Philips Components–Signetics

Document No.	
ECN No.	
Date of Issue	November 12, 1990
Status	Preliminary Specification
Data Communication Products	

SC26C94/SC68C94

Quad universal asynchronous receiver/transmitter (QUART)

DESCRIPTION

The SC26C94/SC68C94 quad universal asynchronous receiver/transmitter (QUART) combines four enhanced Signetics industry-standard UARTs with an innovative interrupt scheme that can vastly minimize host processor overhead. It is implemented using Signetics' high-speed CMOS process that combines small die size and cost with low power consumption.

The operating speed of each receiver and transmitter can be selected independently at one of eighteen fixed baud rates, a 16X clock derived from a programmable counter/timer, or an external 1X or 16X clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the Octal UART particularly attractive for dual-speed channel applications such as clustered terminal systems.

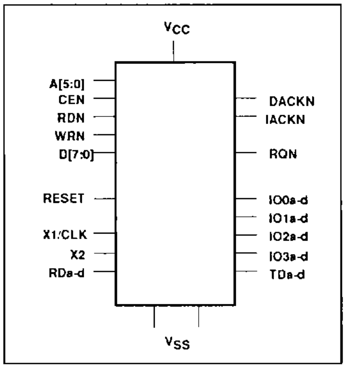
Each receiver is buffered with eight character FIFOs (first-in-first-out memories) and one shift register to minimize the potential for receiver overrun and to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking capability is provided to disable a remote UART transmitter when the receiver buffer is full. (RTS control)

The SC26C94/SC68C94 provides a power-down mode in which the oscillator is stopped and 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

- Four Signetics industry-standard UARTs
- Eight byte receive FIFOs for each UART
- 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 Non-standard rates to 230.4K baud
  - User-defined rates from the programmable counter/timer associated with each block
  - External 1X or 16X clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - Normal (full-duplex), automatic echo, local loop back, remote loopback
- Eight byte transmit FIFOs for each UART
- Programmable interrupt priorities
- Identification of highest priority interrupt
- Global interrupt register set provides data from interrupting channel
- Vectored interrupts with programmable vector format
- IACKN and DTACKN signals
- Built-in baud rate generator with choice of 18 rates

PIN CONFIGURATIONS



- Four I/O pins per UART for modem controls, clocks, etc.
- Power down mode
- High-speed CMOS technology
- 52-pin PLCC and 48-pin DIP
- Commercial and industrial temperature ranges available
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply
- Two multifunction programmable 16-bit counter/timers
- 1MHz 16X mode operation
- 45ns data bus release time
- "Watch Dog" timer for each receiver

Quad universal asynchronous receiver/transmitter (QUART)

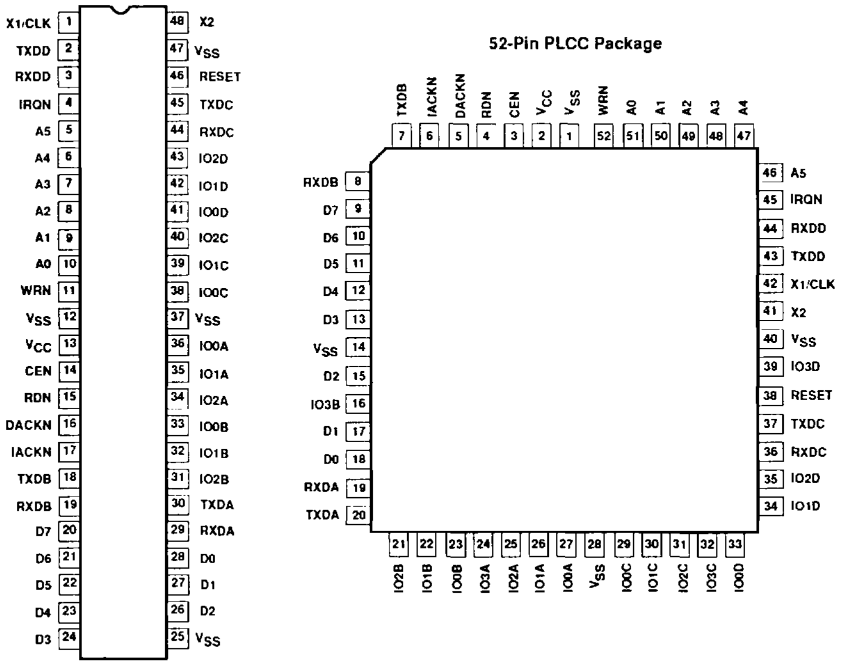
SC26C94/SC68C94

ORDERING INFORMATION

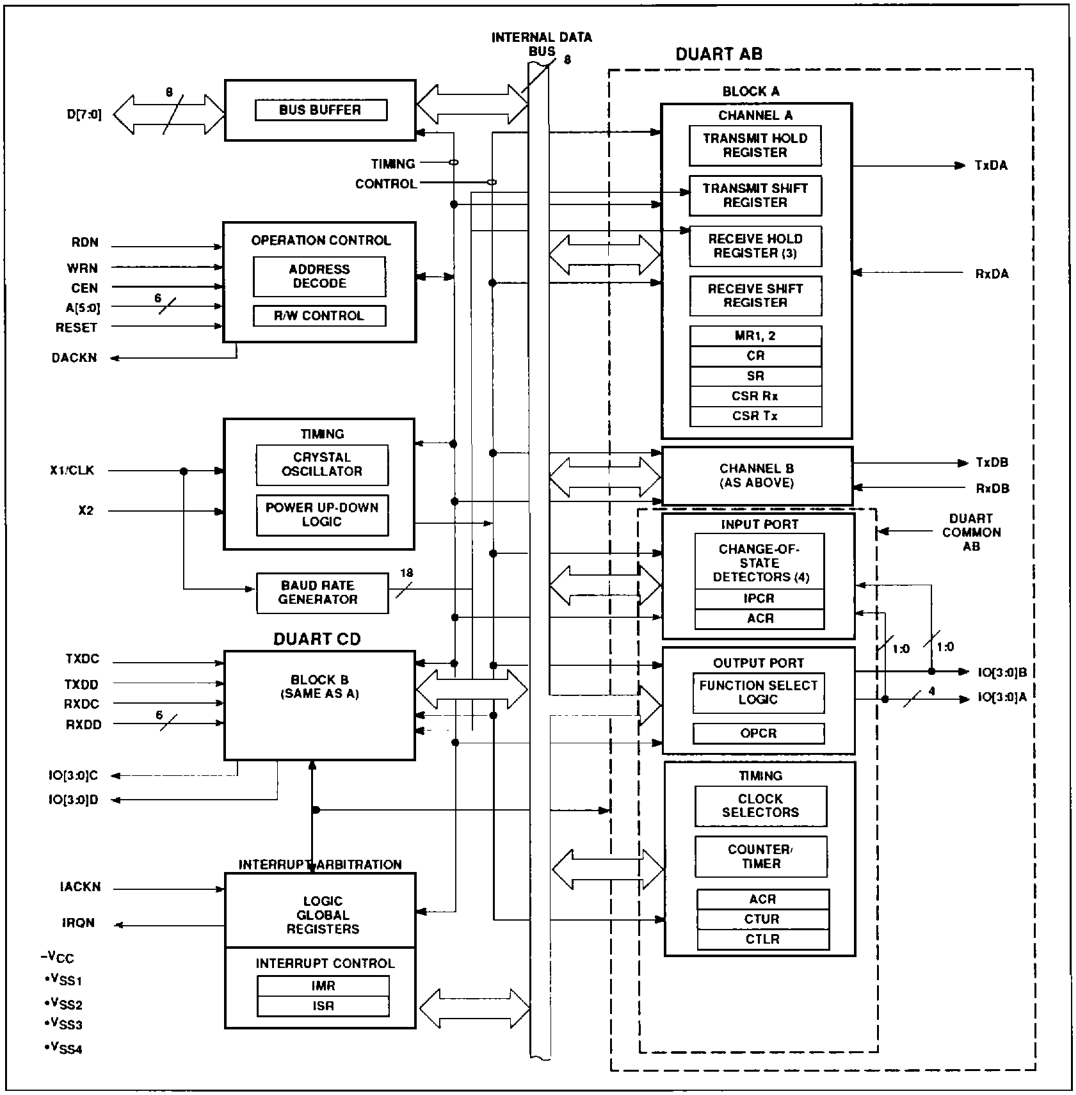
PACKAGES	V_{CC} = +5V ±10%, T_A = 0°C to +70°C	V_{CC} = +5V ±10%, T_A = –40°C to +85°C
Plastic Dual In-Line Package	SC26C94C1N48	SC26C94A1N48
Plastic Leaded Chip Carrier	SC26C94C1A52	SC26C94A1A52

NOTE:
Pin Grid Array (PGA) package version is available from Philips Components Military Division.

