8 High-Speed Input/Output Unit

CHAPTER 8
HIGH-SPEED INPUT/OUTPUT UNIT

The 8XC196KC/KD contains four peripherals that together form a flexible, timer/counter-based, high-speed input/output (HSIO) unit. The peripherals that make up the HSIO are Timer 1, Timer 2, the High-Speed Input (HSI) module, and the High-Speed Output (HSO) module (see Figure 8-1). The HSIO can measure pulse widths, generate waveforms, and create periodic interrupts with very little CPU overhead. This chapter describes each module within the HSIO unit.

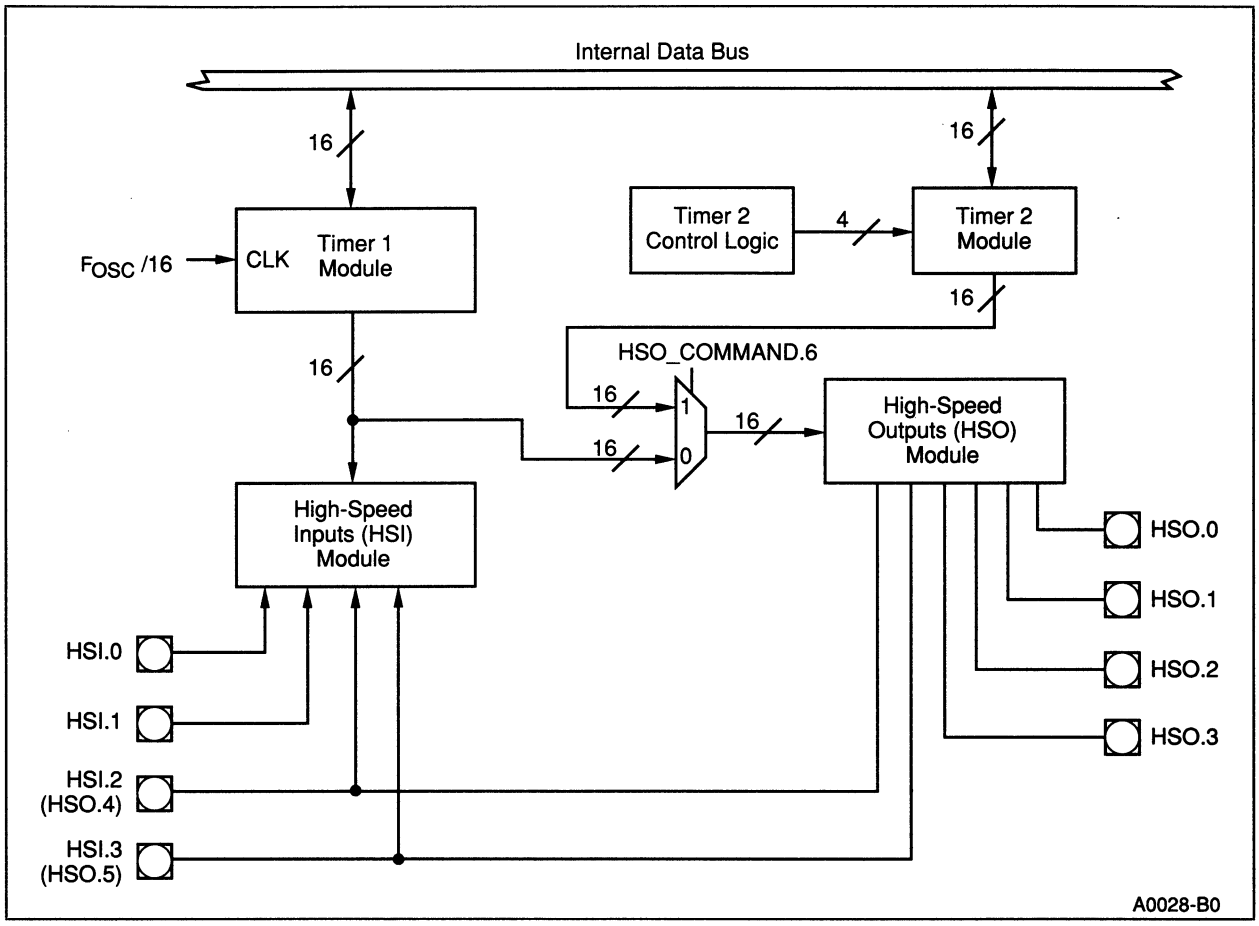

Figure 8-1. HSIO Block Diagram

8.1. TIMERS

The 8XC196KC/KD has two 16-bit timers, Timer 1 and Timer 2. The HSI module always uses Timer 1 as its time base. The HSO module uses either Timer 1 or Timer 2 as its time base.

8.1.1. Timer 1 Functional Overview

Timer 1 is a standard 16-bit, free-running timer that is incremented once every eight state times. Timer 1 is always the time base for the HSI module. It can also be selected as the time base for the HSO module. The two bytes of the TIMER1 register contain the Timer 1 count. You may initialize Timer 1 to a value other than zero by writing to the TIMER1 register in HWindow 15. If you change the TIMER1 value after initiating the HSI module, you may corrupt relative references between HSI events. Also, changing the TIMER1 value after initiating the HSO module with Timer 1 as the time base may cause skipped HSO commands.

8.1.2. Timer 2 Functional Overview

Timer 2 is a programmable 16-bit counter that can be used as the time base for the HSO module, as an up/down counter, or as an extra counter. It can also capture external events. Timer 2 can be clocked either internally or externally. When external clocking is selected, you may select either the T2CLK or HSI.1 pin as the clock source. The maximum count rate for Timer 2 is once every state time (fast increment mode) or every eight state times (normal mode).

The two bytes of the TIMER2 register contain the value of Timer 2. You may initialize Timer 2 to a value other than zero by writing to the TIMER2 register in HWindow 0. When using Timer 2 as the HSO time base, changing the TIMER2 value after initiating the HSO module may cause the HSO to skip commands. See "Timer Precautions" on page 8-9 for additional information.

8.1.3. Programming the Timer 2 Module

Table 8-1 lists the registers that affect the performance and function of the Timer 2 module. Appendix C, "8XC196KC/KD Registers," describes the registers in detail.

Table 8-1. Timer 2 Control and Status Registers

Register Mnemonic	Register Name	Description
TIMER2	Timer 2	Contains the value of Timer 2.
T2CAPTURE	Timer 2 Capture	A rising edge on P2.7 causes the value of Timer 2 to be captured into this register and generates a Timer 2 Capture interrupt (INT11).
INT_MASK INT_MASK1	Interrupt Mask Interrupt Mask 1	Enables or disables the Timer 2 interrupts.
IOC0	Input/Output Control Register 0	Selects the external clock and reset sources for Timer 2.
IOC1	Input/Output Control Register 1	Selects the interrupt source for the Timer Overflow interrupt (INT07).
IOC2	Input/Output Control Register 2	Enables or disables the fast increment mode and the up/down counter function, and selects the overflow boundary for the Timer 2 Overflow interrupt (INT12).
IOC3	Input/Output Control Register 3	Selects the internal or external clock source for Timer 2.

Table 8-2 lists the individual bits and port pins that control Timer 2 options. Column two lists the result of setting the bit or port pin. Column three lists the result of clearing the bit or port pin.

Table 8-2. Timer 2 Control Bits and Pins

Register Bits/ Port Pins	Bit = 1	Bit = 0
IOC0.1	Reset Timer 2 each write	No action
IOC0.3	Enables external reset	Disables external reset
IOC0.5	HSI.0 is the external reset source	T2RST (P2.4) is the external reset source
IOCO.7	HSI.1 is the external clock source	T2CLK (P2.3) is the external clock source
IOC1.3	Enable Timer 2 as source of Timer Overflow interrupt (INT00)	Disables Timer 2 as source of Timer Overflow interrupt (INT00)
IOC2.0	Enables fast increment mode	Disables fast increment mode
IOC2.1	Enables downcount feature	Count up only
IOC2.5	Interrupt on 7FFFH/8000H boundary	Interrupt on 0FFFFH/0000H boundary
T2UP-DN (P2.6)	Count down if IOC2.1 = 1	Count up if IOC2.1 = 1
T2CAPTURE (P2.7)	Captures TIMER2 into T2CAPTURE when a positive transition occurs and the logic level remains stable for at least one state time	—

8.1.3.1. SELECTING THE CLOCK SOURCE

Timer 2 counts both negative and positive transitions. It can be clocked either internally or externally, depending upon the state of the T2_ENA bit in the IOC3 register (IOC3.0). Setting IOC3.0 selects an internal clock source; clearing it selects an external clock source (see Figure 8-2).

