CPI008/6-87

CHIPS AND TECHNOLOGIES, INC.

INTERFACING THE 80386 TO NUMERIC COPROCESSOR USING CHIPS AND TECHNOLOGIES 8230 CHIPSETTM

Rev. 1.2
July 12, 1987

Chips and Technologies, Inc.
521 Cottonwood Drive
Milpitas, California 95035

REVISION SUMMARY BETWEEN REV 1.1. AND 1.2.

1. READY386 output has beem modified to ensure a faster rise time. The READY 386 is no longer generated by the RES387 16L8 PAL. The READYO# from the 387 is delayed by one CLK2 to ensure a 1 wait state operation (The 301 does not support 0 WS non-pipelined accesses).

2. The STEN input to the 387 is not generated by the Q# output of the 74F74 flip-flop. As per Intel's recommendation, the STEN input is strapped high.

3. The PEREQ# output to the 386 has been modified to delay PEREQ to the processor following deactivation of BUSY#.

4. The AF32# from the 306 is not used. Instead, AF32# is generated externally by monitoring A31, MIO#, MALE# and SCLK. The 306 AF32#, fails to output AF32# during the third access of a numeric co-processor instruction (operations requiring three opcode fetches). The 306 AF32# can be used, if one additional wait state is introduced for numeric operation.

INTERFACING THE 80386 TO THE NUMERICS COPROCESSORS

1. INTRODUCTION

Numeric processors extend the register and instruction set of the 80386 microprocessor architecture and adds numeric capability as well as support for floating point, extended integer and BCD data types. The numeric co-processor enhances the overall system performance by executing numeric instructions in parallel with the 80386 microprocessor.

The 80386 communicates with the numeric co-processors through I/O addresses 800000F8H and 800000FCH. The I/O addresses are automatically generated by the 386 when it encounters a numeric instruction. Communication of instructions and data operand transfers between the 80386 and the numeric coprocessor are handled by three control signals:

1. BUSY#, when active, indicates that the coprocessor is busy executing a command.

2. PEREQ# when active, indicates that the numeric coprocessor is ready to initiate data transfer.

3. ERROR# when active, indicates that an unmasked error condition exists.

The 80386 supports two numeric co-processors:

- The 80287 performs 16-bit data transfers and interfaces directly to the 80286 microprocessor. It can interface to the 80386 with additional glue logic.

- The 80387 performs 32-bit data transfers and interfaces directly to the 386 microprocessor. The 80386/80387 interface offers two to three times the performance of the 80386/80287 interface.

This Application-Brief discusses the 386/387 and 386/287 AT compatible hardware interface when using Chips and Technology's CS8230-AT CHIPSet. The first section provides a brief overview of the numeric co-processor implementation on the PC-AT. The section also discusses the differences between the 80387 and the 80287 coprocessor and the differences between the 80386 and the 286 microprocessor when interfacing to numeric co-processors. Those who are familiar with the PC-AT numeric co-processor implementation may wish to skip this section. The second section provides a detailed design description of the 80386/80387 PC-AT compatible interface. The 386/287 interface is discussed in section 3. Detailed schematics and PAL codes have been included in the Appendix.

1.1. PC-AT Numeric Co-processor Hardware Interface Description

The PC-AT uses Intel's 80287 numeric co-processor. The co-processor derives its clock input from the processor clock. This clock is internally divided by three to generate the co-processor clock. The 80287 is accessed as an I/O device at location 00F8H, 00FAH and 00FCH. The communication between the processor and the numeric co-processor is through these I/O addresses.

The 80287 uses the BUSY# output to inform the host processor that it is currently executing a numeric instruction.

In the event of an error during a numeric operation, the BUSY# signal from the numerics is extended (by latching BUSY# during an error condition) to prevent the processor from executing new numeric instruction before the error handling routine is invoked.

The error signal generates a hardware interrupt (INT 13H). The error handler routine is then invoked, which in turn clears the BUSY latch by an 8-bit write to the I/O port address F0H with D0~ D7 equal to zeros. The control is then transferred to the NMI interrupt handler routine.

The 80287, like the 80286 and the 80386 microprocessors, support two modes of operation : the Real Address mode, and the Protected mode. Following power on reset or an I/O write operation to port F1H, the 287 operates in the Real Address mode. The 287 can be programmed to operate in the protected mode by executing an SETPM ESC instruction.

1.2. Differences Between the 387 and the 287 Numeric Co-processor

1. 80287 performs 16-bit data transfers while the 387 performs 32-bit data transfers. The 386 can interface to the 287 with additional logic. However, the 286 supports only the 287.

2. The 387 requires a longer RESET pulse width.

3. The phase of the 387's internal bus controller clock must be the same phase as the 386's internal clock. This enables the 387 to monitor all bus cycles to determine if the 386 is operating in the pipelined or the non-pipelined mode. The 80287 does not monitor CPU bus cycles.

4. The 287 ERROR# output is activated during an unmasked error condition. In the PC-AT implementation, the ERROR# input to the processor is strapped high. When an error occurs during a numeric operation, external logic is used to generate an INT 13H. The interrupt service routine then performs the error processing and reporting.

   The 387 ERROR# output on the otherhand serves two functions:

   a. The 387 activates the ERROR# output when a error occurs of the type not masked by the coprocessor's control register

   b. Following RESET, the ERROR# input is used to inform the 80386 if it is interfacing to the 80387 or the 80287. **If the 386 ERROR# input is held low following the falling edge of RESET until the 386 receives the READY# for the first cycle, the 386 assumes a 387 is installed.** Otherwise a 287 is assumed.

   The current PC-AT implementation has to be modified when using the 387 to allow proper numeric coprocessor interface recognition.

1.3. Differences between the 80386 and the 80286 when interfacing to co-processors

The 80287 supports a PEACK# signal, that in conjunction with the PREQ signal provides the hand-shake mechanism for data transfers between the processor and the numerics interface.

The 80386 does not support a PEACK signal because it knows the exact length of the operand being transferred. After the ESC instruction has been sent to the numeric co-processor, the 80386 processor expansion channel will initiate a data transfer when it receives a PEREQ from the numeric interface.

One direct impact of this operand transfer mechanism in an AT compatible environment is that when an error occurs, the 387/287 will flag the error condition and abort the execution of the instruction. The 386 on the other hand is waiting to complete the requested operand transfers.

To prevent the 386 from idling during the error condition, when ERROR# goes active, the PEREQ (**Intel recommends that PEREQ be activated following the de-activation of BUSY#**) is extended to allow the 386 to complete the operand transfer.