PIN CONFIGURATIONS



BLOCK DIAGRAM



Quad universal asynchronous receiver/transmitter (QUART) SC26C94/SC68C94

PIN DESCRIPTION

MNEMONIC	TYPE	NAME AND FUNCTION
CEN	I	**Chip Select:** Active low input that, in conjunction with RDN or WRN, indicates that the control processor is trying to access a QUART register.
A5:0	I	**Address Lines:** These inputs select a SC26C94/SC68C94 register to be read or written by the control processor.
D7:0	I/O	**8-bit Bidirectional Data Bus:** Used by the control processor to read and write SC26C94/SC68C94 registers.
RDN	I	**Read Strobe:** Active low input. When this line is asserted simultaneously with CEN, the SC26C94/SC68C94 places the contents of the register selected by A[5:0] on the D[7:0] lines.
WRN	I	**Write Strobe:** Active low input. When this line is asserted simultaneously with CEN, the SC26C94/SC68C94 writes the data on D[7:0] into the register selected by A[5:0].
DACKN	O	**Data ACKnowledge:** Active low, open-drain output to the control processor, which is asserted subsequent to a read or write operation. For a read operation, assertion of DACKN indicates that register data is valid on D[7:0]. For a write operation, it indicates that the data on D[7:0] has been captured into the indicated register. This signal corresponds to READYN on 80x86 processors and DTACKN on 680x0 processors.
IRQN	O	**Interrupt Request:** This active low open-drain output to the control processor indicating that one or more of the UART channels has reached an interrupt value which exceeds that pre-programmed by host software. The IRQN can be used directly as a 680x0 processor input, it must be inverted for use as an 80x86 interrupt input. This signal requires an external pull-up resistor.
IACKN	I	**Interrupt ACKnowledge:** Active low input indicating that the control processor is acknowledging an interrupt requested by this device. The SC26C94/SC68C94 responds to the assertion of this signal by placing an interrupt vector on D[7:0] and asserting DACKN.
TDa-d	O	**Transmit Data:** Serial outputs from the four UARTs
RDa-d	I	**Receive Data:** Serial inputs to the four UARTs
I/O0a-d	I/O	**Input/Output 0:** A multi-use input or output signal for each UART. These pins can be used as general purpose inputs, Clear to Send inputs, 1X or 16X Transmit Clock outputs or general purpose outputs. Change-of-state detection is provided for these pins.
I/O1a-d	I/O	**Input/Output 1:** A multi-use input or output signal for each UART. These pins can be used as general purpose or 1X or 16X transmit clock inputs, or general purpose 1X or 16X receive clock outputs. Change-of-state detection is provided for these pins. In addition, I/O1a and I/O1c can be used as Counter/Timer inputs and I/O1b and I/O1d can be used as Counter/Timer outputs.
I/O2a-d	I/O	**Input/Output 2:** A multi-use input or output signal for each UART. These pins can be used as general purpose inputs, 1X or 16X receive clock inputs, general purpose outputs, RTS output or 1X or 16X receive clock outputs.
I/O3a-d	I/O	**Input/Output 3:** A multi-use input or output signal for each UART. These pins can be used as general purpose inputs, 1X or 16X transmit clock inputs, general purpose outputs, or 1X or 16X transmit clock outputs.
RESET	I	**Master Reset:** Active high reset for the SC26C94/SC68C94 logic. Must be asserted at power-up, may be asserted at other times that the system is to be reset and restarted. Registers reset. MR pointer, CIR, IRQN, DTACKN, IVR Interrupt Vector, Power Down, Test registers, FIFO pointers, Baud rate generator, Error Status, Watch Dog Timers, IMR, Change of State detectors, counter/timer to timer, Transmitter and Receiver controllers.
X1/CLK	I	**Crystal 1 or Communication Clock:** This pin can be connected to one side of a 3.6864MHz or a 7.3728MHz crystal, or can be connected to a TTL-level clock with the same frequency.
X2	O	**Crystal 2:** If a crystal is used, this pin should be connected to its other terminal. If a TTL-level clock is applied to X1, this pin should be left unconnected.

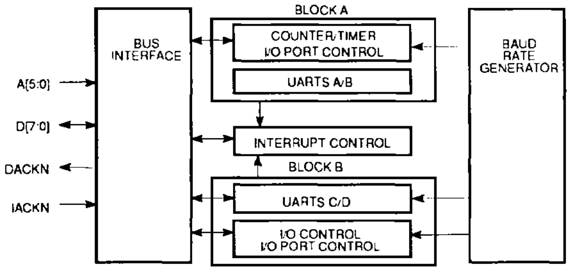


Figure 1. Channel Architecture

Table 1. QUART Registers

A5:0	READ (RDN = Low)	WRITE (WRN = Low)
000000	Mode Register a (MR0a, MR1a, MR2a)	Mode Register a (MR0a, MR1a, MR2a)
000001	Status Register a (SRa)	Clock Select Register a (CSRa)
000010	Reserved	Command Register a (CRa)
000100	Receive Holding Register a (RxFIFOa)	Transmit Holding Register a (TxFIFOa)
001000	Input Port Change Reg ab (IPCRab)	Auxiliary Control Reg ab (ACRab)
000101	Interrupt Status Reg ab (ISRab)	Interrupt Mask Reg ab (IMRab)
000110	Counter/Timer Upper ab (CTUab)	Counter/Timer Upper Reg ab (CTURab)
000111	Counter/Timer Lower ab (CTLab)	Counter/Timer Lower Reg ab (CTLRab)
001000	Mode Register b (MR0b, MR1b, MR2b)	Mode Register b (MR0b, MR1b, MR2b)
001001	Status Register b (SR1b, SR2b)	Clock Select Register b (CSRb, CSRb)
001010	Reserved	Command Register b (CRb)
001011	Receive Holding Register b (RxFIFOb)	Transmit Holding Register b (TxFIFOb)
001100	Output Port Register ab (OPRab)	Output Port Register ab (OPRab)
001101	Input Port Register ab (IPRab)	I/O Port Register a (IOPRa)
001110	Start Counter ab	I/O Port Register b (IOPRb)
001111	Stop Counter ab	Reserved
010000	Mode Register c (MR0c, MR1c, MR2c)	Mode Register c (MR0c, MR1c, MR2c)
010001	Status Register c (SRc)	Clock Select Register c (CSRc)
010010	Reserved	Command Register c (CRc)
010011	Receive Holding Register c (RxFIFOc)	Transmit Holding Register c (TxFIFOc)
010100	Input Port Change Reg cd (IPCRcd)	Auxiliary Control Reg cd (ACRcd)
010101	Interrupt Status Reg cd (ISRcd)	Interrupt Mask Reg cd (IMRcd)
010110	Counter/Timer Upper cd (CTUcd)	Counter/Timer Upper Reg cd (CTURcd)
010111	Counter/Timer Lower cd (CTLcd)	Counter/Timer Lower Reg cd (CTLRcd)
011000	Mode Register d (MR0d, MR1d, MR2d)	Mode Register d (MR0d, MR1d, MR2d)
011001	Status Register d (SRd)	Clock Select Register d (CSRd)
011010	Reserved	Command Register d (CRd)
011011	Receive Holding Register d (RxFIFOd)	Transmit Holding Register d (TxFIFOd)
011100	Output Port Register cd (OPRcd)	Output Port Register cd (OPRcd)
011101	Input Port Register cd (IPRcd)	I/O Port Register c (IOPRc)
011110	Start Counter cd	I/O Port Register d (IOPRd)
011111	Stop Counter cd	Reserved
100000	Bidding Control Register a (BCRa)	Bidding Control Register a (BCRa)
100001	Bidding Control Register b (BCRb)	Bidding Control Register b (BCRb)
100010	Bidding Control Register c (BCRc)	Bidding Control Register c (BCRc)
100011	Bidding Control Register d (BCRd)	Bidding Control Register d (BCRd)
100100	Reserved	Power Down
100101	Reserved	Power Up
100110	Reserved	Disable DACKN
100111	Reserved	Enable DACKN
101000	Current Interrupt Register (CIR)	Reserved
101001	Global Interrupt Channel Reg (GICR)	Interrupt Vector Register (IVR)
101010	Global Int Byte Count Reg (GIBCR)	Update CIR
101011	Global Receive Holding Reg (GRxFIFO)	Global Transmit Holding Reg (GTxFIFO)
101100	Interrupt Control Register (ICR)	Interrupt Control Register (ICR)
101101	Reserved	Test 3
101101	Reserved	Set X1/CLK Normal
101111	Reserved	Set X1/CLK to divide by two
110000–111000	Reserved	Reserved
111001	Test Mode	Test Mode
111010–111111	Reserved	Reserved

FUNCTIONAL BLOCKS

The QUART is composed of four Signetics industry-standard UARTs, each having a separate transmit and receive channel.

The Basic UART cells in the QUART are configured with 8-byte Receive FIFOs and 8-byte Transmit FIFOs. Hardware supports interrupt priority arbitration based on the number of bytes available in the transmit and receive FIFOs. Attempts to push a full FIFO or pop an empty FIFO do not affect the count.

Baud Rate Generator

The baud rate generator used in the QUART is the same as that used in other Signetics industry standard UARTs. It provides 18 basic Baud rates from 50 baud to 38,400 baud. It has been enhanced to provide to provide other baud rates up to 230,400 baud based on a 3.6364MHz clock. With a 7.272800MHz clock 460,800 baud is available.

BLOCK DIAGRAM

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

Channel Blocks

There are two 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 QUART.

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 control processor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Mode registers (MR) 0, 1 and 2 are accessed via an address counter. This counter is set to one (1) by reset for compatibility with other Signetics UARTs. It is set to 0 via a command to the Command Register (CR) The address counter is incremented with each access to the MR until it reaches 2 at which time it remains at 2. All subsequent accesses to the MR will be to MR2 until the MR counter changed by a reset or an MR counter command.

The Mode Registers control the basic configuration of the UART channels. There is one for each UART. (Transmitter/receiver pair)

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.

Oscillator

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, X1 must be driven and X2 left floating as shown in Figure 8. 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

Baud Rate Generator

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 16X the actual baud rate The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection, by the receiver and transmitter, of any of these baud rates or an external timing signal.

Counter/Timer (C/T)

The counter timer is a 16-bit programmable divider that operates in one of three modes. counter, timer, time out. In the timer mode it generates a square wave. In the counter mode it generates a time delay. In the time out mode it monitors the time between received characters. In this mode it is acting as a programmable watch dog timer.

The C/T uses the numbers loaded into the Counter/Timer Lower Register (CTLR) and the Counter/Timer Upper Register (CTUR) as its divisor.

There are two counter/timers in the QUART; one for each DUART. The counter/timer clock source and mode of operation (counter or timer) is selected by the Auxiliary Control Register bits 6 to 4 (ACR[6:4]). The output of the counter/timer may be used for a baud rate and/or may be output to the I/O pins for some external function that may be totally unrelated to data transmission. The counter/timer also sets the counter/timer ready bit in the Interrupt Status Register (ISR) when its output transitions from 1 to 0.

A register read address is reserved to start counter/timer command and a second register read address is reserved to issue a stop command. The START command always loads the contents of CTUR, CTLR, to the counting registers. The STOP command always resets the ISR[3] bit in the interrupt status registers. See Table 1.

Timer Mode

In the timer mode a symmetrical square wave is generated whose half period is equal in time to division of the selected counter/timer clock frequency by the 16-bit number loaded in the CTLR CTUR. Thus, the frequency of the counter/timer output will be equal to the counter/timer clock frequency divided by twice the value of the CTUR CTLR. While in the timer mode the ISR bit 3 (ISR[3]) will be set each time the counter/timer transitions from 1 to 0. (High to low) This continues regardless of issuance of the stop counter command. ISR[3] is reset by the stop counter command. NOTE: Reading of the CTU and CTL registers in the timer mode is not meaningful

Counter Mode

In the counter mode the counter/timer counts the value of the CTLR CTUR down to zero and then sets the ISR[3] bit and sets the counter/timer output from 1 to 0. It then rolls over to 65,365 and continues counting with no further observable effect.

