AP-730

APPLICATION NOTE

Interfacing the 82C59A to Intel 186 Family Processors

Sean Kohler
Hillarie Klempner
Art Prateepvanich
Aquiles Saenz
Application Engineers

Intel Corporation
5000 West Chandler Boulevard
Chandler, AZ 85226

March 5, 1996

Order Number: 272822-001

Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products.

Intel retains the right to make changes to these specifications at any time, without notice. Microprocessor products may have minor variations to this specification known as errata.

*Other brands and names are the property of their respective owners.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-548-4725

COPYRIGHT © INTEL CORPORATION, 1996

Interfacing the 82C59A to Intel 186 Family Processors

1.0 INTRODUCTION

Many engineers have found that their applications require more external interrupt requests than are provided on the Intel 186 family of embedded processors. Intel P82C59A-2 programmable interrupt controllers (PIC) can be added to a design to increase the number of available interrupt requests; this application note explains how.

This application note contains the following:

- A reference design
- An analysis of the timings for the reference design and design considerations related to this timing data
- Design considerations for cascading 82C59As
- Source code for the programs used to test the reference design and design considerations

2.0 HARDWARE DESCRIPTION

Intel's 80C186XL, 80C186EA, and 80C186EB processors have integrated interrupt controllers which provide a maximum of seven interrupt request lines. Only four lines (80C186XL and 80C186EA processors) or three lines (80C186EB processor) can be used for external requests. The 80C186EC processor has two integrated 82C59A compatible interrupt controllers providing 15 maskable interrupt request lines (8 external). To increase the number of external requests, you must add an external programmable interrupt controller, such as an Intel P82C59A-2.

The 82C59A is an industry standard interrupt controller which provides eight interrupt request sources. In an interrupt driven environment, the 82C59A functions as a manager that accepts requests from various sources and then prioritizes them for the processor. The 82C59A has built-in features for expandability with other 82C59As. This allows up to 64 levels of interrupt priorities.

The 82C59A has one address line and seven addressable registers. Addressing seven registers through one address line is achieved by a unique sequential programming method that must be used to initialize the chip.

Though the 82C59A is fully compatible with 80186 microprocessors, there are design considerations when interfacing 82C59As to today's high speed 186s. These design considerations are discussed later in this document.

2.1 Interface Description

The 82C59A interface used in this reference design is shown in Figure 1. The 82C59A is added as a secondary master interrupt controller to the 25-MHz 80C186XL processor. Connecting SP#/EN# to V_{CC} places the 82C59A in non-buffered master mode. Connecting the latched address 1 signal (LA1) of the processor to the A0 input of the 82C59A makes interfacing the 82C59A possible without any external data steering logic. The other connections are self explanatory.

This application note is a template designers can use to build an 82C59A based system using a 186 family processor. The reference design is specific to the 80C186XL processor, but the principles described can also be used with 80C186EA, 80C186EB and 80C186EC processors, with these exceptions:

80C186EA processor

- Some AC specifications are slightly different.

80C186EB processor

- Some AC specifications are different, and there are differences when programming the chip select unit.
- The ARDY pin used in the reference design has a similar function as the READY pin of the 80C186EB processor but setup and hold timings may be different.

80C186EC processor

- The 80C186EC processor has two integrated 82C59As. For information on cascading additional external 82C59As, refer to the *80C186EC/80C188EC Microprocessor User's Manual*.

Refer to the 80C186EA, 80C186EB and 80C186EC processor datasheets and user's manuals (listed in Table 5) for more information.

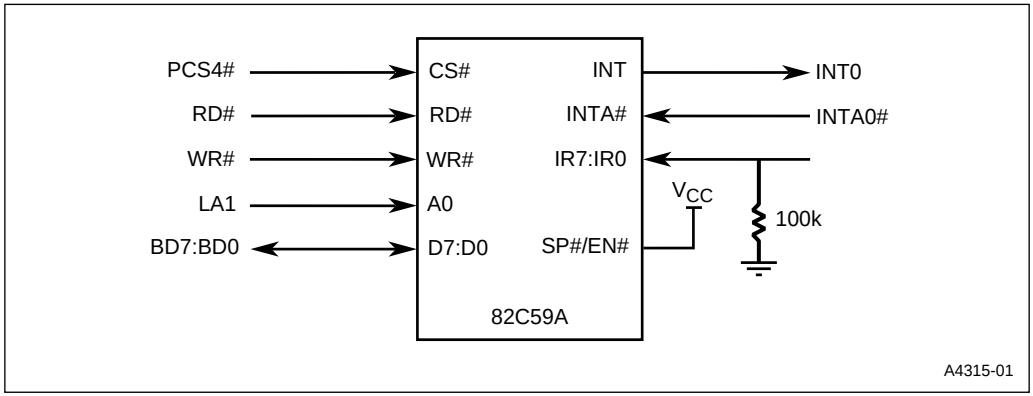

Figure 1. 82C59A Reference Design

3.0 TIMING ANALYSIS

The calculations presented in this section were made using an Intel N80C186XL-25 processor and an Intel P82C59A-2. The processor was run at 25 MHz.

