Signetics

SCC2698

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

Microprocessor Products

Product Specification

DESCRIPTION

The Signetics SCC2698 Octal Universal Asynchronous Receiver/Transmitter (Octal-UART) is a single-chip MOS-LSI communications device that provides an eight-channel, full-duplex asynchronous receiver/transmitter in a single package. It is fabricated with Signetics' CMOS technology, which combines the benefits of High density and Low power consumption.

The operating speed of each receiver and transmitter can be selected independently as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× 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 Octal-UART particularly functional for dual-speed channel applications such as clustered terminal systems.

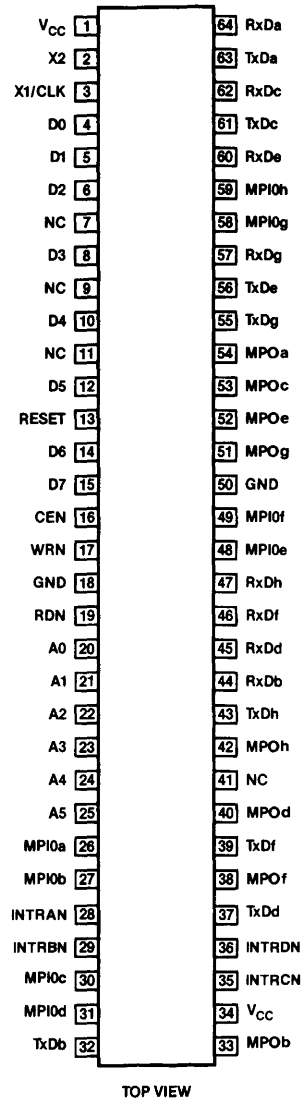
The receiver is quadruply buffered to minimize the potential of receiver overrun, or to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking capability is provided to disable a remote UART transmitter when the receiver buffer is full.

The Octal-UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes. The Octal-UART is fully TTL compatible and operates from a single +5V power supply.

FEATURES

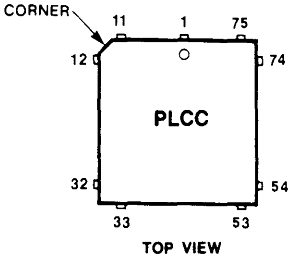
- **Eight full-duplex asynchronous receiver/transmitters**
- **Quadruple buffered receiver data register**
- **Programmable data format:**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Baud rate for the receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **Four user-defined rates derived from the programmable counter/timer associated with each of the four blocks**
  - **External 1× or 16× 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**
- **Four multi-function programmable 16-bit counter/timers**
- **Four interrupt outputs with eight maskable interrupting conditions for each output**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply with Low power mode**

PIN CONFIGURATIONS



TOP VIEW

CD09382S



TOP VIEW

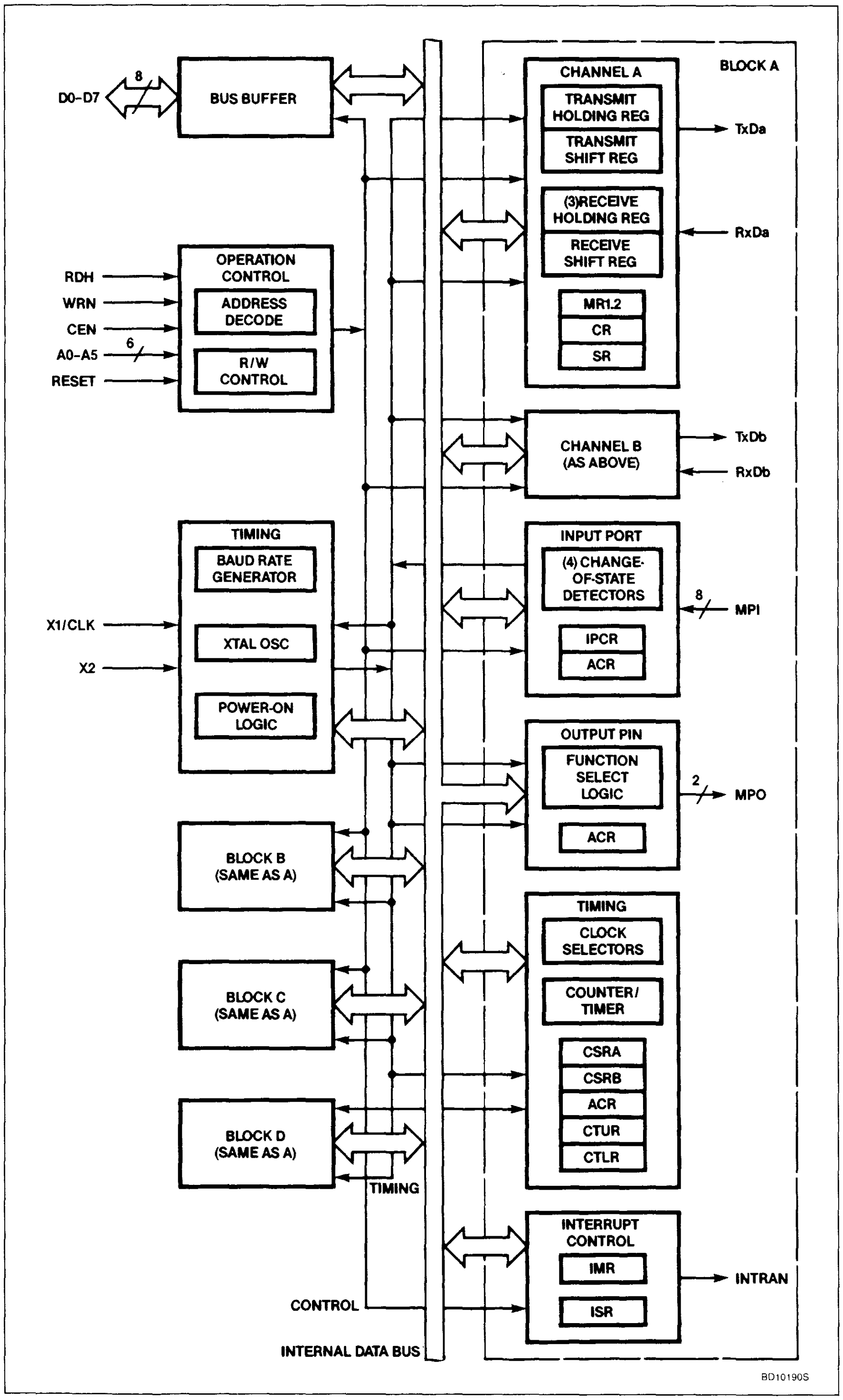
CD0045PS

PLCC Pin Functions (Next Page)

ORDERING INFORMATION

DESCRIPTION	V_{CC} = +5V ± 5%, T_A = 0 to +70°C
Plastic DIP	SCC2698AC1N64
Plastic LCC	SCC2698AC1A84

BLOCK DIAGRAM



PIN CONFIGURATIONS (Continued)

PLCC

Pin	Function	Pin	Function
1	TxDa	43	MPOb
2	MPI3g	44	MPI2d
3	RxDa	45	V_{CC}
4	MPI3h	46	INTRCN
5	V_{CC}	47	INTRDN
6	X2	48	MPI3c
7	X1/CLK	49	TxDd
8	D0	50	MPI3d
9	D1	51	MPOd
10	D2	52	TxDf
11	D3	53	MPOf
12	D4	54	MPOh
13	D5	55	TxDh
14	MPI1a	56	RxDb
15	RESET	57	RxDd
16	D6	58	RxDf
17	D7	59	RxDh
18	CEN	60	MPI1e
19	WRN	61	MPI0e
20	GND	62	MPI1f
21	MPI1b	63	MPI0f
22	RDN	64	MPI2e
23	A0	65	GND
24	MPI2a	66	MPI2f
25	A1	67	MPOg
26	MPI2b	68	MPI3e
27	A2	69	MPOe
28	MPI3a	70	MPI3f
29	A3	71	MPOc
30	MPI3b	72	MPOa
31	A4	73	TxDg
32	A5	74	TxDe
33	MPI0a	75	RxDg
34	MPI0b	76	MPI0g
35	INTRAN	77	MPI0h
36	INTRBN	78	MPI1g
37	MPI0c	79	RxDe
38	MPI1c	80	MPI1h
39	MPI0d	81	TxDc
40	MPI1d	82	MPI2g
41	TxDb	83	RxDc
42	MPI2c	84	MPI2h

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 - D7	4 - 6, 8, 10, 12, 14, 15	8 - 13, 16, 17	I/O	**Data Bus:** Active-High 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the Octal-UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is Low. When the CEN input is High, the data bus is in the 3-State condition.
CEN	16	18	I	**Chip Enable:** Active-Low input. When Low, data transfers between the CPU and the Octal-UART are enabled on D0 - D7 as controlled by the WRN, RDN, and A0 - A5 inputs. When CEN is High, the Octal-UART is effectively isolated from the data bus and D0 - D7 are placed in the 3-State condition.
WRN	17	19	I	**Write Strobe:** Active-Low input. A Low on this pin while CEN is Low causes the contents of the data bus to be transferred to the register selected by A0 - A5. The transfer occurs on the trailing (rising) edge of the signal.
RDN	19	22	I	**Read Strobe:** Active-Low input. A Low on this pin while CEN is Low causes the contents of the register selected by A0 - A5 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0 - A5	20 - 25	23, 25, 27, 29, 31, 32	I	**Address Inputs:** Active-High address inputs to select the Octal-UART registers for read/write operations.
RESET	13	15	I	**Reset:** Master reset. A High on this pin clears the status register (SR), clears the Interrupt Mask Register (IMR), clears the Interrupt Status Register (ISR), clears the output port configuration register (OPCR), places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (High) state, and stops the counter/timer.
INTRAN - INTRDN	28, 29, 35, 36	35, 36, 46, 47	O	**Interrupt Request:** This Active-Low open drain output is asserted on occurrence of one or more of eight maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s).
X1/CLK	3	7	I	**Crystal 1:** Crystal or external clock input. When using the crystal oscillator, this pin serves as the connection for one side of the crystal. If a crystal is not used, an external clock is supplied at this input. An external clock (or crystal) is required even if the internal Baud Rate Generator is not utilized. This clock is used to drive the internal Baud Rate Generator, as an optional input to the timer/counter, and to provide other clocking signals required by the chip.
X2	2	6	I	**Crystal 2:** Connection for other side of crystal. If an external source is used instead of a crystal, this should be open or connected as shown in Figure 6.
RxDa - RxDh	64, 44, 62, 45, 60, 46, 57, 47	3, 56, 83, 57, 79, 58, 75, 59		**Receiver Serial Data Input:** The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
TxDa - TxDh	63, 32, 61, 37, 56, 39, 55, 43	1, 41, 81, 49, 74, 52, 73, 55	O	**Transmitter Serial Data Output:** Transmitter serial data output. The least significant bit is transmitted first. This output is held in the marking (High) condition when the transmitter is idle or disabled and when the Octal-UART is operating in local loopback mode. If external transmitter clock is specified, the data is shifted on the falling edge of the transmitter clock.
MPOa - MPOh	54, 33, 53, 38, 52, 40, 51, 42	72, 43, 71, 51, 69, 53, 67, 54	O	**Multi-Purpose Output:** One of the following functions can be selected for this output pin by programming the output port configuration register: **RTSN** — Request to send Active-Low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. **C/TO** — Counter/timer output **TxC1**× — 1× clock for the transmitter **TxC16**× — 16× clock for the transmitter **RxC1**× — 1× clock for the receiver **RxC16**× — 16× clock for the receiver **TxRDY** — Transmitter holding register empty signal **RxRDY/FFULL** — Receiver FIFO not empty/full signal
MPI0a - MPI0h	26, 27, 30, 31, 48, 49, 58, 59	33, 34, 37, 39, 61, 63, 76, 77	I	**Multi-Purpose Input 0:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the IPCR or input port register. **CTSN** — Clear-to-Send Active-Low input.
MPI1a - MPI1h	NC	14, 21, 38, 40, 60, 62, 78, 80	I	**Multi-Purpose Input 1:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the IPCR or input port register. **CTCLK** — Counter/timer external clock input. Only channels a, c, e, and g change to C/T inputs; channels b, d, f and h stays GPI.
MPI2a - MPI2h	NC	24, 26, 42, 44, 64, 66, 82, 84	I	**Multi-Purpose Input 2:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the input port register. **TCLK** — Transmitter external clock input. This may be a 1× or 16× clock as programmed by CSR[3:0].
MPI3a - MPI3h	NC	28, 30, 48, 50, 68, 70, 2, 4	I	**Multi-Purpose Input 3:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the input port register. **RCLK** — Receiver external clock input. This may be a 1× or 16× clock as programmed by CSR[7:4].
V_{CC}	1, 34	5, 45	I	**Power Supply:** +5V supply input
GND	18, 50	20, 65	I	**Ground**

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

SCC2698

BLOCK DIAGRAM

As shown on the block diagram, the Octal-UART consists of: data bus buffer, interrupt control, operation control, timing, and eight receiver and transmitter channels. The eight channels are divided into four different blocks each block independent of each other (see Figure 1).

BLOCK A CHANNELS a, b	BLOCK C CHANNELS e, f
BLOCK B CHANNELS c, d	BLOCK D CHANNELS g, h

Figure 1. Channel Architecture

Channel Blocks

There are four blocks (Figure 1), each containing two sets of receiver/transmitters. In the following discussion, the description applies to block A which contains channels a and b. However, the same information applies to all channel blocks.

Data Bus Buffer

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

Interrupt Control

A single interrupt output (INTRN) is provided which is asserted on the occurrence of any of the following internal events:

- Transmit holding register ready for each channel
- Receive holding register ready or FIFO full for each channel
- Change in break received status for each channel
- Counter reached terminal count
- Change in MPI input

Associated with the interrupt system are the Interrupt Mask Register (IMR) and the Interrupt Status Register (ISR). The IMR can be programmed to select only certain conditions, of the above, to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. However, the bits of the ISR are not masked by the IMR.

