ADVANCED DIGITAL

MULTI SLAVE

Product Reference Manual

PRELIMINARY COPY

ADC
ADVANCED DIGITAL CORPORATION

USA OFFICE
5432 PRODUCTION DRIVE
HUNTINGTON BEACH, CA 92649
TELEPHONE : (714) 891-4004
TELEX: 183210 ADVANCED HTBH

UNITED KINGDOM OFFICE
27 PRINCESS STREET
HANOVER SQUARE, LONDON W1R8NQ
UNITED KINGDOM
409-0077 / 409-3351
TLX 265840 FINEST

MULTI SLAVE PRODUCT REFERENCE MANUAL

Revision A.0

Advanced Digital Corporation
5432 Production Drive
Huntington Beach, California
92649 USA
Telephone: (714) 891-4004
Telex: 183210 ADVANCED HTBH

Advanced Digital (U.K.) Ltd.
27 Princess Street
Hanover Square
London, U.K.
(01) 409-0077

CONTENTS

Illustrations:

Tables:

SECTION I

INTRODUCTION

This Product Reference Manual provides information to install, utilize and maintain the Advanced Digital Corp.'s Multi Slave S-100 bus compatible computer. The Multi Slave is a single printed circuit board, conforming to the IEEE-696 form factor of 5.25 inches by 10 inches (13.33 cm by 25.4 cm), providing three independent 8 MHz Z80 central processors, each with two 64k byte banks of memory, two serial I/O ports, and a counter/timer.

UNPACKING/PACKING INSTRUCTIONS

When the Multi Slave is delivered by a transfer company, it must be carefully inspected for damage. Prior to accepting delivery, carefully inspect the shipping container for obvious damage. If damage is evident, note it on the waybill and require that the delivery agent sign the waybill. Notify the transfer company immediately, and submit a damage report to the carrier.

Remove the Multi Slave and any accessory items from the shipping container. Retain the shipping container any packing material for possible reshipment. Leave the Multi Slave in its anti-static envelope until installation time.

INSTALLATION

After verifying that the intended enclosure for the Multi Slave will provide adequate power and air flow, remove the Multi Slave from its anti-static envelope. Inspect and verify that the configuration jumpers on the Multi Slave are correct for the Multi Slave's intended utilization. Attach I/O cables to the connectors provided for serial I/O as required. Insert the Multi Slave printed circuit board into a suitable S-100 bus slot connector.

WARRANTY AND WARRANTY RETURN PROCEDURE

Advanced Digital Corporation warrants that its products will be free from defects in material and workmanship for a period of 360 days of shipment from the factory.

If a customer experiences a defect in either workmanship or materials during the warranty period, notify your supplier immediately. Your supplier may repair the Multi Slave or determine if some other action is to be taken. In the event that a return of the Multi Slave is deemed neccessary, obtain a RETURN MATERIAL AUTHORIZATION (RMA) NUMBER from your supplier.

Repack the Multi Slave and any accessory items in the original packing material and ship it in accordance with your supplier's shipping instructions. Make sure the RMA number is clearly marked on the shipping label. Your supplier will not accept delivery of a return shipment without the proper RMA number.

****
* W A R N I N G *
****

The Multi Slave as delivered does not generate, use, or radiate radio frequency energy. However, after installation and application of power, the Multi Slave may generate, use, or radiate radio frequency energy. Advanced Digital Corporation recommends that the Multi Slave be installed in an enclosure which complies with the provisions for computing devices pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference.

FEATURES AND OPTIONS

The Multi Slave provides the following features:

- Three independent 8-bit, 8 MHz Z80 Microprocessors
- Each CPU has 128k of memory, configured as two 64k byte banks, with a 1k to 16k area of common (shared) memory.
- Two asynchronous serial I/O ports per CPU. Level conversion to EIA RS-232C or RS-422 standards is provided via the PS/NET (paddle card) accessory.
- Independent baud rate selection on each serial port
- Counter/Timer providing real time clock capability
- User selectable PROM, jumper configurable for one of the following: 2716, 2732, 2764, 27128, or 27256.
- IEEE-696 S-100 Bus Compatibility
- Automatic power-on/reset bootstrap loader and resident monitor/debugger utility.

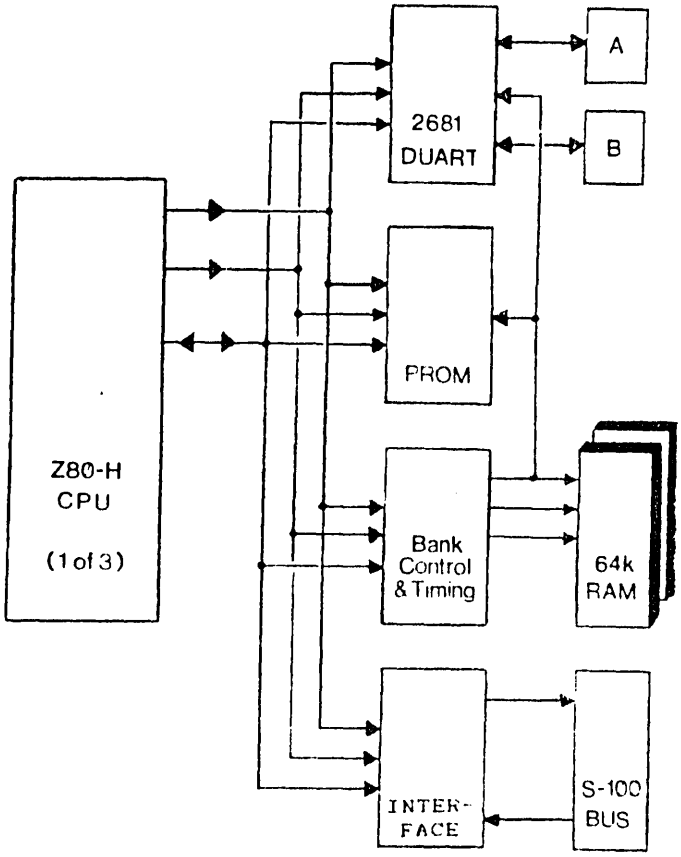


Figure 1-1 Multi Slave Block Diagram

FUNCTIONAL FLOW

Figure 1-1 illustrates the major functional components of the Multi Slave computer board. Initially, immediately following the power-on/reset event, all three processors are held in a reset state. Each processor must be individually activated by a network master. Once the master has activated a Multi Slave processor, the selected slave CPU then begins executing the instructions provided by the onboard EPROM. Depending upon user response and intervention, the processor will either initiate the execution of the resident Monitor/Debug program, or begin the download request sequence to receive an operating system.

MULTI SLAVE PRODUCT SPECIFICATION

Physical and Environmental:

Form Factor	IEEE-696 S-100 standard
Size	5.25 inches x 10.0 inches x .75 inches
Weight	14 oz.
Temperature	
operating	0 to 50 degrees Celcius
storage	-65 to 150 degrees Celcius
Humidity	0 to 95%, non-condensing
Altitude	0 to 10,000 feet (operating)

Power Requrements:

	+5VDC @ xx.x Amps (xx Watts)
Power Regulation	On board, providing 4.75 to 5.25VDC
Cooling	1-5 CFM (cubic feet per minute) air flow

Power Supply Requirements:

Unregulated	+ 7VDC Minimum
	+11VDC Average
	+25VDC Maximum (peak)
	+14.5VDC Minimum
	+21.5VDC Average
	+35.0VDC Maximum (peak)
	-14.5VDC Minimum
	-21.5VDC Average
	-35.0VDC Maximum (peak)

Functional Specification:

Processor	Zilog Z80H
Memory	128k Dynamic RAM
Processor Clock	8.000 MHz
Serial Controller	Signetics 2681 DUART

