Quad universal asynchronous receiver/transmitter (QUART) SC68C94

DESCRIPTION

The 68C94 quad universal asynchronous receiver/transmitter (QUART) combines four enhanced Philips Semiconductors industry-standard UARTs with an innovative interrupt scheme that can vastly minimize host processor overhead. It is implemented using Philips Semiconductors' 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 QUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

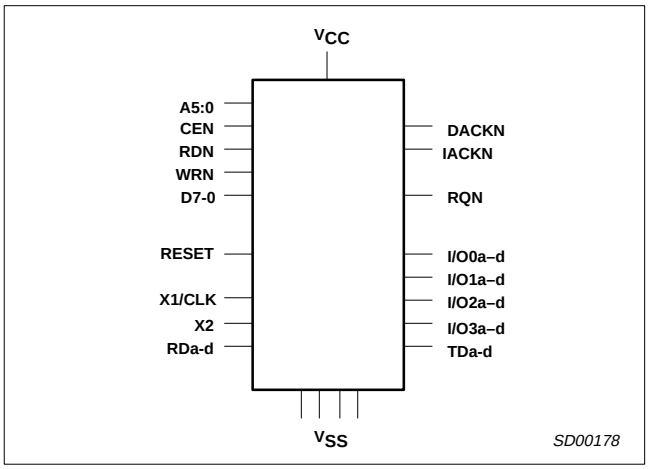
Each receiver is buffered with eight character FIFOs (first-in-first-out memories) and one shift register to minimize the potential for receiver overrun and to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking capability is provided to disable a remote UART transmitter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

The 68C94 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 QUART is fully TTL compatible and operates from a single +5V power supply.

FEATURES

- New low overhead interrupt control
- Four Philips Semiconductors industry-standard UARTs
- Eight byte receive FIFO and eight byte transmit FIFO 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:
  - 23 fixed rates: 50 to 230.4K baud Non-standard rates to 1.0M baud
  - User-defined rates from the programmable counter/timer associated with each of two blocks
  - External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation

PIN CONFIGURATIONS



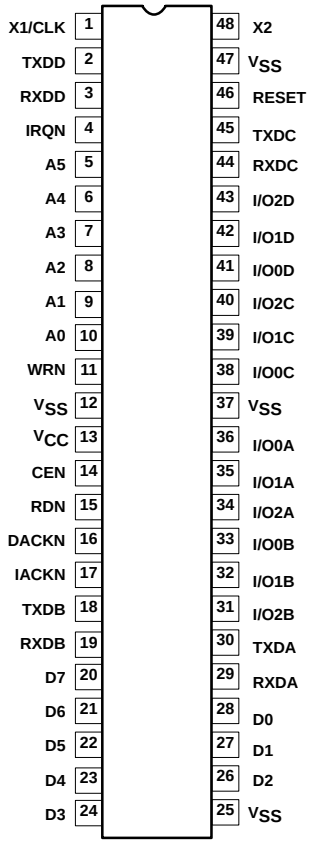
- Programmable channel mode
  - Normal (full-duplex), automatic echo, local loop back, remote loopback
- Programmable interrupt priorities
- Identification of highest priority interrupt
- Global interrupt register set provides data from interrupting channel
- Vectored interrupts with programmable vector format
- IACKN and DTACKN signals
- Built-in baud rate generator with choice of 18 rates
- 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 with low power mode
- Two multifunction programmable 16-bit counter/timers
- 1MHz 16x mode operation
- 30ns data bus release time
- "Watch Dog" timer for each receiver

ORDERING INFORMATION

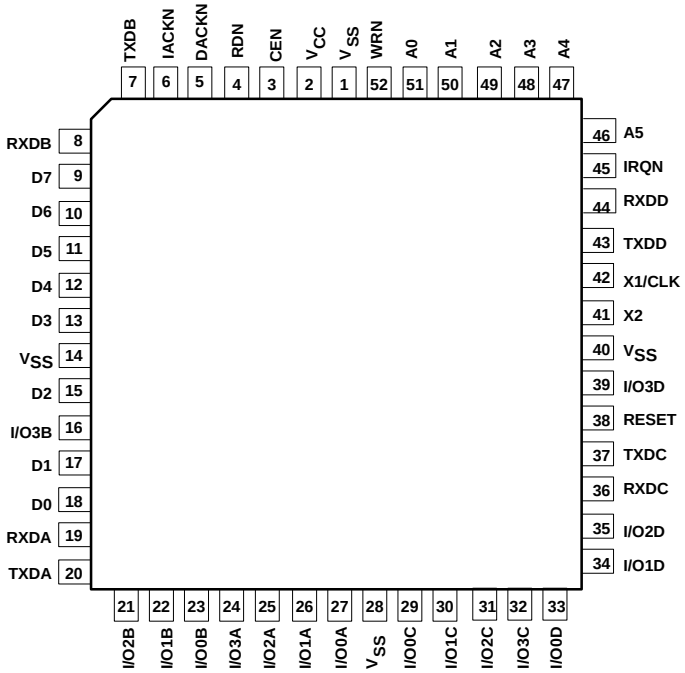
PACKAGES	COMMERCIAL	INDUSTRIAL	DWG #
	V_{CC} = +5V ±10%, T_A = 0°C to +70°C	V_{CC} = +5V ±10%, T_A = –40°C to +85°C	
48-Pin Plastic Dual In-Line Package (DIP)	SC68C94C1N	SC68C94A1N	SOT240-1
52-Pin Plastic Leaded Chip Carrier (PLCC)	SC68C94C1A	SC68C94A1A	SOT238-3

PIN CONFIGURATIONS

48-Pin Dual-In-Line Package



52-Pin PLCC Package



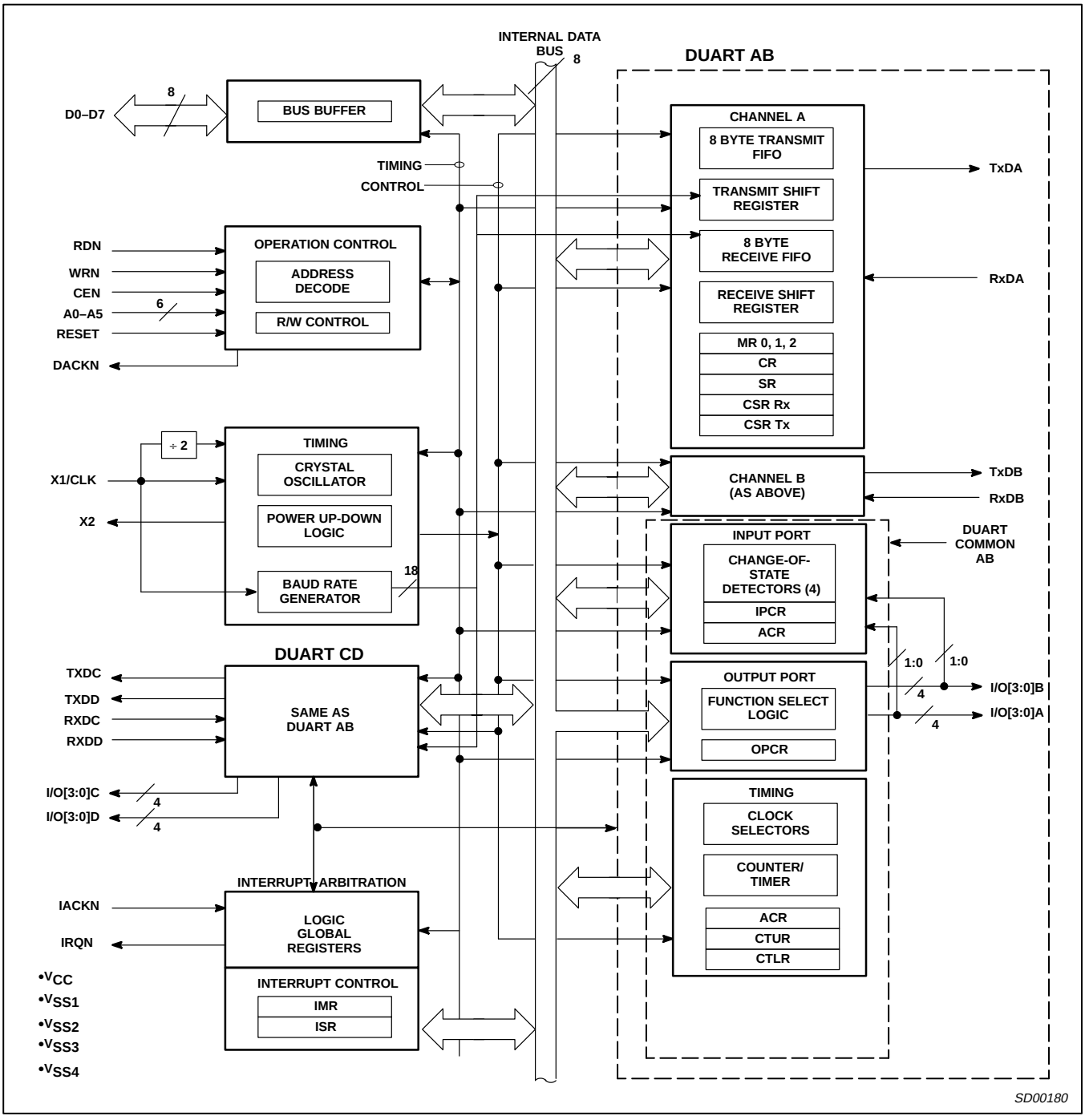
SD00179

ABSOLUTE MAXIMUM RATINGS[1, 2]

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

BLOCK DIAGRAM



PIN DESCRIPTION

MNEMONIC	TYPE	NAME AND FUNCTION
CEN	I	**Chip Select:** Active low input that, in conjunction with RDN or WRN, indicates that the host MPU is trying to access a QUART register. CEN must be inactive when IACKN is asserted.
A5:0	I	**Address Lines:** These inputs select a 68C94 register to be read or written by the host MPU.
D7:0	I/O	**8-bit Bidirectional Data Bus:** Used by the host MPU to read and write 68C94 registers.
RDN	I	**Read Strobe:** Active low input. When this line is asserted simultaneously with CEN, the 68C94 places the contents of the register selected by A5:0 on the D7:0 lines.
WRN	I	**Write Strobe:** Active low input. When this line is asserted simultaneously with CEN, the 68C94 writes the data on D7:0 into the register selected by A5:0.
DACKN	O	**Data ACKnowledge:** Active low, open-drain output to the host MPU, which is asserted subsequent to a read or write operation. For a read operation, assertion of DACKN indicates that register data is valid on D7:0. For a write operation, it indicates that the data on D7:0 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 host MPU indicating that one or more of the enabled UART interrupt sources 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 indicates host MPU is acknowledging an interrupt requested. The 68C94 responds by placing an interrupt vector or interrupt vector modified on D7-D0 and asserting DACKN. This signal updates the CIR register in the interrupt logic. CEN must be high during this cycle.
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 68C94 logic. Must be asserted at power-up, may be asserted at other times that the system is to be reset and restarted. OSC set to divide by 1, MR pointer set to 1, DACKN enabled, I/O pins to input. Registers reset: OPR, CIR. IRQN, DTACKN, IVR Interrupt Vector, Power Down, Test registers, FIFO pointers, Baud rate generator, Error Status, Watch Dog Timers, Change of State detectors, counter/timer to timer, Transmitter and Receiver controllers and all interrupt bits. If reset pin is not used, then first chip access should be to celar 'power-down' mode.
X1/CLK	I	**Crystal 1 or Communication Clock:** This pin is normally connected to one side of a 3.6864MHz or a 7.3728MHz crystal, or can be connected to an external clock up to 8MHz.
X2	O	**Crystal 2:** If a crystal is used, this pin should be connected to its other terminal. If an external clock is applied to X1, this pin should be left unconnected.
V_{CC}, V_{SS}		**Power and grounds:** respectively.

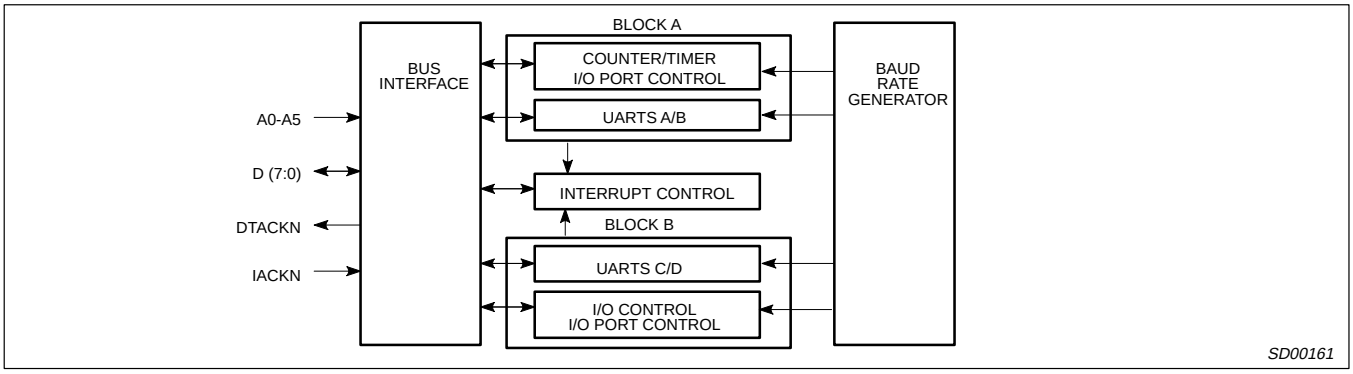


Figure 1. Channel Architecture

Table 1. QUART Registers[1]

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)
000011	Receive Holding Register a (RxFIFOa)	Transmit Holding Register a (TxFIFOa)
000100	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 (SRb)	Clock Select Register b (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/OPCRa (I/O Port Control Reg a)
001110	Start Counter ab	I/OPCRb (I/O Port Control Reg b)
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/OPCRc (I/O Port Control Reg c)
011110	Start Counter cd	I/OPCRd (I/O Port Control Reg d)
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 Interrupting 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	BRG Rate. 00 = low; 01 = high
101110	Reserved	Set X1/CLK divide by two[2]
101111	Reserved	Set X1/CLK Normal[2]
110000–111000	Reserved	Reserved
111001	Test Mode	Test Mode
111010–111111	Reserved	Reserved

NOTES:
1. Registers not explicitly reset by hardware reset power up randomly.
2. In X1/CLK divide by 2 all circuits receive the divided clock except the BRG and change-of-state detectors.