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. The functions performed by the CPU read and write operations are shown in Table 1.

Mode registers 1 and 2 are accessed via an auxiliary pointer. The pointer is set to MR1 by RESET or by issuing a reset pointer command via the command register. Any read or write of the mode register while the pointer is at MR1 switches the pointer to MR2. The pointer then remains at MR2 so that subsequent accesses are to MR2, unless the pointer is reset to MR1 as described above.

Timing Circuits

The timing block consists of a crystal oscillator, a Baud Rate Generator, a programmable 16-bit counter/timer for each block, and two clock selectors.

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external clock is used instead of a crystal, both X1 and X2 are driven using a configuration similar to the one in Figure 6. Also, an arrangement where X2 is floating can be used, however, the input High voltage must be capable of attaining 4.4V. The clock serves as the basic timing reference for the Baud Rate Generator (BRG), the counter/timer, and other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied even if the internal BRG is not used.

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. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection, by the receiver and transmitter, of any of these baud rates or an external timing signal.

There are four C/Ts in the Octal-UART, one for each block. The C/T operation is programmed by ACR[6:4]. One of eight timing sources can be used as the input to the C/T. The output of the C/T is available to the clock selectors and can also be programmed by OPCR[2:0] for channel a and OPCR[6:4] for channel b, to be output on the MPOa or MPOb pin, respectively.

In the timer mode, the C/T generates a square wave whose period is twice the number of clock periods loaded into the C/T upper and lower registers. The counter ready bit in the ISR is set once each cycle of the square wave. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be affected. In this mode, the C/T runs continuously and does not recognize the stop C/T command (the command only resets the counter ready bit in the ISR). Receipt of a start C/T command causes the C/T to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start counter command. Upon reaching terminal count, the counter ready bit in the ISR is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO 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 the counter ready bit 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 following a stop 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 eight bits of the counter 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 eight bits to the upper eight bits occurs between the times both halves of the counter are read. However, a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR.

Receiver and Transmitter

The Octal-UART has eight full-duplex asynchronous receiver/transmitters. The operating frequency for the receiver and transmitter can be selected independently from the Baud Rate Generator, the counter timer, or from an external input.

Registers associated with the communications channel are the Mode Registers (MR1 and MR2), the Clock Select Register (CSR), the Command Register (CR), the Status Register (SR), the Transmit Holding Register (THR), and the Receive Holding Register (RHR).

Transmitter

The transmitter accepts parallel data from the CPU and converts it 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 trans-

Table 1. Register Addressing

A5	A4	A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	0	0	MR1a, MR2a	MR1a, MR2a
0	0	0	0	0	1	SRa	CSRa
0	0	0	0	1	0	Reserved*	CRa
0	0	0	0	1	1	RHRa	THRa
0	0	0	1	0	0	IPCRA	ACRA
0	0	0	1	0	1	ISRA	IMRA
0	0	0	1	1	0	CTUA	CTURA
0	0	0	1	1	1	CTLA	CTLRA
0	0	1	0	0	0	MR1b, MR2b	MR1b, MR2b
0	0	1	0	0	1	SRb	CSRb
0	0	1	0	1	0	Reserved*	CRb
0	0	1	0	1	1	RHRb	THRb
0	0	1	1	0	0	Reserved*	Reserved*
0	0	1	1	0	1	Input port A	OPCRA
0	0	1	1	1	0	Start C/T A	Reserved*
0	0	1	1	1	1	Stop C/T A	Reserved*
0	1	0	0	0	0	MR1c, MR2c	MR1c, MR2c
0	1	0	0	0	1	SRc	CSRc
0	1	0	0	1	0	Reserved*	CRc
0	1	0	0	1	1	RHRc	THRc
0	1	0	1	0	0	IPCRB	ACRB
0	1	0	1	0	1	ISRB	IMRB
0	1	0	1	1	0	CTUB	CTURB
0	1	0	1	1	1	CTLB	CTLRB
0	1	1	0	0	0	MR1d, MR2d	MR1d, MR2d
0	1	1	0	0	1	SRd	CSRd
0	1	1	0	1	0	Reserved*	CRd
0	1	1	0	1	1	RHRd	THRd
0	1	1	1	0	0	Reserved*	Reserved*
0	1	1	1	0	1	Input port B	OPCRB
0	1	1	1	1	0	Start C/T B	Reserved*
0	1	1	1	1	1	Stop C/T B	Reserved*
1	0	0	0	0	0	MR1e, MR2e	MR1e, MR2e
1	0	0	0	0	1	SRe	CSRe
1	0	0	0	1	0	Reserved*	CRe
1	0	0	0	1	1	RHRe	THRe
1	0	0	1	0	0	IPCRC	ACRC
1	0	0	1	0	1	ISRC	IMRC
1	0	0	1	1	0	CTUC	CTURC
1	0	0	1	1	1	CTLC	CTLRC
1	0	1	0	0	0	MR1f, MR2f	MR1f, MR2f
1	0	1	0	0	1	SRf	CSRf
1	0	1	0	1	0	Reserved*	CRf
1	0	1	0	1	1	RHRf	THRf
1	0	1	1	0	0	Reserved*	Reserved*
1	0	1	1	0	1	Input port C	OPCRC
1	0	1	1	1	0	Start C/T C	Reserved*
1	0	1	1	1	1	Stop C/T C	Reserved*
1	1	0	0	0	0	MR1g, MR2g	MR1g, MR2g
1	1	0	0	0	1	SRg	CSRg
1	1	0	0	1	0	Reserved*	CRg
1	1	0	0	1	1	RHRg	THRg
1	1	0	1	0	0	IPCRD	ACRD
1	1	0	1	0	1	ISRD	IMRD
1	1	0	1	1	0	CTUD	CTURD
1	1	0	1	1	1	CTLD	CTLRD
1	1	1	0	0	0	MR1h, MR2h	MR1h, MR2h
1	1	1	0	0	1	SRh	CSRh
1	1	1	0	1	0	Reserved*	CRh
1	1	1	0	1	1	RHRh	THRh
1	1	1	1	0	0	Reserved*	Reserved*
1	1	1	1	0	1	Input port D	OPCRD
1	1	1	1	1	0	Start C/T D	Reserved*
1	1	1	1	1	1	Stop C/T D	Reserved*

NOTES:

*Reserved registers should never be read during normal operation since they are reserved for internal diagnostics.

ACR	= Auxiliary control register
CR	= Command register
CSR	= Clock select register
CTL	= Counter/timer lower
CTLR	= Counter/timer lower register
CTU	= Counter/timer upper
CTUR	= Counter/timer upper register
MR	= Mode register
SR	= Status register
THR	= Tx holding register
RHR	= Rx holding register
IPCR	= Input port change register
ISR	= Interrupt status register
IMR	= Interrupt mask register
OPCR	= Output port configuration register

mission 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 SR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. In the 16× clock mode, this also resynchronizes the internal 1× transmitter clock so that transmission of the new character begins with minimum delay.

The transmitter can be forced to send a break (continuous Low condition) by issuing a start break command via the CR. The break is terminated by a stop break command. If the transmitter is disabled, it continues operating until the character currently being transmitted and the character in the THR, if any, are completely sent out. Characters cannot be loaded into the THR while the transmitter is disabled.

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. 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 again each 16× clock for 7½ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× 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 samples the input. This continues 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 RHR and the RxRDY bit in the SR is set to a one. 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 (i.e. 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 and overrun error (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), only one character consisting of all zeros will be loaded into the FIFO and the received break bit in the SR is set to 1. The RxD input must return to a High condition for two successive clock edges of the 1× clock (internal or external) before a search for the next start bit begins.

TIME OUT MODE

Under certain conditions the user may want to set the receiver to interrupt the CPU when the receiver FIFO becomes full. This can be accomplished by programming MR1[6] = 1. If a message that is only one or two characters long is received, the FIFO is not full so that ISR[1] does not set and the CPU is not interrupted. The CPU will not know that there is data in the receive FIFO. The time-out mode provides the user with a time-out interrupt via the C/T. If a character is received and the FIFO does not become full, a pre-selected period of delay can be timed out by the C/T and the CPU interrupted.

This mode is enabled by writing the appropriate command to the command register. Writing an "AX" to CRA or CRB will invoke the time-out mode for that channel. Writing a "CX" to CRA or CRB will reset the time-out mode. CTU and CTL must be loaded with a value greater than the normal receive character period. Each time a received character is transferred from the shift register to the RHR, the C/T is reloaded with the value in CTU and CTL and then restarted. If the C/T is allowed to end the count, the counter ready bit (ISR[3]) will be set. If IMR[3] is set, an interrupt will occur.

RECEIVER FIFO

The RHR consists of a First-In-First-Out (FIFO) with a capacity of three characters. Data is loaded from the receive shift register into the top-most empty position of the FIFO. The RxRDY bit, in the Status Register (SR), is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three 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 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 appended to each data character in the FIFO. 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 SR should be read prior to reading the corresponding data character.

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 exists, 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 on receipt of the start bit of the new (overrunning) character.

WAKE-UP MODE

In addition to the normal transmitter and receiver operation described above, the Octal-UART incorporates a special mode which provides automatic wake-up of the receiver through address frame recognition for multiprocessor communications. This mode is selected by programming bits MR1[4:3] to '11'.

In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, whose receivers 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 MR1[2]; MR1[2] = 0 transmits a zero in the A/D bit position which identifies the corresponding data bits as data, MR1[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.

While 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, but discards the received character if the received A/D bit is a zero. 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 (SR[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

MULTI-PURPOSE INPUT PIN

The inputs to this unlatched 8-bit port for each block can be read by the CPU, by performing a read operation as shown in Table 1. A High input results in a logic one while a Low input results in a logic zero. When the input port pins are read on the 84-pin PLCC, they will appear on the data bus in alternating pairs. (i.e., DB0 = MPI0a, DB1 = MPI1a, DB2 = MPI0b, DB3 = MPI1b, DB4 = MPI2a, DB5 = MPI3a, DB6 = MPI2b, DB7 = MPI3b. Although this example is shown for input port 'A', all input ports will have a similar order).

The MPI pins can be programmed as an input to one of several Octal-UART circuits. The function of the pin is selected by programming the appropriate control register. Change-of-state detectors are provided for MPI0 and MPI1 for each channel in each block. A High-to-Low or Low-to-High transition of the inputs lasting longer than 25 to 50μs sets the MPI change-of-state bit in the Interrupt Status Register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU, by setting the corresponding bit in the Interrupt Mask Register.

The input port pulse detection circuitry uses a 38.4kHz sampling clock, derived from one of the Baud Rate Generator taps. This produces a sampling period of slightly more than 25μs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples be observed at the new logic level. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs coincident with the first sample pulse. (The 50μs time refers to the condition where the change-of-state is just missed and the first change-of-state is not detected until after an additional 25μs.)

MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1× or 16× transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see OPCR [2:0] and OPCR [6:4] — MPO Output Select).

REGISTERS

The operation of the Octal-UART is programmed by writing control words into the appropriate registers. Operational feedback is provided via registers which can be read by the CPU. Addressing the registers is described in Table 1.

If the contents of the MR, the CSR, the OPCR and ACR are changed while the receiver(s) and transmitter(s) are enabled, the registers will not be updated until both the transmitter(s) are empty and the receiver(s) disabled. The receiver(s) will be disabled, at the completion of the character being received, at the time of the register change. Normally these registers should not be updated without first making sure that the receiver(s) and transmitter(s) are disabled, and the C/T is stopped.

The bit formats of the Octal-UART registers are depicted in Table 2. These are shown for block A. The bit format for the other blocks is the same.

MR1 — Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET or by a set pointer command applied via CR. After reading or writing MR1, the pointers are set at MR2.

MR1[7] — Receiver Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the receiver. This output is manually asserted and negated by commands applied via the command register. MR1[7] = 1 causes RTSN to be automatically negated upon receipt of a valid start bit if the receiver FIFO is full. RTSN is reasserted when an empty FIFO position is available. This feature can be used to prevent overrun in the receiver, by using the RTSN output signal, to control the CTS input of the transmitting device.

MR1[6] — Receiver Interrupt Select

This bit selects either the receiver ready status (RxRDY) or the FIFO full status (FFULL) to be used for CPU interrupts.

MR1[5] — Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). 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 accumulation (logical-OR) of the status for all characters coming to the top of the FIFO, since the last reset error command was issued.

MR1[4:3] — 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. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

MR1[2] — Parity Type Select

This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1[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 'wake-up' mode, it selects the polarity of the A/D bit.

MR1[1:0] — 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.

MR2 — Mode Register 2

MR2 is accessed when the channel a MR pointer points to MR2, which occurs after any access to MR1. Accesses to MR2 do not change the pointer.

MR2[7:6] — Mode Select

The Octal-UART can operate in one of four modes: MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[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 TxD 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 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 communications continue normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[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 TxD output is held High.
4. The RxD 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 MR2[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxD 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., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not 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.

When switching in and out of the various modes, the selected mode is activated at the completion of all transmitted and received characters. Likewise, if a mode is deselected, the device will switch out of the mode at the completion of the current transmit and/or receive characters.

MR2[5] — Transmitter Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the transmitter. This output is manually asserted and negated by appropriate commands issued via the command register. MR2[5] = 1 causes RTSN to be reset automatically one bit time after the characters in the transmit shift register and in the THR (if any) are completely transmitted (includes 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: MR2[5] = 1.
2. Enable transmitter.
3. Assert RTSN via command.
4. Send message.
5. Verify the next to last character of the message is being sent by waiting until transmitter ready is asserted. Disable transmitter after the last character is loaded into the THR.
6. The last character will be transmitted and RTSN will be reset one bit time after the last stop bit.

MR2[4] — Clear-to-Send Control

The state of this bit determines if the CTSN input (MPI) controls the operation of the transmitter. If this bit is 0, CTSN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSN each time it is ready to send a character. If it is asserted (Low), the character is transmitted. If it is negated (High), the TxD output remains in the marking state and the transmission is delayed until CTSN goes Low. Changes in CTSN, while a character is being transmitted, do not affect the transmission of that character. This feature can be used to prevent overrun of a remote receiver.