Reading the C/T in the counter mode outputs the present state of the C/T. If the C/T is not stopped, a read of the C/T may result in changing data on the data bus.

Timeout Mode

The timeout mode uses the received data stream to control the counter. Each time a received character is transferred from the shift register to the RxFIFO, the counter is restarted. If a new character is not received before the counter reaches zero count, the counter ready bit is set, and an interrupt can be generated This mode can be used to indicate when data has been left in the Rx FIFO for more than the programmed time limit. If the receiver has been programmed to interrupt the CPU when the receive FIFO is full, and the message ends before the FIFO is full, the CPU will not be interrupted for the remaining characters in the RxFIFO.

By programming the C/T such that it would time out in just over one character time, the above situation could be avoided. The processor would be interrupted any time the data stream had stopped for more than one character time. NOTE: This is very similar to the watch dog timer controlled by MR0. The difference is in the programmability of the delay time.

This mode is enabled by writing the appropriate command to the command register. Writing an 'Ax' to CRA or CRB will invoke the timeout mode for that channel. Writing a 'Cx' to CRA or CRB will disable the timeout mode. The timeout mode should only be used by one channel at a time.

The timeout mode disables the regular START/STOP counter commands and puts the C/T into counter mode under the control of the received data stream. Each time a received character is transferred from the shift register to the RxFIFO, the C/T is stopped after one C/T clock, reloaded with the value in CTUR and CTLR and then restarted on the next C/T clock. If the C/T is allowed to end the count before a new character has been received, the counter ready bit, ISR[3], will be set. If IMR[3] is set, this will generate an interrupt. Since receiving a character restarts the C/T, the receipt of a character after the C/T has timed out will clear the counter ready bit, ISR[3], and the interrupt. Invoking the 'Set Timeout Mode On' command, CRx='Ax', will also clear the counter ready bit and stop the counter until the next character is received.

The counter timer is controlled with six commands: Start/Stop C/T, Read/Write Counter/Timer lower register and Read/Write Counter/Timer upper register. These commands have slight differences depending on the mode of operation. Please see the detail of the commands under the CTLR CTUR Register descriptions.

Receiver and Transmitter

The QUART has four 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 (MR0, MR1 and MR2) Clock Select Register (CSR), Command Register (CR), Status Register (SR), Transmit FIFO (TxFIFO), and the Receive FIFO (RxFIFO). The transmit and receive FIFOs are each eight characters deep. The receive FIFO also stores three status bits with each character.

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 transmission of the stop bits, if a new *character is not available in the* TxFIFO, 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 in the TxFIFO. In the 16X clock mode, this also re-synchronizes the internal 1X 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 characters currently being transmitted and the character in the TxFIFO, if any, are completely sent out. Characters cannot be loaded in the TxFIFO 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 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode).

If RxD is sampled High, the start bit is invalid and the search for a valid start bit begins again. If RxD is still Low, a valid start bit is assumed and the receiver 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 data is then transferred to the RxFIFO 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 RxFIFO 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 in 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 1X clock (internal or* external) before a search for the next start bit begins.

RECEIVER FIFO

The RxFIFO consists of a first-in-first-out (FIFO) with a capacity of eight 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 eight stack positions are filled with data. The number of filled positions is encoded into a 3-bit value. This value is sent to the interrupt bidding logic where it is used to generate an interrupt. A read of the RxFIFO, 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.

NOTE: The number of filled positions in the RxFIFO is coded as one (1) less than the actual number. Thus, three filled positions is coded as two, 8 filled is coded as 7, etc.

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 RxFIFO 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 upon receipt of the start bit of the new (overrunning) character.

A "watch dog" timer is associated with each receiver. Its interrupt is enabled by MR0[7]. The purpose of this timer to alert the control processor that characters are in the RxFIFO which have not been read. This situation may occur at the end of a transmission when the last few characters received are not sufficient to cause an interrupt.

This counter times out after 64 bit times. It is reset each time a character is transferred from the Receive shift register to the RxFIFO.

WAKE-UP MODE

In addition to the normal transmitter and receiver operation described above, the QUART incorporates a special mode which provides automatic "wakeup" of a receiver through address frame (or character) recognition for multi-processor or multi-station communications. This mode is selected by programming MR1[4:3] to '11'.

In this mode of operation a 'master' station transmits an address character to the several 'slave' stations on the line. The address character is identified by setting its parity bit to 1. The slave stations will usually have their receivers partially enabled as a result of setting MR1[4:3] to 11. When the receivers see a parity bit set to one they will load that character to the RxFIFO and set the RxRDY bit in the status register. The user would usually set the receiver interrupt to occur on RxRDY as well. (All characters whose parity bits are set to 0 will be ignored) The local processor at the slave station will read the 'address' character just received. The master will normally follow an address character(s) with data characters. Since the data characters transmitted by the master will have their parity bits set to zero, stations other than the addressed one(s) will ignore the data.

A transmitted character consists of a start bit, the programmed number of data and stop bits and an "address/data" bit. The parity bit is used as the address or data indicator. The polarity of the A/D bit is selected by setting MR1[2] to zero or one; zero indicates that the current byte is data, while one indicates that the current byte is addresses. The desired polarity of the A/D bit (parity) should be programmed before the TxFIFO is loaded.

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 in the RxFIFO 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 then transferred to the CPU via the RxFIFO. In either case, the data bits are loaded in the data FIFO while the A/D bit is loaded in 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.

INPUT OUTPUT (I/O) PINS

There are 16 multi-use pins; four for each UART. These pins are accessed and controlled via the Input Port Register (IPR), I/O Port Control Register (IOPCR), Input Port Change Register (IPCR), and Output Port Register (OPR). They may be individually programmed to be inputs or outputs.

I/O0x and I/O1x pins have change of state detectors The change of state detectors sample the input ports every 26.04µs and set the change bit in the IPCR if the pin has changed since it was last read. Whether the pins are programmed as inputs or outputs the change detectors still operation and report changes accordingly. See the register descriptions of the I/O ports for the detailed use of these features.

Interrupt Priority Logic

The interrupt logic compares all active interrupts in the QUART, periodically selecting the highest priority interrupt for comparison to an Interrupt Threshold value User programmable register fields allow the system programmer to tailor which interrupt conditions are more important than others. Programmable interrupt priorities allow timely response to critical interrupts or simply allow a single CPU to service a greater number of interrupt situations than it could using more conventional methods.

Overview

The interrupt logic produces a numeric code that identifies the highest priority interrupt condition currently pending. This code is compared against a programmable Interrupt Threshold register which determines whether the IRQN pin is asserted. If the code is currently greater than the programmed value, IRQN is asserted. In the QUART there are 18 interrupt sources:

1. Four receiver data transfer/space filled functions
2. Four receiver break detected conditions
3. Four transmitter FIFO space available events
4. Two counter timer interrupts
5. Four change of state detectors

Each interrupt source is enabled or disabled by the appropriate Interrupt Mask Register (IMR) bit as Signetics UARTs have always been (the receiver timeout function is enabled by a bit in the new MR0 register). Those interrupt sources that are enabled provide the interrupt priority logic with an 8 bit value, considered as an unsigned integer for purposes of this discussion. Those interrupt sources that are disabled do not provide a value to the arbitration logic. This can be accomplished by forcing an all zero value of by inhibiting the EVAL signal, described below, in the disabled interrupt sources.

This integer (called a "bid" since each interrupt is vying for attention) is a concatenation of fixed fields and user programmable fields. The fixed fields are channel number, interrupt type and (for receiver and transmitter types) byte count values. During the "bid arbitration" process, all the bids from enabled sources are presented, simultaneously, to an internal Interrupt Bus. The bidding system and formats are discussed in more detail in the following sections.

The interrupt arbitration logic insures that the interrupt with the numerically largest "bid" value will be the only source driving the Interrupt Bus at the end of the arbitration period. The winner must continue to drive the Interrupt Bus long enough to insure capture by the Interrupt Bus Latches. At the beginning of the next arbitration cycle, all "enabled bidders" drive their current values onto the Interrupt Bus.

The value of the winning bid is compared to the Interrupt Threshold field of the Interrupt Control register which determines if an interrupt should be generated by asserting IRQN Winning bids with values below the interrupt threshold do not generate an interrupt.

Priority Arbitration and Bidding

Each of the five "types" of interrupts has slightly different "bid" value, as follows:

Receivers

# rcv'd	rEr	1	1	Chan #
3	1	1	1	2

Transmitters

0	# avail	1	0	Chan #
1	3	1	1	2

Break Detect

Programmable	1	0	0	Chan #
3	1	1	1	2

Change of State

Programmable	0	0	1	Chan #
3	1	1	1	2

Counter/Timer

Programmable	0	1	0	1	Chan #
2	1	1	1	1	2

Bits shown above as '0' or '1' are hard-wired inputs to the arbitration logic. Their presence allows determination of the interrupt type and they insure that no bid will have a value of all zeros (a condition that is indistinguishable from not bidding at all). They also serve to set a default priority among the non-receive/transmit types when the programmable fields are all zeros.

The channel number always occupies the two LSBs. Inclusion of the channel number insures that a bid value generated will be unique and that a single "winner" will drive the Interrupt Bus at the end of the arbitration interval. The channel number portion of each UARTs bid is hard-wired with UARTa being channel number 0 and so forth.

As can be seen above, bits 4:2 of the winning bid value can be used to identify the type of interrupt, including whether data was received correctly or not. Like the Channel number field, these bits are hard-wired for each interrupt source.

The "# rcv'd" and "# avail" fields indicate the number of bytes present in the receiver FIFO and the number of empty bytes in the transmitter FIFO, respectively. For both these fields, the count is one less than the actual number of bytes available.

NOTE: When there are zero bytes in the receiver's FIFO, it does NOT bid. Similarly, a full transmitter FIFO makes NO bid. In the case where all bids have been disabled by the Interrupt Mask Register or as a result of their byte counts, the active-low Interrupt Bus will return FFh. This value always indicates no interrupt source is active and IRQN should be negated.