FUNCTIONAL BLOCKS

The QUART is composed of four Philips Semiconductors 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, counter/timers, change of state detectors, break detect or receiver error. 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 Philips Semiconductors 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 an 8.0MHz clock rates to 500K baud. Other rates are available by setting the BRG rate to high at address 2D hex or setting Test 1 on at address 39 hex. See Table 3. These two modes are controlled by writing 00 or 01 to the addresses above. They are both set to 00 on reset. External Rx and Tx clocks yield rates to 1MHz in the 16X mode.

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.

Channel Blocks

There are two blocks (Block Diagram), 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 microprocessor 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 or a command 1x to the Command Register for compatibility with other Philips Semiconductors software. It is set to 0 via a command Bx 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 is 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, power up/down logic and a divide by 2 selector. Closely associated with the timing block are two 16-bit counter/timers; one for each DUART.

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 10. 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 X1 pin always supplies the clock for the baud rate generator. The X1 pin also has a feature such that it may be divided by 2. The divide by two mode must always be used whenever the X1 pin is above 4MHz. The baud rate generator supplies the standard rates when X1 is at 3.6864MHz. In the divide by 2 mode, all circuits receive the divide by two clock except baud rate generator and I/O pin change-of-state detectors. 7.3738MHz clock doubles standard baud rates.

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. The eighteen BRG rates are grouped in two groups. Eight of the 18 are common to each group. The group selection is controlled by ACR[7]. See the Baud Rate Table 3. 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

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. 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 block. 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 (see Table 1) is reserved to issue a start counter/timer command and a second register read address is reserved to issue a stop command. The value of D(7:0) is ignored. 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 register.

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.

When the C/T is used to generate a baud rate *and* the C/T is selected through the CSR then the receivers and/or transmitter will be operating in the 16x mode. Calculation for the number 'n' to program the counter timer upper and lower registers is shown below.

n=2 x 16 x Baud rate desired/(C/T Clock Frequency

Often this division will result in a non-integer number; 26.3 for example. One can only program integer numbers to a digital divider. Therefore 26 would be chosen. This gives a baud rate error of 0.3/26.3 which is 1.14%; well within the ability of the asynchronous mode of operation.

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.The time-out mode forces the C/T into the timer mode. 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 time of MR0. The difference is in the programmability of the delay time and that the watchdog timer is restarted by either a receiver load to the RxFIFO or a system read from it.

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. Only one receiver should use this mode at a time. However, if both are on, the timeout occurs after both receivers have been inactive for the timeout period. The start of the C/T will be on the logical or of the two receivers.

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.

Time Out Mode Caution

When operating in the special time out mode, it is possible to generate what appears to be a "false interrupt", i.e., an interrupt without a cause. This may result when a time-out interrupt occurs and then, BEFORE the interrupt is serviced, another character is received, i.e., the data stream has started again. (The interrupt latency is longer than the pause in the data strea.) In this case, when a new character has been receiver, the counter/timer will be restarted by the receiver, thereby withdrawing its interrupt. If, at this time, the interrupt service begins for the previously seen interrupt, a read of the ISR will show the "Counter Ready" bit not set. If nothing else is interrupting, this read of the ISR will return a x'00 character.

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.

If the transmitter is disabled it continues operating until the character currently being transmitted and any characters in the TxFIFO, including parity and stop bits, have been transmitted. New data cannot be loaded to the TxFIFO when the transmitter is disabled.

The transmitter can be forced to send a break (a continuous low condition) by issuing a START BREAK command via the CR register. The break is terminated by a STOP BREAK command or a transmitter reset..

TxFIFO

The TxFIFO empty positions are encoded as a three bit number for presentation to the bidding logic. The coding will equal the number of bytes that remain to be filled. That is, a binary number of 101 will mean five bytes may be loaded; 111 means 7, etc. Eight positions will be indicated by a binary 111 *and* the FIFO empty bit will be set.

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 via the receiver FIFO.

The receiver operates in two modes: the 1X and 16X. The 16X mode is the more robust of the two. It allows the receiver to establish a phase relation to the remote transmitter clock within 1/16 of a bit time and also allows validation of the start bit. The 1X mode does not validate the start bit and assumes that the receiver clock rising edge is centered in the data bit cell. The use of the 1X mode implies that the transmitter clock is available to the receiver.

When operating in the 16X mode and after the receiver has been enabled the receiver state machine will look for a high to low transition on the RxD input. The detection of this transition will cause the divider being driven by the 16X clock to be reset to zero and continue counting. When the counter reaches 7 the RxD input is sampled again and if still low a valid START BIT will be detected. If the RxD input is high at count 7 then an invalid start bit will have been sensed and the receiver will then look for another high to low transition and begin validating again.

When a valid start bit is detected the receiver state machine allows the 16X divider circuit to continue counting 0 to 15. Each time the receiver passes count 7 (the theoretical center of the bit time) another data bit is clocked into the receiver shift register until the proper number of bits have been received including the parity bit, if used, and 1/2 stop bit. After the STOP BIT is detected the receiver state machine will wait until the next falling edge of the 1X clock and then clock the assembled character and its status bits into the receiver FIFO on the next rising edge of the 1X clock. The delay from the detection of the STOP BIT to the loading of the character to the RxFIFO will be from one half to one and one half X1 clock periods, or twice that if X1/2 is used. Receiver Status Register bits for FIFO READY, FIFO FULL, parity error, framing error, break detect will also set at this time. The most significant bits for data characters less than eight bits will be 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 “Change of Break” bit in the ISR at position 2 or 6 is also set at this time. Note that the “Change of Break” bit will set again when the break condition terminates. The RxD input must return to high for two (2) clock edges of the X1 crystal clock for the receiver to recognize the end of the break condition and begin the search for a start bit. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.**

NOTE: If the RxD input is low when the receiver is enabled and remains low for at least 9/16 of a bit time a valid start bit will be seen and data (probably random) will be clocked into the receiver FIFO. If the line remains low for a full character time plus a stop bit then a break will be detected.

Each receiver is equipped with a watchdog timer. This timer is enabled by MR0[7] and counts 64 RxC1X clocks. Its purpose is to alert the controlling CPU that data is in the FIFO which has not been read. This situation may occur at the end of a message when the last group of characters was not long enough to cause an interrupt.

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; 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 actual number filled positions. Seven filled will be coded as 7. Eight filled positions will be coded as 7 *and* the RxFIFO full status bit will be set.

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.

Watchdog Timer

A “watchdog” timer is associated with each receiver. Its interrupt is enabled by MR0[7]. The purpose of this timer is alerting the control processor that characters are in the RxFIFO which have not been read and/or the datastream has stopped. 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 or a read of the RxFIFO is executed.

WAKE-UP MODE (MULTI-DROP OR 9-BIT)

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 receiver sees a one in the parity position, it considers it an address bit and loads 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 local processor will test for an address match for

this station and if match occurs it will enable the local receiver and receive the following data characters. 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.

NOTE: The time between address and data fields must be enough for the local processor to test the address character and enable the receiver. At bit times approaching 10μs this may begin to be a point of concern.

The parity (Address/Data) bit should not be changed until the last stop bit of an address has been sent. Similarly the A/D bit should not be changed to address until the last stop bit has been sent. Either of these conditions will be indicated by an active TxEMT bit in the SR.

The parity bit is not part of the TxFIFO. It is in the transmitter state machine. However, it could be controlled in the FIFO if 5, 6 or 7 bit data was transmitted by using a 6, 7 or 8 bit character. The most significant bit would then be in the 'parity' position and represent the A/D bit. The design of the UART is based, however, on the A/D bit being controlled from the MR register.

Parity should be changed immediately before the data bytes will be loaded to the transmitter.

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 addressed. The desired polarity of the A/D bit (parity) should be programmed before the TxFIFO is loaded.

The receiver should be enabled before the beginning of the first data bit. The time required is dependent on the interrupt latency of the slave receivers. The transmitter is able to start data immediately after the address byte has been sent.

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 (I/OPCR), Input Port Change Register (IPCR), and Output Port Register (OPR). They may be individually programmed to be inputs or outputs. See Table 5.

I/O0x and I/O1x pins have change of state detectors. The change of state detectors sample the input ports every 26.04μs (with the X1 clock at 3.686400MHz) 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 operate and report changes accordingly. See the register descriptions of the I/O ports for the detailed use of these features.

A read of the IPCR resets the I/O COS (change Of State) detectors.

Interrupt Priority System

The interrupt control for the QUART has been designed to provide very low interrupt service overhead for the controlling processor while maintaining a high degree of flexibility in setting the importance of interrupts generated in different functional blocks of the device.

This is accomplished by allowing each function of the QUART (18 total) which may cause an interrupt to generate a variable numeric code which contains the identity of the source, channel number and severity level. This code is compared (at the X1 clock rate or the X1 clock rate divided by 2) to an interrupt threshold. When the interrupting source generates a code that is numerically greater than the interrupt threshold the IRQN is asserted

This is referred to as the bidding process. The winning bid contains, in different fields, all the characteristics of the winning bidder. This data may be used in several ways to steer the controlling processor to the proper type and amount of service required (usually the amount of service refers to the number of bytes written to the transmitter or read from the receiver). Access to the winning bidder is provided via the CIR (Current Interrupt Register), interrupt vectors, modified interrupt vectors and Global registers.

NOTE: IRQN is essentially a level output. It will go active on an interrupt condition and stays active until all interrupting sources are serviced.

IRQN is designed to be an open drain active low level output. It will go low under the control of the arbitration system and remain low until the arbitration has determined that no more sources require service.

When only one Rx or Tx is interrupting, it is possible to see the IRQN assert more than once if, during an access to the FIFO, the CEN input is inactive for more than two cycles of the X1 clock or X1 divide by 2 if that feature is enabled.

IACKN may be thought of as a special read input. Driving IACKN low will update the CIR and then read the Interrupt Vector Register or the Interrupt Vector Register modified by the CIR.

Functional Description of the Interrupt Arbitration

For the purpose of this description, a 'source' is any one of the 18 QUART circuits that may generate an interrupt. The QUART contains eighteen sources which may cause an interrupt:

1. Four receiver data FIFO filled functions.
2. Four receiver BREAK detect functions.
3. Four transmitter FIFO space available functions.
4. Four "Change of State" detectors.
5. Two counter/timers.

The interrupt logic at each source produces a numeric code that identifies its interrupt priority condition currently pending. This code is compared to a programmable Interrupt Threshold via the arbitration logic which determines if the IRQN should be asserted. The arbitration logic only judges those possible interrupt sources which have been allowed to bid via the IMR (Interrupt Mask Register).

The arbitration logic produces a value which is the concatenation of the channel number, interrupt type, FIFO fill level and user-defined fields. The channel number and interrupt type fields are hardwired. During the "bid arbitration" process all bids from enabled sources are presented, simultaneously, to an internal interrupt bus. The bidding system and formats are discussed in more detail in 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 arbitration

period follows the period of the X1 clock. The maximum speed is 4.0MHz. If a higher speed X1 clock is used then the X1 clock “divide by 2” feature must be used.

The value of the winning bid determined during the arbitration cycle is compared to the “Interrupt Threshold” contained in the ICR (Interrupt Control Register). If the winning bid exceeds the value of the ICR the IRQN is asserted.

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

SD00162

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.

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 will 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 1/2 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.

The channel address for C/T ab will be encoded as channel B (01)

The channel address for C/T cd will be encoded as channel D (11)

As shown in Figure 4, the bid arbitration process is controlled by the EVAL/HOLDN signal derived from the oscillator clock.

Receipt of an IACKN signal from the host MPU latches the latest “winning bid” from the latched Interrupt Bus into the Current Interrupt Register (CIR). This logic is diagrammed in Figure 5.

If the IACKN falling edge of Figure 4 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 when the interrupting source is cleared.

Once the interrupt is cleared, the programmable value lowered or its byte count value 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.

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 or execution of the “Update CIR” command.

The Current Interrupt Register and associated read logic is shown in Figure 5. Interrupting channel number and the three bit interrupt type code and FIFO fill level 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 CIR7:5 field will 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 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 5. 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 a value equal to the highest programmable filed.

In effect, once latched by an IACK or the Update CIR command, 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.

