T-52-33-13

National Semiconductor

NS32202-10 Interrupt Control Unit

General Description

The NS32202 Interrupt Control Unit (ICU) is the interrupt controller for the Series 32000® microprocessor family. It is a support circuit that minimizes the software and real-time overhead required to handle multi-level, prioritized interrupts. A single NS32202 manages up to 16 interrupt sources, resolves interrupt priorities, and supplies a single-byte interrupt vector to the CPU.

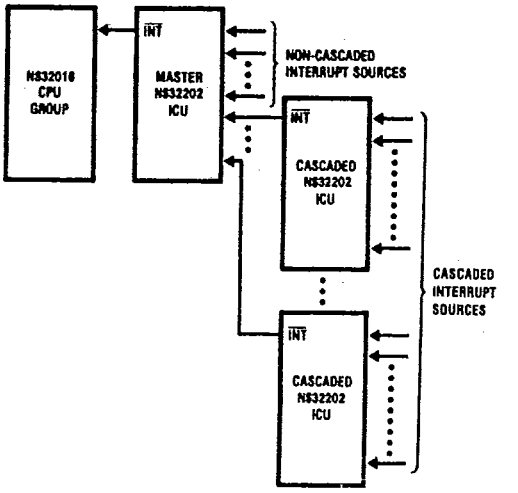
The NS32202 can operate in either of two data bus modes: 16-bit or 8-bit. In the 16-bit mode, eight hardware and eight software interrupt positions are available. In the 8-bit mode, 16 hardware interrupt positions are available, 8 of which can be used as software interrupts. In this mode, up to 16 additional ICUs may be cascaded to handle a maximum of 256 interrupts.

Two 16-bit counters, which may be concatenated under program control into a single 32-bit counter, are also available for real-time applications.

Features

- 16 maskable interrupt sources, cascadable to 256
- Programmable 8- or 16-bit data bus mode
- Edge or level triggering for each hardware interrupt with individually selectable polarities
- 8 software interrupts
- Fixed or rotating priority modes
- Two 16-bit, DC to 10 MHz counters, that may be concatenated into a single 32-bit counter
- Optional 8-bit I/O port available in 8-bit data bus mode
- High-speed XMOS™ technology
- Single, +5V supply
- 40-pin, dual in-line package

Basic System Configuration



TL/EE/5117-1

4

Table of Contents

List of Illustrations

1.0 Product Introduction

The NS32202 ICU functions as an overall manager in an interrupt-oriented system environment. Its many features and options permit the design of sophisticated interrupt systems.

Figure 1-1 shows the internal organization of the NS32202. As shown, the NS32202 is divided into five functional blocks. These are described in the following paragraphs:

1.1 I/O BUFFERS AND LATCHES

The I/O Buffers and Latches block is the interface with the system data bus. It contains bidirectional buffers for the data I/O pins. It also contains registers and logic circuits that control the operation of pins G0/IR0, . . . ,G7/IR14 when the ICU is in the 8-bit bus mode.

1.2 READ/WRITE LOGIC AND DECODERS

The Read/Write Logic and Decoders manage all internal and external data transfers for the ICU. These include Data, Control, and Status Transfers. This circuit accepts inputs from the CPU address and control buses. In turn, it issues commands to access the internal registers of the ICU.

1.3 TIMING AND CONTROL

The Timing and Control Block contains status elements that select the ICU operating mode. It also contains state machines that generate all the necessary sequencing and control signals.

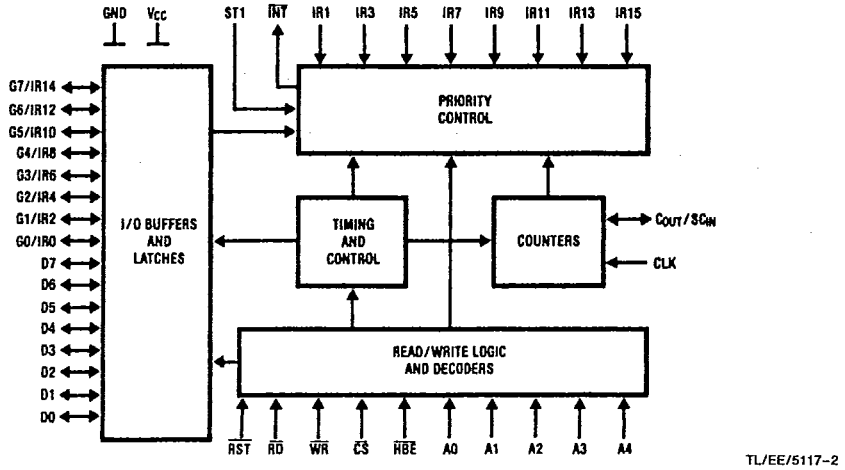
1.4 PRIORITY CONTROL

The Priority Control Block contains 16 units, one for each interrupt position. These units provide the following functions.

- Sensing the various forms of hardware interrupt signals e.g. level (high/low) or edge (rising/falling)
- Resolving priorities and generating an interrupt request to the CPU
- Handling cascaded arrangements
- Enabling software interrupts
- Providing for an automatic return from interrupt
- Enabling the assignment of any interrupt position to the internal counters
- Providing for rearrangement of priorities by assigning the first priority to any interrupt position
- Enabling automatic rotation of priorities

1.5 COUNTERS

This block contains two 16-bit counters, called the H-counter and the L-counter. These are down counters that count from an initial value to zero. Both counters have a 16-bit register (designated HCSV and LCSV) for loading their restarting values. They also have registers containing the current count values (HCCV and LCCV). Both sets of registers are fully described in Section 3.



TL/EE/5117-2

FIGURE 1-1. NS32202 ICU Block Diagram

1.0 Product Introduction (Continued)

The counters are under program control and can be used to generate interrupts. When the count reaches zero, either counter can generate an interrupt request to any of the 16 interrupt positions. The counter then reloads the start value from the appropriate registers and resumes counting. *Figure 1–2* shows typical counter output signals available from the NS32202.

The maximum input clock frequency is 2.5 MHz.

A divide-by-four prescaler is also provided. When the prescaler is used, the input clock frequency can be up to 10 MHz.

When intervals longer than provided by a 16-bit counter are needed, the L- and H-counters can be concatenated to form a 32-bit counter. In this case, both counters are controlled by the H-counter control bits. Refer to the discussion of the Counter Control Register in Section 3 for additional information. *Figure 1-3* summarizes counter read/write operations.

2.0 Functional Description

2.1 RESET

T-52-33-13

The ICU is reset when a logic low signal is present on the $\overline{RST}$ pin. At reset, most internal ICU registers are affected, and the ICU becomes inactive.

2.2 INITIALIZATION

After reset, the CPU must initialize the NS32202 to establish its configuration. Proper initialization requires knowledge of the ICU register's formats. Therefore, a flowchart of a recommended initialization sequence is shown in (*Figure 3–3*) after the discussion of the ICU registers.

The operation sequence shown in *Figure 3–3* ensures that all counter output pins remain inactive until the counters are completely initialized.

2.3 VECTORED INTERRUPT HANDLING

For details on the operation of the vectored interrupt mode for a particular Series 32000 CPU, refer to the data sheet for

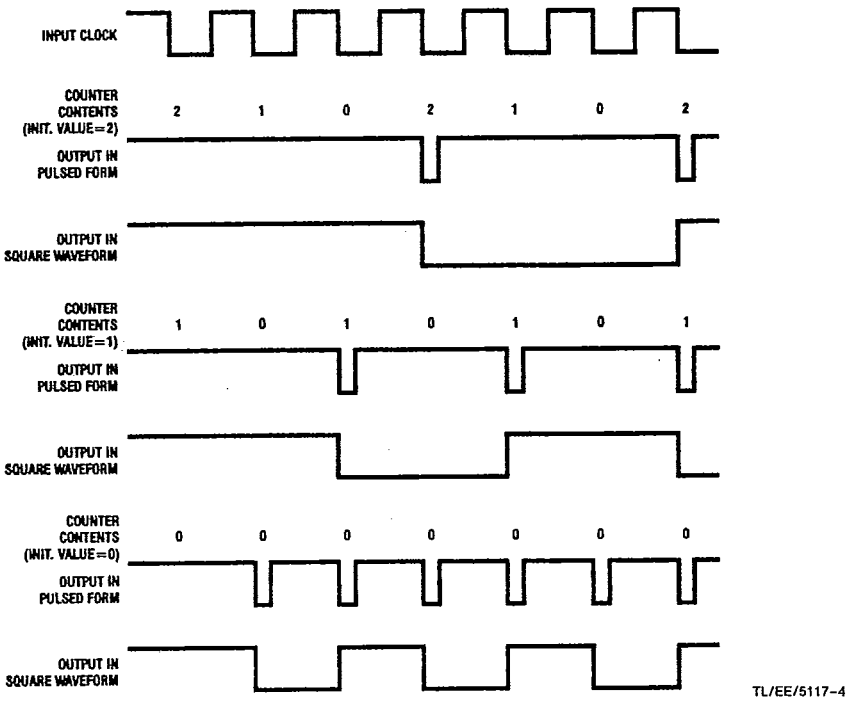
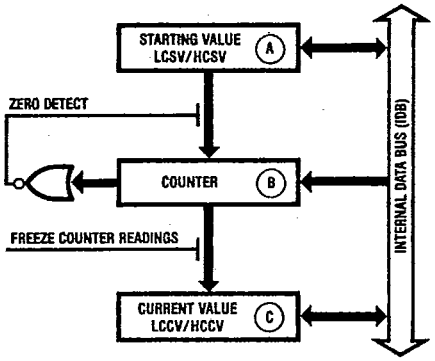


FIGURE 1–2. Counter Output Signals in Pulsed Form and Square Waveform for Three Different Initial Values

2.0 Functional Description (Continued)

that CPU. In this discussion, it is assumed that the NS32202 is working with a CPU in the vectored interrupt mode. Several ICU applications are discussed, including non-cascaded and cascaded operation. *Figures 2–1, 2–2,* and *2–3* show typical configurations of the ICU used with the NS32016 CPU.

