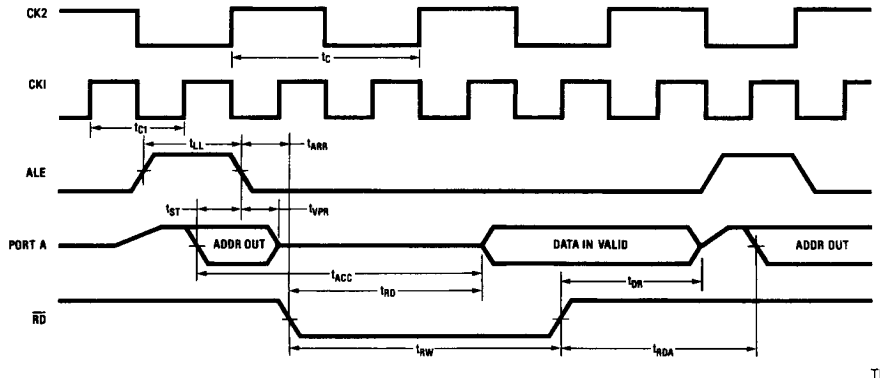


TL/DD/8340-29



TL/DD/8340-2

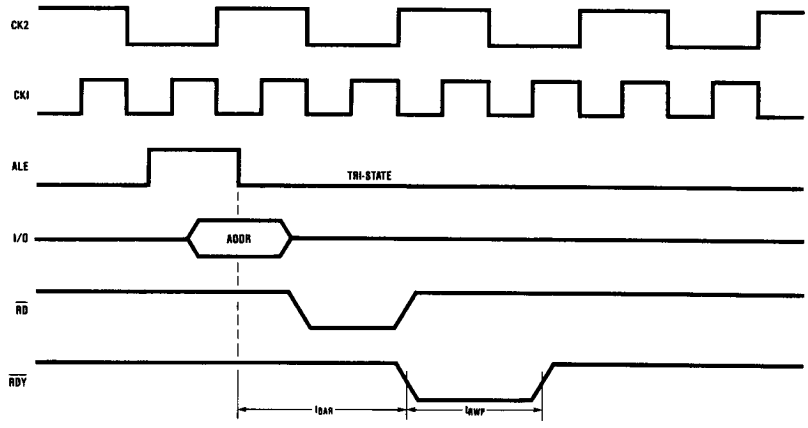
FIGURE 1. Write Cycle



TL/DD/8340-3

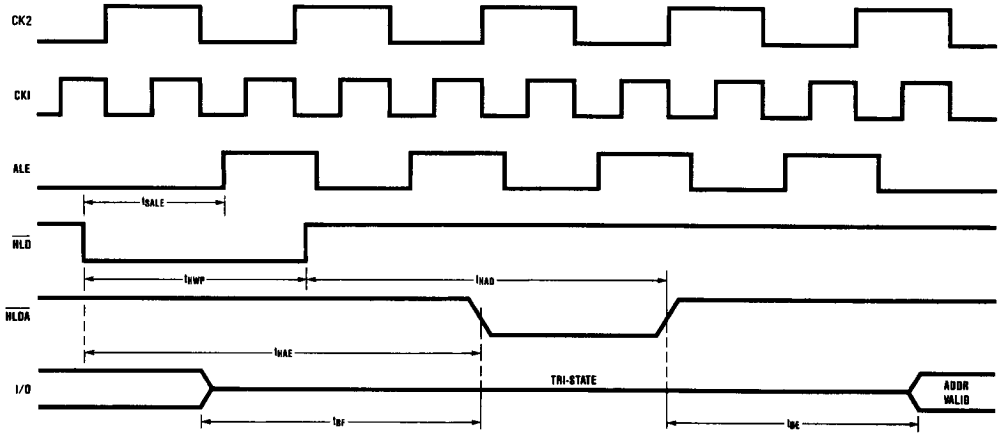
FIGURE 2. Read Cycle

Timing Waveforms (Continued)



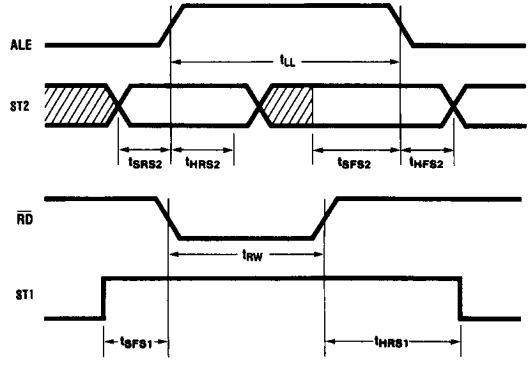
TL/DD/8340-4

FIGURE 3. Ready Mode Timing



TL/DD/8340-5

FIGURE 4. Hold Mode Timing



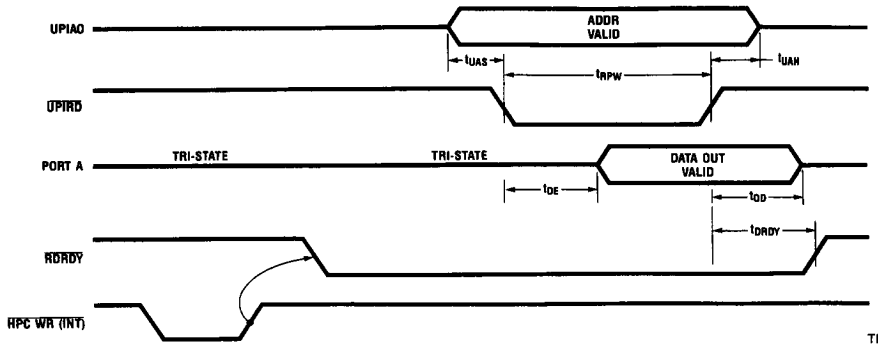
TL/DD/8340-6

TL/DD/8340-7

FIGURE 5. Status Timing

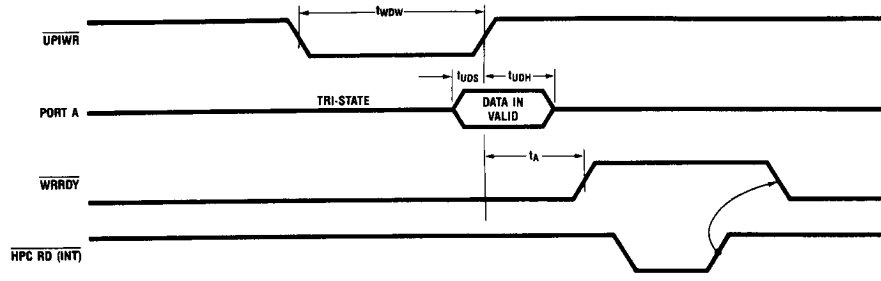
5

Timing Waveforms (Continued)



TL/DD/8340-8

FIGURE 6. UPI Read Timing



TL/DD/8340-9

FIGURE 7. UPI Write Timing

Pin Descriptions

The HPC16040 is available in 68-pin PCC and LCC packages, and a 48-pin ceramic DIP.

I/O PORTS

Port A is a 16-bit bidirectional I/O port with a data direction register to enable each separate pin to be individually defined as an input or output. When accessing external memory, port A is used as the multiplexed address/data bus.

Port B is a 16-bit port with 12 bits of bidirectional I/O similar in structure to Port A. Pins B10, B11, B12 and B15 are general purpose outputs only in this mode. Port B may also be configured via a 16-bit function register BFUN to individually allow each pin to have an alternate function.