This protocol permits the 386 to complete the operation, even after an error occurs. The data transfer process is automatically terminated by the 386.

2.0. 386/387 PC-AT COMPATIBLE INTERFACE

The previous section described the AT compatible numeric hardware interface requirements and the differences between the 386 and the 286 in terms of co-processor interface. This section will describe the techniques of realizing an AT compatible 386/387 interface.

The 386/387 AT compatible interface circuitry is shown in the Appendix along with the PAL codes. Figure 1.0a illustrates the interface in a block diagram form. A brief circuit description is given below:

2.1. PEREQ, INT13 and BUSY generation circuit

The circuitry required to generate the PEREQ, BUSY, and INT13 is shown in Figure 1.0. The PEREQ and BUSY# is latched on the leading edge of the error signal by a 74F74. The Q (INT13)output of the flip-flop is used to generate the interrupt to the 82C206 Integrated Peripheral Controller. The INT13 output is 'Ored' with the 387 BUSY# to generate BUSY output to the 386. **386 PEREQ is generated by 'Oring' the 387 PEREQ and BUSY# inactive with INT13H and BUSY# inactive. The 387's STEN input must be strapped high.** The 74F74 flip-flop is cleared by either a I/O write to port F0H, an INTA# cycle, or by activating RESET3.

2.2. Reset Generation Logic

In a PC-AT compatible numeric interface, there is a provision to reset the numeric co-processor seperately. Intel, however strongly recommends that the 386 and 387 be reset

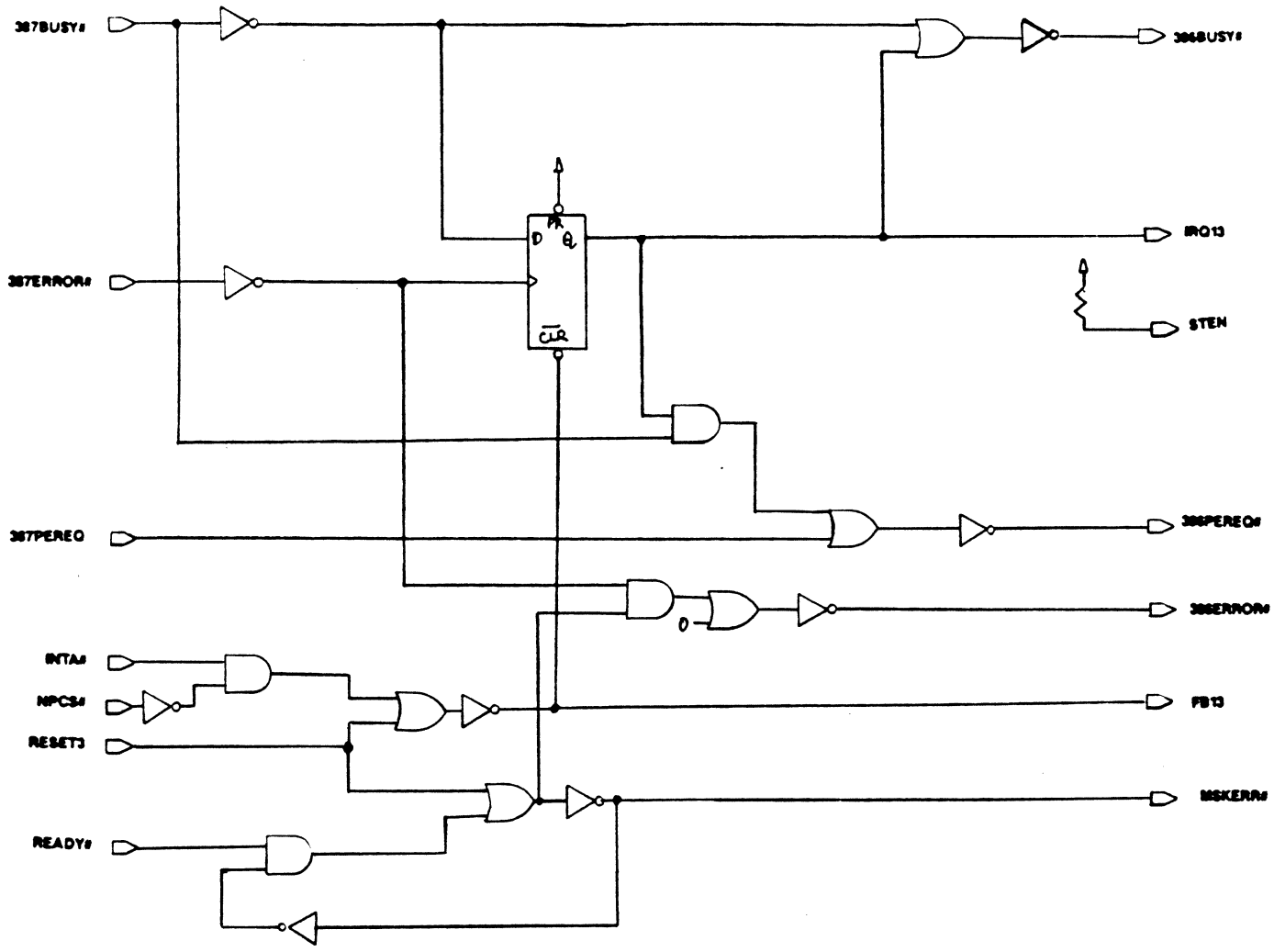


Figure 1.0 BUSY#, PEREG, STEN, IRQB and ERROR# Generation Logic

simultaniously. If you do not intend to support the asynchronous reset feature, then the 386 RESET3 can directly be connected to the 387. If on the otherhand, you intend to support the asynchronous reset feature for compatibility reasons, then additional circuitry is required to support this feature.

The 80387 is reset by performing a write to I/O port F1H with D0~D7 equal to zeros. The CS8230: AT/386 CHIPSet executes a write to this I/O address location as a AT emulation cycle, allowing extension of the cycle with IOCHRDY#.