Table 2. Receiver FIFO 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 bytes in FIFO (Rx FULL)

For the receiver these bits control the number of FIFO positions empty when the receiver will attempt to interrupt. After the reset the receiver FIFO is empty. The default setting of these bits cause the receiver to attempt to interrupt when it has one or more bytes in it.

Table 3. Receiver FIFO 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)

For the transmitter these bits control the number of FIFO positions empty when the receiver will attempt to interrupt. After the reset the transmit FIFO has 8 bytes empty. It will then attempt to interrupt as soon as the transmitter is enabled. The default setting of the MR0 bits (00) condition the transmitter to attempt to interrupt only when it is competely empty. As soon as one byte is loaded, it is no longer empty and hence will withdraw its interrupt request.

INTERRUPT NOTE ON 68C94:

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 even though it is filed above the bid threshold. Although this is not in agreement with the idea that each enabled interrupt source bid with equal authority, it does allow the flexibility of giving particular receiver or transmitters more interrupt importance than others.

This may be used when the Interrupt Threshold is set at or above 100000. Note than in this case the transmitter cannot generate an interrupt. If the interrupt threshold MSBs were set to 011 *and* the ‘Receiver Interrupt Bits’ on the MR registers set to a value other than 00 then the RxFIFO could not generate and interrupt until it had 4, 6 or 8 bytes. This in effect partially defeats the hardwired characteristic that the receiver interrupts should have more importance than the transmitter. This characteristic has been implemented by setting the MSB of the transmitter bid to zero.

Vectored Interrupts

The QUART responds to an Interrupt Acknowledge (IACK) initiated by the host 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

SD00163

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

NOTE: If IACKN is not being used then the command “UPDATE CIR” must be issued for the global and interrupt registers to be updated.

PROGRAMMING UART CONTROL 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.

Table 4. Register Bit Formats, DUART AB. [duplicated for DUART CD]

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

MR0 (Mode Register 0)

Rx Watchdog Timer	RxINT2 bit	TxINT Control	These bits not implemented. They should be considered Reserved.			
0 = off 1 = on	These bits should normally 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	Normally set to 0	0 = Char 1 = Block	00 = With parity 01 = Force parity 10 = No parity 11 = Wake-up mode	0 = Even 1 = Odd	00 = 5 01 = 6 10 = 7 11 = 8

NOTE: *In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.

MR2 (Mode Register 2)

Channel Mode	TxRTS Control	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 C = 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. In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.

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

NOTE: Issuing commands contained in the upper four bits of the “Command Register” should be separated in time by at least three (3) X1 clock edges. Allow four (4) edges if the “X1 clock divide by 2” mode is used. A disabled transmitter cannot be loaded.

SR (Status Register)

Rec’d. Break	Framing Error	Parity Error	Overrun Error	TxEMT	TxRDY	RxFULL	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. Unless reset with the ‘Error Reset’ (CR command 40) or receiver reset, these bits will remain active in the Status Register after the RxFIFO is empty.

ACR (Auxiliary Control Register)

BRG Set Select	Counter/Timer Mode and Source	Delta I/O1b	Delta I/O0b	Delta I/O1a	Delta I/O0a
0 = set 1 1 = set 2	See text	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

Table 4. Register Bit Formats, Duart ab. [duplicated for Duart cd] (continued)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ISR (Interrupt Status Register)							
I/O 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)							
I/O 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]**
CTUR (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)							
I/O3b	**I/O2b**	**I/O3a**	**I/O2a**	**I/O1b**	**I/O0b**	**I/O1a**	**I/O0a**
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

Mode Registers 0, 1 and 2

The addressing of the Mode Registers is controlled by the MR Register pointer. On any access to the Mode Registers this pointer is always incremented. Upon reaching a value of 2 it remains at 2 until changed by a CR command or a hardware reset.

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 is normally set to either 80h or 00h. A read of this register will return 1111 (Fh) in the lower four bits.

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 are normally set to 0 except as noted in the “Interrupt Threshold Calculation” description

MR0[3:0]: These bits are not implemented in the chip. These bits should be be considered “reserved.”

MR1 – Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET, a set pointer command applied via the CR or after an access to MR0. After reading or writing MR1, the pointers are set at MR2.

MR1[7] – Receiver Request-to-Send Flow Control

This bit controls the deactivation of the RTSN output (I/O2x) 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 (the QUART I/O0 pin) of the transmitting device.

Use of this feature requires the I/O2 pin to be programmed as output via the I/OPCR and to be driving a 0 via the OPR. When the RxFIFO is full and the start bit of the ninth character is sensed the receiver logic will drive the I/O2 pin high. This pin will return low when another RxFIFO position is vacant.

MR1[6] – Receiver Interrupt Select 1

This bit is normally set to 0 except as noted in the “Interrupt Threshold Calculation” description. MR1[6] operates with MR0[6] to prevent the receiver from bidding until a particular fill level is attained. For software compatibility this bit is designed to emulate the RxFIFO interrupt function of previous Philips Semiconductors UARTs.

MR1[5] – Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (received break, FE, PE). 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.

In the “Block Error” mode the ORing of the error status bits and the presentation of them to the status register takes place as the bytes enter the RxFIFO. This allows an indication of problem data when the error occurs after the leading bytes have been received. In the character mode the error bits are presented to the status register when the corresponding byte is at the top of the FIFO.

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 (see ‘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.

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.

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

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

NOTE: When the transmitter controls the I/O2 pin (usually used for the RTSN signal) the meaning of the pin is not RTSN at all! Rather it signals that the transmitter has finished transmission. (i.e., end of block).

This bit controls the deactivation of the RTSN output (I/O2) 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. Disable the transmitter after the last byte of the message is loaded to the TxFIFO. At the time the disable command is issued, be sure that the transmitter ready bit is on and the transmitter empty bit is off. If the transmitter empty bit is on (the indication of transmitter underrun) when the disable is issued, the last byte(s) will not be sent.
6. The last character will be transmitted and RTSN will be reset one bit time after the last stop bit.

MR2[4] – Transmitter Clear-to-Send Flow Control
The sate of this bit determines if the CTSN input (I/O0) 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. 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.

RECEIVER NOTE: In all cases, the receiver only checks for a “mark” condition at the center of the stop bit (1/2 to 9/16 bit time into the stop bit position). At this time the receiver has

finished processing the present character and is ready to search for the start bit of the next character.

Table 5. Bit Rate Generator Characteristics Crystal or Clock = 3.6864MHz

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

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

CR – Command Register

CR is used to write commands to the QUART.

CR[7:4] – Miscellaneous Commands

Issuing commands contained in the upper four bits of the "Command Register" should be separated in time by at least three (3) X1 clock edges. Allow four (4) edges if the "X1 clock divide by 2" mode is used. 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.
Only one receiver should use this mode at a time. However, if both are on, the timeout occurs after both receivers have been inactive for the timeout. The start of the C/T will be on the logical 'OR' of the two receivers.

1011 Set MR Pointer to 0.

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 Reserved.

111x Reserved for testing.

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. I/O2x 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	I/O3x – 16X	I/O3x – 16X
1 1 1 1	I/O3x – 1X	I/O3x – 1X

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.

While the transmitter is disabled (or a disable is pending), the TxFIFO may not be loaded.

CR[2] – Enable Transmitter

Enables operation of the transmitter. The TxRDY and TxEMT status bits will be asserted.

CR[1] – Disable Receiver

This command terminates operation of the receiver immediately – a character being received will be lost. However any unread characters in the RxFIFO area are still available. Disable is not the same as a "receiver reset". With a receiver reset any characters not read are 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 6. Baud Rate

	BRG RATE = LOW		BRG RATE = HIGH		TEST 1 = 1	
CSR[7:4]	ACR[7] = 0	ACR[7] = 1	ACR[7] = 0	ACR[7] = 1	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75	300	450	4,800	7,200
0 0 0 1	110	110	110	110	880	880
0 0 1 0	134.5	38.4k	134.5	134.5	1,076	1,076
0 0 1 1	200	150	1200	900	19.2K	14.4K
0 1 0 0	300	300	1800	1,800	28.8K	28.8K
0 1 0 1	600	600	3,600	3,600	57.6K	57.6K
0 1 1 0	1,200	1,200	7,200	7,200	115.2K	115.2K
0 1 1 1	1,050	2,000	1,050	2,000	1,050	2,000
1 0 0 0	2,400	2,400	14.4K	14.4K	57.6K	57.6K
1 0 0 1	4,800	4,800	28.8K	28.8K	4,800	4,800
1 0 1 0	7,200	1,800	7,200	1,800	57.6K	14.4K
1 0 1 1	9,600	9,600	57.6K	57.6K	9,600	9,600
1 1 0 0	38.4k	19.2k	230.4K	115.2K	38.4K	19.2K
1 1 0 1	Timer	Timer	Timer	Timer	Timer	Timer
1 1 1 0	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X
1 1 1 1	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X

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. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 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 'wake-up mode', the parity error bit stores the received A/D (Address/Data) bit.

In the wake-up mode this bit follows the polarity of the A/D parity bit as it is received. A parity of 1 would normally mean address and therefore, the end of a data block.

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.

SRA[3] – Channel A Transmitter Empty (TxEMTA)
This bit will be set when the transmitter underruns, i.e., both the TxEMT and TxRDY bits are set. This bit and TxRDY are set when the transmitter is first enabled and at any time it is re-enabled after either (a) reset, or (b) the transmitter has assumed the disabled state. It is always 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, a pending transmitter disable is executed, the transmitter is reset, or the transmitter is disabled while in the underrun condition.

SR[2] – Transmitter Ready (TxRDY)
This bit, when set, indicates that the TxFIFO has at least one empty location that may be loaded by the CPU. It sets when the transmitter is first enabled. It is cleared when the TxFIFO is full (eight bytes); the transmitter is reset; a pending transmitter disable is executed; the transmitter is disabled when it is in the underrun condition. When this bit is **not** set characters written to the TxFIFO will not be loaded or transmitted; they are lost.

SR[1] – RxFIFO 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] – RxFIFO 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 between two sets of baud rates that are available within each baud rate group generated by the BRG. See Table 3.

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).

The I/O pins available for counter/timer clock source is I/O1a and I/O1c. The counter/timer clock selection is connected to the I/O1 pin and will accept the signal on this pin regardless of how it is programmed by the I/OPCR.

Table 7. ACR[6:4] C/T Clock and Mode Select

[6:4]	Mode	Clock Source
0 0 0	Counter	I/O1 pin
0 0 1	Counter	I/O1 pin divided by 16
0 1 0	Counter	TxC1XA clock of the transmitter
0 1 1	Counter	TxC1XB clock of the transmitter
1 0 0	Timer	I/O1 pin
1 0 1	Timer	I/O1 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
The timer mode generates a squarewave		

ACR[3:0] – I/O1b, I/O0b, I/O1a, I/O0a 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 and thus allow the Change of State Detectors to enter the bidding process. 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 may result 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] – I/O1b, I/O0b, I/O1a, I/O0a Change-of-State Detectors
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] – I/O1b, I/O0b, I/O1a, I/O0a State of I/O Pins
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins during the time the IPCR is read. The IPR is an unlatched register. Data can change during a read.

ISR – Interrupt Status Register

Important: The setting of these bits and those of the IMR are essential to the interrupt bidding process.

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', then the interrupt source represented by this bit is allowed to enter the interrupt arbitration process. It will generate an interrupt (the assertion of INTRN low) only if its bid exceeds the interrupt threshold value. 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 complete status is provided regardless of the contents of the IMR.

ISR[7] – I/O Change-of-State
This bit is set when a change-of-state occurs at the I/O1b, I/O0b, I/O1a, I/O0a 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
Normally the ISR[5] bit being set to one indicates the RxFIFO is filled with one or more bytes and/or the receiver watch dog timer (when enabled) has timed out.

The meaning of ISR[5] is controlled by the MR0[6] and MR1[6] bits which are normally set to 00. The ISR[5] bit setting to one allows the receiver to present its bid to the arbitration logic. This function is explained in the "Interrupt Note On 68C94" and under the "Receiver Interrupt Fill Level".

ISR[5], if set, will reset when the RxFIFO is read. If the reading of the FIFO does not reduce the fill level below that determined by the MR bits, then ISR[5] sets again within two X1 clock times. Further, if the MR fill level is set at 8 bytes AND there is a byte in the receiver shift register waiting for an empty FIFO location, then a read of the RxFIFO will cause ISR[5] to reset. It will immediately set again upon the transfer of the character in the shift register to the FIFO.

NOTE: The setting of ISR[5] means that the receiver has entered the bidding process. It is necessary for this bit to set for the receiver to generate an interrupt. It does **not** mean it **is** generating an interrupt.

ISR[4] – Transmitter Ready Channel b
The function of this bit is programmed by MR0[5:4] (normally set to 00). This bit is set when ever the number of empty TxFIFO positions exceeds or equals the level programmed in the MR0 register. This condition will almost always exist when the transmitter is first enabled. It will reset when the empty TxFIFO positions are reduced to a level less than that programmed in MR0[5:4] or the transmitter is disabled or reset.

The ISR[4] bit will reset with each write to the TxFIFO. If the write to the FIFO does not bring the FIFO above the fill level determined by the MR bits, the ISR[4] bit will set again within two X1 clock times.

NOTE: The setting of ISR[4] means that the transmitter has entered the bidding process. It is necessary for this bit to set for the transmitter to generate an interrupt. It does **not** mean it **is** generating an interrupt.