A peripheral device issues an interrupt request by sending the proper signal to one of the NS32202 interrupt inputs. If the interrupt input is not masked, the ICU activates its Interrupt Output ($\overline{INT}$) pin and generates an interrupt vector byte. The interrupt vector byte identifies the interrupt source in its four least significant bits. When the CPU detects a low level on its Interrupt Input pin, it performs one or two interrupt acknowledge cycles depending on whether the interrupt request is from the master ICU or a cascaded ICU. *Figure 2–4* shows a flowchart of a typical CPU Interrupt Acknowledge sequence.



TL/EE/5117–5

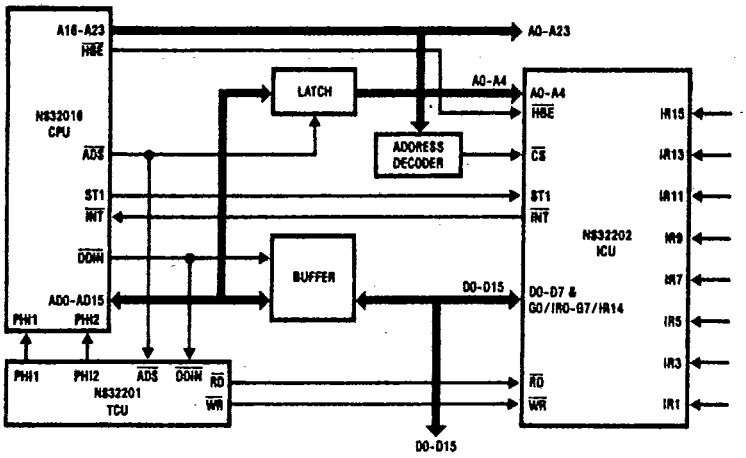
BASIC OPERATIONS:

WRITING TO LCSV/HCSV	[A] ← (IDB)
READING LCSV/HCSV	[A] → (IDB)
WRITING TO LCCV/HCCV	[B] ← (IDB)
(only possible when counters are halted)	[C] ← (IDB)
READING LCCV/HCCV	[C] → (IDB)
(only possible when counter readings are frozen)	
COUNTER COUNTS AND READINGS ARE NOT FROZEN	[C] ← [B]
COUNTER RELOADS STARTING VALUE	[B] ← [A]
(occurs on the clock cycle following the one in which it reaches zero)	

FIGURE 1–3. Counter Configuration and Basic Operations

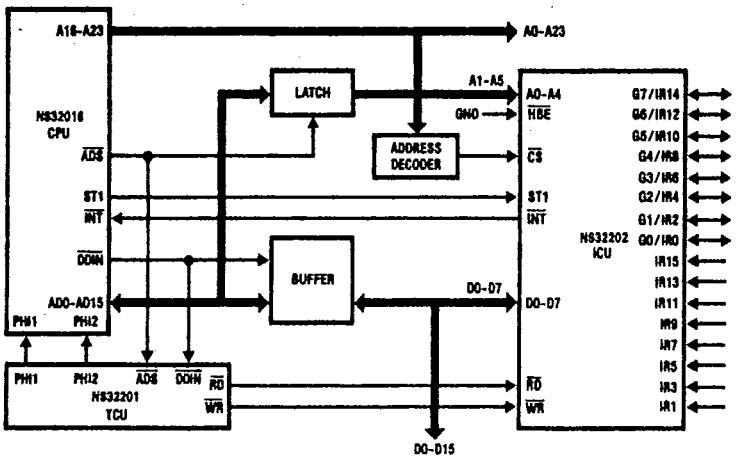
4

2.0 Functional Description (Continued)



TL/EE/5117-6

FIGURE 2-1. Interrupt Control Unit Connections in 16-Bit Bus Mode

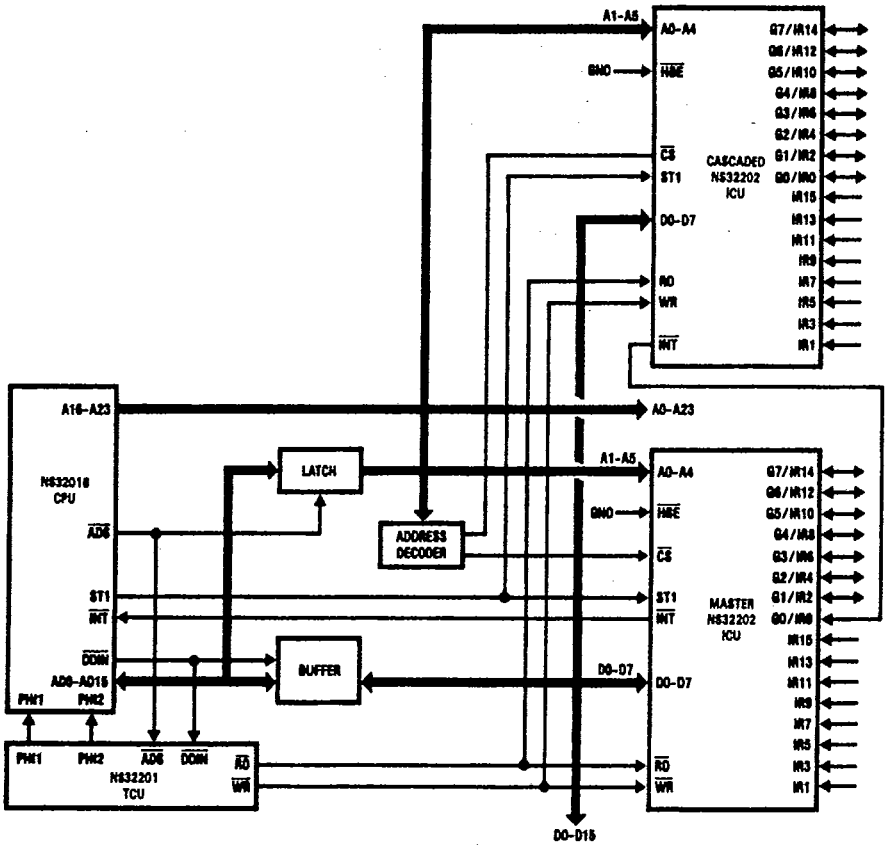


TL/EE/5117-7

NOTE: In the 8-Bit Bus Mode the Master ICU Registers appear at even addresses (A0 = 0) since the ICU communicates with the least significant byte of the CPU data bus.

FIGURE 2-2. Interrupt Control Unit Connections in 8-Bit Bus Mode

2.0 Functional Description (Continued)

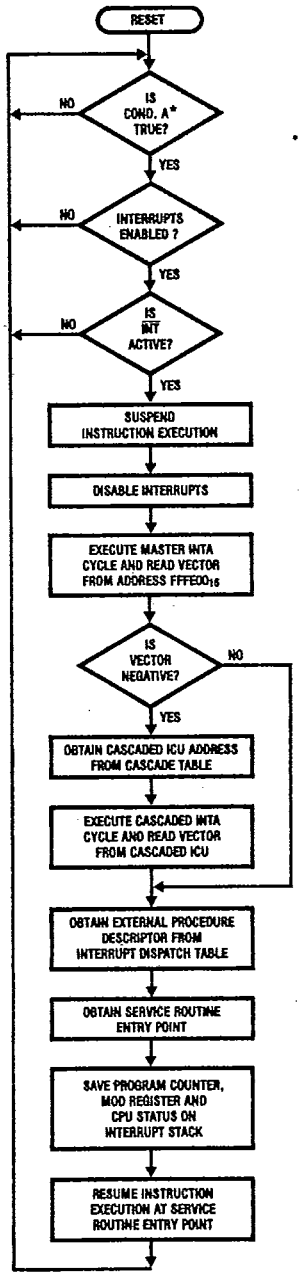


TL/EE/5117-8

FIGURE 2-3. Cascaded Interrupt Control Unit Connections in 8-Bit Bus Mode

4

2.0 Functional Description (Continued)



* Cond. A is true if current instruction is terminated or an interruptible point in a string instruction is reached.

TL/EE/5117–9

FIGURE 2–4. CPU Interrupt Acknowledge Sequence

2.0 Functional Description (Continued)

In general, vectored interrupts are serviced by interrupt routines stored in system memory. The Dispatch Table stores up to 256 external procedure descriptors for the various service procedures. The CPU INTBASE register points to the top of the Dispatch Table. *Figure 2–5* shows the layout of the Dispatch Table. This figure also shows the layout of the Cascade Table, which is discussed with ICU cascaded operation.

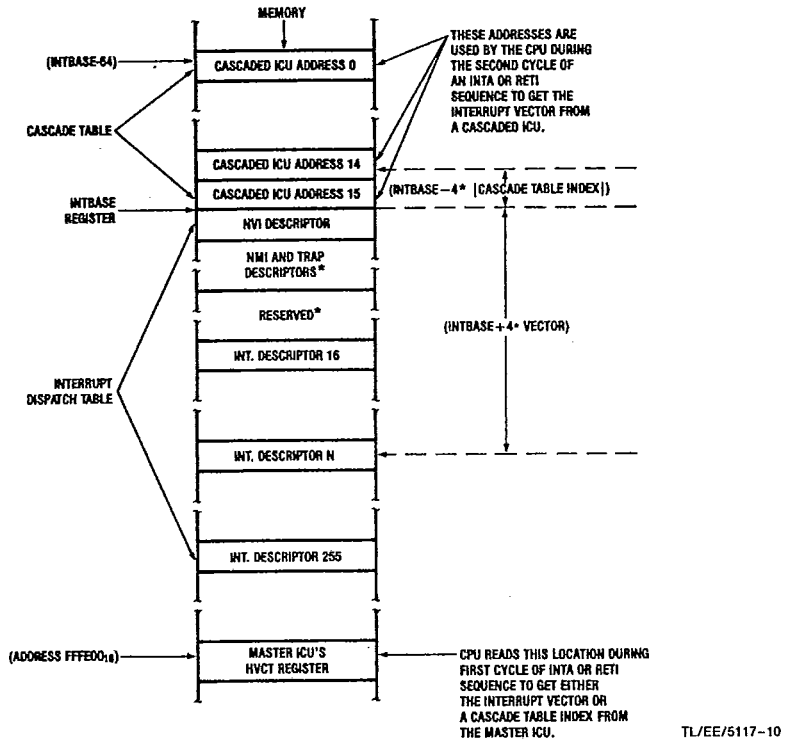
2.3.1 Non-Cascaded Operation. Whenever an interrupt request from a peripheral device is issued directly to the master ICU, a non-cascaded interrupt request to the CPU results. In a system using a single NS32202, up to 16 interrupt requests can be prioritized. Upon receipt of an interrupt request on the $\overline{INT}$ pin, the CPU performs a Master Interrupt-Acknowledge bus cycle, reading a vector byte from address $FFFE00_{16}$. This vector is then used as an index into the dispatch table in order to find the External Procedure Descriptor for the proper interrupt service procedure. The service procedure eventually returns via the Return-from-Interrupt (RET) instruction, which performs a Return-from-Interrupt bus cycle, informing the ICU that it may re-prioritize any interrupt requests still pending. *Figure 2–6* shows a typical CPU RETI sequence. In a system with only one ICU, the vectors provided must be in the range of 0 through 127; this can be ensured by writing 0XXXXXXX into the SVCT register. By providing a negative vector value, the master ICU flags the interrupt source as a cascaded ICU (see below).