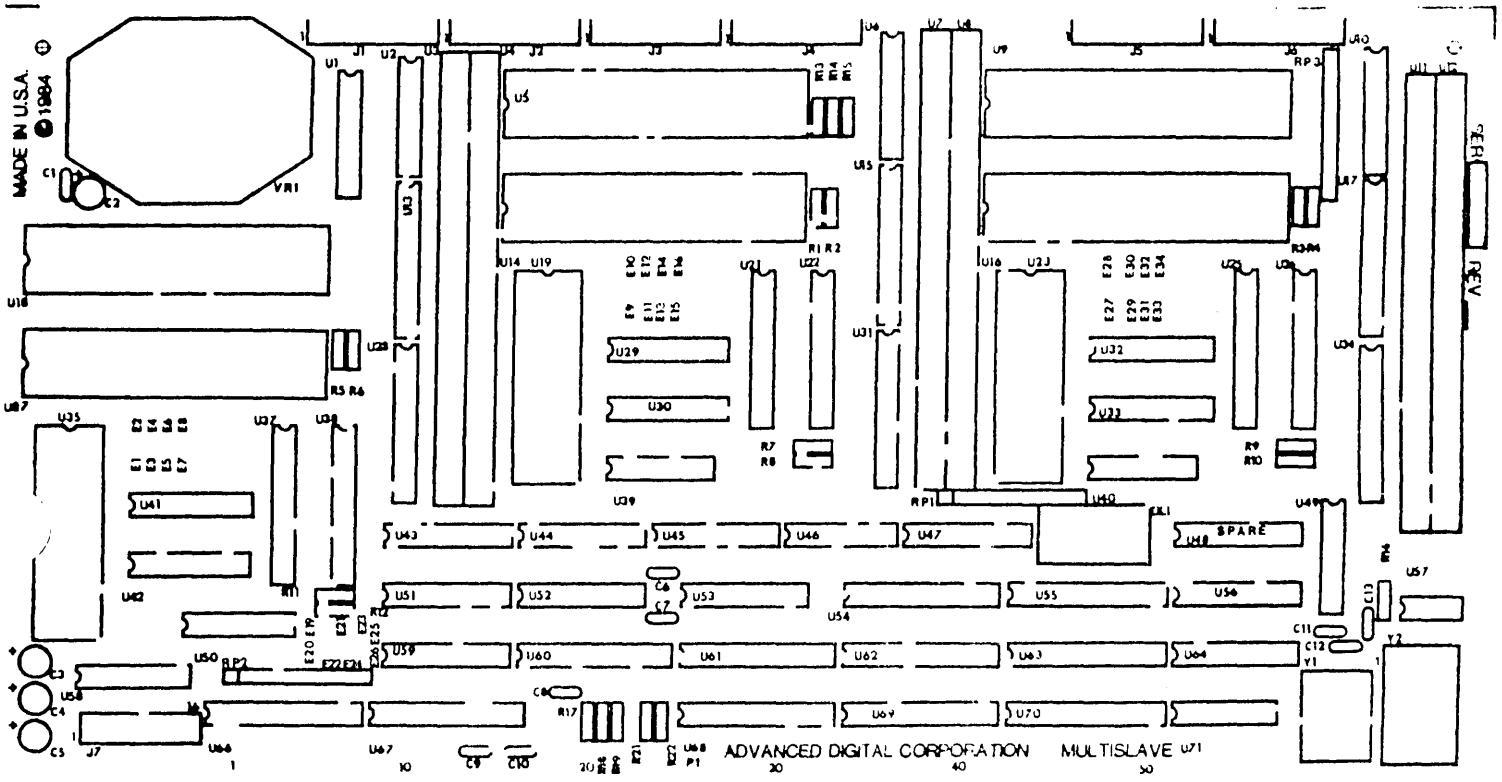


Figure 2-2 Multi Slave Component Layout

The factory standard configuration of the Multi Slave uses a 2764 EPROM, and does not utilize the S-100 Vectored Interrupt Lines. The base address of the board is jumpered for I/O address 90H at jumper block E-19 thru E-26 (lower left quadrant).

GENERAL NOTES

The Multi Slave contains three independent Zilog Z80-H CPUs. The principal I/O device for each processor is the Signetics 2681 DUART (Dual Universal Asynchronous Receiver/Transmitter). This particular device was chosen for its many features, which include an internal dual baud rate generator and counter/timer.

Several types of PROMs are supported by the Multi Slave, based on jumper options. The PROM types supported are as follows: 2716, 2732, 2764, 27128 and 27256. PROM addressing begins at 0000H.

Each processor contains a total of 128k of user available memory, configured as two 64k banks, and is selectable through software.

There is only one interrupt source on the Multi Slave, which is provided by the DUART. Typically, the Z80 CPU will be operated using Mode 1 interrupts (see the Zilog Z80 Technical Manual for a discussion of Z80 interrupt modes).

Each slave is also capable of interrupting the master via the S-100 vectored interrupt lines. This will be discussed in detail later.

Each Z80 CPU is driven by a central 8 MHz oscillator; the three DUARTs are driven by a common 3.6864 MHz crystal.

SECTION II

POWER-ON/RESET EVENTS

Each Multi Slave CPU is held in a reset state until such time as the master enables the operation of a slave. The master enables a slave by issuing an output byte to the slave's base address with bit D4 reset. The PROM is automatically selected, and instruction execution begins at address 0000H.

The ADC supplied PROM initializes both channels of the DUART to 9600 baud, then tests channel A for an available input character. If a character is available, the Multi Slave monitor is activated which issues the following message to the terminal on channel A:

```
Multi Slave Monitor Version 1.x  Generated MM-DD-YY
Copyright (C) 1985  Advanced Digital Corporation

Enter '?' for HELP

*
```

A complete discussion of the Multi Slave monitor may be found in section IV of this manual.

If an input character is not available at serial channel A within approximately 100 milliseconds, the startup program will enter the cold boot process, sending an operating system download request to the master processor over the S-100 bus.

PROM SELECTION JUMPERS

Each CPU has a jumper block to select the PROM type. Use the following table to set these jumpers.

E1-6 E9-16 E27-34

```
1 -- o  o -- 2
3 -- o  o -- 4
5 -- o  o -- 6
7 -- o  o -- 8

TYPICAL JUMPER
     BLOCK
```

PROM TYPE	JUMPER
2716 (2K)	3-5, 4-6
2732 (4K)	5-7, 4-6
2764 (8K)	1-3, 5-7
27128 (16K)	1-3, 6-8, 5-7
27256 (32K)	1-2, 6-8, 5-7

Figure 2-1 Multi Slave PROM Selection Jumpers

INTERRUPT SELECTION JUMPERS

Each Multi Slave CPU may interrupt the master processor on one of two S-100 vectored interrupt lines. Selection of interrupts is as follows:

J7

MULTI SLAVE	JUMPER BLOCK	S-100 BUS
Multi Slave CPU 0	1 -- o o -- 2	INT0* (pin 4)
" " " "	3 -- o o -- 4	INT1* (pin 5)
Multi Slave CPU 1	5 -- o o -- 6	INT2* (pin 6)
" " " "	7 -- o o -- 8	INT3* (pin 7)
Multi Slave CPU 2	9 -- o o -- 10	INT4* (pin 8)
" " " "	11 -- o o -- 12	INT5* (pin 9)
No Connection	13 -- o o -- 14	INT6* (pin 10)
" "	15 -- o o -- 16	INT7* (pin 11)

Figure 2-2 Multi Slave Interrupt Jumper Selection

If the interrupt selection structure shown above is not suitable, wire wrap connections between the jumper block pins may be used in place of jumper plugs, thus allowing any given Multi Slave CPU to use any of the eight S-100 vectored interrupt lines.

