SC28L201

3.3 V, 5 V UART, 3.125 Mbit/s, with 256-byte FIFO

Rev. 01 — 31 October 2005 Product data sheet

1. General description

The SC28L201 is a high performance UART. Its functional and programming features closely match but greatly extend those of previous Philips UARTs. Its configuration on power-up is similar that of the SC26C92. Its differences from the previous Philips UARTs are: 256-character receiver, 256-character transmit FIFOs, 3.3 V and 5 V compatibility, 8 I/O ports for arbitrating interrupt system and overall faster bus and data speeds and is fabricated in an advanced 0.5 micron CMOS process.

It is a member of the IMPACT line of data communications parts.

Pin programming will allow the device to operate with either the Motorola or Intel bus interface by changing the function of some pins (reset is inverted, DACKN, and IACKN enabled, for example).

The Philips Semiconductors SC28L201 Universal Asynchronous Receiver/Transmitter (UART) is a single-chip CMOS-LSI communications device that provides a full-duplex asynchronous receiver/transmitter channel in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system. The use of the Interrupt system provides intelligent interrupt vectors.

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

Each receiver and transmitter is buffered by 256-character FIFOs to nearly eliminate the potential of receiver overrun, transmitter underrun and to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability (Xon/Xoff and RTS/CTS) is provided to disable a remote transmitter when the receiver buffer is full.

Also provided on the SC28L201 is a multipurpose 8-bit I/O for the channel. These can be used as general-purpose I/O ports or can be assigned specific functions (such as clock inputs or status and interrupt outputs) under program control. Normally they will be used for modem control and DMA interface. All ports have change of state detectors and input sections are always active making output signals available to the internal circuits and the control processor.

The SC28L201 is available in a TSSOP48 package. For other package options, contact Philips.

PHILIPS

2. Features

- Member of IMPACT family: 3.3 V or 5.0 V, –40 °C to +85 °C and 80xxx or 68000 bus interface (I/M modes)
- Bit-by-bit real time transmission error check for high data integrity systems
- Full-duplex independent asynchronous receiver/transmitter
- 256 character FIFOs for receiver and transmitter
- Powers up to 9600 baud, 8 bits, no parity, 1 stop bit, interrupt disabled, all I/O set to input
- Pin programming to 68000 or 80xxx bus interface
- Three character recognition system, used as:
  - General purpose character recognition
  - Xon/Xoff character recognition
  - Address recognition Wake-up (multi-drop or 9-bit) mode
  - System provides 4 levels of automation on a recognition event
- Programmable data format
  - 5 to 8 data bits plus parity and 9-bit mode
  - Odd, even, no parity, or force parity
  - $^{9}/_{16}$, 1, 1.5 or 2 stop bits
- 16-bit programmable Counter/Timer
- Programmable baud rate for receiver and transmitter selectable from:
  - 27 fixed rates: 50 Bd to 2.0 MBd (includes MIDI rate)
  - Other baud rates via external clocks and C/T
  - Programmable user-defined rates derived from a programmable Counter/Timer
  - External 1× or 16× clock
- Parity, framing, and overrun error detection
- Line break detection and generation; false start bit detection
- Programmable channel mode
  - Normal (full-duplex)
  - Automatic echo
  - Local loopback
  - Remote loopback
  - Multi-drop mode (also called ‘Wake-up’ or ‘9-bit’)
- Multifunction 13-bit I/O input port
  - Can serve as clock or control inputs
  - Change-of-state detection on eight inputs
  - Inputs have typically > 100 MΩ pull-up resistors
  - Modem and DMA interface
- Versatile arbitrating interrupt system
  - Interrupt system totally supports single query polling
  - Output port can be configured to provide a total of up to six separate interrupt type outputs that may be wire-ORed (switched to open-drain)
  - Each FIFO can be independently programmed for any of 256 interrupt levels
  - Watchdog timer for receiver
- Maximum data transfer rates: 1× clock = 3 Mbit/s; 16× clock = 3.125 Mbit/s

- Automatic Wake-up mode for multi-drop applications
- Start-end break interrupt/status
- Detects break which originates in the middle of a character
- On-chip crystal oscillator
- Power-down mode at less than 10 μA
- Receiver Time-out mode
- Single +3.3 V ± 10 % or +5 V ± 10 % power supply

3. Ordering information

Table 1: Ordering information

V_{DD} = +3.3 V ± 10 % or V_{DD} = +5.0 V ± 10 %; T_{amb} = −40 °C to +85 °C

Type number	Package		
	Name	Description	Version
SC28L201A1DGG	TSSOP48	plastic thin shrink small outline package; 48 leads; body width 6.1 mm	SOT362-1

4. Block diagram

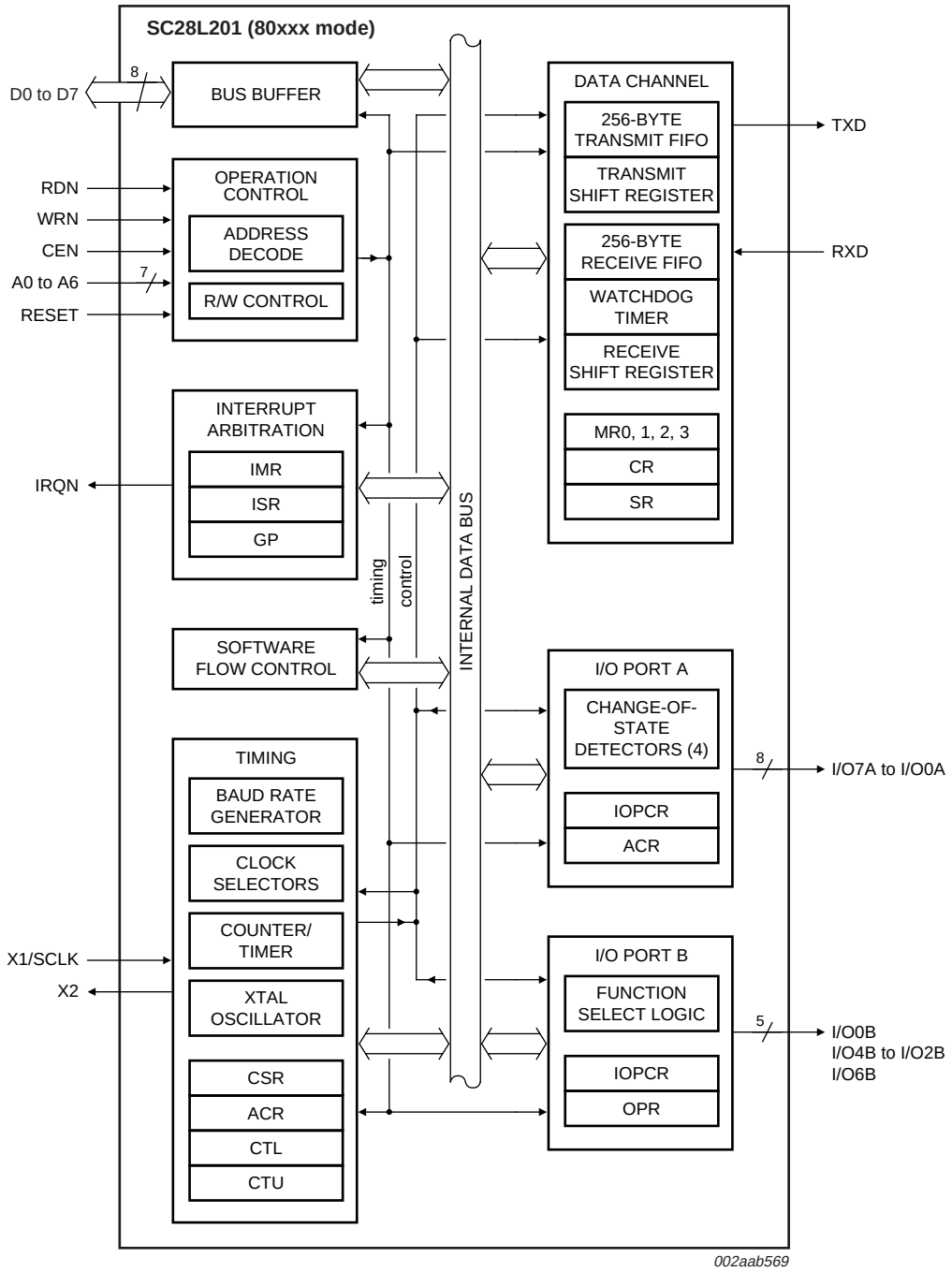


Fig 1. Block diagram of SC28L201 (80xxx mode)

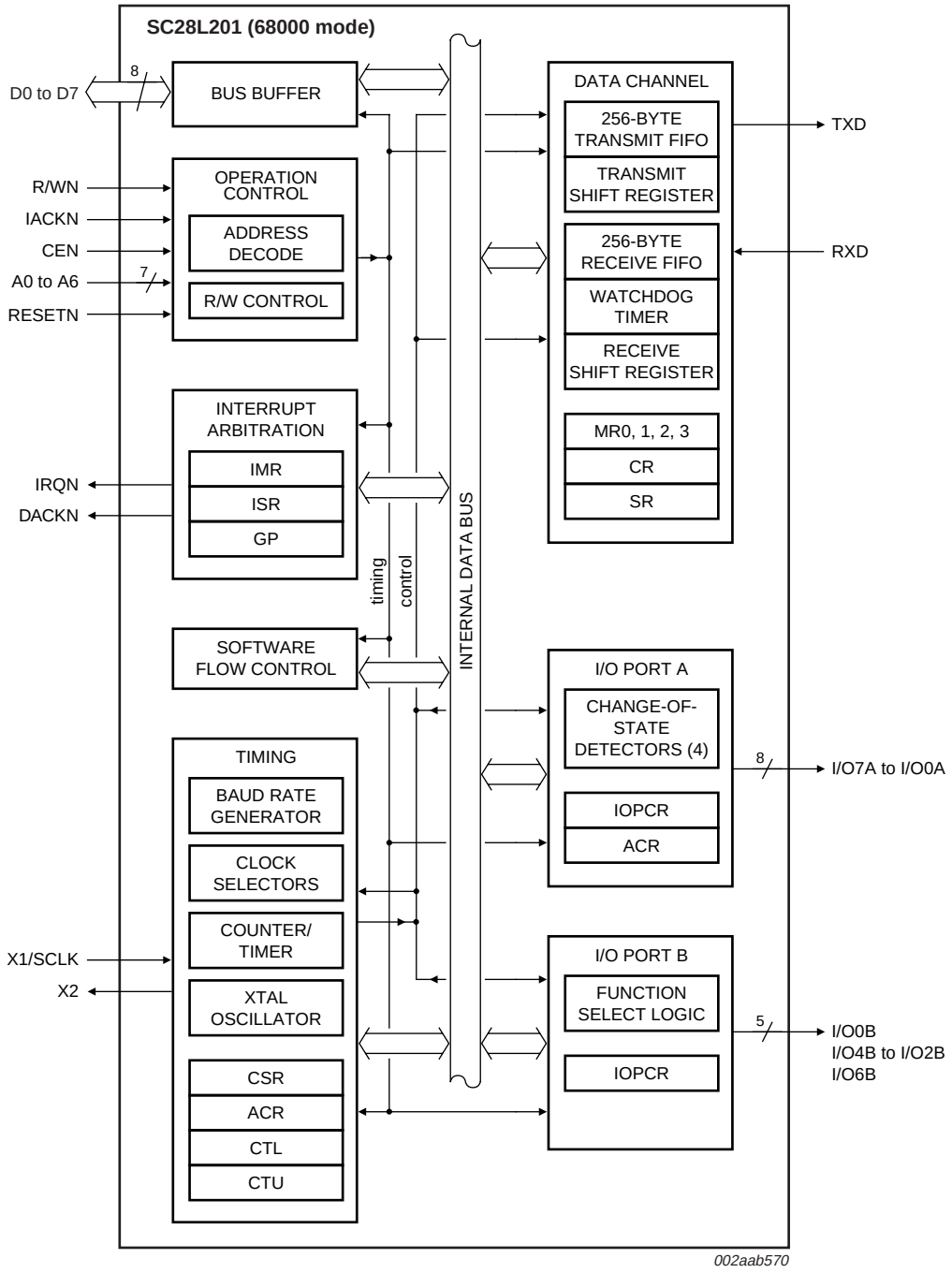


Fig 2. Block diagram of SC28L201 (68000 mode)