When an external clock source is selected, the T2CLK_SRC bit in the IOC0 register (IOC0.7) controls which of two external sources is used. Setting IOC0.7 selects the HSI.1 pin as the external clock source; clearing it selects the T2CLK pin (P2.3) as the external clock source.

When Timer 2 is clocked externally, the FAST_T2_ENA bit in the IOC2 register (IOC2.0) controls the maximum input frequency on the external pin. When FAST_T2_ENA is set, the maximum input rate is one clock per state time (fast increment mode). When FAST_T2_ENA is cleared, the maximum input rate is one clock per eight state times (normal mode).

When Timer 2 is clocked internally, the FAST_T2_ENA bit selects the frequency of the clock input. When FAST_T2_ENA is set, the clock source equals $F_{OSC}/4$ and Timer 2 increments once every state time (fast increment mode). When FAST_T2_ENA is cleared, the clock source equals $F_{OSC}/32$ and Timer 2 increments once every eight state times.

If Timer 2 is the time base for the HSO module, use normal mode. The HSO requires eight state times to sample all CAM entries. HSO events could be missed if Timer 2 uses the fast increment mode. See "Timer Precautions" on page 8-9.

8.1.3.2. SETTING THE COUNT DIRECTION

The T2UD_ENA bit in the IOC2 register (IOC2.1) controls whether Timer 2 counts up only or counts either up or down depending upon the value of the T2UP-DN pin (see Figure 8-2). If IOC2.1 is cleared, Timer 2 always counts up. If IOC2.1 is set and T2UP-DN is low, Timer 2 counts up. If IOC2.1 is set and T2UP-DN is high, Timer 2 counts down.

The T2UP-DN signal must be stable either before or at the same time that the T2CLK signal changes, to ensure that the timer will change direction during the next clock period.

When Timer 2 is the time base for the HSO module, do not change the count direction or events may occur in the wrong order. See "Timer Precautions" on page 8-9 for details.

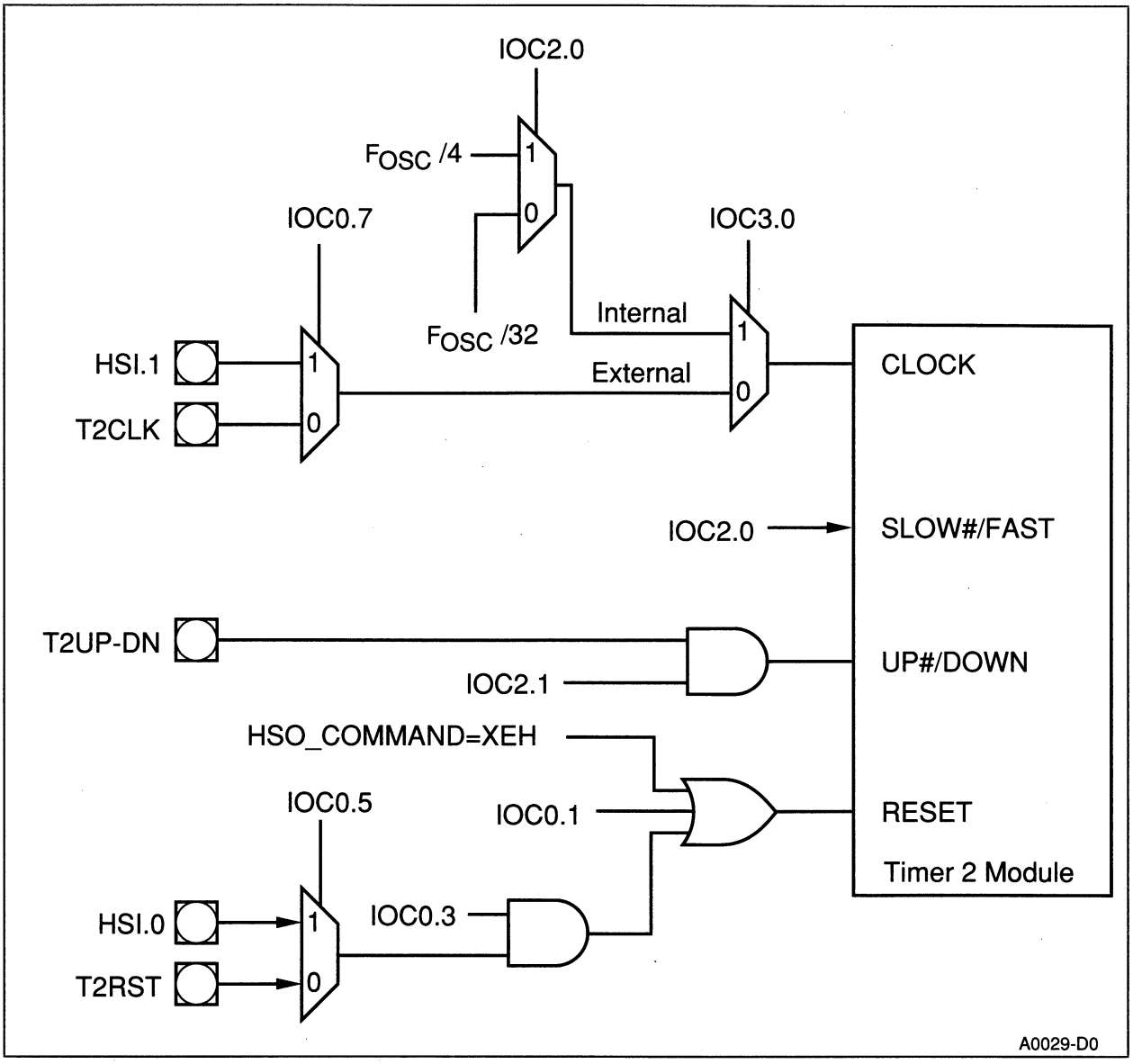

Figure 8-2. Timer 2 Control Logic

8.1.3.3. SELECTING TIMER 2 RESET

Timer 2 can be reset by hardware, software, or the HSO module (see Figure 8-2). Setting the T2RST_ENA bit (IOC0.3) enables an external reset source. Setting the T2RST_SRC bit (IOC0.5) selects the HSI.0 pin and clearing it selects the T2RST (P2.4) pin as the external Timer 2 reset signal. The rising edge of the selected signal resets Timer 2.

Software can reset this timer by setting the SW_T2RST bit (IOC0.1). The HSO can reset this timer by executing the Reset Timer 2 command (CMD_TAG = 0EH). This reset option is particularly useful when using the HSO module to generate pulse width modulated (PWM) outputs. (See "Using the HSO Module to Generate PWM Outputs" on page 8-23.)

8.1.4. Using the External Timer 2 Inputs

The T2UP-DN, T2CLK, T2RST, and T2CAPTURE pins are sampled while CLKOUT is low. These inputs must be stable for at least one full state time and must be valid at least 45 ns before the rising edge of CLKOUT to guarantee recognition.

8.1.4.1. SYNCHRONIZATION

T2RST is always synchronized to an internal, modulo-8 counter. The amount of time it takes to reset the timer depends upon when T2RST is asserted; the timer reset can take from one to nine state times after T2RST is asserted. This is true for both normal and fast increment modes.

During normal increment mode, both T2CLK and T2CAPTURE are also synchronized to the internal, modulo-8 counter. During fast increment mode, these signals pass directly to Timer 2. In all cases, when both signals are asserted simultaneously, the internal design of Timer 2 ensures that the capture will always occur before the clock increments.

8.1.4.2. ASSERTING T2RST, T2CLK, AND T2CAPTURE SIMULTANEOUSLY

Timer 2 behaves predictably when T2RST, T2CLK, and T2CAPTURE occur simultaneously. The behavior of Timer 2 is determined by the time that the signals are asserted in relation to the internal, modulo-8 counter. The internal modulo-8 counter has eight finite *states* numbered one through eight. The counter increments from one state to the next. After it reaches state eight, it rolls over to state one and continues incrementing in an endless loop. The T2CAPTURE and T2CLK signals are sent to Timer 2 during state one; the T2RST signal is sent to Timer 2 during state 2. Since it is **not** possible to synchronize events to this internal counter, you should compensate with either software or hardware to prevent problems.

8.1.4.2.1. Normal Increment Mode

When T2RST, T2CLK, and T2CAPTURE are simultaneously asserted during state 1, the reset occurs first.