MR2[3:0] — 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. In all cases, 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). If an external 1× clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	RxINT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHARACTER	
MR1	0 = no 1 = yes	0 = RxRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = special mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		TxRTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2	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

NOTE:
*Add 0.5 to values shown for 0 – 7, if channel is programmed for 5 bits/character.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSR	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISCELLANEOUS COMMANDS				DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CR	See Text				0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVED BREAK	**FRAMING ERROR**	**PARITY ERROR**	**OVERRUN ERROR**	**TxEMT**	**TxRDY**	**FFULL**	**RxRDY**
SR	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

NOTE:
*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 must be reset when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
		MPOb PIN FUNCTION SELECT			**POWER-DOWN MODE**	**MPOa PIN FUNCTION SELECT**		
OPCR	not used	000 = RTSN 001 = C/TO 010 = TxC (1×) 011 = TxC (16×) 100 = RxC (1×) 101 = RxC (16×) 110 = TxRDY 111 = RxRDY/FF			0 = off 1 = on *	000 = RTSN 001 = C/TO 010 = TxC (1×) 011 = TxC (16×) 100 = RxC (1×) 101 = RxC (16×) 110 = TxRDY 111 = RxRDY/FF		

NOTE:
*Only OPCR[3] in block A controls the power-down mode.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	BRG SET SELECT	**COUNTER/TIMER MODE AND SOURCE**			**DELTA MPI1bINT**	**DELTA MPI0bINT**	**DELTA MPI1aINT**	**DELTA MPI0aINT**
ACR	0 = set 1 1 = set 2	See Text			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IPCR	**DELTA MPI1b**	**DELTA MPI0b**	**DELTA MPI1a**	**DELTA MPI0a**	**MPI1b**	**MPI0b**	**MPI1a**	**MPI0a**
	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

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ISR	**MPI PORT CHANGE**	**DELTA BREAKb**	**RxRDY/ FFULLb**	**TxRDYb**	**COUNTER READY**	**DELTA BREAKa**	**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

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MPI PORT CHANGE INT	**DELTA BREAKb INT**	**RxRDY/ FFULLb INT**	**TxRDYb INT**	**COUNTER READY INT**	**DELTA BREAKa INT**	**RxRDY/ FFULLa INT**	**TxRDYa INT**
IMR	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

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	C/T[15]	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**
CTUR								

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	C/T[7]	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**
CTLR								

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Input Port Register	**MPI3b**	**MPI2b**	**MPI3a**	**MPI2a**	**MPI1b**	**MPI0b**	**MPI1a**	**MPI0a**
	0 = Low	0 = Low	0 = Low	0 = Low	0 = Low	0 = Low	0 = Low	0 = Low
	1 = High	1 = High	1 = High	1 = High	1 = High	1 = High	1 = High	1 = High

CSR — Clock Select Register

CSR[7:4] — Receiver Clock Select
When using a 3.6864MHz crystal or external clock input, this field selects the baud rate clock for the receiver as shown in Table 3.

The receiver clock is always a 16× clock, except for CSR[7:4] = 1111. When MPI3 is selected as the input, MPI3a is for channel a and MPI3b is for channel b.

CSR[3:0] — Transmitter Clock Select
This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3 except as follows.

CSR[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	MPI2 — 16×	MPI2 — 16×
1 1 1 1	MPI2 — 1×	MPI2 — 1×

When MPI2 is selected as the input, MPI2a is for channel a and MPI2b is for channel b.

CR — Command Register

CR is used to write commands to the Octal-UART.

CR[7:4] — Miscellaneous Commands
The encoded value of this field may be used to specify a single command as follows:

0000 No command.

0001 Reset MR pointer. Causes the MR pointer to point to MR1.

0010 Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO pointer is reset to the first location.

0011 Reset transmitter. Resets the transmitter as if a hardware reset had been applied.

0100 Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[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.

0101 Reset break change interrupt. Causes the break detect change bit in the Interrupt Status Register (ISR[2 or 6]) to be cleared to zero.

0110 Start break. Forces the TxD 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 break is delayed until that character or any others loaded after it has been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break.

0111 Stop break. The TxD line will go High (marking) within two bit times. TxD will remain High for one bit time before the next character, if any, is transmitted.

1000 Assert RTSN. Causes the RTSN output to be asserted (Low).

1001 Negate RTSN. Causes the RTSN output to be negated (High).

1010 Set special time out mode with this channel as the channel to restart the C/T as each receive character is transferred from shift register to RHR.

1011 Reserved.

1100 Reset special time out mode.

1101 Reserved.

111x Reserved for testing.

CR[3] — Disable 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.

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

CR[1] — Disable 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 wake-up mode is programmed, the receiver operates even if it is disabled (see Wake-Up Mode).

CR[0] — Enable Receiver
Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

Table 3. Baud Rate

CSR[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	38.4k
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	MPI3 — 16×	MPI3 — 16×
1 1 1 1	MPI3 — 1×	MPI3 — 1×

SR — Channel Status Register

SR[7] — 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 bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the change in break bit in the ISR (ISR[6 or 2]) is set. ISR[6 or 2] is also set when the end of the break condition, as defined above, is detected. The break detect circuitry is capable of detecting breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must last until the end of the next character in order for it to be detected.

SR[6] — Framing Error (FE)
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.

SR[5] — Parity Error (PE)
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 'wake-up' mode, the parity error bit stores the received A/D bit.

SR[4] — Overrun Error (OE)
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.

SR[3] — Transmitter Empty (TxEMT)
This bit is set when the transmitter underruns, i.e., both the transmit holding register and the transmit shift register are empty. However, this bit is not set until one character has been transmitted. 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.

SR[2] — Transmitter Ready (TxRDY)
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, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

SR[1] — FIFO Full (FFULL)
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 FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL is not reset after reading the FIFO once.

SR[0] — Receiver Ready (RxRDY)
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, and no more characters are in the FIFO.

OPCR — Output Port Configuration Register

OPCR[6:4] — MPOb Output Select
This field programs the MPOb output pin to provide one of the following:

000 Request-to-send Active-Low output (RTSN). This output is asserted and negated via the command register. Mode RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.

001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTUR and CTLR. In the counter mode, the output remains High until the 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.

010 The 1× clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1× clock is output.

011 The 16× clock for the transmitter. This is the clock selected by CSR[3:0], and is a 1× clock if CSR[3:0] = 1111.

100 The 1× clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1× clock is output.

101 The 16× clock for the receiver. This is the clock selected by CSR[7:4], and is a 1× clock if CSR[7:4] = 1111.

110 The transmitter register empty signal, which is the same as SR[3].

111 The receiver ready or FIFO full signal.

OPCR[3] — Power-Down Mode Select
This bit, when set, selects the power-down mode. In this mode, the SCC2698 oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the SCC2698 in this mode. This bit is reset with RESET asserted. Note that this bit must be set to a logic 1 before power-down occurs. Only OPCR[3] in block A controls the power-down mode.

OPCR[2:0] — MPOa Output Select
This field programs the MPOa output pin to provide one of the same functions as described in OPCR[6:4].

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, 38.4k, 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 receiver and transmitter.

ACR[6:4] — Counter/Timer Mode and Clock Source Select
This field selects the operating mode of the counter/timer and its clock source (see Table 4).

The MPI pin available as the clock source is MPI a, c, e, and g only.

ACR[3:0] — MPI1b, MPI0b, MPI1a, MPI0a 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] — MPI1b, MPI0b, MPI1a, MPI0a 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 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] — MPI1b, MPI0b, MPI1a, MPI0a 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.

Table 4. ACR[6:4] Operating Mode

[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	MPI pin
0 0 1	Counter	MPI pin divided by 16
0 1 0	Counter	TxC — 1× clock of the transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	MPI pin
1 0 1	Timer	MPI pin 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 — 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 is asserted (Low). 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 is provided regardless of the contents of the IMR.

ISR[7] — MPI Change-of-State

This bit is set when a change-of-state occurs at the MPI1b, MPI0b, MPI1a, MPI0a input pins. It is reset when the CPU reads the IPCR.

ISR[6] — Channel b Change in Break

This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[5] — Receiver Ready or FIFO Full Channel b

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it 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 receive FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read.

If programmed as FIFO full, it 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 FIFO is read and there is no character in the receive shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[4] — Transmitter Ready Channel b

This bit is a duplicate of TxRDY (SR[2]).

ISR[3] — Counter Ready

In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command. It is initialized to '0' when the chip is reset.

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

ISR[2] — Channel a Change in Break

This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[1] — Receiver Ready or FIFO Full Channel a

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it 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 receive FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it 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 FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[0] — Transmitter Ready Channel a

This bit is a duplicate of TxRDY (SR[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 a '1', the INTRN output is asserted (Low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no affect on the INTRN output. Note that the IMR does not mask 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 is 0002_{16}.

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 causes the counter to begin a new cycle using the values in CTU and CTL. 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. The command, however, does not stop the C/T.

The generated square wave is output on MPO 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. Counting begins upon receipt of a start counter command. Upon reaching the terminal count, the counter ready interrupt bit, ISR[3], is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains High until the 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 eight bits of the counter can 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 eight bits to the upper eight bits occurs between the times that both halves of the counter are read. However, a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
V_{CC}	Voltage from V_{CC} to GND[3]	-0.5 to +7.0	V
V_S	Voltage from any pin to GND[3]	-0.5 to V_{CC} ±5%	V
P_D	Power dissipation	1	W

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

SYMBOL	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			V
V_{IH}	Input High-voltage (X1/CLK)		$0.8V_{CC}$	V_{CC}		V
V_{OL}	Output Low-voltage	I_{OL} = 2.4mA			0.4	V
V_{OH}	Output High-voltage (except OC outputs)	I_{OH} = -400μA	2.4			V
I_{IL}	Input leakage current (except MPI pins)	V_{IN} = 0 to V_{CC}	-10		10	μA
I_{IL}	Input leakage current for MPI pins	V_{IN} = 0 to V_{CC}	-50		20	μA
I_{LL}	Data bus 3-State leakage current	V_O = 0 to V_{CC}	-10		10	μA
I_{X1L}	X1/CLK Low-input current	V_{IN} = 0, X2 floated	100	-10	0	μA
I_{X1H}	X1/CLK High-input current	V_{IN} = V_{CC}, X2 floated	0	10	100	μA
I_{OD}	Open-drain output leakage current	V_O = 0 to V_{CC}			10	μA
I_{CC}	Power supply current during power down				2	mA
I_{CC}	Power supply current				30	mA

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 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.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V, as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5V ±5%[4, 5, 6, 7]

SYMBOL	PARAMETER	TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 3)					
t_{RES}	RESET pulse width	200			µs
Bus timing (Figure 4)[8]					
t_{AS}	A0 - A5 setup time to RDN, WRN Low	10			ns
t_{AH}	A0 - A5 hold time from RDN, WRN High	0			ns
t_{CS}[9]	CEN setup time to RDN, WRN Low	0			ns
t_{CH}[9]	CEN hold time from RDN, WRN High	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN Low			200	ns
t_{DF}	Data bus floating after RDN High			80	ns
t_{DS}	Data setup time before WRN High	100			ns
t_{DH}	Data hold time after WRN High	10			ns
t_{RWD}[10]	High time between reads and/or writes	100			ns
MPI and MPO timing (Figure 5)[8]					
t_{PS}	MPI input setup time before RDN Low	0			ns
t_{PH}	MPI input hold time after RDN High	0			ns
t_{PD}	MPO output valid after WRN High			250	ns
Interrupt timing (Figure 6)					
t_{IR}	INTRN High from:				
	Read RHR (RxRDY/FFULL interrupt)			270	ns
	Write THR (TxRDY, TxEMT interrupt)			270	ns
	Reset command (break change interrupt)			270	ns
	Reset command (MPI change interrupt)			270	ns
	Stop C/T command (counter interrupt)			270	ns
	Write IMR (clear of interrupt mask bit)			270	ns
Clock timing (Figure 7)					
t_{CLK}	X1/CLK High or Low time	120			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4	MHz
t_{CTC}	CTCLK High or Low time	120			ns
f_{CTC}	CTCLK frequency	0'		4	MHz
t_{RX}	RxC High or Low time	200			ns
f_{RX}	RxC frequency (16×)	0'		2	MHz
	(1×)	0		1	MHz
t_{TX}	TxC High or Low time	200			ns
f_{TX}	TxC frequency (16×)	0'		2	MHz
	(1×)	0'		1	MHz
Transmitter timing (Figure 8)					
t_{TXD}	TxD output delay from TxC Low			350	ns
t_{TCS}	TxC output delay from TxD output data	0		150	ns
Receiver timing (Figure 9)					
t_{RXS}	RxD data setup time to RxC High	50			ns
t_{RXH}	RxD data hold time from RxC High	100			ns

NOTES:

[7] Test condition for outputs: C_L = 150pF, except interrupt outputs. Test condition for interrupt outputs: C_L = 50pF, R_L = 2.7kΩ to V_{CC}.

[8] Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are ANDed internally. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle.

[9] If CEN is used as the 'strobing' input, the parameter defines the minimum High times between one CEN and the next. The RDN signal must be negated for t_{RWD} to guarantee that any status register changes are valid.

[10] Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.