The high order bit of the transmitter "bid" is always zero. An empty transmit FIFO is, therefore, fixed at a lower interrupt priority than a 5/8 full receive FIFO. Bit 4 of a receiver bid is the Receiver Error Bit (RER). The RER is the OR of the parity, framing and overrun error conditions. The RER does little to modify the priority of receiver interrupts vs. transmitter interrupts. It is output to the Interrupt Bus to allow inclusion of good data vs. problem data information in the Current Interrupt Register.

The high order bits of bids for received break, CoS (Change of State) and Counter/Timer events are all programmable. By programming ones in these fields, the associated interrupt source can be made more significant than most receiver and all transmitter interrupts. Values near zero in these fields makes them lower priority classes of interrupt.

As shown in Figure 2, the bid arbitration process is controlled by the EVAL/HOLDN signal derived from the oscillator clock.

Receipt of an IACKN signal from the control processor latches the latest "winning bid" from the latched Interrupt Bus into the Current Interrupt Register (CIR). This logic is diagrammed in Figure 3.

If the IACKN falling edge of Figure 1 occurs during EVAL time, the result from the last arbitration (captured by the Interrupt Bus latches) is stored in CIR. Otherwise, the next EVAL pulse is inhibited and the value in the Interrupt Bus Latches is stored in CIR.

Clearing the Interrupt

Activities which change the state of the ISR will cause the IRQN to assert or negate. In addition, the accessing of a global or local RxFIFO or TxFIFO reduces the associated byte count for transmitter and receiver data interrupts. If the byte count falls below the threshold value, the interrupt request is withdrawn. Other interrupt conditions are cleared as on previous Signetics UARTs.

Once the interrupts is cleared or its byte count value is reduced by one of the methods listed above, a different bidder (or no bidder at all) will win the on-going arbitration. When the winning bid drops below the Interrupt Threshold Register's value, the IRQN pin will negate.

Interrupt Priority Arbitration Hardware

The hardware that resolves which interrupt has the highest priority is shown conceptually in Figure 3. The rising edge of the EVAL signal begins an arbitration. All interrupt sources drive their "bid" onto the active-low Interrupt Bus. An open drain buffer is employed at each bit position. Interrupt sources that are not participating in the arbitration, i.e., those that are disabled, do not assert their Interrupt Bus drivers.

Each interrupt source in the QUART has its own arbitration logic like that shown in Figure 3. Note that a one in a bit of a bid value corresponds to a low on the Interrupt Bus line to low (1), the result is a low (1) on that Interrupt Bus bit regardless of what the other interrupt sources are doing with respect to that line. Thus each line acts as a wired OR.

At each bit position, the arbitration logic compares the value of that bit of its bid value against the OR'd result on that bit of the Interrupt Bus. If it did not drive the line to low, but the result on the interrupt bus line was low, the logic disables less significant bits, if any, from driving the interrupt bus. In effect, the logic of all the active interrupt sources acts together as a combinatorial network to determine the highest priority active interrupt. This network is best understood as acting from the most significant bit of the interrupt bus, in sequence down to the least significant bit. After a suitable settling time, the value on the interrupt bus reflects the numerically highest bid value among the active interrupt sources. However, while the overall logic network is settling, the logic at what what proves to be the final winning interrupt source may find itself temporarily losing at some of the less significant bits.

The winning value is captured on the trailing edge of EVAL; the pulse with of EVAL must be long enough for a worst-case combination of bid values to resolve.

Arbitration - Aftermath

At the end of the arbitration, i.e., the falling edge of EVAL, the winning interrupt source is driving its Channel number, number of bytes (if applicable) and interrupt type onto the Interrupt Bus. These values are captured into a latch by the trailing edge of EVAL. The output of this latch is used by the Interrupt Threshold comparator; the winning value is captured into another set of latches called the Current Interrupt Register (CIR) at the time of an Interrupt Acknowledge cycle.

The Current Interrupt Register and associated read logic is shown in Figure 3. Interrupting channel number and the three bit interrupt type code are readable via the Internal Data Bus.

The contents of the appropriate receiver or transmitter byte "counter", as captured at the time of IACKN assertion, make up bits 7:5 of the CIR. If the interrupt type stored in the Current Interrupt Register is not a receiver or transmitter data transfer type, the CIR[7:5] field is read as the programmable fields of their respective bid formats.

The buffers driving the CIR to the DBUS also provide the means of implementing the Global Interrupting Channel and Global Byte Count Registers, described in a later section.

The winning bid channel number and interrupt type fields can also be used to generate part of the Interrupt Acknowledge (IACK) Vector, as defined by the Interrupt Control Register.

Interrupt Context

The channel number of the winning "bid" is used by the address decoders to provide data from the interrupting UART channel via a set of Global pseudo-registers. The interrupt Global pseudo-registers are:

1. Global Interrupting Byte Count
2. Global Interrupting Channel
3. Global Receive Holding Register
4. Global Transmit Holding Register

The first two Global "registers" are provided by Current Interrupt Register fields as shown in Figure 3. The interrupting channel number latched in CIR modifies address decoding so that the Receive or Transmit Holding Register for the interrupting channel is accessed during I/O involving the Global Receive and Transmit Holding Registers. Similarly, for data interrupts from the transmitter and receiver, the number of characters available for transfer to the CPU or the number of transmit FIFO positions open is available by reading the Global Interrupt Byte Count Register. For non-data interrupts, a read of the Global Interrupt Byte Count Register yields an undefined value.

In effect, once latched by an IACK, the winning interrupt channel number determines the contents of the global registers. All Global registers will provide data from the interrupting UART channel.

Interrupt Threshold Calculation

The state of IRQN is determined by comparison of the winning "bid" value to the Interrupt Threshold field of the Interrupt Control Register.

The logic of the bidding circuit is such that when no interrupt source has a value greater than the interrupt threshold then the interrupt is not asserted and the CIR (Current Interrupt Register) is set to all ones. When one or more of the 18 interrupt sources which are enabled via the IMR (Interrupt Mask Register) exceed the threshold then the interrupt threshold is effectively disconnected from the bidding operation while the 18 sources now bid against each other. The final result is that the highest bidding source will disable all others and its value will be loaded to the CIR and the IRQN pin asserted low. This all occurs during each cycle of the X1,X2 crystal clock.

Interrupt Note on 26C94.
For the receivers and transmitters, the bidding of any particular unit may be held off unless one of four FIFO fill levels is attained. This is done by setting the RxINT and TxINT bits in MR0 and MR1 to non-zero values. This may be used to prevent a receiver or transmitter from generating an interrupt eventhough it is filled above the bid threshold. Although this is not in agreement with the idea that each enabled interrupt source bid withe qual authority, it does allow the flexibility of giving particular receiver or transmitters more interrupt importance than others.

Receiver interrupt fill level.

MR0[6]	MR1[6]	Interrupt Condition
0	0	1 or more bytes in FIFO (Rx RDY) default
0	1	3 or more bytes in FIFO
1	0	6 or more bytes in FIFO
1	1	8 or more bytes in FIFO (Rx FULL)

MR0[5:4] – Tx interrupt fill level

MR0[5]	MR0[4]	Interrupt Condition
0	0	8 bytes empty (Tx EMPTY) default
0	1	4 or more bytes empty
1	0	6 or more bytes empty
1	1	1 or more bytes empty (Tx RDY)

Vectored Interrupts

The QUART responds to an Interrupt Acknowledge (IACK) initiated by the host MCU by providing an Interrupt Acknowledge Vector on D7:0 The interrupt acknowledge cycle is terminated with a DACKN pulse. The vector provided by the QUART can have one of the three forms under control of the IVC control field (bits 1:0 of the Interrupt Control Register):

With IVC = 00 (IVR only)

IVR7:0
8

With IVC = 01 (channel number)

IVR7:2	Chan #
6	2

With IVC = 10 (type & channel number)

IVR7:5	Type	Chan #
3	3	2

A code of 11 in the Interrupt Vector Control Field of the ICR results in NO interrupt vector being generated. The external data bus will be held in a high impedance state throughout the IACK cycle. A DACKN will be generated normally for the IACK cycle, however.

REGISTERS

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

The bit formats of the QUART registers are depicted in Table 2.

MR0 – Mode Register 0

Mode Register 0 (MR0) is part of the UART configuration registers. It controls the watch dog timer and the encoding of the number of characters received in the RxFIFO. The lower four bits of this register are not implemented in the hardware of the chip. MR0 should be set to only 80h or 00h. A read of this register will return 1111 (Fh) in the lower four bits. See note in Interrupt Threshold Calculation

The MR0 register is accessed by setting the MR Pointer to zero (0) via the command register command 1011 (Bh).

MR0[7]: This bit enables or disables the RxFIFO watch dog timer.
MR0[7] = 1 enable timer
MR0[7] = 0 disable timer
MR0[6 4]: These bits should always be set to zero.
MR0[3:0]: These bits are not implemented in the chip. They could be used to control hardware external to the chip.

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 the 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 re-asserted 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 1
This bit is reserved and should be set to 0. See note in Interrupt Threshold Calculation.

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 transmitted 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 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 QUART 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 re-clocked 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.

Table 2. Register Bit Formats

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

MR0 (Mode Register 0)

Rx Watch-dog Timer	RxINT2 bit	TxINT Control		These bits not implemented. May be used for external control (Reserved for future upgrades)			
0 = off 1 = on	These bits should be set to 0			x	x	x	x

MR1 (Mode Register 1)

RxRTS Control	RxINT1 Select	Error Mode	Parity Mode		Parity Type	Bits per Character	
0 = No 1 = Yes	Always set to 0	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	

MR2 (Mode Register 2)

Channel Mode		TxRTS Conrol	CTS Enable Tx	Stop Bit Length*			
00 = Normal 01 = Auto-echo 10 = Local loop 11 = Remote loop		0 = No 1 = Yes	0 = No 1 = Yes	0 = 0.563 4 = 0.813 8 = 1.563 C = 1.813 1 = 0.625 5 = 0.875 9 = 1.625 D = 1.875 2 = 0.688 6 = 0.938 A = 1.688 E = 1.938 3 = 0.750 7 = 1.000 B = 1.750 F = 2.000			

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

CSR (Clock Select Register)

Receiver Clock Select				Transmitter Clock Select			
See text				See text			

CR (Command Register)

Miscellaneous Commands				Disable Tx	Enable Tx	Disable Rx	Enable Rx
See text				0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes

SR (Status Register)

Rec'd. Break	Framing Error	Parity Error	Overrun Error	TxEMT	TxRDY	FFULL	RxRDY
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.