Event Sequence 1	Contents of TIMER2	Contents of T2CAPTURE
1. Reset Timer 2	0000H	????H
2. Capture External Event	0000H	0000H
3. Increment Timer 2	0001H	0000H

When the signals are simultaneously asserted during states 2–8, the capture and increment occur before the reset.

Event Sequence 2	Contents of TIMER2	Contents of T2CAPTURE
1. Capture External Event	5A56H	5A56H
2. Increment Timer 2	5A57H	5A56H
3. Reset Timer 2	0000H	5A56H

8.1.4.2.2. Fast Increment Mode

When T2RST, T2CLK, and T2CAPTURE are simultaneously asserted during state 1, 3, 4, 5, 6, 7, or 8, the capture and increment occur before the reset.

Event Sequence 1	Contents of TIMER2	Contents of T2CAPTURE
1. Capture External Event	5A56H	5A56H
2. Increment Timer 2	5A57H	5A56H
3. Reset Timer 2	0000H	5A56H

When they are simultaneously asserted during state 2, the capture occurs, then the reset, and finally the increment.

Event Sequence 2	Contents of TIMER2	Contents of T2CAPTURE
1. Capture External Event	5A56H	5A56H
2. Reset Timer 2	0000H	5A56H
3. Increment Timer 2	0001H	5A56H

8.1.5. Timer Interrupts

Three interrupt vectors are associated with Timer 1 and Timer 2.

- Timer Overflow Interrupt
- Timer 2 Overflow Interrupt
- Timer 2 Capture Interrupt

The Timer Overflow Interrupt was a shared vector for both timers on the 8096BH. The other two are new on the 8XC196KC/KD.

The IOS1 register contains flags that indicate which events triggered interrupts. Reading the IOS1 register clears bits 0–5. For this reason, we recommend that you copy the contents of the IOS1 register into a shadow register and then execute bit-test instructions such as JBC or JBS on the shadow register.

8.1.5.1. TIMER OVERFLOW INTERRUPT

Both Timer 1 and Timer 2 can trigger the Timer Overflow interrupt (INT00). Set INT_MASK.0 to enable this interrupt. Set either IOC1.2 (Timer 1) or IOC1.3 (Timer 2) to select the source of the interrupt. When an overflow occurs, a status flag is set in the IOS1 register. A Timer 1 overflow sets IOS1.5 and a Timer 2 overflow sets IOS1.4.

8.1.5.2. TIMER 2 OVERFLOW INTERRUPT

Timer 2 can generate the Timer 2 Overflow interrupt (INT12) instead of the standard Timer Overflow interrupt. This interrupt is enabled by setting INT_MASK1.4. A Timer 2 overflow sets IOS1.4.

Timer 2 can generate the Timer 2 overflow interrupt at either the 0FFFFH/0000H or 7FFFH/8000H boundary. An overflow can occur in either direction. IOC2.5 selects the overflow boundary. When IOC2.5 is set, Timer 2 interrupts at the 7FFFH/8000H boundary. Otherwise, it interrupts at the 0FFFFH/0000H boundary.

8.1.5.3. TIMER 2 CAPTURE INTERRUPT

A positive transition on the T2CAPTURE pin causes the value of Timer 2 to be loaded into the T2CAPTURE register. This event generates a Timer 2 Capture interrupt (INT11) if INT_MASK1.3 is set and T2CAPTURE is asserted for more than two states times.

8.1.6. Timer Precautions

When using the timers as the HSI or HSO time base, following these guidelines will help you to avoid potential problems. The remainder of this section explains the guidelines and outlines the potential problems.

- Use caution when writing to the timer registers, TIMER1 and TIMER2.
- Configure Timer 2 to operate in normal mode (not in fast increment mode).
- Configure Timer 2 to count in only one direction.
- Use caution when resetting Timer 2.
- When Timer 2 is configured to be reset by an external pin, program events to occur when Timer 2 is equal to one rather than zero.

Changing the TIMER1 value after initiating the HSI module may corrupt relative references between HSI events. Also, changing the reference timer value (TIMER1 or TIMER2) after initiating the HSO module may cause programmed HSO events to be missed or to occur in the wrong order.

Because the HSO requires eight state times for a complete CAM search, Timer 2 must operate in its normal operating mode (not in fast increment mode) when it is used as the HSO reference. Clear the FAST_T2_ENA bit (IOC2.0) to select normal operating mode.

Timer 2 should count in only one direction when it is used as the HSO reference. Using Timer 2 as an up/down counter could cause events to be missed or to occur in the opposite order. Also, if Timer 2 oscillates around an entry's time tag, locked entries could occur several times. Clear the T2UD_ENA bit (IOC2.1) to configure Timer 2 as an up counter.

Do not reset Timer2 before its value reaches the highest programmed Timer 2 time in the CAM. The CAM holds an event pending until a time match occurs. If the Timer 2 value is never reached, the event will remain pending until the device is reset or the CAM is cleared.

When Timer 2 is configured to be reset by an external reset pin (IOC0.3 set), do not program events to occur when Timer 2 is equal to zero. If HSI.0 or T2RST (P2.3) resets Timer 2, the event may not occur. The external pins clear Timer 2 asynchronously, and Timer 2 may be incremented to one before the HSO can compare and act upon the CAM entry. Programming events to occur when Timer 2 is equal to one ensures that the HSO has sufficient time to recognize the CAM entry.

8.2. HIGH-SPEED INPUT MODULE

The High-Speed Input (HSI) module monitors four external input pins (HSI.0–HSI.3). When a predefined event occurs on one or more of the pins, the HSI module records the current Timer 1 count value along with a status bit for each input. The HSI module stores data for up to seven events in a 7 × 20-bit FIFO queue and for one additional event in a holding register. Data moves from the FIFO to the holding register and can be read only after it is loaded into the holding register.

The HSI can be programmed to capture each positive transition, each negative transition, every transition (both positive and negative), or every eighth positive transition. This flexibility enables you to use the HSI module to measure a variety of inputs (i.e., pulse widths, periods, duty cycles, phase differences, etc.). Recording every eighth positive transition allows faster pulse rates to be measured and counted.

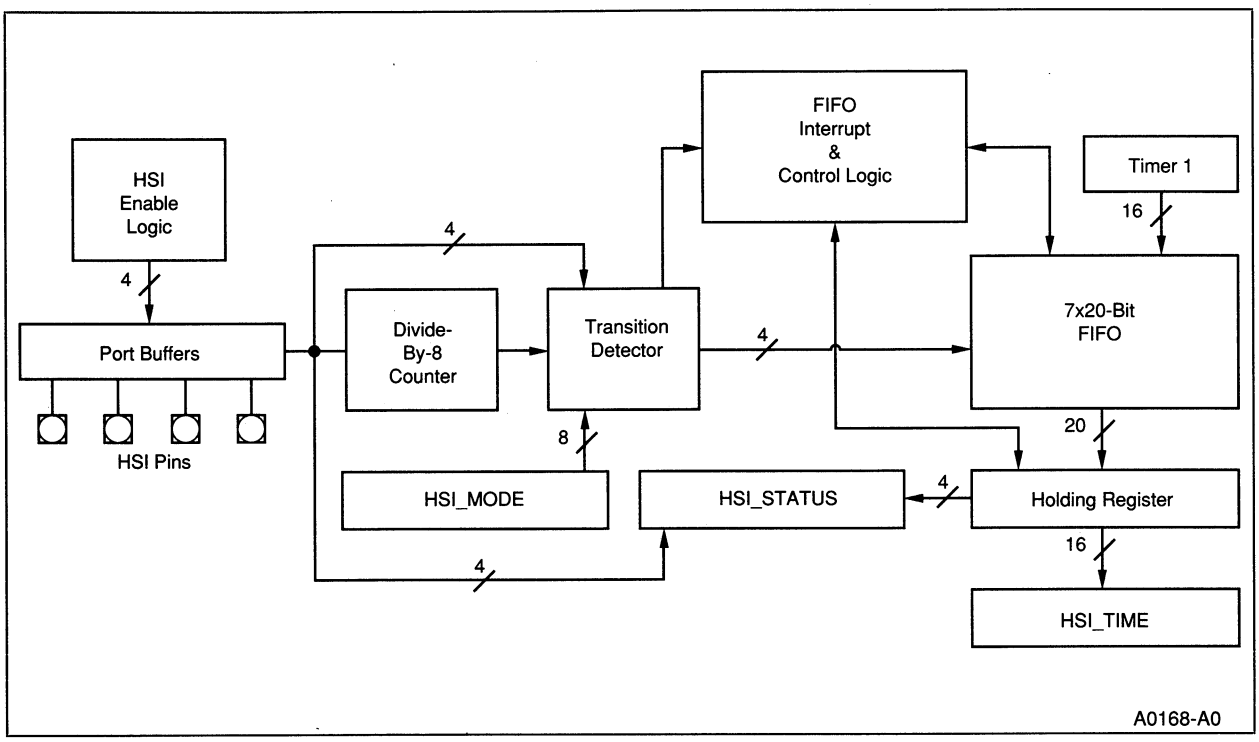

Figure 8-3. HSI Block Diagram

8.2.1. HSI Event Timing Considerations

Events are loaded into the HSI FIFO at the maximum rate of one event every nine state times. Always allow at least nine state times between consecutive events on each pin; otherwise, the HSI may miss an event. If the pin is configured for the eighth transition mode, the maximum rate for positive transitions is eight per 16 state times.

