

THE ZILOG DATACOM FAMILY WITH THE 80186 CPU

Zilog's datacom family evaluation board features the 80186 along with four multiprotocol serial controllers, and allows customers to evaluate these components in an Intel environment.

INTRODUCTION

Zilog's customers need a way to evaluate its serial communications controllers with a central CPU. This App Note (Application Note) explains and illustrates how the datacom family interfaces and communicates with the 80186 on this evaluation board. The board helps the

potential customer to evaluate Zilog's data communications controllers in an Intel environment.

The most advanced and complex component of the serial family is the IUSC. One of the highlights of this App Note is how the IUSC adapts to the 80186 CPU with a minimum of difficulty and a maximum of bus and functional flexibility.

GENERAL DESCRIPTION

The evaluation board includes the following hardware. (Reference two page Schematic diagram at rear of the App Note - Figures 5A and 5B.)

- Intel 80186 Integrated 16-bit Microprocessor
- Zilog Z16C32 Integrated Universal Serial Controller (IUSC™)
- Zilog Z16C33 Monochannel Universal Serial Controller (MUSC™) or USC®
- Zilog Z16C35 Integrated Serial Communications Controller (ISCC™)
- Zilog Z85230 Enhanced Serial Communications Controller (ESCC™) or SCC
- Two 28-pin EPROM sockets, suitable for 2764's through 27512's
- Six 32-pin (or 28-pin) SRAM sockets, suitable for 32K x 8 or 128K x 8 devices

- Four Altera EPLD circuits comprising the glue logic (Figures 1-4 at rear of the App Note) and Evaluation Board Schematic (Figures 5a, 5b)

- RS-232 and RS-422 line drivers and receivers

- Pin headers for configuring and interconnecting the above to serial applications

Notes:

All Signals with a preceding front slash, "/", are active Low, e.g.: /B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

| Connection | Circuit | Device |
|------------|-----------------|-----------------|
| Power | V _{CC} | V _{DD} |
| Ground | GND | V _{SS} |

GENERAL DESCRIPTION (Continued)

Processor

The 80186 may be operated at rates up to 16 MHz. To use the CPU clock for accurate serial bit clocking, a 9.8304 MHz CPU clock can be used. The crystal connected to the processor is 2X the operating frequency.

The processor's 1 Mbyte address space is well filled if the maximum RAM complement is installed. Of the integrated Chip Select outputs provided by the 80186, the /UCS output is used for the EPROMs, and all of the /PCS6-/PCS0 outputs are used for the datacom controllers. A hardware address decoder is used for the SRAMs instead of the 80186's /LCS and /MCS3-/MCS0 outputs because the RAMs must be accessible to the on-chip DMA functions of the ISCC and IUSC as well as the 80186. The 80186 does not decode addresses from external bus masters. Both 8-bit and 16-bit accesses are provided for RAM. The EPROMs are only accessible to the 80186.

The 80186's mid-range memory chip select feature (specifically, the /MCS2 output) is used to give the software a way to hardware Reset the ISCC, IUSC, and (M)USC. This allows a customer's program to operate as if it were in a target system starting from Reset, including the initial write to the Bus Configuration Register (BCR).

The 80186's two integrated DMA channels can be used for any two of the four or six serial data streams in the B side of the (E)SCC and the (M)USC. The "DMA EPLD" derives requests for the 80186's two DMA channels from six inputs, two each for (E)SCC channel B and the one or two channels in the (M)USC. It asserts DREQ0 or DREQ1 (High) if any of the inputs for that channel is low, and the 80186 is not performing an Interrupt Acknowledge cycle. Jumper blocks J22, J23, J24, and J29 control the assignment of the 80186's internal DMA controllers, including provision for a clipped Tx request that is needed if a standard SCC is installed in place of the ESCC. The various possibilities are summarized in Table 1.

Table 1. 80186 DMA Jumper Connections

| To enable the following to use 80186 DMA Channel 0: | Install this jumper: |
|---|----------------------|
| (E)SCC B Rx | J23-1 to J23-2 |
| MUSC Rx or USC A Rx | J22-1 to J22-2 |
| MUSC Tx or USC A Tx | J22-4 to J22-2 |
| USC B Rx | J29-1 to J29-2 |
| USC B Tx | J29-4 to J29-2 |
| To enable the following to use 80186 DMA Channel 1: | Install this Jumper: |
| ESCC B Tx | J24-1 to J24-3 |
| (E)SCC B Tx w/early release | J24-1 to J24-2 |
| MUSC Rx or USC A Rx | J22-1 to J22-3 |
| MUSC Tx or USC A Tx | J22-4 to J22-3 |
| USC B Rx | J29-1 to J29-3 |
| USC B Tx | J29-4 to J29-3 |

If more than one channel among the ESCC B and (M)USC are enabled for one of the 80186's internal DMA channels, software must ensure that only one of the enabled devices makes requests during a given block transfer. This can be done by leaving an entire Receiver or Transmitter idle or disabled, or by programming the device so that the DMA request is not output on the pin.

The ISCC and IUSC handle their own DMA transfers via the 80186's HOLD/HLDA facility.

Note: Either a Z16C33 MUSC or a Z16C30 USC can be installed in socket U5. If this is done, references to the (M)USC herein after may mean the USC as a whole or just its channel A; which one should be clear from the context.

The inputs and outputs associated with the processor's integrated counter/timer facility are brought to the pin

header labelled J26 so that they can be used in applications (Table 2).

Table 2. Counter/Timer Signal Locations

| J26 pin | Signal |
|---------|-------------|
| 1 | Timer In 1 |
| 2 | Timer Out 1 |
| 3 | Timer In 0 |
| 4 | Timer Out 0 |
| 5 | N/C |
| 6 | Ground |

The 80186's integrated interrupt controller is largely bypassed in favor of the traditional Zilogical interrupt daisy-chain structure.

Push buttons are provided for Reset and Non-Maskable Interrupt (NMI). A means to generate an NMI, in response to a Start bit received from the user's PC or terminal, is also provided. The first transmitted Start bit on the RS-232. Console connector J1, after a Reset, also produces an NMI; this feature can be used to find which serial controller channel is connected to the Console connector.

Address Map

EPROM is located at the highest addresses, and its size is programmable in the 80186 for the /UCS output. The

addresses of the datacom controllers are programmed in the 80186 for the /PCS6-/PCS0 outputs, as a block of 128x7=896 bytes starting at a 1 Kbyte boundary. The block can be in I/O space or in a part of memory space that is not used for SRAM or EPROM. The starting 1 Kbyte boundary is called (PBA) in the following sections.

RAM extends upward from address 0.

Using 128K x 8 SRAMs and 64K x 8 EPROMs, the address map might be as shown in Table 3.

Table 3. Suggested Address Map

| | |
|------------------------|---|
| RAM | 00000-BFFFF |
| (E)SCC | D8000, 2, 4, 6 or D8000-D803E (even addrs only) |
| ISCC | D8080-D80FE (even addrs only) |
| (M)USC | D8100-D81FF |
| IUSC | D8200-D837F |
| ISCC-IUSC-(M)USC Reset | DB000-DB7FF (if enabled) |
| 27512 EPROM | E0000-FFFFFF |

EPROM

Two 28-pin EPROM sockets are provided; both must be populated in order to handle the 80186's 16-bit instruction fetches. Jumper header J18 allows the sockets to be compatible with 2764s, 27128s, 27256s, or 27512s; it is jumpered at the factory to match the EPROMs provided. For 27512s only, jumper J18-J2 to J18-J3 and leave J18-J1 open. For 2764s, 27128s, or 27256s, jumper J18-J2 to J18-J1 and leave J18-J3 open.

Note: J18 connects pin 1 of both sockets to either A16 or Vcc. This is done because for 2764s, 27128s, and 27256s, pin 1 is Vpp which may require a high voltage and/or draw more current than a normal logic input. For 2764s and 27128s, a similar jumper might be provided in some designs for pin 27 (/PGM). As long as the address for /UCS is programmed as described in the next paragraph, A15 (which is connected to pin 27) is High whenever /UCS is Low, so that 2764s and 27128s operate correctly.