2.3.2 Cascaded Operation. In cascaded operation, one or more of the interrupt inputs of the master ICU are connected to the Interrupt Output pin of one or more cascaded ICUs. Up to 16 cascaded ICUs may be used, giving a system total of 256 interrupts.

Note: The number of cascaded ICUs is practically limited to 15 because the Dispatch Table for the NS32016 CPU is constructed with entries 1 through 15 either used for NMI and Trap descriptors, or reserved for future use. Interrupt position 0 of the master ICU should not be cascaded, so it can be vectored through Dispatch Table entry 0, reserved for non-vectored interrupts. In this case, the non-vectored interrupt entry (entry 0) is also available for vectored interrupt operation, since the CPU is operating in the vectored interrupt mode.

The address of the master ICU should be $FFFE00_{16}$. (*) Cascaded ICUs can be located at any system address. A list of cascaded ICU addresses is maintained in the Cascade Table as a series of sixteen 32-bit entries.

(*)Note: The CPU status corresponding to both, master interrupt acknowledge and return from interrupt bus cycles, as well as address bit A8, could be used to generate the chip select ($\overline{CS}$) signal for accessing the master ICU during one of the above cycles. In this case the master ICU can reside at any system address. The only limitation is that the least significant 5 or 6 address bits (6 in the 8-bit bus mode) must be zero. The address bit A8 must be decoded to prevent an NMI bus cycle from reading the hardware vector register of the ICU. This could happen, since the NS32016 CPU performs a dummy read cycle from address $FFFF00_{16}$, with the same status as a master INTA cycle, when a non-maskable-interrupt is acknowledged.



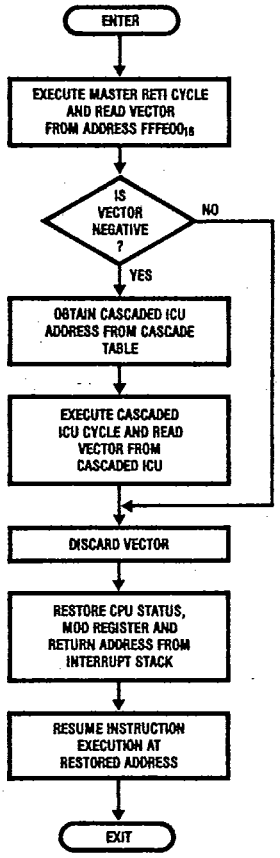
TL/EE/5117–10

* Table entries 1 to 15 should not be used by the ICU since they contain NMI and Trap Descriptors or are reserved for future use. (For more details refer to NS32016 data sheet.)

FIGURE 2–5. Interrupt Dispatch and Cascade Tables

4

2.0 Functional Description (Continued)



TL/EE/5117–11

FIGURE 2–6. CPU Return from Interrupt Sequence

The master ICU maintains a list (in the CSRC register pair) of its interrupt positions that are cascaded. It also provides a 4-bit (hidden) counter (in-service counter) for each interrupt position to keep track of the number of interrupts being serviced in the cascade ICUs. When a cascaded interrupt input is active, the master ICU activates its interrupt output and the CPU responds with a Master Interrupt Acknowledge Cycle. However, instead of generating a positive interrupt vector, the master ICU generates a negative Cascade Table index.

The CPU interprets the negative number returned from the master ICU as an index into the Cascade Table. The Cascade Table is located in a negative direction from the Dispatch Table, and it contains the virtual addresses of the hardware vector registers for any cascaded NS32202s in the system. Thus, the Cascade Table index supplied by the master ICU identifies the cascaded ICU that requested the interrupt.

Once the cascaded ICU is identified, the CPU performs a Cascaded Interrupt Acknowledge cycle. During this cycle, the CPU reads the final vector value directly from the cascaded ICU, and uses it to access the Dispatch Table. Each cascaded ICU, of course, has its own set of 16 unique interrupt vectors, one vector for each of its 16 interrupt positions.

The CPU interprets the vector value read during a Cascaded Interrupt Acknowledge cycle as an unsigned number. Thus, this vector can be in the range 0 through 255.

When a cascaded interrupt service routine completes its task, it must return control to the interrupted program with the same RETI instruction used in non-cascaded interrupt service routines. However, when the CPU performs a Master Return From Interrupt cycle, the CPU accesses the master ICU and reads the negative Cascade Table index identifying the cascaded ICU that originally received the interrupt request. Using the cascaded ICU address, the CPU now performs a Cascaded Return From Interrupt cycle, informing the cascaded ICU that the service routine is over. The byte provided by the cascaded ICU during this cycle is ignored.

2.4 INTERNAL ICU OPERATING SEQUENCE

The NS32202 ICU accepts two interrupt types, software and hardware.

Software interrupts are initiated when the CPU sets the proper bit in the Interrupt Pending (IPND) registers (R6, R7), located in the ICU. Bits are set and reset by writing the proper byte to either R6 or R7. Software interrupts can be masked, by setting the proper bit in the mask registers (R10, R11).

Hardware interrupts can be either internal or external to the ICU. Internal ICU hardware interrupts are initiated by the on-chip counter outputs. External hardware interrupts are initiated by devices external to the ICU, that are connected to any of the ICU interrupt input pins.

Hardware interrupts can be masked by setting the proper bit in the mask registers (R10, R11). If the Freeze bit (FRZ), located in the Mode Control Register (MCTL), is set, all incoming hardware interrupts are inhibited from setting their corresponding bits in the IPND registers. This prevents the ICU from recognizing any hardware interrupts.

Once the ICU is initialized, it is enabled to accept interrupts. If an active interrupt is not masked, and has a higher priority than any interrupt currently being serviced, the ICU activates its Interrupt Output ($\overline{INT}$). *Figure 2–7* is a flowchart showing the ICU interrupt acknowledge sequence.

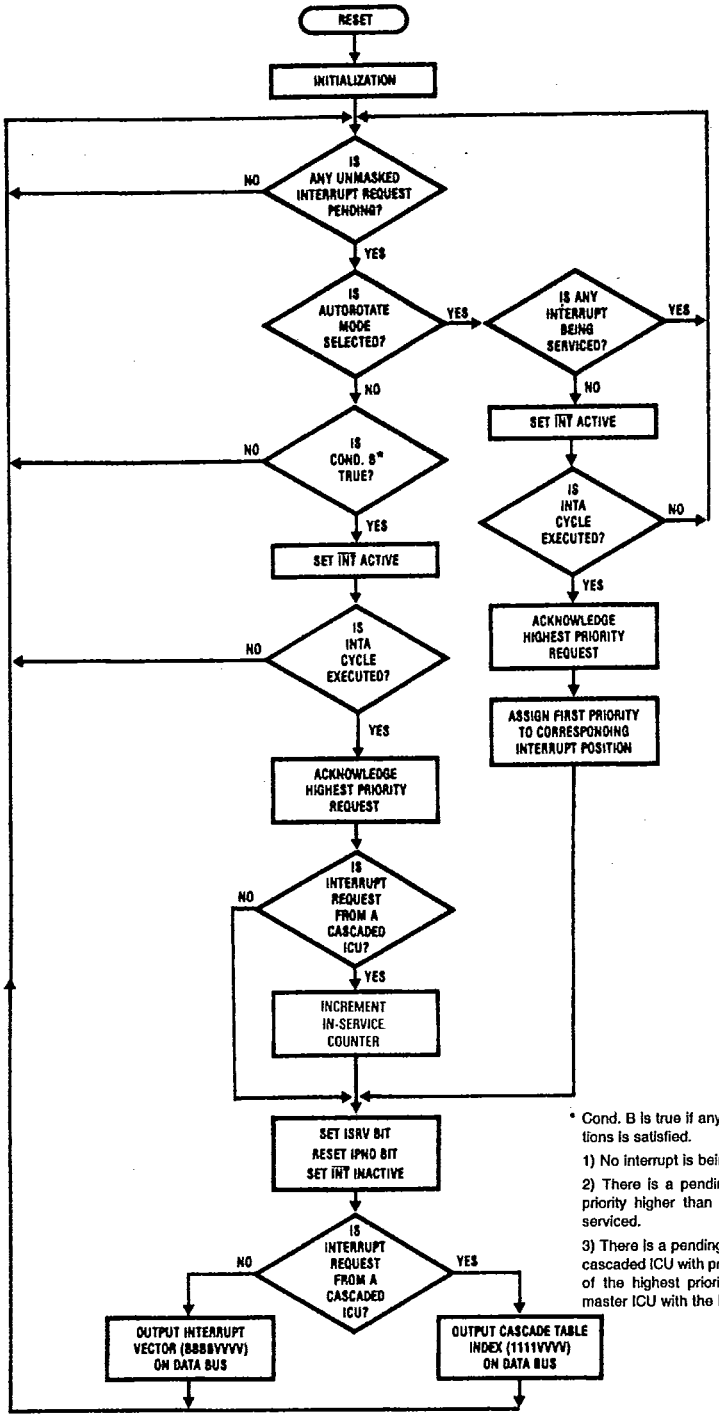
The CPU responds to the active $\overline{INT}$ line by performing an Interrupt Acknowledge bus cycle. During this cycle, the ICU clears the IPND bit corresponding to the active interrupt position and sets the corresponding bit in the Interrupt In-Service Registers (ISRV). The 4-bit in-service counter in the master ICU is also incremented by one if the fixed priority mode is selected and the interrupt is from a cascaded ICU. The ISRV bit remains set until the CPU performs a RETI bus cycle and the 4-bit in-service counter is decremented to zero. *Figure 2–8* is a flowchart showing ICU operation during a RETI bus cycle.