5. Pinning information

5.1 Pinning

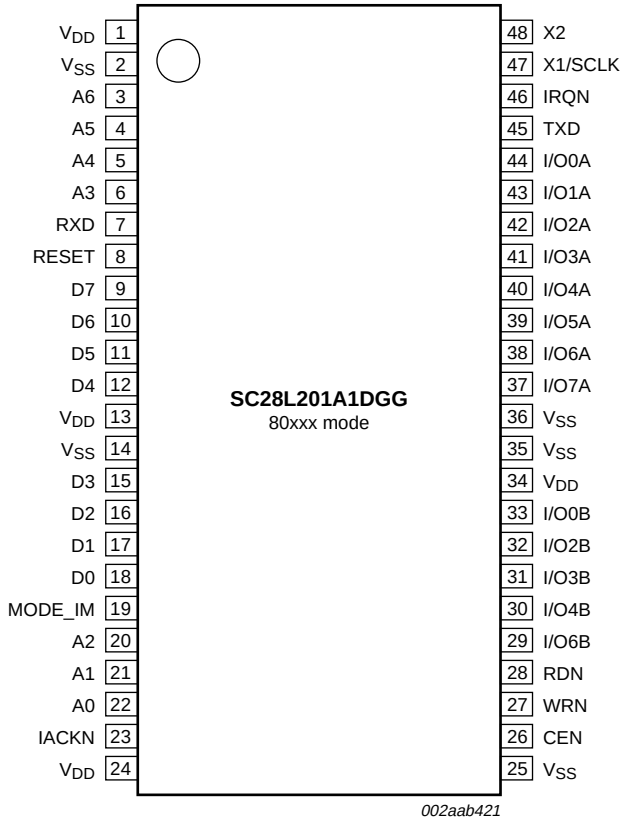


Fig 3. Pin configuration for TSSOP48; 80xxx mode

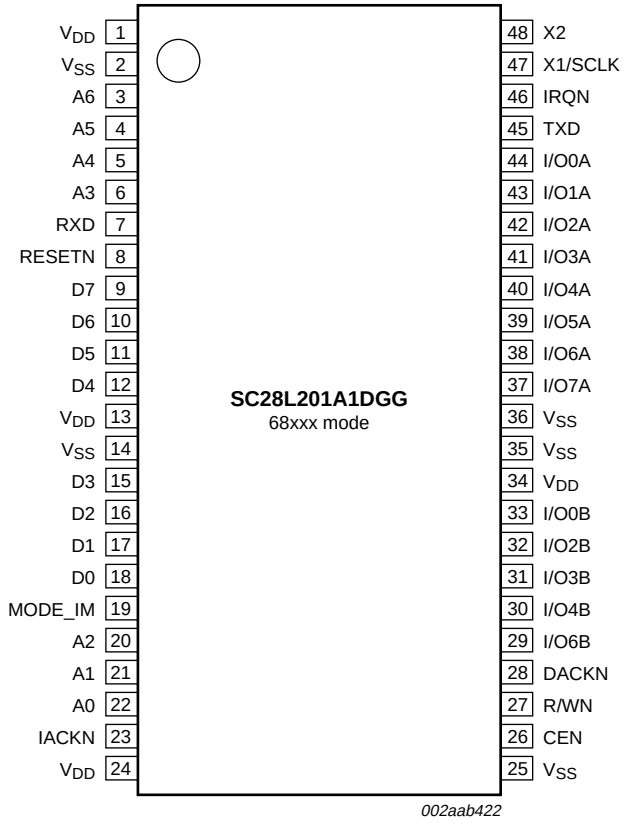


Fig 4. Pin configuration for TSSOP48; 68000 mode

5.2 Pin description

Table 2: Pin description for 80xxx bus interface (Intel)

See Figure 3.

Symbol	Pin	Type	Description
MODE_IM	19	I	**Bus configuration.** When HIGH or not connected configures the bus interface to the conditions shown in this table.
D0 to D7	18, 17, 16, 15, 12, 11, 10, 9	I/O	**Data bus.** Bidirectional 3-state data bus used to transfer commands, data and status between the UART and the CPU. D0 is the least significant bit.
CEN	26	I	**Chip Enable.** Active LOW input signal. When LOW, data transfers between the CPU and the UART are enabled on D[0:7] as controlled by the WRN, RDN and A6 to A0 inputs. When HIGH, places the D[0:7] lines in the 3-state condition.
WRN	27	I	**Write strobe.** When LOW and CEN is also LOW, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.

Table 2: Pin description for 80xxx bus interface (Intel) *...continued*

See Figure 3.

Symbol	Pin	Type	Description
RDN	28	I	**Read strobe.** When LOW and CEN is also LOW, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A6 to A0	3, 4, 5, 6, 20, 21, 22	I	**Address inputs.** Select the UART internal registers and ports for read/write operations.
RESET	8	I	**Reset.** A HIGH level clears internal registers (SR, IMR, ISR, OPR, OPCR), places I/O[7:0] at high-impedance input state, stops the counter/timer, and puts Channel in the inactive state, with the TXD output in the 'mark' (HIGH) state. Sets MR pointer to MR1 9600 baud, 1 start, no parity and 1 stop bit(s).
IRQN	46	O	**Interrupt request.** Active LOW, open-drain output which signals the CPU that one or more of the eleven (11) maskable interrupting conditions are true.
IACKN	23	I	**Interrupt acknowledge.** Active LOW input indicates an interrupt acknowledge cycle. Usually asserted by the CPU in response to an interrupt request. When asserted places the interrupt vector on the bus.
X1/SCLK	47	I	**Crystal 1.** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 12).
X2	48	O	**Crystal 2.** Connection for other side of the crystal. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 12). If X1/SCLK is driven from an external source, this pin must be open or not driving more than 2 CMOS or TTL loads.
RXD	7	I	**Channel A Receiver serial data input.** The least significant bit is received first. 'Mark' is HIGH; 'space' is LOW.
TXD	45	O	**Channel A Transmitter serial data output.** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle or when operating in local loopback mode. 'Mark' is HIGH; 'space' is LOW.
I/O7A to I/O0A	37, 38, 39, 40, 41, 42, 43, 44	I/O	**General-purpose input and output ports.** The character of these pins is controlled by I/OPCR. They may be inputs or outputs and will present many internal clocks and interrupt signals: RTS, CTS, DTR, DSR, and so on. All have change-of-state detectors and the input is always active. These pins are set to input only when addressed from the low order address space. When these pins are configured for interrupt type signals (RXRDY, TXRDY, C/TRDY), they switch to open-drain outputs. Each of these pins has a small pull-up 'resistor' that supplies approximately 5 µA of current.
I/O6B	29	I/O	**Additional general-purpose I/O pins.** They are similar to the above without any connection to the data path or clocks. They have Change-Of-State (COS) detectors and will generate interrupts if enabled. Each of these pins has a small pull-up 'resistor' that supplies approximately 5 µA of current.
I/O4B to I/O2B	30, 31, 32		
I/O0B	33		
V_{DD}	1, 13, 24, 34	power	**Power supply (4 pins).** +3.3 V ± 10 % or +5.0 V ± 10 % supply input. Operation is assured from 2.97 V or 5.5 V. Timing parameters are specified with respect to the V_{DD} being at 3.3 V ± 10 % or 5.0 V ± 10 %.
V_{SS}	2, 14, 25, 35, 36	power	**Ground (5 pins)**

Table 3: Pin description for 68000 bus interface (Motorola)

See Figure 4.

Symbol	Pin	Type	Description
MODE_IM	19	I	**Bus configuration.** When LOW, configures the bus interface to the conditions shown in this table.
D0 to D7	18, 17, 16, 15, 12, 11, 10, 9	I/O	**Data bus.** Bidirectional 3-state data bus used to transfer commands, data and status between the UART and the CPU. D0 is the least significant bit.
CEN	26	I	**Chip Enable.** Active LOW input signal. When LOW, data transfers between the CPU and the UART are enabled on D[0:7] as controlled by the R/WN and A6 to A0 inputs. When HIGH, places the D[0:7] lines in the 3-state condition.
R/WN	27	I	**Read/Write.** Input signal. When CEN is LOW, R/WN HIGH inputs a read cycle, when LOW a write cycle.
IACKN	23	I	**Interrupt acknowledge.** Active LOW input indicates an interrupt acknowledge cycle. Usually asserted by the CPU in response to an interrupt request. When asserted, places the interrupt vector on the bus and asserts DACKN.
DACKN	28	O	**Data transfer acknowledge.** An open-drain active LOW output asserted in a write, read, or interrupt acknowledge cycle to indicate proper transfer of data between the CPU and the UART.
A6 to A0	3, 4, 5, 6, 20, 21, 22	I	**Address inputs.** Select the UART internal registers and ports for read/write operations.
RESETN	8	I	**Reset.** A LOW level clears internal registers (SR, IMR, ISR, OPR, OPCR), places I/O[7:0] A and B at high-impedance input state, stops the counter/timer, and puts Channel in the inactive state, with the TXD output in the ‘mark’ (HIGH) state. Sets MR pointer to MR1 9600 baud, 1 start, no parity and 1 stop bit(s).
IRQN	46	O	**Interrupt request.** Active LOW, open-drain output which signals the CPU that one or more of the eleven (11) maskable interrupting conditions are true.
X1/SCLK	47	I	**Crystal 1.** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 14).
X2	48	O	**Crystal 2.** Connection for other side of the crystal. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 12). If X1/SCLK is driven from an external source, this pin must be open or not driving more than 2 CMOS or TTL loads.
RXD	7	I	**Channel Receiver serial data input.** The least significant bit is received first. ‘Mark’ is HIGH; ‘space’ is LOW.
TXD	45	O	**Channel Transmitter serial data output.** The least significant bit is transmitted first. This output is held in the ‘mark’ condition when the transmitter is disabled, idle or when operating in local loopback mode. ‘Mark’ is HIGH; ‘space’ is LOW.
I/O7A to I/O0A	37, 38, 39, 40, 41, 42, 43, 44	I/O	**General-purpose input and output ports.** The character of these pins is controlled by I/OPCR. They may be inputs or outputs and will present many internal clocks and interrupt signals: RTS, CTS, DTR, DSR, and so on. All have change-of-state detectors and the input is always active. These pins are set to input only when addressed from the low order 16 address space. When these pins are configured for interrupt type signals (RXRDY, TXRDY, C/TRDY), they switch to open-drain outputs. Each of these pins has a small pull-up ‘resistor’ that supplies approximately 5 μA of current.

Table 3: Pin description for 68000 bus interface (Motorola) *…continued*
See Figure 4.