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
See the description of ISR[5]. The channel ‘a’ receiver operation is the same as channel ‘b’.

ISR[0] – Transmitter Ready Channel a
See the description of ISR[4]. Channel “a” transmitter operates in the same manner as channel “b.”

IMR – Interrupt Mask Register

The programming of this register selects which interrupt sources will be allowed to enter the interrupt arbitration process. This register is logically ANDED with the interrupt status register. Its function is to allow the interrupt source it represents to join the bidding process if the corresponding IMR and ISR bits are both 1. It has no effect on the value in the ISR. It does not mask the reading of the ISR.

CTUR and CTLR – Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 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 address at A5–A0 0Eh for C/T ab or read address 1Eh for C/T cd). 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 address at A5–A0 0Fh for C/T ab or read address 1Fh for C/T cd). The command, however, does not stop the C/T. It only resets the ISR[3] bit; the C/T continues to run. The ISR[3] bit will set again as the counter passes through 0. The generated square wave is output on an I/O 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 rolls over to 65535 and continues counting until stopped by the CPU. If I/O 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

The QUART has four I/O pins for each channel. These pins may be individually programmed as an input or output under control of the I/OPCR (I/O Port Control Register). Functions which may use the I/O pins as inputs (Rx or Tx external clock, for example) are always sensitive to the signal on the I/O pin regardless of it being programmed as an input or an output. For example if I/O1a was programmed to output the RxC1X clock and the Counter/Timer was programmed to use I/O pin as its clock input the result would be the Counter/Timer being clocked by the RxC1X clock.

The 16 I/O ports are accessed and/or controlled by five (5) registers: IPR, ACR, I/OPCR, IPCR, OPR. They are shown in Table 8 of this document. Each UART has four pins. Two of these pins have “Change of State Detectors” (COS). These detectors set whenever the pin to which they are attached changes state. (1 to 0 or 0 to 1) The “Change of State Detectors” are enabled via the ACR. When enabled the COS devices may generate interrupts via the IMR and IPCR registers. Note that when the COS interrupt is enabled that any one or more of the four COS bits in the IPCR will enable the COS bidding. Each of the channel’s four I/O lines are configured as inputs on reset.

The Change of State detectors sample the I/O pins at the rate of the 38.4KHz clock. A change on the pin will be required to be stable for at least 26.04µs and as much as 52.08µs for the COS detectors to confirm a change. Note that changes in the X1/clock frequency will effect this stability requirement.

COS detectors are reset by a read of the IPCR.

Table 8. Register Bit Formats, I/O Section

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

IPCR (Input Port Change Register ab) The lower four bits replicate the lower four bits of the IPR. The upper four bits reads state of Change detectors. Change detectors are enabled in ACR[3:0]. (DUART ab)

DeltaI/O1b	Delta I/O0b	Delta I/O1a	Delta I/O0a	I/O1b	I/O0b	I/O1a	I/O0a
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

IPCR (Input Port Change Register cd) The lower four bits replicate the lower four bits of the IPR. The upper four bits reads state of Change detectors. Change detectors are enabled in ACR[3:0]. (DUART cd)

Delta I/O1d	Delta I/O0d	Delta I/O1c	Delta I/O0c	I/O1d	I/O0d	I/O1c	I/O0c
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

I/OPCR (I/O Port Configuration Register) One register for each UART.

I/O3x CONTROL	I/O2x CONTROL	I/O1x CONTROL	I/O0x CONTROL
Two bits for each I/O pin.			

This register controls the configuration of the I/O ports. It defines them as inputs or outputs and controls what sources will drive them in the case of outputs or which functions they will drive when used as an input. Each pin has four functions and hence two bits to control it. Each UART has one eight bit register to control its four I/O ports.

OPR (Output Port Register cd) for DUART cd

I/O3d	I/O2d	I/O3c	I/O2c	I/O1d	I/O0d	I/O1c	I/O0c
One bit for each pin. When I/O pins are configured as “General Purpose Outputs” the pins will be driven to the complement value of its associated OPR bit.							

OPR (Output Port Register ab) for DUART ab

I/O3b	I/O2b	I/O3a	I/O2a	I/O1b	I/O0b	I/O1a	I/O0a
One bit for each pin. When I/O pins are configured as “General Purpose Outputs” the pins will be driven to the complement value of its associated OPR bit.							

This register contains the data for the I/O ports when they are used as ’General Purpose Outputs’ . The bits of the register are controlled by writing to the hex addresses at 0C and 1C. Ones written to the OPR drive the pins to 0; zeros drive the pins to 1. (The pins drive the value of the complement data written to the OPR)

IPR (Input Port Register cd) Reads I/O pins for DUART cd

I/O3d	I/O2d	I/O3c	I/O2c	I/O1d	I/O0d	I/O1c	I/O0c
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

IPR (Input Port Register ab) Reads I/O pins for DUART ab

I/O3b	I/O2b	I/O3a	I/O2a	I/O1b	I/O0b	I/O1a	I/O0a
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

This register reads the state of the ’I/O Ports’. The state of the I/O ports is read regardless of being programmed as inputs or outputs. The IPR can be thought of a just another 8 bit parallel port to the system data bus. The lower four bits of this register are replicated in the lower four bits of the IPCR register.

I/O Port Control Channel A (IOPCRA)

IOPCR[xx] Pin Control Bits	IOPCRa[7:6]	IOPCRa[5:4]	IOPCRa[3:2]	IOPCRa[1:0]
	I/O3A	I/O2A	I/O1A	I/O0A
00 = input	IPR(5), TxC in	IPR(4), RxC in	IPR(1), C/Tab Clk in$^{NO\ TAG}$ TxC in	IPR(0), CTSN
01 = output	OPRab(5)	OPRab(4) RTSN$^{NO\ TAG}$ if IOPCR[5:4] = 01	OPRab(1) RTSN$^{NO\ TAG}$ if IOPCR[5:4] ≠ 01	OPRab(0)

10 = output	TxC 16x	RxC 1x	RxC 16x	TxC 1x
11 = output	TxC 1x	RxC 16x	RxC 1x	TxC 16x

I/O Port Control Channel B (IOPCRB)

IOPCR[xx] Pin Control Bits	IOPCRb[7:6]	IOPCRb[5:4]	IOPCRb[3:2]	IOPCRb[1:0]
	I/O3B	I/O2B	I/O1B	I/O0B
00 = input	IPR(7), TxC in	IPR(6), RxC in	IPR(3), TxC in	IPR(2), CTSN
01 = output	OPRab(7)	OPRab(6) $RTSN^{NO\ TAG}$ if IOPCR[5:4] = 01	OPRab(3) $RTSN^{NO\ TAG}$ if IOPCR[5:4] ≠ 01	OPRab(2)
10 = output	TxC 16x	RxC 1x	C/T ab out	TxC 1x
11 = output	TxC 1x	RxC 16x	RxC 1x	TxC 16x

I/O Port Control Channel C (IOPCRC)

IOPCR[xx] Pin Control Bits	IOPCRc[7:6]	IOPCRc[5:4]	IOPCRc[3:2]	IOPCRc[1:0]
	I/O3C	I/O2C	I/O1C	I/O0C
00 = input	IPR(5), TxC in	IPR(4), RxC in	IPR(1), C/Tcd Clk $in^{NO\ TAG}$ TxC in	IPR(0), CTSN
01 = output	OPRcd(5)	OPRcd(4) $RTSN^{NO\ TAG}$ if IOPCR[5:4] = 01	OPRab(1) $RTSN^{NO\ TAG}$ if IOPCR[5:4] ≠ 01	OPRcd(0)
10 = output	TxC 16x	RxC 1x	RxC 16x	TxC 1x
11 = output	TxC 1x	RxC 16x	RxC 1x	TxC 16x

I/O Port Control Channel D (IOPCRD)

IOPCR[xx] Pin Control Bits	IOPCRd[7:6]	IOPCRd[5:4]	IOPCRd[3:2]	IOPCRd[1:0]
	I/O3D	I/O2D	I/O1D	I/O0D
00 = input	IPR(7), TxC in	IPR(6), RxC in	IPR(3), TxC in	IPR(2), CTSN
01 = output	OPRcd(7)	OPRcd(6) $RTSN^{NO\ TAG}$ if IOPCR[5:4] = 01	OPRcd(3) $RTSN^{NO\ TAG}$ if IOPCR[5:4] ≠ 01	OPRcd(2)
10 = output	TxC 16x	RxC 1x	C/T cd out	TxC 1x
11 = output	TxC 1x	RxC 16x	RxC 1x	TxC 16x

The input part of the I/O pins is always active. The programming of the IOPCR bits to 00 merely turns off the out drivers and places the pin at high impedance.

A read of the IPR register returns the value of the IPR bits as shown above. IPR(5) is at bit position 5 of the data bus. Note that the IPR bit positions do not follow the 0, 1, 2, 3 order of the I/O ports. During a read of the IPR the I/O ports are not latched. Therefore, it is possible to see changing data during the read. Port pins that have clocks on them may not yield valid data during the read.

Since the input circuits of the I/O ports are always active it is possible to direct the port signal back into the port. For example: I/O1 will output the RTS signal. Setting the Counter/Timer (C/T) to be clocked by the I/O1 port will result in the counter counting the number of times RTS goes active. The change of state detectors on I/O0 and I/O1 will, when programmed, always be sensitive to the signal on the port regardless of the source of that port's signal.

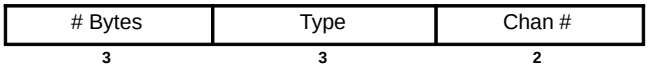
NOTES:

1. Normal configurations place RTSN output on I/O1 and place Tx external clock input on I/O3. For the 48 pin Dual In-Line package, I/O3 is not available. The following options allow flexible I/O programming with the 48 pin package:
   When IOPCR(7:6), the I/O3 control, ≠ 00, then I/O1 becomes available to the transmitter as an external clock.
   When IOPCR(5:4), the I/O2 control, = 01, then I/O2 may be the RTSN signal if MR1(7) = 1 and OPR(4) = 1.
2. I/O1 becomes RTSN when IOPCR(3:2) = 01 and MR1(7) = 1 and OPR(1) = 1. (OPR(3) for channel B)

Registers of the Interrupt System

The CIR, and "Global" registers are updated with the IACKN signal or from the "Update CIR" command at hex address 2A. These registers are not updated when IRQN is asserted since there could be a long time between the assertion of IRQN and the start of the interrupt service routine. (See notes following this section).

Current Interrupt Register (CIR)



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.

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 RxFIFO (GRxFIFO)

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 TxFIFO (GTxFIFO)

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.

C/Tab indicated by Channel code B 01
C/Tcd indicated by Channel code D 11

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 host MPU. The threshold field resets to 00.
The IVC field controls what kind of vector the QUART returns to the host MPU 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, Interrupt Type and Channel number
11 Disable generation of vector during IACK cycle. Returns hex'FF during an IACKN cycle.

The IVC field reset to 00. The ICR is read/write at address 2CH.

Bidding Control Registers (BCRs)

Received Break	State Change	C/T
3	3	2

This register is a transparent latch. It must be set to ensure the expected operation of the arbitration system. The 3 MSBs determine the priority of Received Break Interrupts; they are reset to 000.

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.
BCR Counter/Timer bits reset to 00.

Interrupt Vector (IVR)

The 8 bits of the interrupt vector

Interrupt Vector (IVR-Modified)

Always Used	with IVC = 0x	w/IVC = 01 or 10
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 contained in the CIR. The IVR is write only at address 29H.

DC ELECTRICAL CHARACTERISTICS[5, 6, 7]

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)	0 to +70°C	2.0			V
		–40 to +85°C	2.2			
V_{IH}	Input high voltage (X1/CLK)		$0.8V_{CC}$			V
V_{OL} V_{OH}	Output Low voltage Output High voltage (except OD outputs)	I_{OL} = 4.0mA I_{OH} = –400µA I_{OH} = –100µA	 $0.8V_{CC}$ $0.9V_{CC}$		0.4	V V V
I_{IL} I_{IH}	Input current Low, I/O ports Input current High, I/O ports	V_{IN} = 0 V_{IN} = V_{CC}	–10		 10	µA µA
I_I	Input leakage current	V_{IN} = 0 to V_{CC}	–1		1	µA
I_{ILX1} I_{IHX1}	X1/CLK input Low current X1/CLK input High current	V_{IN} = GND, X2 = open V_{IN} = V_{CC}, X2 = open	–100		 100	µA µA
I_{OZH} I_{OZL}	Output off current High, 3–state data bus Output off current Low, 3–state data bus	V_{IN} = V_{CC} V_{IN} = 0	–1		10 1	µA
I_{ODL} I_{ODH}	Open–drain output Low current in off state: IRQN Open–drain output Low current in off state: IRQN	V_{IN} = 0 V_{IN} = V_{CC}	–1		1	µA
I_{CC}	Power supply current Operating mode	TTL input levels 25°C with X1 = 4MHz			50	mA
	Power down mode*				5	mA

* See UART application note for power down currents less than 5µA.