SLAVE BASE ADDRESS SELECTION

Each Multi Slave CPU occupies four of the master's I/O ports, though only three are actually used by each CPU. Base address selection of the Multi Slave board is as follows:

E19 - E26

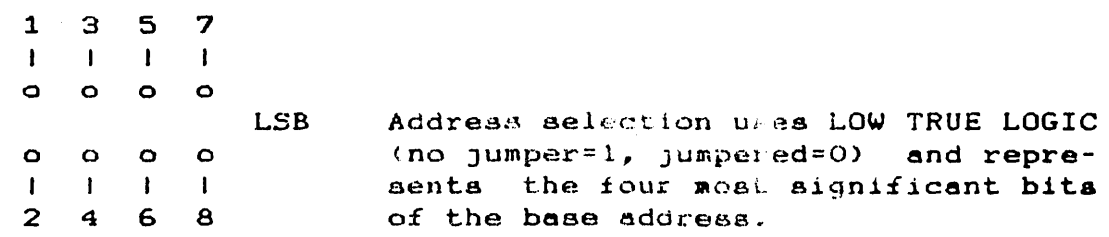


Figure 2-3 Multi Slave Base Address Selection

Example:

BASE ADDRESS	JUMPERS
70H	1-2
80H	3-4, 5-6, 7-8
90H	3-4, 5-6
A0H	3-4, 7-8
B0H	3-4
C0H	5-6, 7-8
D0H	5-6

If, for example, the base address was 80H, the three CPUs would be addressed as follows:

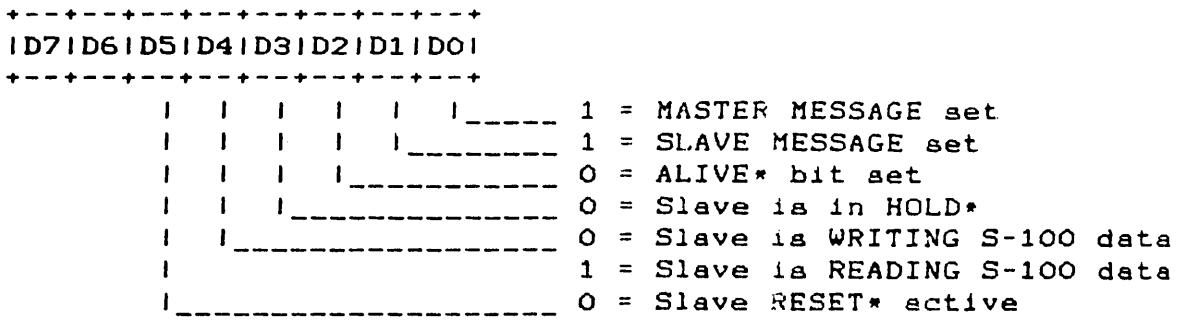
```
CPU #0          80H  ; uses 80H, 81H, and 82H (83H not used)
CPU #1          84H  ; uses 84H, 85H, and 86H (87H not used)
CPU #2          88H  ; uses 88H, 89H, and 8AH (8BH not used)
```

The master (S-100) data port assignments are as follows:

BASE ADDRESS	TYPE	FUNCTION
+0	R/W	Read Slave Status (see details below) Write Control bits
+1	R/W	S-100 Data Port
+2	R/W	Read = De-assert ALIVE* bit Write= De-assert SLAVE MESSAGE bit

Table 2-1 Master S-100 Port Assignments

SLAVE STATUS PORT (read by Master Processor)



STATUS PORT (written by Master Processor)

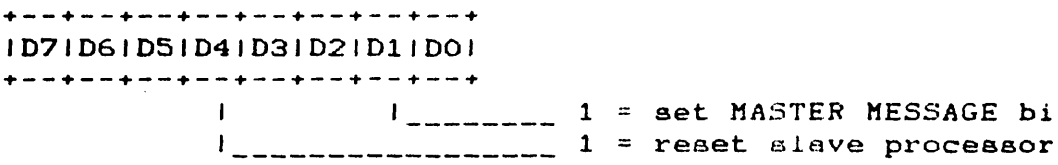


Figure 2-4 Master Status Port Bit Defintions

Note that bits D6 and D7 of the STATUS PORT are not used, and are on (1) when the port is read by the Master Processor. Note also that there are four low active terms; ALIVE*, HOLD*, WRITE*, and RESET*.

MASTER/SLAVE COMMUNICATIONS

For the master processor to send data to the slave, the slave must be reading the S-100 data port. Conversly, for the slave to send data to the master, the slave must write to the S-100 data port before the master reads it. As mentioned earlier, when the slave is reading or writing to the S-100 data port, it is placed in a HOLD (wait) condition, and released to resume processing when the Master Processor writes or reads the corresponding port. This ensures that the master and slave(s) remain in synchronization during bus communication; reliable network operation is achieved when the other status bits (ALIVE*, READ/WRITE*, and HOLD*) are implemented as part of a communications protocol.

The Master Processor may reset a slave processor at any time by issuing an output to the appropriate slave status port with bit D4 set. To reactivate the slave, the master must again issue an output byte to the slave's status port with bit D4 reset, followed by a short post-reset delay. An example of this sequence is shown below:

```
SL_RES:   LD   A,10H          ; SET BIT D4
          OUT  (SLVBASE),A    ; OUTPUT TO SLAVE BASE ADDRESS
          LD   B,16           ; LEAVE THE BIT SET ...
SL_R05:   DJNZ SL_R05         ; ... FOR A SHORT TIME
          XOR  NOT 10H        ; TURN OFF BIT D4
          OUT  (SLVBASE),A    ; UN-RESET THE SLAVE
          LD   B,16           ; SET UP FOR POST-RESET DELAY
SL_R10:   EX   (SP),HL        ;  THIS IS A
          EX   (SP),HL        ;   VERY EFFECTIVE
          DJNZ SL_R10         ;    TIME WASTER
          RET                 ; RETURN TO CALLER
```

The suggested handshake mechanism between the master and slave is as follows (for transmission from slave to master): the slave first asserts the SLAVE MESSAGE bit. The master may see this by polling (reading the slave status port), or the Multi Slave may be configured to cause an interrupt on the master. In either case, once the SLAVE MESSAGE bit has been asserted, the slave CPU then outputs its first data byte to the S-100 bus communications port. Upon doing so, the slave CPU is forced into a wait condition, which the master must verify by testing the WRITE* and HOLD* bits at the slave status port. Once the master has determined that both signals are true, it may then read the byte waiting at the S-100 data port. The slave is released to resume processing, free to continue sending subsequent data bytes, etc.

In a master to slave transmission, the master must assert the MASTER bit, which the slave sees by polling its' status port. Upon detecting the active MASTER bit, the slave must reset it

(indicating to the master that the slave is ready to accept the message), and immediately issue a read to the S-100 data port. Again, the slave processor is forced into a wait condition. which the master must verify by testing the READ and HOLD* bits at the slave status port. Once the master has determined that both signals are active, it may then write the data byte to the S-100 data port. The slave is released to resume processing, free to continue receiving subsequent data bytes, etc.

The following page provides a brief example of this master/slave dialogue:

MASTER PROCESSOR (RECEIVING MESSAGE FROM SLAVE)

```
      IN    A,(SLVSTAT)     ; GET SLAVE STATUS
      BIT   SLVMSG,A        ; IS SLAVE MESSAGE BIT SET?
      RET   NZ              ; NO, RESUME OTHER TASKS (ACTIVE LOW)
      OUT   (MSGRST),A      ; OUTPUT CLEARS SLAVE MESSAGE BIT
                            ; (CONTENTS OF A-REG INSIGNIFICANT)
WAIT1:
      IN    A,(SLVSTAT)     ; GET SLAVE STATUS BITS
      BIT   SLVWRT,A        ; IS SLAVE WRITING TO S-100 PORT?
      JR    NZ,WAIT1        ; LOOP IF NOT
      BIT   SLVHLD,A        ; IS SLAVE IN A HOLD CONDITION?
      JR    NZ,WAIT1        ; LOOP IF NOT
      IN    A,(SLVDATA)     ; ALL IS READY - GET THE DATA BYTE
      RET                   ; AND RETURN TO CALLING TASK
```

MASTER PROCESSOR (SENDING MESSAGE TO SLAVE)

```
      LD    A,00000010B     ; SET MASTER MESSAGE BIT...
      OUT   (SLVSTAT),A     ; ...SO THE SLAVE WILL SEE IT
WAIT2:
      IN    A,(SLVSTAT)     ; GET SLAVE STATUS
      BIT   MASTER,A        ; HAS THE SLAVE RESET IT YET?
      JR    NZ,WAIT2        ; LOOP UNTIL HE HAS DONE SO
WAIT3:
      IN    A,(SLVSTAT)     ; GET SLAVE STATUS AGAIN
      BIT   SLVWRT,A        ; IS SLAVE READING THE S-100 PORT?
      JR    Z,WAIT3         ; LOOP IF NOT
      BIT   SLVHLD,A        ; IS SLAVE IN A HOLD CONDITION?
      JR    NZ,WAIT3        ; LOOP IF NOT
      LD    A,(HL)          ; GET BYTE TO SEND
      OUT   (SLVDATA),A     ; SEND IT TO THE SLAVE
      RET                   ; AND RETURN TO CALLING TASK
```

SLAVE PROCESSOR (RECEIVING MESSAGE FROM MASTER)

```
      IN    A,(MASTER)      ; CLEAR MASTER MESSAGE BIT
      IN    A,(DATAPORT)    ; AND IMMEDIATELY READ THE S-100 PORT
      RET                   ; RETURN TO CALLING TASK
```

SLAVE PROCESSOR (SENDING MESSAGE TO MASTER)

```
      IN    A,(SLVMSG)      ; SET SLAVE MESSAGE BIT
      LD    A,B             ; B-REG HAS DATA BYTE TO SEND
      OUT   (DATAPORT),A    ; OUTPUT DATA BYTE TO S-100 DATA PORT
      RET                   ; RETURN TO CALLING TASK
```

Note that these examples transmit and receive only a single byte at a time and provide only a skeletal communications protocol. The slave may utilize the Z80 block input and output instructions to receive and transmit any number of bytes to the master, though the user should employ a slightly more sophisticated protocol when doing so. In typical networking environments, the first byte transmitted contains the length of the message to follow. Implementation of such networking schemes is left to the user.