Table 2. Register Bit Formats (Continued)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

ACR (Auxiliary Control Register)

BRG Set Select	Counter/Timer Mode and Source			Delta IO1b	Delta IO0b	Delta IO1a	Delta IO0a
0 = set 1 1 = set 2	See text			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

IPCR (Input Port Change Register)

Delta IO1b	Delta IO0b	Delta IO1a	Delta IO0a	IO1b	IO0b	IO1a	IO0a
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

ISR (Interrupt Status Register)

IO 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

IMR (Interrupt Mask Register)

IO Port Change INT	Delta BREAKb INT	RxRDY/ FFULLb INT	TxRDYb INT	Counter Ready INT	Delta BREAKa INT	RxRDY/ FFULLa INT	TxRDYa INT
0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

CTUR (Counter/Timer Upper Register)

C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

CTLR (Counter/Timer Lower Register)

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

IPR (Input Port Register)

IO3b	IO2b	IO3a	IO2a	IO1b	IO0b	IO1a	IO0a
0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High

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 re-clocked 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 all 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 TxFIFO (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 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 of the message is loaded in the TxFIFO.
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 sate of this bit determines if the CTSN input (IO) 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 sate 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 1X clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

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 16X clock, except for CSR[7:4] = 1111. IO2x is external input.

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	IO3x – 16X	IO3x – 16X
1 1 1 1	IO3x – 1X	IO3x – 1X

Table 3. Baud Rate

CSR[7:4]	Test 3 = 0		Test 3 = 1	
	ACR[7] = 0	ACR[7] = 1	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75	50	450
0 0 0 1	110	110	110	110
0 0 1 0	134.5	38.4K	134.5	230.4K
0 0 1 1	200	150	200	900
0 1 0 0	300	300	1800	1,800
0 1 0 1	600	600	3,600	3,600
0 1 1 0	1,200	1,200	7,200	7,200
0 1 1 1	1,050	2,000	1,050	2,000
1 0 0 0	2,400	2,400	14.4K	14.4K
1 0 0 1	4,800	4,800	28.8K	28.8K
1 0 1 0	7,200	1,800	7,200	1,800
1 0 1 1	9,600	9,600	57.6K	57.6K
1 1 0 0	38.4K	19.2K	230.4K	115.2K
1 1 0 1	Timer	Timer	Timer	Timer
1 1 1 0	IO2 – 16X	IO2 – 16X	IO2 – 16X	IO2 – 16X
1 1 1 1	IO2 – 1X	IO2 – 1X	IO2 – 1X	IO2 – 1X

CR – Command Register

CR is used to write commands to the Octal UART.

CR[7:4] – Miscellaneous Commands

The encoded value of this field can 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 TxFIFO, the start of break is delayed until that character or any others loaded after it have 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 Timeout Mode On. The register in this channel will restart the C/T as each receive character is transferred from the shift register to the RxFIFO. The C/T is placed in the counter mode, the START/STOP counter commands are disabled, the counter is stopped, and the Counter Ready Bit, ISR[3], is reset.
1011	Reserved.
1100	Disable Timeout Mode. This command returns control of the C/T to the regular START/STOP counter commands. It does not stop the counter, or clear any pending interrupts. After disabling the timeout mode, a 'Stop Counter' command should be issued.
1101	Set MR pointer to 0.
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 TxFIFO 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 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.

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 1x 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 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 will be set when the transmitter underruns, i.e., both the transmit holding register and the transmit shift register are empty. This bit and TxRDY are set when transmitter is first enabled after a disable transmitter command or reset. It is set after transmission of the last stop bit of a character, if no character is in the TxFIFO awaiting transmission. It is reset when the TxFIFO is loaded by the CPU, or when the transmitter is disabled.

SR[2] – Transmitter Ready (TxRDY)
This bit, when set, indicates that the TxFIFO is ready to be loaded with a character. This bit is cleared when the TxFIFO is full 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 TxFIFO 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 eight 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 RxFIFO, and no more characters are in the FIFO.

ACR – Auxiliary Control Register

ACR[7] – Baud Rate Generator Set Select
This bit selects one of two sets of baud rates 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, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, 19.2k, and 38.4k 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).

Table 4. ACR[6:4] C/T Clock and Mode Select

[6:4]	Mode	Clock Source
0 0 0	Counter	IO pin
0 0 1	Counter	IO pin divided by 16
0 1 0	Counter	TxC–1X clock of the transmitter A or C
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	IO pin
1 0 1	Timer	IO 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

ACR[3:0] – IO1b, IO0b, IO1a, IO0a 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] – IO1b, IO0b, IO1a, IO0a 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] – IO1b, IO0b, IO1a, IO0a Change-of-State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the inputs pins at the time the IPCR is read

ISR – Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output 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] – IO Change-of-State
This bit is set when a change-of-state occurs at the IO1b, IO0b, IO1a, IO0a 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 receiver 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 holding 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[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 ready 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 receiver 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 holding 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 effect 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 into the CTUR/CTLR registers is H'0002'. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half-periods will be. The C/T will not be running until it receives an initial 'Start Counter' command (read at address A3–A0 = 1110). After this, while in timer mode, the C/T will run continuously. Receipt of a subsequent start counter command causes the C/T to terminate the current timing cycle and to begin a new cycle using the values in the CTUR and CTLR

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

In the counter mode, the C/T counts down the number of pulses loaded in CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command Upon reaching the terminal count H'0000', the counter ready interrupt bit (ISR[3]) is set The counter continues counting past the terminal count until stopped by the CPU. If IO 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 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 (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower eight bits to the upper eight bits occurs between the times that both halves of the counter is read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

I/O LOGIC

Another difference between the QUART and most other Signetics UART products is that the QUART has four pins for each channel which can be inputs or outputs, while previous Signetics UART devices have varying number of fixed-direction input pins and output pins.

IPR (for DUART ab)

3b	2b	3a	2a	1b	0b	1a	0a
1	1	1	1	1	1	1	1

The state of all eight pins for each DUART can always be read via the IPR

IPCR (for DUART ab)

Δ1b	Δ0b	Δ1a	Δ0a	1b	0b	1a	0a
1	1	1	1	1	1	1	1

IPR and IPCR are analogous to registers of the same name in the SCC2698.

IOPCR (for DUART ab)

IO3x use	IO2x use	IO1x use	IO0x use
2	2	2	2

The configuration of I/O pins as inputs or outputs in each UART channel is controlled by a register called IOPCR (I/O Port Control Register). This register generally replaces the OPCR (Output Port Control Register) of previous Signetics UARTs. The coding of the IOPCR control fields is shown in Figure 4.

OPR (for DUART ab)

3b	2b	3a	2a	1b	0b	1a	0a
1	1	1	1	1	1	1	1

For I/O pins that are selected for output port control in IOPCR, this register controls their state. The OPRs are read/write at address 0Ch and 1Ch. Read/write output registers optimize the changing of some bits without affecting the state of other bits in the register.

Each of a channel's 4 I/O lines are configured to be inputs upon reset. Each UART channel has two inputs, IO0 and IO1, that are equipped with change detection. The ACR (Auxiliary Control Register) controls interrupt generation by change of state of these inputs. ACR[0] enables change of state interrupt on IO0a, ACR[1] on IO1a, ACR[2] on IO0b and ACR[3] on IO1b.

IOPCR Control of IO3:0 Pins for Each Channel

IOPCR7 IOPCR6	IOPCR5 IOPCR4	IOPCR3 IOPCR2	IOPCR1 IOPCR0
IO3x Control	**IO2x Control**	**IO1x Control**	**IO0x Control**
00 = GPI or TxC in	00 = GPI or RxC in	00 = (Note 2)	00 = GPI or CTS in
01 = OPR3 out	01 = OPR2 (Note 1)	01 = OPR1 out	01 = OPR0 out
10 = TxC16X out	10 = RxC1X out	10 = (Note 3)	10 = TxC1X out
11 = TxC1X out	11 = RxC16X out	11 = RxC1X out	11 = TxC16X out

Note 1: Bit 2 of the OPR is the Request to Send function that is affected by the Assert and Negate RTS commands in CRA or CRB, and by the TxRTS feature if MR2x5 is 1, as well as by writing OPR. The RxRTS function, which is activated if MR1x7 is 1, does not affect OPR2 but merely blocks the output signal whenever the Rx FIFO is full.

Note 2: As for the other three pins, a 00 value in this field makes the IO1 pin an input. IO1 can always be used as a General Purpose Input (GPI). IO1a and IO1c can be used as CTI depending on how ACR6:4 is programmed. If OPCR7:6 is not 00, IO1 can be used as TxC depending on how CSRx3:0 is programmed.

Note 3: A 10 value in this field makes IO1b and IO1d output the CTO signal, and makes IO1a and IO1c output RxC16x.

QUART REGISTERS vs 2698B

As shown in Table 1, registers present in the SCC2698's first four channels are present in the QUART. Furthermore, they retain the same addresses and functionally that they possess in the 2698. Thus, the QUART is compatible with existing software except for the multi-purpose I/O and some interrupt related functions.

Revised Formats for Existing Registers

The function of bit 6 of the MR1 register (RxINT Select) has been superseded by the new interrupt priority logic. This bit is now reserved.

The ACR has analogous but slightly different functions for bits 3:0 as described in the I/O Logic section.

The new format of the IOPCR, shown above, does not provide for the power down bit. The power down function is now set by a write to QUART address 24H; power up is a write to address 25H.

New Registers Required

Current Interrupt Register (CIR)

# Bytes	Type	Chan #
3	3	2

The Channel # field indicates which of the four UARTs has the highest priority interrupt currently outstanding, while the Type field indicates its source within the UART. The Type field is encoded as follows:

000 No Interrupt
001 Change of State
x10 Transmit available
011 Receive available, no error
100 Receiver break change
101 Counter/Timer
111 Receive available, w/errors

With Type = x11, the # Bytes field indicates the count of received bytes available for reading, while with Type = x10 it indicates the number of bytes that can be written to the transmit FIFO. With Type = x0x, the # Bytes field is undefined.