When the ISRV bit is set, the $\overline{INT}$ output is disabled. This output remains inactive until a higher priority interrupt position becomes active, or the ISRV bit is cleared.

An exception to the above occurs in the master ICU when the fixed priority mode is selected, and the interrupt input is connected to the $\overline{INT}$ output of a cascaded ICU. In this case the ISRV bit does not inhibit an interrupt of the same priority.

This is to allow nesting of interrupts in a cascaded ICU.

2.0 Functional Description (Continued)



* Cond. B is true if any one of the following conditions is satisfied.

1) No interrupt is being serviced

2) There is a pending unmasked interrupt with priority higher than that of the interrupt being serviced.

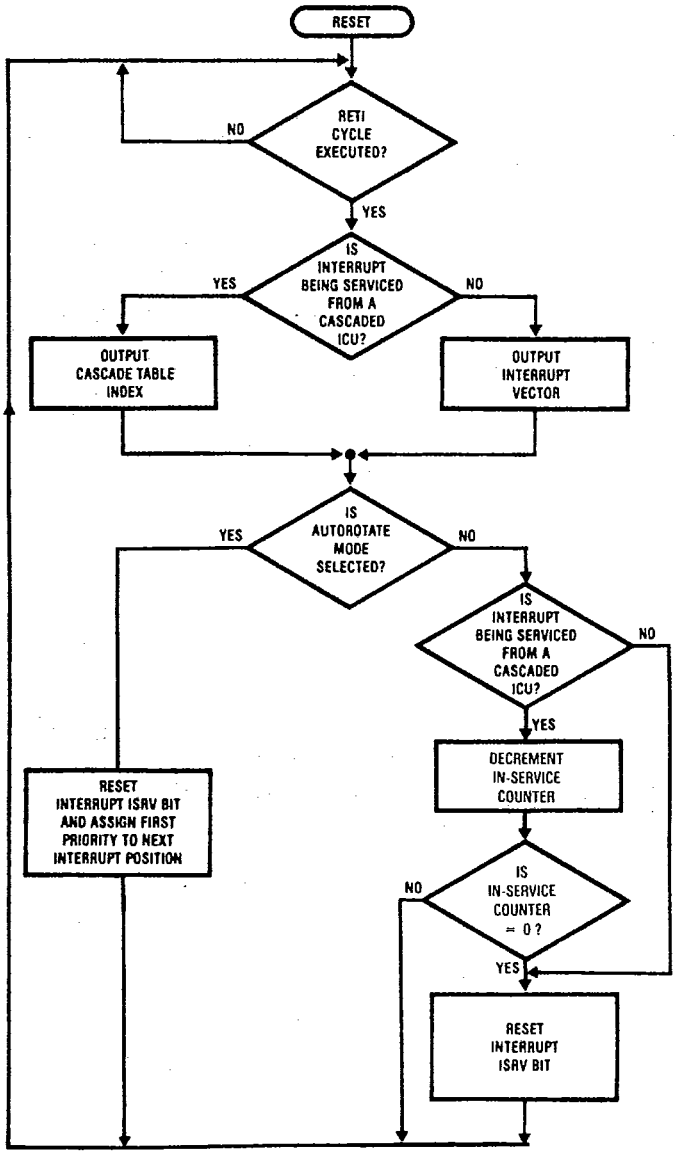
3) There is a pending unmasked interrupt from a cascaded ICU with priority higher or same as that of the highest priority interrupt position in the master ICU with the ISRV bit set.

TL/EE/5117–12

FIGURE 2–7. ICU Interrupt Acknowledge Sequence

4

2.0 Functional Description (Continued)



TL/EE/5117–13

FIGURE 2–8. ICU Return from Interrupt Sequence

2.0 Functional Description (Continued)

2.5 INTERRUPT PRIORITY MODES

The NS32202 ICU can operate in one of four interrupt priority modes: Fixed Priority; Auto-Rotate; Special Mask; and Polling. Each mode is described below.

2.5.1 Fixed Priority Mode

In the Fixed Priority Mode (also called Fully Nested Mode), each interrupt position is ranked in priority from 0 to 15, with 0 being the highest priority. In this mode, the processing of lower priority interrupts is nested with higher priority interrupts. Thus, while an interrupt is being serviced, any other interrupts of the same or lower priority are inhibited. The ICU does, however, recognize higher priority interrupt requests.

When the interrupt service routine executes its RETI instruction, the corresponding ISRV bit is cleared. This allows any lower priority interrupt request to be serviced by the CPU.

At reset, the default priority assignment gives interrupt IR0 priority 0 (highest priority), interrupt IR1 priority 1, and so forth. Interrupt IR15 is, of course, assigned priority 15, the lowest priority. The default priority assignment can be altered by writing an appropriate value into register FPRT (L) as explained in Section 3.9.

Note: When the ICU generates an interrupt request to the CPU for a higher priority interrupt while a lower priority interrupt is still being serviced by the CPU, the CPU responds to the interrupt request only if its internal interrupt enable flag is set. Normally, this flag is reset at the beginning of an interrupt acknowledge cycle and set during the RETI cycle. If the CPU is to respond to higher priority interrupts during any interrupt service routine, the service routine must set the internal CPU interrupt enable flag, as soon during the service routine as desired.

2.5.2 Auto-Rotate Mode

The Auto Rotate Mode is selected when the NTAR bit is set to 0, and is automatically entered after Reset. In this mode an interrupt source position is automatically assigned lowest priority after a request at that position has been serviced. Highest priority then passes to the next lower priority position. For example, when servicing of the interrupt request at position 3 is completed (ISRV bit 3 is cleared), interrupt position 3 is assigned lowest priority and position 4 assumes highest priority. The nesting of interrupts is inhibited, since the interrupt being serviced always has the highest priority.

This mode is used when the interrupting devices have to be assigned equal priority. A device requesting an interrupt, will have to wait, in the worst case, until each of the 15 other devices has been serviced at most once.

2.5.3 Special Mask Mode

The Special Mask Mode is used when it is necessary to dynamically alter the ICU priority structure while an interrupt is being serviced. For example, it may be desired in a particular interrupt service routine to enable lower priority interrupts during a part of the routine. To do so, the ICU must be programmed in fixed priority mode and the interrupt service routine must control its own in-service bit in the ISRV registers.

The bits of the ISRV registers are changed with either the Set Bit Interlocked or Clear Bit Interlocked instructions (SBITIW or CBITIW). The in-service bit is cleared to enable lower priority interrupts and set to disable them.

Note: For proper operation of the ICU, an interrupt service routine must set its ISRV bit before executing the RETI instruction. This prevents the RETI cycle from clearing the wrong ISRV bit.

2.5.4 Polling Mode

The Polling Mode gives complete control of interrupt priority to the system software. Either some or all of the interrupt positions can be assigned to the polling mode. To assign all interrupt positions to the polling mode, the CPU interrupt enable flag is reset. To assign only some of the interrupt positions to the polling mode, the desired interrupt positions are masked in the Interrupt Mask registers (IMSK). In either case, the polling operation consists of reading the Interrupt Pending (IPND) registers.

If necessary, the IPND read can be synchronized by setting the Freeze (FRZ) bit in the Mode Control register (MCTL). This prevents any change in the IPND registers during the read. The FRZ bit must be reset after the polling operation so the IPND contents can be updated. If an edge-triggered interrupt occurs while the IPND registers are frozen, the interrupt request is latched, and transferred to the IPND registers as soon as FRZ is reset.

The polling mode is useful when a single routine is used to service several interrupt levels.

3.0 Architectural Description

The NS32202 has thirty-two 8-bit registers that can be accessed either individually or in pairs. In 16-bit data bus mode, register pairs can be accessed with the CPU word or double-word reference instructions. *Figure 3-1* shows the ICU internal registers. This figure summarizes the name, function, and offset address for each register.

Because some registers hold similar data, they are grouped into functional pairs and assigned a single name. However, if a single register in a pair is referenced, either an L or an H is appended to the register name. The letters are placed in parentheses and stand for the low order 8 bits (L) and the high order 8 bits (H). For example, register R6, part of the Interrupt Pending (IPND) register pair, is referred to individually as IPND(L).