3.1 Read Timing Analysis

Table 1 provides a comparison of critical read cycle timings of the interrupt control unit and the processor.

3.1.1 TOE Read Low to Data Valid

Table 1 illustrates that Read low to data valid (TOE) is violated. After RD# goes active, the 82C59A takes a maximum of 120 ns for its data to become valid. This is an issue because the 80C186XL processor is expecting valid data 52 ns after RD# goes active. This is solved by adding at least two wait states to the 80C186XL processor's read bus cycle, which adds 80 ns because the processor is running at 25 MHz. With two wait states added, the 82C59A can satisfy the requirement of the processor because the processor expects the data to be valid at 132 ns.

Table 1. Read Cycle Wait State Analysis

Memory Device Parameter	Description	Processor Equation	82C59A Equation	Processor Value	82C59A Value	Wait States
TOE	Read low to data valid	$2T_{CLCL}$-T_{CLRL}-T_{DVCL}	T_{ALDV}	52 ns	120 ns	2
TACC	Address valid to data valid	$3T_{CLCL}$-T_{CLAV}-T_{ADLTCH}-T_{DVCL} (T_{ADLTCH} ~10ns)	T_{AHDV}	82 ns	200 ns	3
TCE	Chip enable to data valid	$3T_{CLCL}$-T_{CLCSV}-T_{DVCL}	T_{AHDV}	92 ns	200 ns	3
TDF	Read high to address valid	T_{RHAV}	T_{RDHZ}	25 ns	85 ns	See section 3.1.4

3.1.2 TACC Address Valid to Data Valid

Valid data can be presented from the 82C59A up to 200 ns after the address presented by the processor's address latch becomes valid. The processor is expecting the data to be valid a maximum of 82 ns after the address becomes valid. Inserting three wait states for the read cycle ensures that the processor does not expect data before it is guaranteed to be present on the bus.

3.1.3 TCE Chip Enable to Data Valid

Running the 80C186XL processor at 25 MHz violates the chip select to data valid timing of the 82C59A. After the chip select becomes active, valid data is presented a maximum of 200 ns later from the 82C59A. This poses a problem because the 80C186XL processor is expecting valid data 92 ns after the chip select becomes active, not 200 ns which the 82C59A provides. To solve this issue, three wait states must be added to extend the cycle by 120 ns. With the wait states inserted, the processor expects valid data 212 ns after the chip select goes active.

3.1.4 TDF Read High to Address Valid

The 82C59A can present data on the bus up to 85 ns after the RD# signal goes high. The processor can drive the next address onto the bus 25 ns after RD# goes high. This may cause bus contention. Since this occurs in T4 of the bus cycle, T_{RHAV} cannot be lengthened by adding wait states (other than by slowing the clock rate). To solve this problem the data bus must be buffered.

3.1.5 Read Summary

A three wait state read cycle and a buffered data bus are required when interfacing the 80C186XL processor at 25 MHz to the 82C59A interrupt control unit. Program the chip-select unit to insert wait states on read bus cycles. Refer to the user's manual for your 186 family processor for an explanation on how to insert wait states. A three wait state read cycle and a data buffer solves all of the above violations and provides a successful read cycle from the 82C59A to the processor.

3.2 Write Timing Analysis

Table 2 provides a comparison of critical write cycle timings of the interrupt control unit and the processor.