The first code executed after Reset should program the 80186's Chip Select Control Registers to set up the address ranges for which outputs like /UCS and /PCS6-/PCS0 are asserted. In particular, the UMCS register (address A0H within the 80186's Peripheral Control Block)

must be programmed to correspond to the size of EPROMs used (Table 4).

Table 4. EPROM Address Ranges

| EPROM Type | UMCS Value | EPROM | Address Range |
|------------|------------|-------|---------------|
| | 2764 | FC3C | FC000-FFFFFF |
| | 27128 | F83C | F8000-FFFFFF |
| | 27256 | F03C | F0000-FFFFFF |
| | 27512 | E03C | E0000-FFFFFF |

The three LSBs of the above UMCS values are all 100, which signifies no external Ready/WAIT is used and no wait states are required. If the EPROMs are not fast enough for no-wait-state operation, making the three LSBs 101, 110, or 111 extends EPROM cycles by 1, 2, or 3 wait states, respectively.

RAM

Six 32-pin sockets are provided; they should be populated in pairs, starting with the lower-numbered sockets, to allow for 16-bit accesses. V_{CC} is provided at both pin 32 and pin 30 so that 28-pin 32K x 8 SRAMs can be installed in pins

3-30 of the sockets. Jumper block J19 allows decoding of the Chip Select signals from A17-A16 for 32K x 8 SRAMs or from A19-A18 for 128K x 8 SRAMs. The six standard memory populations are:

| | |
|----------------------------------|---------------------------|
| One pair of 32K x 8 devices: | 64 Kbytes at 00000-0FFFF |
| Two pairs of 32K x 8 devices: | 128 Kbytes at 00000-1FFFF |
| Three pairs of 32K x 8 devices: | 192 Kbytes at 00000-2FFFF |
| One pair of 128K x 8 devices: | 256 Kbytes at 00000-3FFFF |
| Two pairs of 128K x 8 devices: | 512 Kbytes at 00000-7FFFF |
| Three pairs of 128K x 8 devices: | 768 Kbytes at 00000-BFFFF |

J19 is factory set according to the size of the SRAMs provided. For 32K x 8 SRAMs, jumpers are installed between J19-J2 and J19-J3, and between J19-J5 and J19-J6, with J19-J1 and J19-J4 left open. For 128K x 8 SRAMs, jumpers are installed between J19-J1 and J19-J2, and between J19-J4 and J19-J5, with J19-J3 and J19-J6 left open.

32K x 8 SRAMs have cyclic/redundant addressing starting at 40000, 80000, and C0000. The only configuration in which this causes problems is with three pairs of 32K x 8 SRAMs and 27512 EPROMs; in this case, there is a conflict in the range E0000-EFFFF. This conflict can be avoided by any of the following means:

- Using two pairs of 32K x 8 SRAMs;
- Using one pair of 128K x 8 SRAMs;

- Using 27256 EPROMs, or
- Using 27512 EPROMs but programming the size of /UCS like they are 27256s.

Since the /LCS output of the 80186 is not used, the LMCS register in the 80186 is not written with any value.

Programming the Peripheral Chip Selects

The 80186 allows the /PCS6-/PCS0 pins, which in this case select the various datacom controllers, to be asserted for a selected 896-byte block of addresses. The block may reside in either memory or I/O space depending on the values programmed into the PACS and MPCS registers, locations A4H and A8H of the 80186's Peripheral Control Block, respectively. The choice of address space depends on the needs of the customer's application and the configuration of software supplied with the board (Table 5).

Table 5. Three Standard Alternatives for Serial Controller Addressing

| Basic Requirement | Base Address (PBA) | PACS value | MPCS value |
|-----------------------------------|--------------------|------------|------------|
| I/O Space | 8000 | 0838 | 81B8 |
| Memory Space, 32K x 8 SRAMS used | 38000 | 3838 | 81F8 |
| Memory Space, 128K x 8 SRAMS used | D8000 | D838 | 81F8 |

The three LSBs of the PACS value specify the Ready/WAIT handling for the /PCS3-/PCS0 lines which select the (E)SCC, ISCC, and (M)USC. The three LSBs of the MPCS value specify the Ready/WAIT handling for the /PCS4, 5, and 6 lines, which select the IUSC. Both fields are shown here with the LSB's 000, signifying that the 80186 should honor a WAIT on the external Ready/WAIT signal, but that it should not provide any minimum wait.

Programming the Mid-Range Memory to Reset the ISCC, IUSC, and (M)USC

A Reset puts the ISCC, IUSC, and (M)USC in a special and unique state in which the first write to each device implicitly goes to a Bus Configuration Register (BCR) that controls the device's basic bus operation; the BCR is not accessible thereafter. So that this board can serve as a complete development environment for customers'

software, it includes a means whereby software (e.g., the debug monitor) can assert the /RESET input of these three devices. Specifically, assertion of the /MCS2 output of the 80186 causes such a Reset.

The 81 in the MS Byte of the MPCS values, shown in Table 5, makes each of the /MCS3-/MCS0 pins correspond to a 2 Kbyte block of addresses in memory space. The actual active pin addresses are determined by the value written into the MMCS register; location A6H of the 80186's Peripheral Control Block. Table 6 shows suggested MMCS values as a function of the RAM chip size, and the corresponding range of addresses for which any read or write access causes the three controllers to be reset.

Table 6. Address Ranges for Reset

| RAM Size | MMCS value | Address Range for which ISCC, IUSC, and (M)USC are Reset: |
|----------|------------|---|
| 32K x 8 | 3BFF | 3B000-3B7FF |
| 128K x 8 | DBFF | DB000-DB7FF |

The three LSBs of the above MMCS values are 111 so that the longest possible Reset pulse is generated when any of the locations in the indicated range are accessed.

Interrupt Daisy Chain (Priority) Order

Jumper block J25 selects whether the (E)SCC device is at the start or the end of the interrupt daisy chain.

Note that if this feature is not needed, it can be disabled by simply not programming the MMCS register.

| To make the interrupt priority be: | Jumper J25 as follows: |
|---|--|
| (E)SCC highest, IUSC, ISCC, (M)USC lowest | J25-J2 to J25-J3, J25-J4 to J25-J5 (J25-J1, J25X open) |
| IUSC highest, ISCC, MUSC, (E)SCC lowest | J25-J1 to J25-J2, J25-J to J25-J4 (J25-5J, J25X open) |
| IUSC highest, ISCC, USC, (E)SCC lowest | J25X to J25-J2, J25-J3 to J25-J4 (J25-J1, J25-J5 open) |

This variability is provided in part because early versions of the 85230 ESCC had trouble passing an interrupt acknowledge down the daisy chain if it occurred in

response to a lower-priority device's request just as the ESCC was starting to make its own request. Current 85230's don't have the problem.

(E)SCC

Socket U2 can be configured for either an ESCC or SCC, and for versions thereof that use either multiplexed or non-multiplexed address and data. Jumper blocks J20 and J21 select certain signals accordingly. For a part with multiplexed addresses and data (80x30), jumper J20-J1 to J20-J2 and leave J20-J3 open, and jumper J21-J1 to J21-J2 and J21-J4 to J21-J5, leaving J21-J3 and J21-J6 open. With such a part, software can directly address the (E)SCC's registers, and need not concern itself with writing register addresses to Write Register 0 (WR0).

For a part having a non-multiplexed bus (85x30), jumper J20-J2 to J20-J3, J21-J2 to J21-J3, and J21-J5 to J21-J6, leaving J20-J1, J21-J1, and J21-J4 open. In this case, software must handle the (E)SCC by writing register addresses into its WR0 in order to access any register other than WR0, RR0, or the data registers.

Channels A and B can be handled on a polled or interrupt-driven basis. Channel A of the (E)SCC is suggested for connecting the user's PC or terminal for use with the Debug Monitor included in this evaluation kit. Channel B (but not A) can be handled on a DMA basis using the 80186's internal DMA channels, or on a polled or interrupt driven basis.