The CIR is Read only at address 28H

Global Interrupt Byte Count (GIBC)

00000	# Bytes
5	3

The GIBC is not an actual register but simply outputs the interrupting UART's transmit or receive byte counter value. The count, accurate at the time IACKN asserts, is captured in the CIR. The high order 5 bits are read as '0'. The GIBC is read only at address 2AH

Global Interrupt Byte Count (GIBC)

Received Data
8

Like the GIBC., no physical register implementation exists. The correct receiver's FIFO is popped based on the value of the interrupting channel field of the Current Interrupt Register.

If a receiver is not the cause of the current interrupt, a read of the Global RxFIFO will yield a byte containing all ones and NONE of the UART channels' receive FIFOs will be popped. (IMPORTANT)

The GRxFIFO is Read only at address 2BH

Global Transmit Holding Register (GTHR)

Data to be Sent
8

Similar to the GRxFIFO, no physical register implementation exists. The byte is pushed into the correct transmitter's FIFO based on the interrupting channel field of the Current Interrupt Register.

If a transmitter is not the cause of the current interrupt, a write to the Global TxFIFO has no effect.

The GTxFIFO is Write only at address 2BH

Global Interrupting Channel (GICR)

000000	Chan #
6	2

Like the other Global pseudo-registers no hardware register exists. The Channel number field of the Current Interrupt Register padded with leading zeros is output as the GICR. The GICR is Read only at address 29H.

Interrupt Control (ICR)

Threshold	IVC
6	2

The Threshold Field is used by the interrupt comparator to determine if a winning interrupt "bid" should result in interrupting the control processor. This field resets to 3Fh.

The IVC field controls what kind of vector the QUART returns to the control processor during an Interrupt Acknowledge cycle:

00 Output contents of Interrupt Vector Register
01 Output 6 MSBs of IVR and Channel number as 2 LSBs
10 Output 3 MSBs of IVR and Channel number and Interrupt Type
11 Disable generation of vector during IACK cycle

This field reset to 00 and is read/write at address 2CH.

Bidding Control Registers (BCRs)

Rcv'd Break	State Change	C/T
3	3	2

The 3 MSBs determine the priority of Received Break Interrupts; they are reset to 001.

Bits 4:2 determine the priority of Change of Input State interrupts, and are reset to 00.

There is one BCR per UART channel; they can be read or written at addresses 20-23H.

Interrupt Vector (IVR)

Always Used	with IVC = 0x	w/IVC > 00
3	3	2

Holds the constant bits of the interrupt acknowledge vector. As shown, the three MSBs are always used, while the less significant bits can be replaced by the interrupt type code and/or Channel code bits. The IVR is write only at address 29H.

OPEN ISSUES

This specification is Preliminary. Final performance values will follow characterization AC parameters have been taken from first fabrication lots

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature range[2]	Note 4	°C
T_{STG}	Storage temperature range	–65 to +150	°C
V_{CC}	Voltage from V_{DD} to GND[3]	–0.5 to +7.0	V
V_S	Voltage from any pin to ground[3]	-0.5 to V_{CC} +0.5	V
P_D	Power dissipation	1	W

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. See ordering information table for applicable temperature range and operating supply range.

DC ELECTRICAL CHARACTERISTICS[1, 2, 3]

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)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		$0.8V_{CC}$			V
V_{OL}	Output Low voltage	I_{OL} = 4.0mA			0.4	V
V_{OH}	Output High voltage (except OD outputs)	I_{OH} = –400µA	$0.8V_{CC}$			V
			$0.9V_{CC}$			V
I_{IL}	Input current Low, I/O ports	V_{IN} = 0	–10			µA
I_{IH}	Input current High, I/O ports	V_{IN} = V_{CC}			1	µA
I_I	Input leakage current	V_{IN} = 0 to V_{CC}	–1		1	µA
I_{ILX1}	X1/CLK input Low current	V_{IN} = GND, X2 = open	–100			µA
I_{IHX1}	X1/CLK input High current	V_{IN} = V_{CC}, X2 = open			100	µA
I_{OZH}	Output off current High, 3-State data bus	V_{IN} = V_{CC}			10	
I_{OZL}	Output off current Low, 3-State data bus	V_{IN} = 0	–1		1	µA
I_{ODL}	Open-drain output Low current in off state: IRQN	V_{IN} = V_{CC}	–10			
I_{ODH}	Open-drain output Low current in off state: IRQN	V_{IN} = 0			1	µA
I_{CC}	Power supply current Operating mode	TTL input levels 25°C with X1 = 4MHz			50	mA
	Power down mode				TBD	µA

NOTES:

1. Parameters are valid over specified temperature range. See ordering information table for applicable temperature range and operating supply range.
2. All voltage measurements are referenced to ground (GND). For testing, all inputs 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 V_{IL} and V_{IH}, as appropriate
3. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

AC ELECTRICAL CHARACTERISTICS[1, 2, 3, 4] $T_A = 20°C$; $V_{CC} = 5V \pm 10\%$, unless otherwise specified

SYMBOL	FIGURE	PARAMETER		LIMITS			UNIT
				Min	Typ	Max	
Reset timing							
t_{RES}	5	Reset pulse width		200			ns
IO Port timing[8]							
t_{PS}	6	IO input setup time before RDN Low		0			ns
t_{PH}	6	IO input hold time after RDN High		0			ns
Interrupt timing							
t_{IR}	7	INTRN negated or IO output High from: Read RHR (RxRDY/FFULL interrupt) Write THR (TxRDY interrupt) Reset command (break change interrupt) Reset command (IO change interrupt) Stop C/T command (counter interrupt) Write IMR (clear of interrupt mask bit)	With respect to a 3.6864MHz clock on pin X1/CLK			 100 100 100 100 100 100	 ns ns ns ns ns ns
Clock timing							
t_{CLK}	8	X1/CLK low time/high time		125/100			ns
t_{CLK}	8	X1/CLK frequency		2.0	3 6864	4.0	MHz
t_{CTC}	8	Counter/timer clock high or low time		60			ns
f_{CTC}	8	Counter/timer clock frequency		0[11]		8	MHz
t_{RX}	8	RxC high or low time		30			ns
f_{RX}	8	RxC frequency (16X) RxC frequency (1X)		0[11] 0[11]		16 1 0	MHz MHz
t_{TX}	8	TxC high or low time		200			ns
f_{TX}	8	TxC frequency (16X) TxC frequency (1X)		0[11] 0[11]		16 1.0	MHz MHz
Transmitter timing							
t_{TXD}	9	TxD output delay from TxC low 16X				120	ns
t_{TCS}	9	TxD output delay from TxC output		0		50	ns
Receiver timing							
t_{RXS}	10	RxD data setup time to RxC high		100			ns
t_{RXH}	10	RxD data hold time from RxC high		100			ns

NOTES:

1. Parameters are valid over specified temperature range. See ordering information table for applicable temperature range and operating supply range.
2. All voltage measurements are referenced to ground (GND). For testing, all inputs 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 V_{IL} and V_{IH}, as appropriate.
3. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
4. Test condition for interrupt and IO outputs. $C_L = 50pF$, forced current for $V_{OL} = 5.3mA$; forced current for $V_{OH} = 400\mu A$, $R_L = 2.7k\Omega$ to V_{CC}. Test conditions for rest of outputs: $C_L = 150pF$.

AC ELECTRICAL CHARACTERISTICS (PRELIMINARY)[4] T_A = 20°C; V_{CC} = 5V ± 10%, unless otherwise specified.

NO.	FIGURE	CHARACTERISTIC	LIMITS			UNIT
			Min	Typ	Max	
1	1a	Setup: A[5:0] valid to CEN low	10			ns
2	1a	Hold: A[5:0] valid after CEN low	30			ns
3	1a	Access: Later of CEN low and RDN low, to Dnn valid (read)			110	ns
4	1a	Later of CEN low and (RDN or WRN as applicable) low, to DACKN low Normal Operation: From Power Down:			70 + 2 X1 edges 150	ns ns
5	1a	Earlier of CEN high or RDN high, to Dnn released (read)[1]	0		45	ns
6	1a	Earlier of CEN high or (RDN or WRN as applicable) high, to DACKN released	0		30	ns
7	1a	Earlier of CEN high or (RDN or WRN as applicable) high, in one cycle, to later of CEN low and (RDN or WRN as applicable) low, for the next cycle	55			ns
8	1a	Setup, Dnn valid (write) to later of CEN low and WRN low[2]	–30			ns
9	1a	Later of CEN low and WRN low, to earlier of CEN high or WRN high	110			ns
10	1a	Hold: Dnn valid (write) after DACKN low, CEN high or WRN high[3]	0			ns

NOTES:
1. The minimum time indicates that read data will remain valid until the bus master drives CEN and/or RDN to high.
2. The fact that this parameter is negative means that the Dnn line may actually become valid after CEN and WRN are both low.
3. In a Write operation, the bus master must hold the write data valid either until drives CEN and/or WRN to high, or until the QUART drives DACKN to low, whichever comes first.
4. Test condition for interrupt and IO outputs: C_L = 50pF, forced current for V_{OL} = 5.3mA; forced current for V_{OH} = 400μA, R_L = 2.7kΩ to V_{CC}. Test conditions for rest of outputs: C_L = 150pF.

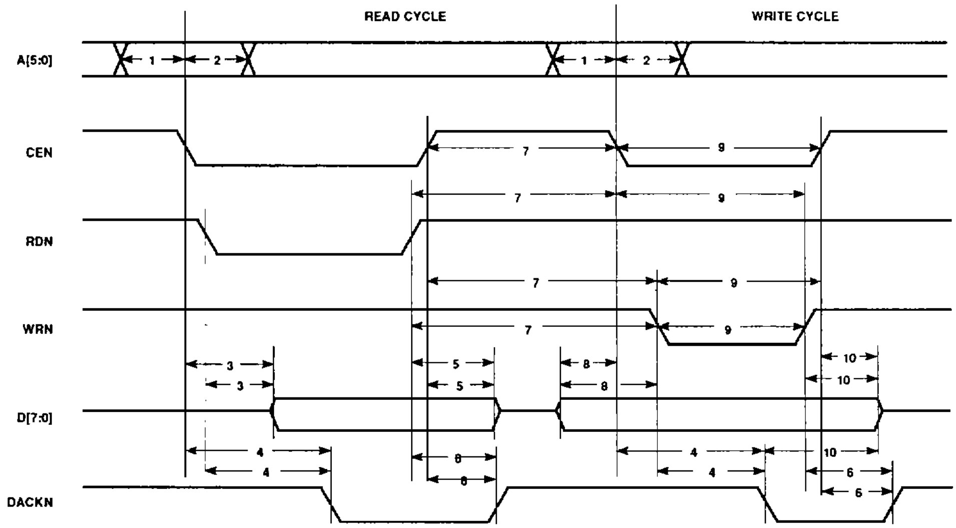


Figure 1a. A Read Cycle Followed by a Write Cycle With DACKN

AC ELECTRICAL CHARACTERISTICS (PRELIMINARY)[1] $T_A = 0°C$; $V_{CC} = 5V \pm 10\%$, unless otherwise specified.