AC ELECTRICAL CHARACTERISTICS[5, 6, 7, 8]

SYMBOL	FIGURE	PARAMETER	LIMITS			UNIT
			Min	Typ	Max	
Reset timing						
t_{RES}	8	Reset pulse width	200			ns
I/O Port timing						
t_{PS}	9	I/O input setup time before RDN Low	0			ns
t_{PH}	9	I/O input hold time after RDN High	0			ns
t_{PD}	9	I/O output valid from WRN High RDN Low			 110 110	 ns ns
Interrupt timing						
t_{IR}	10	IRQN negated or I/O output High from: Read RHR (RxRDY/FFULL interrupt) Write THR (TxRDY interrupt) Reset command (break change interrupt) Reset command (I/O 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}	11	X1/CLK low/high time	125/100			ns
t_{CLK}	11	X1/CLK low/high time (above 4MHz; X1/CLK ÷ 2 active)	56/56			ns
t_{CLK}	11	X1/CLK frequency	0[9]	3.6864	8.0	MHz
t_{CTC}	11	Counter/timer clock high or low time	60			ns
f_{CTC}	11	Counter/timer clock frequency	0[9]		8	MHz
t_{RX}	11	RxC high or low time	30			ns
f_{RX}	11	RxC frequency (16X) RxC frequency (1X)	0[9] 0[9]		16 1.0	MHz MHz
t_{TX}	11	TxC high or low time	30			ns
f_{TX}	11	TxC frequency (16X) TxC frequency (1X)	0[9] 0[9]		16 1.0	MHz MHz
Transmitter timing						
t_{TXD}	12	TxD output delay from TxC low			120	ns
t_{TCS}	12	TxC output delay from TxD output data	–20		+20	ns
Receiver timing						
t_{RXS}	13	RxD data setup time to RxC high	100			ns
t_{RXH}	13	RxD data hold time from RxC high	100			ns

NOTES:

1. Stress above these listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other condition above those indicated in the operation section of the 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.
5. 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.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for interrupt and I/O outputs: C_L = 50pF, R_L = 2.7kΩ to V_{CC}. Test conditions for rest of outputs: C_L = 150pF.
8. 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.
9. This value is not tested, but is guaranteed by design. For t_{CLK} minimum test rate is 2.0MHz.

AC ELECTRICAL CHARACTERISTICS[4]

T_A = 25°C; V_{CC} = 5V ± 10%, unless otherwise specified. Limits shown as nn/nn refer to Commercial/Industrial temperature range. Single numbers apply to both ranges.

NO.	FIGURE	CHARACTERISTIC	LIMITS			UNIT
			Min	Typ	Max	
1	2	Setup: A[5:0] valid to CEN Low	10			ns
2	2	Hold: A[5:0] valid after CEN Low[6]	45			ns
3	2	Access: Later of CEN Low and RDN Low, to Dnn valid (read)			110/115	ns
4	2	Later of CEN Low and (RDN or WRN as applicable) Low, to DACKN Low Normal Operation: From Power Down:	10 + 2 X1 edges[5]		90/122 + 3 X1 edges[5] 150	ns
5	2	Earlier of CEN High or RDN High, to Dnn released (read)[1]	0		30	ns
6	2	Earlier of CEN High or (RDN or WRN as applicable) High, to DACKN released	0		30	ns
7	2	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	50			ns
8	2	Setup, Dnn valid (write) to later of CEN Low and WRN Low[2]	–30			ns
9	2	Later of CEN Low and WRN Low, to earlier of CEN High or WRN High	110/115			ns
10	2	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 I/O outputs: C_L = 50pF, forced current for V_{OL} = 5.3mA; forced current for V_{OH} = 400µA, RL = 2.7kΩ to V_{CC}. Test condition for rest of outputs: C_L = 150pF
5. Consecutive write operations to the upper four bits of the Command Register (CR) require at least three X1/CLK edges; four X1/CLK edges in the 'X1/CLK divide by 2 edges' according to register 2E or 2F setting.
6. Address is latched at leading edge of a read or write cycle.

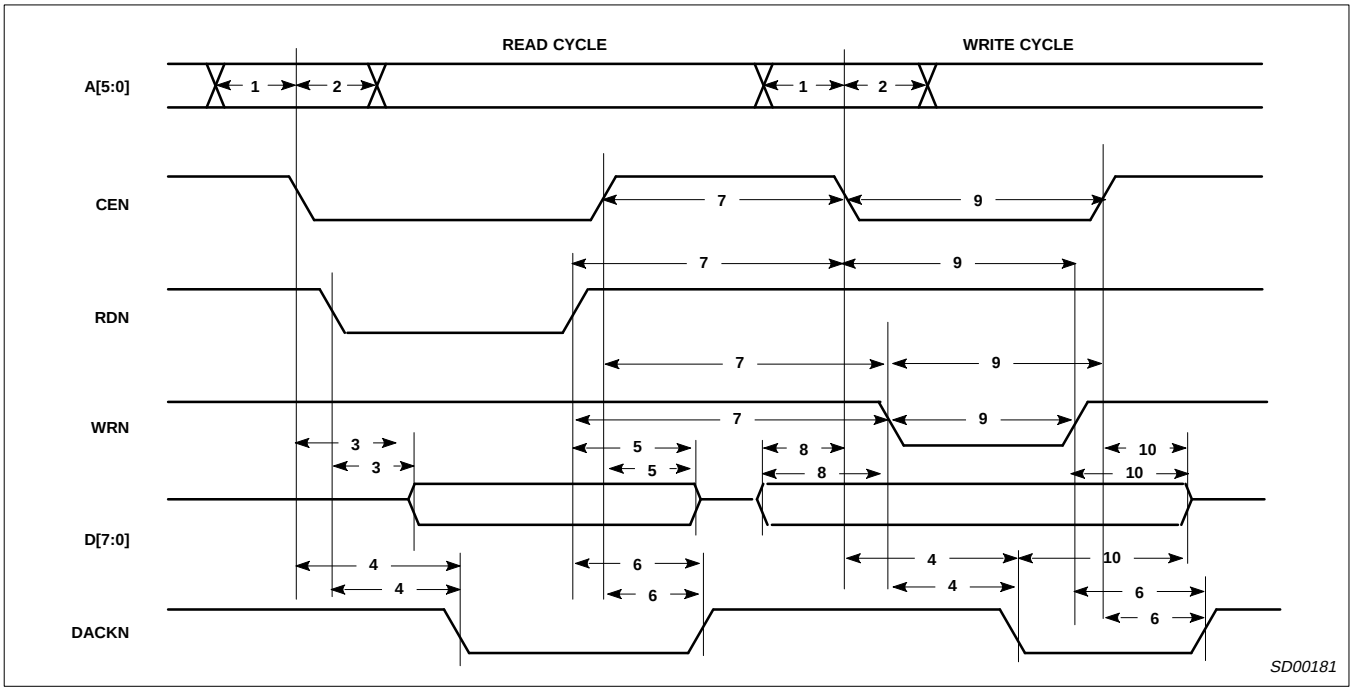


Figure 2. A Read Cycle Followed by a Write Cycle with DACKN

AC ELECTRICAL CHARACTERISTICS T_A = 25°C; V_{CC} = 5V ± 10%, unless otherwise specified. Limits shown as nn/nn refer to Commercial/Industrial temperature range. Single numbers apply to both ranges.

NO.	FIGURE	CHARACTERISTIC	LIMITS			UNIT
			Min	Typ	Max	
1	3	D[7:0] Valid after IACKN Low			110/115	ns
2	3	DACKN Low after IACKN Low	10 + 2 X1 edges[1]		90/122 + 3 X1 edges[1]	ns
3	3	D[7:0] floating after IACKN High	0		30	ns
4	3	DACKN High after IACKN High	0		30	ns
5	3	IACKN High after IACKN Low	110/115			ns

NOTE:
1. Consecutive write operations to the upper four bits of the Command Register (CR) require at least three X1/CLK edges; four X1/CLK edges in the 'X1/CLK divide by 2 edges' according to register 2E or 2F setting.