B0:	TDX	UART Data Output
B1:		
B2:	CKX	UART Clock (Input or Output)
B3:	T2IO	Timer2 I/O Pin
B4:	T3IO	Timer3 I/O Pin
B5:	SO	MICROWIRE/PLUS Output
B6:	SK	MICROWIRE/PLUS Clock (Input or Output)
B7:	$\overline{\text{HLDA}}$	Hold Acknowledge Output
B8:	TS0	Timer Synchronous Output
B9:	TS1	Timer Synchronous Output
B10:	UA0	Address 0 Input for UPI Mode
B11:	$\overline{\text{WRRDY}}$	Write Ready Output for UPI Mode
B12:		
B13:	TS2	Timer Synchronous Output
B14:	TS3	Timer Synchronous Output
B15:	$\overline{\text{RDRDY}}$	Read Ready Output for UPI Mode

When accessing external memory, four bits of port B are used as follows:

B10:	ALE	Address Latch Enable Output
B11:	$\overline{\text{WR}}$	Write Output
B12:	$\overline{\text{HBE}}$	High Byte Enable Output/Input (sampled at reset)
B15:	$\overline{\text{RD}}$	Read Output

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

I0:		
I1:	NMI	Nonmaskable Interrupt Input
I2:	INT2	Maskable Interrupt/Input Capture/$\overline{\text{URD}}$
I3:	INT3	Maskable Interrupt/Input Capture/$\overline{\text{UWR}}$
I4:	INT4	Maskable Interrupt/Input Capture
I5:	SI	MICROWIRE/PLUS Data Input
I6:	RDX	UART Data Input
I7:		

Port D is an 8-bit input port that can be used as general purpose digital inputs.

Port P is a 4-bit output port that can be used as general purpose data, or selected to be controlled by timers 4

Pin Descriptions (Continued)

through 7 in order to generate frequency, duty cycle and pulse width modulated outputs.

POWER SUPPLY PINS

V_{CC} Positive Power Supply (3V to 5.5V)

GND Ground for On-Chip Logic

DGND Ground for Output Buffers

Note: There are two electrically connected V_{CC} pins on the chip, GND and DGND are electrically isolated. Both V_{CC} pins and both ground pins must be used.

CLOCK PINS

CKI The Chip System Clock Input

CKO The Chip System Clock Output (inversion of CKI)

Pins CKI and CKO are usually connected across an external crystal.

CK2 Clock Output (CKI divided by 2)

OTHER PINS

WO This is an active low open drain output that signals an illegal situation has been detected by the Watch Dog logic.

ST1 Bus Cycle Status Output: indicates first opcode fetch.

ST2 Bus Cycle Status Output: indicates machine states (skip, interrupt and first instruction cycle).

RESET is an active low input that forces the chip to restart and sets the ports in a TRI-STATE® mode.

RDY/HLD has two uses, selected by a software bit. It's either a READY input to extend the bus cycle for slower memories, or a HOLD request input to put the bus in a high impedance state for DMA purposes.

NC (no connection) unused at this time.

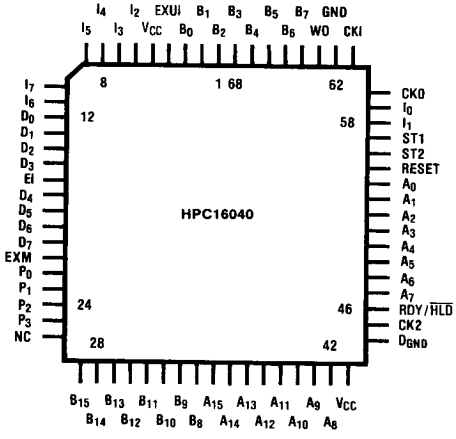
EXM External memory enable (active high) disables internal ROM and maps it to external memory.

EI External interrupt with vector address FFF1:FFF0. (active high)

EXUI External interrupt which is internally OR'ed with the UART interrupt with vector address FFF3:FFF2 (Active Low).

Connection Diagrams

Plastic and Leadless Chip Carriers



TL/DD/8340–10

Top View

Order Number HPC16040E or V
See NS Package Number E68B or V68A

5

Ports A and B

The highly flexible A and B ports are similarly structured. The Port A (see *Figure 9*), consists of a data register and a direction register. Port B (see *Figure 10*) has an alternate function register in addition to the data and direction registers. All the control registers are read/write registers.

The associated direction registers allow the port pins to be individually programmed as inputs or outputs. Port pins selected as inputs, are placed in a TRI-STATE mode by resetting corresponding bits in the direction register.

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.

Primary and secondary functions are multiplexed onto Port B through the alternate function register (BFUN). The secondary functions are enabled by setting the corresponding bits in the BFUN register.

Operating Modes

To offer the user a variety of I/O and expanded memory options, the HPC16040 has four operating modes. The four modes are Single-Chip, Expanded, Single-Chip ROMless, and Expanded ROMless. The four modes are determined by the state of both the External Memory (EXM) pin and the External Access (EA) bit in the PSW Register. The HPC16040 System bus consists of port A and four bits of port B. Port A is defined as the address/data bus and the four bits of port B are referred to as the control bus.

SINGLE-CHIP MODE

In this mode, the HPC16040 functions as a self-contained microcomputer. It can address internal memory consisting of 256 bytes of RAM and 4 kbytes of ROM. All ports are configured as memory mapped I/O ports. The HPC16040 reads 8 bits or 16 bits of data from the ports, depending on whether a byte or word format instruction is used (see *Figure 11*). The EXM pin and EA bit of the PSW Register are both logic "0" during Single-Chip mode signifying that on-chip ROM is being addressed and the range is limited to 4k (see Table II).

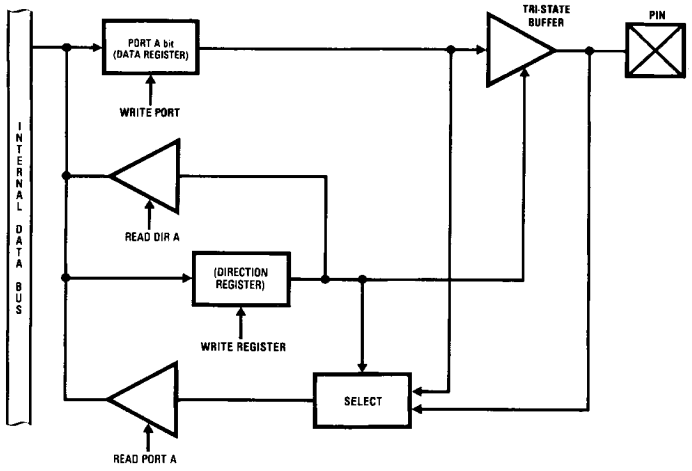
TABLE II. Operating Modes

External Memory Pin (EXM)	External Access Bit (EA)	Operation Mode
0	0	Single Chip
0	1	Expanded
1	0	Single Chip ROMless
1	1	Expanded ROMless