When monitoring repetitive events that occur once per nine state times, a mismatch between the nine-state HSI resolution and the eight-state timer causes one time-tag value to be skipped every nine timer counts. If an event occurs just before the timer increments, and the next event occurs nine state times later, the timer will increment twice between the two events (see Figure 8-4). The result is a FIFO entry with a time tag one count later than expected every nine counts.

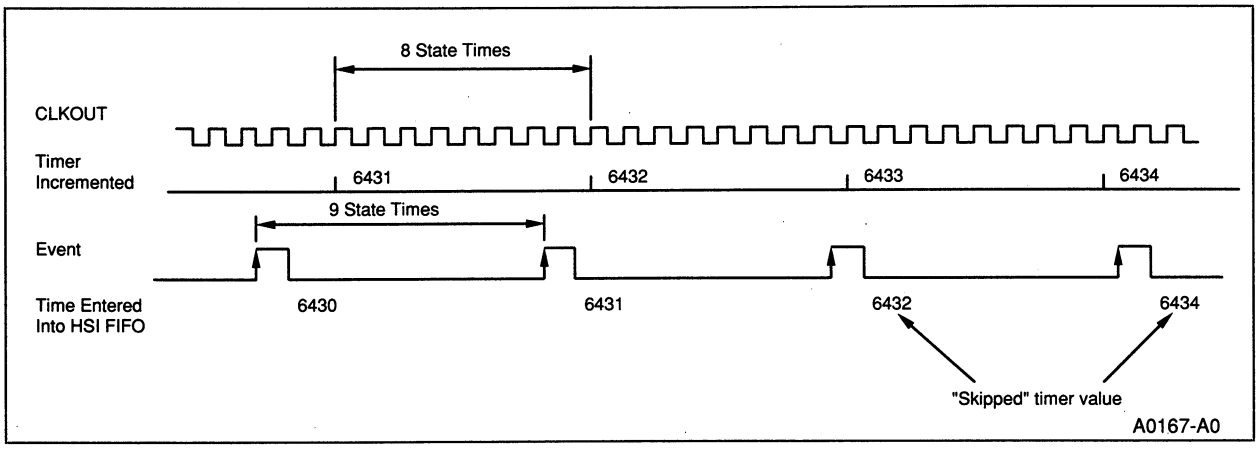

Figure 8-4. Skipped Timer Values

The first event into an empty FIFO will transfer to the holding register after eight state times. If a second event occurs after the first moves to the holding register, it is treated as the first event into an empty FIFO. Additional events are not recorded if both the FIFO and holding register are full.

If the first two events into an empty FIFO (not including the holding register) occur during the same internal phase, both are recorded with the same time tag. Otherwise, if the second event occurs within nine states after the first, its time tag is one count later than the tag time of the first. If this is the *skipped* timer value, the second event's time tag is two counts later than the time tag of the first.

The change detection logic samples the HSI pins while CLKOUT is low. To guarantee that the HSI event is recorded with the correct Timer 1 value, the HSI inputs must either be stable for at least one full state time or be valid at least 45 ns before the rising edge of CLKOUT. If neither condition is met, the HSI input may be sampled in the next CLKOUT. Therefore, if information is to be synchronized to the HSI module, it should be latched on the rising edge of CLKOUT.

The HSI module's internal divide-by-eight counter continuously counts transitions. When you program an HSI input to detect every eighth positive transition, the counter is not reset. It retains its accumulated count and continues counting. If the accumulated count was six, then just two additional positive transitions will trigger an HSI event capture. For this reason, it is a good idea to ignore the first HSI event when using the eight transitions mode.

8.2.2. Reading the Event Data

The event data can be read only after it is loaded into the holding register. Allow eight state times for data to move through the FIFO and into the holding register. When an event moves from the FIFO into the holding register, the HSI_RDY bit is set in the IOS1 register (IOS1.7). The HSI Data Available interrupt (INT02) is generated if the interrupt is enabled and the event is selected as the source of the interrupt. (See "Enabling HSI Interrupts" on page 8-14 for details).

To unload the holding register, first read the HSI_STATUS register. This causes the holding register to load the event status bits for HSI.0-HSI.3 into the even bits of the HSI_STATUS register. When set, the event status bits indicate that an event occurred on the corresponding HSI pin. The odd bits always contain the current state of the HSI.0–HSI.3 pins.

Then read the HSI_TIME register. The holding register loads the Timer 1 count value into the HSI_TIME register. Reading the HSI_TIME register causes the FIFO to load the next event into the holding register. If you read the HSI_TIME register before first reading the HSI_STATUS register, the event status bits are overwritten.

8.2.3. Programming the HSI Module

Table 8-7 lists the registers that affect the performance and function of the HSI module. Refer to Appendix C, "8XC196KC/KD Registers," for detailed information.

Table 8-3. HSI Control and Status Registers

Register Mnemonic	Register Name	Description
HSI_MODE	HSI Mode	Defines the type of transition that will cause an event on each HSI pin.
HSI_STATUS	HSI Status	Contains the event-detection bits and the current input level for each HSI pin.
HSI_TIME	HSI Time	Contains the 16-bit value of TIMER1 at the time of the event.
INT_MASK INT_MASK1	Interrupt Mask Interrupt Mask 1	Enables or disables the HSI interrupts.
IOC0	Input/Output Control Register 0	Enables or disables each HSI pin. Selects alternate functions for HSI.0 and HSI.1.
IOC1	Input/Output Control Register 1	Selects the source for the HSI Data Available interrupt (INT02).
IOS1	Input/Output Status Register 1	Indicates the status of the HSI FIFO.

8.2.3.1. DEFINING AN HSI EVENT

The HSI_MODE register controls, for each HSI pin, which of four types of events trigger a capture into the HSI FIFO: each positive transition, each negative transition, every transition (both positive and negative), or a series of eight positive transitions (see Figure 8-5).