Table 2. Write Cycle Wait State Analysis

Memory Device Parameter	Description	Processor Equation	82C59A Equation	Processor Value	82C59A Value	Wait States
TWC	Write cycle time	$4T_{CLCL}$	T_{WHWL}	160 ns	190 ns	1
TAW	Address valid to write high	$3T_{CLCL}$-T_{ADLTCH} (T_{ADLTCH} ~10ns)	T_{AHWL}+T_{WLWH}	110 ns	190 ns	2
TCW	Chip enable to write high	$3T_{CLCL}$	T_{AHWL}+T_{WLWH}	120 ns	190 ns	2
TWR	Write recover time	T_{WHLH}	T_{WHAX}	1 ns	0 ns	0
TDW	Data valid to write high	$2T_{CLCL}$	T_{DVWH}	80 ns	160 ns	2
TDH	Data hold from write high	T_{WHDX}	T_{WHDX}	30 ns	0 ns	0
TWP	Write pulse width	T_{WLWH}	T_{WLWH}	65 ns	190 ns	3

3.2.1 TWC Write Cycle Time

The write cycle time for the 80C186XL processor is four periods, making the write cycle time 160 ns at 25 MHz. The 82C59A has a write cycle time of 190 ns. Adding one wait state to the processor's write bus cycle ensures that setup and hold times are met.

3.2.2 TAW Address Valid to Write High

Table 2 indicates that Address valid to write high is violated. After the address becomes valid, the 80C186XL processor can deassert WR# in 110 ns. The 82C59A takes a minimum of 190 ns for it to read valid data after the address is valid. This is solved by adding at least two wait states (80 ns) to the 80C186XL processor write bus cycle.

3.2.3 TCW Chip Enable to Write High

Running the 80C186XL processor at 25 MHz violates the chip select to write high timing of the 82C59A. After the chip select becomes active, WR# is high a maximum of 120 ns later. This poses a problem because the 82C59A requires a minimum of 190 ns to read valid data after chip enable is valid. This problem is solved by adding at least two wait states to the 80C186XL processor write bus cycle.

3.2.4 TDW Data Valid to Write High

The 82C59A requires that data is valid 160 ns before WR# goes high. This is a problem because the 80C186XL processor drives WR# high only 80 ns after data becomes valid. The solution is to add two wait states to the write bus cycle.

3.2.5 TWP Write Pulse Width

The 80C186XL processor has a write pulse width of 65 ns. The 82C59A needs a pulse width of 190 ns. This is solved by adding three wait states to the write bus cycle.

3.2.6 Write Summary

The 82C59A has a much slower response time than the 186 processor. In this case, it is necessary to add three wait states to the write cycle of the processor. This ensures that the minimum pulse width of the processor's write signal is no smaller than the pulse width of the 82C59A write signal. Program the chip-select unit to insert wait states on write bus cycles. The user's manual for your 186 family processor provides instructions for inserting wait states.

3.3 Interrupt Acknowledge Timing Analysis

3.3.1 INTA# Active to Data Valid

The 82C59A can present data on the bus 120 ns after the RD# signal goes low. The processor can expect data 95 ns after the RD# signal goes low. This can cause the processor to read invalid data. To prevent this situation, include at least one wait state in the processor's INT/INTA# cycle. (See Table 3.)

3.3.2 INTA# Pulse Width

The processor can broadcast an INTA# pulse width of only 106 ns. This becomes a problem because the 82C59A needs a minimum pulse width of 160 ns. Adding two wait states extends the processor's INTA# pulse width to meet the 82C59A minimum requirement.

3.3.3 Interrupt Acknowledge Summary

Two wait states must be added to the 80C186XL processor INT/INTA# cycle to ensure that the processor and the 82C59A can communicate under all conditions. In this case, two wait states must be added by an external wait state generator, because the 186 processor does not internally support adding wait states to INT/INTA# bus cycles. Figure 2 shows a possible INTA# bus cycle wait state generator. All setup and hold times must be met and the designer must choose appropriate devices.

Table 3. Interrupt Acknowledge Wait State Analysis

Description	Processor Equation	82C59A Equation	Processor Value	82C59A Value	Wait States
INTA# active to data valid	$3T_{CLCL}-T_{CVCTV}-T_{DVCL}$	T_{RLDV}	95 ns	120 ns	1
INTA# pulse width	$3T_{CLCL}-T_{CVCTV}+T_{CVCTX}$	T_{RLRH}	106 ns	160 ns	2