Jumper block J23 allows channel B's /W//REQB output to be used for either a Wait function or a Receive DMA Request function. To use the output for Wait, jumper J23-J2 to J23-J3 and leave J23-J1 open. The Wait function is only significant if the software wants to delay completion of a Read from the (E)SCC's Receive Data register until data is available, and/or if it wants to delay completion of a Write to the Transmit Data register until the previously-written character has been transferred to the Transmit Shift register. These modes are alternatives to checking the corresponding status flags and can be used to achieve operating speeds higher than those possible with such traditional polling, although not as fast as the speeds possible with a DMA approach.

To use the /W//REQB output as a Receive DMA Request, jumper J23-J1 to J23-J2 and leave J23-J3 open.

Jumper block J24 determines how channel B's /DTR/ /REQB output is used. To use this output for the Data Terminal Ready function, jumper J24-J3 to J24-J4 and leave J24-J1 and J24-J2 open. To use this output directly as a Transmit DMA Request (using the ESCC's early-release capability), jumper J24-J1 to J24-J3 and leave J24-J2 and J24-J4 open. To drive the Transmit DMA Request with a clipped version of this signal that is forced High earlier than a standard SCC drives it High, jumper J24-J1 to J24-J2 and leave J24-J3 and J24-J4 open.

The "SCC EPLD" handles the (E)SCC's signalling requirements. Among other things, this EPLD configures the (E)SCC socket's pins 35 and 36 for either a multiplexed or non-multiplexed part, based on whether J20 is jumpered to connect the 80186 ALE signal to one of its input pins. If the device detects high-going pulses on this input, it drives corresponding low-going Address Strobe pulses onto (E)SCC pin 35 and drives low-going Data Strobe pulses onto (E)SCC pin 36.

If the SCC EPLD's pin 9 stays at Ground, the part drives Read strobes onto pin 36 and drives delayed Write strobes onto pin 35, for a non-multiplexed 85x30 device.

While the ESCC's relaxed timing capability allows the 80186's /WR output to be connected directly to the /WR input of a non-multiplexed ESCC, the SCC EPLD delays start of an SCC's write cycle until write data is valid, even though this is not necessary for an ESCC.

The SCC EPLD also generates the clipped-DMA-request signal mentioned in connection with J24, and logically ORs Reset onto pins 35 and 36. The device also tracks the two IACK cycles provided by the 80186 for each Interrupt Acknowledge cycle. For a multiplexed address/data port, it drives the address strobe (only) on the first cycle, and it provides the /RD or /DS pulse needed by the (E)SCC (only) on the second cycle. The "DMA EPLD" provides the INTACK signal needed by the (E)SCC.

The (E)SCC is only accessible at even addresses. For a non-multiplexed part (85x30), the following four register locations are repeated throughout the even addresses from (PBA) through (PBA)+126:

| | |
|------------------------------|-----------------------------------|
| (PBA), (PBA)+8,... (PBA)+120 | Channel B Command/Status register |
| (PBA)+2, +10,... (PBA)+122 | Channel B Data register |
| (PBA)+4, +12, ... (PBA)+124 | Channel A Command/Status register |
| (PBA)+6, +14, ... (PBA)+126 | Channel A Data register |

For a multiplexed part (80 x 30), the Select Shift Left command (D1-0=11) should be written to Channel B's WR0 before any other registers are accessed. Then the

basic (E)SCC register map occurs twice in the even addresses from (PBA) through (PBA)+126:

| | |
|------------------------------|--------------------------|
| (PBA), (PBA)+2, ... (PBA)+30 | Channel B registers 0-15 |
| (PBA)+32, +34, ... (PBA)+62 | Channel A registers 0-15 |
| (PBA)+64, +66, ... (PBA)+94 | Channel B registers 0-15 |
| (PBA)+96, +98, ... (PBA)+126 | Channel A registers 0-15 |

The redundant addressing of the (E)SCC is used to control a feature that can be used by software to allow the user to interrupt software execution from his keyboard. If the (E)SCC is read at an address with A6-A5=11 (for a multiplexed part this means in the higher-addressed A channel), a mode is set in which a low on the console Received Data line (i.e., a Start bit on pin 3 of the J1

connector) causes a Non-Maskable Interrupt on the 80186. The mode is cleared by Reset, or when the (E)SCC is read at an address with A6-A5=10 (on a multiplexed part, in the higher-addressed B channel). The NMI handler should do the latter fairly quickly to prevent subsequent data bits on Received Data from causing further NMIs.

ISCC

Since the 80186 processor provides multiplexed addresses and data, the ISCC is configured to use the addresses on the AD lines. Therefore, software can address the various ISCC registers directly, and need not be concerned with writing register addresses into the indirect address fields of the ISCC's WR0 and CCAR.

Because the ISCC includes four DMA channels, its Channel A and B Transmitters and Receivers can be handled on a polled, interrupt-driven, and/or DMA basis, in any mixture.

Since the ISCC can only be programmed as an 8-bit device on the AD7-AD0 lines, it occupies only the even-addressed bytes within its address range, (PBA)+128 through (PBA)+254.

The first write to this address range, after a Reset, implicitly writes the ISCC's Bus Configuration Register (BCR). To match up with the rest of the board's hardware, this first write should be a byte write that stores the hexadecimal value C6 in any even address in the first half of the ISCC's address range [(PBA)+128 through (PBA)+190]. Details of this transaction are as follows:

- The High induced by a pull-up resistor on the ISCC's A/B input selects the WAIT protocol on the /WAIT//RDY pin, which corresponds to how the 80186 works. (In subsequent register accesses, the A/B selection is taken from A5 of the multiplexed address.)

- A Low on the ISCC's SCC//DMA input, which is connected to A6, is required by the internal logic of the ISCC. This is why the BCR write is restricted to the first half of the ISCC's address range.
- As with all transactions between the 80186 and ISCC, the address must be even because the ISCC only accepts slave-mode data on the AD7-AD0 pins.
- The MSB of the data (D7) is 1 to enable the Byte Swap feature, so that when the ISCC's DMA controller is reading transmit data from RAM, it takes alternate bytes from AD7-AD0 and AD15-AD8.
- D6 of the data is 1 so that when the ISCC's DMA controller is reading transmit data from RAM, it takes even-addressed bytes from D7-D0 and odd-addressed bytes from D15-D8 (same function as the 80186).
- D2-D1 of the data are 11 to select double-pulsed mode for the ISCC's /INTACK input. Again, this is how the 80186 works.
- D0 of the data is 0 to select Shift Left Address mode so that the ISCC subsequently takes register addressing from the AD5-AD1 lines rather than from AD4-AD0. This is because the 80186 is a 16-bit processor that locates even-addressed bytes on AD7-AD0 and odd-addressed bytes on AD15-AD8, but the ISCC only accepts slave-mode writes on the AD7-AD0 pins.

- The fact that the ISCC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the ISCC's slave-mode address map is as follows:

| | |
|--------------------------------|--------------------------------------|
| (PBA)+128, 130, ..., (PBA)+190 | DMA Controller Registers |
| (PBA)+192, 194, ..., (PBA)+222 | ISCC Serial Channel B registers 0-15 |
| (PBA)+224, 226, ..., (PBA)+254 | ISCC Serial Channel A registers 0-15 |

(M)USC

Since the 80186 processor provides multiplexed addresses and data, the (M)USC is configured to use the addresses on the AD lines. Therefore, the software need not write register addresses into the indirect address field of the (M)USC's CCAR.

The (M)USC's Transmitter and Receiver can be handled on a polled or interrupt-driven basis. In addition, any two of the Receivers and Transmitters in the (M)USC and Channel B of the (E)SCC can be handled on a DMA basis, using the 80186's integrated DMA controllers.