The following paragraphs give detailed descriptions of the registers shown in *Figure 3-1.*

3.1 HVCT — HARDWARE VECTOR REGISTER (R0)

The HVCT register is a single register that contains the interrupt vector byte supplied to the CPU during an Interrupt Acknowledge (INTA) or Return From Interrupt (RETI) cycle. The HVCT bit map is shown below:

7	6	5	4	3	2	1	0
B	B	B	B	V	V	V	V

3.0 Architectural Description (Continued)

REG. NUMBER AND ADDRESS IN HEX.		REG. NAME	REG. FUNCTION
	R0 (00_{16})	HVCT —	HARDWARE VECTOR
	R1 (01_{16})	SVCT —	SOFTWARE VECTOR
R3 (03_{16})	R2 (02_{16})	ELTG —	EDGE/LEVEL TRIGGERING
R5 (05_{16})	R4 (04_{16})	TPL —	TRIGGERING POLARITY
R7 (07_{16})	R6 (06_{16})	IPND —	INTERRUPTS PENDING
R9 (09_{16})	R8 (08_{16})	ISRV —	INTERRUPTS IN-SERVICE
R11 ($0B_{16}$)	R10 ($0A_{16}$)	IMSK —	INTERRUPT MASK
R13 ($0D_{16}$)	R12 ($0C_{16}$)	CSRC —	CASCADED SOURCE
R15 ($0F_{16}$)	R14 ($0E_{16}$)	FPRT —	FIRST PRIORITY
	R16 (10_{16})	MCTL —	MODE CONTROL
	R17 (11_{16})	OCASN —	OUTPUT CLOCK ASSIGNMENT
	R18 (12_{16})	CIPTR —	COUNTER INTERRUPT POINTER
	R19 (13_{16})	PDAT —	PORT DATA
	R20 (14_{16})	IPS —	INTERRUPT/PORT SELECT
	R21 (15_{16})	PDIR —	PORT DIRECTION
	R22 (16_{16})	CCTL —	COUNTER CONTROL
	R23 (17_{16})	CICTL —	COUNTER INTERRUPT CONTROL
R25 (19_{16})	R24 (18_{16})	LCSV —	L-COUNTER STARTING VALUE
R27 ($1B_{16}$)	R26 ($1A_{16}$)	HCSV —	H-COUNTER STARTING VALUE
R29 ($1D_{16}$)	R28 ($1C_{16}$)	LCCV —	L-COUNTER CURRENT VALUE
R31 ($1F_{16}$)	R30 ($1E_{16}$)	HCCV —	H-COUNTER CURRENT VALUE

FIGURE 3-1. ICU Internal Registers

3.0 Architectural Description (Continued)

The BBBB field is the bias which is programmed by writing $BBBB0000_2$ to the SVCT register (R1). The VVVV field identifies one of the 16 interrupt positions. The contents of the HVCT register provide various information to the CPU, as shown in *Figure 3-2*.

Note 1: The ICU always interprets a read of the HVCT register as either an INTA or RETI cycle. Since these cycles cause internal changes to the ICU, normal programs must never read the ICU HVCT register.

Note 2: If the HVCT register is read with ST1 = 0 (INTA cycle) and no unmasked interrupt is pending, the binary value BBBB1111 is returned and any pending edge-triggered interrupt in position 15 is cleared.

If the auto-rotate priority mode is selected, the FPRT register is also cleared, thus preventing any interrupt from being acknowledged. In this case a re-initialization of the FPRT register is required for the ICU to acknowledge interrupts again.

If a read of the HVCT register is performed with ST1 = 1 (RETI cycle), the binary value BBBB1111 is returned.

If the auto-rotate mode is selected, a priority rotation is also performed.

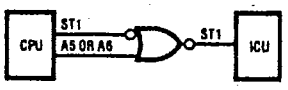
3.2 SVCT — SOFTWARE VECTOR REGISTER (R1)

The SVCT register is a copy of the HVCT register. It allows the programmer to read the contents of the HVCT register without initiating a INTA or RETI cycle in the ICU. It also allows a programmer to change the BBBB field of the HVCT register. The bit map of the SVCT register is the same as for the HVCT register.

During a write to SVCT, the four least significant bits are unaffected while the four most significant bits are written into both SVCT and HVCT (R1 and R0).

The SVCT register is updated dynamically by the ICU. The four least significant bits always contain the vector value that would be returned to the CPU if a INTA or RETI cycle were executed. Therefore, when reading the SVCT register, the state of the CPU ST1 pin is used to select either pending interrupt data or in-service interrupt data. For example, if the SVCT register is read with ST1 = 0 (as for an INTA cycle), the VVVV field contains the encoded value of the highest priority pending interrupt. On the other hand, if the SVCT register is read with ST1 = 1, the VVVV field contains the encoded value of the highest priority in-service interrupt.

Note: If the CPU ST1 output is connected directly to the ICU ST1 input, the vector read from SVCT is always the RETI vector. If both the INTA and RETI vectors are desired, additional logic must be added to drive the ICU ST1 input. A typical circuit is shown below. In this circuit, the state of the ICU ST1 input is controlled by both the CPU ST1 output and the selected address bit.



TL/EE/6117-14

3.3 ELTG — EDGE/LEVEL TRIGGERING REGISTERS (R2, R3)

The ELTG registers determine the input trigger mode for each of the 16 interrupt inputs. Each input is assigned a bit in this register pair. An interrupt input is level-triggered if its bit in ELTG is set to 1. The input is edge-triggered if its bit is cleared. At reset, all bits in ELTG are set to 1.

If odd-numbered interrupt positions must be used for software interrupts, the edge triggering mode must be selected and the corresponding interrupt inputs should be prevented from changing state.

3.4 TPL — TRIGGERING POLARITY REGISTERS (R4, R5)

The TPL registers determine the polarity of either the active level or the active edge for each of the 16 interrupt inputs. As with the ELTG registers, each input is assigned a bit. Possible triggering modes for the various combinations of ELTG and TPL bits are shown below.

ELTG BIT	TPL BIT	TRIGGERING MODE
0	0	Falling Edge
0	1	Rising Edge
1	0	Low Level
1	1	High Level

Software interrupt positions are not affected by their TPL bits. At reset, all TPL bits are set to 0.

Note 1: If edged-triggered interrupts are to be handled, the TPL register should be programmed before the ELTG register.

This prevents spurious interrupt requests from being generated during the ICU initialization from edge-triggered interrupt positions.

Note 2: Hardware interrupt inputs connected to cascaded ICUs must have their TPL bits set to 0.

3.5 IPND — INTERRUPT PENDING REGISTERS (R6, R7)

The IPND registers track interrupt requests that are pending but not yet serviced. Each interrupt position is assigned a bit in IPND. When an interrupt is pending, the corresponding bit in IPND is set. The IPND data are used by the ICU to generate interrupts to the CPU. These data are also used in polling operations.

	INTA CYCLE (ST1 = 0)		RETI CYCLE (ST1 = 1)	
	Highest priority pending interrupt is from:		Highest priority in-service interrupt was from:	
BBBB	cascaded ICU	any other source	cascaded ICU	any other source
	1111	programmed bias*	1111	programmed bias*
VVVV	encoded value of the highest priority pending interrupt		encoded value of the highest priority in-service interrupt	

*The Programmed bias for the master ICU must range from 0000 to 0111_2 because the CPU interprets a one in the most significant bit position as a Cascade Table Index indicator for a cascaded ICU.

FIGURE 3-2. HVCT Register Data Coding

3.0 Architectural Description (Continued)

The IPND registers are also used for requesting software interrupts. This is done by writing specially formatted data bytes to either IPND(L) or IPND(H). The formats differ for registers R6 and R7. These formats are shown below:

IPND(L) (R6) — S0000PPP

IPND(H) (R7) — S0001PPP

Where: S = Set (S = 1) or Clear (S = 0)

PPP = is a binary number identifying one of eight bits

Note: The data read from either R6 or R7 are different from that written to the register because the ICU returns the register contents, rather than the formatted byte used to set the register bits.

The ICU automatically clears a set IPND bit when the pending interrupt request is serviced. All pending interrupts in a register can be cleared by writing the pattern 'X1XXXXXX' to it (X = don't care). To avoid conflicts with asynchronous hardware interrupt requests, the IPND registers should be frozen before pending interrupts are cleared. Refer to the Mode Control Register description for details on freezing the IPND registers.

At reset, all IPND bits are set to 0.

Note: The edge sensing mechanism used for hardware interrupts in the NS32202 ICU is a latching device that can be cleared only by acknowledging the interrupt or by changing the trigger mode to level sensing. Therefore, before clearing pending interrupts in the IPND registers, any edge-triggered interrupt inputs must first be switched to the level-triggered mode. This clears the edge-triggered interrupts; the remaining interrupts can then be cleared in the manner described above. This applies to clearing the interrupts only. Edge-triggered interrupts can be set without changing the trigger mode.

3.6 ISRV — INTERRUPT IN-SERVICE REGISTERS (R8, R9)

The ISRV registers track interrupt requests that are currently being serviced. Each interrupt position is assigned a bit in ISRV. When an interrupt request is serviced by the ICU, its corresponding bit is set in the ISRV registers. Before generating an interrupt to the CPU, the ICU checks the ISRV registers to ensure that no higher priority interrupt is currently being serviced.

Each time the CPU executes an RETI instruction, the ICU clears the ISRV bit corresponding to the highest priority interrupt in service. The ISRV registers can also be written into by the CPU. This is done to implement the special mask priority mode.

At reset, the ISRV registers are set to 0.

Note: If the ICU initialization does not follow a hardware reset, the ISRV register should be cleared during initialization by writing zeroes into it.

3.7 IMSK — INTERRUPT MASK REGISTERS (R10, R11)

Each NS32202 interrupt position can be individually masked. A masked interrupt source is not acknowledged by the ICU. The IMSK registers store a mask bit for each of the ICU interrupt positions. If an interrupt position's IMSK bit is set to 1, the position is masked.

The IMSK registers are controlled by the system software. At reset, all IMSK bits are set to 1, disabling all interrupts.

Note: If an interrupt must be masked off, the CPU can do so by setting the corresponding bit in the IMSK register. However, if an interrupt is set pending during the CPU instruction that masks off that interrupt, the CPU may still perform an interrupt acknowledge cycle following that instruction since it might have sampled the $\overline{INT}$ line before the ICU deasserted it. This could cause the ICU to provide an invalid vector. To avoid this problem, the above operation should be performed with the CPU interrupt disabled.

3.8 CSRC — CASCADED SOURCE REGISTERS (R12, R13)

The CSRC registers track any cascaded interrupt positions. Each interrupt position is assigned a bit in the CSRC registers. If an interrupt position's CSRC bit is set, that position is connected to the $\overline{INT}$ output of another NS32202 ICU, i.e., it is a cascaded interrupt.

At reset, the CSRC registers are set to 0.