Symbol	Pin	Type	Description
I/O6B	29	I/O	**Additional general-purpose I/O pins.** They are similar to the above without any connection to the data path or clocks. They have Change-Of-State (COS) detectors and will generate interrupts if enabled.
I/O4B to I/O2B	30, 31, 32		
I/O0B	33		
V_{DD}	1, 13, 24, 34	power	**Power supply (4 pins).** +3.3 V ± 10 % or +5.0 V ± 10 % supply input. Operation is assured from 2.97 V or 5.5 V. Timing parameters are specified with respect to the V_{DD} being at 3.3 V ± 10 % or 5.0 V ± 10 %.
V_{SS}	2, 14, 25, 35, 36	power	**Ground (5 pins)**

6. Functional description

Refer to the block diagrams shown in Figure 1 and Figure 2.

The SC28L201 is composed of several functional blocks. They are listed in the approximate order of hierarchy as seen from the pins of the device.

- Bus interface. Motorola or Intel format
- Timing circuits
- I/O ports
- UART
- Transmitter and receiver
- Transmitter real time error test
- FIFO structures
- Arbitrating interrupt structure
- Character and address recognition
- Flow control
- Test and software compatibility with previous Philips UARTs

6.1 Brief description of functional blocks

6.1.1 Bus interface: the two basic modes of bus interface

The bus interface operates in ‘68000’ or ‘86xxx’ format as selected by the MODE_IM pin. The signals used by this section are the Address, Data bus, Chip select, read/write, Data acknowledge and Interrupt acknowledge and Interrupt request. Assertion of DACKN requires two edges of the SCLK after the assertion of CEN. The default mode is the 86xxx mode. Pin or register programming may change it to the 68000 mode.

6.1.2 Timing circuits

6.1.2.1 Crystal oscillator

The crystal oscillator is the main timing element for the SC28L201. It is nominally set at 14.7456 MHz. Operation with a crystal as a frequency standard is specified from 7 MHz to 16.2 MHz. The use of an external clock allows all frequencies to 50 MHz. Clock prescalers are provided to match various available system clocks to those needed for baud rate generation.

Remark: If an external clock is used, X2 should not drive more than 2 CMOS or 2 TTL equivalents.

6.1.2.2 Fixed rate Baud Rate Generator (BRG)

The BRG is driven by the X1/SCLK input through a programmable prescale divider. It generates all of the 27 ‘fixed’ internal baud rates. This baud rate generator is designed to generate the industry standard baud rates from a 14.7456 MHz crystal or clock frequency. X1/SCLK frequencies different from 14.7456 MHz will cause the ‘fixed’ baud rates to change by **exactly** the ratio of 14.7456 to the different frequency.

6.1.2.3 Counter/Timer

The two Counter/Timers are programmable 16-bit ‘down’ counters. It provides miscellaneous baud rates, timing periods and acts as an extra watchdog timer for the receivers. It has 8 programmable clock sources derived from internal and external signals. It may also act as a character counter for the receiver. Interrupts from the Counter/Timer are generated as it passes through zero.

6.1.2.4 Programmable BRG (PBRG)

This is another 16-bit programmable counter to generate only baud rates or miscellaneous clock frequencies. Its output is available to the receiver and transmitter and may be delivered to I/O ports. It has 8 programmable clock sources derived from internal and external signals.

6.1.3 I/O ports

The SC28L201 is provided with 14 I/O ports. These ports are true input and/or output structures and are equipped with a change of state detector. The input circuit of these pins is **always** active. Under program control the ports my display internal signals or static logic levels. The functions represented by the I/O ports include hardware flow control. Modem signals, signals for interrupt conditions or various internal clocks and timing intervals.

Each I/O pin has a change of state detector attached to it. These are used to alert the processor to slow or infrequent signals, modem signals, alarm, power alerts, and so on. For the signals to qualify for Change-Of-State (COS) detection, the signal must be stable for a time of 25 µs to 50 µs (one to two cycles of the 38.4 kBd clock).

The input logic of these pins is **always** active, even when defined as an output. Therefore, it would be possible for the chip to count the number of times the RTS or CTS signal occurred, thus giving an indication of interrupt latency.

6.1.4 UART

The UART is a fully independent, full duplex and provides all normal asynchronous functions: 5 to 8 data bits, parity odd or even, programmable stop bit length, false start bit detection. Also provided are 256-byte FIFOs, Xon/Xoff software flow control, RTS/CTS hardware flow control. The 9-bit mode address recognition with automatic RS485 turnaround. The BRG, Counter/Timer, or external clocks provide the baud rates. The receivers and transmitters may operate in either the '1×' or '16×' modes.

The control section recognizes two address schemes. One is the subset of the other: a four (4) bit and a seven (7) bit address spaces. The purpose of this is to provide a large degree of software compatibility with previous Philips/Signetics UARTs.

6.1.5 Transmitter and receiver

The transmitter and receiver are independent devices capable of full duplex operation. Baud rates, interrupt and status conditions are under separate control. Transmitter has automatic simplex 'turnaround'. Receiver has RTS and Xon/Xoff flow control and a three-character recognition system.

6.1.6 Transmitter real time error check

This is a circuit used to verify that the correct data arrived at the destination. It is done real time with one or two bit times of programmable delay. The purpose of this circuit is to improve the response time of detecting problem data channels and to relieve the processor burden and delay of checking data returned for validation.

The function is that the receiver returns the data received back to the transmitting station where it is compared to a delayed version of the data sent. If an error occurs, and interrupt may be generated for the particular bit that is in error. This is essentially a loopback condition where circuits internal to the UART delay and compare the data returned.

It is suggested that a very high priority be set in the interrupt arbitration bidding control register when the real time error detection is in use.

6.1.7 FIFO structures

The FIFO structure is 256 bytes for each of the two FIFOs in the UART. They are organized as 11-bit words for the receiver and 8-byte words for the transmitter. The interrupt level may be set at any value from 0 to 255. The interrupt level is independently set for each FIFO.

FIFO interrupt and DMA fill/empty levels are controlled by the RXFIL and TXFIL registers which may set any level of the from 0 to 255. The signals associated with the FIFO fill levels are available to the I/O pins (for interrupt or DMA) and to the arbitrating interrupt system for fine tuning of the arbitration authority.

6.1.8 Intelligent interrupt arbitration

The interrupt system uses a highly programmable arbitrating technique to establish when an interrupt should be presented to the processor. The advantageous feature of this system is the presentation of the context of the interrupt. It is presented in both a current interrupt register and in the interrupt vector. The context of the interrupt shows the interrupting channel, identifies which of the 11 possible sources in requesting interrupt service and in the case of a receiver or transmitter gives the current fill level of the FIFO.

The content of the current interrupt register also drives the Global Registers of the interrupt system. These registers are indirect addresses (pointers) to the interrupt source requesting service.

Programming of Bid Control Registers allows the interrupt level of any source to be varied at any time over a range of 256 levels.

6.1.9 Character and address recognition

The character recognition system is designed first and foremost as a general-purpose system that can give an interrupt on the reception or transmission on any of three user-defined characters. A subblock of this system is the special function related to Xon/Xoff flow control and the '9-bit mode'.

The recognition block stores up to three characters. The recognition is done on a byte boundary and sets status and interrupt when recognition events occur. Three modes of automatic operation are provided for the in-band flow control (Xon/Xoff) and three modes of automatic operation are provided for address recognition (9-bit or multi-drop mode). Both in-band flow control and address recognition may also be completely under the control of the host processor.

A subset of the recognition system is Xon/Xoff character recognition and the recognition of the multi-drop address character. If Xon/Xoff or multi-drop function is enabled the recognition system passes the information about the recognition event to the appropriate receiver or transmitter state machine for execution. In any case, the information about a recognition event is available to the interrupt system and to the control processor.

Another subset of the character recognition is recognition of the address character itself (the character value) used in the multi-drop or 9-bit mode. Here also four levels of automatic operation are available. The most interrupt efficient is the 'auto-wake/auto-doze' level which relieves the processor of any tasks.

6.1.10 Flow control

Flow control is implemented in either the traditional RTS/CTS protocol or in the inbound Xon/Xoff method. Both may be controlled by fully/partially automatic methods or by interrupt generation.

6.1.11 Test modes

The three test modes, auto echo, local loopback, and remote loopback, are provided to verify UART function and processor interface integrity at the system level. The local loopback, however, is directed a little more toward the control processor to the UART interface. Through it the software developer may verify all of the interrupt, flow control; the hardware designer may verify all of the timing and pin connections. This information is obtained without any recourse to external test equipment, logic analyzers or terminals.

The auto echo and remote loopback are meant to test the communication channel after it is established that the processor to UART interface is well established.

7. Detailed descriptions

Remark: For the convenience of the reader, some paragraphs of the following sections are repeated in descriptions of closely linked functions described in other sections.

7.1 Bus interface

The bus interface operates in two modes selected by the MODE_IM pin. If this pin is HIGH or left open, the I or 80xxx mode, the signal DACKN signal is not generated or used, and data flow to and from the chip is controlled by the state of the CEN, RDN, WRN pin combination. If the MODE_IM pin is tied LOW, the M or 68000 mode, the data is written to the device when the DACKN pin is asserted LOW by the UART. Read data is presented by a delay from CEN active.

The Host interface in the 80xxx mode is comprised of the signal pins CEN, WRN, RDN. The data is written to the chip on the rise of CEN or WRN, whichever one occurs first. Data is presented to the bus on the condition of both CEN and RDN being LOW. In the read condition, the data bus is returned to high-impedance on the rise of CEN or RDN, whichever one returns HIGH first.

When in the M or 68000 mode, several control pins change function to provide the signals CEN, R/WN, IACKN, DACKN. When CEN is LOW and R/WN is LOW, data will be written to the chip when the DACKN occurs. In a read condition, data will be presented to the bus when CEN is LOW and R/WN is HIGH. DACKN will signal when the data on the bus is valid. The data bus will return to high-impedance with CEN returns HIGH.

The assertion if IACKN (in either mode) will cause the interrupt vector or the interrupt vector modified by the context of the interrupt source to be placed on the bus. In the 68000 mode, the assertion of IACKN will also generate a DACKN cycle. Addressing of the various functions of the UART is through the address bus A[6:0]. Data is presented on the 8-bit data bus.

7.1.1 DACKN cycle (68000 mode)

When operating in the 68000 mode, bus cycle completion is indicated by the DACKN pin (an open-drain signal) going LOW. The timing of DACKN is controlled by GCCR[7:6] where three time delays area available. The delay begins with the falling edge of CEN. DACKN is presented after $\frac{1}{2}$ to three periods of X1/SCLK. The minimum time will be two edges of the X1/SCLK and will be realized when the bus cycle begins just before the transition of X1/SCLK. Usually in this mode the address and data are set up with respect to the leading edge of the bus cycle. Timing diagrams for this mode are drawn with DACKN in consideration. When CEN is withdrawn before DACKN occurs, the generation of the DACKN signal and bus cycle will be terminated. In this case, the bus timing will return to that of Intel-type timing for that particular cycle. This timing should not be less than the minimum read or write pulse.

The DACKN pin is an open-drain driver. At the termination of an access to the SC28L201 DACKN drives the pin to high-impedance until the next DACKN cycle. This will occur at the termination of the CEN or IACKN cycle.

Remark: The faster 86xxx timing may be used in the 68000 mode **if** the bus cycles are faster than $\frac{1}{2}$ period of the SCLK clock. Withdrawing CEN before DACKN prevents the generation of DACKN. In this case, bus timing is effectively that of the 86xxx mode.