BOOTSTRAP SOFTWARE

The following program will provide the user with a bootstrap facility which follows the protocol set forth in the prior section. This program is identical to that contained in the Multi Slave monitor PROM, and assumes that an intermediate loader will be received from the master.

```
SLVMSG    EQU  20H        ; I/O READ SETS SLAVE MESSAGE BIT
SALIVE    EQU  60H        ; I/O READ SETS SLAVE ALIVE BIT
SDATA     EQU  0070H      ; B=MESSAGE LENGTH, C=PORT ADDRESS
SHMEM     EQU  8000H      ; ONLY ADDRESSES ABOVE 8000H ARE
                          ; ACCESSABLE WHILE PROM IS ENABLED
BOOT:
     IN   A,(SLVMSG)      ; SET SLAVE MESSAGE BIT
     IN   A,(SALIVE)      ; SET SLAVE ALIVE BIT
     LD   BC,SDATA        ; B=BYTE COUNT (0=256 BYTES)
                          ; C=PORT ADDRESS
     OUT  (C),B           ; SEND BYTE COUNT FIRST
     LD   HL,SHMEM        ; ADDRESS TO STORE INCOMING MESSAGE
                          ; (WHICH WILL BE THE INTERMEDIATE
                          ; LOADER PROGRAM)
     INIR                 ; RECEIVE ENTIRE MESSAGE FROM MASTER
     JP   SHMEM           ; BRANCH & EXECUTE INTERMEDIATE LOADER
```

I/O ADDRESS MAP

Each Multi Slave CPU has an identical I/O address map. The port assignments are as follows:

ADDRESS (Hex)	TYPE	FUNCTION
00-0F	R/W	DUART Data/Control (see Signetics 2681 Documentation for details)
10-1F	---	Not Used
20-2F	R	Asserts SLAVE MESSAGE bit
30-3F	R	De-asserts MASTER MESSAGE bit
40-4F	W	BANK/PROM SELECT (details follow)
50-5F	R	STATUS PORT (MASTER and SLAVE bits)
60-6F	R	Asserts SLAVE ALIVE* bit
70-7F	R/W	S-100 DATA PORT

Table 2-2 I/O Port Assignments

The user should be aware that although it would appear that the S-100 data ports would collide with one another, they are actually physically separated in the hardware. This is explained a bit more clearly by the following diagram:

```
SLAVE PROCESSOR NUMBER                    MASTER PROCESSOR
----------------------                    ----------------
#0 -- I/O ADDRESS 70H.....................I/O BASE ADDRESS + 0
#1 -- I/O ADDRESS 70H.....................I/O BASE ADDRESS + 4
#2 -- I/O ADDRESS 70H.....................I/O BASE ADDRESS + 8
```

```
SLAVE STATUS PORT (read by the Slave, read only)
------------------------------

+--+--+--+--+--+--+--+--+
|D7|D6|D5|D4|D3|D2|D1|D0|
+--+--+--+--+--+--+--+--+
                  |  |
                  |  |----- 1 = MASTER message active
                  ---------- 1 = SLAVE message active
```

Figure 2-5 Slave Status Port Bit Definitions

BANK AND PROM SELECT PORT

The BANK/PROM SELECT port specifies which of the two 64k memory banks is to be active, and whether or not the PROM is to be selected. The bit definitions for this port are as follows:

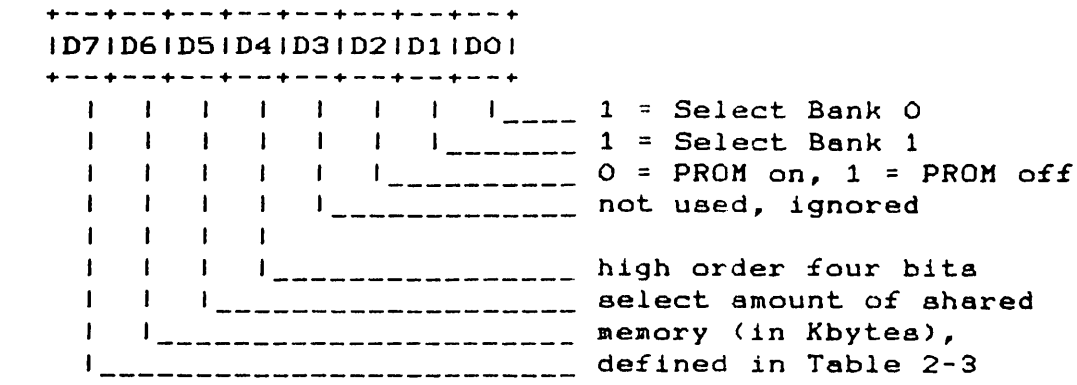


Figure 2-6 Bank and Prom Select Port Bit Definitions

Bits D4-D7 Hex Value	Shared Amount
0	16k
1	15k
2	14k
3	13k
4	12k
5	11k
6	10k
7	9k
8	8k
9	7k
A	6k
B	5k
C	4k
D	3k
E	2k
F	1k

Table 2-3 Bank Select Bit Definitions

NOTE: Bank 0 and 1 are mutually exclusive and must not be set active at the same time. When the PROM is selected, only memory locations 8000H through OFFFFH may be accessed for RAM read and write operations.

SERIAL I/O CONTROLLER

The Multi Slave provides two independent serial I/O channels per CPU, both contained in one serial I/O controller, the Signetics 2681 Dual Asynchronous Receiver/Transmitter (DUART). The DUART contains two internal, independent baud rate generators, capable of producing 18 user selectable baud rates. Level conversion to EIA RS-232 or RS-422 level is provided by an external circuit assembly called the PS-NET.

SERIAL CHANNEL CONNECTORS

Connectors for each serial I/O channel are located across the top of the Multi Slave board (see figure 1-1 for exact placement). Each connector has the following pinout:

PIN NO.	SIGNAL NAME		DIRECTION
1	DCD	Data Carrier Detect	input
2	DSR	Data Set Ready	input
* 3	***	see note below	input/output
4	RXD	Receive Data	input
5	CTS	Clear to Send	input
6	TXD	Transmit Data	output
7	RTS	Request to Send	output
8	DTR	Data Terminal Ready	output
9	CLK	Tx/Rx Clock	input/output
10	GND	Signal Ground	ground
11	N/C		
12	+16 VDC		supply voltage
13	-16 VDC		supply voltage
14	+5 VDC		supply voltage

* On channel A, this term is RNG (ring detect) and may be connected to the ring detect line on a modem. This line has no connected on channel B.

Table 2-4 Serial I/O Cable Connector

COUNTER/TIMER CIRCUIT

The counter/timer circuit is contained on the DUART device. As its name implies, it may be used as a timer (programmable divider) or a counter, providing an appropriate indication when the specified countdown value has reached zero.

It should be noted that unlike the Z80 CTC, the DUART's counter/-timer uses a 16-bit countdown register, providing much greater flexibility in its application.

Appendix H contains the data sheet for the Signetics 2681 device.