EXPANDED MODE

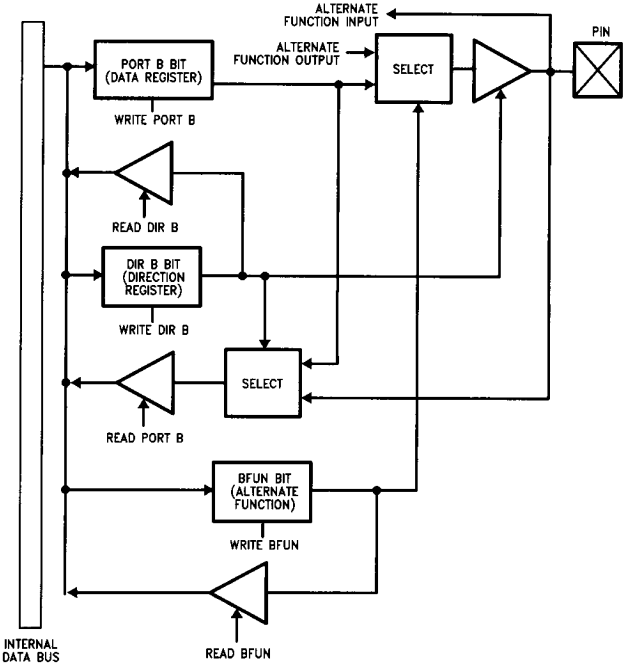
The Expanded mode (see *Figures 12* and *13*) is entered by setting the EA bit in the PSW Register. The HPC16040 can operate within the full 64 kbytes of address space. The 64 kbytes of addressable memory includes all on-chip memory because the EXM pin is grounded during this mode. The external memory may be any combination of RAM and ROM. External memory can be accessed with the data bus defined as either 8 bits wide or 16 bits wide. The System bus may be configured in the 8-bit mode by pulling the $\overline{\text{HBE}}$ pin high at reset. Upon entering the expanded mode, port A

Operating Modes (Continued)



TL/DD/8340-13

FIGURE 9. Port A: I/O Structure



TL/DD/8340-32

FIGURE 10a. Structure of Port B Pins B0, B1, B2, B5, B6 and B7 (Typical Pins)

5

Operating Modes (Continued)

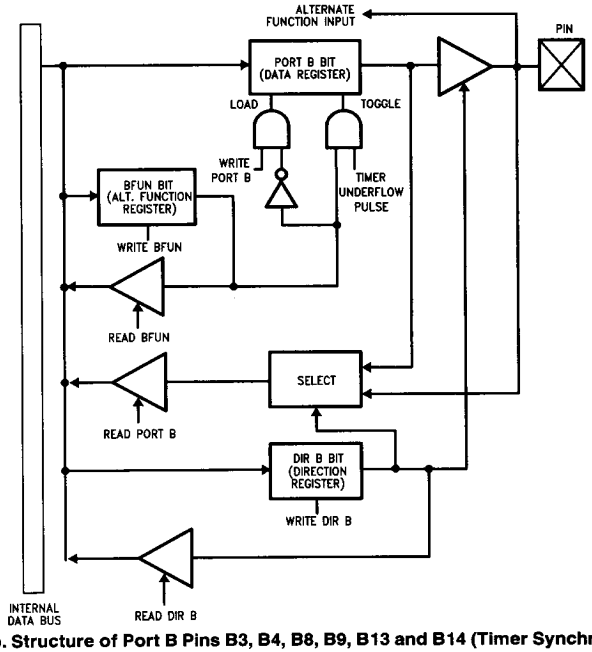


FIGURE 10b. Structure of Port B Pins B3, B4, B8, B9, B13 and B14 (Timer Synchronous Pins)

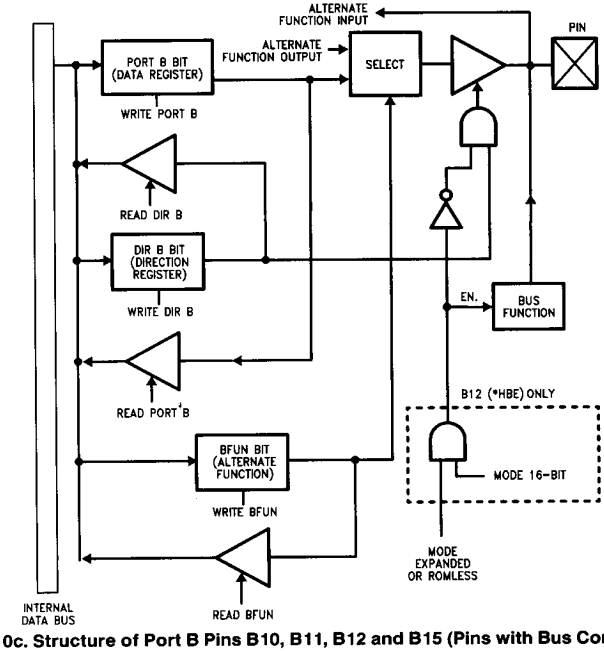


FIGURE 10c. Structure of Port B Pins B10, B11, B12 and B15 (Pins with Bus Control Roles)

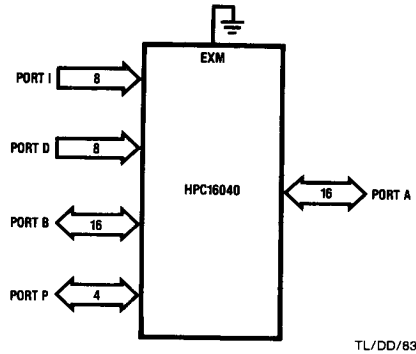
Operating Modes (Continued)

becomes the Address/Data bus. Four bits of port B become ALE, $\overline{WR}$, $\overline{HBE}$ and $\overline{RD}$ signals. The $\overline{RD}$ and $\overline{WR}$ signals are generated only if the selected address is off-chip. The $\overline{HBE}$ is generated only in the 16-bit bus configuration.

ROMless MODES

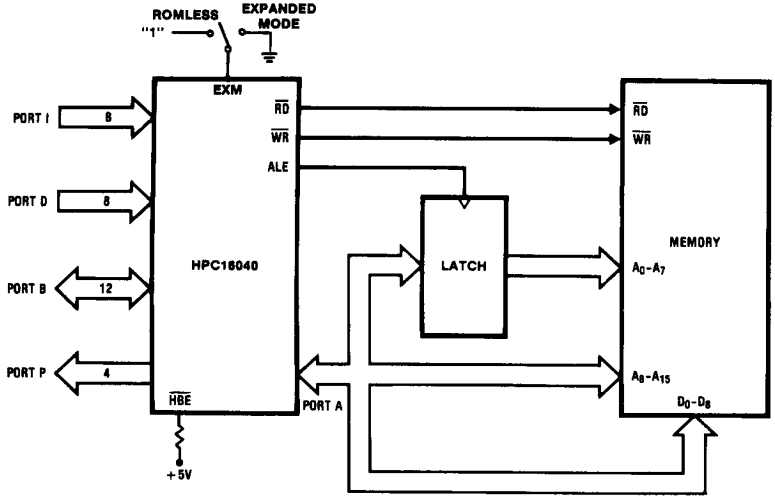
There are two ROMless modes; Single-Chip ROMless and Expanded ROMless. Both ROMless modes are entered by pulling the EXM pin high, (see *Figure 12* and *13*). The EA bit in the PSW Register determines whether the HPC16040 addresses the Single-Chip memory range of 4 kbytes or the Expanded range of 64 kbytes, (see Table II for this information). In both ROMless modes, the HPC16040 continues to use the internal 256 bytes of RAM. The external 4k or 64k of addressed memory may be any combination of RAM and ROM. The address space corresponding to internal ROM is mapped into external memory.

Note: The HPC16040 uses 16-bit words for stack memory. Therefore, when using the 8-bit mode, User's Stack must be in internal RAM.