Jumper block J22 connects the (M)USC's /RxREQ and /TxREQ outputs to the "DMA EPLD" that makes the DMA Requests to the 80186. As shipped from the factory, jumpers are installed between J22-J1 and J22-J2, and between J22-J3 and J22-J4. In this configuration, the (M)USC's /RxREQ drives the 80186 DREQ0, and (M)USC /TxREQ drives the 80186 DREQ1. To reverse this assignment, jumper J22-J1 to J22-J3 and J22-J2 to J22-J4. To disconnect the (M)USC from one or both of the 80186's DMA channels, remove one or both jumpers (put them in a safe place in case you change your mind). Jumper block J29 provides the same connection-variability for the /RxREQ and /TxREQ outputs of Channel B of a USC.

Since the 80186's DMA channels are not capable of fly-by operation, the (M)USC's /RxACK and /TxACK pins have no dedicated function. They can be used for Request to Send and Data Terminal Ready; the two signals are lightly pulled up since they are not driven after Reset.

The (M)USC can be programmed using 16-bit data on the AD15-AD0 lines or 8-bit data on AD15-AD8 and AD7-AD0. It makes the distinction between 8-bit and 16-bit operations as part of its address map rather than through a control input. The PS pin of an MUSC, or the A//B pin of a USC, is connected to a latched version of 80186 A7. The D//C pin of the (M)USC is grounded. The overall address

range of the (M)USC is 256 bytes, between (PBA)+256 and (PBA)+511.

The first write to this address range, after a Reset, implicitly writes the (M)USC's Bus Configuration Register (BCR). To match the rest of the board's hardware, this first write should be a 16-bit write, storing the hex value 0007 at any address in the second half of the (M)USC's range [any address in (PBA)+384 through 510, i.e., in the A channel of a USC]. Details of this transaction are as follows:

- The High on the PS or A//B input, which is connected to A7, selects the WAIT protocol on the /WAIT//RDY pin, corresponding to how the 80186 works.
- The MSB of the data (D15) is 0 because a separate non-multiplexed address is not wired to pins AD13:8 of the (M)USC.
- Bits 14-3 are required to be all zeros by the (M)USC's internal logic.
- D2 of the data is 1 to tell the (M)USC that the data bus is 16 bits wide.
- D1 of the data is 1 to select double-pulsed mode for the (M)USC's /INTACK input. This is how the 80186 works.
- D0 of the data is 1 to select Shift Right Address mode so that the (M)USC subsequently takes register addressing from the AD6-AD0 lines rather than from AD7-AD1.
- The fact that the (M)USC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the (M)USC address map is as follows:

| Starting Addr | Ending Addr | Registers Accessed |
|---------------|-------------|--|
| (PBA)+256 | (PBA)+319 | 16-bit access to MUSC regs or USC channel B regs |
| (PBA)+320 | (PBA)+383 | 8-bit access to MUSC regs or USC channel B regs |
| (PBA)+384 | (PBA)+447 | 16-bit access to MUSC regs or USC channel A regs |
| (PBA)+448 | (PBA)+511 | 8-bit access to MUSC regs or USC channel A regs |

Note: To maximize compatibility, program an MUSC using the second half of this range, (PBA)+384 through (PBA)+511.

While the ESCC and ISCC can drive their Baud Rate Generators from their PCLK inputs, the (M)USC has no such input. The 80186 clock output SYSCLK is brought to

pins 7 of J9, J10, and J12, at which point it can be jumpered to pin 9 or 8 so that it is routed to the /TxC or /RxC pin of the device.

IUSC

Since the 80186 processor provides multiplexed addresses and data on the AD lines, the IUSC is configured to use these addresses. Software need not write register addresses into the indirect address fields of the IUSC's CCAR and DCAR.

The IUSC's two DMA channels allow its Receiver and Transmitter to be handled on a polled, interrupt-driven, or DMA basis, in any combination.

The IUSC can be programmed using 16-bit data on the AD15-AD0 lines or 8-bit data on AD15-AD8 and AD7-AD0. The distinction between 8-bit and 16-bit operations is made as part of the address map rather than via a control input. The D//C pin of the IUSC is driven from A7 during slave cycles, and the S//D pin is driven from A8. The overall address range of the IUSC is 384 bytes from (PBA)+512 through (PBA)+895.

The first write to this address range, after a Reset, implicitly writes the IUSC's Bus Configuration Register (BCR). To match up with the rest of the board's hardware, this first write is a 16-bit write, storing the recommended hex value 00F7 at any word address in the range (PBA)+768 through (PBA)+830. Details of this transaction are as follows:

- The High on the IUSC's S//D input, which is connected to A8, selects the WAIT protocol on the /WAIT//RDY pin, which is how the 80186 works.
- It may not be required for this initial write, but it is good programming form for A6 to be zero since this is a word write. This and the previous point determine the recommended address range.
- The MSB of the data (D15) is 0 because a separate non-multiplexed address is not wired to pins AD13:8 of the IUSC.
- Bits 14-8 are more or less required to be all 0 by the IUSC's internal logic.

- D7-D6 are 11 to allow the DMA controllers to do either 16-bit transfers, or alternating byte transfers on AD7-AD0 for even-addressed bytes and on AD15-AD8 for odd-addressed bytes. This is compatible with 80186 byte ordering.
- D5-D4 of the data are 11 to select double-pulsed mode for the IUSC's /INTACK input. Again, this is how the 80186 works.
- D3 of the data is 0 to select open-drain mode on the IUSC's /BUSREQ pin. The board's control logic also drives this signal low when the ISCC asserts its Bus Request output.
- D2 of the data is 1 to tell the IUSC that the data bus is 16 bits wide.
- D1 of the data is 1 to select open-drain mode on the IUSC's /INT pin which is OR-tied with the interrupt request from the (E)SCC.
- D0 of the data is 1 to select Shift Right Address mode, so that the IUSC subsequently takes register addressing from the AD6-AD0 lines rather than from AD7-AD1.
- The fact that the IUSC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the IUSC slave-mode address map is as follows:

IUSC (Continued)

| Starting Addr | Ending Addr | Registers Accessed |
|---------------|-------------|---|
| (PBA)+512 | (PBA)+575 | 16-bit access to IUSC Transmit DMA registers |
| (PBA)+576 | (PBA)+639 | 8-bit access to IUSC Transmit DMA registers |
| (PBA)+640 | (PBA)+703 | 16-bit access to IUSC Receive DMA registers |
| (PBA)+704 | (PBA)+767 | 8-bit access to IUSC Receive DMA registers |
| (PBA)+768 | (PBA)+831 | 16-bit access to IUSC Serial Controller registers |
| (PBA)+832 | (PBA)+895 | 8-bit access to IUSC Serial Controller registers |

While the ESCC and ISCC can drive their Baud Rate Generators from their PCLK inputs, the IUSC cannot do this from its CLK input. The 80186 clock output SYSCLK is brought to pins 7 of J9, J10, and J12 at which point it can be jumpered to pin 9 or 8 so that it is routed to the /TxC or /RxC pin of the device.

Since the IUSC contains its own DMA channels, its /RxREQ and /TxREQ pins have no dedicated function. They can be used for Request to Send and Data Terminal Ready; the two signals are lightly pulled up to allow for the fact that they are not driven after Reset.

SERIAL INTERFACING

The serial I/O pins of the four serial controllers are connected to the six connector blocks labelled J5 through J10. In addition, the port pins of the IUSC are connected to the J11 connector block, and the port pins of an MUSC or the B channel of a USC are connected to J12. These connector blocks can be interconnected for communication between on-board serial controllers, or they can be connected to the user's custom communications hardware on another board. As a third option, they can be connected to three on-board serial interfaces via the connector blocks labelled J13 through J15.

Two of the on-board serial interfaces use EIA-RS-232 signal levels and pin arrangement. 25-pin D connectors J1A or J2A are configured as DTE, while J1B and J2B are configured as DCE. These serial interfaces are used by

connecting one of J5-J10 to J13 or J14, respectively. J1B is typically used for connection to the user's PC or terminal.

The third on-board serial interface uses EIA-422 signal levels on connector J3A, J3B, or J4, and is used by connecting one of J5-J10 to J15. The 25-pin D connector J3A uses the DTE pin arrangement put forth in the EIA-530 standard. J3B is a DCE version of EIA-530, while the 8-pin circular DIN connector, J4, is compatible with the Apple Macintosh Plus and later Macintoshes, and thus with AppleTalk/LocalTalk equipment.