When operating in the 86xxx mode, DACKN is not generated. Data is written on the termination of CEN or WRN, whichever one occurs first. Read data is presented from the leading edge of the read condition (CEN and RDN both LOW).

In the 68000 mode, data is written to the registers on the rise of CEN or the fall of DACKN, whichever one occurs first. Data on a read cycle will become valid with respect to the fall of CEN. It will always be valid at the fall of DACKN. The bus returns to high-impedance when either CEN or RDN returns to a logical 1 (HIGH).

7.1.2 IACKN cycle, update CIR

(Valid for both interrupt and polled service modes.)

When the host CPU responds to the interrupt, it will usually assert the IACKN signal LOW. This will cause the intelligent interrupt system of the UART to generate an IACKN cycle in which the condition of the interrupting source is determined. When IACKN asserts, the last valid of the interrupt arbitration cycle is captured in the CIR. The value captured presents all of the important details of the highest priority interrupt at the moment the IACKN (or the Update CIR command) was asserted. Due to system interrupt latency the interrupt condition captured by the CIR may not be the condition that caused the initial assertion of the interrupt. Recall that any number of interrupts can occur at the same time. Nearly all interrupt events are totally asynchronous to each other and will depend on a variety of internal or external clocks with various times or being enabled or disabled.

The UART will respond to the IACKN cycle with an interrupt vector. The interrupt vector may be a fixed value, the content of the Interrupt Vector Register, or when interrupt vector modification is enabled via ICR, it may contain codes for the interrupt type and/or interrupting channel. This allows the interrupt vector to steer the interrupt service **directly** to the proper service routine. The interrupt value captured in the CIR remains until another IACKN or Update CIR command is given to the UART. The interrupting channel and interrupt type fields of the CIR set the current interrupt context of the UART. The channel component of the interrupt context allows the use of Global Interrupt Information registers that appear at fixed positions in the register address map. For example, a read of the Global GIBCR will read the channel FIFO byte count if the CIR interrupt context is the receiver. At another time read of the GIBCR will show the transmitter byte count if the interrupt context is that of the transmitter interrupt, and so on. Global registers exist to facilitate qualifying the interrupt parameters and for writing to and reading from FIFOs without explicitly addressing them. They are essentially an indirect address to the content of the CIR.

The CIR will load with 0x00 if IACKN or Update CIR is asserted when the arbitration circuit is **not** asserting an interrupt. In this condition there is no arbitration value that exceeds the threshold value. When Interrupt vector modification is active in this situation the interrupt vector bits associated with the CIR will all be zero. A zero type field indicates nothing with in the UART is requiring processor service.

Remark: IACKN is essentially a special read action where the value of the interrupt vector is presented to the data bus.

7.2 Timing circuits

7.2.1 Crystal oscillator

The crystal oscillator operates directly from a crystal, tuned between 7.0 MHz and 16.2 MHz connected across the X1/SCLK and X2 inputs with a minimum of external components. BRG values listed for the clock select registers correspond to a 14.7456 MHz crystal frequency. Use of different frequencies will change the ‘standard’ baud rates by precisely the ratio of 14.7456 MHz to the different crystal frequency.

An external clock up to 50 MHz frequency range may be connected to X1/SCLK pin. If an external clock is used instead of a crystal, X1/SCLK **must** be driven and X2 left floating or driving a load of not more than 2 CMOS or TTL equivalents. The X1/SCLK clock serves as the basic timing reference for the baud rate generator (BRG) and is available to the programmable BRG (PBRG), counter-timers, control logic and the UART receivers and transmitters.

7.2.2 Baud rage generator (BRG)

The baud rate generator operates from the oscillator or external X1/SCLK clock input and generates 27 commonly-used data communications baud rates (including MIDI) ranging from 50 baud to 921.6 kBd. These common rates may be increased (up to 3.125 MBd) when faster clocks are used on the X1/SCLK clock input. (See Section 8.2.5 “Receiver Clock Select Register (RxCSR) and Transmitter Clock Select Register (TxCSR)”.) All of these are available simultaneously for use by any receiver or transmitter. The clock outputs from the BRG are at 16× (the actual baud rate).

Please see Section 7.2.3 “Counter/Timer” for a description of the frequency error that the asynchronous protocol may tolerate. Depending on character length it varies from 4.1 % to 6.7 %.

7.2.3 Counter/Timer

The two Counter/Timers are programmable 16-bit dividers that are used for generating miscellaneous clocks or generating time-out periods or counting characters received by the receivers. Interrupts may be generated any time the counter passes through 0x00. The counter/timer clocks may be used simultaneously by receiver, transmitter, I/O pin, time-out logic, or interrupt.

7.2.3.1 Counter/Timer programming

The Counter/Timer is a 16-bit programmable divider that operates in one of four modes: character count, counter, timer, and time-out. Character count counts characters. The Timer mode generates a square wave or a pulse. If a square wave is programmed, the counter counts down once for the HIGH portion, and once for the LOW portion of the square wave. In the Pulse mode it counts down and outputs a pulse one-clock cycle. Recall that the input to the counter may be from many places other than the X1 crystal clock.

In the Counter mode it generates a time delay. In this mode, the counter effectively stops at the end of the time-out, it does not continue until another START/STOP timer command sequence is given. In the Time-out mode, it monitors the time between received characters. If the time between any two characters is longer than the programmed time, and interrupt is generated. This activity is similar to a receiver watchdog timer, but the true meaning is that the data has stopped. The watchdog action indicates there is data in the

RXFIFO that has not been read. The Counter/Timer uses the numbers loaded into the Counter/Timer Lower Register (CTPL and the Counter/Timer Upper Register (CTPU) as its divisor. 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 Section 8.4.7 “Counter/Timer Preset Upper and Counter/Timer Preset Lower (CTPU, CTPL)”.

Whenever the these timers are selected via the receiver or transmitter Clock Select register, their output will be configured as a 16× clock for the respective receiver or transmitter. Therefore, one needs to program the timers to generate a clock 16 times faster than the data rate. The formula for calculating ‘n’, the number loaded to the CTPU and CTPL registers, based on a particular input clock frequency is shown in Equation 1.

For the Timer mode, the formula is as follows:

$$n = \frac{\text{C/T clock input frequency}}{(2 \times 16 \times \text{desired baud rate})} \quad (1)$$

(If the pulse mode is selected, then ‘2’ in the divisor should be ‘1’. This doubles the C/T output speeds for any given input clock.)

Remark: ‘n’ may assume a value of 1. In previous Philips data communications controllers this value was not allowed. The Counter/Timer Clock Select Register (CTCS) controls the Counter/Timer input frequency.

The frequency generated from the above formula will be at a rate 16 times faster than the desired baud rate. The transmitter and receiver state machines include divide-by-16 circuits, which provide the final frequency and provide various timing edges used in the qualifying the serial data bit stream.

Often this division will result in a non-integer value: 26.3 for example. One may only program integer numbers to a digital divider. Therefore, 26 (0x001A) would be chosen. If 26.7 were the result of the division then 27 (0x001B) would be chosen. This gives a baud rate error of 0.3/26.3 or 0.3/26.7 that yields a percentage error of 1.14 % or 1.12 % respectively, well within the ability of the asynchronous mode of operation. Higher input frequency to the counter reduces the error effect of the fractional division.

One should be cautious about the assumed benign effects of small errors since the other receiver or transmitter with which one is communicating may also have a small error in the precise baud rate. In a clean communications environment using one START bit, eight data bits, and one STOP bit, the total difference allowed between the transmitter and receiver frequency is approximately 4.6 %. Less than eight data bits will increase this percentage.

7.2.4 Programmable Baud Rate Generators (PBRG)

There are two PBRG Counters, used only for random baud rate generation. The two PBRG Timers are programmable 16-bit dividers that are used for generating miscellaneous clocks. These clocks may be used by the receiver, transmitter, counter/timers or I/O pin at any time in any combination.

Each timer unit has eight different clock sources available to it, as described in Section 8.4.6 “Programmable BRG Clock Source, 0 and 1 (PBRGCS)”. Note that the timer run and stop controls are also contained in this register. The PBRG counters generate a symmetrical square wave whose half period is equal in time to the division of the selected PBRG Timer clock source by the number loaded to the PBRGPU and PBRGPL Preset Registers. Thus, the output frequency will be the clock source frequency divided by twice the 16-bit value loaded to these registers. This is the result of counting down once for the high portion of the output wave and once for the low portion.

Whenever the these timers are selected via the receiver or transmitter Clock Select register, their output will be configured as a 16× clock for the respective receiver or transmitter. Therefore, one needs to program the timers to generate a clock 16 times faster than the data rate. The formula for calculating ‘n’, the number loaded to the PBRGPL and PBRGPU registers, is the same as shown in Equation 1.

7.3 I/O ports

Thirteen (13) I/O ports are provided for the UART. They may be programmed to be inputs or outputs. The input circuits are always active whether programmed as and input or an output. In general a 2-bit code in the **I/OPCR** (I/O Port Control Register) controls what function these pins will present. All I/O ports default to high-impedance input state on power-up. All 13 I/O pins have a small pull-up ‘resistor’ that provides approximately 5 μA current.

Remark: When calling software written for legacy two-channel UARTs manufactured by Philips (Signetics), be sure I/O pins are set to input where the legacy software expected an input. Declare I/O pins as output where the legacy software expected an output.

7.3.1 Input characteristics of the I/O ports

The I/O pins are configured individually to be inputs or outputs. As inputs they may be used to bring external data to the bus, as clocks for internal functions or external control signals. Each I/O pin has a ‘Change-of-State’ detector. The change detectors are used to signal a change in the signal level at the pin (either 0-to-1 or 1-to-0 transitions). The level change on these pins must be stable for approximately 25 μs to 50 μs (two edges of the internally generated 38.4 kHz baud rate clock) before the detectors will signal a valid change. These are typically used for interface signals from modems to the UART and from there to the host.

7.3.2 Output port of the I/O ports

The OPR, I/OPCR, MR, and CR registers may control the I/O pins when configured as outputs. (For the control in the lower 16 position address space, the control register is the OPCR.) Via appropriate programming, the pins of the output port may be configured as another parallel port to external circuits, or they may represent internal conditions of the UART. When this 8-bit port is used as a general-purpose output port, the output port pins drive inverse logic levels of the individual bits in the Output Port Register (OPR). The OPR register is set and reset by writing to the SOPR and ROPR addresses (seeSection 8.5.10 “Bidding Control Register, Break Change (BCRBRK)” and Section 8.5.11 “Bidding Control Register, Change-Of-State (BCRCOS)”). The output pins will drive the same data polarity of the OPR registers. The I/OPCR (or the OPCR) register conditions these output pins to be controlled by the OPR or by other signals in the chip. Output ports are driven HIGH on hardware reset.

When legacy code is called using the lower 16 address portions (0x00 to 0x0F), the I/O pins will be switched to input. Legacy code would expect to see the I/OA pins to be input and the I/OB pins to be output driving HIGH as a default condition. In calling legacy code, this condition must be accounted for.

7.4 UART operation

7.4.1 Receiver and transmitter

The Dual UART has two full duplex asynchronous receiver/transmitters. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter, or from an external input. Registers that are central to basic full-duplex operation are the mode registers (MR0, MR1 and MR2), the clock select registers (RxCSR and TxCSR), the command register (CR), the status register (SR), the transmit holding register (TxFIFO), the receive holding register (RxFIFO), interrupt status register (ISR) and interrupt mask register (IMR). MR3 controls the automatic activity or the Xon/Xoff flow control, Address recognition, multi-drop (‘9-bit’ mode) and general purpose character recognition. Because MR3 does not exist in legacy UARTs, these features should be disabled before legacy code is loaded.