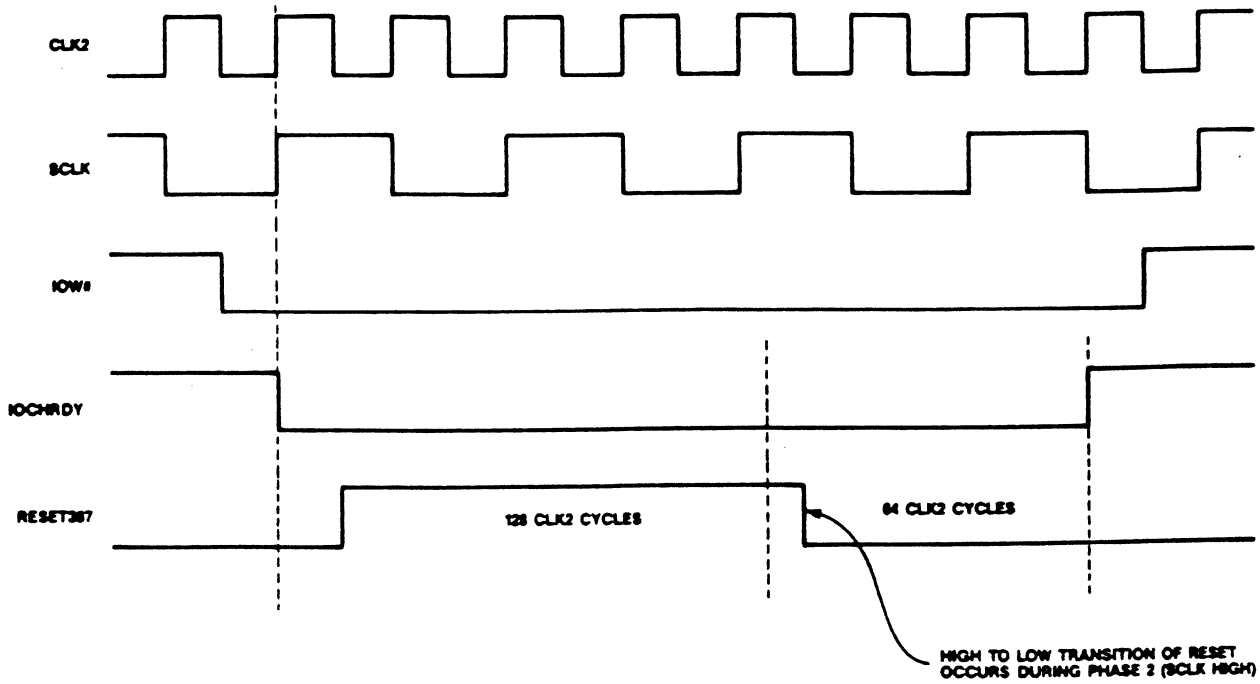


Figure 2.0 Timing Relationship Between IOW#, RESET and IOCHRY

The RESET generation and synchronization logic extends the cycle sufficiently to meet the reset pulse width requirement of the 387 and allow sufficient time for synchronization. The timing relationship between the IOW, RESET and IOCHRDY is shown in Figure 2.0 and the logic required to satisfy the timing relationship is illustrated in Figure 3.0.

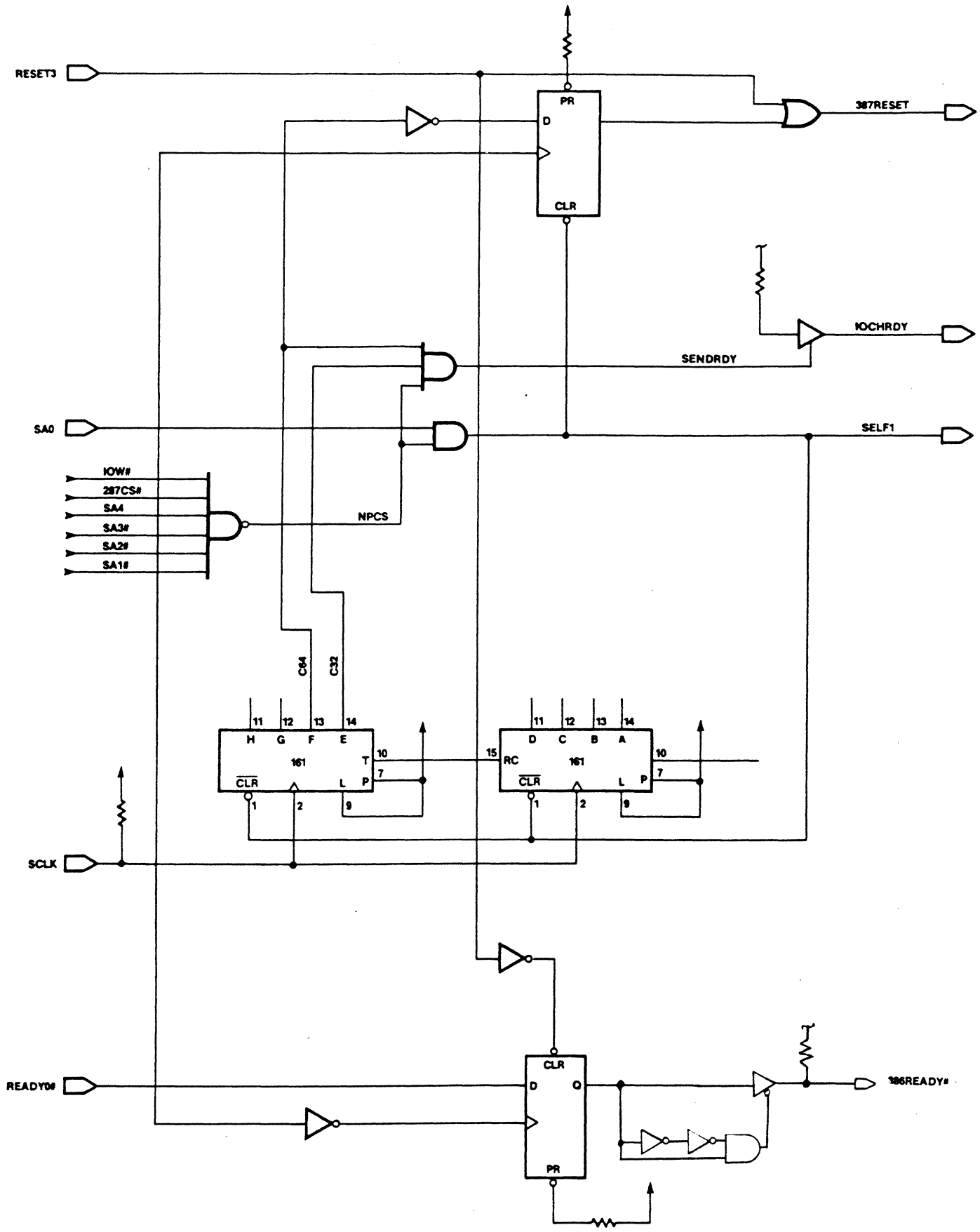


Figure 3.0 80387 RESET Generation Logic

Three conditions have to be satisfied to properly reset the 387 and re-synchronize the 387 to the 386:

1. RESET to the 80387 has to be held high for at least 78 CLK2 cycles

2. The high to low transition of reset should occur during phase 2 of the 386 CLK2. This is to ensure that the internal bus controller clock of the 387 is in phase with the 386 internal clock.