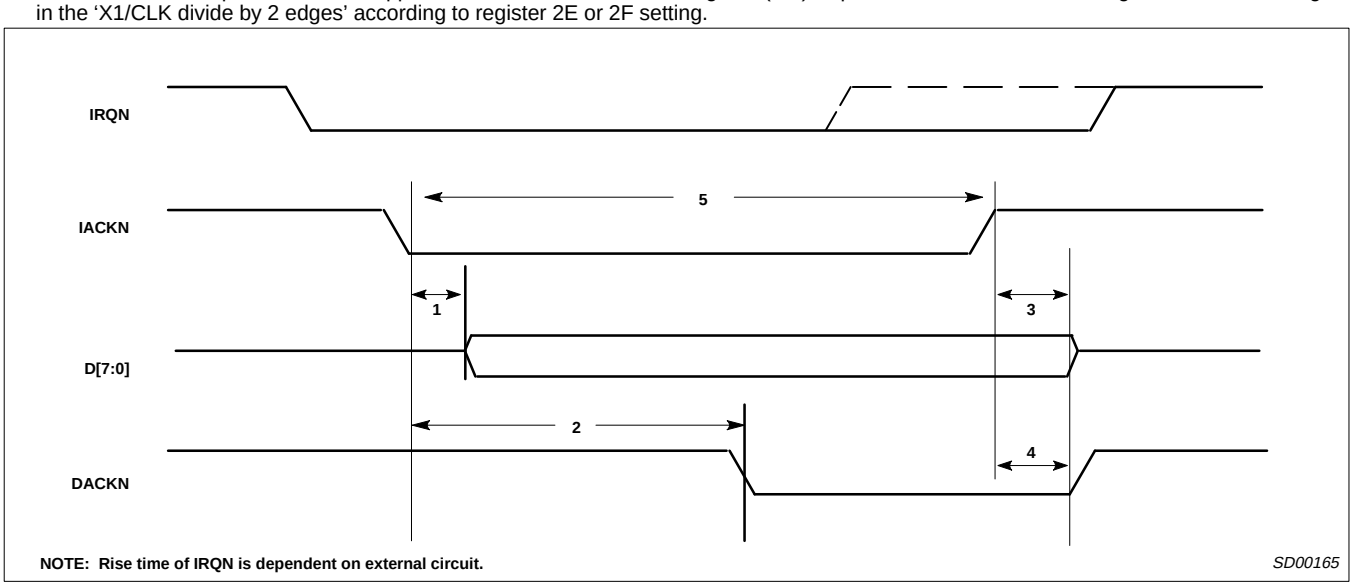


Figure 3. Interrupt Knowledge (IACKN) Timing

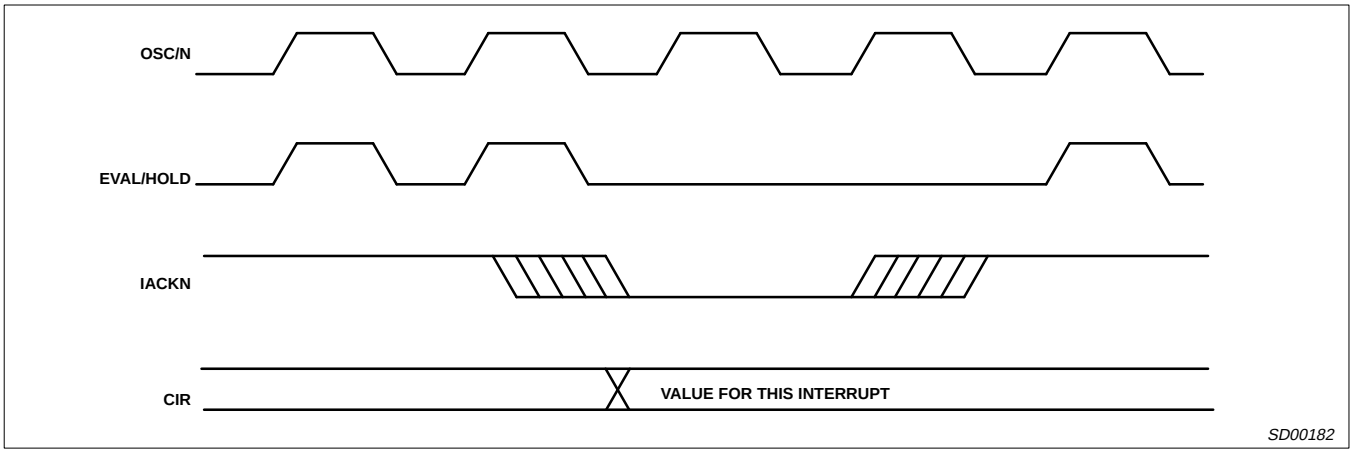


Figure 4. Interrupt Bid Arbitration Timing

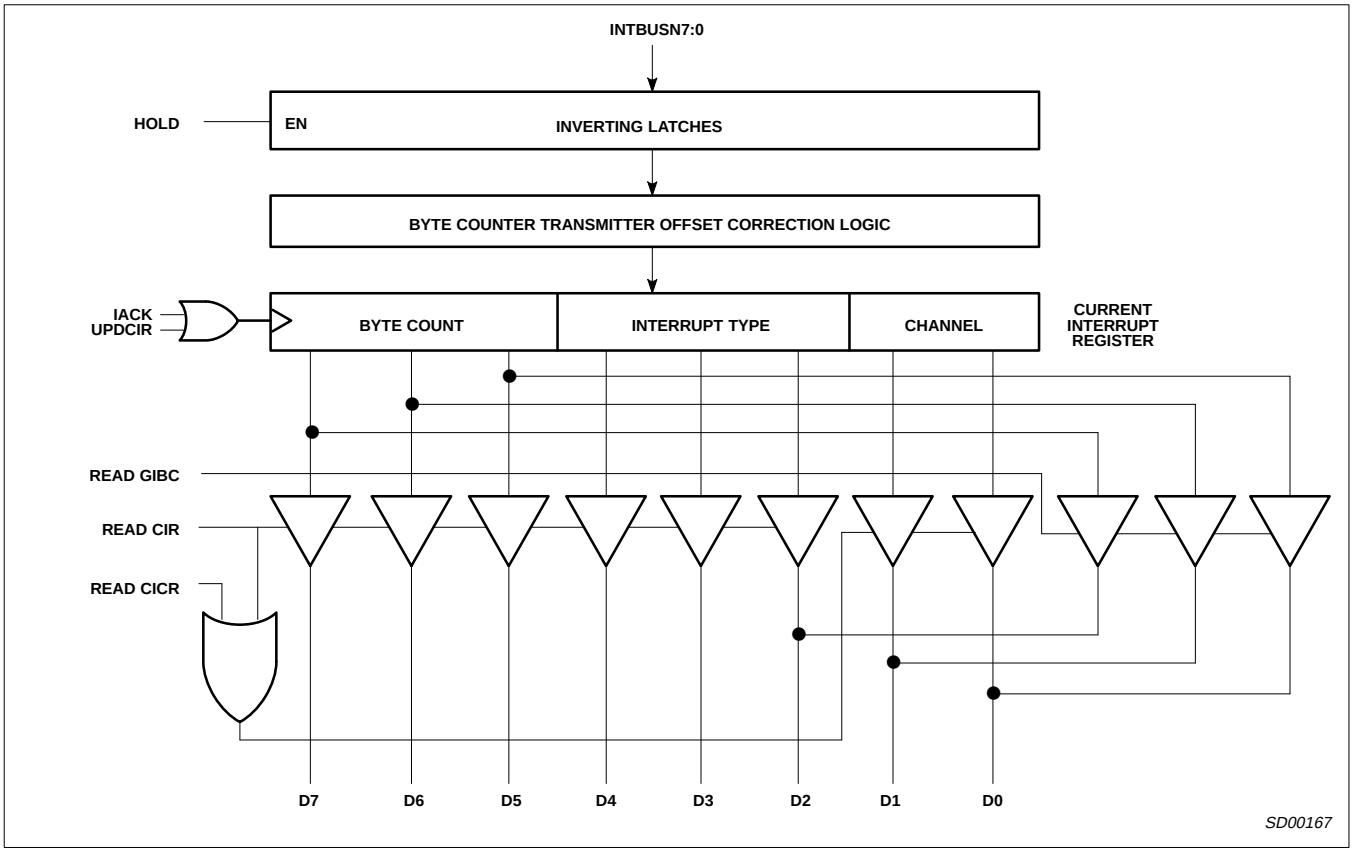


Figure 5. Current Interrupt Register Logic

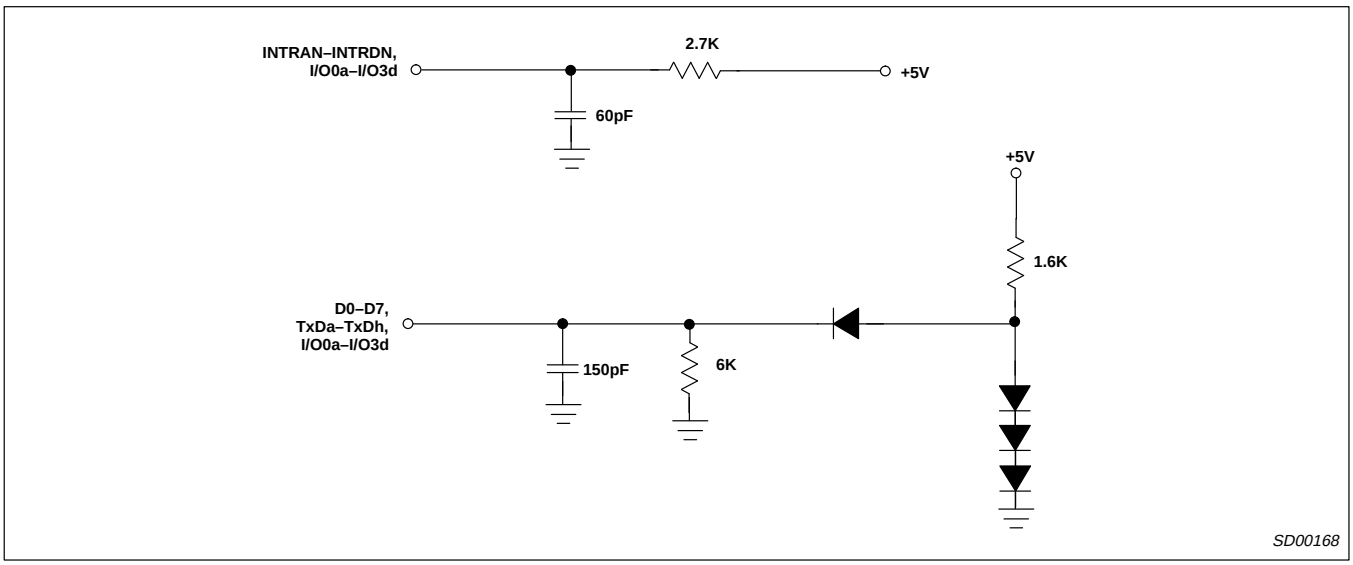


Figure 6. Test Conditions on Outputs

Figure 7. Reset Timing

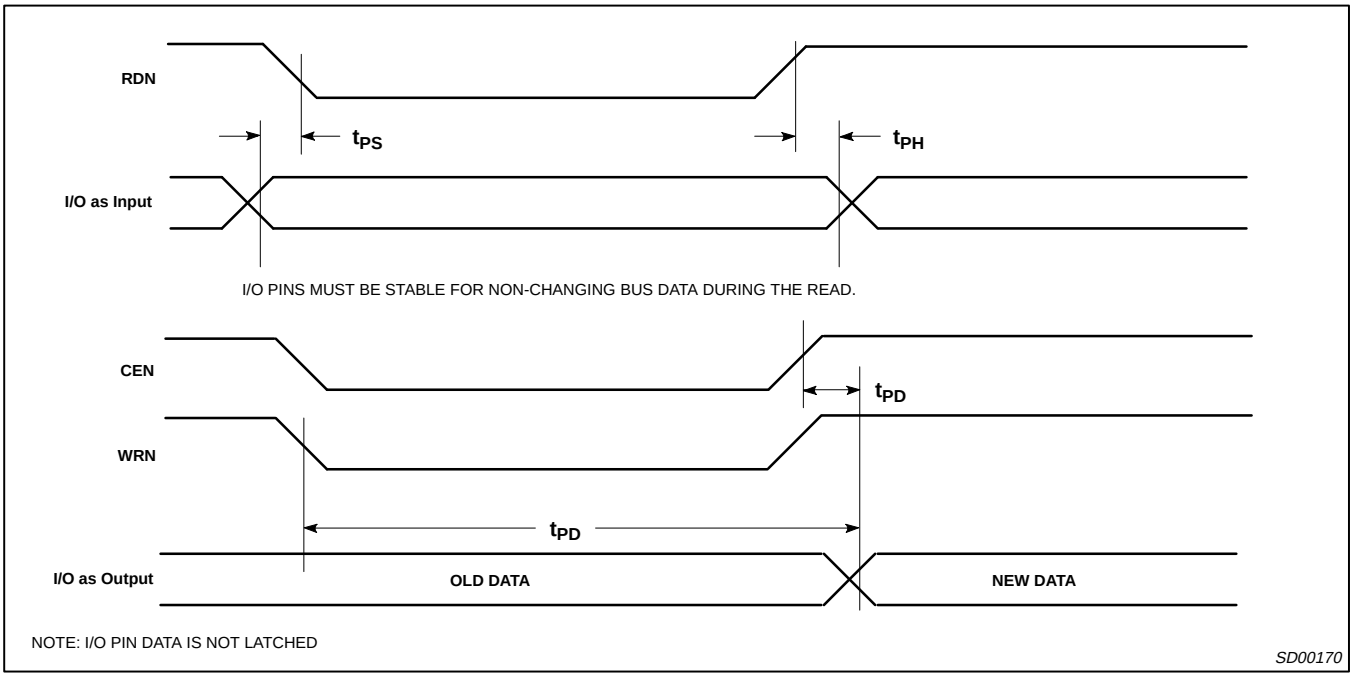
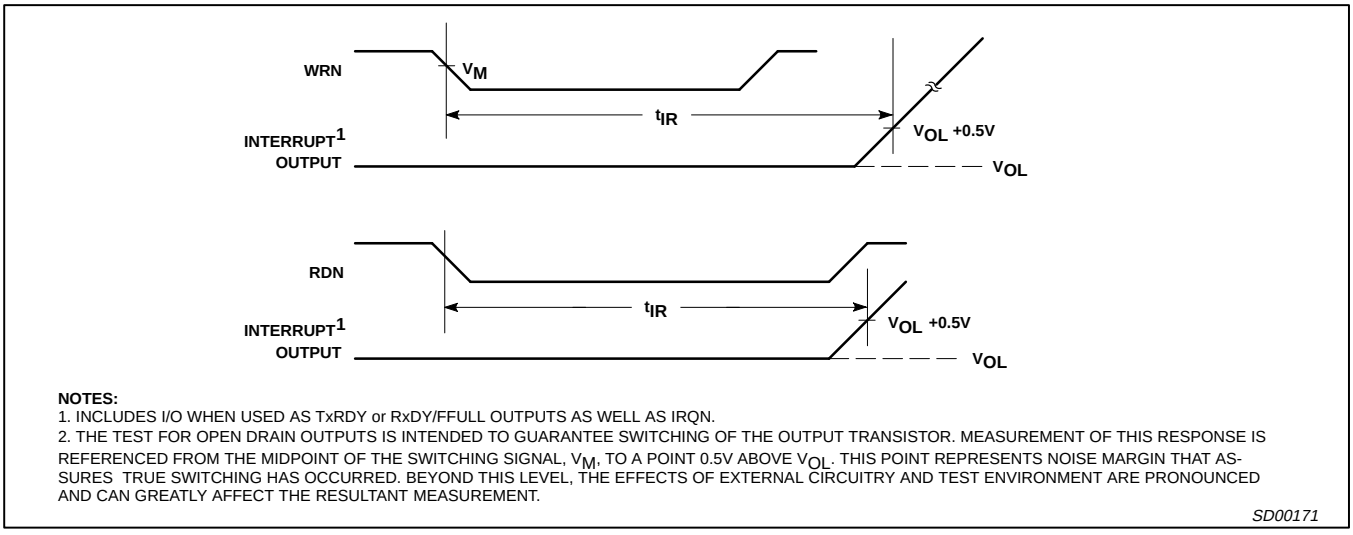


Figure 8. I/O Port Timing



NOTES:
1. INCLUDES I/O WHEN USED AS TxRDY or RxDY/FFULL OUTPUTS AS WELL AS IRQN.
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 9. Interrupt Timing