TL/DD/8340–15

FIGURE 11. Single-Chip Mode

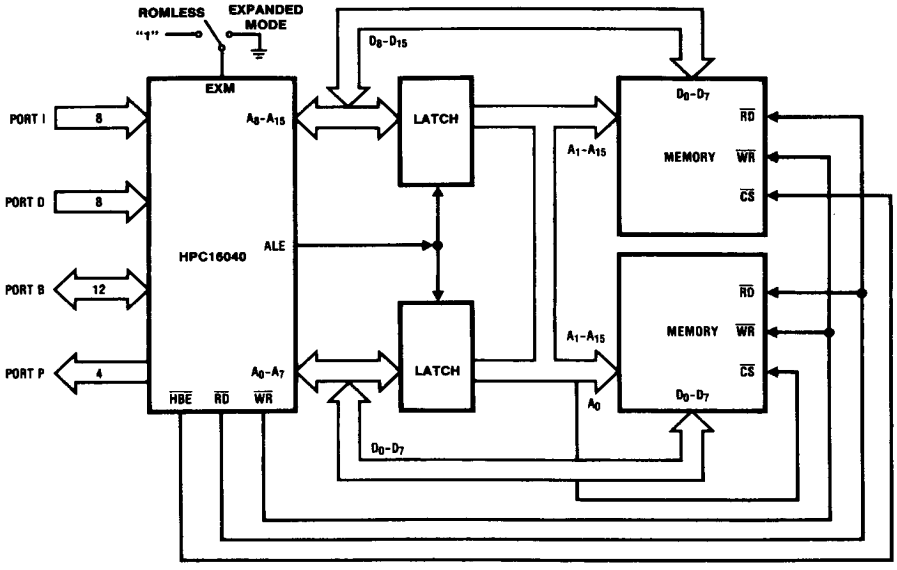


TL/DD/8340–16

FIGURE 12. 8-Bit External Memory

5

Operating Modes (Continued)



TL/DD/8340–17

FIGURE 13. 16-Bit External Memory

Wait States

The HPC16040 provides four software selectable Wait States that allow access to slower memories. The Wait States are selected by the state of two bits in the PSW register. Additionally, the RDY input may be used to extend the instruction cycle, allowing the user to interface with slow memories and peripherals.

Power Save Modes

Two power saving modes are available on the HPC16040: HALT and IDLE. In the HALT mode, all processor activities are stopped. In the IDLE mode, the on-board oscillator and timer T0 are active but all other processor activities are stopped. In either mode, all on-board RAM, registers and I/O are unaffected.

HALT MODE

The HPC16040 is placed in the HALT mode under software control by setting bits in the PSW. All processor activities, including the clock and timers, are stopped. In the HALT mode, power requirements for the HPC16040 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 RESET or the NMI. The RESET input reinitializes 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 HPC16040 is placed in the IDLE mode through the PSW. In this mode, all processor activity, except the on-board oscillator and Timer T0, is stopped. External interrupt (EXUI) is shared with the UART interrupt. This interrupt is level-low sensitive. To select this interrupt disable the ERI and ETI UART interrupt bits in the ENUI register. To select the UART interrupt leave this pin floating or tie it high. As with the HALT mode, the processor is returned to full operation by the RESET or NMI inputs, but without waiting for oscillator stabilization. A timer T0 overflow will also cause the HPC16040 to resume normal operation.

HPC16040 Interrupts

Complex interrupt handling is easily accomplished by the HPC16040's vectored interrupt scheme. There are eight possible interrupt sources as shown in Table III.

TABLE III. Interrupts

Vector Address	Interrupt Source	Arbitration Ranking
$FFFF:FFFE	RESET	0
$FFFD:FFFC	Nonmaskable 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 on the UART transmit/receive complete or external on EXUI	6
$FFF1:FFF0	External interrupt on EI pin	7

Interrupt Arbitration

The HPC16040 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 III. The interrupt on Reset has the highest rank and is serviced first.

Interrupt Processing

Interrupts are serviced after the current instruction is completed except for the RESET, which is serviced immediately.

RESET and EXUI are level-LOW-sensitive interrupts and EI is level-HIGH-sensitive. All other 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 HPC16040 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)

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 be serviced, 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. The occurrence 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 HPC16040 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 affect). 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)

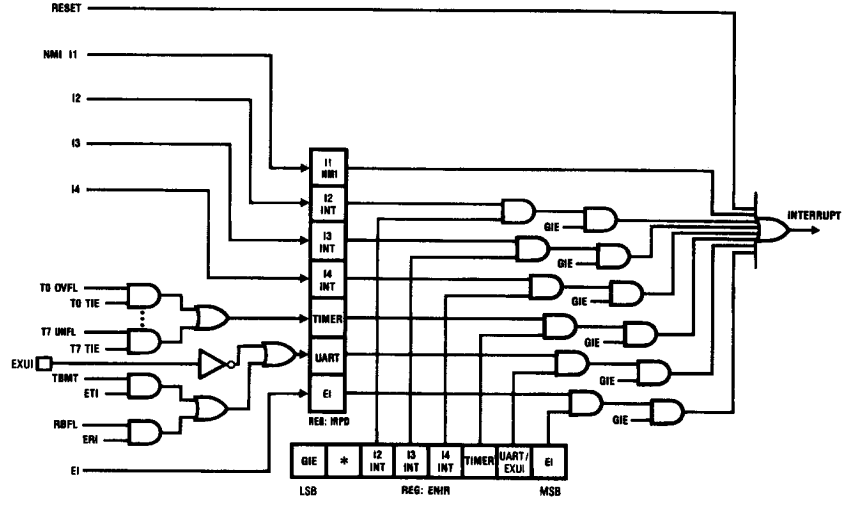
Three bits of the register select the input polarity of the external interrupt on I2, I3, and I4.

Servicing the Interrupts

The Interrupt, once acknowledged, pushes the program counter (PC) onto the stack thus incrementing the stack pointer (SP) twice. 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 14* shows the Interrupt Enable Logic.

Reset

The RESET input initializes the processor and sets ports A, B, and P in the TRI-STATE condition. RESET is an active-low Schmitt trigger input. The processor vectors to FFFF:FFFE and resumes operation at the address contained at that memory location (which must correspond to an on board location).



TL/DD/8340–18

FIGURE 14. Interrupt Enable Logic

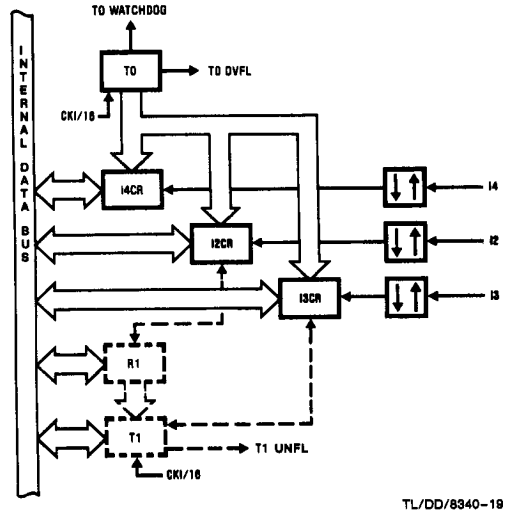
Timer Overview

The HPC16040 contains a powerful set of flexible timers enabling the HPC16040 to perform extensive timer functions; not usually associated with microcontrollers.

The HPC16040 contains eight 16-bit timers. Each timer has an associated 16-bit register. Timer T0 is a free-running timer, counting up at a fixed CKI/16 (Clock Input/16) rate. It is used for Watch Dog 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. A control bit in the register TMMODE configures timer T1 and its associated register R1 as capture registers I3CR and I2CR. 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 (see *Figure 15*).