The serial interface connectors are summarized in the following tables:

Table 7. Controller Port Connectors

| To use the following serial controller channel with off-board or on-board serial hardware: | Connect to this (these) 10-pin connector block(s): |
|--|--|
| (E)SCC Channel A | J5 |
| (E)SCC Channel B | J6 |
| ISCC Channel A | J7 |
| ISCC Channel B | J8 |
| IUSC | J9 (J11 for Port pins) |
| (M)USC | J10 (J12 for MUSC Port pins or USC channel B) |

Table 8. On-Board Line Driver/Receiver Connectors

| To use a serial chip controller with the following on-chip serial interface: | Connect the connector(s) from the previous table to: |
|--|--|
| J1A or J1B EIA-RS-232 Console | J13 |
| J2A or J2B EIA-RS-232 | J14 |
| RS-422 differential: J3A or J3B EIA-530 or J4 Circular-8 (DIN) | J15 |

The pin-out of the J5-J10 connectors is fairly consistent, among the various serial controllers: but of necessity not identical because of differences

Table 9. Pin Assignments of Standard Controller Connectors

| Pin# | J5: (E)SCC | J6: (E)SCC | J7,8: ISCC | J9: IUSC | J10: MUSC | J12: USC |
|------|------------|-------------------|------------|----------|--------------|----------|
| | A pin | B pin | pin | pin | or USC A pin | B pin |
| 1 | TxD | TxD | TxD | TxD | TxD | TxD |
| 2 | RxD | RxD | RxD | RxD | RxD | RxD |
| 3 | /RTS | /RTS | /RTS | (N/C) | /RxACK | /RxACK |
| 4 | /CTS | /CTS | /CTS | /CTS | /CTS | /CTS |
| 5 | /DTR | /DTR or (N/C) [1] | /DTR | (N/C) | /TxACK | /TxACK |
| 6 | /DCD | /DCD | /DCD | /DCD | /DCD | /DCD |
| 7 | /SYNC | /SYNC | /SYNC | (SYSCLK) | (SYSCLK) | (SYSCLK) |
| 8 | /RTxC | /RTxC | /RTxC | /RxC | /RxC | /RxC |
| 9 | /TRxC | /TRxC | /TRxC | /TxC | /TxC | /TxC |
| 10 | GND | GND | GND | GND | GND | GND |
| 11 | NA | NA | NA | /TxREQ | /TxREQ | /TxREQ |
| 12 | NA | NA | NA | /RxREQ | /RxREQ | /RxREQ |

Note:

[1] Controlled by the J24 jumper block: must be N/C if (E)SCC channel B transmitter is to be handled by an 80186 DMA channel.

The ground pins are included as signal references with off-board hardware.

enough opposing inputs and outputs as needed to make the communication protocol meaningful.

When interconnecting between two connectors among J5-J10, DO NOT jumper corresponding pins straight across, as this connects outputs to outputs and inputs to inputs. Rather, connect at least each pin 1 to the other pin 2, and

The pin-out of the 12-pin J13-J15 connectors is similar to that of J5-J10, but more extensive. To allow for the "DCE" connectors that were added in revision "B" of the board, J13 and J14 are 16-pin headers and J15 is a 14-pin one:

Table 10. Pin Assignments of Line Driver/Receiver Connectors

| Pin # | J13-J14 | J13-J14 | J15 | J15 | Direction/where used |
|-------|------------|------------|------------|------------|----------------------|
| | DTE signal | DCE signal | DTE signal | DCE signal | |
| 1 | TxD | RxD | TxD | RxD | Output to J1-J4 |
| 2 | RxD | TxD | RxD | TxD | Input from J1-J4 |
| 3 | /RTS | /CTS | /RTS | /CTS | Output to J1-J3 |
| 4 | /CTS | /RTS | /CTS | /RTS | Input from J1-J4 [3] |
| 5 | /DTR | /DSR | /DTR | /DSR | Output to J1-J4 |
| 6 | /DSR | /DTR | /DSR | /DTR | Input from J1-J4 |

Note:

[3] Various conventions have been used to combine synchronous clock inputs and modem control inputs on Apple Macintosh connectors similar to J4, as described in a later section.

SERIAL INTERFACING (Continued)

Table 10. Pin Assignments of Line Driver/Receiver Connectors

| Pin # | J13-J14 DTE signal | J13-J14 DCE signal | J15 DTE signal | J15 DCE signal | Direction/where used |
|-------|-----------------------|-----------------------|-------------------|-------------------|------------------------------|
| 7 | | /DCD | | /DCD | Output to J1B, J2B, J3B |
| 8 | /DCD | | /DDC | | Input from J1A, J2A, J3A, J4 |
| 9 | | | | | |
| 10 | GND | GND | GND | GND | |
| 11 | | /RxC | | /RxC | Output to J1B, J2B, J3B |
| 12 | /RxC | | /RxC | | Input from J1A, J2A, J3A |
| 13 | /TxCO | /TxCI | /TxCO | /TxCI | Output to J1-3 |
| 14 | /TxCI | /TxCO | /TxCI | /TxCO | Input from J1-3 [3] |
| 15 | | /RI | | | Output to J1B, J2B |
| 16 | /RI | | | | Input from J1A, J2A |

Note:

[3] Various conventions have been used to combine synchronous clock inputs and modem control inputs on Apple Macintosh connectors similar to J4, as described in a later section.

Comparison of the two preceding charts leads to several conclusions:

- Pins 1-5 can always be jumpered straight across from a J5-J10 connector block to a J13-J15 connector block.
- In a synchronous environment, the Transmit clock can be either driven or received and the Receive clock can be received from the DTE connector or sent on the DCE connector.

The 10-pin J11 and J12 jumper blocks provide for connections to the Port pins of the IUSC and (M)USC, respectively. As with J5-J10, these connections may be to the customer's off-board custom circuits and/or to certain pins in the J13-J15 blocks. The following pin assignment is determined so that if a 2-channel USC is plugged into the (M)USC socket, J12 has the same pin-out for the USC's B channel as do J5-J10 for other channels.

Table 11. Pin Assignments of Controller Port Connectors

| Pin # | J11: IUSC Signal | J12: (M)USC Signal |
|-------|---------------------------|---------------------------|
| 1 | PORT1 (Clock 1 In) | PORT1 |
| 2 | PORT4 (Xmit TSA Gate Out) | PORT4 (Xmit TSA Gate Out) |
| 3 | (N/C) | (N/C) |
| 4 | PORT0 (Clock 0 In) | PORT0 |
| 5 | (N/C) | (N/C) |
| 6 | PORT3 (Rcv TSA Gate Out) | PORT3 (Rcv TSA Gate Out) |
| 7 | (N/C) | (SYSCLK) |
| 8 | PORT5 (Rcv Sync Out) | PORT5 (Rcv Sync Out) |
| 9 | PORT2 | PORT2 |
| 10 | GND | GND |
| 11 | PORT6 (Rcv Sync In) | PORT6 (Rcv Sync In) |
| 12 | PORT7 (Xmit Complete Out) | PORT7 (Xmit Complete Out) |

Finally, an unpopulated 4-pin oscillator socket is included on the board with its output connected to a single jumper/wire-wrap pin. This socket can be populated with a user-supplied oscillator and connected to various clock pin(s) among J5-J15.

Sensing which Serial Controller Channel is connected to the Console

In order to use the software provided with this evaluation board, one of the serial controller channels must be connected to a Personal Computer (or a dumb terminal) via the J1 and J13 connectors. Some versions of this software may restrict the choice to (E)SCC Channel A or the (M)USC, depending on the user's applications needs, but there is nothing in the hardware that limits the choice of which serial channel is used for the Console. However, on the J1-J4 (J13-J15) side there are two things that are special about the J1/J13 section as compared to the others. One is the provision for a Non-Maskable Interrupt in response to a received Start bit, as described earlier in the section on (E)SCC addressing.