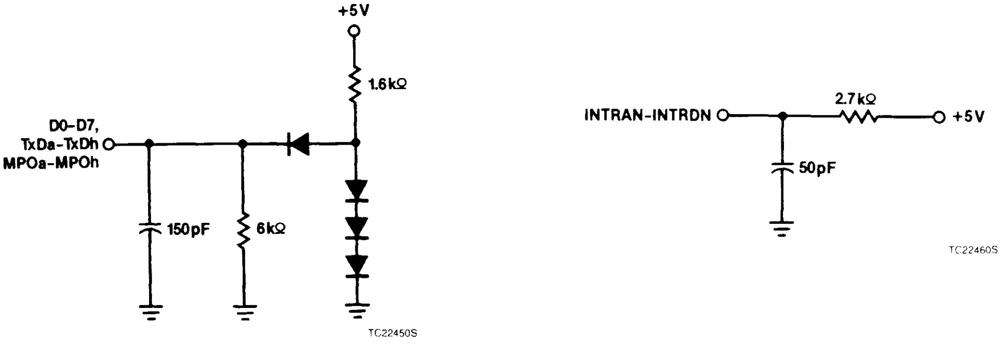


Figure 2. Test Conditions on Outputs

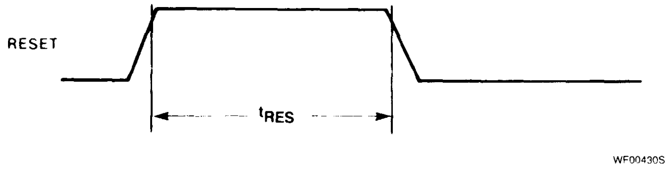


Figure 3. Reset Timing

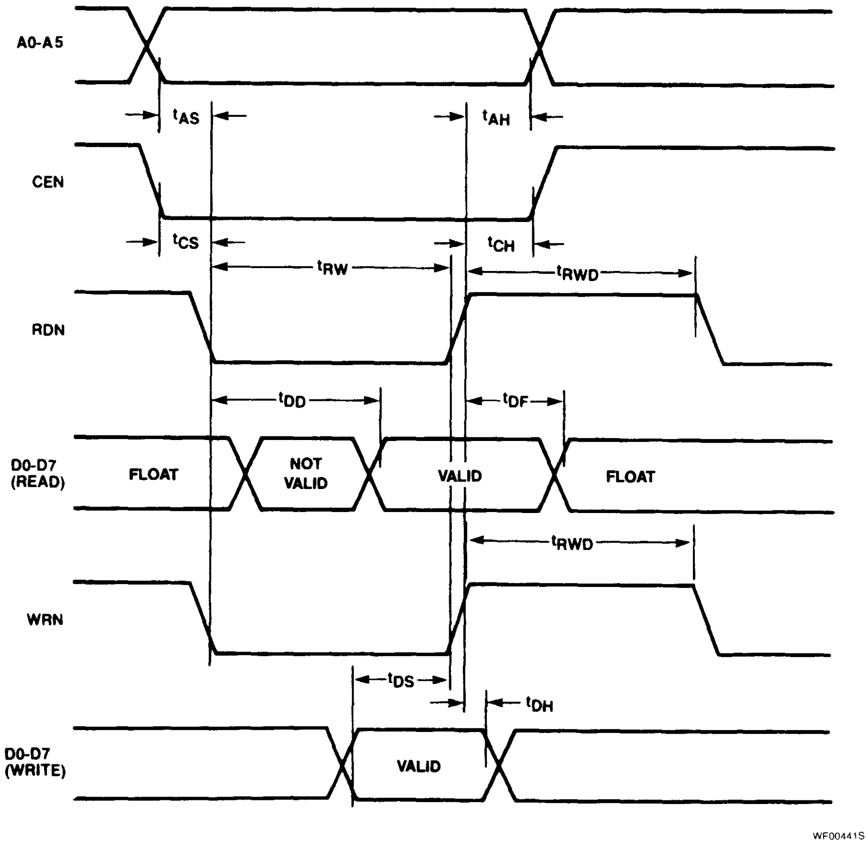


Figure 4. Bus Timing

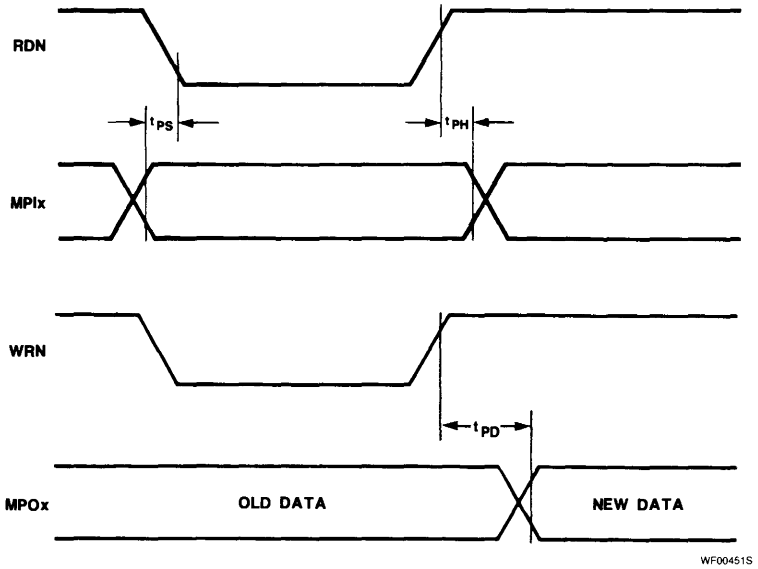
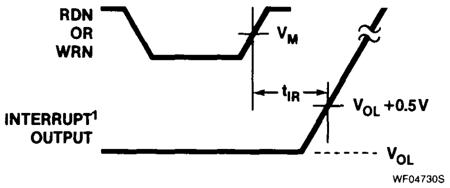


Figure 5. Port Timing



NOTES:

1. INTRN or MPO when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5V above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 6. Interrupt Timing

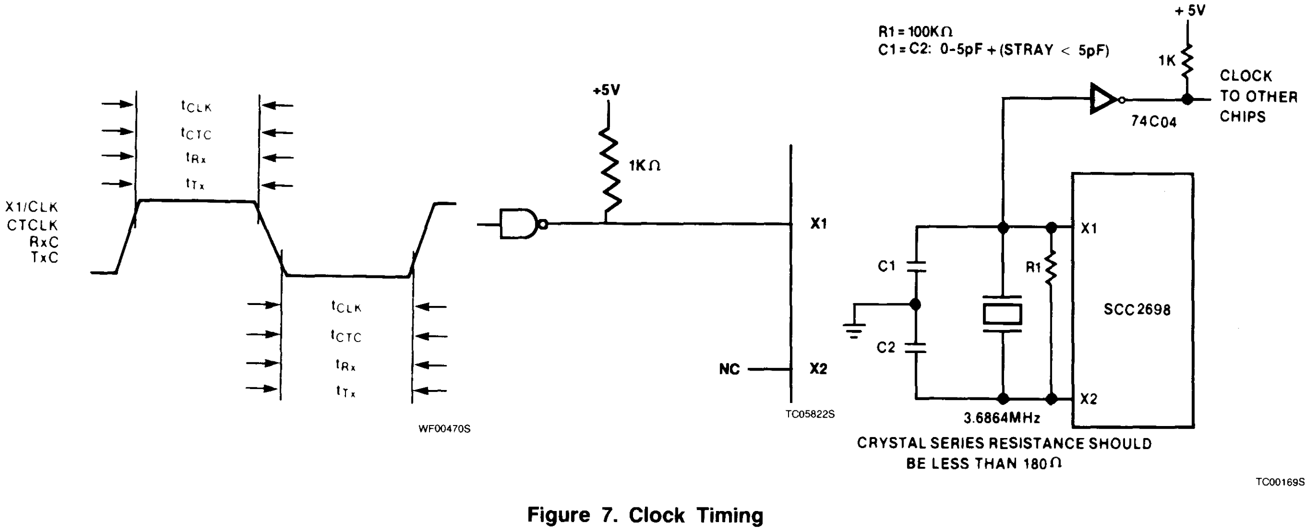


Figure 7. Clock Timing

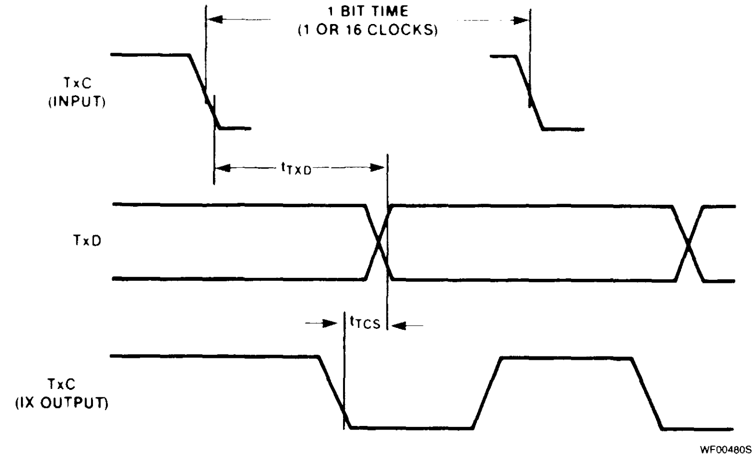


Figure 8. Transmit

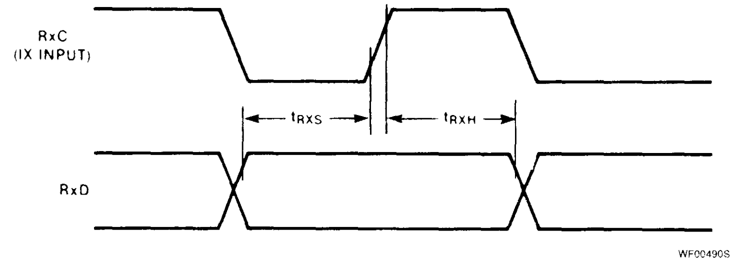
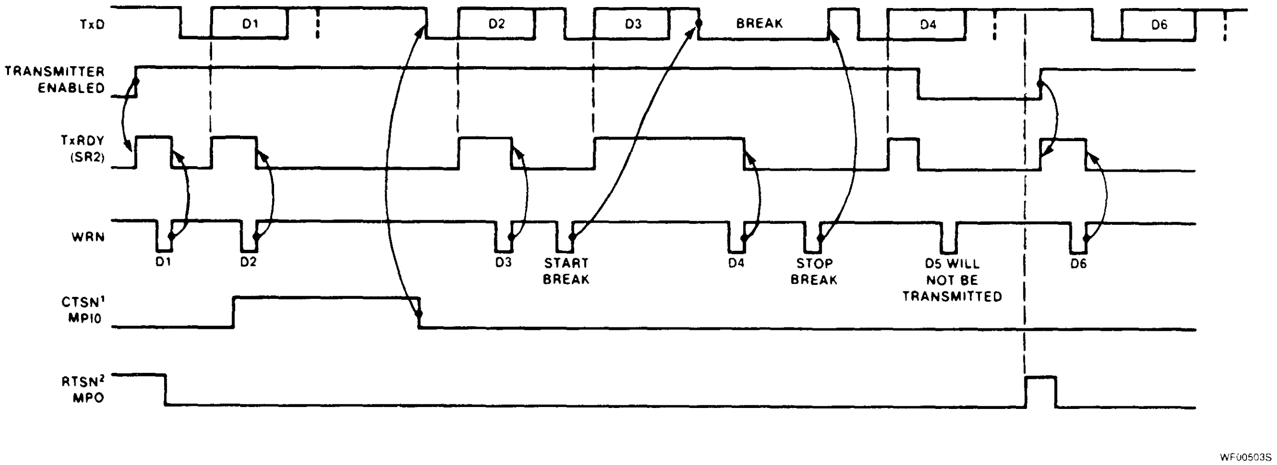
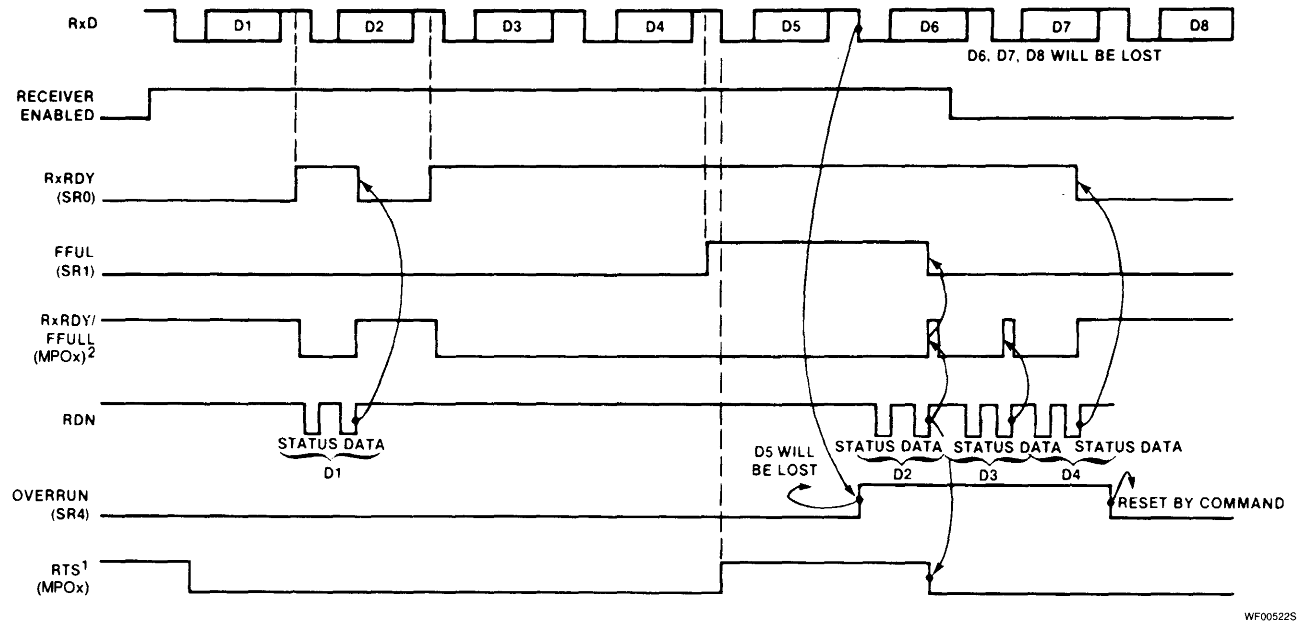


Figure 9. Receive



NOTES:
1. Timing shown for MR2[4] = 1.
2. Timing shown for MR2[5] = 1.