SECTION III

FAULT ISOLATION

Fault isolation is the process of identifying a fault and the resultant cause of the fault to the lowest possible level. This section deals with hardware fault isolation and is generally independent of software considerations.

Prior to any attempt at fault isolation, a test environment must be validated. Generally, the test environment will consist of an S-100 chassis, motherboard, power supply, S-100 extender card, and a known good Multi Slave. Validation consists of removing all other S-100 circuit cards from the chassis, and any other devices loading the +8, +16, and/or -16 VDC power supplies.

Having removed all circuit boards from the motherboard, verify that the following voltages referenced to ground (S-100 bus pin 50 and 100) are within the tolerances listed below:

S-100 PIN	DEFINITION	MINIMUM	AVERAGE	MAXIMUM
1	+ 8 VDC	+7.0	+11.0	+25.0
51	+ 8 VDC	+7.0	+11.0	+25.0
2	+16 VDC	+14.5	+21.5	+35.0
52	-16 VDC	-35.0	-21.5	-14.5

The above conditions must be met before proceeding with the next test.

Step 1: Visual Verification

Inspect the suspect Multi Slave to verify that components are correctly installed and properly seated in their sockets. Components may be compared against a known good Multi Slave. All DIP components have the same pin 1 orientation.

Step 2: On Board +5 VDC Regulation

Remove power from the motherboard. Insert the S-100 extender card into a suitable slot in the motherboard, then insert the Multi Slave into the extender card socket. Apply power and measure the voltage at U1-16. This voltage must be between 4.75 and 5.25 VDC.

Step 3: Clock Verification

Verify the clock frequencies at the following locations:

U57-7	80ns	central clock distributed to all CPU's.
U49-8	150ns	central SIO clock distributed to all DUART's.

Step 5: Memory Verification

Verify memory row and column addressing by monitoring pin 1 and pin 19 on each AM2965 memory driver circuit.

Step 6: Monitor Verification

Data Set Ready (J1-2, J3-2, J5-2) MUST be low for the monitor to issue a message to the console. Verify that the PS/NET-1 card is properly configured and connected to the Multi Slave, and that the terminal's baud rate is set to 9600. Ensure that the Multi Slave card is in a reset-hold condition by pressing the system reset button on the computer's front panel. Then, using a monitor program on the Master Processor, activate the desired slave CPU by issuing an OUT (slave command port),40H. The Multi Slave monitor should issue its logon message as described in Section 2 of this manual. Press any key on the slave console within two seconds after "un-resetting" it. The Multi Slave monitor should then be ready to accept commands.

STEP 7: Verification of other Major Components

Verification of other major Multi Slave components requires development of short software routines which will provide scope loops to support the analysis of Multi Slave signals some of the more commonly required routines have been incorporated in the Multi Slave Monitor program. These tests include memory and I/O read and write loops. See the monitor command list for further information.

```
* O9X 1Ø
* I9X
DE
* O9X Ø2
```

SECTION IV

THE MULTI SLAVE MONITOR

The Multi Slave contains a very powerful monitor program, providing numerous features not found in most simple monitors. Some of the highlights include:

- o Full function decimal/hexadecimal calculator (expression evaluator)
- o Z80 disassembler
- o Offset variable for LIST and DUMP functions
- o Comprehensive memory diagnostics
- o Printer echo (all console input & output may be echoed to the printer) with user selectable baud rate
- o Scope loops - memory and I/O read/write loops

The basic monitor command structure is as follows:

COMMAND_LETTER [V1 [,V2 [,V3]]] <CR>

where V1, V2, and V3 are variables (command parameters), and unless indicated otherwise, are hexadecimal values. Parameters may or may not be required depending upon the command. Note that ALL commands are terminated by a carriage return <CR>, and fields within brackets ([]) are optional.

MONITOR COMMANDS

B	BOOT this slave system by issuing an operating system download request to the master. For specific details about the download request program, see "Master/Slave Communications" in section II of this manual.
C X1,X2,X3	COMPARE the contents of memory, starting at address X1 to address X2, for X3 bytes. If a mismatch occurs, the contents of both addresses will be displayed.
D X1 [X2]]	DUMP the contents of memory beginning at address 0000H (if the DUMP command has not been previously invoked), or continue at the last address plus one, or at address X1 for 256 bytes (or thru address X2). SEE OFFSET ($) COMMAND.

F X1,X2,X3	FILL memory from address X1 thru X2 with data value X3.
G X1	GO to address X1 (via a CALL) and execute the instructions at that address.
H	See the next subsection for details of this command.
I X1	INPUT data from I/O port address X1 and display it on the console. To display the contents of the next sequential port, enter a carriage return; to display the contents of the prior port, enter a minus (-) sign.
K [N1]	Display the current bank number, or switch to bank N1.
L [X1 [X2]]	LIST, using Z80 mnemonics, the instructions beginning at address 0000H (if the LIST command has not been previously invoked), or continue at the last address plus one, or at address X1 for 18 lines of instructions, or thru address X2. SEE OFFSET ($) COMMAND.
M X1,X2,X3	MOVE the contents of memory beginning at address X1 thru address X2 to address X3.
O X1,X2	OUTPUT data byte X2 to I/O port address X1.
P [D1]	Toggle the PRINTER online or offline, or set the printer baud rate to value D1 (decimal).
S X1	SET the contents of address X1. The current contents of the memory location will be displayed. Enter <CR> to advance to the next address, (-) to go back to the prior address, two hexadecimal characters (0-9, A-F) to change the hex value, or (,A) to change the contents to ASCII value A.
T [X1 [,X2]]	TEST memory beginning at 0000H thru the highest possible address (the starting address of the monitor -1), or starting at address X1 (thru address X2). A plus sign (+) will be displayed with each successful pass. Any address which fails will be displayed at the console, along with the expected and failing data pattern. Upon completion of the test (one complete pass at all specified addresses), the test will be terminated and a message will be displayed at the console.

Z X1,X2 [,X3] Perform a Scope Loop test specified by function X1, as follows:

0 = Memory Read Loop
1 = Memory Write Loop
2 = I/O Read Loop
3 = I/O Write Loop

Field X2 is the memory address or I/O port address, and X3 is the data value to be written to the specified memory or I/O address. Note that field X3 is required only if the test function is a write operation.

$ [X1] Display the current offset value, or change it to value X1. This offset will be added to the address specified in the DUMP and LIST commands.

MONITOR ASPECTS AND CONSIDERATIONS

The Multi Slave monitor takes two important factors into consideration; self preservation, and the possibility of an active master processor on the S-100 bus during monitor execution. Self preservation implies the protection of the memory region in which the monitor resides.

Interrupts are enabled while the monitor is executing; the interrupt service routine performs one important task, that of setting the SLAVE ALIVE bit at each 16.666 ms interrupt interval. This ensures that the master processor will not attempt to reset the slave; the slave always appears to be "alive."

Any command which modifies the contents of memory performs a test of the target address to ensure that it is not 1) the Z80 Mode 1 interrupt vector address or any portion of it (i.e. locations 0038H, 0039H or 003AH), and 2) an address within the monitor. In case 1, the SET, TEST, FILL and MOVE commands will simply skip over these locations. In case 2, an error message will be issued to the console.

There are two I/O address groups which must be accessed with care. The first group is 20H thru 2FH; an I/O read in this range asserts the SLAVE MESSAGE bit, indicating to the master processor that the slave is requesting service. Typically, this will initiate the operating system download sequence.

The second I/O address group is 70H thru 7FH, the "gateway" to the S-100 bus. An I/O read or write in this range causes the CPU to enter a WAIT condition, terminated only after the master has read from or written to the corresponding communications port. In most cases, this will be fatal to the monitor.

To prevent an inadvertent read or write to these ports, the monitor will prompt with:

CONFIRM (Y/N):

when it encounters a read or write request to any one of the aforementioned addresses. A single keystroke reply is required, either 'Y' to perform the requested function, or any other key to terminate it.

OTHER FEATURES AND FACILITIES

The Multi Slave monitor provides a means of obtaining hardcopy output of all console I/O. When hardcopy output is desired, execution of the 'P' command will display the 'PRINTER ON' message at the console, with all subsequent console I/O being echoed to the printer. The next invokation of the 'P' command terminates the printer output, and displays the 'PRINTER OFF' message to the console.