The timers T2 and T3 have selectable clock rates. 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 control register DIVBY programs the clock input to timers T2 and T3 (see *Figure 16*).

The timers T1 through T7 in conjunction with their registers form Timer-Register pairs. The registers hold the pulse duration values. 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.



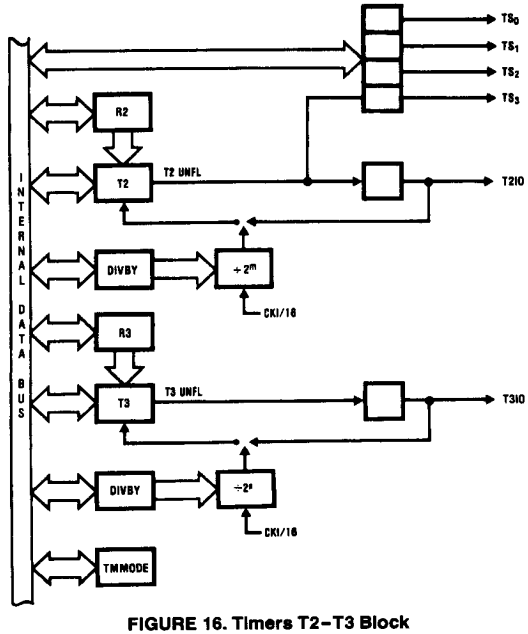
TL/DD/8340-19

FIGURE 15. Timers T0–T1 Block

SYNCHRONOUS OUTPUTS

The flexible timer structure of the HPC16040 simplifies pulse generation and measurement. 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 underflows (see *Figure 16*).

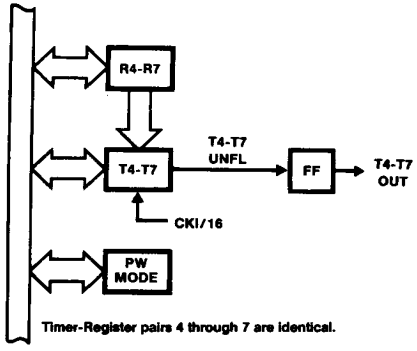
Timer/register pairs 4–7 form four identical units which can generate synchronous outputs on port P (see *Figure 17*).



TL/DD/8340-20

FIGURE 16. Timers T2–T3 Block

Timer Overview (Continued)



TL/DD/8340–21

FIGURE 17. Timers T4–T7 Block

Timer Registers

There are four control registers that program the timers. The divide by (DIVBY) register programs the clock input to timers T2 and T3. The timer mode register (TMMODE) contains control bits to start and stop timers T1 through T3. It also contains bits to latch and enable interrupts from timers T0 through T3. The control register PWMODE similarly programs the pulse width timers T4 through T7 by allowing them to be started, stopped, and to latch and enable interrupts on underflows. The PORTP register contains bits to preset the outputs and enable the synchronous timer output functions.

Timer Applications

The use of Pulse Width Timers for the generation of various waveforms is easily accomplished by the HPC16040.

Frequencies can be generated by using the timer/register pairs. A square wave is generated when the register value is a constant. The duty cycle can be controlled simply by changing the register value.

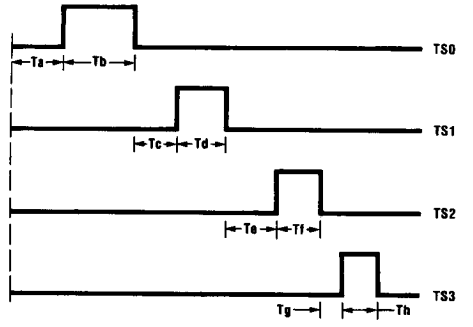
TL/DD/8340–22

FIGURE 18. Square Wave Frequency Generation

Synchronous outputs based on Timer T2 can be generated on the 4 outputs TS0–TS3. Each output can be individually programmed to toggle on T2 underflow. Register R2 contains the time delay between events. *Figure 19* is an example of synchronous pulse train generation.

Watch Dog Logic

The Watch Dog Logic monitors the operations taking place and signals upon the occurrence of any illegal activity. The illegal conditions that trigger the Watch Dog logic are potentially infinite loops and illegal addresses. Should the Watch Dog register not be written to before Timer T0 overflows



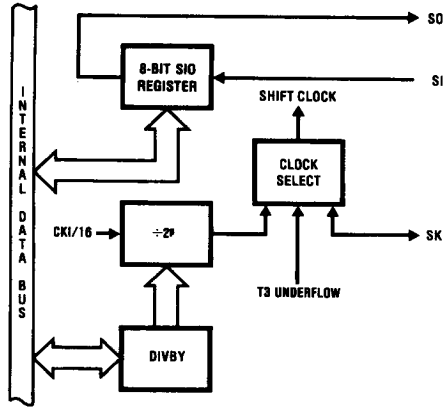
TL/DD/8340–23

FIGURE 19. Synchronous Pulse Generation

twice, or more often than once every 4096 counts, an infinite loop condition is assumed to have occurred. An illegal condition also occurs when the processor generates an off-chip address when in the Single-Chip mode. The illegal condition forces the Watch Out (WO) pin low. The WO pin is an open drain output and can be connected to the RESET or NMI inputs or to the users external logic.

MICROWIRE/PLUS

MICROWIRE/PLUS is used for synchronous serial data communications (see *Figure 20*). MICROWIRE/PLUS 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.



TL/DD/8340–24

FIGURE 20. MICROWIRE/PLUS

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

5

MICROWIRE/PLUS Operation

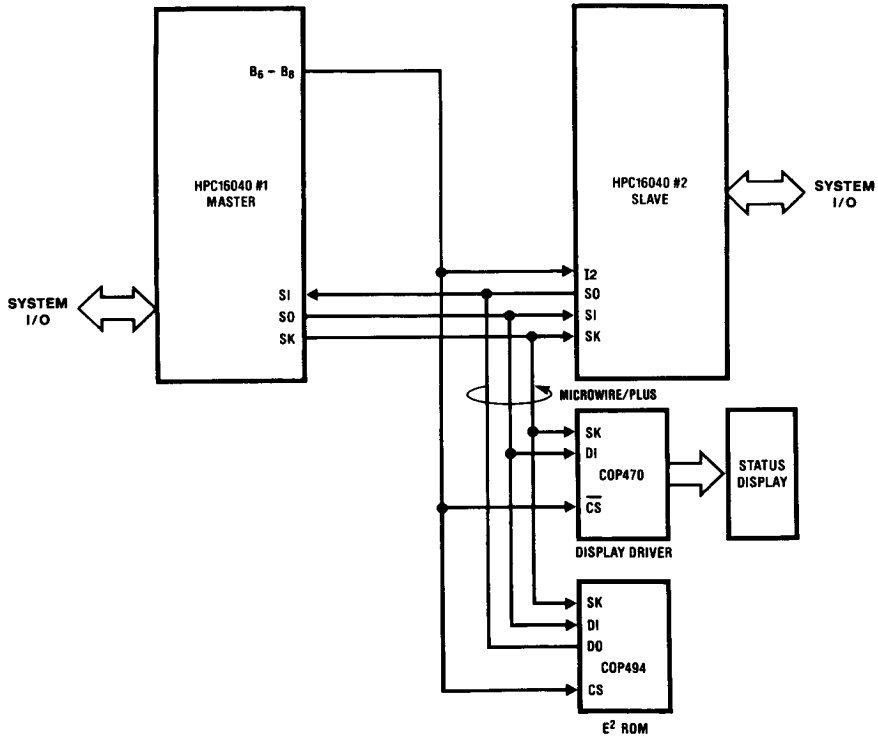
The HPC16040 can enter the MICROWIRE/PLUS mode as the master or a slave. A control bit in the IRCD register determines whether the HPC16040 is the master or slave. The shift clock is generated when the HPC16040 is configured as a master. An externally generated shift clock on the SK pin is used when the HPC16040 is configured as a slave. When the HPC16040 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.