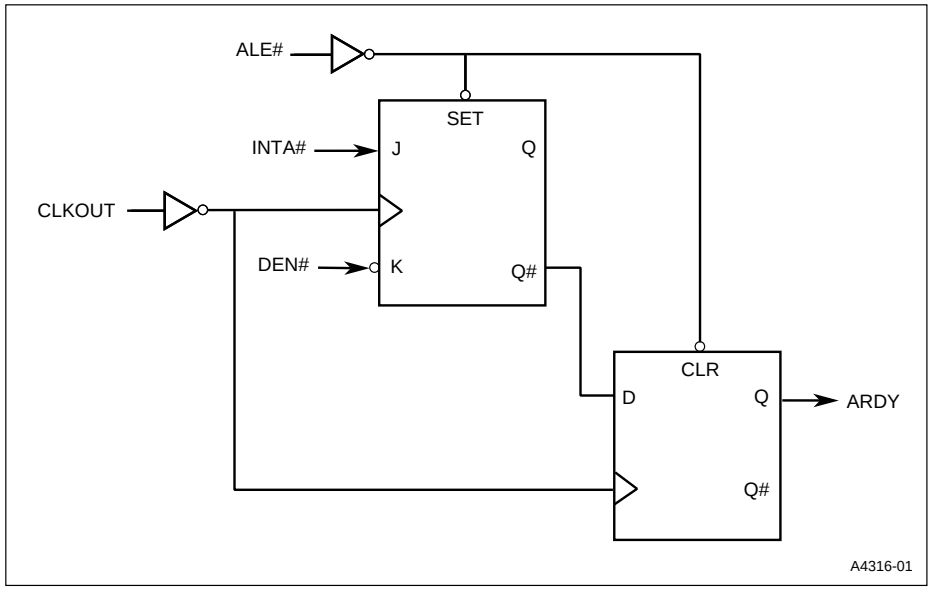

Figure 2. INTA# Cycle Wait State Generator

3.4 Design Considerations for Cascaded 82C59As

Some 186 family microprocessors might not generate interrupt acknowledge bus cycles. These components, listed in Table 4, sense the interrupt asserted on the INT*n* line and acknowledge the interrupt internally, but do not assert INTA*n*#.

Since INTA*n*# is not asserted, the 82C59A does not clear its in-service bit, and does not pass the interrupt vector to the processor. The processor, expecting the 82C59A to place the interrupt type on the data bus, reads erroneous data from the data bus and interprets it as an interrupt vector. This erroneous vector is interpreted as a valid vector, which usually results in a system failure.

Two conditions exist in which components requiring the design consideration exhibit the above mentioned symptoms:

- INTA1# bus cycles are not generated for INT1 assertions when INT1 is configured for Cascade Mode and when an interrupt of higher priority than INT1 occurs before INT1 is acknowledged.
- INTA0# bus cycles are not generated for INT0 assertions when INT0 is configured for Cascade Mode and when an interrupt of higher priority than INT0 occurs before INT0 is acknowledged. Note that after reset, INT0 has a default higher priority than INT1.

Table 4 identifies the processors that require this design consideration.

To avoid the problem:

- Do not use both INT1 and INT0 in cascade mode.
- Make sure that the cascaded interrupt used is configured to be the highest priority interrupt.
- Use 186 processors which do not contain the errata.

There is also a hardware workaround:

1. Weakly pull data bus lines 0 to 7 to a known value (this example uses FFH) to force a defined vector value on the bus when the problem occurs.
2. Include an interrupt service routine in the software to acknowledge the actual interrupt. (See the software example provided in Appendix B.)

3. On the 186 family processor, program cascaded interrupt inputs to be level sensitive.

4. Write a simple service routine for the interrupt type defined in step 1.

5. Write a simple service routine for 82C59A interrupt 7 (only for cases with two external interrupt controllers).

6. Issue non-specific End of Interrupt commands inside INT0 and INT1 service routines (only for cases with two external interrupt controllers).

For a more detailed explanation, refer to FaxBack document #2025 "80C18xXL, 80C18xEA, and 80C18xEB: INTx/INTAx# Errata". (To have this document sent to your fax machine, call 1-800-525-3019.)

3.4.1 Devices Affected

The devices affected may be identified by the ninth character of the alphanumeric Intel FPO number below the product code number. The ninth character is an identifier that relates to the stepping as indicated in Table 4.

Table 4. Devices Requiring the INT*n*/INTA*n*# Design Consideration

Device(1)	Stepping	9th Character
80C186XL/80C188XL	B	A
80C186EA/80C188EA	A	A
80C186EA/80C188EA	B	B
80L186EA/80L188EA	B	B
80C186EB/80C188EB	A	A(2)
80C186EB/80C188EB	B0	B
80C186EB/80C188EB	B1	C
80L186EB/80L188EB	A	A(2)
80L186EB/80L188EB	B0	B
80L186EB/80L188EB	B1	C

NOTES:

1. None of the 80C186/80C188 and 80C186EC/80C188EC processors require the INT*n*/INTA*n*# design consideration.
2. A-Step material may be identified by no ninth character.