Software can use the other special feature of the J1/J13 section, after a Reset, to sense which serial channel is connected to the Console port. A Reset signal (from power-on or the Reset button, but not from the Reset-the-ISC, etc., address decode as described earlier) puts the "NMI" EPLD in a special mode wherein the first Start bit on the Console's Transmit Data lead causes an NMI. This feature can be used in a start-up procedure like the following, to tell which serial controller channel is used for the Console:

For each serial controller channel that the software can use for the Console:

1. Initialize the channel.
2. Send a NUL character to the channel.
3. Wait a short time to see if an NMI occurs. If so, the current channel is the Console. If not, go on to the next serial channel and try again.

If none of the allowed serial channels produces an NMI, the user has not properly jumpered any J5-J10 connector block to the J13 block.

Basic software should use the serial controller channel for the Console in a very basic, polled way. Because of this and because of similarities between the (E)SCC and the ISC, and between the (M)USC and the IUSC, note that software allows the Console to be connected to either the (E)SCC channel A or to the (M)USC; in fact, it includes most of the code necessary to use any of the six serial controller channels for the Console.

Notes on J4/Macintosh/AppleTalk/LocalTalk

The J4 connector is similar to that offered on various Macintosh systems. The ESCC and ISCC are particularly well adapted for use with this port, and development of USC family capability for AppleTalk/LocalTalk is of interest.

The J3 and J4 connectors cannot be used simultaneously. The J16 jumper block controls whether the RS-422 driver for Transmit Data is turned "on" and "off" under control of the associated Request to Send signal, as on the Mac, or is "on" full time, which is more suitable for the use of J3. To put the TxD driver under control of RTS, jumper J16-1 to J16-J2 and leave J16-J3 open. For full-time drive on TxD (and also the J3 RTS pins), jumper J16-J2 to J16-J3 and leave J16-J1 open.

The J17 jumper block controls whether the reception of Data Carrier Detect and Clear to Send is differential (on J3) or unbalanced, as on J4. To use differential signalling from J3, remove all jumpers from J17.

On the initial Macintosh and subsequent ones as well, Apple did the unbalanced signalling backward from standard RS-423 and RS-232 polarity for the CTS lead (also called HSK and HSKI). If you are developing code for Macintosh hardware, you can preserve Mac compatibility by jumpering J17-J3 to J17-J5 and J17-J4 to J17-J6. This grounds the CTS- lead and connects the CTS+ lead to J4-J2. It also (assuming a standard source at the other end) inverts CTS to the opposite sense from that expected by the serial controller for functions such as auto-enabling. To make the CTS input of the serial controller have its normal (low-true) sense, jumper J17-J3 to J17-J4, and J17-J5 to J17-J6— this grounds the CTS+ lead and connects the CTS- lead to J4-J2.

The DTR (HSKO) output is provided in Apple systems from Mac Plus onward and has standard RS-423 (and RS-232) polarity.

The DCD input on J4-J7 is provided in Apple systems from the Mac II and SE onward, and also has standard polarity on Apple hardware. Jumper J17-J1 to J17-J2 to ground the "+" input of the receiver; the "-" lead is connected to J4-J7.

SERIAL INTERFACING (Continued)

With jumpers installed to make DCD and CTS unbalanced, J4 can also be used for an additional RS-232 serial link. Connect a "Mac to Hayes modem" cable to J4, and optionally a null modem interconnect module to the other end. The cable internally grounds the RxD+ and TxD+ leads so that RxD- and TxD- act like RS-232 signals.

Macintosh systems also include provisions for synchronous clock inputs. It is not known whether these features are used by any applications, or attached hardware. On all known Macs, the SCC's TRxC pin is driven from the same signal as CTS; to be compatible with this feature, connect J15-J4 to pins 4 and 9 of the selected connector among J5-J10.

On the Mac SE, Mac II, and later models, a multiplexing scheme is provided on SCC channel A's RTxC pin to drive from either the same signal as DCD, or from an on-board 3.672 MHz clock. (Channel B always had the 3.672 MHz clock.) The former capability can be provided by connecting J15-J6 to pins 6 and 8 of the selected connector among J5-J10. The latter capability can be only approximated using the 80186 clock with different baud rate divisors, or by using another oscillator. (The board includes an unpopulated 4-pin oscillator socket that might be useful in this regard.)

JUMPER SUMMARY

Table 12 includes only those connector blocks intended to be populated by 2-pin option jumpers. J1-J15 and J26 are actual connectors meant for use with cables, jumper wires, or wire-wrapped connections.

Table 12. Two-Pin Option Jumpers

| Jumpers | Installed | Open |
|--------------------------------|---|--|
| J9-J7 thru -9 | 7 to 8: 80186 SYSCLK is IUSC /RxC 7 to 9: 80186 SYSCLK is IUSC /TxC | 8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C |
| J10-J7 thru -9 | 7 to 8: 80186 SYSCLK is MUSC (USC A) /RxC 7 to 9: 80186 SYSCLK is MUSC (USC A) /TxC | 8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C |
| J12-J7 thru -9 | 7 to 8: 80186 SYSCLK is USC B /RxC 7 to 9: 80186 SYSCLK is USC B /TxC | 8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C |
| J16-J1 thru -3 | 1 to 2: J3, J4 TxD driven when RTS 2 to 3: J3, J4 TxD, RTS driven full-time | Must install one or the other |
| J17-J1 to -2 J17-J3 thru -6 | Unbalanced DCD- on J3 or J4 3 to 5 and 4 to 6: CTS+ on J4-J2 3 to 4 and 5 to 6: CTS- on J3 or J4 | Differential DCD+, DCD- on J3 Differential CTS+, CTS- on J3 |
| J18-J1 thru -3 | 1 to 2: 2764, 27128, 27256 EPROMs 2 to 3: 27512 EPROMs | Must install one or the other |
| J19-J1 thru -6 | 1 to 2 and 4 to 5: 128K x 8 SRAMs 2 to 3 and 5 to 6: 32K x 8 SRAMs | Must install one way or the other |
| J20-J1 thru -3 | 1 to 2: U2 contains 80C30 or 80230 2 to 3: U2 contains 85C30 or 85230 | Must install one way or the other |
| J21-J1 thru -6 | 1 to 2 and 4 to 5: U2 contains 80C30 or 80230 2 to 3 and 5 to 6: U2 contains 85C30 or 85230 | Must install one way or the other |
| J22-J1 thru -4 | 1 to 2: MUSC (USC A) RxREQ on DMA 0 1 to 3: MUSC (USC A) RxREQ on DMA 1 2 to 4: MUSC (USC A) TxREQ on DMA 0 3 to 4: MUSC (USC A) TxREQ on DMA 1 | 1: MUSC (USC A) Rx no DMA 4: MUSC (USC A) Tx no DMA |
| J23-J1 thru -3 | 1 to 2: (E)SCC B RxRQ on DMA 0 2 to 3: (E)SCC B Wait function | (E)SCC B neither Rx DMA nor Wait |
| J24-J1 thru -4 | 1 to 2: clipped SCC B TxREQ on DMA 1 1 to 3: direct ESCC B TxREQ on DMA 1 3 to 4: /DTR output from ESCC B | (E)SCC B neither Tx DMA nor /DTR |
| J25-J1 thru - 5 and J25X | 1 to 2 and 3 to 4: (E)SCC last on IACK chain, MUSC second to last J25X to 2 and 3 to 4: (E)SCC last, USC 2nd to last 2 to 3 and 4 to 5: (E)SCC first on IACK chain | Must be one of these three ways |
| J28-J1 thru -6 | 1 to 2: 80186 SYSCLK is (E)SCC PCLK 3 to 4: 80186 SYSCLK is ISCC PCLK 5 to 6: 80186 SYSCLK is IUSC CLK | Connect some other clock to 2, 4, or 6 |
| J29-J1 thru -4 | 1 to 2: USC B RxREQ on DMA 0 1 to 3: USC B RxREQ on DMA 1 2 to 4: USC B TxREQ on DMA 0 3 to 4: USC B TxREQ on DMA 1 | 1: USC B Rx no DMA 4: USC B Tx no DMA |