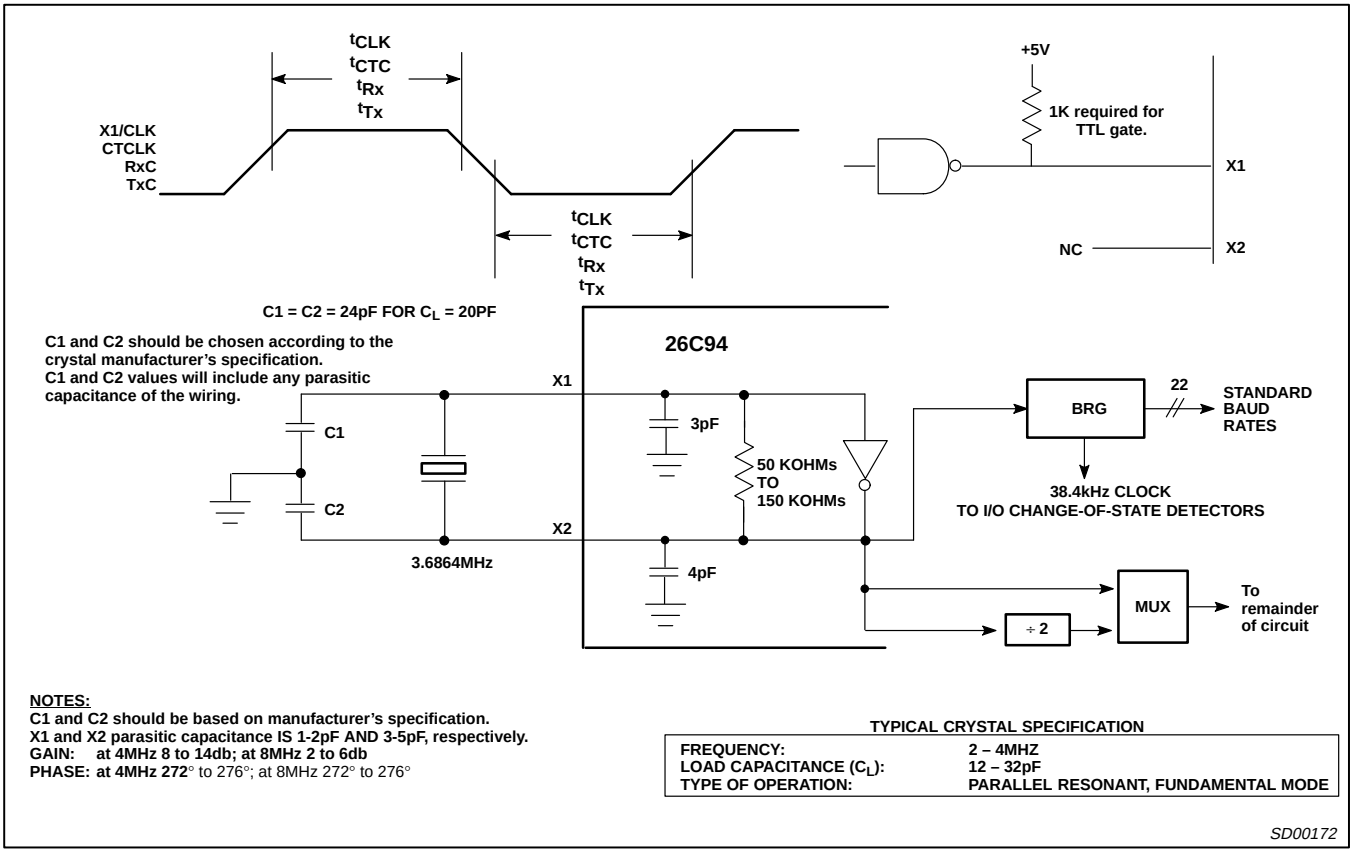


Figure 10. Clock Timing

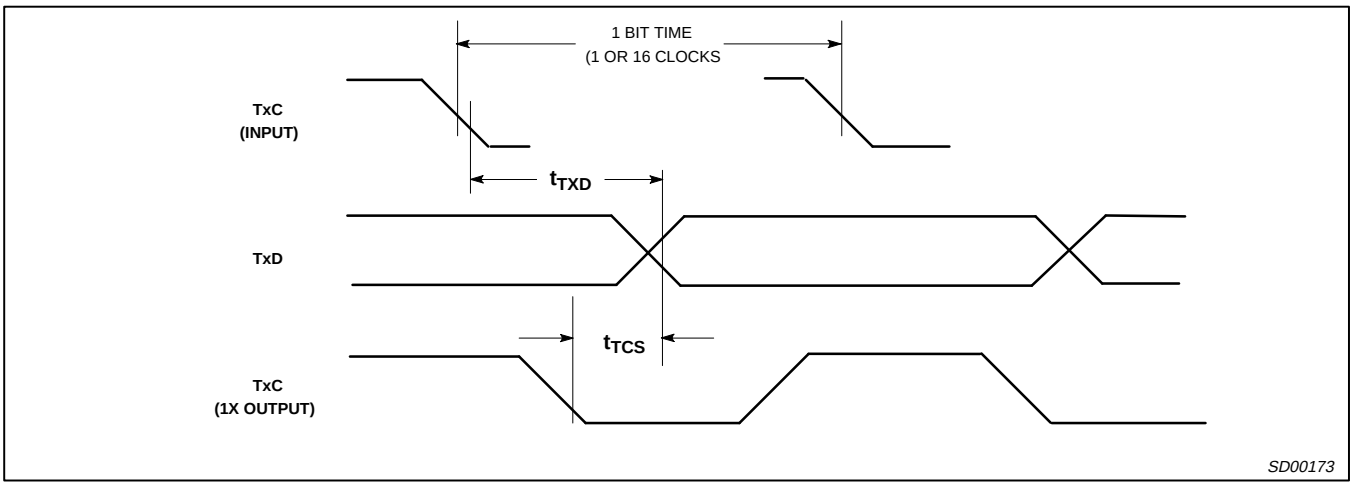


Figure 11. Transit Clock Timng

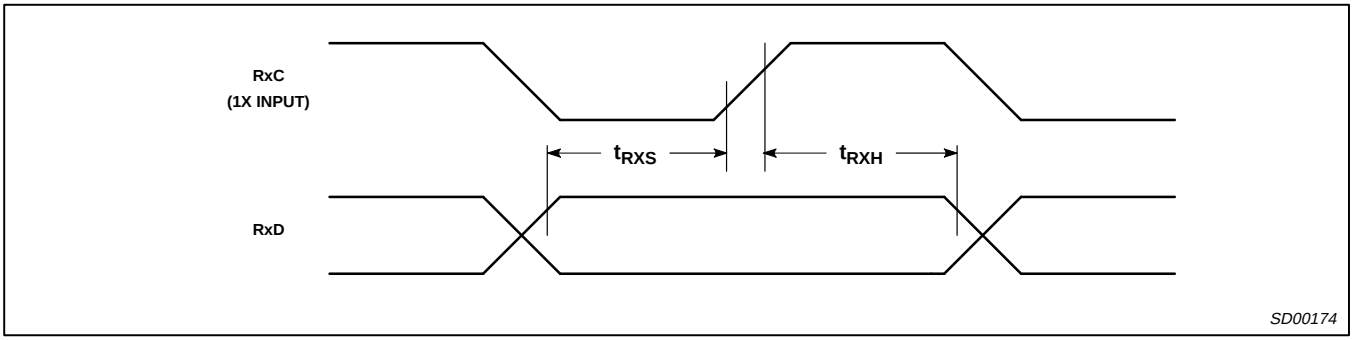


Figure 12. Receive Clock Timing

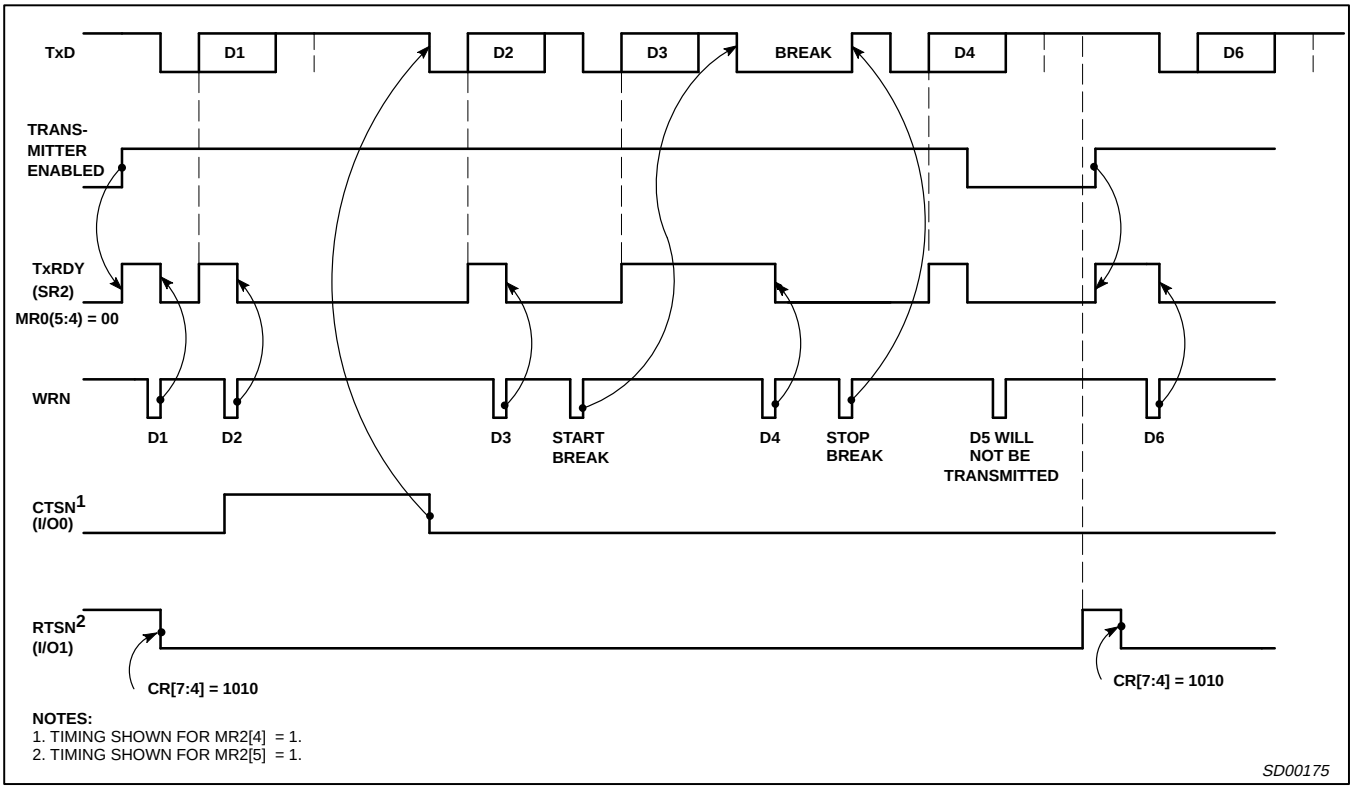


Figure 13. Transmitter Data Timing

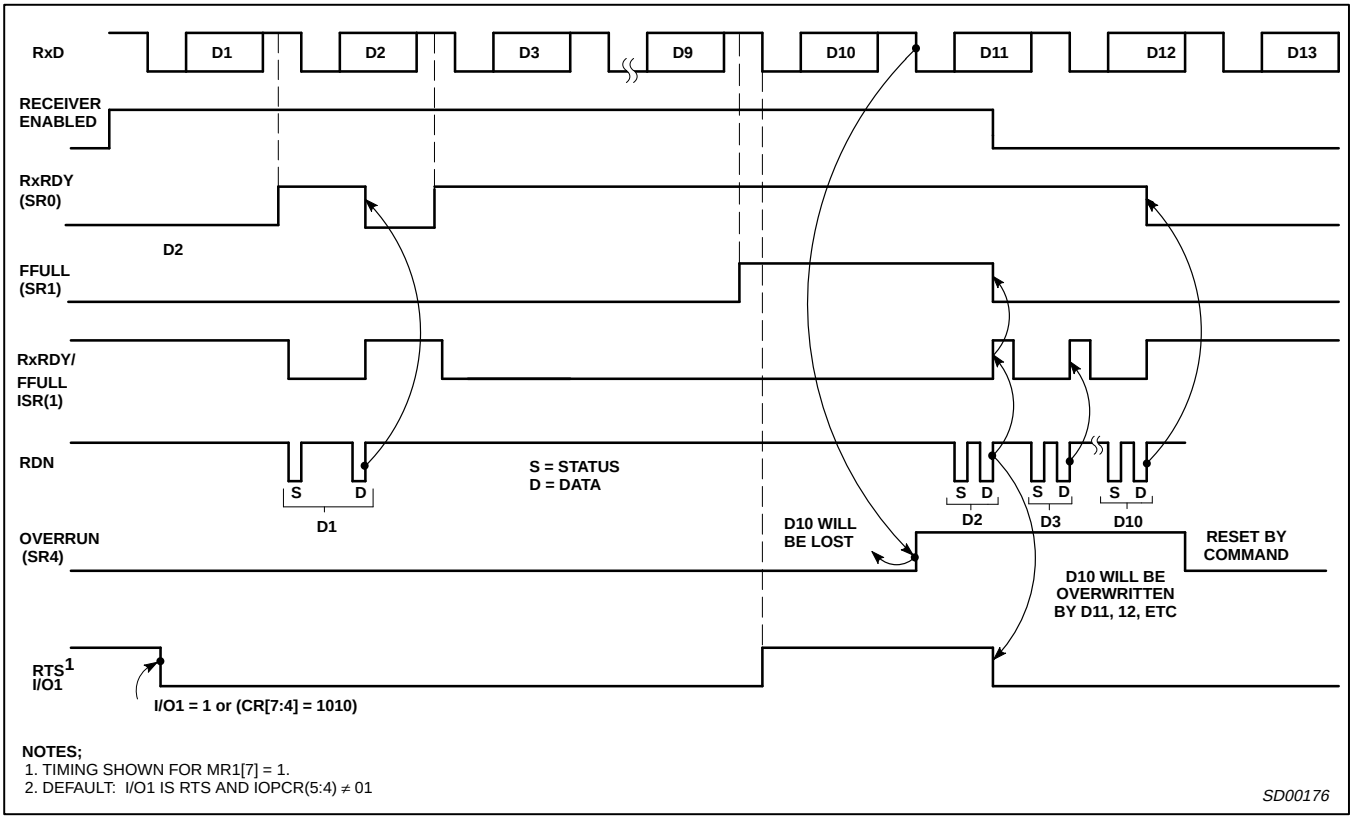


Figure 14. Receiver Data Timing

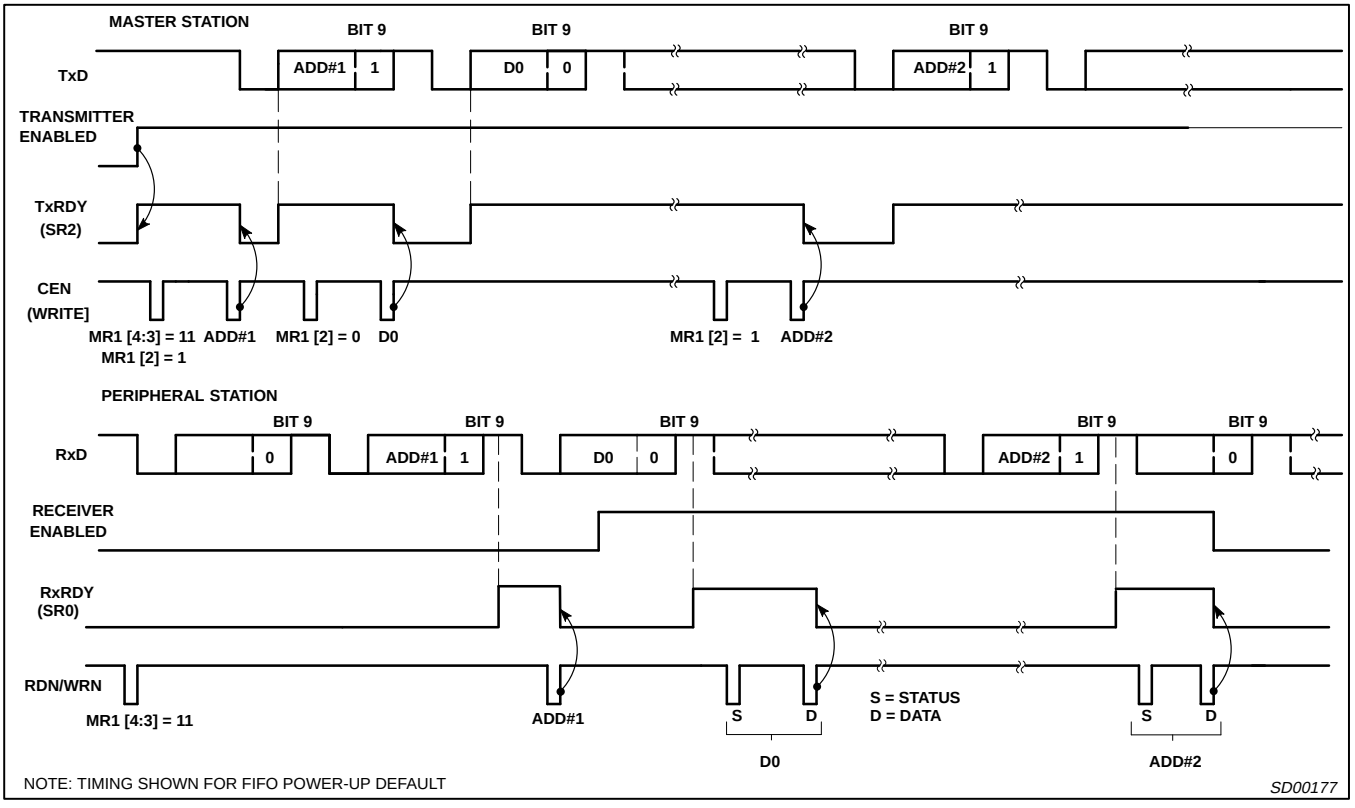


Figure 15. Wake-Up Mode

INTERRUPT NOTES

The following is a brief description of the new QUART “Bidding” interrupt system, interrupt vector and the use of the Global registers.

The new features of the QUARTs have been developed to greatly reduce the microprocessor time required to service uart interrupts. Bus cycle times have also been enhanced. By use of the new Current Interrupt Register (CIR) the speed of a polled system is also improved. For example programming the SCC2692 to interrupt on TxRDY and RxFUL would generate four interrupts for every six characters processed along with at least two additional accesses to the chip for each interrupt. This amounts to two non–data chip accesses per character. In the 68C94 this has been reduced to 0.25 non data accesses per character; an eight fold improvement. In certain conditions use of the global registers will yield a greater improvement.

The QUART has 18 possible sources which can be programmed to generate an interrupt:

- 4 Receiver channels
- 4 Transmitter Channels
- 4 Received ”Break” conditions
- 4 Change of State Detectors (a total of 8 ports)
- 2 Counter/Timers

These sources are encoded in such a way that they generate a unique value. This value is defined by chip hardwire programming, user programming, and the source’s present condition. The values the sources generate are compared (at the X1 clock rate) to a user defined Interrupt Threshold value contained in the ICR (Interrupt Control Register). When the source’s value exceeds the threshold the interrupt is generated. It is the source’s value which is captured in the CIR.

The heart of the interrupt speed enhancement is attained by allowing the interrupting source to encode its channel, interrupt type and, if appropriate, the number of FIFO bytes requiring service. This information is coded and transferred the CIR (Current Interrupt Register) at the time IACKN is asserted or the command ’Update CIR’ is executed. Upon an interrupt the processor may read this register and in one access determine the ”who, what and how much”. This CIR value is used to drive the interrupt vector modification (when used) and the new ”Global” registers.

“Global” Registers

The ”Global Registers” are effectively pointers which use the contents of the CIR to direct a read or write operation to Rx or Tx or other source which is currently interrupting. There are four global registers defined in the register map:

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

The global receive and transmit registers operate as an indirect address. The data read from the global receive register will be that of the currently interrupting receiver; the data written to the global transmit register will go to the currently interrupting transmitter. The interesting point here is that under certain circumstances an interrupt can be serviced without an interrogation of the chip.

For completeness it should be noted that the global registers are not physical devices. Reads of the Global Byte and Channel registers give the Byte count or Channel number, respectively, (right justified) of the interrupting channel. The CIR data is mapped to these ”registers”.

In these identifiers the receivers are biased to have highest priority. The identifier bits and the channel number bits are hardwired on the chip. Normally the non–data interrupts would be programmed to a low value. The programmable fields can, in some cases, make these sources higher than a full receiver.

It would seem that a 11 programmed in the upper counter/timer bits, for example, would cause it to interrupt nearly all the time. This is not true . A counter/timer that has not timed out will not bid. In a similar fashion a receiver FIFO that is empty or a transmitter FIFO that is full will not bid

In general terms the threshold value programmed in the ICR (Interrupt Control Register) will reflect some fill level of the eight character transmit and receive FIFOs that allow processor service without underrun or overrun occurring.

Table 9. “Bidding Format”

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION
Rx Byte count			Error 1	1	1	Channel No.		Receiver bid With error
Rx Byte count			no Error 0	1	1	Channel No.		Receiver bid No error
0	Tx Byte Count			1	0	Channel No.		Transmit bid
Programmable			1	0	0	Channel No.		Receive Break
Programmable			0	0	1	Channel No.		Change of State
Programmable		0	1	0	1	Channel No.		Counter/Timer

NOTES:

1. The ones and zeros above represent the hardwired positions.
2. Note the format of bits 4:2. They represent the identity of the interrupting source.
3. Bids with the highest number of contiguous MSBs win the bid.
   - 1 1 1 Receiver with error
   - 0 1 1 Receiver without error
   - x 1 0 Transmitter
   - 1 0 0 Receiver Break detect
   - 0 0 1 Change of State
   - 1 0 1 Counter/Timer
   - 0 0 0 No interrupt

Note that interrupt threshold value in the ICR is 6 bits long. This value is aligned with the bid arbitration logic such that it bids only through the most significant 6 bits. The result of this is that the channel value does not 'bid'. However the logic is such that other parts of the bid being equal the condition of the highest channel will be captured in CIR. The increasing order of the channels is A, B, C, D. Thus channel D is the "strongest" of the four.

It could be that the giving the highest strength to channel D may, from time to time, not be what would be most desired. Further it may be desired to alter the authority of a channel's bid. This may be done by setting the Rx and/or Tx interrupt bits in MR0 and MR1 to values different than zero. This will have the effect of not allowing the associated receiver or transmitter to bid until its FIFO reaches a particular fill level. Although this compromises the idea of the bidding interrupt scheme, it is entirely safe to use. In fact it is setting of MR0 and MR1 interrupt bits to zero that causes the receiver to stop bidding when it is empty and causes the transmitter to stop bidding when it is full. Altering the MR0 and MR1 interrupt bits only changes the level at which the Rx & Tx bidding is stopped.