4.0 RELATED INFORMATION

To order Intel literature call 1-800-548-4725, or contact:

Intel Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641

Table 5. Related Information

Document Name	Order#
80C186EA/80C188EA Microprocessor User's Manual	270950
80C186EB/80C188EB Microprocessor User's Manual	270830
80C186EC/80C188EC Microprocessor User's Manual	272047
80C186XL/80C188XL Microprocessor User's Manual	272164
Embedded Microprocessors Databook for processor related datasheets	272396
Peripheral Components Databook see the *82C59A-2 CHMOS Programmable Interrupt Controller* Datasheet	296467

APPENDIX A
REFERENCE DESIGN TEST PROGRAM

Example A-1 is a listing of the software used to test this design. This design was tested on a modified 80C186XL/EA Evaluation Board. Any initialization code not shown was performed by the RISM firmware on the evaluation board.

You can download a copy of this file from Intel's application BBS. Using a terminal program and a modem, call 503-264-7999 and respond to the system prompts.

Example A-1. Reference Design Test Program Listing (Sheet 1 of 7)

```
$MOD186
NAME _82C59_INTERFACE

PCB_BASE        EQU     0FF00H

;INTERRUPT CONTROL REGISTERS
IMASK           EQU     PCB_BASE + 028H
I0CON           EQU     PCB_BASE + 038H
MPCS            EQU     PCB_BASE + 0A8H
PACS            EQU     PCB_BASE + 0A4H
LED             EQU     488H            ;ADDRESS OF THE LED
EOI             EQU     0FF22H
MASTER_TYPE     EQU     32              ;BASE TYPE OF THE MASTER ICU
INT0_TYPE       EQU     MASTER_TYPE + 0
INT1_TYPE       EQU     MASTER_TYPE + 1
INT2_TYPE       EQU     MASTER_TYPE + 2
INT3_TYPE       EQU     MASTER_TYPE + 3
INT4_TYPE       EQU     MASTER_TYPE + 4
INT5_TYPE       EQU     MASTER_TYPE + 5
INT6_TYPE       EQU     MASTER_TYPE + 6
INT7_TYPE       EQU     MASTER_TYPE + 7

;PCS BASE ADDRESS AT 3FFH (DETERMINED BY iRISM CODE)

PCS4            EQU     600H    ;LOWER BOUNDARY OF PCS4
EXT_PIC_P0      EQU     PCS4    ;INTERRUPT CONTROL REGISTER
EXT_PIC_P1      EQU     PCS4+2  ;INTERRUPT CONTROL REGISTER
PCSVAL_MSK      EQU     03H     ;CHIP SELECT VALUE WITH 3 WAIT STATES

CODE   SEGMENT AT 0100H
       ASSUME CS:CODE
MAIN:  CLI             ;DISABLE INTERRUPTS
       CALL INITPCS    ;INITIALIZE THE PERIPHERAL CHIP SELECT
       CALL INITVECT   ;INITIALIZE THE INTERRUPT VECTOR TABLE
       CALL INITICU    ;INITIALIZE THE INTERRUPT CONTROL UNIT
       STI             ;ENABLE INTERRUPTS
       JMP  $          ;WAIT FOR INTERRUPT TO OCCUR
```

Example A-1. Reference Design Test Program Listing (Sheet 2 of 7)

```
;*******************************************************************************
;INITIALIZES THE TIMER0 INTERRUPT VECTOR LOCATION

INITVECT  PROC
          XOR AX, AX      ;CLEARS THE ACCUMULATOR
          MOV DS, AX      ;CLEARS THE DATA SEGMENT

;INITIALIZE THE INTERRUPT VECTOR

      MOV DI, INT0_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR0  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR0   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT1_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR1  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR1   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT2_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR2  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR2   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT3_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR3  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR3   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT4_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR4  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR4   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT5_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR5  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR5   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT6_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR6  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR6   ;POINTER TO THE IP LOCATION IN THE TABLE

      MOV DI, INT7_TYPE*4               ;MOVE INTERRUPT VECTOR LOC INTO DI
      MOV WORD PTR DS:[DI], OFFSET ISR7  ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG ISR7   ;POINTER TO THE IP LOCATION IN THE TABLE

      RET

INITVECT       ENDP
```