Note 1: If any cascaded ICU is used, the CSRC register should be cleared during initialization (if the initialization does not follow a hardware reset) by writing zeroes into it. This should be done before setting the bits corresponding to the cascaded interrupt positions. This operation ensures that the 4-bit in-service counters (associated with each interrupt position to keep track of cascaded interrupts) always get cleared when the ICU is re-initialized.

Note 2: Only the Master ICU should have any CSRC bits set. If CSRC bits are set in a cascaded ICU, incorrect operation results.

3.9 FPRT — FIRST PRIORITY REGISTERS (R14, R15)

The FPRT registers track the ICU interrupt position that currently holds first priority. Only one bit of the FPRT registers is set at one time. The set bit indicates the interrupt position with first (highest) priority.

The FPRT registers are automatically updated when the ICU is in the auto-rotate mode. The first priority interrupt can be determined by reading the FPRT registers. This operation returns a 16-bit word with only one bit set. An interrupt position can be assigned first priority by writing a formatted data byte to the FPRT(L) register. The format is shown below:

7	6	5	4	3	2	1	0
X	X	X	X	F	F	F	F

Where: XXXX = Don't Care

FFFF = A binary number from 0 to 15 indicating the interrupt position assigned first priority.

Note: The byte above is written only to the FPRT(L) register. Any data written to FPRT(H) is ignored.

At reset the FFFF field is set to 0, thus giving interrupt position 0 first priority.

3.10 MCTL — MODE CONTROL REGISTER (R16)

The contents of the MCTL set the operating mode of the NS32202 ICU. The MCTL bit map is shown below.

7	6	5	4	3	2	1	0
CFRZ	COUTD	COUTM	CLKM	FRZ	unused	NTAR	T16N8

3.0 Architectural Description (Continued)

CFRZ — Determines whether or not the NS32202 counter readings are frozen. When frozen, the counters continue counting but the LCCV and HCCV registers are not updated. Reading of the true value of LCCV and HCCV is possible only while they are frozen.

CFRZ = 0 => LCCV and HCCV Not Frozen

CFRZ = 1 => LCCV and HCCV Frozen

COUTD — Determines whether the COUT/SCIN pin is an input or an output. COUT/SCIN should be used as an input only for testing purposes. In this case an external sampling clock must be provided otherwise hardware interrupts will not be recognized.

COUTD = 0 => COUT/SCIN is Output

COUTD = 1 => COUT/SCIN is Input

COUTM — When the COUT/SCIN pin is programmed as an output (COUTD=0), this bit determines whether the output signal is in pulsed form or in square wave form.

COUTM = 0 => Square Wave Form

COUTM = 1 => Pulsed Form

CLKM — Used only in the 8-bit Bus Mode. This bit controls the clock wave form on any of the pins G0/IR0, . . . ,G3/IR6 programmed as counter output.

CLKM = 0 => Square Wave Form

CLKM = 1 => Pulsed Form

FRZ — Freeze Bit. In order to allow a synchronous reading of the interrupt pending registers (IPND), their status may be frozen, causing the ICU to ignore incoming requests. This is of special importance if a polling method is used.

FRZ = 0 => IPND Not Frozen

FRZ = 1 => IPND Frozen

NTAR — Determines whether the ICU is in the AUTO-ROTATE or FIXED Priority Mode. In AUTO-ROTATE mode, the interrupt source at the highest priority position, after being serviced, is assigned automatically lowest priority. In this mode, the interrupt in service always has highest priority and nesting of interrupts is therefore inhibited.

NTAR = 0 => Auto-Rotate Mode

NTAR = 1 => Fixed Mode

T16N8 — Controls the data bus mode of operation.

T16N8 = 0 => 8-Bit Bus Mode

T16N8 = 1 => 16-Bit Bus Mode

At reset, all MCTL bits except COUTD, are reset to 0. COUTD is set to 1.

3.11 OCASN — OUTPUT CLOCK ASSIGNMENT REGISTER (R17)

Used only in the 8-bit Bus Mode. The four least significant bits of this register control the output clock assignments on pins G0/IR0, . . . ,G3/IR6. If any of these bits is set to 1, the clock generated by either the H-Counter or the H+L-Counter will be output to the corresponding pin. The four most significant bits of OCASN are not used. At Reset the four least significant bits are set to 0.

Note: The interrupt sensing mechanism on pins G0/IR0, . . . ,G3/IR6 is not disabled when any of these pins is programmed as clock output. Thus, to avoid spurious interrupts, the corresponding bits in register IPS should also be set to zero.

3.12 CIPTR — COUNTER INTERRUPT POINTER REGISTER (R18)

The CIPTR register tracks the assignment of counter outputs to interrupt positions. A bit map of this register is shown below.

7	6	5	4	3	2	1	0
H	H	H	H	L	L	L	L

Where: HHHH = A 4-bit binary number identifying the interrupt position assigned to the H-Counter (or the H+L-counter if the counters are concatenated).

LLLL = A 4-bit binary number identifying the interrupt position assigned to the L-counter.

Note: Assignment of a counter output to an interrupt position also requires control bits to be set in the CICTL register. If a counter output is assigned to an interrupt position, external hardware interrupts at that position are ignored.

At reset, all bits in the CIPTR are set to 1. (This means both counters are assigned to interrupt position 15.)

3.13 PDAT — PORT DATA REGISTER (R19)

Used only in the 8-bit Bus Mode. This register is used to input or output data through any of the pins G0/IR0, . . . ,G7/IR14 programmed as I/O ports by the IPS register. Any pin programmed as an output delivers the data written into PDAT. The input pins ignore it. Reading PDAT provides the logical value of all I/O pins, INPUT and OUTPUT.

3.14 IPS — INTERRUPT/PORT SELECT REGISTER (R20)

Used only in the 8-bit Bus Mode. This register controls the function of the pins G0/IR0, . . . ,G7/IR14. Each of these pins is individually programmed as an I/O port, if the corresponding bit of IPS is 0; as an interrupt source, if the corresponding bit is 1. The assignment of the H-Counter output to G0/IR0, . . . ,G3/IR6 by means of reg. OCASN overrides the assignment to these pins as I/O ports or interrupt inputs.

At Reset, all the IPS bits are set to 1.

Note: Whenever a bit in the IPS register is set to zero, to program the corresponding pin as an I/O port, any pending interrupt on the corresponding interrupt position will be cleared.

3.15 PDIR — PORT DIRECTION REGISTER (R21)

Used only in the 8-bit Bus Mode. This register determines the direction of any of the pins G0/IR0, . . . ,G7/IR14 programmed as I/O ports by the IPS register. A logic 1 indicates an input, while a logic 0 indicates an output.

At Reset, all the PDIR bits are set to 1.

3.16 CCTL — COUNTER CONTROL REGISTER (R22)

The CCTL register controls the operating modes of the counters. A bit map of CCTL is shown below.

7	6	5	4	3	2	1	0
CCON	CFNPS	COUT1	COUT0	CRUNH	CRUNL	CDCRH	CDCRL

CCON — Determines whether the counters are independent or concatenated to form a single 32-bit counter (H+L-Counter). If a 32-bit counter is selected, the bits corresponding to the H-

3.0 Architectural Description (Continued)

Counter will control the H+L-Counter, while the bits corresponding to the L-Counter are not used.

CCON = 0 => Two 16-bit Counters

CCON = 1 => One 32-bit Counter

CFNPS Determines whether the external clock is prescaled or not.

CFNPS = 0 => Clock Prescaled (divided by 4)

CFNPS = 1 => Clock Not Prescaled.

COUT1 & COUT0 These bits are effective only when the COUT/SCIN pin is programmed as an OUTPUT (COUTD bit in reg. MCTL is 0). Their logic levels are decoded to provide different outputs for COUT/SCIN, as detailed in the table below:

COUT1	COUT0	COUT/SCIN Output Signal
0	0	Internal Sampling Oscillator
0	1	Zero Detect Of L-Counter
1	0	Zero Detect Of H-Counter
1	1	Zero Detect Of H+L-Counter*

*If the H- and L-Counters are not concatenated and COUT1/COUT0 are both 1, the COUT/SCIN pin is active when either counter reaches zero.

CRUNH Determines the state of either the H-Counter or the H+L-Counter, depending upon the status of CCON.

CRUNH = 0 => H-Counter or H+L-Counter Halted

CRUNH = 1 => H-Counter or H+L-Counter Running

CRUNL Effective only when CCON = 0. This bit determines whether the L-Counter is running or halted.

CRUNL = 0 => L-Counter Halted

CRUNL = 1 => L-counter Running

CDCRH Effective only when CRUNH =0 (Counter Halted). This bit is the single cycle decrement signal for either the H-Counter or the H+L-Counter.

CDCRH = 0 => No Effect

CDCRH = 1 => Decrement H-Counter or H+L-Counter

CDCRL Effective only when CRUNL = 0 and CCON = 0. This bit is the single cycle decrement signal for the L-Counter.

CDCRL = 0 => No Effect

CDCRL = 1 => Decrement L-Counter

Note: The bits CDCRL and CDCRH are set when a logic 1 is written into them, but, they are automatically cleared after the end of the write operation. This is needed to accomplish the decrement operation. Therefore, these bits always contain 0 when read.

Reset does not affect the CCTL bits.

3.17 CICTL — COUNTER INTERRUPT CONTROL REGISTER (R23)

The CICTL register controls the counter interrupts and records counter interrupt status. Interrupts can be generated from either of the 16-bit counters. When the counters are concatenated, the interrupt control is through the H-Counter control bits. In this case the CIEL bit should be set to zero to avoid spurious interrupts from the L-Counter. A bit map of the CICTL register is shown following.

7	6	5	4	3	2	1	0
CERH	CIRH	CIEH	WENH	CERL	CIRL	CIEL	WENL

CERH H-Counter Error Flag. This bit is set (1) when a second interrupt request from the H-Counter (or H+L-Counter) occurs before the first request is acknowledged.

CIRH H-Counter Interrupt Request. It is set (1) when an interrupt is pending from the H-Counter (or H+L-Counter). It is automatically reset when the interrupt is acknowledged.