NO.	FIGURE	CHARACTERISTIC	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
1	1b	A[5:0] Setup time to RDN WRN Low	10			ns
2	1b	A[5:0] Hold time from RDN WRN Low	30			ns
3	1b	CEN Setup time to RDN WRN Low[2]	0			ns
4	1b	CEN Hold time from RDN WRN High[2]	0			ns
5	1b	RDN WRN Pulse width Low	110			ns
6	1b	D[7:0] Data Valid after CEN and RDN Low			110	ns
7	1b	D[7:0] Data bus floating after RDN or CEN High			45	ns
8	1b	D[7:0] Data bus setup time before WRN or CEN High	75			ns
9	1b	D[7:0] Hold time after WRN or CEN High	0			ns
10	1b	Time between Reads and/or Writes[3]	55			ns

NOTES:

1 Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. 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.

2. The RDN signal must be negated for t_{RWD} guarantee that any status register changes are valid

3. Consecutive write operations to the command register require at least three rising edges of the X1 clock between writes.

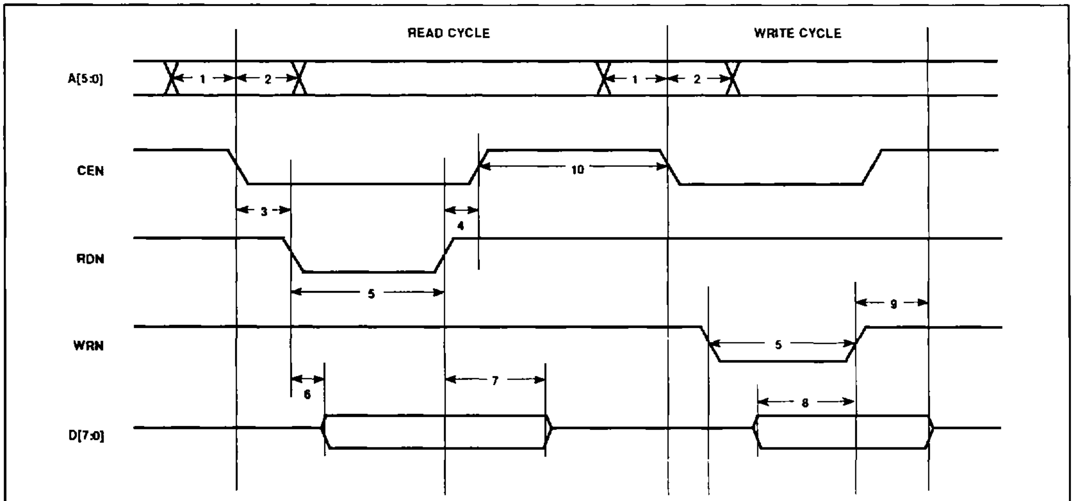


Figure 1b. A Read Cycle Followed by a Write Cycle Without DACKN

AC ELECTRICAL CHARACTERISTICS (PRELIMINARY) T_A = 0°C; V_{CC} = 5V ± 10%, unless otherwise specified.

NO.	FIGURE	CHARACTERISTIC	LIMITS			UNIT
			Min	Typ	Max	
1	1c	D[7:0] Valid after IACKN Low			110	ns
2	1c	DACKN Low after IACKN Low	30		10 + 2 X1 edges	ns
3	1c	D[7:0] floating after IACKN High			45	ns
4	1c	DACKN High after IACKN High			45	ns

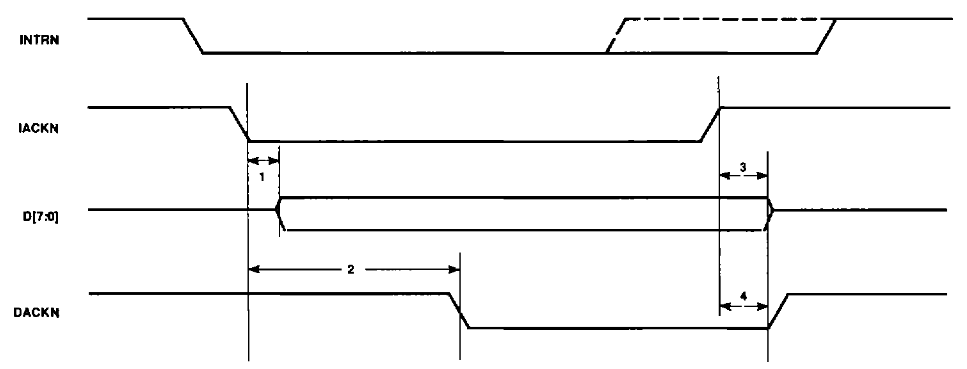


Figure 1c. Interrupt Knowledge (IACKN) Timing

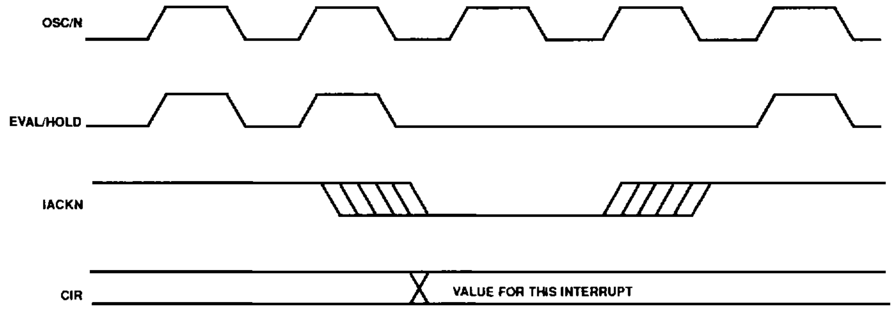


Figure 2. Interrupt Bid Arbitration Timing.

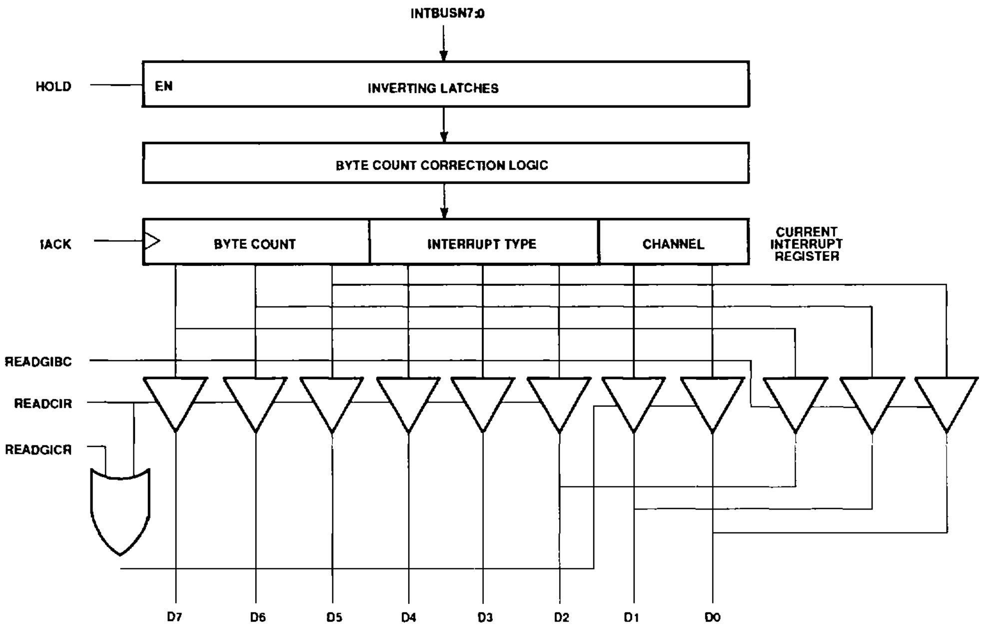


Figure 3. Current Interrupt Register Logic

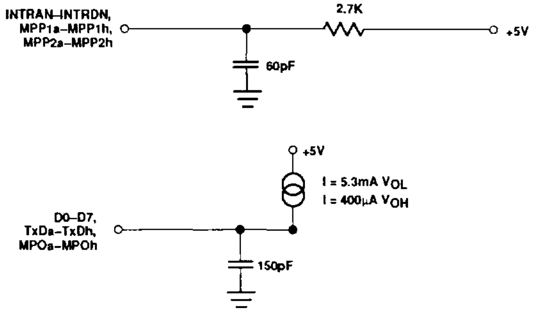


Figure 4. Test Conditions on Outputs

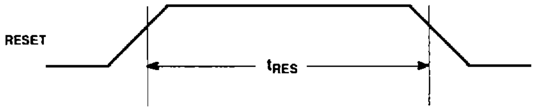


Figure 5. Reset Timing

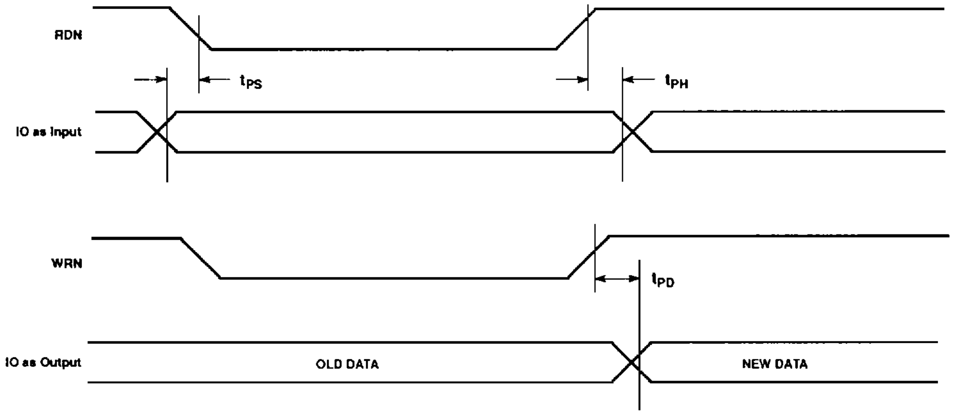
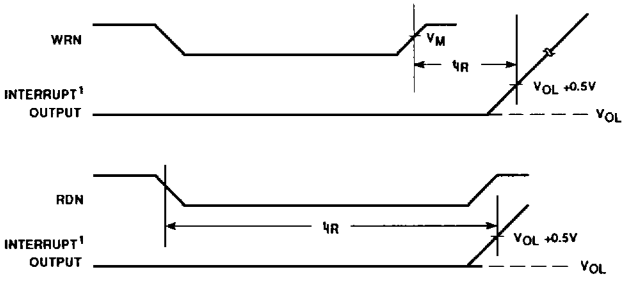