3. After RESET is de-activated, at least 50 CLK2 cycles are required before a new numeric instruction can be executed. This delay is also generated by the logic in Figure 3.0).

2.2.1. OPERATION

When an I/O write to the port F1 is performed, the SELF1 output is activated, which clears the cascaded counter. The C64 output (see Figure 3.0) is inverted and synchronized to SCLK to generate the SRESET. This output is high for 64 SCLK cycles following a write to port address F1H. The falling edge of SRESET occurs during the high to low transition of SCLK, thereby ensuring that reset occurs during Phase 2 of the 386 internal clock. This enables the 387 to maintain synchronization with the 386. The circuit also ensures that following a write to I/O port F1, the reset to the 387 is held high for at least 64 SCLK (128 386 CLK2 cycles) cycles, thereby satisfying the reset high time requirements of the 387 (>> 78 CLK2 cycles). The IOCHRDY# output is forced low until C64 and C32 goes active (after 32 SCLK cycles). This prevents the processor from executing a numeric instruction till 32 SCLK (64 387 CLK2 cycles) cycles following the de-activation of SRESET (Note: The 387 requires at least 50 387 CLK2 cycles for initialization) After 96 SCLK cycles, the IOCHRDY# is released, allowing the processor to resume execution of instructions.

NOTE 1:

During the reset synchronization sequence, the 386 will not honor any refresh requests. The worst case refresh delay is:

192 x 386 CLK2 period

@ 16 MHz, 386 CLK2 period is 62.5 ns.

= 192 x 62.5

= 12 microseconds.

The worst case refresh cycle delay is 12 microseconds. The DRAMs require 128 rows to be refreshed every 2 ms. The PC-AT implements a distributed refresh scheme whereby each row is refreshed every 15 microseconds.

Therefore 128 rows are refreshed every 128 x 15 = 1920 or 1.92 microseconds. Even if one refresh cycle is delayed by 12 microseconds (hopefully we will not have more than one numeric error in 2 ms) the 2 ms requirement is still not violated (1920 + 12 = 1.932 <

2 ms).

The number of RESETs required within a 2 ms interval to violate the refresh requirements can be computed as follows:

The refresh can be delayed by approximately 12 microseconds (worst case : refresh request coincides with an error condition) during an error condition. The refresh implementation on the PC-AT can tolerate a refresh delay of 80 microseconds (2.0 - 1.920 = 0.080 milliseconds). Therefore, if more than six numeric errors occur during a 2 millisecond interval, the DRAM refresh requirement will be violated.

NOTE 2:

Intel recommends that 386/387 should be reset simultaniously. The asynchronous reset feature can easily be disabled by removing the SRESET term in the RES387 generation logic or by deleting the reset synchronization circuitry (in fact removing the reset synchronization circuitry greatly simplifies the numeric co-processor interface).

The present RES387 equation is as follows:

RES387 = SRESET # RESET3; SRESET is the asynch reset input.

To disable asynchronous reset, the following modification is recommended:

RES387 = RESET3;

NOTE3:

Forcing AF32 during pipelined operation

The AF32# to the 82C301 should be active during the start of all local bus accesses (including accesses to the 387 co-processors). The AF32 allows the 82C301 to determine which state machine should be invoked for the current access, ie. either CPU or AT state machine. The 82C306 AF32# is activated during numeric co-processor accesses. The 82C306 AF32# generation circuit functions properly during non-pipelined accesses. In the non-pipelined mode, the AF32# is activated during 32-bit memory accesses as well as 387 accesses. But, when operating in the pipe-lined mode, AF32# is not activated during the third cycle of back to back accesses. This is caused due to the improper clearing of the latch that generates AF32# during pipe-lined cycles. When AF32# is not detected during the third access, the 82C301 attempts to initiate the AT state machine causing improper operation. The above anomaly can easily be corrected by making the following modifications to the numeric co-processsor interface:

Forced AF32# generation during numeric co-processor access:

Requirements for the AF32# generation logic are as follows:

1. AF32# has to be activated at least 26 ns following the trailing edge of MALE#.

2. AF32# must be deactivated before the trailing edge of the MALE# for the next

cycle to prevent bus contention. If a numeric operation is followed by an access to DRAMs, the 82C302 generates AF32# for the the 32-bit memory accesss. The forced AF32# generation logic should remove AF32# before the 82C302 asserts it.

3. AF32# generation logic should be a tri-state output.

To prevent AF32# contention, it is recommended that the forced FA32# be removed at least 1 SCLK prior to the trailing edge of MALE#.

```
!306AF32 :=!MALE & SCLK & A31 & !MIO;      activate AF31 during A31 & !MIO
          # !306AF32 & !SCLK;               keep AF32 active till SCLK is high
```

2.3. Error Generation Logic

The ERROR input to the 386 should be low following the reset sequence till the first ready is detected to allow for proper numeric interface recognition. After READY is detected, the ERROR to the 386 must remain high till the next system reset.

The circuit that satisfies the above condition is illustrated in Figure 1.0. When RESET3 is activated, the 386ERROR# output is activated. The 386ERROR# is held active till the first ready is detected.

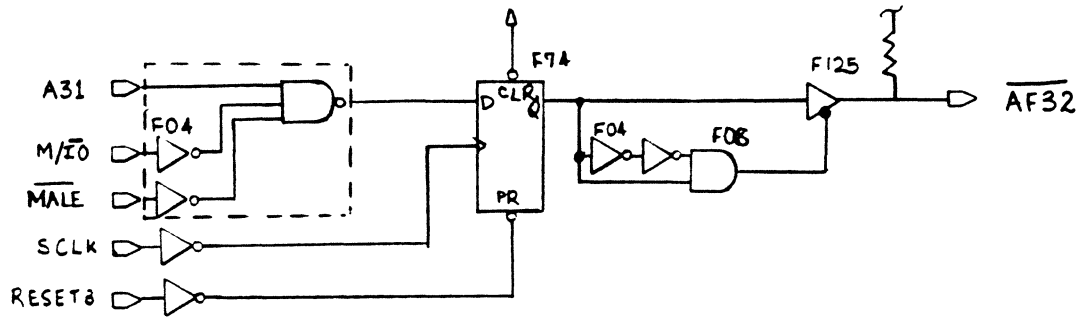


FIG 4.0 PROPOSED AF32# GENERATION CIRCUIT.