The default printer baud rate is 9600 baud; this may be changed by using the second form of the 'P' command:

Pn <CR>

where 'n' is the desired baud rate. The baud rates currently supported are 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 7200, 9600, 19,200 and 38,400. There is one special case here, the 134.5 baud rate; this value should be entered as 134 (without the decimal fraction).

All monitor commands may be terminated with the ESC(ape) key. Console (and printer) output may be temporarily suspended by entering control-S (^S) and resumed by entering control-Q (^Q).

A monitor command line may be 'deleted' by entering control-X (^X). This causes the cursor to return to the beginning of the current line, immediately to the right of the asterisk (*) prompt character. If hardcopy output is enabled, a carriage return/line feed/space sequence is issued to the printer so that the next command line is not typed over the current line.

All other control characters are invalid. The monitor will issue the bell character to the console in place of the control character.

All lower case characters entered on the console are converted to upper case.

THE 'H' COMMAND - EXPRESSION EVALUATOR

Pocket calculators with such capabilities as hexadecimal display and Boolean functions are a convenient tool, but when one isn't handy, such calculations by hand are tedious at best. For this reason, an expression evaluator has been included in the Multi Slave monitor program.

The expression evaluator has a total of 17 operators, as follows:

&	Dyadic AND
*	Dyadic MULTIPLY
+	Dyadic ADD or monadic PLUS
-	Dyadic SUBTRACT or monadic MINUS
/	Dyadic DIVIDE (two's complement)
//	Dyadic REMAINDER
<	Dyadic LESS THAN
>	Dyadic GREATER THAN
<=	Dyadic LESS THAN OR EQUAL
==	Dyadic EQUAL
>=	Dyadic GREATER THAN OR EQUAL
<<	Dyadic or monadic ROTATE LEFT
>>	Dyadic or monadic ROTATE RIGHT
|	Dyadic INCLUSIVE OR
||	Dyadic EXCLUSIVE OR
~=	Dyadic NOT EQUAL
~	Monadic NOT (one's complement)

Constants may occur in two forms, numbers or strings. Numbers may be in decimal or hexadecimal notation. A decimal number is simply a string of digits; a hexadecimal number is prefixed by either a dollar sign ($) or enclosed in single quotes prefixed by an X (e.g. $FFFF or X'FFFF'). Expressions are evaluated to 16 bits of precision (modulo 65536, or $FFFF) using two's complement arithmetic, with no check for overflow. Strings are any sequence of characters enclosed in single quotes. Two consective single quotes are required to represent one single quote within a string. Null strings are ignored.

Expressions have the form:

[[OPERAND_1 [OPERATOR]] OPERAND_2

where OPERATOR is one of the aforementioned dyadic or monadic operators, and OPERAND_1 and OPERAND_2 are constants or expressions. Dyadic operators require both operands, monadic operators require only OPERAND_2.

All operators have equal precedence, and expressions are evaluated from left to right. Parentheses around an expression may be used to alter precedence; the innermost expression in parenthesis is evaluated first.

The dyadic operators are used to form expressions which evaluate to either one or zero, indicating a true or false condition respectively. A true expression yields a result of one; a false comparison produces a zero result.

The left and right rotation operators (<< and >>) perform a 16 bit rotate (not shift) operation; rotation implies wraparound of bits. Shifting can be accomplished with the multiply and divide operators. When used as monadic operators, operands are rotated one bit. When used as dyadic operators, the first operand specifies the number of bits to rotate.

If a string constant appears in an expression with operators, the value of the string is the ASCII code of the first character in the string. An exception to this rule occurs when two strings appear with a relational operator; then a character by character comparison of the two strings is performed, using the ASCII value to determine relative order. If the two strings are of unequal length, the shorter is padded on the right with spaces.

EXAMPLES:

Expression	Evaluates to:
3	3
(4)	4
-(+3)	$FFFD
~3	$FFFC
9//8	1
1+2*3	9
1+(2*3)	7
5==5	1 (indicates a true condition)
$FFFF<0	0 ($FFFF=65535)
3+(2~=6)	4 (3+1, where 1=true)
<<2	4
>>3	$8001
3>>$F0	$001E
'A'+1	$0042

S-100 BUS PINOUT

S-100 Bus Pin	Mnemonic	Active State	Source	Notes
01	+8 Volts	----	Bus	
02	+16 Volts	----	Bus	
03	XRDY	High	Slave	Unused
04	VI0*	Low (OC)	Slave	Vectored Int 0
05	VI1*	Low (OC)	Slave	Vectored Int 1
06	VI2*	Low (OC)	Slave	Vectored Int 2
07	VI3*	Low (OC)	Slave	Vectored Int 3
08	VI4*	Low (OC)	Slave	Vectored Int 4
09	VI5*	Low (OC)	Slave	Vectored Int 5
10	VI6*	Low (OC)	Slave	Vectored Int 6
11	VI7*	Low (OC)	Slave	Vectored Int 7
12	NMI*	Low (OC)	Slave	Unused
13	PWRFAIL*	Low	Bus	Unused
14	TMA3*	Low (OC)	Master	Unused
15	A18	High	Master	Unused
16	A16	High	Master	Unused
17	A17	High	Master	Unused
18	SDSB*	Low (OC)	Master	Unused
19	CDSB*	Low (OC)	Master	Unused
20	GND	----	Bus	
21	----	----	----	Unused
22	ADSB*	Low (OC)	Master	Unused
23	DODSB*	Low (OC)	Master	Unused
24	phi	High	Master	Unused
25	pSTVAL*	Low	Master	Unused
26	pHLDA	High	Master	Unused
27	----	----	----	Unused
28	----	----	----	Unused
29	A05	High	Master	Address bit 5
30	A04	High	Master	Address bit 4
31	A03	High	Master	Address bit 3
32	A15	High	Master	Address bit 15
33	A12	High	Master	Address bit 12
34	A09	High	Master	Unused
35	DO1	High	Master	Data Out bit 1
	DATA1	High	M/S	Unused 1
36	DO0	High	Master	Data Out bit 0 (LSB)
	DATA0	High	M/S	Unused
37	A10	High	Master	Unused
38	DO4	High	Master	Data Out bit 4
	DATA4	High	M/S	Unused
39	DO5	High	Master	Data Out bit 5
	DATA5	High	M/S	Unused
40	DO6	High	Master	Data Out bit 6
	DATA6	High	M/S	Unused
41	DI2	High	Slave	Data In Bit 2
	DATA10	High	M/S	Unused
42	DI3	High	Slave	Data In bit 3
	DATA11	High	M/S	Unused

S-100 Bus Pin	Mnemonic	Active State	Source	Notes
43	DI7	High	Slave	Data In bit 7
	DATA15	High	M/S	Unused
44	sM1	High	Master	Unused
45	sOUT	High	Master	Output Bus Cycle
46	sINP	High	Master	Input Bus Cycle
47	sMEMR	High	Master	Unused
48	----	----	----	Unused (sHLTA)
49	CLOCK	High	Master	Unused
50	GND	----	Bus	Ground
51	+8 Volts	----	Bus	
52	-16 Volts	----	Bus	
53	GND	----	Bus	Ground
54	SLV CLR*	Low	Master	Unused
55	TMA0*	Low (OC)	Master	Unused
56	TMA1*	Low (OC)	Master	Unused
57	TMA2*	Low (OC)	Master	Unused
58	sXTRQ*	Low	Master	Unused
59	A19	High	Master	Unused
60	SIXTN*	Low (OC)	Master	Unused
61	A20	High	Master	Unused
62	A21	High	Master	Unused
63	A22	High	Master	Unused
64	A23	High	Master	Unused
65	----	----	----	Unused
66	----	----	----	Unused
67	----	----	----	Unused (PHANTOM*)
68	----	----	----	Unused (MWRT)
69	----	----	----	Unused
70	GND	----	Bus	Ground
71	----	----	----	Unused
72	RDY	High (OC)	Slave	Unused
73	INT*	Low (OC)	Slave	Unused
74	HOLD*	Low (OC)	Master	Unused
75	RESET*	Low (OC)	Bus	System Reset
76	pSYNC	High	Master	Bus Transfer control
77	pWR*	Low	Master	Data Bus Valid
78	pDBIN	High	Master	Data In Strobe
79	A0	High	Master	Address bit 0 (LSB)
80	A1	High	Master	Address bit 1
81	A2	High	Master	Address bit 2
82	A6	High	Master	Address bit 6
83	A7	High	Master	Address bit 7
84	A8	High	Master	Unused
85	A13	High	Master	Unused
86	A14	High	Master	Unused
87	A11	High	Master	Unused
88	DO2	High	Master	Data Out bit 2
	DATA2	High	M/S	Unused
89	DO3	High	Master	Data Out bit 3
	DATA3	High	M/S	Unused
90	DO7	High	Master	Data Out bit 7
	DATA7	High	M/S	Unused