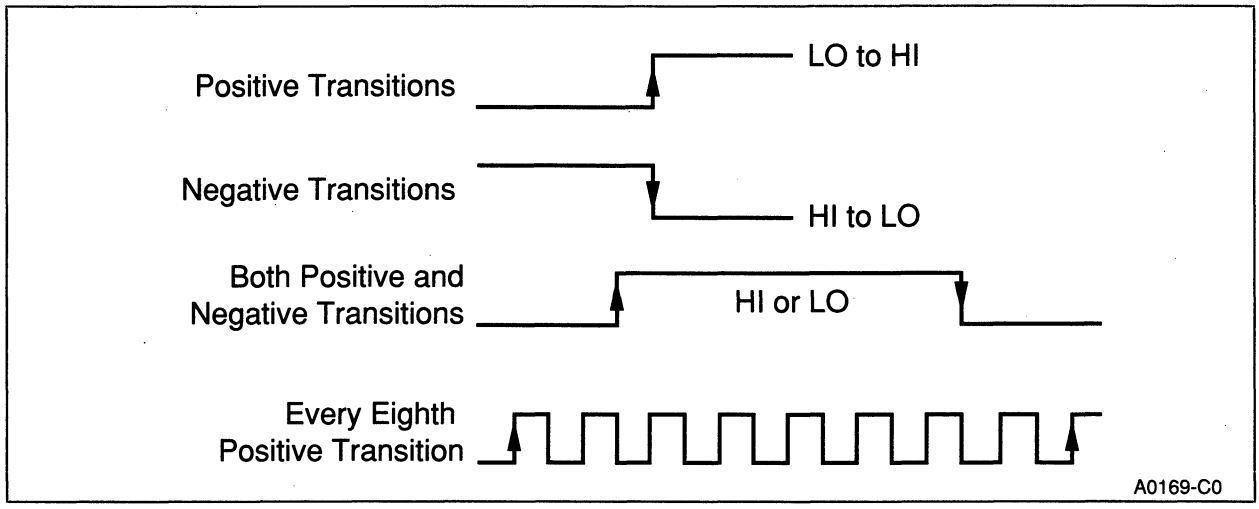

Figure 8-5. HSI Event Options

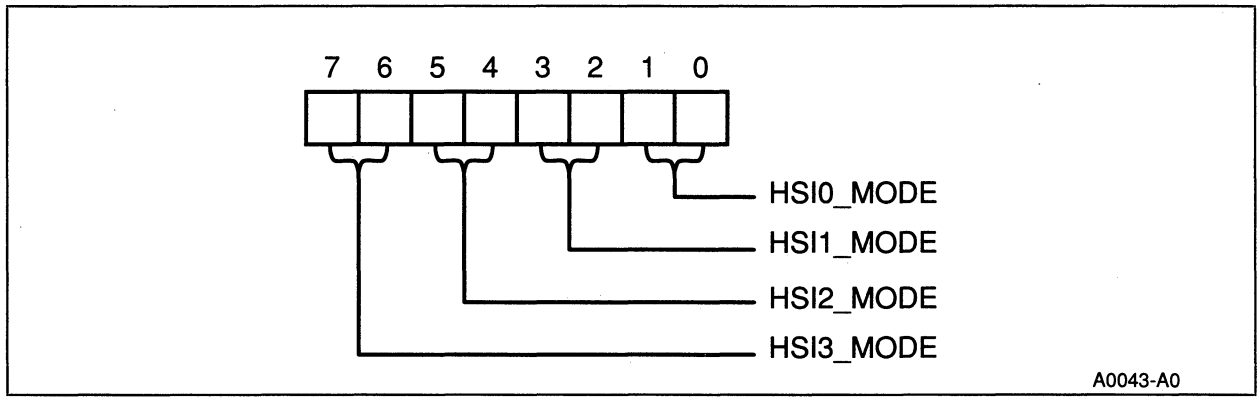

Figure 8-6. HSI_MODE Register

Each two-bit field within the HSI_MODE register is programmed to define the transition mode for the corresponding HSI input (see Figure 8-6 and Table 8-4).

Table 8-4. Transition Mode Encoding

HSI*x*_MODE	Description
00	Eight positive transitions trigger a capture into the HSI FIFO.
01	Each positive transition triggers a capture into the HSI FIFO.
10	Each negative transition triggers a capture into the HSI FIFO.
11	Every transition (both positive and negative) triggers a capture into the HSI FIFO.

8.2.3.2. ENABLING HSI INTERRUPTS

The HSI can generate an interrupt when any of the following events occur.

- The fourth entry is loaded into the FIFO.
- The sixth entry is loaded into the FIFO.
- An entry moves from the FIFO into the holding register.

In addition, the HSI.0 pin can be configured to function as an independent, external interrupt. It need not be enabled as an HSI input to function as an external interrupt.

The INT_MASK and INT_MASK1 registers contain bits that enable and disable each interrupt. Table 8-5 briefly describes the conditions that cause each HSI interrupt and then lists its interrupt number and priority.

Table 8-5. HSI Interrupts

Interrupt Source	Generated when	Interrupt Name	Priority (Note 1)
HSI FIFO Full (Note 2)	INT_MASK1.6 = 1 and the sixth entry is loaded into the FIFO. Holding register entries are not counted.	HSI FIFO Full (INT14)	14
	IOC1.7 = 1, INT_MASK.2 = 1, INT_MASK1.6 = 0, and the sixth entry is loaded into the FIFO. Holding register entries are not counted.	HSI Data Available (INT02)	2
HSI FIFO 4	INT_MASK1.2 = 1, and the fourth entry is loaded into the FIFO. Holding register entries are not counted.	HSI FIFO 4 (INT10)	10
HSI.0 External	INT_MASK1.4 = 1 and the HSI.0 signal transitions from low to high and remains high for at least one state time. This interrupt can be configured to function as an independent, external interrupt, the HSI.0 need not be enabled as an input to the HSI module.	HSI.0 Pin (INT04)	4
HSI Data Available	IOC1.7 = 0, INT_MASK.2 = 1, and an entry moves from the FIFO into the holding register. This interrupt indicates that at least one HSI event has occurred and is ready for processing.	HSI Data Available (INT02)	2

NOTES:

1. 15 is the highest priority; 0 is the lowest.
2. The sixth FIFO entry can trigger either the HSI Data Available interrupt (INT02) or the HSI FIFO Full interrupt (INT14) but should not be configured for both.

8.2.3.3. ENABLING AND DISABLING THE HSI PINS

You can individually enable or disable the HSI input function of each HSI pin by setting or clearing the corresponding bit in the IOC0 register (see Figure 8-7 and Appendix C, "8XC196KC/KD Registers," for a complete description of the IOC0 register).

Setting the appropriate bit in the IOC0 register connects the pin to the HSI transition detector and divide-by-eight counter. Clearing the bit disconnects the pin from the HSI logic, but not from the HSI_STATUS register (see Figure 8-7). Disabling the HSI input function does not disable or prevent the use of each pin's additional functions.

8.2.3.4. ASSIGNING ADDITIONAL FUNCTIONS TO HSI PINS

You can optionally program the HSI pins to provide alternate functions. Table 8-6 lists the additional functions for each HSI output pin. Column three describes how to select the additional function.

Table 8-6. Additional Functions for HSI Pins

HSI Function Name	Additional Functions	Selected by
HSI.0	HSI.0 Pin interrupt (INT04)	Setting INT_MASK1.4 enables the HSI.0 Pin interrupt (INT04). Assert HSI.0 for greater than two state times to guarantee recognition.
	Timer 2 reset source	Setting IOC0.5 selects HSI.0 as the Timer 2 reset source. When enabled, a rising edge on HSI.0 resets Timer 2.
HSI.1	Timer 2 clock source	Clearing IOC0.7 and IOC3.0 selects the HSI.1 pin as the Timer 2 clock source.
HSI.2	HSO.4	Setting IOC1.4 enables the output function of HSO.4. Note that the HSI and HSO functions can be active at the same time.
HSI.3	HSO.5	Setting IOC1.6 enables the output function of HSO.5.Note that the HSI and HSO functions can be active at the same time.