DMA/EPLD LOGIC

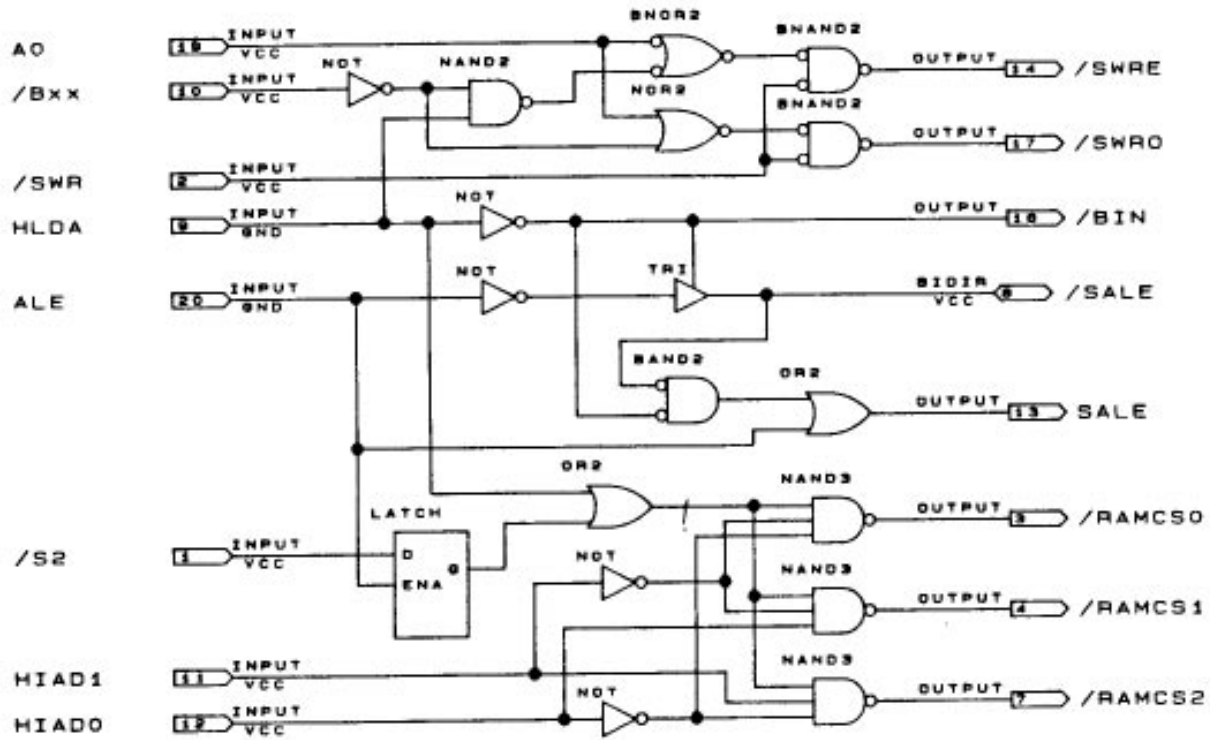


Figure 1. Control EPLD for 186 Board

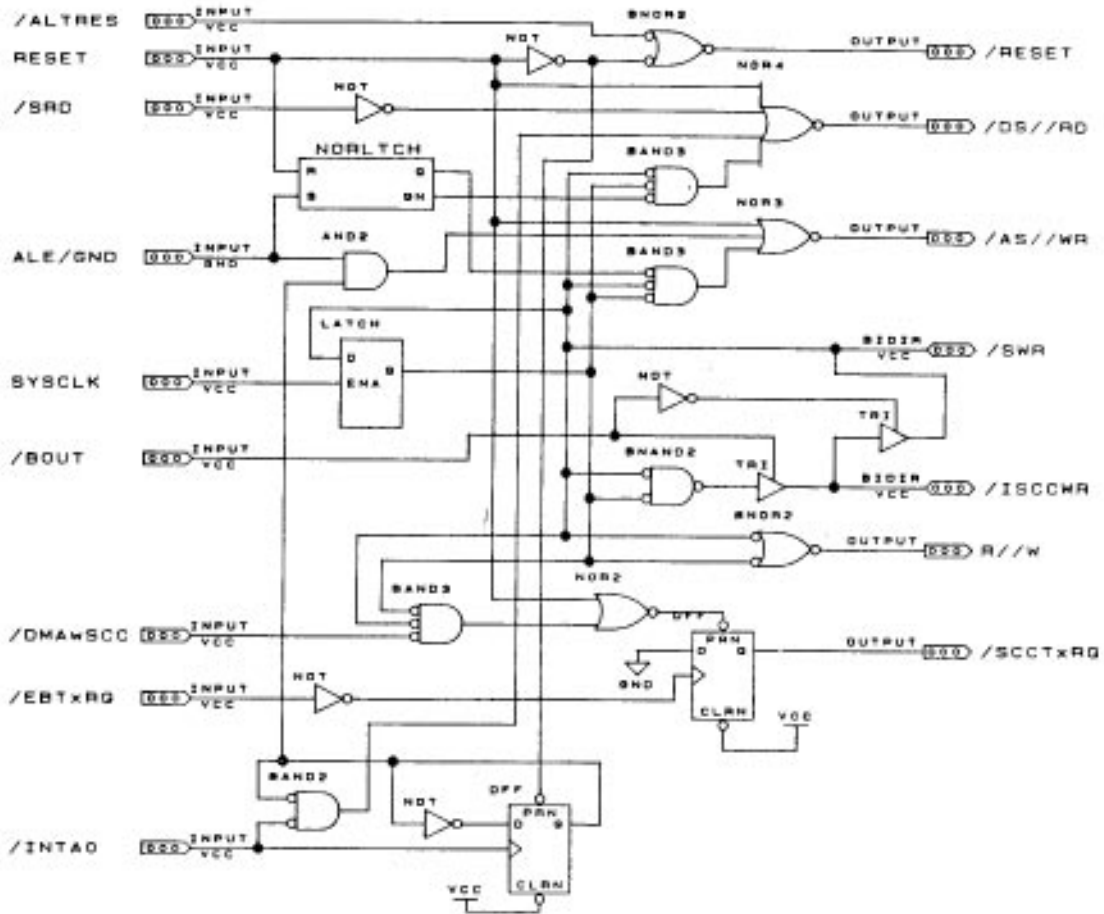


Figure 2. SCC EPLD for 186 Board

DMA/EPLD LOGIC (Continued)