Figure 10. Transmitter Timing



NOTES:
1. Timing shown for MR1[7] = 1.
2. Shown for OPCR[6:4, 2:0] = 111 or 000 and MR1[6] = 0.

Figure 11. Receiver Timing

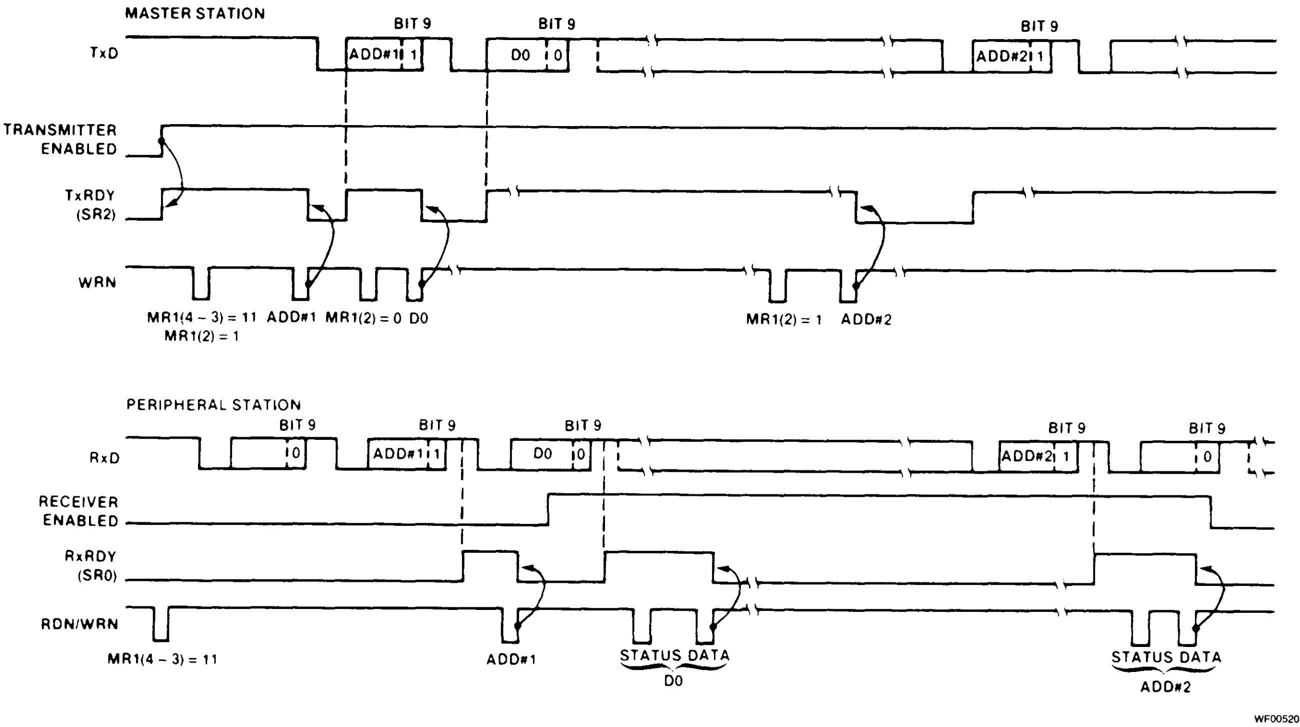


Figure 12. Wake-Up Mode

DEFINITIONS

Data Sheet Identification	Product Status	Definition
Objective Specification	**Formative or In Design**	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	**Preproduction Product**	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	**Full Production**	This data sheet contains Final Specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Signetics makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS
Signetics' Products are not designed for use in life support appliances, devices, or systems where malfunction of a Signetics Product can reasonably be expected to result in a personal injury. Signetics' customers using or selling Signetics' Products for use in such applications do so at their own risk and agree to fully indemnify Signetics for any damages resulting in such improper use or sale.

Signetics
a subsidiary of U.S. Philips Corporation

Signetics Corporation
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088-3409
Telephone 408/991-2000

98-8026-980 7298/DLR/5MCR10887

Signetics

AN410

SCC2698

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

Application Note

Microprocessor Division

DEVICE ARCHITECTURE

The Signetics SCC2698 Octal-UART is composed of four blocks, each logically equivalent to a 2681 or 2692 DUART. Each block is composed of two channels, a counter/timer, and an interrupt control section. The channels are matched to the blocks as shown in figure 1. The blocks are indicated by capital letters A, B, C, and D; the channels are indicated by lower-case letters a, b, c, d, e, f, g, and h. All registers act either on a block or an individual channel.

BLOCK A CHANNELS a, b	BLOCK C CHANNELS e, f
BLOCK B CHANNELS c, d	BLOCK D CHANNELS g, h

Figure 1. Channel Architecture

Registers that affect a block:
IPCR/ACR
ISR/IMR
CTU/CTUR
CTL/CTLR
IPR/OPCR
START C/T
STOP C/T

Registers that affect a channel:
MR1/MR2
SR/CSR
CR
RHR/THR

X1/CLK SOURCES

The SCC2698 must have a clock source connected to the X1 input at all times. It can be supplied by a crystal between the X1 and X2 pins, or by driving an external clock into the X1/CLK input. The frequency must be between 2.0 and 4.0MHz for correct device operation; 3.6864MHz is the nominal frequency which is used to obtain the standard baud rates listed for the internal baud rate generator.

X1/X2 Crystal

The SCC2698 oscillator circuitry consists of an inverting amplifier and a feedback resistor which are used to implement a Pierce oscillator (see figure 2). This circuitry will cause the crystal attached between the X1 and X2 pins to go into anti-resonant (parallel) operation. So, while a number of crystal and capacitor combinations will work, obtaining a parallel calibrated crystal and adjusting the external capacitor values until the total circuit capacitance matches the capacitance specified for the crystal will result in the most accurate frequency value. Using 24pF capacitors and the parallel crystal recommended below will give accurate, reliable results. The frequency will vary slightly depending on the amount of stray capacitance in the individual circuit, but will typically be off no more than 0.01%. The frequency can be adjusted by trimming the external capacitors; larger capacitors lower the oscillator's frequency and smaller ones raise it.

A source for the 3.6864MHz crystal is: Saronix, Palo Alto, CA. From California, call (800) 422-3355; outside California call (800) 227-8974. Request part number NYP037-20.

Externally Driven Clock

The most important point in using an external source to drive the X1/CLK input is to meet the V_{IH} specification of $0.8V_{CC}$ (4.0V at V_{CC} = 5.0V). This can be insured by using an open collector buffer with a pull-up resistor to Vcc to drive the X1 input. Also, when driving a clock into X1, be sure to leave the X2 pin open; grounding it will kill the oscillation.

BAUD RATE GENERATION TECHNIQUES

There are 18 standard baud rates available using the internal baud rate generator when the X1/CLK frequency is 3.6864MHz. These are selected by ACR[7] and by CSR [7:4] for the receiver and CSR[3:0] for the transmitter.

The baud rate generator table follows:

Baud Rate Generator Table

CSR[7:4] (or [3:0])	ACR[7] = 0	ACR[7] = 1
0000	50	75
0001	110	110
0010	134.5	38.4K
0011	200	150
0100	300	300
0101	600	600
0110	1,200	1,200
0111	1,050	2,000
1000	2,400	2,400
1001	4,800	4,800
1010	7,200	1,800
1011	9,600	9,600
1100	38.4K	19.2K

The baud rate generator can also be used to generate other baud rates by using a different X1/CLK frequency. For this case, each ACR[7] and CSR combination gives a different division ratio. The division ratio table follows:

Division Ratio Table

CSR[7:4] (or [3:0])	ACR[7] = 0	ACR[7] = 1
0000	73,728	49,152
0001	33,536	33,536
0010	27,392	96
0011	18,432	24,576
0100	12,288	12,288
0101	6,144	6,144
0110	3,072	3,072
0111	3,520	1,840
1000	1,536	1,536
1001	768	768
1010	512	2,048
1011	384	384
1100	96	192

The baud rate can be calculated by dividing the X1/CLK frequency by the appropriate division ratio. For example, if the X1/CLK frequency = 3MHz, ACR[7] = 0 and CSR = CC hex, the division ratio is 96 and both the receiver and transmitter will use a 31.25K baud rate.

Externally Generated Baud Rate Clock

An externally generated baud rate clock can be used for each transmitter and receiver using multi-purpose inputs 2 and 3. These inputs are only available in the 84-pin PLCC package. MPI3 is used as a 16X clock source for the receiver by programming CSR[7:4] = 1110 and as a 1X clock source by programming CSR[7:4] = 1111. MPI2 is used as a 16X clock source for the transmitter by programming CSR[3:0] = 1110 and a 1X clock source by programming CSR[3;0] = 1111. The maximum frequency that can be used as a 16X clock is 2MHz, which results in a baud rate of 125Kbps. The maximum frequency that can be used as a 1X clock is 1MHz, for a maximum baud rate of 1Mbps.

Counter/Timer as 16X Baud Rate Clock

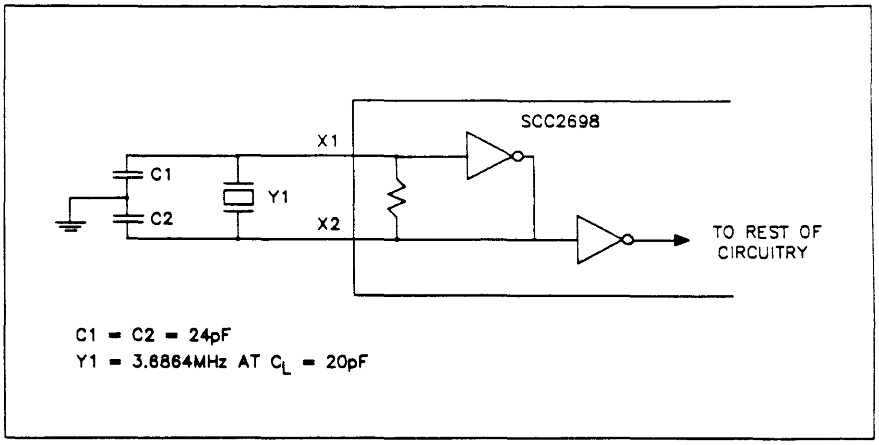
The counter/timer can be used in timer mode to divide the X1/CLK or an external clock. The output of the C/T is internally connected as a 16X clock source for the receiver by programming CSR[7:4] = 1101 and as a 16X clock source for the transmitter by programming CSR [3:0] = 1101. The clock source for the timer is selected by ACR[6:4], as follows:

ACR[6:4]	Timer Clock Source
100	MPI1 pin*
101	MPI1 pin divided by 16*
110	X1/CLK
111	X1/CLK divided by 16

*The MPI1 pin available as a clock source is MPI1 a, c, e and g only.

In addition, the CTUR and CTLR registers must be programmed with the divisor value for the timer. The minimum allowable value to program is 0002 hex. The timer will generate a square wave with a period of twice the number of timer clock periods programmed into CTUR/CTLR. The resultant baud rate is calculated by:

$$\left(\frac{\text{timer clock}}{2 \text{ X CTUR/CTLR VALUE}}\right) \div 16$$



C1 = C2 = 24pF
Y1 = 3.6864MHz AT C_L = 20pF

Figure 2. Pierce Oscillator Using SCC2698 On-Board Oscillator Circuitry

The maximum baud rate available in this manner is 62.5Kbps. This is obtained with an X1/CLK = 4MHz and programming as shown in example 1.

Counter/Timer as 1X Baud Rate Clock

The timer can also be used as a 1X clock source for the transmitter by externally connecting the C/T output to the MPI2 (TCLK) input. In addition, the C/T output can be connected to the MPI3 (RCLK) input to provide a 1X clock source for the receiver, but care must be taken to have the timer output synchronized with the incoming data to ensure accurate data reception. These inputs are only available in the 84-pin PLCC packaged part. The C/T is set up as described in the last section. The resultant baud rate is calculated by:

$$\left(\frac{\text{timer clock}}{2 \text{ X CTUR/CTLR VALUE}}\right)$$

The maximum baud rate available in this manner is 1Mbps. This is obtained with an X1/CLK = 4MHz, MPO externally connected to MPI2 and MPI3 and programmed as shown in example 2.

RTS/CTS FLOW CONTROL

One way to achieve flow control with the SCC2698 is to have the request to send (RTS) output, controlled by the receiver, connected to the clear to send (CTS) input, which enables the transmitter. RTS is controlled by the receiver when MR1[7] = 1, and the multi-purpose output (MPO) is used as the RTS output when OPCR[6:4] = 0 and OPCR [2:0] = 0.

Initially, the RTS output must be asserted by writing CR[7:4] = 1000 immediately after enabling the receiver. After this, RTS will automatically negate upon receipt of a valid start bit if the receiver FIFO is full, and will reassert when an empty FIFO position is available. CTS enables the transmitter and MPI0 is used as the CTS input pin when MR2[4] = 1. When CTS is negated, the transmitter will complete transmitting a character already in progress, but will not transmit a character waiting in the THR. The TxD output will then go into the marking state and the transmitter clock will be stopped. The Tx empty bit will not be set (even if the transmitter is empty) until the transmitter clock starts running again. When CTS is reasserted, the transmitter will start again, transmitting if a character is waiting in the THR or setting the empty bit if not. Be careful if the transmitting device is not a Signetics part, as some devices will

Example 1

```
CSR=DD HEX            ; Rx & Tx USE TIMER AS 16X BAUD RATE
                      ; CLOCK
ACR[7]=DON'T CARE
ACR[6:4]=110          ; X1/CLK IS TIMER 1X CLOCK SOURCE
CTUR/CTLR=0002 HEX    ; TIMER HAS DIVISOR OF 2
```

Example 2

```
CSR=FF HEX            ; Rx USES MPI3 AND Tx USES MPI2 AS A
                      ; 1X BAUD RATE CLOCK
ACR[7]=DON'T CARE
ACR[6:4]=110          ; X1/CLK IS TIMER 1X CLOCK SOURCE
CTUR/CTLR=0002 HEX    ; TIMER HAS DIVISOR OF 2
OPCR[6:4]=001         ; MPO IS C/T OUTPUT
```

Note that this example is shown to further demonstrate the device's versatility in baud rate generation. For most applications, this method is good for the transmit clock, but may not be a recommended method for accurate data reception, unless the timer clock source is synchronized with the incoming data.