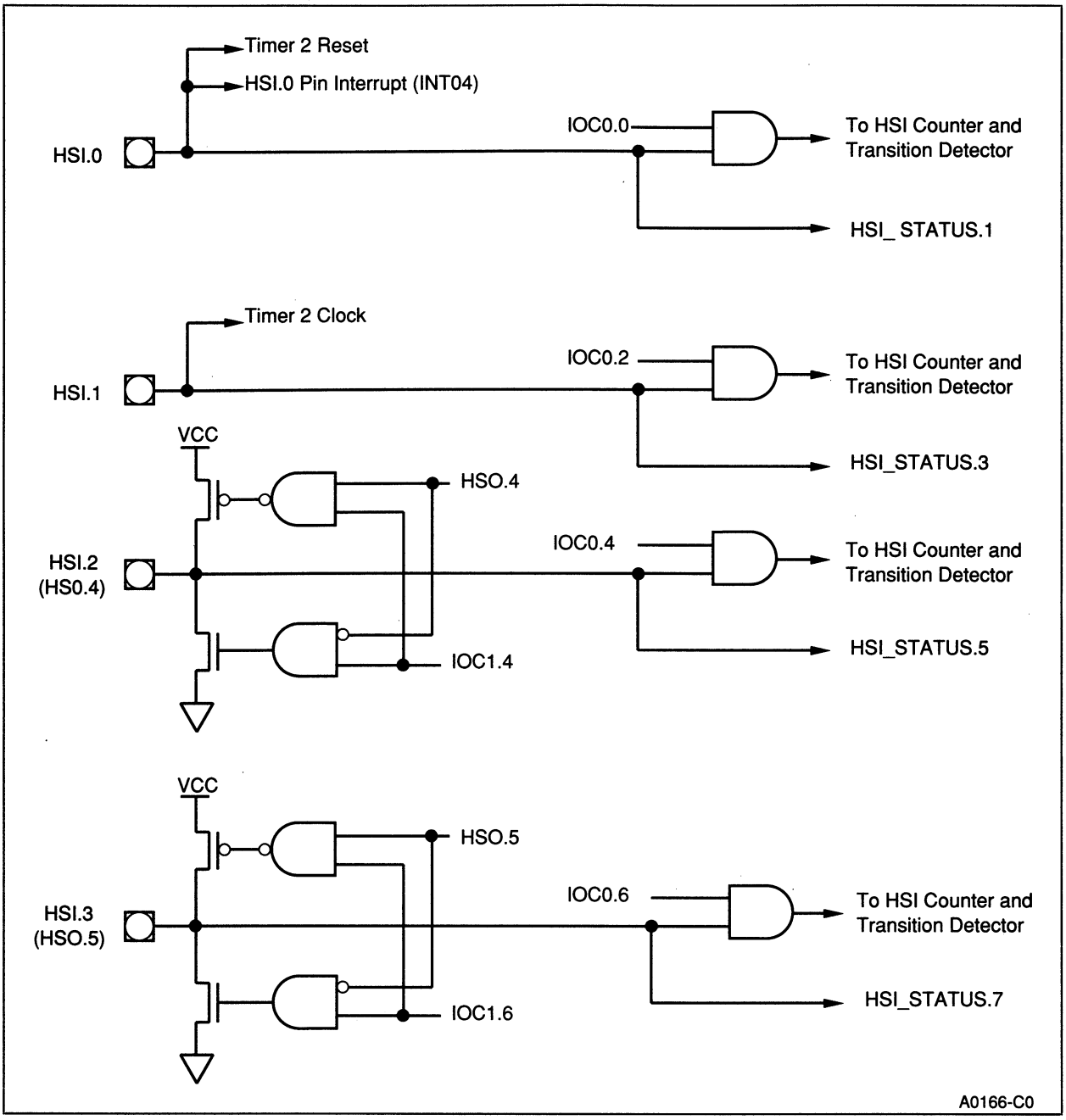

Figure 8-7. HSI Input Options

8.3. HIGH-SPEED OUTPUT MODULE

The High-Speed Output (HSO) module triggers events at specified times using either Timer 1 or Timer 2 as the time reference. These programmable events include starting an A/D conversion, resetting Timer 2, generating up to four software time delays, and setting or clearing one or more of the HSO outputs (HSO.0–HSO.5). The HSO module stores up to eight pending events and their specified times in a Content-Addressable Memory (CAM) file. Figure 8-8 is a block diagram of the HSO module.

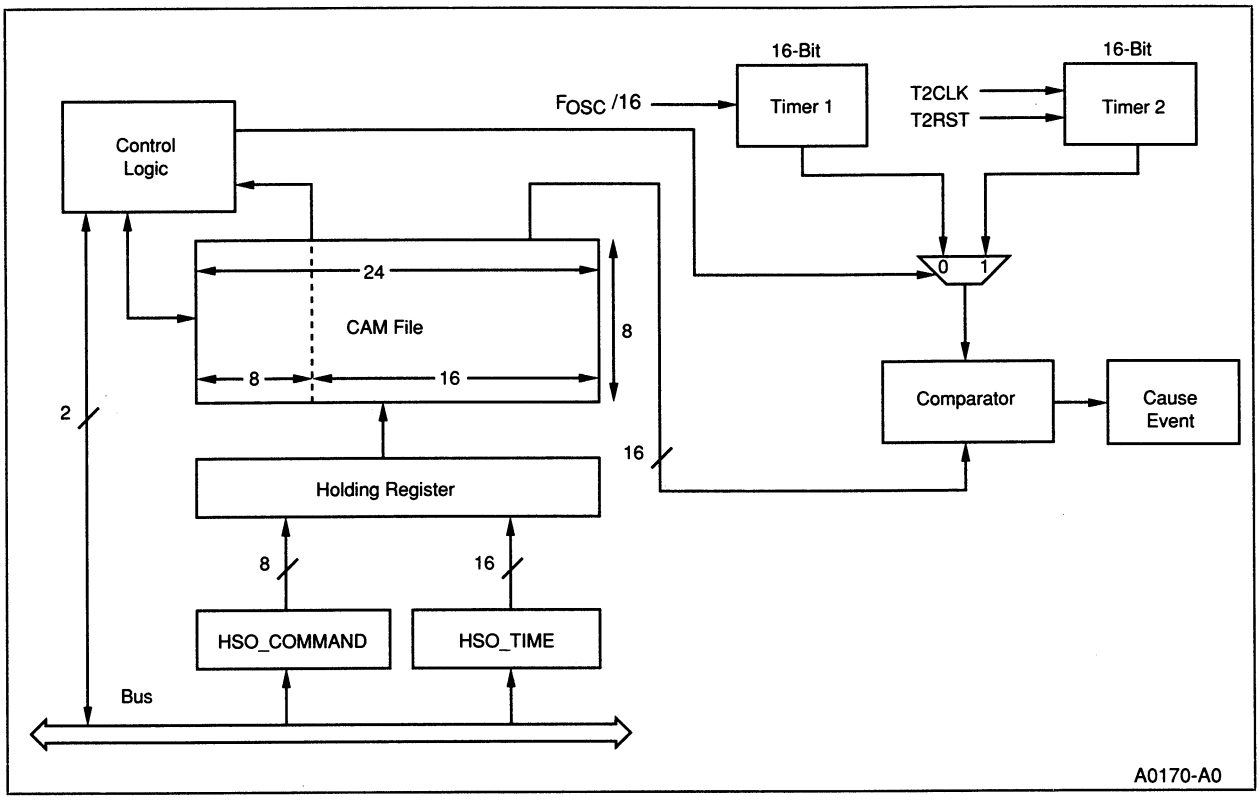

Figure 8-8. HSO Block Diagram

8.3.1. HSO Functional Overview

The CAM file is the main component of the HSO module. This file stores up to eight commands. Each CAM entry is 24 bits wide. Sixteen bits are loaded from the HSO_TIME register to specify the action time, and eight bits are loaded from the HSO_COMMAND register to specify the nature of the action, whether it will generate an interrupt, and whether Timer 1 or Timer 2 is the reference.

NOTE

When Timer 2 is used as the HSO module's reference timer, it must operate in normal mode (not fast increment mode) and should count in only one direction. See "Timer Precautions" on page 8-9 for additional information.

The HSO compares all eight CAM entries with the timer value before triggering an event. It takes one state time to compare each CAM entry, so eight state times are needed for a complete CAM search. The HSO triggers the event when the selected timer matches the programmed time value.

The HSO can trigger 15 different events, which are classified as *external* or *internal*. The *external events* are those that toggle one or more of the HSO outputs; the *internal events* are those that set up the software timers, reset Timer 2, and start an A/D conversion. All of these events set flags that can optionally generate interrupts. External events generate the High-Speed Output interrupt (INT03); internal events generate the Software Timer interrupt (INT05).

The four software timers provide a means for generating interrupts at predefined times. They are most commonly used to trigger interrupt routines that must occur at regular intervals. At the specified time, the HSO sets a flag in the IOS1 register; if the interrupt is enabled, it also generates a Software Timer interrupt. If more than one software timer expires within the same time frame, multiple status bits are set. The interrupt service routine can examine the IOS1 register to determine which software timer(s) caused the interrupt.

8.3.2. Programming the HSO Module

Table 8-7 lists the registers that affect the performance and function of the HSO module. Refer to Appendix C, "8XC196KC/KD Registers," for detailed information.

Table 8-7. HSO Control and Status Registers

Register Mnemonic	Register Name	Description
HSO_COMMAND	HSO Command	Determines what event or events the HSO will trigger at the specified time.
HSO_TIME	HSO Time	Specifies the time at which an HSO command is to be executed.
INT_MASK	Interrupt Mask	Enables or disables the HSO interrupts.
IOC1	Input/Output Control Register 1	Enables or disables HSO.4 and HSO.5 as outputs.
IOC2	Input/Output Control Register 2	Enables and disables command locking within the HSO CAM file; can also clear the HSO CAM.
IOS0	Input/Output Status Register 0	Indicates the current state of the HSO pins, holding register, and CAM. Writing to the bits can set or clear the corresponding HSO pin.
IOS1	Input/Output Status Register 1	Contains flags that indicate which internal events triggered interrupts.
IOS2	Input/Output Status Register 2	Contains flags that indicate which external HSO events have occurred.

8.3.2.1. PROGRAMMING HSO EVENTS

The contents of the HSO_TIME and HSO_COMMAND registers together define each HSO event. The HSO_TIME register specifies the time at which an HSO command is to be executed. Bits 0–3 of the HSO_COMMAND register contain the command that determines what event or events the HSO will trigger at the specified time (see Figure 8-9). Table 8-9 lists the HSO command tags. The HSO_COMMAND register also determines whether the event will generate an interrupt, controls whether a pin toggle command sets or clears the affected pin(s), selects the timer reference for the HSO command, and controls whether the command is locked into the CAM or cleared upon execution (see Table 8-9).