MICROWIRE/PLUS Application

Figure 21 illustrates a MICROWIRE/PLUS arrangement for an automotive application. The microcontroller-based system could be used to interface to an instrument cluster and various parts of the automobile. The diagram shows two HPC16040 microcontrollers interconnected to other MICROWIRE peripherals. HPC16040 #1 is set up as the master and initiates all data transfers. HPC16040 #2 is set up as a slave answering to the master.

The master microcontroller interfaces the operator with the system and could also manage the instrument cluster in an automotive application. Information is visually presented to the operator by means of a VF display controlled by the COP470 display driver. The data to be displayed is sent serially to the COP470 over the MICROWIRE/PLUS link. Data such as accumulated mileage could be stored and retrieved from the EEPROM COP494. The slave HPC16040 could be used as a fuel injection processor and generate timing signals required to operate the fuel valves. The master processor could be used to periodically send updated values to the slave via the MICROWIRE/PLUS link. To speed up the response, chip select logic is implemented by connecting an output from the master to the external interrupt input on the slave.



TL/DD/8340–25

FIGURE 21. MICROWIRE/PLUS Application

HPC16040 UART

The HPC16040 contains a software programmable UART. The UART (see *Figure 22*) 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 (8 or 9 bits) and the value of the ninth bit in transmission. The ENUR register flags framing 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 and enabling or disabling the UART's Wake-up 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 and enabling or disabling transmit and receive interrupts.

The baud rate clock for the Receiver and Transmitter can be selected for either an internal or external source using two bits in the ENUI register. The internal baud rate is programmed by the DIVBY register. The baud rate may be selected from a range of 8 Hz to 64 kHz in binary steps or T3 underflow. By selecting a 9.83 MHz crystal, all standard baud rates from 75 baud to 38.4 kBaud can be generated. The external baud clock source comes from the CKX pin. The Transmitter and Receiver can be run at different rates by selecting one to operate from the internal clock and the other from an external source.

The HPC16040 UART supports two data formats. The first format for data transmission consists of one start bit, eight data bits and one or two stop bits. The second data format for transmission consists of one start bit, nine data bits, and one or two stop bits. Receiving formats differ from transmission only in that the Receiver always requires only one stop bit in a data frame.

UART Wake-up Mode

The HPC16040 UART features a Wake-up Mode of operation. This mode of operation enables the HPC16040 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 HPC16040 with an interrupt. The processor then examines the content of the receiver buffer to decide whether it has been addressed and whether to accpet subsequent data.

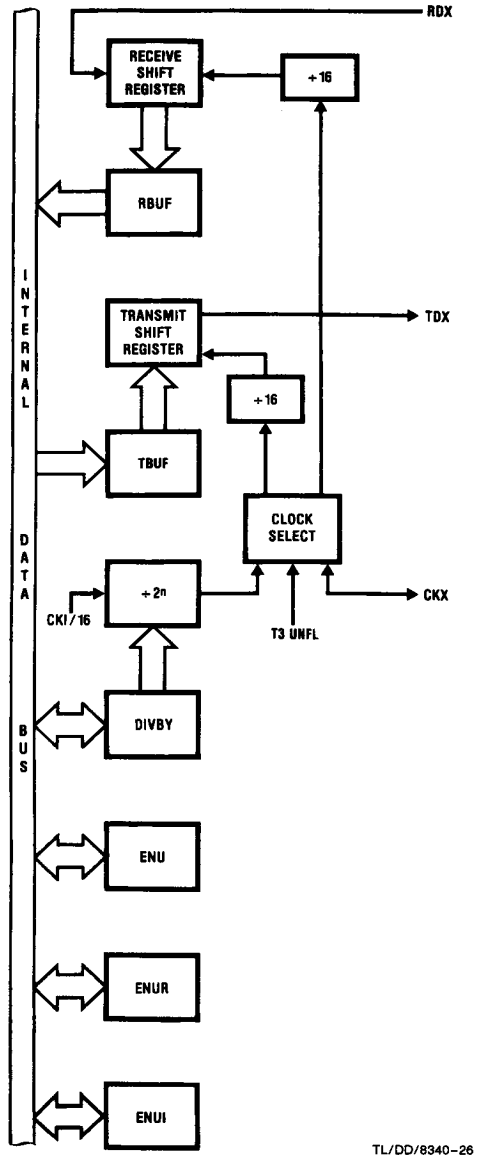


FIGURE 22. UART Block Diagram

5

Universal Peripheral Interface

The Universal Peripheral Interface (UPI) allows the HPC16040 to be used as an intelligent peripheral to another processor. The UPI could thus be used to tightly link two HPC16040's and set up systems with very high data exchange rates. Another area of application could be where a HPC16040 is programmed as an intelligent peripheral to a host system such as the Series 32000 microprocessor. *FIGURE 23* illustrates how a HPC16040 could be used an an intelligent peripherial for a Series 32000-based application.

The interface consists of a Data Bus (port A), a Read Strobe ($\overline{URD}$), a Write Strobe ($\overline{UWR}$), a Read Ready Line ($\overline{RDRDY}$), a Write Ready Line ($\overline{WRRDY}$) and one Address Input (UA0). The data bus can be either eight or sixteen bits wide.

The $\overline{URD}$ and $\overline{UWR}$ inputs may be used to interrupt the HPC16040. The $\overline{RDRDY}$ and $\overline{WRRDY}$ outputs may be used to interrupt the host processor.

The UPI contains an Input Buffer (IBUF), an Output Buffer (OBUF) and a Control Register (UPIC). In the UPI mode, port A on the HPC16040 is the data bus. UPI can only be used if the HPC16040 is in the Single-Chip mode.

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. The HPC16040 supports shared memory access with two pins. The pins are the RDY/$\overline{HLD}$ input pin and the $\overline{HLDA}$ output pin. The user can software select either the Hold or Ready function by the state of a control bit. The $\overline{HLDA}$ output is multiplexed onto port B.

The host uses DMA to interface with the HPC16040. The host initiates a data transfer by activating the $\overline{HLD}$ input of the HPC16040. In response, the HPC16040 places its system bus in a TRI-STATE Mode, freeing it for use by the host. The host waits for the acknowledge signal ($\overline{HLDA}$) from the HPC16040 indicating that the sytem 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 HPC16040 resumes normal operations.