7.4.2 Transmitter status bits

The SR (Status Register) contains two bits that show the condition of the transmitter FIFO. These bits are TxRDY and Tx Idle. TxRDY means the TxFIFO has space available for one or more bytes; Tx Idle means the TxFIFO is completely empty and the last stop bit has been completed: the transmitter is underrun. Tx Idle can not be active without TxRDY also being active. These two bits will go active upon initial enabling of the transmitter.

The transmitter status bits are normally cleared by servicing the interrupt condition they represent or by Tx reset or Tx disable commands.

Transmission resumes and the Tx Idle bit is cleared when the CPU loads at least one new character into the TxFIFO. The TxRDY will not extinguish until the TxFIFO is completely full. The TxRDY bit will always be active when the transmitter is enabled and there is at lease one open position in the TxFIFO.

The transmitter is disabled by a hardware reset, a transmitter reset in the command register or by the transmitter disable bit also in the command register (CR). The transmitter must be explicitly enabled via the CR before transmission can begin. Note that characters cannot be loaded into the TxFIFO while the transmitter is disabled, hence it is necessary to enable the transmitter and then load the TxFIFO. It is not possible to load the TxFIFO and then enable the transmission.

Note the difference between transmitter disable and transmitter reset.

Either hardware or software may cause the reset action. When reset the transmitter stops transmission immediately. The transmit data output will be driven HIGH, transmitter status bits set to zero and any data remaining in the TxFIFO is effectively discarded.

The transmitter disable is controlled by the Tx Enable bit in the command register. Setting this bit to zero will not stop the transmitter immediately but will allow it to complete any tasks presently underway. It is only when the last character in the TxFIFO and its stop bit(s) have been transmitted that the transmitter will go to its disabled state. While the

transmitter enable/disable bit in the command register is at zero the TxFIFO will not accept any more characters and the Tx Idle and TxRDY bits of the status register set to zero.

7.4.3 Transmission of 'break'

Transmission of a break character is often needed as a synchronizing condition in a data stream. The 'break' is defined as a start bit followed by all zero data bits by a zero parity bit (if parity is enabled) and a zero in the stop bit position. The forgoing is the minimum time to define a break. The transmitter can be forced to send a break (continuous LOW condition) by issuing a start break command via the CR. Once the break starts, the TXD output remains LOW until the host issues a command to 'stop break' via the CR or the transmitter is issued a software or hardware reset. In normal operation the break is usually much longer than one character time.

7.4.4 1x and 16x modes, transmitter

The transmitter clocking has two modes: 16× and 1×. Data is always sent at the 1× rate. However. the logic of the transmitter may be operated with a clock that is 16 times faster than the data rate or at the same rate as the data, that is, 1×. All clocks selected internally for the transmitter (and the receiver) will be 16× clocks. Only when an external clock is selected may the transmitter logic and state machine operate in the 1× mode. The 1× or 16× clocking makes little difference in transmitter operation. (This is not true in the receiver.) In the 16×-clock mode, the transmitter will recognize a byte in the TxFIFO within $^{1}/_{16}$-bit time to $^{2}/_{16}$-bit time and thus begin transmission of the start bit. In the 1× mode this delay may be up to 2 bit times.

7.4.5 Transmitter FIFO

The FIFO configuration of the SC28L201 is as 256 8-bit words. Interrupt levels may be set to any level within the FIFO size and may be set differently for each FIFO. Logic associated with the FIFO encodes the number of empty positions for presentation to the interrupt arbitration system. The encoding value is the number of empty positions. Thus, an empty TxFIFO will bid with the value or 255; when full it will not bid at all; one position empty bids with the value 0. A Full TxFIFO will not bid since no character is available.

Normally TxFIFO will present a bid to the arbitration system whenever it has one or more empty positions. The bits of the TxFIFO Interrupt Level in MR0[5:4] allow the user to modify this characteristic so that bidding will not start until one of four levels (one or more filled, empty, 16 filled, 240 filled, full) have been reached. As will be shown later, this feature may be used to make moderate improvements in the interrupt service efficiency. A similar system exists for the Receiver.

7.4.6 Transmitter

The SC28L201 is conditioned to transmit data when the transmitter is enabled through the Command Register. The transmitter of the SC28L201 indicates to the CPU that it is ready to accept a character by setting the ISR TxRDY bit in the Status Register. This condition can be programmed to generate an interrupt request at I/O4 or IRQN. When the transmitter is initially enabled the TxRDY and Tx Idle bits will be set in the Status Register. When a character is loaded to the transmit FIFO, the Tx Idle bit will be reset. The Tx Idle bit will not set until the transmit FIFO is empty and the transmit shift register has finished transmitting the stop bit of the last character written to the transmit FIFO.

The TxRDY bit is set whenever the transmitter is enabled and the TxFIFO is not full. Data is transferred from the holding register to Transmit Shift Register when it is idle or has completed transmission of the previous character. Characters cannot be loaded into the TxFIFO while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TXD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the TxFIFO, the TXD output remains High and the Tx Idle bit in the Status Register (SR) will be set to 1. Transmission resumes and the Tx Idle bit is cleared when the CPU loads a new character into the TxFIFO.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous LOW condition by issuing a ‘send break’ command. The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the Command Register before resuming operation.

If CTS option of hardware flow control is enabled (MR2[4] = 1), the CTS input at I/O[0]A must be LOW in order for the character to be transmitted. The transmitter will check the state of the CTS input at the beginning of each character transmitted. If it is found to be HIGH, the transmitter will delay the transmission of any characters until the CTS has returned to the LOW state. CTS going HIGH during the serialization of a character will not affect that character.

An interesting result of the I/O pin input circuit always being active is that it gives software control of transmitter activity. Programming the MR2[4] to ‘1’ gives I/O[0]A (CTSN) control of the transmitter. Thus, if software drives I/O[0]A HIGH or LOW, the transmission of data is started or stopped by direct software commands.

The transmitter can also control the RTSN outputs, I/O[0]B, via MR2[5]. When this mode of operation is set (often referred to as the RS-485 method) the meaning of the I/O[0]B signal is all bytes loaded to the transmitters FIFO have been transmitted including the last stop bit(s). See Section 8.2.3 “Mode Register 2 (MR2)” for enabling this automatic function.

7.4.7 Receiver operation

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), framing error or break condition, and presents the assembled character and its status condition to the CPU via the RxFIFO. Three status bits are FIFOed with each character received. The RxFIFO is really 11 bits wide: eight data bits and 3 status bits. Unused FIFO bits for character lengths less than 8 bits are set to zero.

It is important to note that in the asynchronous protocol the receiver logic considers the entire message to be contained within the start bit to the stop bit. It is not aware that a message may contain many characters. **The receiver returns to its Idle mode at the end of each stop bit.** As described below it immediately begins to search for another start bit, which is normally, of course, immediately forthcoming.

7.4.7.1 1x and 16x modes, receiver

The receiver operates in one of two modes: 1× and 16×. Of the two, the 16× is by far more robust and the preferred mode. Although the 1× mode may allow a faster data rate, it does not provide for the alignment of the receiver 1× data clock to that of the transmitter. This strongly implies that the 1× clock of the remote transmitter is available to the receiver; the two devices are physically close to each other.

The 16× mode operates the receiver logic at a rate 16 times faster than the 1× data rate. This allows for validation of the start bit length, the validation of level changes at the receiver serial data input (RXD), and the validation of the stop bit length. Of most importance in the 16× mode is the ability of the receiver logic to align the phase of the internally generated receiver 1× data clock to that of the received start bit of the remote transmitter. This occurs with an accuracy of less than $^{1}/_{16}$-bit time.

7.4.7.2 Receiver

The receiver of the SC28L201 is conditioned to receive data when enabled through the Command Register. The receiver looks for a HIGH-to-LOW (mark-to-space) transition of the start bit on the RXD input pin. If a transition is detected, the state of the RXD pin is sampled each 16× clock for $7^{1}/_{2}$ clock periods (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode). If RXD is sampled HIGH, (that is the start bit was LOW less than $^{7}/_{16}$-bit to $^{1}/_{2}$-bit time) the start bit is judged invalid and the search for another valid start bit begins immediately. If RXD is still LOW, a valid start bit is assumed and the receiver then continues to sample the input at one-bit time intervals at the theoretical center of the bit. When the proper number of data bits and parity bit (if used) have been assembled, and one half-stop bit has been detected the receiver loads the byte to the FIFO. The least significant bit is received first. The data is then transferred to the Receive FIFO and the ISR RxRDY bit in the SR is set to ‘1’. This condition can be programmed to generate an interrupt at IRQN or I/O[4]B. If the character length is less than 8 bits, the most significant unused bits in the RxFIFO are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received with the stop bit at a zero level (framing error) and RXD remains LOW for at least another $^{1}/_{2}$-bit time after the stop bit was sampled, then the receiver operates as if a new start bit had been detected. It then continues assembling the next character.

The error conditions of parity error, framing error, and overrun error (if any) are written to the SR at the received character boundary. This is just before the RxRDY status bit is set.

A break condition is detected when RXD is LOW for the entire character including the parity bit, if used, and stop bit. When a break is found a character consisting of all zeros will be loaded into the RxFIFO, the received break bit in the SR and the change of break bit in the ISR are set to 1 and the receiver ready is set in the SR. The RXD input must **return to HIGH for two (2) clock edges** of the RxC1x clock for the receiver to recognize the end of the break condition. At the end of the break condition the search for the next start bit begins.

Two edges of the RxC1x clock will usually require a HIGH time of one RxC1x clock period or 3 RxC1x edges since the clock of the controller is usually not synchronous to nor in phase with the RxC1x clock.

7.4.7.3 Receiver status bits

There are five (5) status bits that are evaluated with each byte (or character) received: received break, framing error, parity error, overrun error, and change of break. The first three are appended to each byte and stored in the RxFIFO.

The overrun error and change of break are not necessarily associated with the byte presently being received. They are developed by the receiver state machine. They will persist until a command to rest them is issued to the command register. A change of break occurs on a beginning and the end of a break condition. The meaning of overrun is that data has not been lost. All data before the overrun flag is set if valid and available.

The receiver status bits are normally cleared by servicing the interrupt condition they represent or by Rx reset or Rx disable commands or the several error reset commands in the Command Register (CR).

The ‘received break’ will always be associated with a zero byte in the RxFIFO. It means that zero character was a break character and not a zero data byte. The reception of a break condition will always set the ‘change of break’ status bit in the Interrupt Status Register (ISR).

The ‘change of break’ condition is reset by a reset error status command in the Command Register.

A framing error occurs when a non-zero character was seen and that character has a zero in the stop bit position.

The parity error indicates that the receiver-generated parity was not the same as that sent by the transmitter.

The framing, parity and received break status bits are reset when the associated data byte is read from the RxFIFO since these ‘error’ conditions are attached to the byte that has the error.

The overrun error occurs when the RxFIFO is full, the receiver shift register is full, and another start bit is detected. At this moment the receiver has 257 valid characters and the start bit of the 258th has been seen. At this point the host has approximately $^{6}/_{16}$-bit time to read a byte from the RxFIFO or the overrun condition will be set. The 258th character then overruns the 257th, and the 258th the 259th, and so on until an open position in the RxFIFO is seen (‘seen’ meaning at least one byte was read from the RxFIFO).

Overrun is cleared by a use of the ‘error reset’ command in the Command Register.

The fundamental meaning of the **overrun** is that data has been lost. Data in the RxFIFO remains valid. The receiver will begin placing characters in the RxFIFO as soon as a position becomes vacant.