Example A-1. Reference Design Test Program Listing (Sheet 3 of 7)

```
;*******************************************************************************
ISR0    PROC

;NOTE THAT AN EOI HAS TO BE ISSUED FOR BOTH THE 82C59A-2 AND THE 80C186XL's
;INTERNAL INTERRUPT CONTROL UNIT

      MOV DX, LED              ;TURN ON THE LED
      MOV AL, 0FEH
      OUT DX, AL

      MOV DX, EXT_PIC_P0       ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI              ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                     ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR0           ENDP

;*******************************************************************************
ISR1    PROC

      MOV DX, LED              ;TURN ON THE LED
      MOV AL, 0FDH
      OUT DX, AL

      MOV DX, EXT_PIC_P0       ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI              ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                     ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR1           ENDP
```

Example A-1. Reference Design Test Program Listing (Sheet 4 of 7)

```
;*******************************************************************************
ISR2    PROC

      MOV DX, LED               ;TURN ON THE LED
      MOV AL, 0FBH
      OUT DX, AL

      MOV DX, EXT_PIC_P0        ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI               ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                      ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR2           ENDP

;*******************************************************************************
ISR3    PROC

      MOV DX, LED               ;TURN ON THE LED
      MOV AL, 0F7H
      OUT DX, AL

      MOV DX, EXT_PIC_P0        ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI               ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                      ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR3           ENDP

;*******************************************************************************
ISR4    PROC

      MOV DX, LED               ;TURN ON THE LED
      MOV AL, 0EFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0        ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL
      MOV DX, EOI               ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
```

Example A-1. Reference Design Test Program Listing (Sheet 5 of 7)

```
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR4           ENDP

;*******************************************************************************
ISR5    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0DFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR5           ENDP

;*******************************************************************************
ISR6    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0BFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR6           ENDP
```

Example A-1. Reference Design Test Program Listing (Sheet 6 of 7)

```
;*******************************************************************************
ISR7    PROC

      MOV DX, LED              ;TURN ON THE LED
      MOV AL, 7FH
      OUT DX, AL

      MOV DX, EXT_PIC_P0       ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI              ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                     ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR7          ENDP

;*******************************************************************************
INITICU        PROC

;SETTING UP INTERNAL INTERRUPT CONTROLLER REGISTER FOR CASCADE MODE

      MOV DX, I0CON            ;INITIALIZE INTERRUPT 0 FOR CASCADING
      MOV AX, 0030H            ;SET CASCADE BIT, MAKE INT0 THE HIGHEST PRIORITY
      OUT DX, AX               ;INITIALIZE TO CASCADE MODE

;INITIALIZING THE MASTER CONTROLLER

      MOV DX, EXT_PIC_P0       ;INITIALIZATION COMMAND WORD 1 FOR THE SLAVE
      MOV AL, 13H              ;SET BIT TO INDICATE IT IS THE ONLY 8259
                               ;IN THE SYSTEM
      OUT DX, AL               ;MOVE VALUE

      MOV DX, EXT_PIC_P1       ;INITIALIZATION COMMAND WORD 2 FOR THE MASTER
      MOV AL, MASTER_TYPE      ;SET UP THE MASTER BASE TYPE ---------------
      OUT DX, AL

      MOV DX, EXT_PIC_P1       ;INITIALIZATION COMMAND WORD 4 FOR THE MASTER
      MOV AL, 01H              ;8086 MODE, NO SFNM, NO AEOI, NO BUF
      OUT DX, AL

      MOV DX, EXT_PIC_P1      ;SETTING THE OPERATION COMMAND WORD 1 (MASK REG)
      MOV AL, 00H             ;UNMASK ALL OF THE INTERRUPT ON THE EXTERNAL MASTER
      OUT DX, AL              ;SET THE MASK REGISTER
```

Example A-1. Reference Design Test Program Listing (Sheet 7 of 7)

```
      MOV DX, IMASK             ;UNMASK ONLY INT0
      MOV AX, 00EDH             ;CLEAR ONLY INTERRUPT 0
      OUT DX, AX                ;MASK ALL INTERRUPTS BUT INT0

      RET                      ;RETURN FROM PROCEDURE

INITICU        ENDP

;*******************************************************************************
INITPCS        PROC

      MOV DX, MPCS              ;INITIALIZE THE CHIP SELECT
      IN AX, DX
      OR AX, PCSVAL_MSK         ;CHIP SELECT VALUE WITH 3 WAIT STATES
      OUT DX, AX

      RET                      ;RETURN FROM PROCEDURE

INITPCS        ENDP

;*******************************************************************************
CODE    ENDS
        END MAIN
```

APPENDIX B
DESIGN CONSIDERATION TEST PROGRAM

This program assumes an implementation of the hardware workaround discussed in “Design Considerations for Cascaded 82C59As” on page 5.