Table 8-8. CMD_TAG Encoding

CMD_TAG (in Hex)	Command Mnemonic	Definition
00	HSO0	Switch High-Speed Output 0
01	HSO1	Switch High-Speed Output 1
02	HSO2	Switch High-Speed Output 2
03	HSO3	Switch High-Speed Output 3
04	HSO4	Switch High-Speed Output 4
05	HSO5	Switch High-Speed Output 5
06	HSO01 *	Switch High-Speed Outputs 0 and 1
07	HSO23 *	Switch High-Speed Outputs 2 and 3
08	SWT0	Program Software Timer 0
09	SWT1	Program Software Timer 1
0A	SWT2	Program Software Timer 2
0B	SWT3	Program Software Timer 3
0C	HSOALL *	Switch High-Speed Outputs 0, 1, 2, 3, 4, 5
0D	—	Reserved; do not use
0E	T2RST	Reset Timer 2
0F	A_D	Start an A/D Conversion

* In these configurations, two or more pins are set or cleared simultaneously.

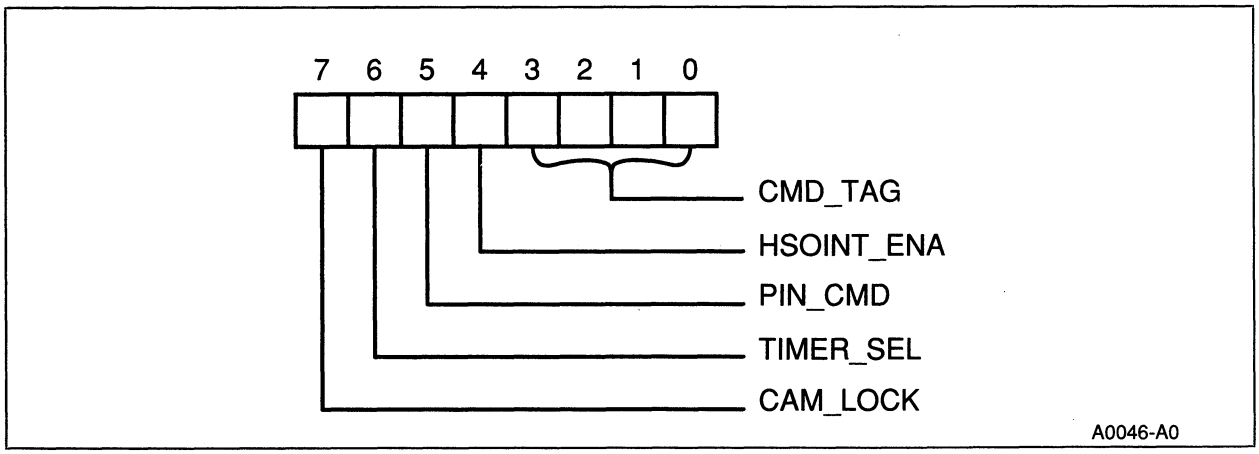

Figure 8-9. HSO_COMMAND Register

Table 8-9. HSO_COMMAND Register Bit Descriptions

Bit Number(s)	Bit Mnemonic	Bit Name	Description
0–3	CMD_TAG	HSO Commands	Determines what event or events will occur at the time specified in HSO_TIME. (See the "CMD_TAG Encoding" table.)
4	HSOINT_ENA	Enable/ Disable HSO Interrupt	Determines whether an HSO event generates an interrupt. 1= generate an interrupt 0= no interrupt When this bit is set, programmed HSO pin events generate the High-Speed Output interrupt (INT03, 2006H) and internal events generate the Software Timer interrupt (INT05, 200AH). When an interrupt is generated, the HSO event status bits in the IOS1 and IOS2 registers must be interrogated to determine which event caused the interrupt.
5	PIN_CMD	Set/Clear Selected HSO Pin	Determines whether the effect of a CMD_TAG command sets or clears the specified pin or pins. 1= Set the pin(s) 0= Clear the pin(s)
6	TIMER_SEL	Select Timer 1/ Timer 2	Selects the timer reference for the HSO command. 1= Select Timer 2 0= Select Timer 1
7	CAM_LOCK	Lock Entry Into CAM	When IOC2.6 is set (command locking enabled), this bit controls whether the HSO command is locked into the CAM or cleared after execution. 1= Lock command into CAM 0= Clear command from CAM after execution Writing a "1" to IOC2.7 clears all entries (locked or not) from the CAM, as does a chip reset.

NOTE

Before programming the HSO_COMMAND and HSO_TIME registers, read either IOS0.7 or IOS0.6 to verify that the HSO holding register is empty. If IOS0.7 is cleared, the holding register is empty. If IOS0.6 is cleared, the holding register is empty **and** at least one CAM register is empty. Writing to HSO_TIME when the holding register is not empty overwrites the current holding register value.

To enter a command into the CAM file, first write to the HSO_COMMAND register to define the event, and then write to the HSO_TIME register to specify the relative time at which the event is to occur. Writing to the HSO_TIME registers loads the HSO holding register. The command is held in the holding register until an empty CAM register is available, at which time the command enters the CAM file. It can take up to eight state times for a command to move from the holding register to the CAM file, so always allow at least eight state times between consecutive writes to the HSO_COMMAND and HSO_TIME registers. Otherwise, the command in the holding register can be overwritten before it is loaded into the CAM file.

NOTE

Commands in the holding register will not execute, even if their time tag is reached. Commands must be in the CAM file to execute.

The value in HSO_TIME can be either a specific timer value or a value that is offset from the current timer value. Use the standard load instruction to program a specific time. Otherwise, use a three-operand ADD instruction to specify a value that is offset from the current timer value.

```
LDB HSO_COMMAND, #what_to_do                ; Loads command
ADD HSO_TIME, TIMER1, #when_to_do_it        ; ADD_3op
```

To provide proper synchronization, the minimum time that should be loaded into a timer is *current_timer_value* + 2. Smaller values may cause the timer match to occur 65,636 counts later than expected. This restriction applies to both Timer 1 and Timer 2.

Use care when writing to the HSO_COMMAND and HSO_TIME registers. An interrupt can occur after the command tag is written to HSO_COMMAND and before the time value is loaded into HSO_TIME. If the interrupt routine writes to these registers, its command tag overwrites the main program's command tag in the CAM file, so the main program's command is never executed. Disabling interrupts when writing to the HSO module prevents this problem. (See "Modifying Interrupt Priorities" in Chapter 5.)

8.3.2.2. ENABLING THE HSO INTERRUPTS

Setting the HSOINT_ENA bit (HSO_COMMAND.4) causes an HSO event to generate an interrupt. The HSO module can generate two different interrupts: the High-Speed Output interrupt (INT03) and the Software Timer interrupt (INT05).

External events generate the High-Speed Output interrupt, if the interrupt is enabled (INT_MASK.3 set). The status bits related to the High-Speed Output interrupt are IOS2.0–IOS2.5. *Internal events* generate the Software Timer interrupt, if the interrupt is enabled (INT_MASK.5 set). The status bits related to the Software Timer interrupt are IOS1.0–IOS1.3 for the software timer commands and IOS2.6–IOS2.7 for Timer 2 reset and A/D conversion start, respectively.

When an interrupt is generated, read the HSO event status bits in the IOS1 and IOS2 registers to determine which event caused the interrupt. Reading the IOS1 register clears bits 0–5; reading the IOS2 register clears all bits. For this reason, we recommend that you copy the contents of the registers into shadow registers and then execute bit-test instructions such as JBC or JBS on the shadow registers.

8.3.2.3. LOCKING CAM ENTRIES

The LOCK_ENA bit in the IOC2 register (IOC2.6) enables and disables command locking. When this bit is set, the CAM_LOCK bit (HSO_COMMAND.7) controls whether individual commands are locked into the CAM or cleared after execution. When CAM_LOCK is set, the command remains in the CAM after execution. This feature makes it easy to generate periodic events or waveforms. Clearing CAM_LOCK causes the HSO to clear the command from the CAM after execution.

Locked entries provide the ability to program periodic or repetitive events, while minimizing the software overhead. One of the most useful features is that locked entries enable the HSO to generate multiple pulse width modulated (PWM) outputs, with minimum software overhead. (See "Using the HSO Module to Generate PWM Outputs.") A locked A/D conversion command causes the HSO to generate multiple A/D conversions. When Timer 2 is used as the HSO reference, a locked T2RST command can generate periodic events; events with HSO_TIME less than the Timer 2 reset value occur repeatedly as Timer 2 resets. Locked software timer commands can schedule recurrent software tasks; interrupt service routines can execute the tasks without having to program another HSO software timer command.