Example 3

```
RXRDY:    MOVE.B   #$00,$30000       ; DUMMY WRITE
          MOVE.B   SR,D3             ; READ STATUS REGISTER
          BTST     #0,D3             ; CHECK FOR RXRDY
          BEQ      RXRDY             ; IF NOT, KEEP CHECKING
```

transmit both the character in the transmitter shift register and the character in the transmitter holding register when CTS is negated. If this occurs, the receiver may be overrun by a fifth character.

Modem control configurations may require RTS to be controlled by the transmitter, asserted for the entire time the message is being sent, and negated on completion of the transmission. RTS is controlled by the transmitter when MR2[5] = 1, and MPO is used as the RTS output when OPCR[6:4] = 0 and OPCR[2:0] = 0. The RTS output must be asserted by writing CR[7:4] = 1000 after the transmitter has been enabled and before the first byte of the message is loaded into the THR. The transmitter should be disabled after the last character of the message is loaded into the THR. This last character will be transmitted and RTS will be negated one bit time after the last stop bit.

MULTI–PURPOSE INPUTS

There are four multi–purpose inputs provided for each channel for the 84–pin PLCC package, MPI0, MPI1, MPI2 and MPI3. The DIP package has one multi–purpose input for each channel, MPI0. The current state for each MPI can be read from the input port register (IPR). Each input can be used as a general purpose input, to be interpreted as the user desires. In addition, each input can be programmed to provide a specific defined input, as follows:

MPI0

- Current state also in the input port change register (IPCR).
- Has a change of state indicator in IPCR, which can also be used to generate an interrupt.
- When MR2[4] = 1, it is used as CTSN input to enable the transmitter.

MPI1

- Current state also in the input port change register (IPCR).
- Has a change of state indicator in IPCR, which can also be used to generate an interrupt.
- When ACR[6:4] = 000, MPI1a, MPI1c, MPI1e, and MPI1g are used as a 1X counter clock source.
  When ACR[6:4] = 001, MPI1a, MPI1c, MPI1e, and MPI1g are used as a 16X counter clock source.
  When ACR[6:4] = 100, MPI1a, MPI1c, MPI1e, and MPI1g are used as a 1X timer clock source.
  When ACR[6:4] = 101, MPI1a, MPI1c, MPI1e, and MPI1g are used as a 16X timer clock source.
- For all four of these programmed cases, MPI1b, MPI1d, MPI1f, and MPI1h stay as general purpose inputs, since there is only one C/T clock for each block.

MPI2

- When CSR[3:0] = 1110, it is used as the transmitter 16X baud rate clock input.
- When CSR[3:0] = 1111, it is used as the transmitter 1X baud rate clock input.

MPI3

- When CSR[7:4] = 1110, it is used as the receiver 16X baud rate clock input.
- When CSR[7:4] = 1111, it is used as the receiver 1X baud rate clock input.

BUS INTERFACE

The internal control signals for strobing read and write cycles are obtained by internally logically ANDing CEN with RDN, and CEN with WRN. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle. However, the RDN line cannot be left asserted with just the CEN line pulsed. The RDN signal must be negated between reads, because the status register can only be updated at that time.

When using a 68000 type bus interface with a static R/WN output, either a hardware or a software method of pulsing the RDN line must be designed in. An example of a hardware solution is to logically AND the CEN signal with the inversion of R/WN to provide RDN (see attached schematic). RDN can also be negated by the software, the inversion of R/WN can be used for RDN, and a dummy write can be performed between consecutive reads. For example, in a polling loop of the status register, see example 3.