Remark: Precaution must be taken when reading an overrun FIFO. There will be 256 valid characters in the receiver FIFO. There will be one character in the receiver shift register. However it will **not** be known if more than one ‘over-running’ character has been received since the overrun bit was set. The 257th character received and read as valid but it will not be known how many characters were lost between the two characters of the 256th and 257th reads of the RxFIFO. In the 8-bit mode, the numbers 8 and 9 replace the numbers 256 and 257 above.

The ‘change of break’ means that either a break has been detected or that the break condition has been cleared. This bit is available in the ISR. The break change bit being set in the ISR and the received break bit being set in the SR will signal the beginning of a break. At the termination of the break condition only the change of break in the ISR will be set. After the break condition is detected the termination of the break will only be recognized when the RXD input has returned to the HIGH state for **two** successive edges of the 1× clock; $\frac{1}{2}$-bit to 1-bit time (see above).

The receiver is disabled by reset or via CR commands. A disabled receiver will not interrupt the host CPU under any circumstance in the **normal** mode of operation. If the receiver is in the multi-drop or special mode, it will be partially enabled and thus may cause an interrupt. Refer to Section 8.2.2 “Mode Register 1 (MR1)” for more information.

7.4.7.4 Receiver FIFO

The receiver buffer memory is a 256-byte FIFO with three status bits appended to each data byte. (The FIFO is then 256 11-bit ‘words’.) The receiver state machine gathers the bits from the Receiver Shift Register and the status bits from the receiver logic and writes the assembled byte and status bits to the RxFIFO shortly after the stop bit has been sampled. Logic associated with the FIFO encodes the number of filled positions for presentation to the interrupt arbitration system. The encoding is always the number of filled positions. Thus, a full RxFIFO will bid with the value of 255 **and** the Status Register RxFULL bit is set. The RxFULL bit means 256 characters. When empty, it will not bid at all. One position occupied bids with the value ‘1’. An empty FIFO will not bid since no character is available.

Normally RxFIFO will present a bid to the arbitration system whenever it has one or more filled positions. The bits of the RxFIFO Interrupt Offset Level (RxFIL) of MR0 allow the user to modify this characteristic so that bidding will not start until one of four levels has been reached.

As will be shown later, this feature may be used to make slight improvements in the interrupt service efficiency. A similar system exists in the transmitter.

7.4.7.5 RxFIFO status bits, status reporting modes

This description applies to the upper three bits in the Status Register. These three bits are not ‘in the status register’; they are part of the RxFIFO. The three status bits at the output of the RxFIFO are presented as the upper three bits of the Status Register.

The error status of a character, as reported by a read of the SR (status register upper three bits) can be provided in two ways, as programmed by the error mode control bit in the mode register: Character mode or the Block mode. The Block mode may be further modified (via a CR command) to set the status bits as the characters enter the FIFO or as they are read from the FIFO.

In the Character mode, status is provided on a character-by-character basis as the characters are read from the RxFIFO: the ‘status’ applies only to the character at the output of the RxFIFO (the next character to be read).

In the Block mode (on entry), the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the input of the RxFIFO since the last reset error command was issued. In this mode each of the status bits stored in the RxFIFO are passed through a latch as they are sequentially written to the receiver FIFO. If any of the characters has an error bit set that latch will set and remain set until it is reset

with a ‘receiver reset’ issued from the Command Register or a chip reset is issued. The purpose of this mode is indicating an error in the data block as opposed to an error in a character. This mode improves receiver service efficiency. In modern systems with low error rates, it is more efficient to ask for retransmit of a block error data than to analyze it on a byte by byte system.

The above paragraph describes the Block mode activity as the data is entered to the RxFIFO. Normally the status would be read only once: at the beginning of the service to the receiver interrupt. If an error is not set then the entire amount of data in the RxFIFO would be read without any more reading of the receiver status. This effectively doubles the efficiency of reading the receiver RxFIFO.

The use of the Block mode on Exit passes the data and error conditions as the RxFIFO is read. Here the final read of the status register would be after the last byte was read from the RxFIFO. This delays the knowledge of an error condition until after the data has been read.

The latch used in the block mode to indicate ‘problem data’ is usually set as the characters are read out of the RxFIFO. Via a command in the CR, the latch may be configured to set as error characters are loaded to the RxFIFO. This gives the advantage of indicating ‘problem data’ up to 256 (or the FIFO size) characters earlier.

In either mode, reading the SR does not affect the RxFIFO. The RxFIFO address is advanced only when the RxFIFO is read. Therefore, the SR should be read **prior** to reading the corresponding data character.

If the RxFIFO is full when a new character is received, the character is held in the receiver shift register until a position is available in the RxFIFO. At this time there are 257 valid characters in the RxFIFO. If an additional character is received while this state exists, the contents of the RxFIFO 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.

7.4.7.6 Wake-up mode

(Also referred to as the ‘9-bit’, ‘multi-drop’, ‘party line’ or Special mode.)

In the use of this mode, the parity bit is used to distinguish between an address byte and a data byte. The purpose is to allow data to be directed to a particular station from a master station. A station is addressed by a byte with the parity bit set to ‘1’. The data for that station is sent following the address and all the data bytes have the parity bit set to ‘0’.

The SC28L201 provides four modes of this common asynchronous ‘party line’ protocol where the parity bit is used to indicate that a byte is address data or information data. Three automatic modes and the default Host operated mode are provided. The automatic mode has several sub-modes (see below). In the full automatic mode the internal state machine devoted to this function will handle all operations associated with address recognition, data handling, receiver enables and disables. In both modes the meaning of the parity bit is changed. A ‘1’ usually means address, a ‘0’ means data.

Its purpose is to allow several receivers connected to the same data source to be individually addressed. Of course addressing could be by group also. Normally the ‘Master’ would send an address byte to all receivers ‘listening’. The remote receiver will be ‘looking’ at the data stream for its address. Upon recognition of its address the receiver

will enable itself to receive the following data stream. Upon receipt of an address not its own it would then disable itself. As described below appropriate status bits are available to describe the operation. Again, for this mode an ‘address byte’ is a byte that has the bit in the parity position set to logical 1.

The use of the multi-drop mode usually implies a ‘master and slave’ configuration, but is not required, of the several UART stations so programmed. The software control should allow time for the slave stations to respond to the receipt of an address bit. Often a reply from the addressed station is expected to confirm the receipt of the address. Please see Section 8.2.4 “Mode Register 3 (MR3)”.

Enabling the Wake-up mode: (This mode is variously referred to as ‘9-bit’ or ‘multi-drop’.)

This mode is selected by programming bits MR1[4:3] (the parity bits) to ‘11’. The wake-up feature has four modes of operation: one strictly under processor control and three automatic. These modes are controlled by bits 6, 1, 0 in the MR3 register. Bit 6 controls the loading of the address byte to the RxFIFO and MR3[1:0] determines the sub-mode as shown in the following list.

MR3[1:0] = 00 Normal Wake-up Mode (default), which is the same as previous UARTs, and is therefore under full control of the processor. The Host controls operation via interrupts it receives and commands it writes to the UART Command Registers (CR).

Normal wake-up (the default configuration): The enabling of the Wake-up mode executes a partial enabling of the receiver state machine. Even though the receiver has been reset, the Wake-up mode will override the disable and reset condition.

In the default (mode ‘00’ above and the least efficient) configuration for this mode of operation, a ‘master’ station transmits an address character followed by data characters for the addressed ‘slave’ station. The slave stations, whose receivers are normally disabled (not reset), examine the received data stream. Upon recognition of its address bit **(this is the parity bit redefined to indicate the associated byte is an address byte, not the address itself)** interrupts the CPU (by setting RxRDY). The CPU (host) compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit; the programmed number of data bits, an address/data (A/D) bit and the programmed number of stop bits. The CPU selects the polarity of the transmitted A/D bit by programming bit MR1[2]. MR1[2] = 0 transmits a zero in the A/D bit position which identifies the corresponding data bits as **data**. MR1 [2] = 1 transmits a one in the A/D bit position which identifies the corresponding data bits as an **address**. The CPU should program the mode register prior to loading the corresponding data bytes into the TxFIFO.

While in this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the 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 the receiver is enabled, all received characters are transferred to the CPU via the RxFIFO. In either case when the address character is recognized the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SR[5]). Framing error, overrun error, and break

detect operate normally whether or not the receiver is enabled. When the automatic modes are in operation the loading of the address character to the FIFO is controlled by the MR0[6] bit.

The several automatic controls. These modes are concerned with the recognition of the address character itself:

- MR3[1:0] = 01; Auto wake.

  Enable receiver on address recognition for this station. Upon recognition of its assigned address the local receiver will be enabled by the character recognition state machine and normal receiver communications with the host will be established. The address just received may be discarded (stripped from the data stream) or loaded to the RxFIFO depending on the programming of MR0[6].

- MR3[1:0] = 10; Auto Doze.

  Disable receiver on address recognition, not for this station. Upon recognition of an address character that is not its own, in the Auto Doze mode, the receiver will be disabled by the character recognition state machine and the address just received either discarded or loaded to the RxFIFO depending on the programming of MR0[6].

- MR3[1:0] = 11; Auto wake and doze.

  Both modes described above. The programming of MR3[1:0] to 11 will enable both the auto wake and auto doze features. The address just received may be discarded (stripped from the data stream) or loaded to the RxFIFO depending on the programming of MR0[6].

The enabling of the Wake-up mode executes a partial enabling of the receiver state machine. Even though the receiver has been reset, the Wake-up mode will override the disable and reset conditions.

7.4.7.7 Receiver reset and disable

Please note the difference between Receiver disable and reset.

Receiver disable stops the receiver immediately. Data being assembled in the Receiver Shift Register is lost. Data and status in the FIFO is preserved and may be read. A re-enable of the receiver after a disable will cause the receiver to begin assembling characters at the next start bit detected.

Receiver reset will discard the present shift register data, reset the receiver ready bit (RxRDY), clear the status of the byte at the top of the FIFO and re-align the FIFO read/write pointers. This effectively clears the receiver FIFO, although the FIFO data is not altered.

7.4.7.8 Receiver watchdog timer

A ‘watchdog timer’ is associated with each receiver. Its interrupt is enabled by the ‘watchdog’ bits of the Watchdog, Character Address, and X enable register (WCXER). The purpose of this timer is to alert the control processor that characters are in the RxFIFO which have not been read. This situation may occur at the end of a transmission when the last few characters received are not sufficient to cause an interrupt. The Watchdog counter times out after 64 bit times. It is reset each time a read of the RxFIFO is executed or a write to the RxFIFO is executed by the receiver state machine.

7.4.7.9 Receiver Time-out mode

Remark: This is similar to the watchdog above, but its more precise meaning is that the receiver data stream has stopped.

In addition to the watchdog timer described in Section 7.4.7.8, the counter/timer may be used for a similar function. Its programmability, of course, allows much greater precision of time-out intervals.

The Time-out mode uses the received data stream to control the counter. Each time a received character is transferred from the shift register to the RxFIFO, the counter is restarted. If a new character is not received before the counter reaches zero count, the counter ready bit is set, and an interrupt can be generated. This mode can be used to indicate when data has been left in the RxFIFO for more than the programmed time limit. Otherwise, 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 may not know there is data left in the FIFO. The CTPU and CTPL value would be programmed for just over one character time, so that the CPU would be interrupted as soon as it has stopped receiving continuous data. This mode can also be used to indicate when the serial line has been marking for longer than the programmed time limit. In this case, the CPU has read all of the characters from the FIFO, but the last character received has started the count. If there is no new data during the programmed time interval, the counter ready bit will get set, and an interrupt can be generated.