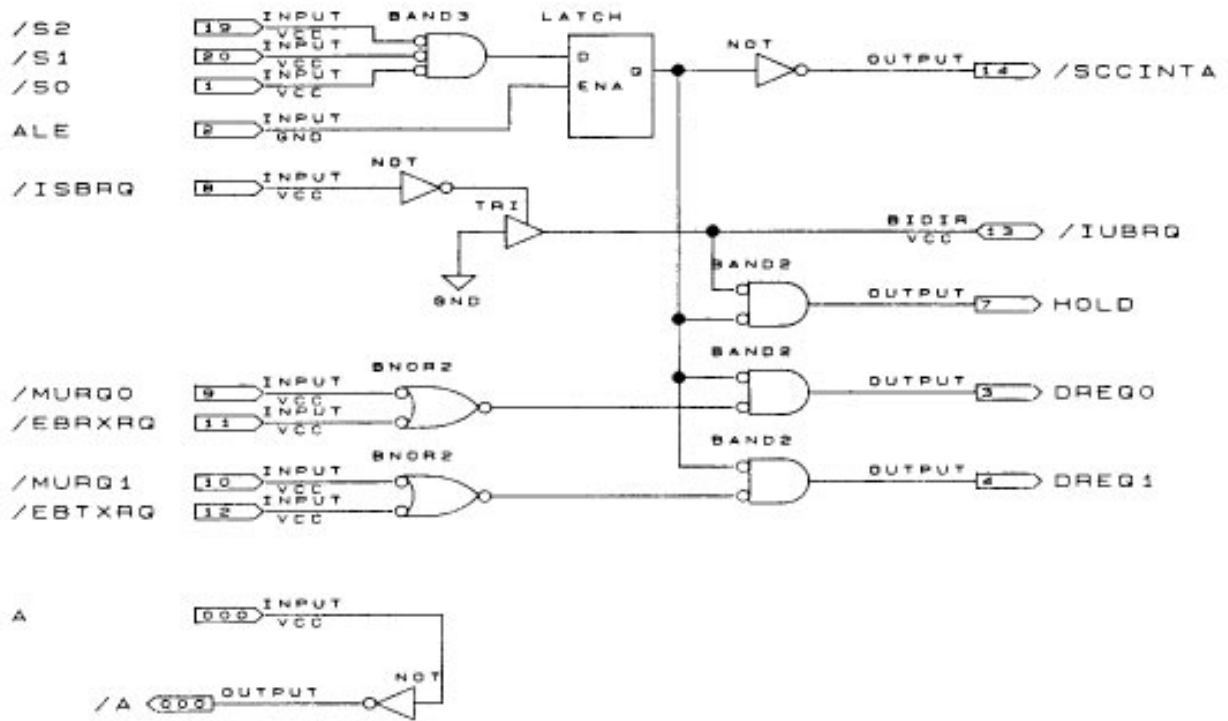


Figure 3. DMA EPLD for 186 Board

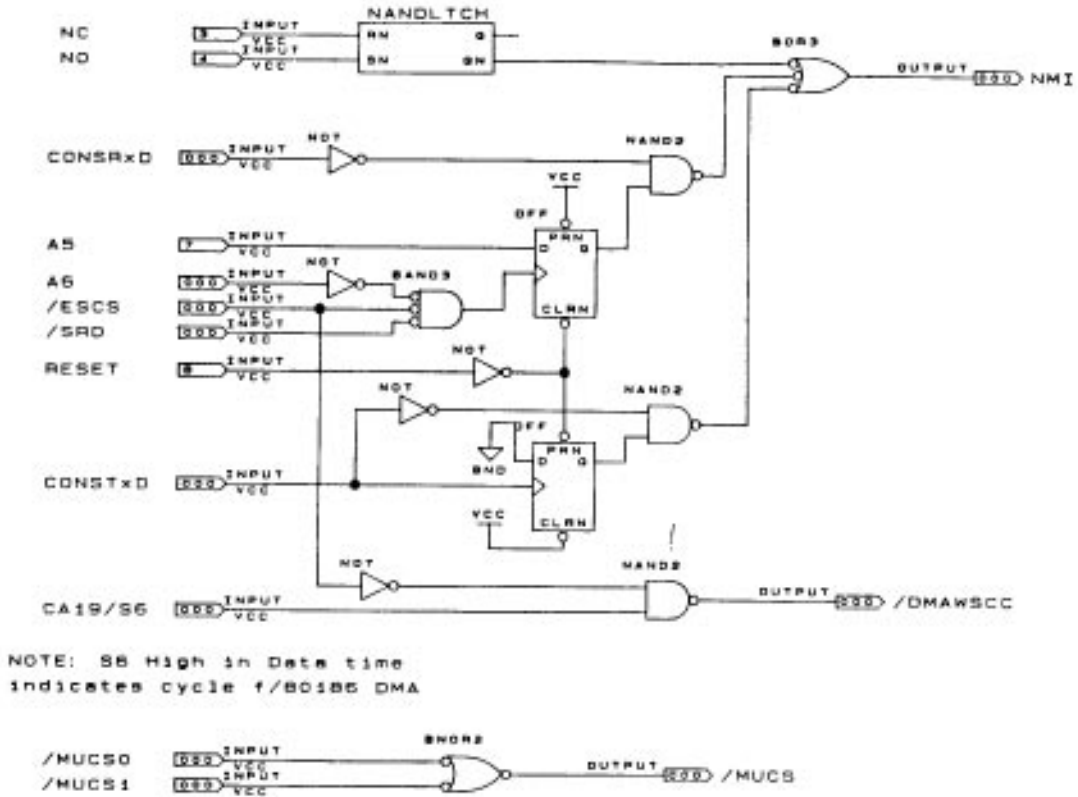


Figure 4. NMI Field for 186 Board

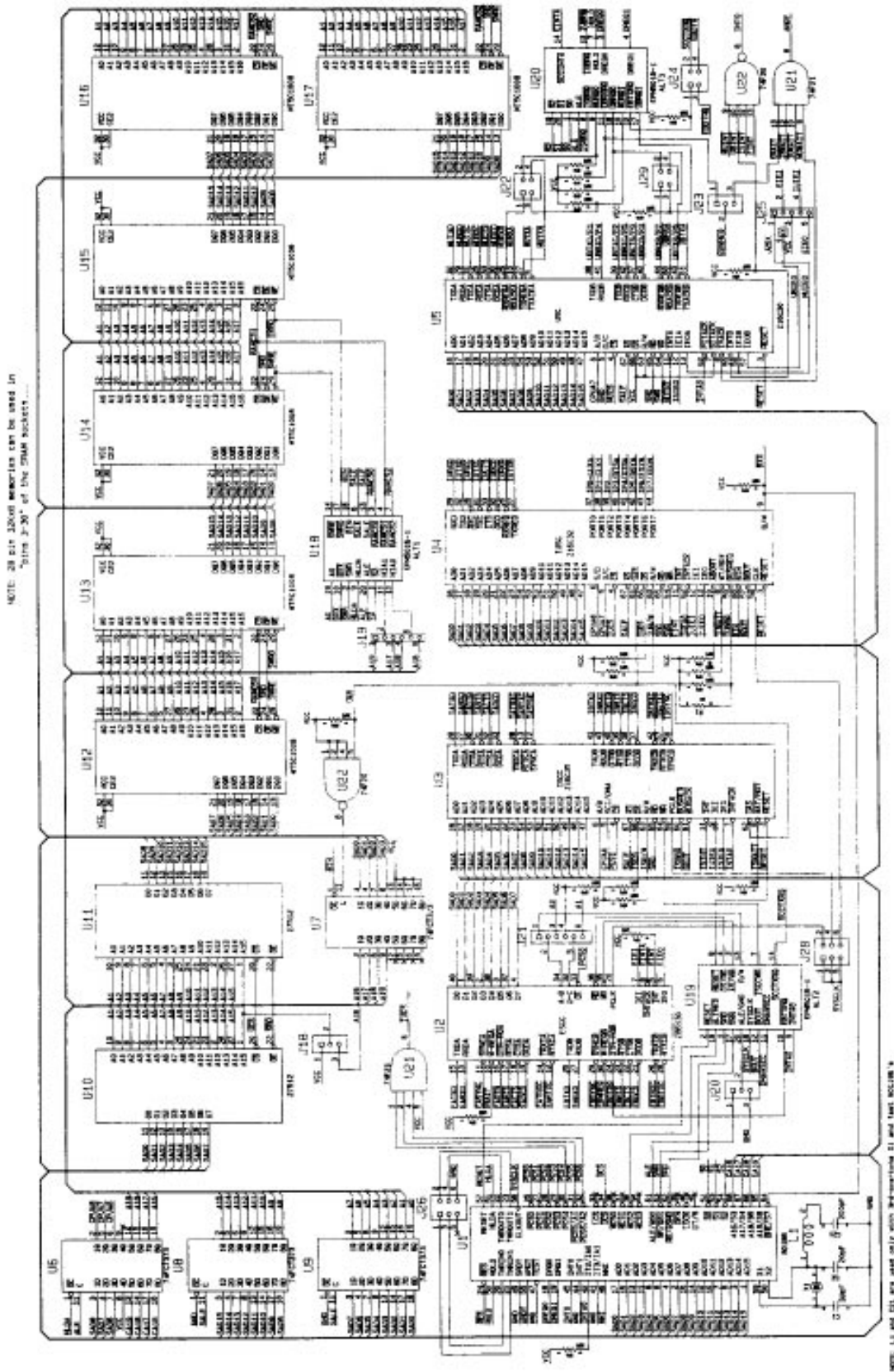


Figure 5. Schematic of the Evaluation Board