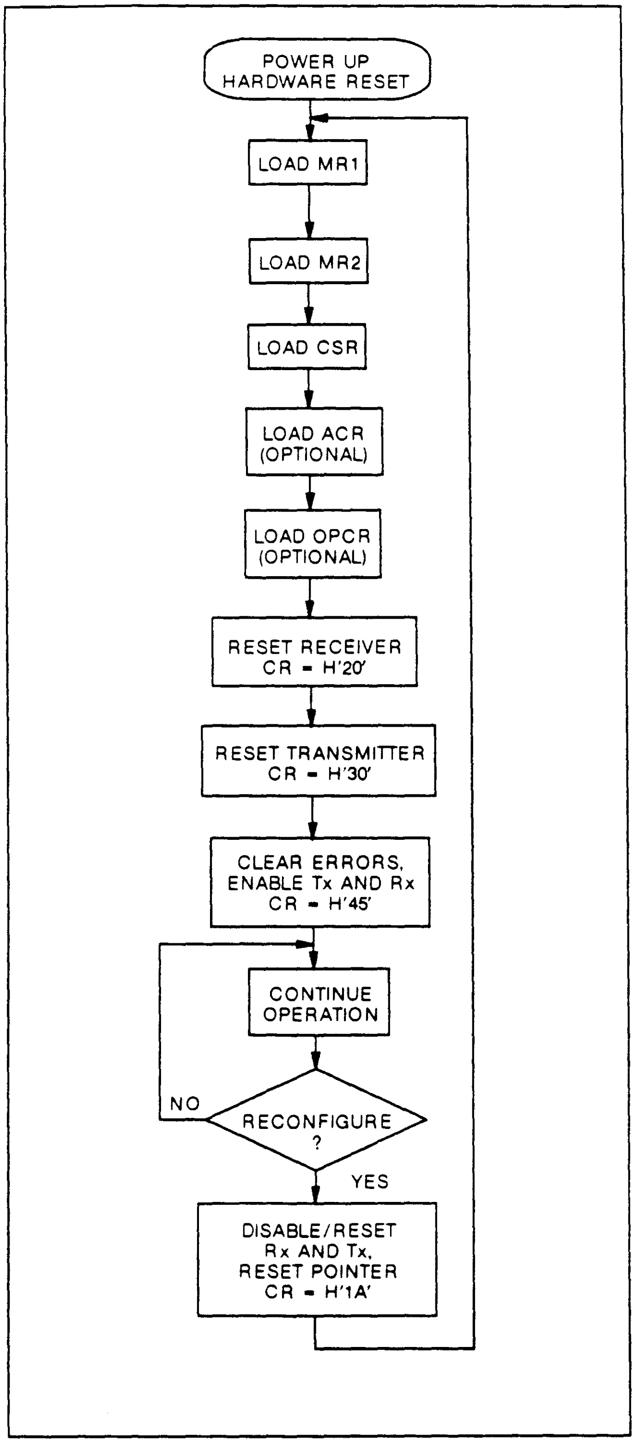


Figure 3. Basic Initialization

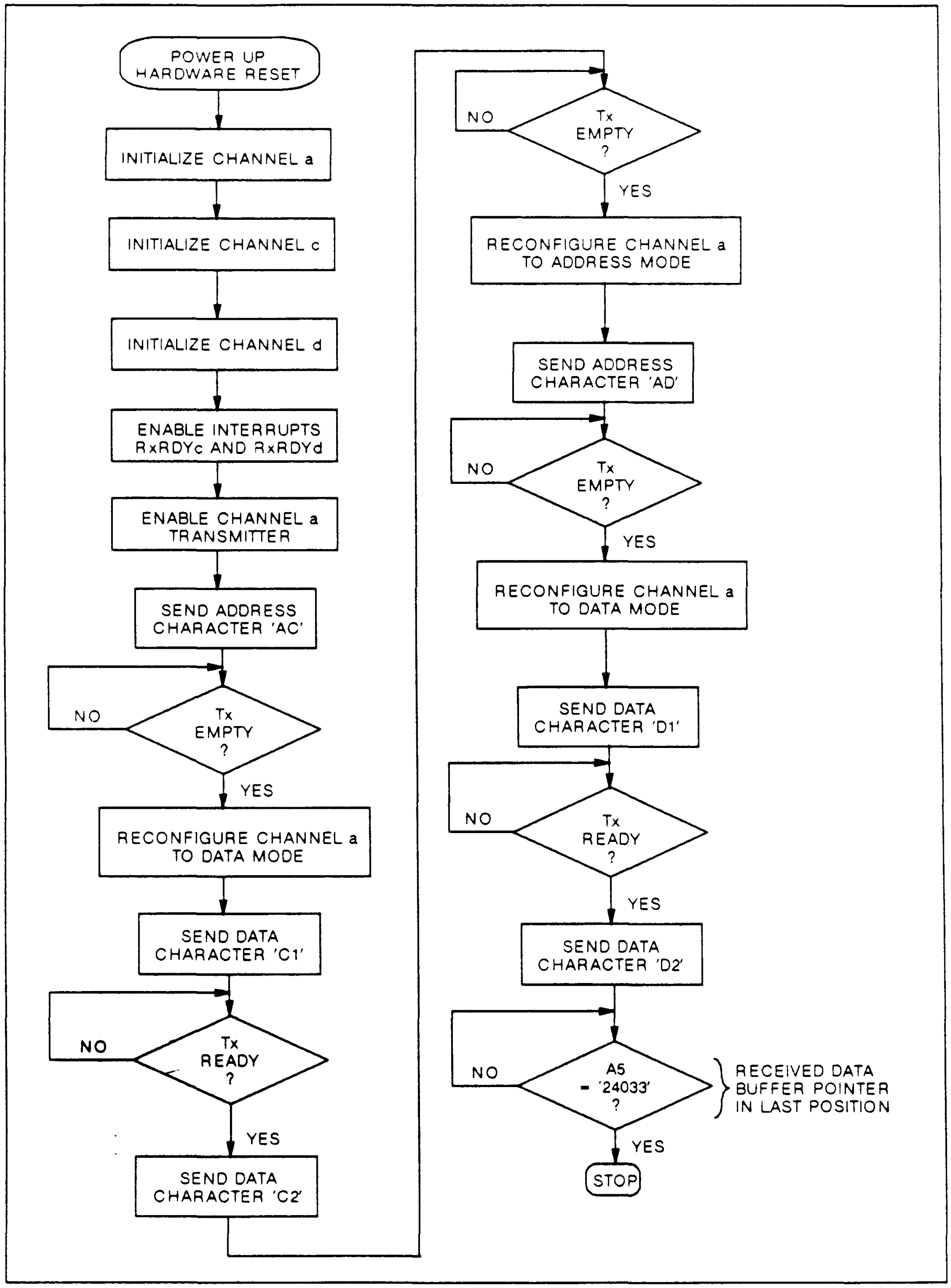


Figure 4. Wake-up Mode Example, Main Program Flowchart

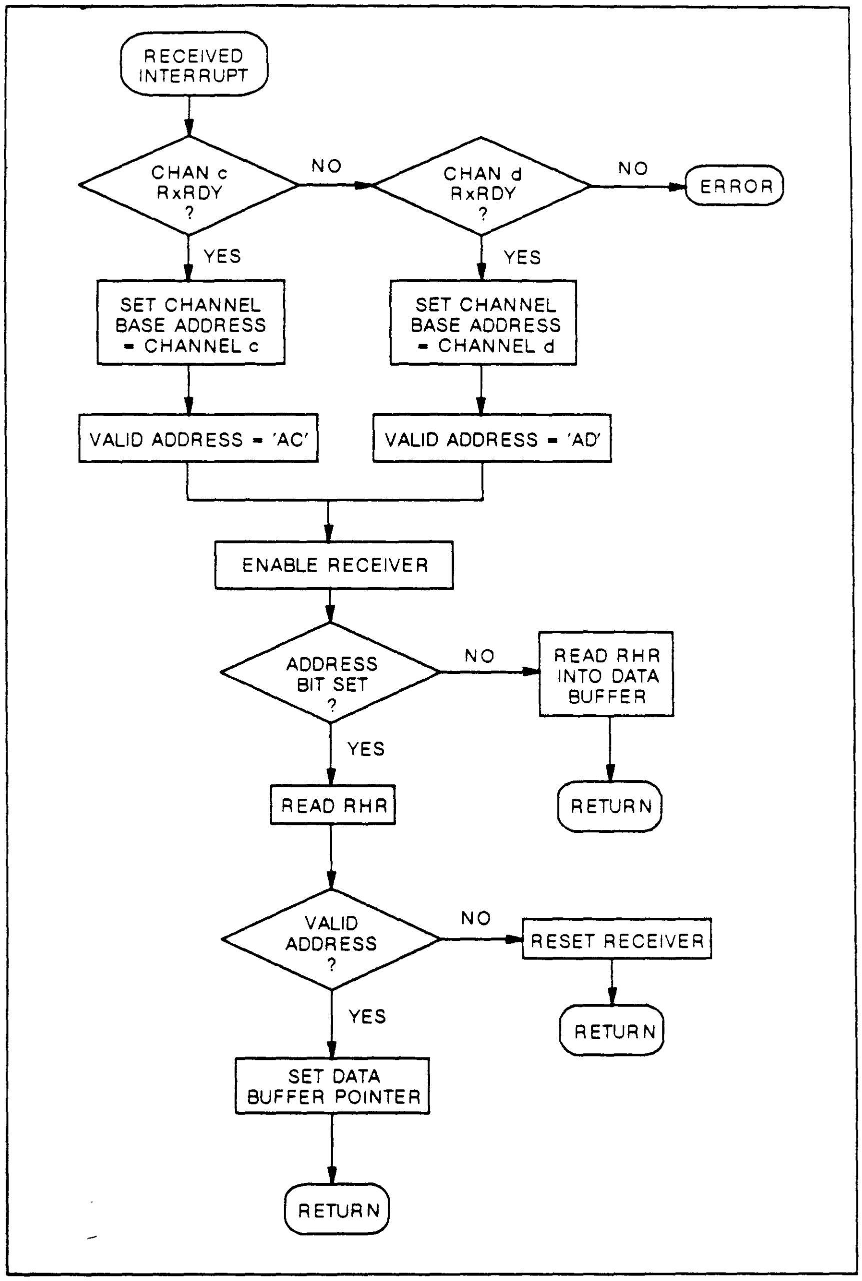


Figure 5. Wake-up Mode Example, Interrupt Routine Flowchart

LOCAL LOOPBACK MODE PROGRAM EXAMPLE

```
; THIS IS PROGRAM 2698 LOCAL LOOP
; IT SEPARATELY TESTS EACH CHANNEL IN LOCAL LOOPBACK
; MODE, SENDING 256 CHARACTERS AND COMPARING EACH
; CHARACTER SENT WITH EACH ONE RECEIVED.

; D.IBARRA JULY 1987

;       BEGIN
CHANa   EQU     $74001        ;CHANNEL BASE ADDRESSES
CHANb   EQU     $74011
CHANc   EQU     $74021
CHANd   EQU     $74031
CHANe   EQU     $74041
CHANf   EQU     $74051
CHANg   EQU     $74061
CHANh   EQU     $74071
;
BLOCKA  EQU     $74001        ;BLOCK BASE ADDRESSES
BLOCKB  EQU     $74021
BLOCKC  EQU     $74041
BLOCKD  EQU     $74061
;
MR1     EQU     $0            ;CHANNEL REGISTER OFFSETS
MR2     EQU     $0
STATR   EQU     $2
CSR     EQU     $2
CR      EQU     $4
RHR     EQU     $6
THR     EQU     $6
;
IPCR    EQU     $8            ;BLOCK REGISTER OFFSETS
ACR     EQU     $8
ISR     EQU     $A
IMR     EQU     $A
CTU     EQU     $D
CTUR    EQU     $D
CTL     EQU     $E
CTLR    EQU     $E
IPR     EQU     $1A
OPCR    EQU     $1A
STRTCT  EQU     $1C
STOPCT  EQU     $1E
;
START:  MOVEA.L #CHANa,A2      ;TEST CHAN a
        JSR     INIT
        JSR     TEST
        MOVEA.L #CHANb,A2      ;TEST CHAN b
        JSR     INIT
        JSR     TEST
        MOVEA.L #CHANc,A2      ;TEST CHAN c
        JSR     INIT
        JSR     TEST
        MOVEA.L #CHANd,A2      ;TEST CHAN d
        JSR     INIT
        JSR     TEST
        MOVEA.L #CHANe,A2      ;TEST CHAN e
        JSR     INIT
        JSR     TEST
        MOVEA.L #CHANf,A2      ;TEST CHAN f
```

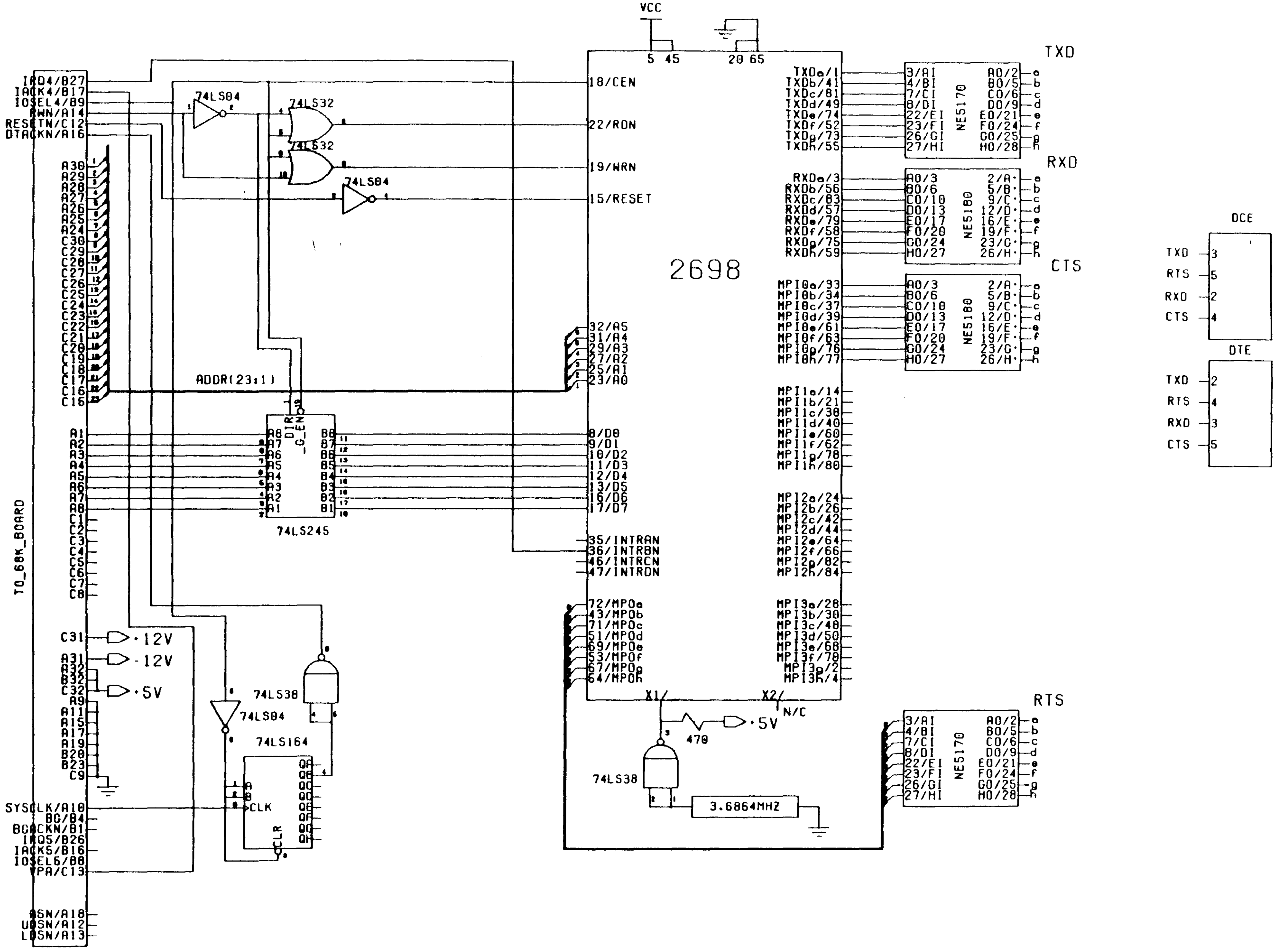
```
       JSR     INIT
       JSR     TEST
       MOVEA.L #CHANg,A2    ;TEST CHAN g
       JSR     INIT
       JSR     TEST
       MOVEA.L #CHANh,A2    ;TEST CHAN h
       JSR     INIT
       JSR     TEST
;
       MOVEA.L #0,A2        ;CLEAR ADDRESS REG.
STOP:  TRAP    #15

;
;-------SUBROUTINES-----------
;
INIT: MOVE.B #$1A,CR[A2]    ;DISABLE TX & RX, RESET POINTER
      MOVE.B #$13,MR1[A2]   ;NO PARITY, 8 BITS
      MOVE.B #$87,MR2[A2]   ;LOCAL LOOP, STOP=1
      MOVE.B #$66,CSR[A2]   ;TXC=RXC=BRG,1200 BAUD
      MOVE.B #$20,CR[A2]    ;RESET RECEIVER
      MOVE.B #$30,CR[A2]    ;RESET TRANSMITTER
      MOVE.B #$45,CR[A2]    ;CLR ERRORS, ENABLE TX-RX
      RTS
;
TEST: MOVE.W #$100,D7       ;CLEAR SEND REGISTER
AGAIN:SUBI.B #$1,D7         ;DEC. SEND CHAR.
      BEQ    DONE           ;UNTIL D7=0, THEN DONE
      MOVE.B D7,THR[A2]     ;TRANSMIT CHAR
      BSR    RXRDY          ;WAIT FOR RXRDY
      MOVE.B RHR[A2],D1     ;RECEIVE CHAR INTO D1
      CMP.B  D7,D1          ;SENT CHAR=RECEIVE CHAR ?
      BEQ    AGAIN          ;IF SO, KEEP SENDING
      TRAP   #15            ;STOP IF FAIL
DONE: RTS
;
RXRDY:MOVE.B STATR[A2],D3   ;STATUS REG. TO D3
      BTST   #0,D3          ;IS RECEIVER READY ?
      BEQ    RXRDY          ;IF NOT, WAIT
      RTS
;
      END    START
```

WAKE-UP MODE PROGRAM EXAMPLE

```
;  THIS IS PROGRAM 2698 WAKEUP
;  IT USES CHAN. a AS THE MASTER STATION, AND
;  CHANNELS c and d AS SLAVE RECEIVING STATIONS.
;  CHAN. a TRANSMITTER IS EXTERNALLY CONNECTED
;  TO c and d RECEIVERS. THIS PROGRAM IS INTERRUPT
;  DRIVEN, THE SLAVE STATIONS WILL INTERRUPT ON RXRDY.
;
;  D. IBARRA JULY 1987
;       BEGIN
;
CHANa    EQU    $74001         ;CHANNEL BASE ADDRESSES
CHANb    EQU    $74011
CHANc    EQU    $74021
CHANd    EQU    $74031
CHANe    EQU    $74041
CHANf    EQU    $74051
CHANg    EQU    $74061
CHANh    EQU    $74071
;
BLOCKA   EQU    $74001         ;BLOCK BASE ADDRESSES
BLOCKB   EQU    $74021
BLOCKC   EQU    $74041
BLOCKD   EQU    $74061
;
MR1      EQU    $0             ;CHANNEL REGISTER OFFSETS
MR2      EQU    $0
STATR    EQU    $2
CSR      EQU    $2
CR       EQU    $4
RHR      EQU    $6
THR      EQU    $6
;
IPCR     EQU    $8             ;BLOCK REGISTER OFFSETS
ACR      EQU    $8
ISR      EQU    $A
IMR      EQU    $A
CTU      EQU    $D
CTUR     EQU    $D
CTL      EQU    $E
CTLR     EQU    $E
IPR      EQU    $1A
OPCR     EQU    $1A
STRTCT   EQU    $1C
STOPCT   EQU    $1E
;
START:   MOVEA.L #CHANa,A1
         MOVEA.L #CHANa,A2    ;INIT CHAN a
         JSR     INIT
         MOVEA.L #CHANc,A2    ;INIT CHAN c
         JSR     INIT
         MOVEA.L #CHANd,A2    ;INIT CHAN d
         JSR     INIT
         MOVEA.L #BLOCKB,A3   ;ENABLE RXRDY INTERRUPTS
         JSR     SETINT
;
         MOVE.B #$04,CR[A1]   ;ENABLE TX CH. a
         MOVE.B #$AC,THR[A1]  ;SEND ADDRESS CHAR
         JSR     DTMODE       ;CHANGE TO DATA MODE
         MOVE.B #$C1,THR[A1] ;SEND DATA
         JSR     TXRDY       ;WAIT FOR TXRDY
         MOVE.B #$C2,THR[A1] ;SEND DATA
         JSR     ADMODE      ;CHANGE TO ADDRESS MODE
         MOVE.B #$AD,THR[A1] ;SEND ADDRESS CHAR
         JSR     DTMODE      ;CHANGE TO DATA MODE
         MOVE.B #$D1,THR[A1] ;SEND DATA
         JSR     TXRDY       ;WAIT FOR TXRDY
         MOVE.B #$D2,THR[A1] ;SEND DATA
;
WTDN:    MOVE.L A5,D1        ;MOVE DATA BUFFER POINTER TO D1
         CMP.L #$24033,D1    ;IS IT IN LAST POSITION ?
         BNE WTDN            ;IF NOT, WAIT UNTIL DONE
         MOVEA.L #0,A2       ;CLEAR ADD. REG.
         TRAP #15            ;TO INDICATE NORMAL END
;
;-----------------SUBROUTINES--------------
;
INIT:    MOVE.B #$1A,CR[A2]  ;DISABLE TX AND RX, RESET POINTER
         MOVE.B #$1F,MR1[A2] ;WAKEUP, 8 BITS, A/D=1
         MOVE.B #$07,MR2[A2] ;NORMAL, STOP=1
         MOVE.B #$66,CSR[A2] ;TXC=RXC=BRG, 1200 BAUD
         MOVE.B #$20,CR[A2]  ;RESET RX
         MOVE.B #$30,CR[A2]  ;RESET TX
         MOVE.B #$40,CR[A2]  ;CLEAR ERRORS
         RTS
;
SETINT: LEA.L  INT,A6        ;PUT INT. ROUTINE ADD. INTO A6
        MOVE.L A6,$10476     ;ADD. TO AUTO VECTOR #4 LOCATION
        MOVE.B #$22,IMR[A3]  ;INT. ON RXRDY BOTH CHANNELS
        RTS
;
DTMODE: JSR    TXEMTY        ;WAIT FOR TXEMPTY
        MOVE.B #$1A,CR[A1]   ;DISABLE TX AND RX, RESET POINTER
        MOVE.B #$1B,MR1[A1]  ;WAKEUP, 8 BITS, A/D=0
        MOVE.B #$20,CR[A1]   ;RESET RX
        MOVE.B #$30,CR[A1]   ;RESET TX
        MOVE.B #$44,CR[A1]   ;CLEAR ERRORS,ENABLE TX
        RTS
;
ADMODE: JSR    TXEMTY        ;WAIT FOR TXEMPTY
        MOVE.B #$1A,CR[A1]   ;DISABLE TX AND RX, RESET POINTER
        MOVE.B #$1F,MR1[A1]  ;WAKEUP, 8 BITS, A/D=1
        MOVE.B #$20,CR[A1]   ;RESET RX
        MOVE.B #$30,CR[A1]   ;RESET TX
        MOVE.B #$44,CR[A1]   ;CLEAR ERRORS,ENABLE TX
        RTS
;
TXEMTY: MOVE.B STATR[A1],D1  ;MOVE STATUS REG. TO D1
        BTST   #3,D1         ;IS TX EMPTY ?
        BEQ    TXEMTY        ;IF NOT, WAIT
        RTS
;
TXRDY:  MOVE.B STATR[A1],D1  ;MOVE STATUS REG. TO D1
        BTST   #2,D1         ;IS TXRDY ?
        BEQ    TXRDY         ;IF NOT, WAIT
        RTS
;
```

```
;-------------INTERRUPT ROUTINE-------------
;
INT:    MOVE.B  ISR[A3],D2      ;MOVE ISR TO D2
        BTST    #1,D2           ;IS CH c RXRDY ?
        BEQ     CHD             ;IF NOT, CHECK CH d
        MOVE.B  #$AC,D5         ;PUT CH c ADD. INTO D5
        MOVEA.L #CHANc,A2       ;SET BASE ADD.=CH c
        BRA     CHKAD           ;GO CHECK A/D BIT
CHD:    BTST    #5,D2           ;IS CH d RXRDY ?
        BEQ     ERROR           ;IF NOT, ERROR
        MOVE.B  #$AD,D5         ;PUT CH d ADD. INTO D5
        MOVEA.L #CHANd,A2       ;SET BASE ADD.=CH d
CHKAD:  MOVE.B  #$01,CR[A2]     ;ENABLE RX
        MOVE.B  STATR[A2],D3 ;READ STATUS REG. TO D3
        BTST    #5,D3           ;IS IT ADDRESS CHAR. ?
        BEQ     DATA            ;IF NOT, GO READ DATA
        MOVE.B  RHR[A2],D4      ;READ ADDRESS CHAR.
        CMP     D4,D5           ;IS IT MY ADDRESS ?
        BEQ     SETBUF          ;IF IT IS, SET DATA BUFFER
        MOVE.B  #$20,CR[A2]     ;IF NOT, RESET RX
        RTE                     ;RETURN
;
SETBUF: MOVEA.L A2,A5           ;DATA BUFFER ADDRESS
        SUBA.L  #$50000,A5      ;=CH. BASE - OFFSET
        RTE                     ;RETURN
;
DATA:   MOVE.B  RHR[A2],[A5]+;READ DATA CHAR.
        RTE                     ;RETURN
;
ERROR:  TRAP    #15
;
        END     START
```