8.3.2.4. CLEARING THE CAM FILE

Three occurrences can remove an entry from the CAM:

- if the specified timer matches the programmed value and the event is not locked
- the device is reset
- the CAM_CLR bit (IOC2.7) is set

To clear locked events, set the CAM_CLR bit (IOC2.7) or reset the device. Either of these options will clear the entire CAM.

8.3.2.5. CANCELING AN EVENT

You can cancel an *external event* by writing the opposite event with the same time tag to the CAM. For example, if a command that sets HSO.1 when TIMER1 = 1234 is followed by a command that clears HSO.1 when TIMER1 = 1234, HSO.1 does not toggle. However, both commands remain in the CAM until either TIMER1 = 1234 (if the entries are not locked), the device is reset, or the CAM_CLR bit (IOC2.7) is set.

You can cancel an *internal event* only by setting the CAM_CLR bit (IOC2.7) or resetting the device.

8.3.2.6. ENABLING THE HSO.4 AND HSO.5 PINS

The HSO.4 and HSO.5 outputs are multiplexed with the HSI.2 and HSI.3 inputs, respectively. These pins can be enabled for both functions (see Figure 8-7 on page 8-16). Setting the HSO4_ENA bit (IOC1.4) enables HSO.4 as an output; setting the HSO5_ENA bit (IOC1.6) enables HSO.5.

8.3.2.7. USING HSO.0–HSO.6 AS OUTPUT PINS

The HSO outputs can provide up to six additional output pins if the HSO function is not needed. To use the HSO pins as standard outputs, enable the pins and then set or clear them by writing to the IOS0 register in HWindow 15.

8.3.3. Using the HSO Module to Generate PWM Outputs

Both the HSO module and the PWM modules can generate a rectangular pulse train that varies in duty cycle and period. With proper components, a highly accurate 8-bit D/A converter can be made using either the HSO or the PWM outputs. (See "Generating Analog Outputs" in Chapter 10.)

The HSO module can generate either a single PWM waveform or multiple PWM waveforms. A single waveform requires a programmed *period* and *clear_time* for the output pin and two HSO commands: one to set the pin and one to clear it. Either Timer 1 or Timer 2 can be used as the reference.

Generating multiple waveforms is done in a similar way, using Timer 2 as the reference. Multiple waveform generation requires a programmed *period* for Timer 2, a set time (*PWMx_ST*) and a clear time (*PWMx_CT*) for each output pin, and multiple HSO commands: one to reset Timer 2, several to set the pins, and several to clear the pins.

The following example shows a common way to generate multiple PWMs using HSO.0, HSO.1, and HSO.2 as the PWM outputs and Timer 2 as the reference timer. Figure 8-10 shows the three resulting PWMs.

```
SET_0:      LDB HSO_COMMAND, #1110000B      ;set HSO.0 when
            LD HSO_TIME, #PWM0_ST           ;timer2 = pwm0_st
            SKIP R0                         ;wait 4 state times
            SKIP R0                         ;wait 4 state times
SET_1:      LDB HSO_COMMAND, #1110001B      ;set HSO.1 when
            LD HSO_TIME, #PWM1_ST           ;timer2 = pwm1_st
            SKIP R0                         ;wait 4 state times
            SKIP R0                         ;wait 4 state times
SET_2:      LDB HSO_COMMAND, #1110010B      ;set HSO.2 when
            LD HSO_TIME, #PWM2_ST           ;timer2 = pwm2_st
            SKIP ZERO_REG                   ;wait 4 state times
            SKIP ZERO_REG                   ;wait 4 state times
CLEAR_0:    LDB HSO_COMMAND, #1100000B      ;clear HSO.0 when
            LD HSO_TIME, #PWM0_ST           ;timer2 = pwm0_ct
            SKIP ZERO_REG                   ;wait 4 state times
            SKIP ZERO_REG                   ;wait 4 state times
CLEAR_1:    LDB HSO_COMMAND, #1100001B      ;clear HSO.1 when
            LD HSO_TIME, #PWM1_ST           ;timer2 = pwm1_ct
            SKIP ZERO_REG                   ;wait 4 state times
            SKIP ZERO_REG                   ;wait 4 state times
CLEAR_2:    LDB HSO_COMMAND, #1100010B      ;clear HSO.2 when
            LD HSO_TIME, #PWM2_ST           ;timer2 = pwm2_ct
            SKIP ZERO_REG                   ;wait 4 state times
            SKIP ZERO_REG                   ;wait 4 state times
RESET:      LDB HSO_COMMAND, #11001110B     ;reset Timer 2 when
            LD HSO_TIME, #PERIOD            ;timer2 = period
```

Refer to Application Note AP-466, "Using the 80C196KB," for additional information and software examples that show how to use the HSO module to generate single and multiple PWMs.

8.3.4. HSO Output Timing

The timing of the HSO pins is synchronized to the specified timer. All external HSO pins that are due to change at a certain timer value will change just after the timer is incremented. Internally, the timer changes every eight state times during Phase 1. From an external perspective, the HSO pin should change just before the falling edge of CLKOUT and should be stable by its rising edge. Information from the HSO can be latched on the CLKOUT rising edge. Internal HSO events also occur when the reference timer increments.

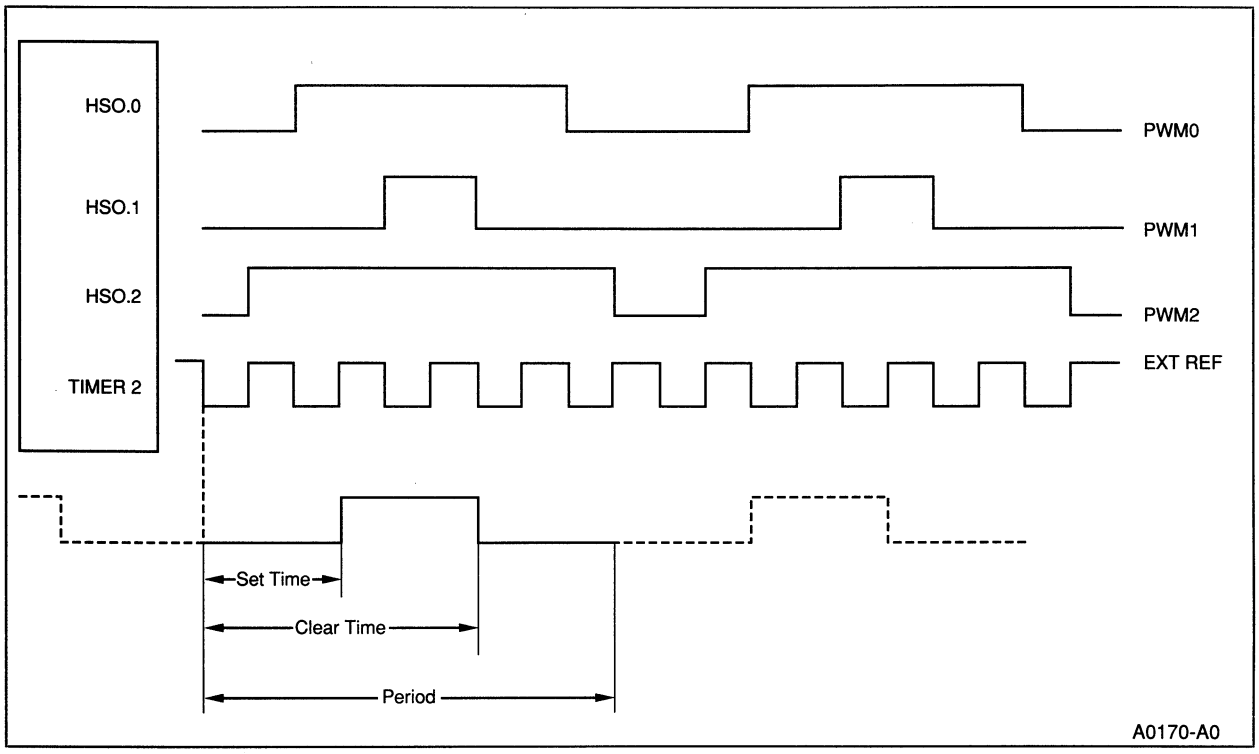

Figure 8-10. Example PWM Waveforms