Writing the appropriate command to the Command Register enables the Time-out mode. Writing an ‘Ax’ to CR will invoke the Time-out mode for that channel. Writing a 0xCx to CR will disable the Time-out mode. CTPU and CTPL should be loaded with a count-down value that, with the selected clock, will generate a time period greater than the normal receive character period. The Time-out 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 1 C/T clock, reloaded with the value in CTPU and CTPL 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, interrupt arbitration for the C/T will begin. Invoking the ‘Set Time-out Mode On’ command, CRx = ‘Ax’, clears the counter ready bit and stops the counter until the next character is received.

Exiting the Time-out mode, CR command 0x0C, will clear the counter ready bit.

7.4.8 Arbitrating interrupt structure

Remark: The one or two bus cycle identification of the interrupt source, type, FIFO fill level, and channel is the principle advantage of this system. It is equally effective in both interrupt of polled method of UART service. Its most efficient use is with the use of the interrupt vector and IACKN. IACKN is totally effective in the polled mode. The intelligence of this system may be completely defeated by merely setting the arbitration value in the ICR to 0x00 and not using the CIR. One would then rely on traditional interrupt service by searching and testing various status registers on the chips assertion of IRQN.

The center point of this system is the value contained in Current Interrupt Register (CIR). This register contains the FIFO byte count, a unique identification number for each interrupt source and a channel number for multichannel UARTs. The CIR is continuously

updated by an arbitration or bidding unit that selects the highest value presented by the various interrupt sources that may be enabled at any time. The values used are under the control of the software and the FIFO fill counters.

The arbitration is exercised over the several systems within the UART that may generate an interrupt. These will be referred to as ‘interrupt sources’. There are 11 in all and may of those have several sublevels. In general the arbitration is based on the fill level of the receiver FIFO or the empty level of the transmitter FIFO. The FIFO levels are encoded into an 8-bit number, which is concatenated to the channel number and source identification code. All of this is compared (via the bidding or arbitration process) to a user defined ‘threshold’. Whenever a source exceeds the numerical value of the threshold the interrupt will be generated.

Interrupt sources that do not have a FIFO are each provided with a ‘programmable field’ that will determine their importance in the arbitration and type identification process. (See Table 4.)

At the time of interrupt acknowledge (IACKN) the source which has the highest bid (not necessarily the source that caused the interrupt to be generated) will be captured in a Current Interrupt Register (CIR). This register will contain the complete definition of the interrupting source: channel, types of interrupt (receiver, transmitter, change-of-state, and so on) and FIFO fill level. The value of the bits in the CIR are used to drive the interrupt vector and global registers such that controlling processor may be steered directly to the proper service routine. A single read operation to the CIR provides all the information needed to qualify and quantify the most common interrupt sources.

Using IACKN automatically provides an update of the CIR and presents that data as an interrupt vector or an interrupt vector modified. Without the use of the IACKN, and update CIR command will be issued followed by a read of the CIR.

The interrupt sources are:

- Receiver without error
- Receiver with error for each channel
- Receiver Watchdog Time-out Event
- Transmitter
- Change in break received status per channel
- Rx loopback error
- Change of state on channel input pins
- Xon/Xoff character recognition
- Counter/Timer
- Address character recognition
- No interrupt active (very useful in polled service and as a test value to terminate interrupt service) when CIR = 0x00

Transmit FIFO empty level and Receiver FIFO fill levels are unique for each channel and may be set at any level.

Associated with the interrupt system are the Interrupt Mask Register (IMR) and the Interrupt Status Register (ISR) resident in each UART. Programming of the IMR selects which of the above sources may enter the arbitration process. The IMR enables the interrupt. Only the bidders in the ISR whose associated bit in the IMR is set to one (1) will be permitted to enter the arbitration process. The ISR can be read by the host CPU to determine all currently active interrupting conditions. For convenience of reading the ISR, the MR1[6] bit, when set, allows the reading of the ISR masked by the bits of the IMR.

7.4.8.1 Enabling and activating interrupt sources

An interrupt source becomes enabled when writing a one to the proper Interrupt Mask Register bit (IMR) activates its interrupt capability. An interrupt source can never generate an IRQN or have its 'bid' or interrupt number appear in the CIR unless the source has been enabled by the appropriate bit in an IMR.

An interrupt source is active if it is presenting its bid to the interrupt arbiter for evaluation. Most sources have simple activation requirements. The watchdog timer, break received, Xon/Xoff or Address Recognition and change-of-state interrupts become active when the associated events occur and the arbitration value generated thereby exceeds the threshold value programmed in the Interrupt Control Register (ICR).

The transmitter and receiver functions have additional controls to modify the condition upon which the initiation of interrupt bidding begins: the TxINT and RxINT fields of the MR0 and MR2 registers. These fields can be used to start bidding or arbitration when the RxFIFO is not empty, 50 % full, 75 % full, or 100 % full. For the transmitter it is not full, 50 % empty, 75 % empty, and empty.

Example: To increase the probability of transferring the contents of a nearly full RxFIFO, do not allow it to start bidding until 50 % or 75 % full. This will prevent its relatively high priority from winning the arbitration process at low fill levels. A high threshold level could accomplish the same thing, but may also mask out low priority interrupt sources that must be serviced. Note that for fast channels and/or long interrupt latency times using this feature should be used with caution since it reduces the time the host CPU has to respond to the interrupt request before receiver overrun occurs.

7.4.8.2 Setting interrupt priorities

The bid or interrupt number presented to the interrupt arbiter is composed of character counts, channel codes, fixed and programmable bit fields. The interrupt values are generated for various interrupt sources as shown in Table 4. The value represented by the bits 11:4 in Table 4 are compared against the value represented by the 'Threshold'. The 'Threshold', bits 10:0 of the ICR (Interrupt Control Register), is aligned such that bit 0 of the threshold is compared to bit 1 of the interrupt value generated by any of the sources. Whenever the value of the interrupt source is greater than the threshold the interrupt will be generated.

The codes form bits 4:1 drive part of the interrupt vector modification and the Global Interrupt Type Register. The codes are unique to each source type and identify them completely.

Table 4: Interrupt values

Type	Bit 11 to bit 4		Bit 3	Bit 2	Bit 1	Bit 0
receiver without error	RxFIFO filled Byte Count		0	0	1	channel number
receiver with error	RxFIFO filled Byte Count		1	0	1	channel number
receiver watchdog	RxFIFO filled Byte Count		1	0	0	channel number
transmitter	TxFIFO empty Byte Count		0	1	0	channel number
change of break	programmed field	1	1	1	0	channel number
Rx loopback error	programmed field	1	1	1	1	channel number
change-of-state	programmed field	0	1	1	0	Port 0 or 1
Xon/Xoff	programmed field	0	1	1	1	channel number
Counter/Timer	programmed field	1	0	0	0	Counter 0 or 1
address recognition	programmed field	1	0	1	1	channel number
no interrupt	0	0	0	0	0	-
threshold	bits 7:0 of Interrupt Control Register (ICR)	0	0	0	0	0

7.4.8.3 Interrupt arbitration and IRQN generation

Interrupt arbitration is the process used to determine that an interrupt request should be presented to the host. The arbitration is carried out between the ‘Interrupt Threshold’ and the ‘sources’ whose interrupt bidding is enabled by the IMR. The interrupt threshold is part of the ICR (Interrupt Control Register) and is a value programmed by the user. The ‘sources’ present a value to the interrupt arbiter. That value is derived from four fields: the channel number, type of interrupts source, FIFO fill level, and a programmable value. The interrupt request (IRQN) will be asserted only when one or more of these values exceeds the threshold value in the interrupt control register will.

Following assertion of the IRQN the host will either assert IACKN (Interrupt Acknowledge) or will use the command to ‘Update the CIR’. At the time either action is taken the CIR will capture the value of the source that is prevailing in the arbitration process. (Call this value the winning bid.)

The SCLK drives the arbitration process. It evaluates the 12 bits of the arbitration bus at $\frac{1}{2}$ the SCLK rate developing a value for the CIR every two SCLK cycles. New arbitration values presented to the arbitration block during an arbitration cycle will be evaluated in the next arbitration cycle.

For sources other than receiver and transmitters the user may set the high order bits of an interrupt sources bid value, thus tailoring the relative priority of the interrupt sources. The fill level of their respective FIFOs controls the priority of the receivers and transmitters. The more filled spaces in the RxFIFO the higher the bid value; the more empty spaces in the TxFIFO the higher its priority. Channels whose programmable high order bits are set will be given interrupt priority higher than those with zeros in their high order bits, thus allowing increased flexibility. The transmitter and receiver bid values contain the character counts of the associated FIFOs as high order bits in the bid value. Thus, as a receivers RxFIFO fills, it bids with a progressively higher priority for interrupt service. Similarly, as empty space in a transmitters TxFIFO increases, its interrupt arbitration priority increases.

The programmable fields allow the software to adjust the authority or value of the bid for those devices not having a FIFO.

For example: The break condition is sometimes used to signal a starting point in a continuous stream of data. A Continuous running weather report or stock market 'ticker-tape' report needs breaks in the data so that a receiver knows where the data starts. Once start of the break is detected it is important to reset the 'change of break' interrupt so that this bit can signal the condition of the break ending. This is signaled by the SC28L201 the setting another change of break event in the ISR. Since it is assumed the data will be starting very soon after the end of break it is important to give the change of break condition a high priority. This may be accomplished by setting the arbitration value for the 'change of break' to a high value. The value in the 'change of break programmable field' in Table 4 would be 0x7F.

7.4.8.4 IACKN cycle, update CIR

When the host CPU responds to the interrupt, it will usually assert the IACKN signal LOW. This will cause the UART to generate an IACKN cycle in which the condition of the interrupting device is determined. When IACKN asserts, the last valid interrupt number is captured in the CIR. The value captured presents most of the important details of the highest priority interrupt at the moment the IACKN (or the 'Update CIR' command) was asserted.

The UART will respond to the IACKN cycle with an interrupt vector. The interrupt vector may be a fixed value, the content of the Interrupt Vector Register, or when 'Interrupt Vector Modification' is enabled via ICR, it may contain codes for the interrupt type and/or interrupting channel. This allows the interrupt vector to steer the interrupt service **directly** to the proper service routine. The interrupt value captured in the CIR remains until another IACKN cycle occurs or until an 'Update CIR' command is given to the UART. The interrupting channel and interrupt type fields of the CIR set the current 'interrupt context' of the UART.

The CIR drives the several Global Interrupt Information registers that appear at fixed positions in the register address map. For example, a read of the Global Byte Count return the value of the FIFO fill level of the receiver or transmitter depending on which one is interrupting. Global registers exist to facilitate qualifying the interrupt parameters and for writing to and reading from FIFOs without explicitly addressing them. They are essentially indirect addresses to characteristics of the interrupting source.

The CIR will load with 0x00 if IACKN or Update CIR is asserted when the arbitration circuit is **not** asserting an interrupt. In this condition there is no arbitration value that exceeds the threshold value. When Interrupt vector modification is active in this situation the interrupt vector bits associated with the CIR will all be zero.