You can download a copy of this file from Intel’s application BBS. Using a terminal program and a modem, call 503-264-7999 and respond to the system prompts.

Example B-1. Design Consideration Test Program (Sheet 1 of 7)

```
$MOD186
NAME _82C59_INTERFACE_ERRATA

PCB_BASE        EQU     0FF00H

;INTERRUPT CONTROL REGISTERS
IMASK           EQU     PCB_BASE + 028H
I0CON           EQU     PCB_BASE + 038H
MPCS            EQU     PCB_BASE + 0A8H
PACS            EQU     PCB_BASE + 0A4H

LED             EQU     488H            ;ADDRESS OF THE LED
EOI             EQU     0FF22H
MASTER_TYPE     EQU     32              ;BASE TYPE OF THE MASTER ICU

;PCS BASE ADDRESS AT 3FFH (DETERMINED BY iRISM CODE)

PCS4            EQU     600H    ;LOWER BOUNDARY OF PCS4
EXT_PIC_P0      EQU     PCS4    ;INTERRUPT CONTROL REGISTER
EXT_PIC_P1      EQU     PCS4+2  ;INTERRUPT CONTROL REGISTER
PCSVAL_MSK      EQU     03H     ;CHIP SELECT VALUE WITH 3 WAIT STATES


CODE    SEGMENT AT 0100H
        ASSUME CS:CODE, DS:CODE


MAIN:   CLI             ;DISABLE INTERRUPTS
        CALL INITPCS    ;INITIALIZE THE PERIPHERAL CHIP SELECT
        CALL INITVECT   ;INITIALIZE THE INTERRUPT VECTOR TABLE
        CALL INITICU    ;INITIALIZE THE INTERRUPT CONTROL UNIT
        STI             ;ENABLE INTERRUPTS
        JMP  $
```

Example B-1. Design Consideration Test Program (Sheet 2 of 7)

```
;*******************************************************************************

;INITIALIZES THE TIMER0 INTERRUPT VECTOR LOCATION

INITVECT       PROC

      XOR AX, AX      ;CLEARS THE ACCUMULATOR
      MOV DS, AX      ;CLEARS THE DATA SEGMENT

;INITIALIZE THE INTERRUPT VECTOR

      MOV DI, 255*4                  ;USED FOR HARDWARE WORKAROUND
                                     ;255 BECAUSE WE HAVE PULLUPS ON THE DATABUS
      MOV WORD PTR DS:[DI], OFFSET E_ISR ;POINTER TO THE CS LOCATION IN THE TABLE
      MOV WORD PTR DS:[DI+2], SEG E_ISR  ;POINTER TO THE IP LOCATION IN THE TABLE

      RET

INITVECT       ENDP

;*******************************************************************************
E_ISR   PROC

      MOV DX, EXT_PIC_P0      ;THE OPERATION COMMAND WORD 3
                              ;(POLL SEQUENCE)
      MOV AX, 000CH           ;POLL=1 AND D5:4=01
      OUT DX, AL              ;ISSUE POLL COMMAND
      NOP
      NOP
      NOP
      IN  AL, DX              ;THE 82C59A-2 DEPOSITS THE POLL STATUS BYTE
      AND AL, 07H
      TEST AL, 00H            ;SEE IF HIGHEST PRIORITY (IR0) IS PENDING
      JE  ISR0                ;IF TRUE JUMP TO ISR0
      TEST AL, 01H            ;SEE IF IR1 IS PENDING
      JE  ISR1                ;IF TRUE JUMP TO ISR1
      TEST AL, 02H            ;SEE IF IR2 IS PENDING
      JE  ISR2                ;IF TRUE JUMP TO ISR2
      TEST AL, 03H            ;SEE IF IR3 IS PENDING
      JE  ISR3                ;IF TRUE JUMP TO ISR3
      TEST AL, 04H            ;SEE IF IR4 IS PENDING
      JE  ISR4                ;IF TRUE JUMP TO ISR4
```