S-100 Bus Pin	Mnemonic	Active State	Source	Notes
91	DI4	High	Slave	Data In bit 4
	DATA12	High	M/S	Unused
92	DI5	High	Slave	Data In bit 5
	DATA13	High	M/S	Unused
93	DI6	High	Slave	Data In bit 6
	DATA14	High	M/S	Unused
94	DI1	High	Slave	Data In bit 1
	DATA9	High	M/S	Unused
95	DI0	High	Slave	Data In bit 0
	DATA8	High	M/S	Unused
96	sINTA	High	Master	Unused
97	sWO*	Low	Master	Unused
98	ERROR*	Low (OC)	Slave	Unused
99	POC*	Low	Bus	Unused
100	GND	---	Bus	Ground

(OC) = open collector
Unused = pin not implemented on Multi Slave

PROM SELECTION JUMPER BLOCKS

```
1 -- o  o -- 2
3 -- o  o -- 4
5 -- o  o -- 6
7 -- o  o -- 8

TYPICAL JUMPER
     BLOCK
```

PROM TYPE	JUMPER
2716 (2K)	3-5, 4-6
2732 (4K)	5-7, 4-6
2764 (8K)	1-3, 5-7
27128 (16K)	1-3, 6-8, 5-7
27256 (32K)	1-2, 6-8, 5-7

S-100 BUS VECTORED INTERRUPT JUMPERS

MULTI SLAVE	JUMPER BLOCK	S-100 BUS
Multi Slave CPU 0	1 -- o o -- 2	INTO* (pin 4)
" " " "	3 -- o o -- 4	INT1* (pin 5)
Multi Slave CPU 1	5 -- o o -- 6	INT2* (pin 6)
" " " "	7 -- o o -- 8	INT3* (pin 7)
Multi Slave CPU 2	9 -- o o -- 10	INT4* (pin 8)
" " " "	11 -- o o -- 12	INT5* (pin 9)
No Connection	13 -- o o -- 14	INT6* (pin 10)
" "	15 -- o o -- 16	INT7* (pin 11)

BASE ADDRESS SELECTION

```
(7)(6)(5)(4)

 1  3  5  7
 |  |  |  |
 o  o  o  o
MSB              LSB
 o  o  o  o
 |  |  |  |
 2  4  6  8
```

Address selection uses LOW TRUE LOGIC (no jumper=1, jumpered=0) and represents the four most significant bits of the base address.

Example:

BASE ADDRESS	JUMPERS
70H	1-2
80H	3-4, 5-6, 7-8
90H	3-4, 5-6
AOH	3-4, 7-8
BOH	3-4
COH	5-6, 7-8
DOH	5-6

SERIAL I/O CONNECTORS

PIN NO.	SIGNAL NAME		DIRECTION
1	DCD	Data Carrier Detect	input
2	DSR	Data Set Ready	input
* 3	***	see note below	input/output
4	RXD	Receive Data	input
5	CTS	Clear to Send	input
6	TXD	Transmit Data	output
7	RTS	Request to Send	output
8	DTR	Data Terminal Ready	output
9	CLK	Tx/Rx Clock	input/output
10	GND	Signal Ground	ground
11	N/C		
12	+16 VDC		supply voltage
13	-16 VDC		supply voltage
14	+5 VDC		supply voltage

* On channel A, this term is RNG (ring detect) and may be connected to the ring detect line on a modem. This line has no connected on channel B.

I/O PORT ASSIGNMENTS

ADDRESS (Hex)	TYPE	FUNCTION
00-0F	R/W	DUART Data/Control (see Signetics 2681 Documentation for details)
10-1F	---	Not Used
20-2F	R	Asserts SLAVE MESSAGE bit
30-3F	R	De-asserts MASTER MESSAGE bit
40-4F	W	BANK/PROM SELECT (details follow)
50-5F	R	STATUS PORT (MASTER and SLAVE bits)
60-6F	R	Asserts SLAVE ALIVE* bit
70-7F	R/W	S-100 DATA PORT

MULTI SLAVE SCHEMATICS

Item	Qty.	Part No.	Description	Reference
1	5	74LS367	BUS DRIVER	U1,U47,U52,U53,U64
2	3	74LS590	8 BIT COUNTER	U2,U6,U10
3	3	SN2681	DUART	U5,U9,U18
4	6	AM2965	MEMORY DRIVER	U13,U15,U17,U28,U31 U34
5	3	Z80H	CPU	U14,U16,U27
6	6	74LS138	3 TO 8 DECODER	U29,U32,U41,U43,U44 U51
7	3	74LS85	4 BIT COMPARATOR	U30,U33,U42
8	3	74LS74	DUAL D FLIP FLOP	U39,U40,U58
9	3	74LS279	QUAD LATCH	U45,U56,U59
10	1	7406	HEX INVERTER	U46
11	2	74LS00	QUAD NAND GATE	U49,U50
12	3	74LS273	OCTAL D FLIP FLOP	U54,U55,U66
13	6	74LS373	OCTAL LATCH	U61,U62,U63,U68,U69 U70
14	1	74LS393	4 BIT COUNTER	U71
15	1	74LS688	8 BIT ID COMPARATOR	U67
16	1	DS0026	CLOCK DRIVER	U57
17	3	PAL16R4A	PROG ARRAY LOGIC	U21,U25,U37
18	3	PAL16L8A	PROG ARRAY LOGIC	U22,U26,U38
19	1	PAL16L8	PROG ARRAY LOGIC	U60
20	3	2764JL25	EPROM	U19,U23,U35
21	3		4.7K OHM PACK	RP1,RP2,RP3
22	12		47 OHM 10% 1/4W	R1-R12
23	1		1K 10% 1/4W	R16
24	5		10 OHM 10% 1/4W	R17-R19,R21,R22
25	5		47 PF CAPACITOR	C6-C10
26	1		5 PF CAPACITOR	C11
27	1		10 PF CAPACITOR	C12
28	1		100 PF CAPACITOR	C13
29	1		6.8 UF 25WVDC	C2
30	3		10 UF 25WVDC	C3-C5
31	1		.1 UF CERAMIC CAP	C1
32	6	TM4164EL9	64K x 8 SIP DRAM	U3,U4,U7,U8,U11,U12
33	1		8 PIN DIP SOCKET	U57
34	8		14 PIN DIP SOCKET	U39,U40,U46,U49,U50 U58,U71,DL1
35	19		16 PIN DIP SOCKET	U1,U2,U6,U10,U29,U30 U32,U33,U41,U42,U44, U45,U47,U51,U52,U53, U56,U59,U64
36	23		20 PIN DIP SOCKET	U13,U15,U17,U21,U22, U25,U26,U28,U31,U34, U37,U38,U54,U55,U60, U61,U62,U63,U66,U67, U68,U69,U70
37	3		28 PIN DIP SOCKET	U19,U23,U35
38	6		40 PIN DIP SOCKET	U5,U9,U14,U16,U18, U27
39	1		3.6864 MHz XTAL	Y1
40	1	NCT050C	8.00 MHz OSCILLATOR	Y2
41	1	DL6135	30Ns DELAY LINE	DL1

Item	Qty.	Part No.	Description	Reference
42	6		14 PIN HEADER	J1-J6
43	4		8 PIN HEADER	E1-E8,E19-E26 E9,E16,E27-E34
44	1		16 PIN HEADER	J7
45	1	78H05	5V/5A REGULATOR	VR1
46	2		PCB EJECTORS	---
47	1		HEAT SINK [REF VR1]	---
48	2		6-32 x 3 SCREWS	---
49	2		6-32 NUTS	---
50	2		#6 LOCK WASHER	---