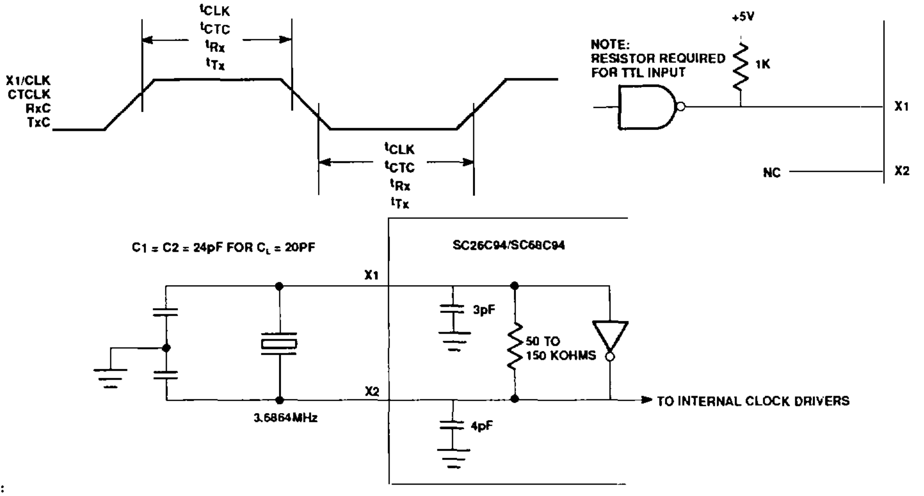
Figure 6. IO Port Timing



NOTES:

1. Includes IO when used as TxRDY or RxDY/FFULL outputs as well as INTRN.
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 7. Interrupt Timing



NOTE:
C1 and C2 should be based on manufacturer's specification.
X1 and X2 parasitic capacitance is 1-2pF and 3-5pF, respectively.

TYPICAL CRYSTAL SPECIFICATION

FREQUENCY:	2 – 4MHZ
LOAD CAPACITANCE (C_L):	12 – 32pF
TYPE OF OPERATION:	PARALLEL RESONANT, FUNDAMENTAL MODE

Figure 8. Clock Timing

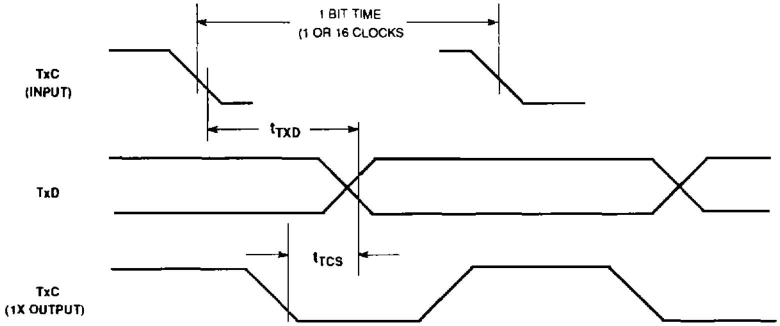


Figure 9. Transmit Timing

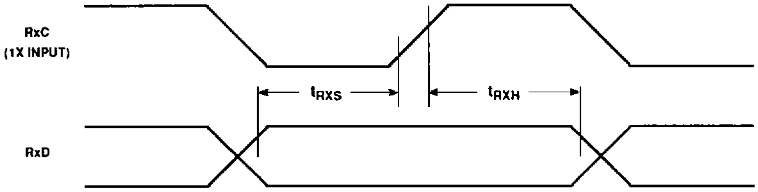
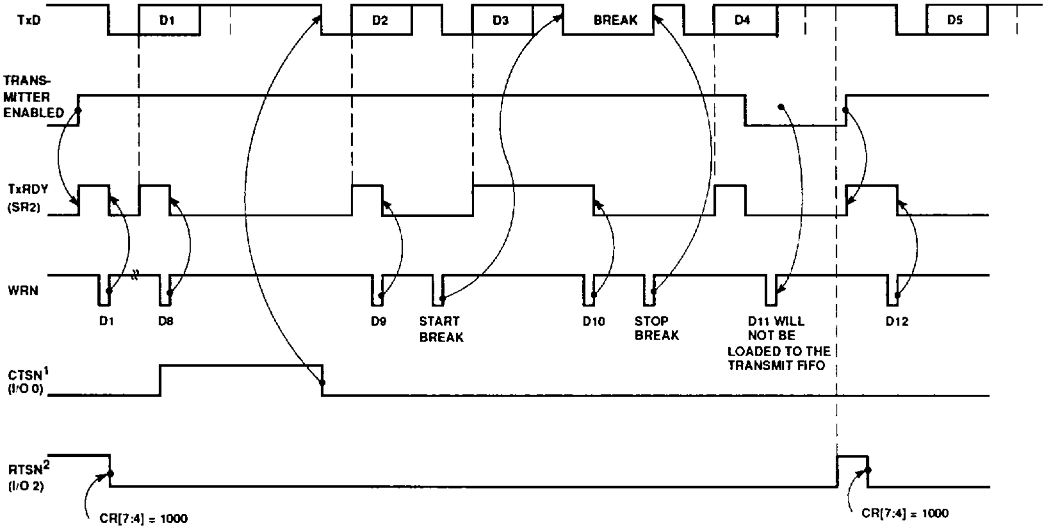
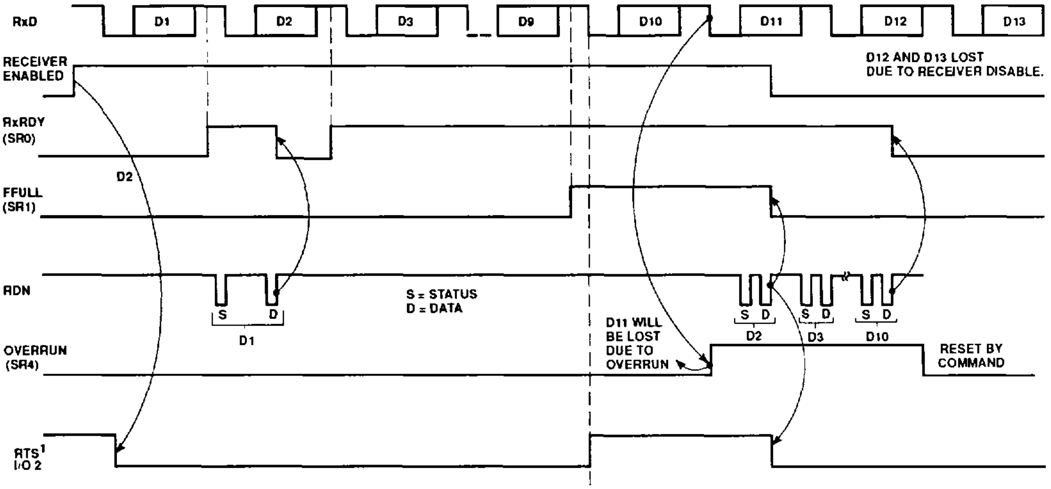


Figure 10. Receive Timing



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

Figure 11. Transmitter Timing



NOTES;
1 Timing shown for MR1[7] = 1.

Figure 12. Receiver Timing

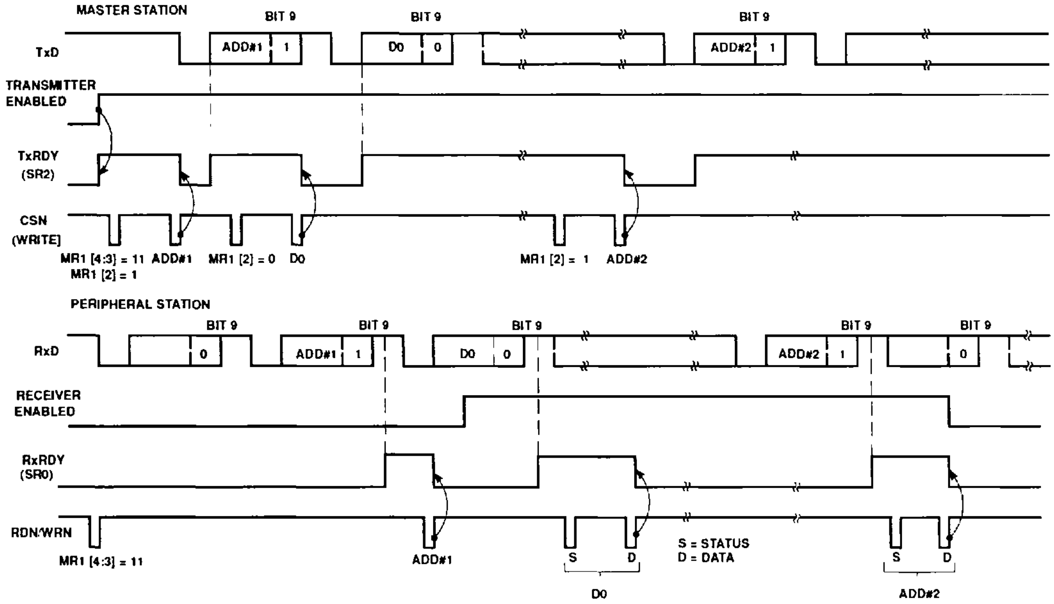


Figure 13. Wake-Up Mode