See the "Interrupt Note on 68C94" which refers to the use of the MR registers in controlling the Rx and Tx bidding.

In normal operation the character of an interrupt will be controlled by the above registers in conjunction with the IMR (Interrupt Mask Register (one for each DUART)) . The function of the IMR will be to enable bidding of any particular source. Recall that the QUART has 18 functions which may generate an interrupt.

The format of the interrupt vector is controlled by the ICR[1:0] bits. The formats are shown in Table 7. The purpose of the vector modification is to allow the interrupting source (either channel or type and channel) to direct the processor to appropriate service routine. We have found that some users wish to use extremely tight loops for the service routines and find the addition of several tests of status bytes to be very 'expensive' in processor time.

Table 10. Configuration of Interrupt Vector for the QUART

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Interrupt vector for → ICR[1:0]=00	INTERRUPT VECTOR FORMATS (Controlled by ICR[1:0])							
	IVR[7:0]							
	Full interrupt vector							
Interrupt vector for → ICR[1:0]=01								
	IVR[7:2]						ICR[1:0]	
	Interrupt vector 6 MSBs						Channel number	
Interrupt vector for → ICR[1:0]=10								
	IVR[7:5]			ICR[4:2]			ICR[1:0]	
	Interrupt vector 3 MSBs			Interrupt type			Channel number	
Interrupt vector for → ICR[1:0]=11 (Inhibit)								
	Inhibit vector output. (Set bus to FFh)							
	CURRENT INTERRUPT REGISTER FORMAT CIR[7:0]							
	Rx or Tx byte count			Interrupt type: R/Tx CT COS BRK			Channel number	
	INTERRUPT CONTROL REGISTER FORMAT ICR[0:7]							
	Interrupt threshold ICR[7:2]						Interrupt vector format ICR[1:0]	

NOTE ON QUART INTERFACE TO ITS CONTROLLING PROCESSOR

The QUART, has been designed to interface in either the synchronous interrupt environment (without DACKN) or the asynchronous interrupt environment (with DACKN). The 80xxx devices of Intel design are usually operated in a synchronous interrupt mode while those of Motorola design, 68xxx devices, operate in an asynchronous interrupt mode.

Note: Synchronous and asynchronous interrupt modes are not in any way associated with synchronous or asynchronous data transmission.

The QUART has been designed with the pins required to service either interface. In general then it is probable that in any application some of the interface pins will not be used. This note discusses what is required for the "text book" connections of the two methods. It should be noted that features of either method are not mutually exclusive.

The interface pins are all active low. (at V_{SS} or ground) The pins used for normal reading and writing to the QUART (the generation of a bus cycle) are CEN (Chip Enable), RDN (Read Enable), WRN (Write Enable). The pins used in the interrupt service are IRQN (Interrupt Request), IACKN (Interrupt Acknowledge). The pin used for data transfer is DACKN (Data Acknowledge). IRQN and DACKN are open drain outputs.

DACKN signaling can be enabled or disabled via writing to address 27h or 26h respectively. Note that if DACKN is enabled that writing to the QUART will occur on the falling edge of DACKN. The use of hardware reset (required at power up) enables DACKN.

The Asynchronous Interface

Those familiar with 68xxx I/O will note the use of the two pins RDN and WRN to be in conflict with 68xxx devices use of the one R/WN pin. The R/WN must be inverted such that the R/WN may drive the WRN input while the inversion of R/WN drives the RDN input. It is good practice to condition the inversion of R/WN such that RDN will not become active on the termination of a write to the QUART while CEN is still asserted. These short periods of read could upset FIFO pointers in the chip.

During a read of the QUART DACKN signals that valid data is on the data bus. During a write to the QUART DACKN signals that data placed on the bus by the control processor has been written to the

addressed register. The generation of DACKN begins with the start of a bus cycle (Read, Write or Interrupt Acknowledge) and then requires two edges of the X1 clock plus typically 70ns for its assertion.

In this mode the writing of data to the QUART registers occurs on the falling edge of DACKN or the rising edge of the combination of CEN and WRN which ever occurs first. This requires that the data to be written to the QUART registers be valid with respect to the leading edge of the combination of CEN and WRN. (In the synchronous mode it is the trailing edge)

IACKN updates the CIR (Current Interrupt Register) and places the Interrupt Vector or Modified Interrupt Vector on the bus if the Interrupt Vector is used.

The Synchronous Interface

In this mode the DACKN and IACKN are usually not used. Here data is written to the QUART on the trailing edge of the combination of CEN and WRN. The placing of data on the bus during a read cycle begins with the leading edge of the combination of CEN and RDN.

The read cycle will terminate with the rise of CEN or RDN which ever one occurs first. In this mode bus cycles are usually setup to be the minimum time required by the QUART and hence will be faster than bus cycles that are defined by the DACKN signal. DACKN should be turned off in this mode.

When IACKN is not used or is not available the command at 2Ah should be used to update the CIR (Current Interrupt Register). This register is normally updated by IACKN in response to the IRQN. Note that the CIR is not updated by IRQN since there could be a long time between the assertion of IRQN and the start of the interrupt service routine. During this time it is quite possible that another interrupt with a higher priority occurs. It is the CIR that contains the information that describes the interrupt source and its priority. It is therefor recommended that the first operation upon entering the interrupt service routine is the updating of the CIR. (Recall that the contents of the GLOBAL registers reflect the content of the CIR)

Summary

In the asynchronous mode all of the interface pins are usually used. The synchronous mode usually will not use the IACKN and DACKN. However there is no conflict in the quart if both modes are used in the same application. (i.e. More than one device may control the QUART) The principles to keep in mind are:

1. When IACKN is not used the CIR should be updated via command.
2. If DACKN is not used it should be disabled.
3. When in the asynchronous mode be sure DACKN is enabled.
4. With 68xxx type controllers the RDN signal must be generated.