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

DESCRIPTION

The Signetics SCN2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

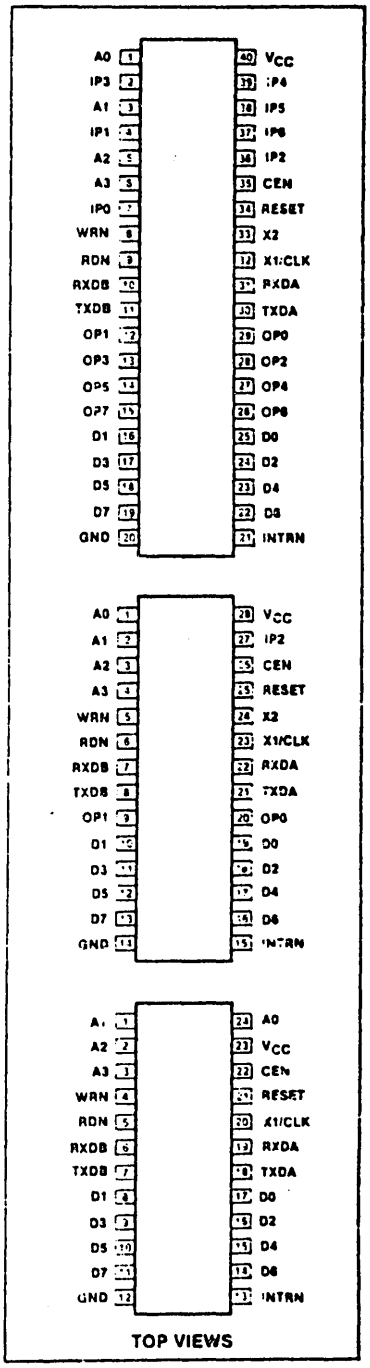
Also provided on the SCN2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2681 is available in three package versions to satisfy various system requirements: 40-pin and 28-pin, both 0.6" wide DIPs, and a compact 24-pin, 0.4" wide, DIP.

FEATURES

- Dual full-duplex asynchronous receiver/transmitter
- Quadruple buffered receiver data registers
- Programmable data format
  - —5 to 8 data bits plus parity
  - —Odd, even, no parity or force parity
  - —1, 1.5 or 2 stop bits programmable in 1/16 bit increments
- Programmable baud rate for each receiver and transmiter selectable from:
  - —18 fixed rates: 50 to 38.4K baud
  - —One user defined rate derived from programmable timer/counter
  - —External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - —Normal (full duplex)
  - —Automatic echo
  - —Local loopback
  - —Remote loopback
- Multi-function programmable 16-bit counter/timer
- Multi-function 7-bit input port
  - —Can serve as clock or control inputs
  - —Change of state detection on four inputs
- Multi-function 8-bit output port
  - —Individual bit set/reset capability
  - —Outputs can be programmed to be status/interrupt signals
- Versatile interrupt system
  - —Single interrupt output with eight maskable interrupting conditions
  - —Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs
- Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec
- Automatic wake-up mode for multidrop applications
- Start-end break interrupt/status
- Detects break which originates in the middle of a character
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply

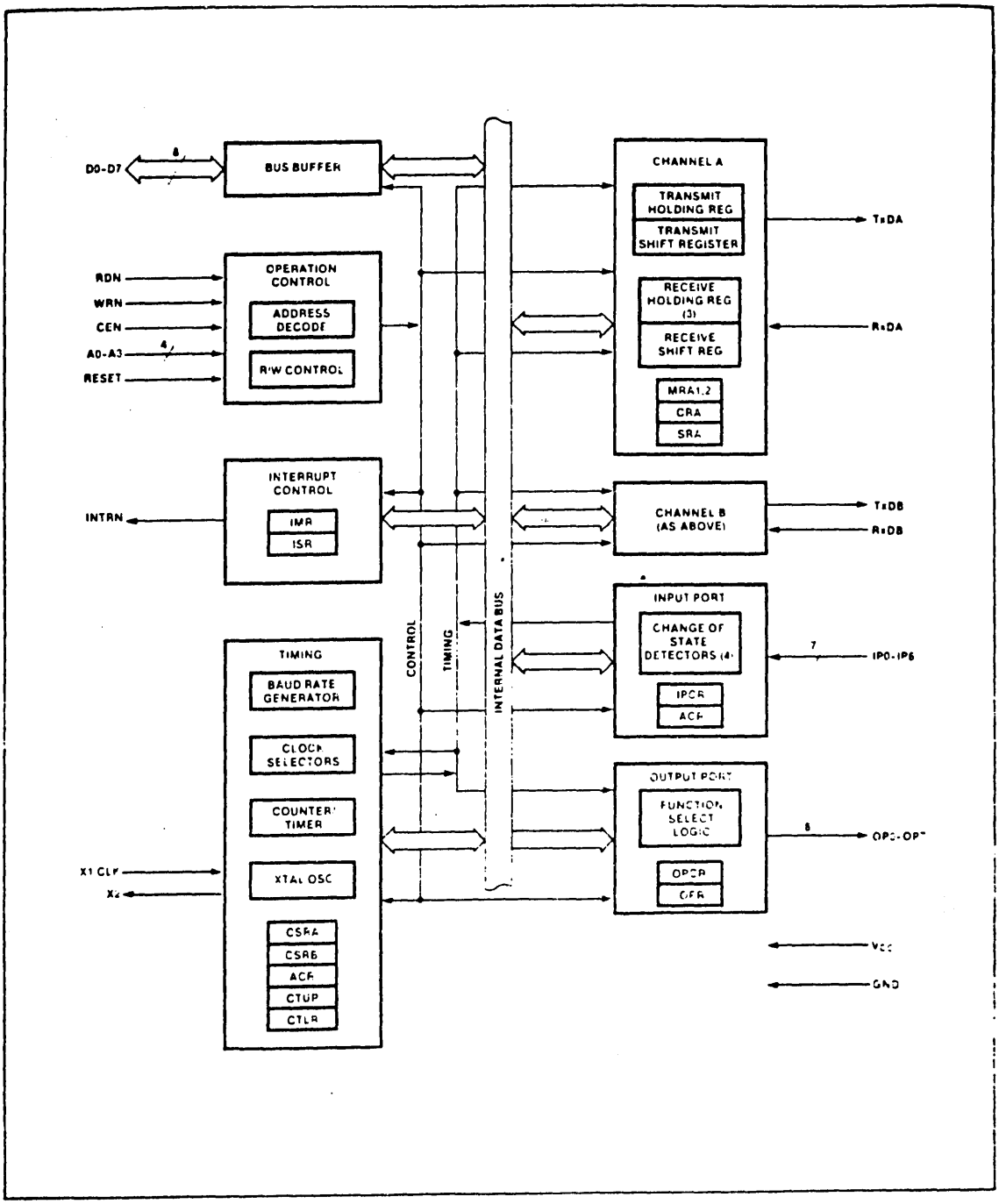
PIN CONFIGURATION



TOP VIEWS

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

BLOCK DIAGRAM



DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

PIN DESIGNATION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0-D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the WRN, RDN and A0-A3 inputs. When high, places the D0-D7 lines in the 3-state condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0-A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0-OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
X2	X	X		O	**Crystal 2:** Connection for other side of the crystal. Should be connected to ground if a crystal is not used. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

PIN DESIGNATION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** + 5V supply input
GND	X	X	X	I	**Ground**

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3-OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the 2681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. D_7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50μs will set the corresponding bit in the input port with change register. The bits are cleared when this register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR)

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. If it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the

receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the ac-

Table 1 2681 REGISTER ADDRESSING

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

Table 2 REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA CSRB	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA CRB	not used— must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7-5) from the top of the FIFO together with bits 4-0. These bits are cleared by a "reset error status" command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Table 2 REGISTER BIT FORMATS (Continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

cumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.

3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate CLOCK = 3.6864MHz	
CSRA[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP6—16X	IP6—16X
1 1 1 1	IP6—1X	IP6—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received. further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following.

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

Table 3 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE
Duty cycle of 16X clock is 50% ± 1%

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7]= 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

Table 4 ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		4.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.