TO_68K_BOARD
IRQ4/B27
IACK4/B17
IOSEL4/B9
RWN/A14
RESETN/C12
DTACKN/A16
A30
A29
A28
A27
A26
A25
A24
C30
C29
C28
C27
C26
C25
C24
C23
C22
C21
C20
C19
C18
C17
C16
C16
ADDR(23:1)
A1
A2
A3
A4
A5
A6
A7
A8
C1
C2
C3
C4
C5
C6
C7
C8
C31
+12V
A31
-12V
A32
B32
C32
+5V
A9
A11
A15
A17
A19
B20
B23
C9
SYSCLK/A10
BG/B4
BGACKN/B1
IRQ5/B26
IACK5/B16
IOSEL6/B8
VPA/C13
ASN/A18
UDSN/A12
LDSN/A13
74LS04
74LS32
74LS32
74LS04
74LS245
DIR
_G_EN
A8
A7
A6
A5
A4
A3
A2
A1
B8
B7
B6
B5
B4
B3
B2
B1
74LS38
74LS04
74LS164
A
B
CLK
CLR
QA
QB
QC
QD
QE
QF
QG
QH
VCC
5 45
20 65
2698
18/CEN
22/RDN
19/WRN
15/RESET
32/A5
31/A4
29/A3
27/A2
25/A1
23/A0
8/D0
9/D1
10/D2
11/D3
12/D4
13/D5
16/D6
17/D7
35/INTRAN
36/INTRBN
46/INTRCN
47/INTRDN
72/MPOa
43/MPOb
71/MPOc
51/MPOd
69/MPOe
53/MPOf
67/MPOg
64/MPOh
X1
X2
N/C
470
+5V
74LS38
3.6864MHZ
TXDa/1
TXDb/41
TXDc/81
TXDd/49
TXDe/74
TXDf/52
TXDg/73
TXDh/55
RXDa/3
RXDb/56
RXDc/83
RXDd/57
RXDe/79
RXDf/58
RXDg/75
RXDh/59
MPI0a/33
MPI0b/34
MPI0c/37
MPI0d/39
MPI0e/61
MPI0f/63
MPI0g/76
MPI0h/77
MPI1a/14
MPI1b/21
MPI1c/38
MPI1d/40
MPI1e/60
MPI1f/62
MPI1g/78
MPI1h/80
MPI2a/24
MPI2b/26
MPI2c/42
MPI2d/44
MPI2e/64
MPI2f/66
MPI2g/82
MPI2h/84
MPI3a/28
MPI3b/30
MPI3c/48
MPI3d/50
MPI3e/68
MPI3f/70
MPI3g/2
MPI3h/4
TXD
NE5170
3/AI
4/BI
7/CI
8/DI
22/EI
23/FI
26/GI
27/HI
A0/2
B0/5
C0/6
D0/9
E0/21
F0/24
G0/25
H0/28
a
b
c
d
e
f
g
h
RXD
NE5180
A0/3
B0/6
C0/10
D0/13
E0/17
F0/20
G0/24
H0/27
2/A
5/B
9/C
12/D
16/E
19/F
23/G
26/H
CTS
NE5180
RTS
NE5170
DCE
TXD 3
RTS 5
RXD 2
CTS 4
DTE
TXD 2
RTS 4
RXD 3
CTS 5

Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Signetics makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS

Signetics' Products are not designed for use in life support appliances, devices, or systems where malfunction of a Signetics Product can reasonably be expected to result in a personal injury. Signetics' customers using or selling Signetics' Products for use in such applications do so at their own risk and agree to fully indemnify Signetics for any damages resulting in such improper use or sale.

Signetics
a subsidiary of U.S. Philips Corporation

Signetics Corporation
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088-3409
Telephone 408/991-2000

Signetics Microprocessor Division

Data Sheet Errata

SCC2698 February 8,1988

DIP Pinout Correction

There is an error on the first page of the data sheet (December 1986 and July 13, 1987) in the pin configuration diagram for the 64 pin DIP. Pin number 38 should be labeled MPOd and pin 40 should be labeled MPOf (the data sheets have these two pins transposed). The pin description section of the data sheet is correct.

Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustration purposes only.

SCC2698A vs. SCC2698B Comparison

An enhanced version of the SCC2698, to be marked SCC2698Bxyznn (where xyznn represent the various temperature range and package options), is currently in development to replace the "A" version. It will be possible to substitute the "B" version for the "A" version if care is taken in design. Engineering samples of the SCC2698B will be available in Q1 1988 with production in the 84 pin PLCC and 64 pin DIP to follow in Q2 1988. The SCC2698A version will be discontinued when customers have successfully converted to the "B" version.

The "B" version corrects two anomalies which may cause operational problems in some applications of the current silicon, increases the number of outputs, and speeds testing. The following is a description of the differences between the SCC2698A and SCC2698B:

1. A bug in the change of state detector logic which could cause the occasional miss of a change will be corrected. No hardware or software changes are required. The inputs are automatically sampled every 25 uS at the standard 3.6864 MHz crystal frequency. The miss will occur in the "A" version if the flag bits are being read while a new change is being loaded by the sampling circuit. This will be corrected on the SCC2698B by preventing new changes from being loaded while a read is occurring.

2. Several control registers have been changed from transparent latches to edge triggered registers to insure that transient conditions on the bus are no longer passed through the registers. The "A" version contained complex register update logic which could result in some application problems. This change has permitted the removal of the register update logic. Programs written for the "A" version will work with the "B" version.

   The SCC2698A's "Register Update Logic" attempts to prevent programming errors from affecting part operation. When a write to a control register is performed, it is delayed internally until the receiver(s) and transmitter(s) have completed their current operations. If the update would affect only one channel, then the delay will end when that channel's transmitter and receiver have each completed their current operation and gone idle. If the operation would affect both channels within a block (ie change the counter timer mode) then the delay will not end until both receivers and transmitters have completed their current operations. If a transmitter is being fed a series of bytes it could lock out the update for a long period of time. The registers controlled by the update logic were MR1, MR2, CSR, OPCR and ACR. The update logic automatically disables the receiver(s) at the end of the current byte. This update logic could cause the following application problems:

   a) One or both receivers within a block may become disabled (stop functioning), when the function of an I/O pin is changed, the mode of the timer counter is changed, or any changes are made to the configuration of either receiver or transmitter.

   b) Changes to the mode of the counter timer, the mode of the I/O pins, or the configuration of either transmitter or receiver may be delayed in taking effect from microseconds to hours later.

In removing this logic for the SCC2698B version, the MR1, MR2, CSR, OPCR and ACR registers have been made edge triggered. Previously these registers were transparent latches. This change insures that register bits that are written with their previous value will not have their functions affected by the write, since transient conditions on the bus are no longer passed through the registers. For previous products such as the SCN2681, we have recommended, in AN405, that botn the transmitter and receiver be disabled by software, while either is being reconfigured, to avoid problems caused by transient conditions on the bus. In the SCC2698B it will now only be necessary to disable the function being reconfigured. For example; the CSR registers each contain 4 bits to control the receive clock source and 4 more bits to control the transmit clock source. With the edge triggered registers in the SCC2698B only the transmitter will have to be disabled if only its clock source is being modified.

3. Sixteen pins that were input only on the SCC2698A, may now be programmed to operate as outputs on the SCC2698B by programming OPCR bit 7 to a one. The pins in question are MPI2a-h and MPI3a-h and are only available on the 84 pin PLCC package option. The new outputs provide transmitter ready (TXRDY) and receiver ready (RXRDY) interrupts for each of the channels while still maintaining a Request To Send output (RTSN). OPCR(7), which is not used in the 2698A, should be programmed to zero to maintain software compatibility. The enhanced pins will become inputs on reset maintaining compatibility with the "A" version.

When programmed as inputs, the MPI2 and MPI3 pins operate as previously to provide general purpose inputs (GPI), Transmit Clock Inputs (TXC), or Receive Clock Inputs (RXC). As outputs, MPI2 provides the transmit holding register empty signal (TXRDY) and MPI3 provides the receiver FIFO not empty / full signal (RXRDY/FFULL). These signals were previously only available on the MPO pin which was often needed for the request to send (RTSN) signal.

There are four OPCRs in the SCC2698 for blocks A through D. Setting bit 7 of a given OPCR will convert both of the MPI2 pins and both of the MPI3 pins into outputs with the functions described above.

With appropriate programming the following I/O functions can now be simultaneously provided for each channel:

Outputs	RXRDY	Interrupt
	TXRDY	Interrupt
	RTS	Modem control
Inputs	DSR	Modem control
	CTS	Modem control
	RCLK	Receiver clock

The MPI2 and MPI3 pins each have a P-channel pull up resistor to provide a logic 1 when they are left unconnected as inputs. The output driver is an open drain N-channel transistor.

In summary for the SCC2698B:

OPCR(7) = 0 MPI2x will be a GPI or TXC input and
MPI3x will be a GPI or RXC input

OPCR(7) = 1 MPI2x will be a TXRDY output and
MPI3x will be a RXRDY/FFULL output.

4. Pin 41 of the 64 pin DIL package was a no connect on the SCC2698A, but becomes a test input on the SCC2698B. When using the DIL package, neither CSR(7:4) nor CSR(3:0) should be programmed to Hex 'E' or 'F'. These values will cause the "A" version to attempt to use inputs that are not available in the 64 pin package option (MPI2 and MPI3 pins), and will cause the "B" version to use the test pin as an external baud rate clock input common to all channels. Voltages on pin 41 should not exceed the absolute maximum ratings. This pin may be tied high, tied low or left open.

For Further assistance contact in Signetics' Microprocessor Division:
Ron Mitchell - Data Communications Marketing (408) 991-3512 or
Debi Ibarra - Data Communications Applications (408) 991-3509.

Signetics Microprocessor Division

Advance Notice

SCC2698B Enhanced Octal UART

□ Features of the SCC2698B

□ Maintaining Compatibility with the SCC2698A

An enhanced version of the SCC2698, to be marked SCC2698Bxyznn is currently in development to replace the "A" version. The SCC2698A will be discontinued, so it is very important that current customers be made aware of the differences in order to ensure software compatibility. It will be possible to substitute the "B" version if care is taken in the software design. Production of the SCC2698B will start in Q2 1988. The SCC2698A version will be discontinued when customers have successfully converted to the "B" version.

The "B" version corrects two anomalies which may cause operational problems in some applications of the current silicon, increases the number of outputs, and speeds testing. The following is a brief summary of the changes:

1. A bug in the change of state detector logic which could cause the occasional miss of a change will be corrected. No hardware or software changes are required.

2. Several control registers have been changed from transparent latches to edge triggered registers to insure that transient conditions on the bus are no longer passed through the registers. The "A" version contained complex register update logic which could result in some application problems. This change has permitted the removal of the register update logic. Programs written for the "A" version will work with the "B" version.

3. Sixteen pins (MPI2a-h and MPI3a-h) that were formally input only, may now be programmed to operate as outputs by programming OPCR(7) to a one. The new outputs provide the transmitter ready (TXRDY) and receiver ready (RXRDY) interrupts for each of the channels while still maintaining a Request To Send output (RTSN). **OPCR(7), which is not used in the 2698A, should be programmed to zero to maintain software compatibility.** The enhanced pins will become inputs on reset maintaining compatibility with the "A" version.

4. Pin 41 of the 64-pin DIL package was a no connect on the SCC2698A, but becomes a test input on the SCC2698B. When using the DIL package, neither CSR(7:4) nor CSR(3:0) should be programmed to hex 'E' or 'F'. These values will cause the "A" version to attempt to use inputs that are not available in the 64-pin package option, and will cause the "B" version to use the test pin as an external clock source. Voltages on pin 41 should not exceed the absolute maximum ratings. This pin may be tied high, tied low or left open.

If required, a more detailed description of these changes may be obtained from Ron Mitchell in Signetics Data Communications Marketing at (408) 991-3512.

November 19, 1987