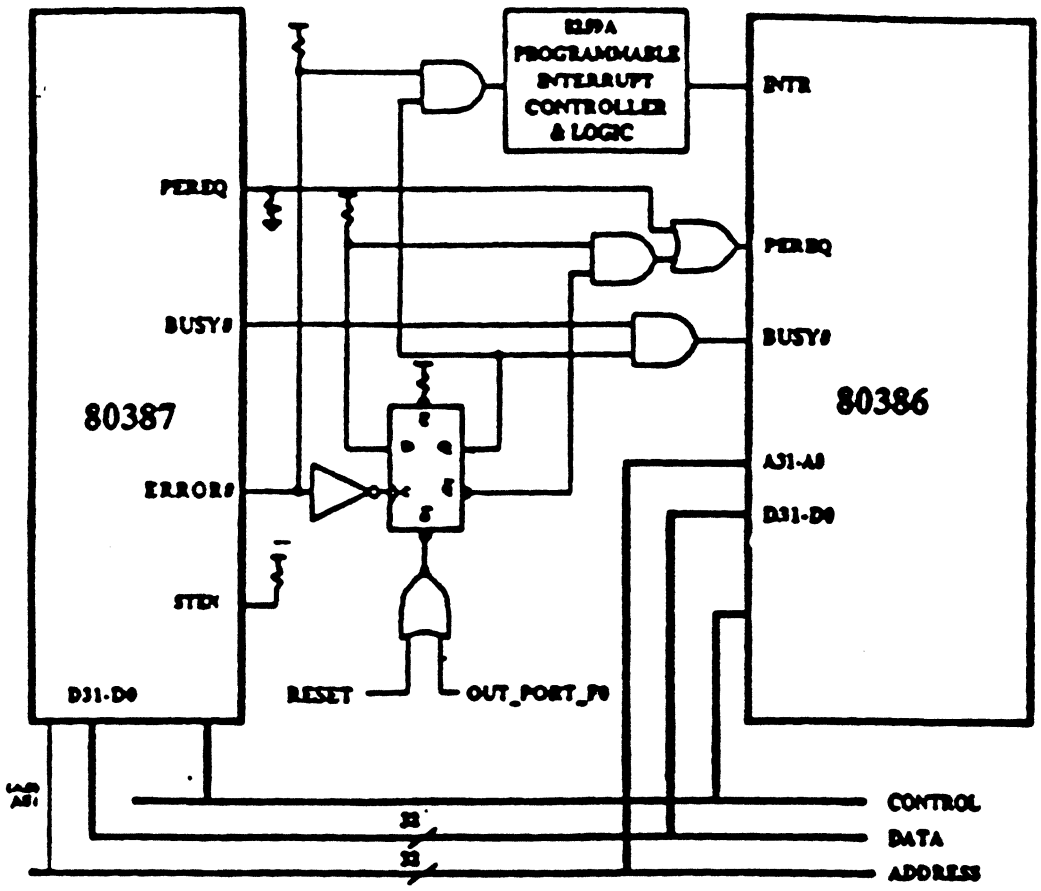


Figure 1.0a PC-AT Compatible 386/387 Interface as recommended by Intel (B0-step 80387 Stepping Information)

3. 386/287 PC-AT COMPATIBLE NUMERIC INTERFACE

If the ERROR# input to the 386 is sampled high following the high to low transition of reset, the 386 assumes it is interfacing either to the 287 co-processor or no co-processor interface exists. It is the responsibility of the software to exercise the numeric interface to determine the presence of a 287 numeric co-processor.

The 80386 performs all the necessary bus cycles to transfer data to or from the 287 on the lower half of the data bus. The 80386 automatically converts 32 bit data transfers into two sixteen bit 287 data transfers. When the 386 detects a numeric instruction, it generates one or more I/O cycles to port address 800000F8H and 800000FCH.

External bus control logic is required to translate 386 bus signals to 287 compatible signals. When using Chips and Technology Chipset, the bus controller function is automatically performed by the 82C301 bus controller. The only additional logic required is the BUSY#, PEREQ, RESET, ERROR#, IOCS16#, NPCS# and IRQ13 generation logic. This logic can be integrated into a single 82S153 PAL.

Figure 4 illustrates the 386/287 interface circuitry. The PEACK# output of the 287 is strapped high. The 287 data bus D0~D15 is connected to RD0~RD15 out of the 82C305. The XIOR# and the XIOW# out of the 82C301 connects directly to the 80287 NPRD# and NPWR# inputs.

3.1. 80287 Clock Generation

The 82C301 generates the 386 processor clock (CLK2), the SCLK and the ATS clock. CLK2 is derived from the CLK2IN input of the 82C301 (If CLK2IN is 32 MHz then CLK2 equals 32 Mhz). The 82C301 provides software controlled selection of the ATSCLK clock used for the AT state machine. The ATSCLK provides the clock input to the 287. This clock input is divided by three to generate the internal clock.

3.2. OPERATION

1. 386 PEREQ is activated when either 287 PEREQ or IRQ13 is active.

2. 287 RESET is acticvated when either the system reset (RESET4) or an I/O write to port F1H is performed. When interfacing the 386 to the 287, reset synchronization is not required as the 287 does not monitor 386 bus cycles.

3. IRQ13 is generated when BUSY and ERROR from the 287 occur at the same time. The IRQ13 will then remain active until RESET or I/O write to port F1 is executed.

4. The 82C304 activates 287CS. for addresses 0E0H to FFH. This output has to be qualified with A3 to ensure that the 287CS is activated only for port accesses to addresses F8H or FCH. The NPCS is not activated during IRQ13.