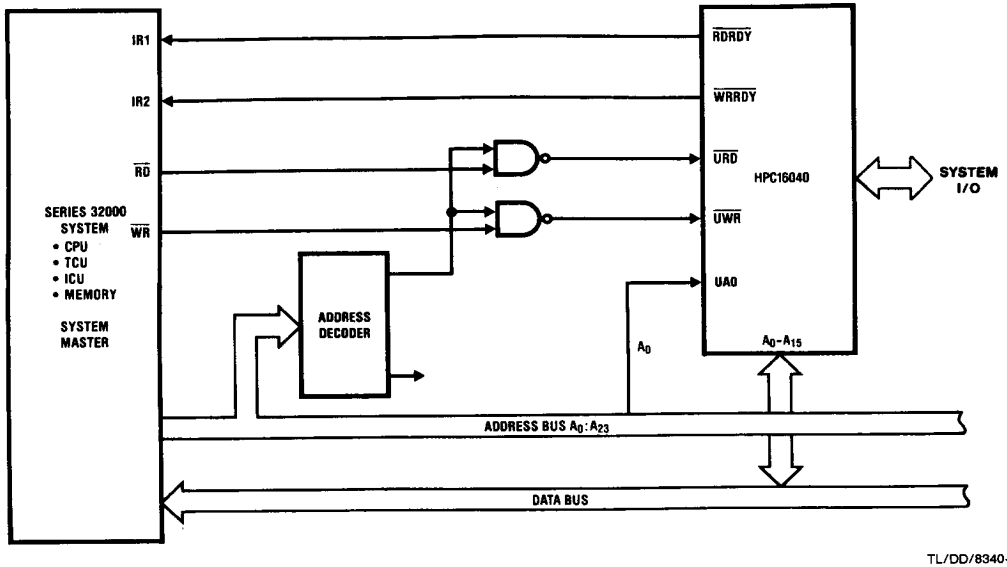
FIGURE 24 illustrates an application of the shared memory interface between the HPC16040 and a Series 32000 system.

Memory

The HPC16040 has been designed to offer flexibility in memory usage. A total address space of 64 kbytes can be addressed with 4096 bytes of ROM and 256 bytes of RAM available on the chip itself. The ROM may contain program instructions, constants or data. The ROM and RAM share the same address space allowing instructions to be executed out of RAM.

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 the B, X and SP registers. Memory can be addressed as words or bytes. Words are always addressed on even-byte boundaries. The HPC16040 uses memory-mapped organization to support registers, I/O and on-chip peripheral functions.

The HPC16040 memory address space extends to 64 kbytes and registers and I/O are mapped as shown in Table IV.



TL/DD/8340–27

FIGURE 23. HPC16040 as a Peripheral: (UPI Interface to Series 32000 Application)

Shared Memory Support (Continued)

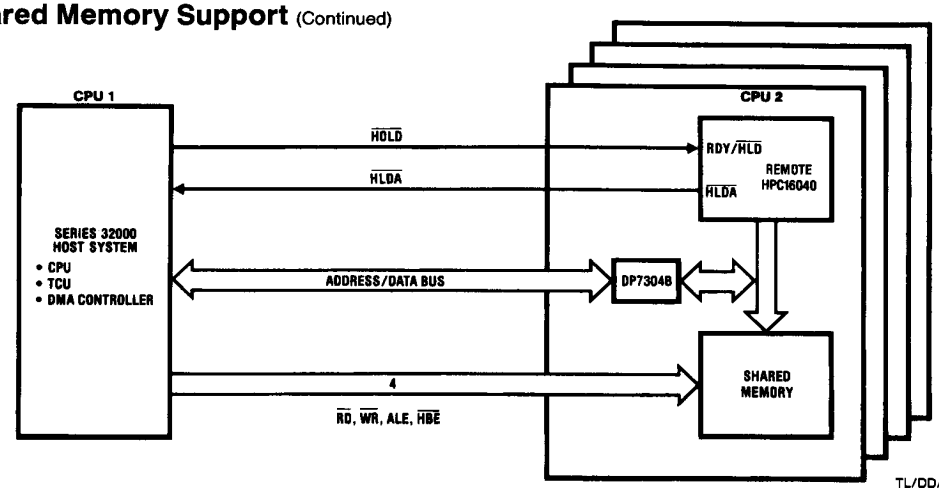


FIGURE 24. Shared Memory Application: HPC16040 Interface to Series 32000 System

TABLE IV. Memory Map

FFFF:FFF0	Interrupt Vectors	USER MEMORY
FFEF:FFD0	JSRP Vectors	
FFCF:FFCE		
: :	On-Chip ROM	
F001:F000		
EFFF:EFFE		
: :	External Expansion	
0201:0200	Memory	
01FF:01FE		
: :	On-Chip RAM	
01C1:01C0		
0195:0194	Watchdog Address	Watchdog Logic
0192	T0CON Register	Timer Block T0:T3
0191:0190	TMMODE Register	
018F:018E	DIVBY Register	
018D:018C	T3 Timer	
018B:018A	R3 Register	
0189:0188	T2 Timer	
0187:0186	R2 Register	
0185:0184	I2CR Register/ R1	
0183:0182	I3CR Register/ T1	
0181:0180	I4CR Register	
0153:0152	Port P Register	Timer Block T4:T7
0151:0150	PWMODE Register	
014F:014E	R7 Register	
014D:014C	T7 Timer	
014B:014A	R6 Register	
0149:0148	T6 Timer	
0147:0146	R5 Register	
0145:0144	T5 Timer	
0143:0142	R4 Register	
0141:0140	T4 Timer	

0128	ENUR Register	UART
0126	TBUF Register	
0124	RBUF Register	
0122	ENUI Register	
0120	ENU Register	
0104	Port D Input Register	
00F5:00F4	BFUN Register	PORTS A & B CONTROL
00F3:00F2	DIR B Register	
00F1:00F0	DIR A Register / IBUF	
00E6	UPIC Register	UPI CONTROL
00E3:00E2	Port B	PORTS A & B
00E1:00E0	Port A / OBUF	
00DE	Microcode ROM Dump	PORT CONTROL & INTERRUPT CONTROL REGISTERS
00DD:00DC	HALT Enable Register	
00D8	Port I Input Register	
00D6	SIO Register	
00D4	IRCD Register	
00D2	IRPD Register	
00D0	ENIR Register	
00CF:00CE	X Register	HPC16040 CORE REGISTERS
00CD:00CC	B Register	
00CB:00CA	K Register	
00C9:00C8	A Register	
00C7:00C6	PC Register	
00C5:00C4	SP Register	
00C3:00C2	(reserved)	
00C0	PSW Register	
00BF:00BE	On-Chip	USER RAM
: :	RAM	
0001:0000		

5

HPC16040 CPU

The HPC16040 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 can also output the carry bit to a 1-bit C register.

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.

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.

Program (PC) Register

The 16-bit PC register addresses program memory.

Addressing Modes

ADDRESSING MODES—ACCUMULATOR AS DESTINATION

Register Indirect

This is the "normal" mode of addressing for the HPC16040 (instructions are single-byte). The operand is the memory addressed by the B register (or X register for some instructions).

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—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.

Double Register Indirect Using the B and X Registers

Used only with Reset, Set and IF 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.