Example B-1. Design Consideration Test Program (Sheet 3 of 7)

```
      TEST AL, 05H            ;SEE IF IR5 IS PENDING
      JE  ISR5                ;IF TRUE JUMP TO ISR5
      TEST AL, 06H            ;SEE IF IR6 IS PENDING
      JE  ISR6                ;IF TRUE JUMP TO ISR6

      JMP  ISR7               ;IR7 MUST BE PENDING -> JUMP TO ISR7

      IRET                    ;PRECAUTIONARY IRET (SHOULD NOT BE NEEDED)
E_ISR          ENDP

;*******************************************************************************
ISR0    PROC

;NOTE THAT AN EOI HAS TO BE ISSUED FOR BOTH THE 82C59A-2 AND THE 80C186XL's
;INTERNAL INTERRUPT CONTROL UNIT

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0FEH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ; ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR0           ENDP

;*******************************************************************************
ISR1    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0FDH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL
      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
```

Example B-1. Design Consideration Test Program (Sheet 4 of 7)

```
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR1           ENDP

;*******************************************************************************
ISR2    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0FBH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR2           ENDP

;*******************************************************************************
ISR3    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0F7H
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX
```

Example B-1. Design Consideration Test Program (Sheet 5 of 7)

```
      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR3           ENDP

;*******************************************************************************
ISR4    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0EFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR4           ENDP

;*******************************************************************************
ISR5    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0DFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR5           ENDP
```

Example B-1. Design Consideration Test Program (Sheet 6 of 7)

```
;*******************************************************************************
ISR6    PROC
      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 0BFH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR6          ENDP

;*******************************************************************************
ISR7    PROC

      MOV DX, LED             ;TURN ON THE LED
      MOV AL, 07FH
      OUT DX, AL

      MOV DX, EXT_PIC_P0      ;ISSUE NON-SPECIFIC EOI FOR 82C59A-2
      MOV AL, 20H
      OUT DX, AL

      MOV DX, EOI             ;ISSUE NON-SPECIFIC EOI FOR 80C186XL
      MOV AX, 8000H
      OUT DX, AX

      IRET                    ;RETURN FROM INTERRUPT SERVICE ROUTINE
ISR7          ENDP

;*******************************************************************************
INITICU    PROC

;SETTING UP INTERNAL INTERRUPT CONTROLLER REGISTER FOR CASCADE MODE

      MOV DX, I0CON           ;INITIALIZE INTERRUPT 0 FOR CASCADING
      MOV AX, 0037H           ;SET CASCADE BIT, MAKE INT0 THE LOWEST PRIORITY
```

Example B-1. Design Consideration Test Program (Sheet 7 of 7)

```
      OUT DX, AX              ;INITIALIZE TO CASCADE MODE
;INITIALIZING THE MASTER CONTROLLER

      MOV DX, EXT_PIC_P0      ;INITIALIZATION COMMAND WORD 1 FOR THE SLAVE
      MOV AL, 13H             ;SET BIT TO INDICATE IT IS THE ONLY 8259
                              ;IN THE SYSTEM
      OUT DX, AL              ;MOVE VALUE

      MOV DX, EXT_PIC_P1      ;INITIALIZATION COMMAND WORD 2 FOR THE MASTER
      MOV AL, MASTER_TYPE     ;CAN BE ANY VALUE IN THIS EXAMPLE
      OUT DX, AL

      MOV DX, EXT_PIC_P1      ;INITIALIZATION COMMAND WORD 4 FOR THE MASTER
      MOV AL, 01H             ;8086 MODE, NO SFNM, NO AEOI, NO BUF
      OUT DX, AL

      MOV DX, EXT_PIC_P1      ;SETTING THE OPERATION COMMAND WORD 1 (MASK REG)
      MOV AL, 00H             ;UNMASK ALL OF THE INTERRUPT ON THE EXTERNAL MASTER
      OUT DX, AL              ;SET THE MASK REGISTER

      MOV DX, IMASK           ;UNMASK ONLY INT0
      MOV AX, 00EDH           ;CLEAR ONLY INTERRUPT 0
      OUT DX, AX              ;MASK ALL INTERRUPTS BUT INT0

      RET                     ;RETURN FROM PROCEDURE

INITICU        ENDP

;*******************************************************************************
INITPCS        PROC

      MOV DX, MPCS            ;INITIALIZE THE CHIP SELECT
      IN AX, DX
      OR AX, PCSVAL_MSK       ;CHIP SELECT VALUE WITH 3 WAIT STATES
      OUT DX, AX

      RET                     ;RETURN FROM PROCEDURE

INITPCS        ENDP
;*******************************************************************************

CODE    ENDS
        END MAIN
```