CIEH H-Counter Interrupt Enable. When it is set, the H-Counter (or H+L-Counter) interrupt is enabled.

WENH H-Counter Control Write Enable. When WEHN is set (1), bits CERH, CIRH, and CIEH can be written.

CERL L-Counter Error Flag. This bit is set (1) when a second interrupt request from the L-Counter occurs before the first request is acknowledged.

CIRL L-Counter Interrupt Request. It is set (1) when an interrupt is pending from the L-Counter. It is automatically reset when the interrupt is acknowledged.

CIEL L-Counter Interrupt Enable. When it is set (1), the L-Counter interrupt is enabled.

WENL L-Counter Control Write Enable. When WENL is set (1), bits CERL, CIRL, and CIEL can be written.

Note: Setting the write enable bits (WENH or WENL) and writing any of the other CICTL bits are concurrent operations. That is, the ICU will ignore any attempt to alter CICTL bits if the proper write enable bit is not set in the data byte.

At reset, all CICTL bits are set to 0. However, if the counters are running, the bits CIRL, CERL, CIRH and CERH may be set again after the reset signal is removed.

3.18 LCSV/HCSV — L-COUNTER STARTING VALUE/ H-COUNTER STARTING VALUE REGISTERS (R24, R25, R26, AND R27)

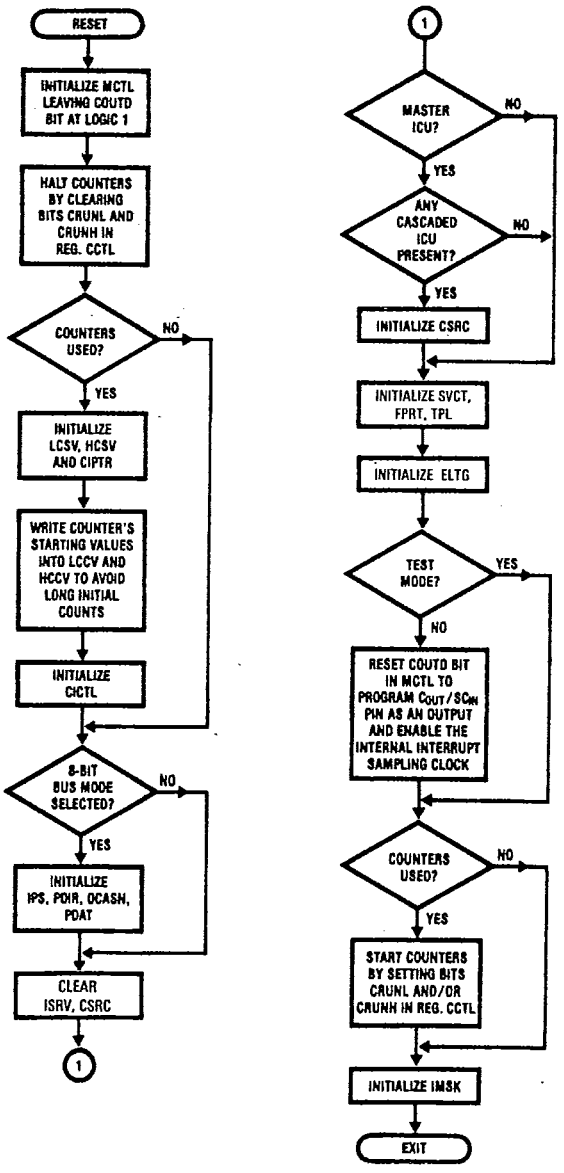
The LCSV and HCSV registers store the start values for the L-Counter and H-Counter, respectively. Each time a counter reaches zero, the start value is automatically reloaded from either LCSV or HCSV, one clock cycle after zero count is reached. Loading LCSV or HCSV from the CPU must be synchronized to avoid writing the registers while the reloading of the counters is occurring. One method is to halt the counters while the registers are loaded.

When the 16-bit counters are concatenated, the LCSV and HCSV registers hold the 32-bit start count, with the least significant byte in R24 and the most significant byte in R27.

3.19 LCCV/HCCV — L-COUNTER CURRENT VALUE/ H-COUNTER CURRENT VALUE REGISTERS (R28, R29, R30, AND R31)

The LCCV and HCCV registers hold the current value of the counters. If the CFRZ bit in the MCTL register is reset (0), these registers are updated on each clock cycle with the current value of the counters. LCCV and HCCV can be read only when the counter readings are frozen (CFRZ bit in the

3.0 Architectural Description (Continued)



TL/EE/5117–15

FIGURE 3–3. Recommended ICU's Initialization Sequence

3.0 Architectural Description (Continued)

MCTL register is 1). They can be written only when the counters are halted (CRUNL and/or CRUNH bits in the CCTL register are 0). This last feature allows new initial count values to be loaded immediately into the counters, and can be used during initialization to avoid long initial counts.

When the 16-bit counters are concatenated, the LCCV and HCCV registers hold the 32-bit current value, with the least significant byte in R28 and the most significant byte in R31.

3.20 REGISTER INITIALIZATION

Figure 3–3 shows a recommended initialization procedure for the ICU that sets up all the ICU registers for proper operation.

4.0 Device Specifications

4.1 NS32202 PIN DESCRIPTIONS

4.1.1 Power Supply

Power (V_{CC}): +5V DC Supply

Ground (GND): Power Supply Return

4.1.2 Input Signals

Reset ($\overline{RST}$): Active low. This signal initializes the ICU. (The ICU initializes to the 8-bit bus mode.)

Chip Select ($\overline{CS}$): Active low. This signal enables the ICU to respond to address, data, and control signals from the CPU.

Addresses (A0 through A4): Address lines used to select the ICU internal registers for read/write operations.

High Byte Enable ($\overline{HBE}$): Active low. Enables data transfers on the most-significant byte of the Data Bus. If the ICU is in the 8-bit Bus Mode, this signal is not used and should be connected to either GND or V_{CC}.

Read ($\overline{RD}$): Active low. Enables data to be read from the ICU's internal registers.

Write ($\overline{WR}$): Active low. Enables data to be written into the ICU's internal registers.

Status (ST1): Status signal from the CPU. When the Hardware Vector Register is read, this signal differentiates an INTA cycle from an RETI cycle. If ST1 = 0 the ICU initiates an INTA cycle. If ST1 = 1 an RETI cycle will result.

Interrupt Requests (IR1, IR3..., IR15): These eight inputs are used for hardware interrupts. Each may be individually triggered in one of four modes: Rising Edge, Falling Edge, Low Level, or High Level.

Counter Clock (CLK): External clock signal to drive the ICU internal counters.

4.1.3 Output Signals

Interrupt Output ($\overline{INT}$): Active low. This signal indicates that an interrupt is pending.

4.1.4 Input/Output Signals

Data Bus 0–7 (D0 through D7): Eight low-order data bus lines used in both 8-bit and 16-bit bus modes.

General Purpose I/O Lines (G0/IR0, G1/IR2,...,G7/IR14): These pins are the high-order data bits when the ICU is in the 16-bit bus mode. When the ICU is in the 8-bit bus mode, each of these can be individually assigned one of the following functions:

- Additional Hardware Interrupt Input (IR0 through IR14)
- General Purpose Data Input
- General Purpose Data Output
- Clock Output from H-Counter (Pins G0/IR0 through G3/IR6 only)

It should be noted that, for maximum flexibility in assigning interrupt priorities, the interrupt positions corresponding to pins G0/IR0,...,G7/IR14 and IR1,...,IR15 are interleaved.

Counter or Oscillator Output/Sampling Clock Input (COUT/SCIN): As an output, this pin provides either a clock signal generated by the ICU internal oscillator, or a zero detect signal from one or both of the ICU counters. As an input, it is used for an external clock, to override the internal oscillator used for interrupt sampling. This is done only for testing purposes.

4.0 Device Specifications (Continued)

4.2 ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to GND	−0.5V to +7.0V
Power Dissipation	1.5 Watt

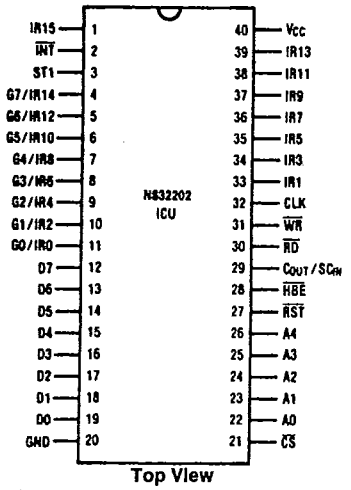
Note: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under Electrical Characteristics.*

4.3 ELECTRICAL CHARACTERISTICS

T_A = 0° to 70°C, V_{CC} = +5V ± 5%, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage				0.8	V
V_{IH}	Input High Voltage		2.0			V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400 μA	2.4			V
I_L	Leakage Current (Output and I/O Pins in TRI-STATE/Input mode)	0.4 ≤ V_{IN} ≤ V_{CC}	−20		20	μA
I_I	Input Load Current	V_{in} = 0 to V_{CC}	−20		20	μA
I_{CC}	Power Supply Current	I_{out} = 0, T = 0°C			300	mA

Connection Diagram



TL/EE/5117–3

Top View

Order Number NS32202D-6, NS32202D-10
See NS Package Number D40C

FIGURE 4–1

4

4.0 Device Specifications (Continued)

T-52-33-13

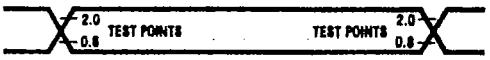
4.4 SWITCHING CHARACTERISTICS

4.4.1 Definitions

All the timing specifications given in this section refer to 0.8V or 2.0V on the input and output signals as illustrated in *Figure 1*, unless specifically stated otherwise.