5. The 386 ERROR# input is strapped high and COPEN# is strapped low..

6. The 386 BUSY# is activated if any of the following conditions hold true:

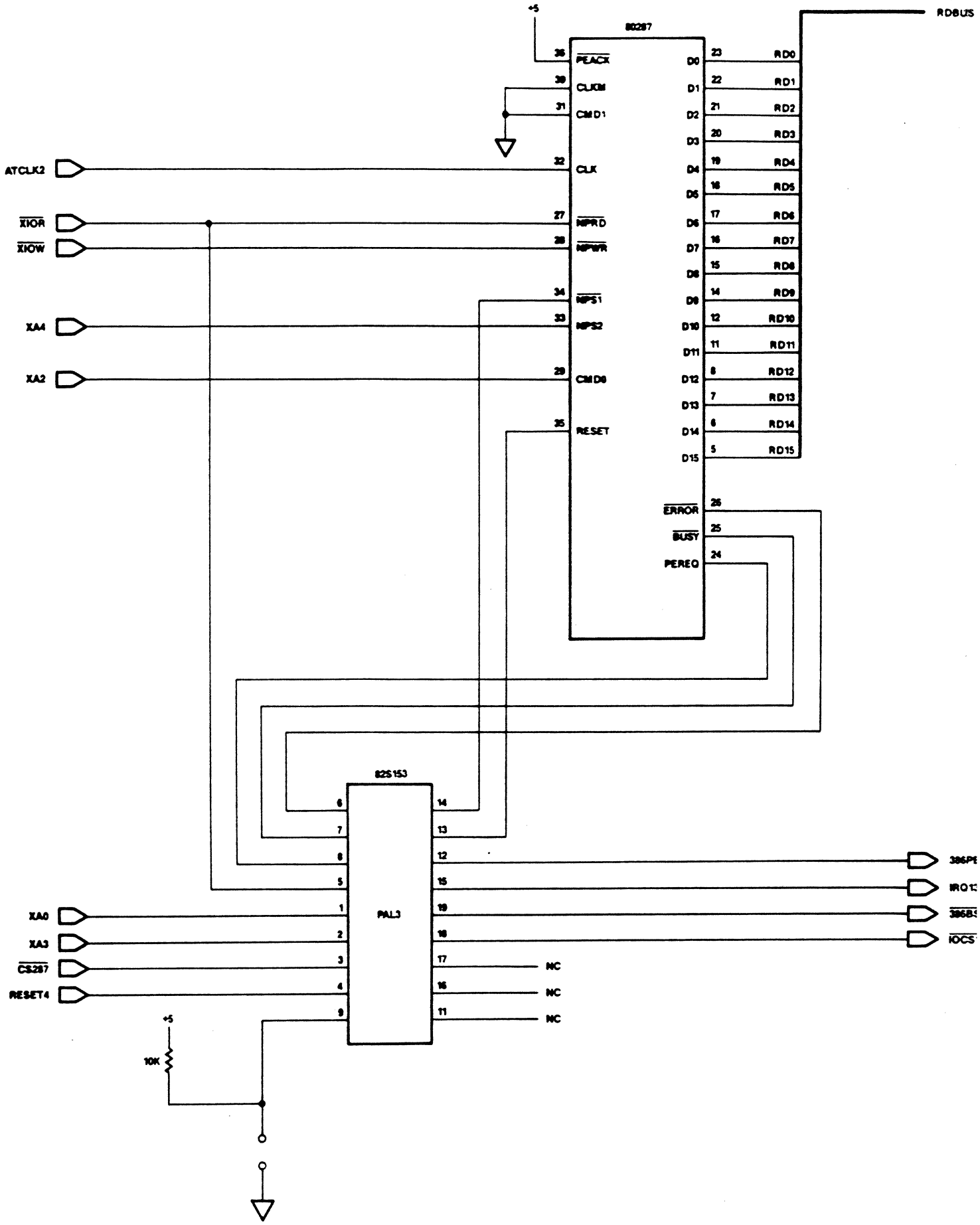


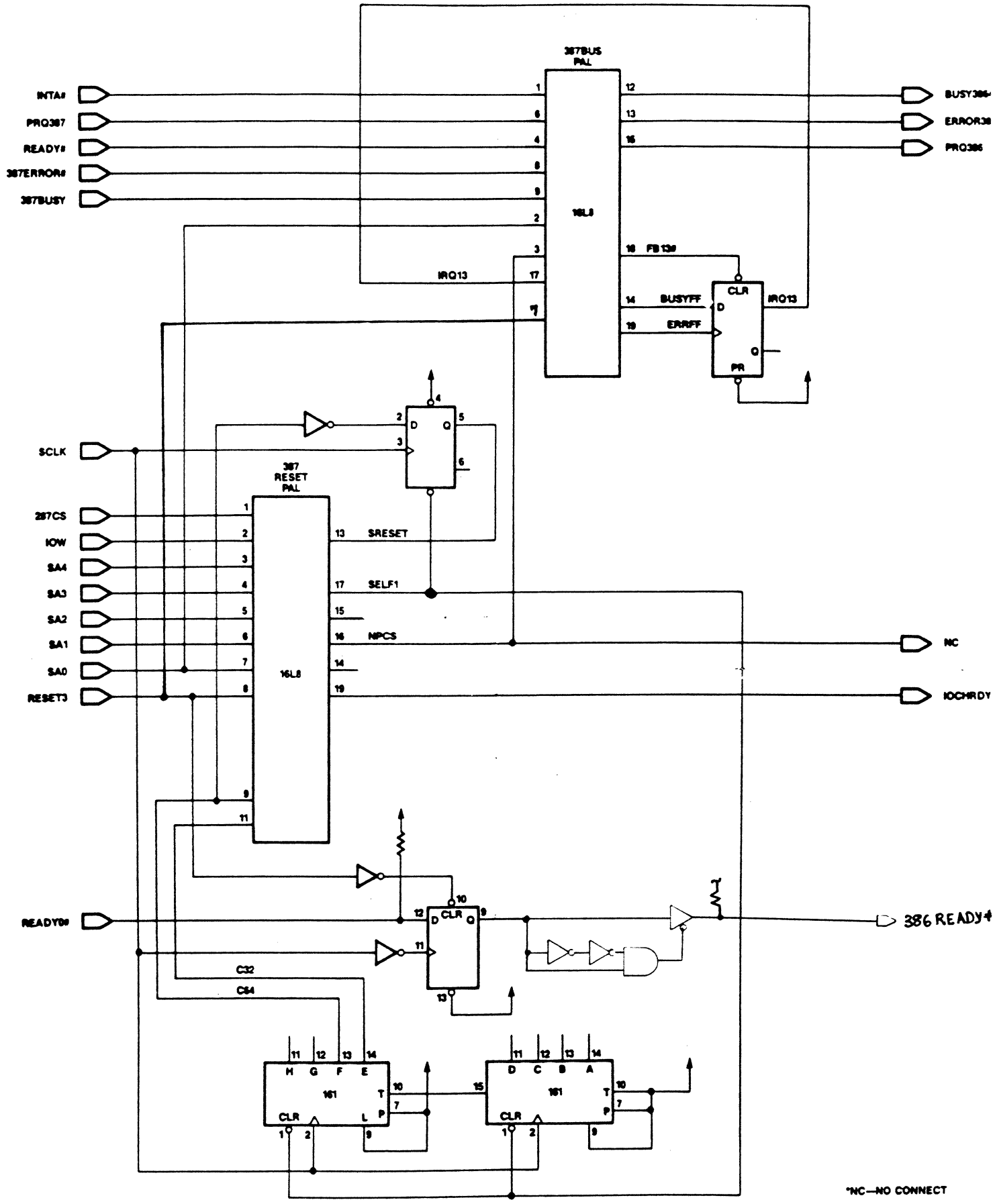
Figure 4.0 80386/80287 PC-AT Compatible Numeric Interface

1. 287 BUSY# and COPEN# are both active
2. IRQ13 and COPEN# are both active
3. NPCS and COPEN# are both inactive

7. IOCS16# to the 82C301 is activated when the 287 is accessed to inform the byte conversion logic on the 82C301 that it is interfacing to a sixteen bit I/O peripheral.