HPC Instruction Set Description

Mnemonic	Description	Action
ARITHMETIC INSTRUCTIONS		
ADD	Add	MA + Meml → MA carry → C
ADC	Add with carry	MA + Meml + C → MA carry → C
DADC	Decimal add with carry	MA + Meml + C → MA (Decimal) carry → C
SUBC	Subtract with carry	MA − Meml + C → MA carry → C
DSUBC	Decimal subtract w/carry	MA − Meml + C → MA (Decimal) carry → C
MULT	Multiply (unsigned)	MA*Meml → MA & X, 0 → K, 0 → C
DIV	Divide (unsigned)	MA/Meml → MA, rem. → X, 0 → K, 0 → C
IFEQ	If equal	Compare MA & Meml, Do next if equal
IFGT	If greater than	Compare MA & Meml, Do next if MA > Meml
AND	Logical and	MA and Meml → MA
OR	Logical or	MA or Meml → MA
XOR	Logical exclusive-or	MA xor Meml → MA
MEMORY MODIFY INSTRUCTIONS		
INC	Increment	Mem + 1 → Mem
DECSZ	Decrement, skip if 0	Mem − 1 → Mem, Skip next if Mem = 0

HPC Instruction Set Description (Continued)

Mnemonic	Description	Action
BIT INSTRUCTIONS		
SET	Set bit	1 → Mem.bit (bit is 0 to 7 immediate)
RESET	Reset bit	0 → Mem.bit
IF	If bit	If Mem.bit is true, do next instr.
MEMORY TRANSFER INSTRUCTIONS		
LD	Load	Meml → MA
	Load, incr/decr X	Mem(X) → A, X ±1 (or 2) → X
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
POP	Pop Stack to Memory	SP−2 → SP, W(SP) → W
LDS	Load A, incr/decr B, Skip on condition	Mem(B) → A, B ±1 (or 2) → B, Skip next if B greater/less than K
XS	Exchange, incr/decr B, Skip on condition	Mem(B) ⟷ A,B ±1 (or 2) → B, Skip next if B greater/less than K
REGISTER LOAD IMMEDIATE INSTRUCTIONS		
LD A	Load A immediate	imm → A
LD B	Load B immediate	imm → B
LD K	Load K immediate	imm → K
LD X	Load X immediate	imm → 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 of 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
SET C	Set C	1 → C
RESET C	Reset C	0 → C
IF C	IF C	Do next if C = 1
IFN C	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
RETS	Return then skip next	SP−2 → SP,W(SP) → PC, & skip
RETI	Return from interrupt	SP−2 → SP,W(SP) → PC, interrupt re-enabled

Note: 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

Memory Usage

Number Of Bytes For Each Instruction (number in parenthesis is 16-Bit field)

	Using Accumulator A						To Direct Memory			
	Reg Indir. (B)	Reg Indir. (X)	Direct	Indir	Index	Immed.	Direct *	Direct **	Immed. *	Immed. **
LD	1	1	2(4)	3	4(5)	2(3)	3(5)	5(6)	3(4)	5(6)
X	1	1	2(4)	3	4(5)	—	—	—	—	—
ST	1	1	2(4)	3	4(5)	—	—	—	—	—
ADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
SBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DSBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
ADD	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
MULT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
DIV	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFEQ	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFGT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
AND	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
OR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
XOR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)

*8-bit direct address
**16-bit direct address

Instructions that modify memory directly

	(B)	(X)	Direct	Indir	Index	B&X
SET	1	2	3(4)	3	4(5)	1
RESET	1	2	3(4)	3	4(5)	1
IF	1	2	3(4)	3	4(5)	1
DECSZ	3	2	2(4)	3	4(5)	
INC	3	2	2(4)	3	4(5)	

Immediate Load Instructions

	Immed.
LD B,*	2(3)
LD X,*	2(3)
LD K,*	2(3)
LD BK,*,*	3(5)

Register Indirect Instructions with Auto Increment and Decrement

Register B With Skip		
	(B+)	(B−)
LDS A,*	1	1
XS A,*	1	1

Register X		
	(X+)	(X−)
LD A,*	1	1
X A,*	1	1

Instructions Using A and C

CLR	A	1
INC	A	1
DEC	A	1
COMP	A	1
SWAP	A	1
RRC	A	1
RLC	A	1
SHR	A	1
SHL	A	1
SET	C	1
RESET	C	1
IF	C	1
IFN	C	1

Transfer of Control Instructions

JSRP	1
JSR	2
JSRL	3
JP	1
JMP	2
JMPL	3
JID	1
JIDW	1
NOP	1
RET	1
RETS	1
RETI	1

Stack Reference Instructions

	Direct
PUSH	2
POP	2

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 HPC16040 has been designed to be extremely code-efficient. The HPC16040 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 HPC16040, and the code savings over other popular microcontrollers has been considerable—often the jobs take less than one-half the memory!

Reasons for this saving of code include the following:

SINGLE BYTE INSTRUCTIONS

The majority of instructions on the HPC16040 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. Most other micros need 2-byte instructions for any short jumps.

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 INSTRUCTIONS FOR DATA MOVEMENT AND PROGRAM LOOPING

The HPC16040 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 the B register
3. Compare the B register versus the K register
4. Generate a conditional skip if B is greater than 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.

The one exception to the above is with the IRPD register. A LOAD IMMEDIATE instruction is to be the only instruction used to clear a bit or bits in this register (see Interrupt Pending Register section).

DECIMAL ADD AND SUBTRACT

This instruction is needed to interface with the decimal user world.

It can handle both 16-bit words and 8-bit bytes.

The 16-bit capability saves code since many variables can be stored as one piece of data and the programmer does not have to break his data into two bytes. Many applications store most data in 4-digit variables. The HPC16040 supplies 8-bit byte capability for 2-digit variables and literal variables.

MULTIPLY AND DIVIDE INSTRUCTIONS

The HPC16040 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.

Development Support

The MOLE (Microcontroller On-Line Emulator) is a low cost development system and emulator for all microcontroller products. These include COPs, TMP, 8050U and the HPC Family of Products. The MOLE consists of a BRAIN Board, Personality Board and optional host software.

The purpose of a MOLE is to provide the user with a tool to write and assemble code, emulate code for the target microcontroller and assist in both the software & hardware debugging of the system.

It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, PROM programming and diagnostic operations.

It contains three serial ports because multiple ports are usually needed to optionally connect to a terminal, a host system, a printer or modem, or to connect to other MOLEs in a multi-MOLE environment.

MOLE can be used in either a stand alone mode or in conjunction with selected host systems, i.e., those using CP/M or PC-DOS. Communicating via RS-232 port.

Dial-A-Helper is a service provided by the MOLE applications group. If a user is having difficulty in getting a MOLE to operate in a particular mode or it is acting peculiar, he can contact us via his system and a modem. He can leave messages on our electronic bulletin board which we will respond to, or he can arrange for us to actually take control of his system via modem for debugging purposes.

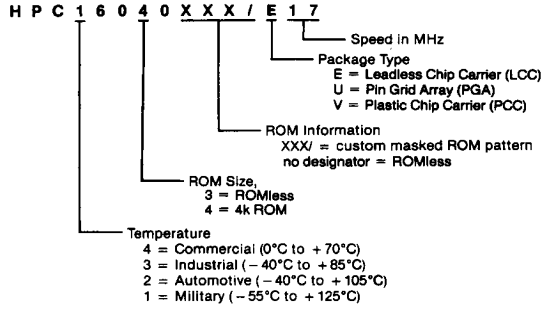
The applications group can then cause his system to execute various commands and try to resolve the customer's problem by actually getting the customer's system to respond. 99% of the time the problem is resolved. This allows us to respond in minutes instead of days when applications help is needed.

The system can also be used to download available applications software.

Part Selection

The HPC family includes devices with many different options and configurations to meet various application needs. The number HPC16040 has been generically used throughout this datasheet to represent the whole family of parts. The following chart explains how to order various options available when ordering HPC family members.

Note: All options may not currently be available.



TL/DD/8340-12

FIGURE 8. HPC Family Part Numbering Scheme

Examples

HPC46030E17 — ROMless, Commercial temp. (0°C to 70°C), LCC

HPC16040XXX/U17 — 4k masked ROM, Military temp. (−55°C to +125°C), PGA

HPC26040XXX/V17 — 4k masked ROM, Automotive temp. (−40°C to +105°C), PCC