Abbreviations:

L.E.—leading edge R.E.—rising edge
T.E.—trailing edge F.E.—falling edge



TL/EE/5117–16

FIGURE 4–2. Timing Specification Standard

4.4.1.1 Timing Tables

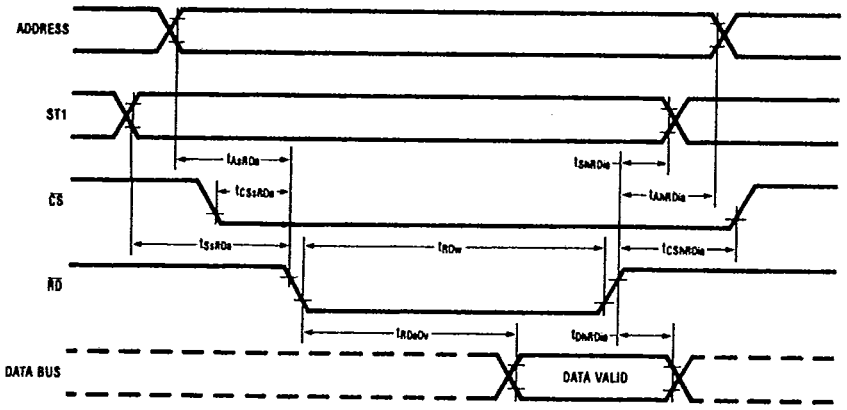
Symbol	Figure	Description	Reference/Conditions	NS32202-10		Units
				Min	Max	
READ CYCLE						
t_{AhRDia}	4-3	Address Hold Time	After $\overline{RD}$ T.E.	10		ns
t_{AsRDa}	4-3	Address Setup Time	Before $\overline{RD}$ L.E.	35		ns
$t_{CShRDia}$	4-3	$\overline{CS}$ Hold Time	After $\overline{RD}$ T.E.	15		ns
t_{CSsRDa}	4-3	$\overline{CS}$ Setup Time	Before $\overline{RD}$ L.E.	30		ns
t_{DhRDia}	4-3	Data Hold Time	After $\overline{RD}$ T.E.	5	50	ns
t_{RDaDv}	4-3	Data Valid	After $\overline{RD}$ L.E.		150	ns
t_{RDw}	4-3	$\overline{RD}$ Pulse Width	At 0.8V (Both Edges)	160		ns
t_{SsRDa}	4-3	ST1 Setup Time	Before $\overline{RD}$ L.E.	35		ns
t_{ShRDia}	4-3	ST1 Hold Time	After $\overline{RD}$ T.E.	−30		ns
WRITE CYCLE						
t_{AhWRia}	4-4	Address Hold Time	After $\overline{WR}$ T.E.	10		ns
t_{AsWRa}	4-4	Address Setup Time	Before $\overline{WR}$ L.E.	35		ns
$t_{CShWRia}$	4-4	$\overline{CS}$ Hold Time	After $\overline{WR}$ T.E.	15		ns
t_{CSsWRa}	4-4	$\overline{CS}$ Setup Time	Before $\overline{WR}$ L.E.	30		ns
t_{DhWRia}	4-4	Data Hold Time	After $\overline{WR}$ T.E.	10		ns
t_{DsWRia}	4-4	Data Setup Time	Before $\overline{WR}$ T.E.	70		ns
t_{WRiaPf}	4-4	Port Output Floating	After $\overline{WR}$ T.E. (To PDIR)		200	ns
t_{WRiaPv}	4-4	Port Output Valid	After $\overline{WR}$ T.E.		200	ns
t_{WRw}	4-4	$\overline{WR}$ Pulse Width	At 0.8V (Both Edges)	160		ns

4.0 Device Specifications (Continued)

4.4.1.1 Timing Tables (Continued)

Symbol	Figure	Description	Reference/Conditions	NS32202-10		Units
				Min	Max	
OTHER TIMINGS						
t_{COUTl}	4-8	Internal Sampling Clock Low Time	At 0.8V (Both Edges)	50		ns
t_{COUTp}	4-8	Internal Sampling Clock Period		400		ns
t_{SCINh}	4-7	External Sampling Clock High Time	At 2.0V (Both Edges)	100		ns
t_{SCINl}	4-7	External Sampling Clock Low Time	At 0.8V (Both Edges)	100		ns
t_{SCINp}	4-7	External Sampling Clock Period		800		ns
t_{Ch}	4-9	External Clock High Time (Without Prescaler)	At 2.0V (Both Edges)	100		ns
t_{Chp}	4-9	External Clock High Time (With Prescaler)	At 2.0V (Both Edges)	40		ns
t_{Cl}	4-9	External Clock Low Time (Without Prescaler)	At 0.8V (Both Edges)	100		ns
t_{Clp}	4-9	External Clock Low Time (With Prescaler)	At 0.8V (Both Edges)	40		ns
t_{Cy}	4-9	External Clock Period (Without Prescaler)		400		ns
t_{Cyp}	4-9	External Clock Period (With Prescaler)		100		ns
t_{GCOUTl}	4-9	Counter Output Transition Delay	After CLK F.E.		300	ns
t_{COUTw}	4-9	Counter Output Pulse Width in Pulsed Form	At 0.8V (Both Edges)	50		ns
t_{ACKIR}	4-5	Interrupt Request Delay	After Previous Interrupt Acknowledge	500		ns
t_{IRld}	4-5	$\overline{INT}$ Output Delay	After Interrupt Request Active		800	ns
t_{IRw}	4-5	Interrupt Request Pulse Width in Edge Trigger	At 0.8V (Both Edges)	50		ns
t_{RSTw}		$\overline{RST}$ Pulse Width	At 0.8V (Both Edges)	400		ns

4.4.1.2 Timing Diagrams

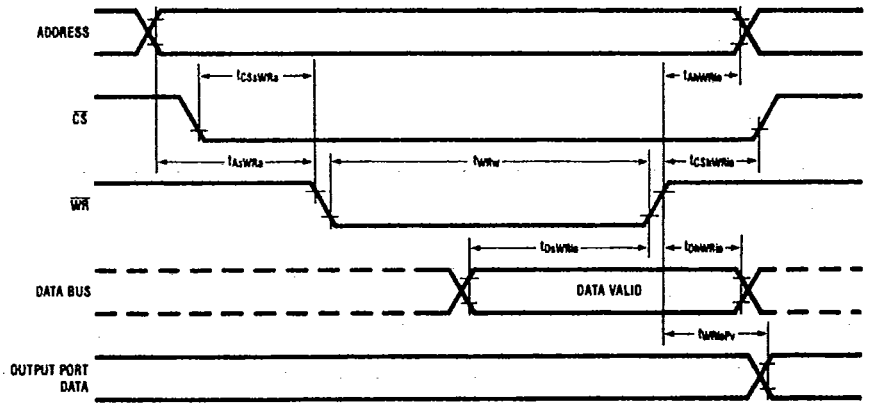


TL/EE/5117-17

FIGURE 4-3. READ/INTA Cycle

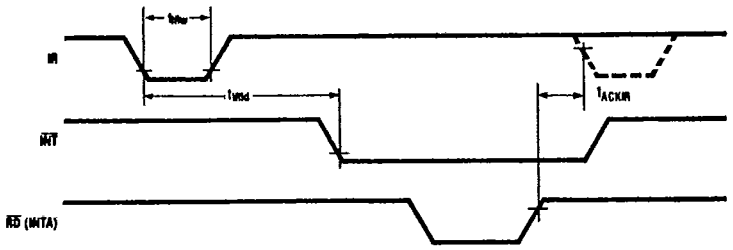
T-52-33-13

4.0 Device Specifications (Continued)



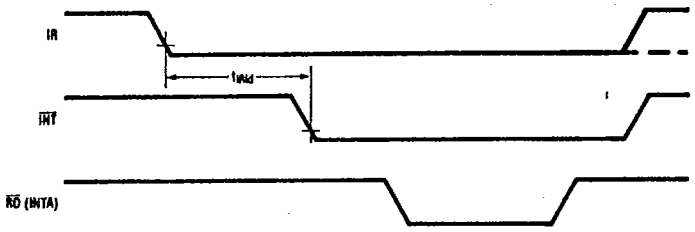
TL/EE/5117–18

FIGURE 4–4. Write Cycle



TL/EE/5117–19

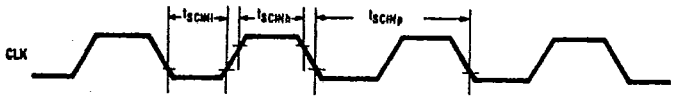
FIGURE 4–5. Interrupt Timing in Edge Triggering Mode



TL/EE/5117–20

FIGURE 4–6. Interrupt Timing in Level Triggering Mode

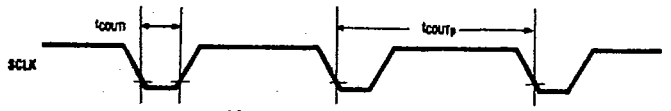
4.0 Device Specifications (Continued)



TL/EE/5117-21

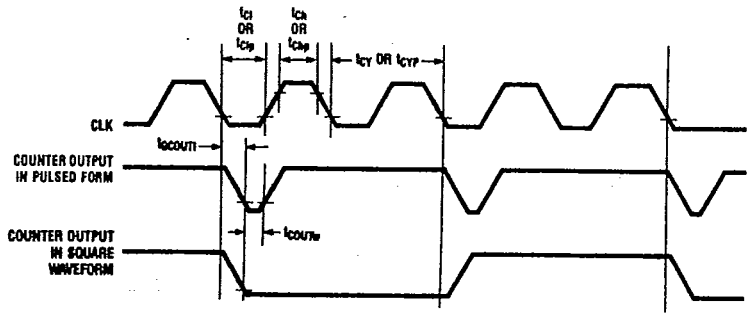
Note: Interrupts are sampled on the rising edge of CLK.

FIGURE 4-7. External Interrupt-Sampling-Clock to be Provided at Pin COUT/SCIN When in Test Mode



TL/EE/5117-22

FIGURE 4-8. Internal Interrupt-Sampling-Clock Provided at Pin COUT/SCIN



TL/EE/5117-23

FIGURE 4-9. Relationship Between Clock Input at Pin CLK and Counter Output Signals at Pins COUT/SCIN or G0/R0,...,G3/R6, in Both Pulsed Form and Square Waveform