If the design does not use the Chips and Technology CHIPSet, then an external bus controller should be used to generate the numeric co-processor interface. The Bus controller logic should guarantee 80287 timing requirements, particularly the minimum command inactive time (Tcmdi). Using Chips 80386 chipset, AT compatible interface can be realized using a single 82S153 PAL (16L8 PAL can also be used).

APPENDIX



Appendix A. 80386/80387 PC-AT Compatible Numeric Interface

```
PARTNO   387RES ;
NAME     387RES ;
DATE     3/2/87 ;
REV      02 ;
DESIGNER  Chips and Technologies Inc. ;
COMPANY   Chips and Technologies Inc. ;
ASSEMBLY  80387 RESET SUPPORT ;
LOCATION  U51 ;

/*****************************************************************/
/* This device controls RESET timing and IOCHRDY drive during    */
/* 387 reset caused by IO operations. The RESET feature can be   */
/* deleted if asynchronous reset feature is not required         */
/*****************************************************************/
/*  Allowable Target Device Types:  PAL 16L8                     */
/*****************************************************************/

/** Inputs **/

PIN 1      = N_287CS  ; /*  '287 Chip Select from '304 */
PIN 2      = N_IOW    ; /*  !XIOW */
PIN 3      = SA04     ; /*  S-BUS A04 */
PIN 4      = SA03     ; /*  S-BUS A03 */
PIN 5      = SA02     ; /*  S-BUS A02 */
PIN 6      = SA01     ; /*  S-BUS A01 */
PIN 7      = SA00     ; /*  S-BUS A00 */
PIN 8      = RESET3   ; /*  CPU RESET */
PIN 9      = C64      ; /*  Count 64 from external counter */
PIN 11     = C32      ; /*  Count 32 from external counter */
PIN 13     = SRESET   ; /*  Sync reset destined for '387   */
PIN 15     = READYIN  ; /*  Delayed READYO# from '387      */

/* The highlighted input pins should remain unconnected if    */
/* asynchronous reset is not required                         */

/** Outputs **/

PIN 12     = RES387   ; /*  RESET to '387      */
PIN 16     = !NPCS    ; /*  Fully decoded F0,F1 for other PAL */
PIN 17     = SELF1    ; /*  Select F1 - UNCLEAR counters */
PIN 19     = !IOCHRDY ; /*  IOCHRDY drive to bus */
PIN 18     = SENDRDY  ; /*  Send out IOCHRDY */

/* The highlighted output pins are no connect if you do not want to */
/* reset the 387 independently                                      */

/** Declarations and Intermediate Variable Definitions **/


/** Logic Equations **/
```

```
NPCS = !N_IOW & !N_287CS & SA04 & !SA03 & !SA02 & !SA01 ;

RES387 = RESET3 # SRESET ;

/* SRESSET term can be deleted if it is not required to reset */
/* the 387 independent of the 386                             */

SENDRDY = !( C32 & C64 ) & NPCS & SA00 ;

IOCHRDY.OE = SENDRDY ;

IOCHRDY = 'b'1 ;

SELF1 = NPCS & SA00 ;

/* The highlighted terms can be deleted if it is not required */
/* to reset the 387 independent of the 386.                   */
```

```
PARTNO   387BUS ;
NAME     387BUS ;
DATE     3/8/87 ;
REV      00 ;
DESIGNER  Chips and Technologies Inc. ;
COMPANY   Chips and Technologies Inc. ;
ASSEMBLY  80387 BUSY LATCHING ;
LOCATION  U52 ;

/*****************************************************************/
/* This device controls BUSY latching, PEREQ, ERROR, BUSY, and   */
/* IRQ13 generation to the processor.                            */
/*****************************************************************/
/*  Allowable Target Device Types:  PAL  16L8                    */
/*****************************************************************/

/** Inputs **/

PIN 1      = N_INTA   ; /*  INTA#  */
PIN 2      = UNUSED   ; /*  unused */
PIN 3      = N_NPCS   ; /*  F0 OR F1   */
PIN 4      = N_READY  ; /*  READY#     */
PIN 6      = PRQNP    ; /*  PEREQ from 387 */
PIN 7      = RESET3   ; /*  CPU RESET  */
PIN 8      = N_ERRNP  ; /*  ERROR from 387 */
PIN 9      = N_BUSNP  ; /*  BUSY from 387  */
PIN 17     = IRQ13    ; /*  Latched IRQ from FF */

/** Outputs **/

PIN 12     = !BUSCPU  ; /*  BUSY# to '386   */
PIN 13     = !ERRCPU  ; /*  ERROR# TO '386  */
PIN 14     = !BUSFF   ; /*  BUSY to Flip-Flop */
PIN 15     = PRQCPU   ; /*  PEREQ to CPU    */
PIN 16     = !MSKERR  ; /*  Internal signal to pass ERROR */
PIN 19     = !ERRFF   ; /*  ERROR to Flip-Flop */
PIN 18     = !FB13    ; /*  Signal to reset FF  */

/** Declarations and Intermediate Variable Definitions **/


/** Logic Equations **/

BUSCPU = !N_BUSNP & N_KILLB # IRQ13 & N_KILLB ;

ERRCPU = !N_ERRNP & MSKERR ;

BUSFF  = N_BUSNP ;

PRQCPU = PRQNP & IRQ13 & BUSY;
```

```
MSKERR = RESET3 # N_READY & MSKERR ;

ERRFF  = N_ERRNP ;

FB13   = RESET3 # N_INTA & !N_NPCS ;
```

```
CUPL Version 2.10 a Serial # 2-00003-006
Copyright (c) 1983,84,85,86 Personal CAD Systems, Inc.
CREATED Feb 02 1987

        PARTNO      287INF ;
        NAME 287INF ;
        DATE  3/2/87 ;
        REV   01;
        COMPANY     Chips and Technology ;
        ASSEMBLY    80287/80386 interface ;

/* 287 Co-processor interface to the 386 CPU */

/* Allowable Target Devices:    82S153 */

/** inputs **/

pin 1  = XA0    ; /* Address bit 0 */
pin 2  = XA3    ; /* Address bit 3 */
pin 3  = CS287  ; /* co-processor chip select */
pin 4  = RESET4         ; /* System Reset */
pin 5  = XIOW  ; /* I/O Write strobe */
pin 6  = 287ERR         ; /* Error from 287 */
pin 7  = 287BSY         ; /* Busy from 287 */
pin 8  = 287PER         ; /* Extension Request from 287 */
pin 9  = COPEN          ; /* Co-processor interface enable */

/** outputs **/

pin 12 = 386PER         ; /* Extension request to the 386 */
pin 13 = 287RES         ; /* Co-processor reset */
pin 14 = !NPCS; /* Co-processor chip select */
pin 15 = IRQ13 ; /* System Interrupt 13 */
pin 16 = ERR1  ; /* Internal Signal */
pin 17 = DELERR         ; /* Internal Signal */
pin 18 = !IOCS16        ; /* 16-Bit I/O device bus signal */
pin 19 = !387BSY        ; /* Busy to 386 */

/** Logic Equations */

386PER=         287PER & !COPEN
                # IRQ13;

287RES=         XA0 & !XA3 & !CS287 & !XIOW
                # RESET4;

NPCS= !CS287 & !XA3 & !IRQ13;

IRQ13=          !287BSY & !287ERR & !DELERR & XA3 & !RESET4 & !COPEN
                # !287BSY & !287ERR & !DELERR & XIOW & !RESET4 & !COPEN
                # !287BSY & !287ERR & !DELERR & CS287 & !RESET4 & !COPEN
                # IRQ13 & !RESET4 & XA3 & !COPEN
                # IRQ13 & !RESET4 & XIOW & !COPEN
                # IRQ13 & !RESET4 & CS287 & !COPEN;
```

```
ERR1 =!287ERR & !COPEN;

DELERR =    !287ERR & ERR1 & !COPEN;

IOCS16 =    !CS287 & XA3 & !IRQ13;

IOCS16.OE = NPCS;

386BSY =    !287BSY & !COPEN
            # IRQ13 & !COPEN
            # NPCS & COPEN;
```

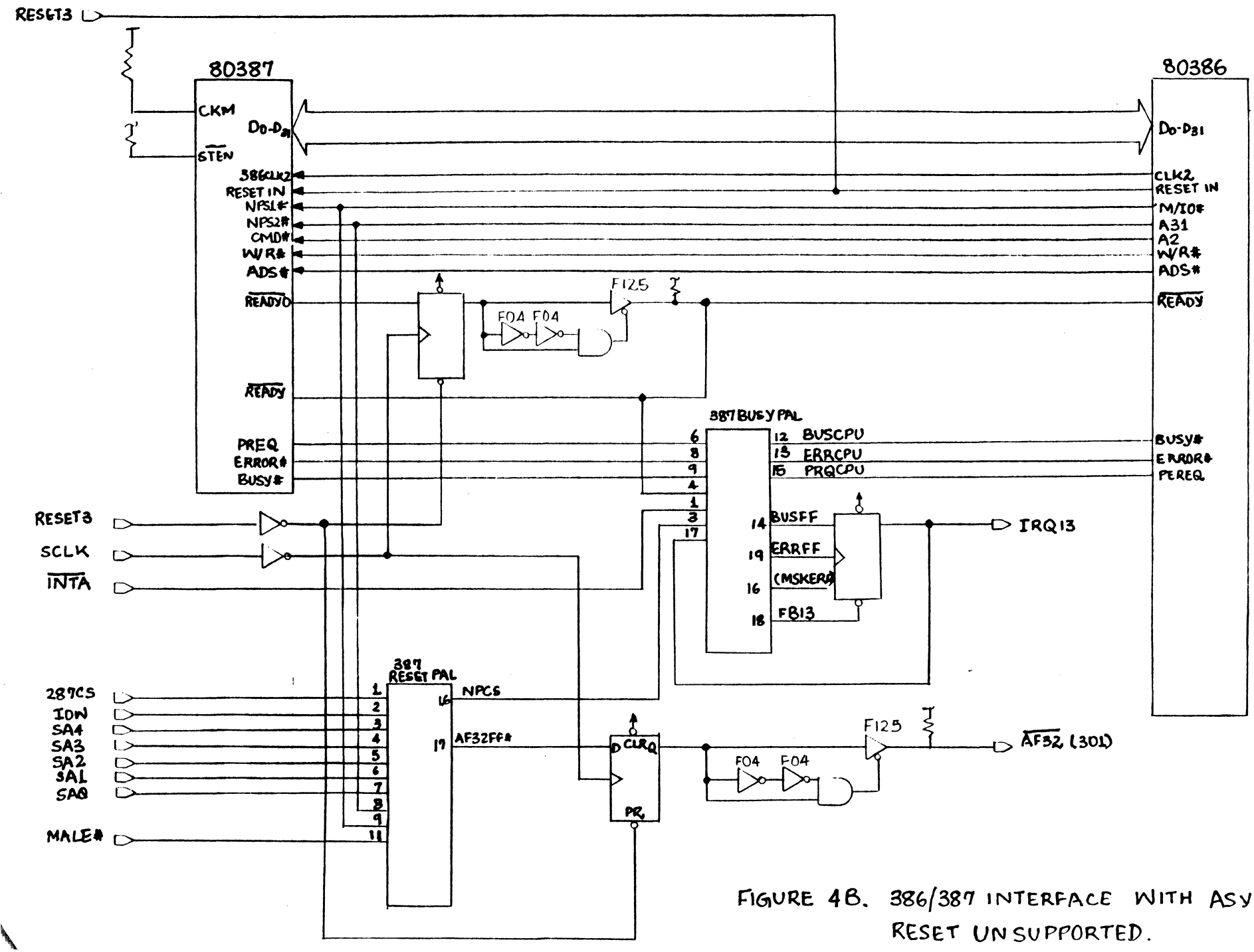


FIGURE 4B. 386/387 INTERFACE WITH ASYNC RESET UNSUPPORTED.