7.4.8.5 Global registers

The Global Registers, 10 in all, are driven by the interrupt system. They are defined by the content of the CIR (Current Interrupt Register) as a result of an interrupt arbitration. In other words they are indirect registers pointed to by the content of the CIR. The list of global register follows:

- GIBCR: the byte count of the interrupting FIFO
- GICR: channel number of the interrupting channel

  The GCIR will read 0 except for Counter/Timer 1 and I/O[n]B pins
- GITR: type identification of interrupting channel
- GRxFIFO: pointer to the interrupting receiver FIFO

- GTxFIFO: pointer to the interrupting transmitter FIFO

A read of the GRxFIFO will give the content of the RxFIFO that presently has the highest bid value. The purpose of this system is to enhance the efficiency of the interrupt system.

7.4.8.6 Polling (normal and using the CIR)

The ‘arbitrating interrupt system’ will reduce the polling overhead to one or two bus cycles depending on the use of IACKN. This interrupt system does not need either the IRQN (interrupt request) or the IACKN (interrupt acknowledge) signal. It only requires an update CIR command. Recall that if nothing has exceeded preset levels, including 0x00, the CIR will read 0x00, meaning no service is required. A minimal polling loop would be: update CIR, read CIR, if 0x00 exit.

Many users prefer polled to interrupt driven service where there are not a large number of fast data channels and/or the host CPUs other interrupt overhead is low. The UART is functional in this environment.

The most efficient method of polling is the use of the ‘update CIR’ command (with the interrupt threshold set to zero) followed by a read of the CIR. This dummy write cycle will perform the same CIR capture function that an IACKN falling edge would accomplish in an interrupt driven system. A subsequent read of the CIR, at the same address, will give information about an interrupt, if any. If the CIR type field contains 0s, no interrupt is awaiting service. If the value is non-zero, the fields of the CIR may be decoded for type; channel and character count information. Optionally, the global interrupt registers may be read for particular information about the interrupt status or use of the global RxD and TXD registers for data transfer as appropriate. The interrupt context will remain in the CIR until another update CIR command or an IACKN cycle is initiated by the host CPU occurs. The CIR loads with 0x00 if Update CIR is asserted when the arbitration circuit has **not** detected an arbitration value that exceeds the threshold value of the ICR. The global registers and CIR may be used as ‘vectors’ to the service type required.

Traditional methods of polling status registers may also be used. Their lower efficiency may be greatly offset by use of the UCIR (Update Current Interrupt Register) command and the read of the CIR. They reduce the many reads and tests of status registers to only one read and one write. This would normally be accomplished by setting the interrupt threshold to zero. Then the moment any system within the UART needs service the next poll of the CIR would return a non zero value and the type field will inform the processor which of the possible 11 systems needs service. In the case of the FIFOs the number of bytes to be written or read is also available.

7.4.9 Character and address recognition (also used for multi-drop, Xon/Xoff systems)

Three programmable characters are provided for the character recognition for each channel. Each character is general purpose in nature and may be set to only cause an interrupt or to initiate some rather complex operations specific to ‘multi-drop’ address recognition or in-band Xon/Xoff flow control.

Character recognition system continually examines the incoming data stream. Upon the recognition of a character bits appropriate for the character recognized are set in the Xon/Xoff Interrupt Status Register (XISR) and in the Interrupt Status Register (ISR). The setting of these bit(s) will initiate any of the automatic sequences or and/or an interrupt that may have enabled via the MR3 register.

Remark: Reading the XISR Clears the status bits associated with the recognition.

The characters of the recognition system are fully programmable. The Xon/Xoff characters will be set to the standard characters if the hardware or software reset is used.

The character recognition circuits are basically designed to provide general-purpose character recognition. Additional control logic has been added to allow for Xon/Xoff flow control and for recognition of the address character in the multi-drop or Wake-up mode. This logic also allows for the generation of interrupts in either the general-purpose recognition mode or the specific conditions mentioned above.

The generality of the above provides a modicum of compatibility to BOP (Bit Oriented Protocol) where the generation and detection of 'flags' is required. Parts of usually synchronous BOP protocols (HDLC in particular) are beginning to show up in asynchronous formats.

7.4.9.1 Character stripping

The MR0[7:6] register provides for stripping (meaning not set to the RxFIFO) the characters used for character recognition. Recall that the character recognition may be conditioned to control several aspects of the communication. However this system is first a character recognition system. The status of the various states of this system is reported in the XISR and ISR registers. The character stripping of this system allows for the removal of the specified control characters from the data stream: two for the Xon /Xoff and one for the wake-up. Via control in the MR0[7:6] register these characters may be discarded (stripped) from the data stream when the recognition system 'sees' them or they may be sent on the RxFIFO. Whether they are stripped or not the recognition system will process them according to the action requested; flow control, wake-up, interrupt generation, and so on. Care should be exercised in programming the stripping option if noisy environments are encountered. If a normal character were corrupted to an Xoff character the transmitter would be stopped. If that character were now stripped from the FIFO stack, then that stripping action would make it difficult to determine the cause of transmitter stopping.

When character stripping is invoked and a recognition character is received that has **an error bit set** that character is sent to the RxFIFO even though character stripping is active.

7.4.10 Flow control (Xon/Xoff)

This section describes in-band flow control or Xon/Xoff signaling. For the RTS/CTS hardware (out-of-band) control see MR1[7] and MR2[4] descriptions.

The flow control is accomplished via the character recognition system giving recognition information to the flow control processor. Xon and Xoff are special characters used by a receiver to start and stop the remote transmitter that is sending it data. Several modes of manual and automatic flow control are available by programming mode control.

7.4.10.1 Mode control

Xon/Xoff mode control is accomplished via the MR3[3:2].

00 — Xon/Xoff processing disabled. The host will control Rx, Tx.

01 — Auto Tx control. Tx is stopped/started when Xoff/Xon is received.

10 — Auto Rx control. Receiver commands Tx to send Xoff at trigger level.

11 — Auto Rx and Tx control. Receiver commands Tx to send Xoff as the receiver fills and commands the Tx to send Xon when Rx FIFO fill level is lowered. This results in total automatic control. No processor interrupt is required.

Note that MR3[7] controls the stripping of Xoff/Xon characters.

0 — Xon/Xoff characters are sent to the Rx FIFO

1 — Xon/Xoff characters are discarded

The MR3[7] functions regardless of the setting of MR3[3:2]. This allows for general purpose character recognition and processing (see Section 7.4.9.1 “Character stripping”).

7.4.10.2 Xon/Xoff characters

The programming of these characters is usually done individually. The standard Xon/Xoff characters are: Xon is 0x11, Xoff 0x13. Any enabling of the Xon/Xoff functions will use the contents of the Xon and Xoff character registers as the basis on which recognition is predicated.

7.4.10.3 Host mode (least efficient)

When neither the auto-receiver or auto-transmitter modes are set, the Xon/Xoff logic is operating in the host mode. In host mode, all activity of the Xon/Xoff logic is initiated by commands to the CRx. The Xoff command forces the transmitter to disable exactly as though an Xoff character had been received by the RxFIFO. The transmitter will remain disabled until the chip is reset or the CR[7:3] = 1 0110 (Xoff resume) command is given. In particular, reception of an Xon or disabling or re-enabling the transmitter will **not** cause resumption of transmission. Redundant CRTXxx commands, that is ‘CRTXon, CRTXon’, are harmless, although they waste time. A CRTXon may be used to cancel a CRTXoff (and vice versa) but both may be transmitted depending on the command timing with respect to that of the transmitter state machine.

7.4.10.4 Auto-transmitter mode

When a channel receiver loads an Xoff character into the RxFIFO, the channel transmitter will finish transmission of the current character and then stop transmitting. A transmitter so idled can be restarted by the receipt of an Xon character by the receiver or by a hardware or software reset. The last option results in the loss of the un-transmitted contents of the TxFIFO. When in this mode and waiting for an Xon signal from the receiver, the CR (Command Register) commands for the transmitter are not effective.

While idle in this mode and waiting for an Xon signal from the receiver, data may be written to the TxFIFO and it continue to present its fill level to the interrupt arbiter and maintains the integrity of its status registers.

Use of ‘00’ as an Xon/Xoff character is complicated by the Receiver break operation which loads a ‘00’ character on the RxFIFO. The Xon/Xoff character detectors do not discriminate in this case from an Xon/Xoff character received through the RXD pin.

Remark: To be recognized as an Xon or Xoff character, the receiver must have room in the RxFIFO to accommodate the character. An Xon/Xoff character that is received resulting in a receiver overrun does not effect the transmitter nor is it loaded into the RxFIFO, regardless of the state of the Xon/Xoff transparency bit, MR3[7].

7.4.10.5 Receiver mode

Since the receiving FIFO resources in the UART are limited, some means of controlling a remote transmitter is desirable in order to lessen the probability of receiver overrun. The UART provides two methods of controlling the data flow. There is a hardware-assisted means of accomplishing control, the so-called out-of-band flow control, and an in-band flow control method.

The out-of-band flow control is implemented through the CTSN-RTSN signaling via the I/O ports. The operation of these hardware handshake signals is described in the receiver and transmitter discussions.

In-band flow control is a protocol for controlling a remote transmitter by embedding special characters within the message stream, itself. Two characters, Xon and Xoff, which do not represent normal printable character take on flow control definitions when the Xon/Xoff capability is enabled. Flow control characters received may be used to gate the channel transmitter on and off. This activity is referred to as Auto-transmitter mode. To protect the channel receiver from overrun, fixed fill levels (hardware set at 240 characters) of the RxFIFO may be employed to automatically insert Xon/Xoff characters in the transmitters data stream. This mode of operation is referred to as Auto-receiver mode. Commands issued by the host CPU via the CR can simulate all these conditions.

7.4.10.6 Auto-receive and transmit (most efficient)

This is a combination of both modes.

Remark: Xon/Xoff characters: The Xon/Xoff character with errors will be accepted as valid. The user has the option sending or not sending these characters to the FIFO. Error bits associated with Xon/Xoff will be stored normally to the receiver FIFO.

The channel's transmitter may be programmed to automatically transmit an Xoff character without host CPU intervention when the RxFIFO fill level exceeds a fixed limit (240). In this mode it will transmit an Xon character when the RxFIFO level drops below a second fixed limit (16). A character from the TxFIFO that has been loaded into the TXD shift register will continue to transmit. Character(s) in the TxFIFO that have not been loaded to the transmitter shift register are unaffected by the Xon or Xoff transmission. They will be transmitted after the Xon/Xoff activity concludes.

If the fill level condition that initiates Xon activity negates before the flow control character can begin transmission, the transmission of the flow control character will not occur.

Hardware keeps track of Xoff characters sent that are not rescinded by an Xon. This logic is reset by writing MR3[3:2] to '00'. If the user drops out of Auto-receiver mode while the XISR shows Xoff as the last character sent, the Xon/Xoff logic would **not** automatically send the negating Xon.

The disable CRTX command (of the Command Register, 0x18) can be used to cleanly terminate any pending CRTX commands.

Remark: In **no** case will an Xon/Xoff character transmission be aborted. Once the character is loaded into the TX Shift Register, transmission continues until completion or a chip reset or transmitter reset is encountered. The kill CRTX command has no effect in either of the Auto modes.

7.4.10.7 Xon/Xoff interrupts

The Xon/Xoff logic generates interrupts **only** in response to recognizing either of the characters in the XonCR or XoffCR (Xon or Xoff Character Registers). The transmitter activity initiated by the Xon/Xoff logic or any CR command does **not** generate an interrupt. The character comparators operate regardless of the value in MR3[3:2]. Hence the comparators may be used as general-purpose character detectors by setting MR3[3:2] = 00 and enabling the Xon/Xoff interrupt in the IMR.

The UART can present the Xon/Xoff recognition event to the interrupt arbiter for IRQN generation. The IRQN generation may be masked by setting bit 4 of the Interrupt Mask Register, IMR. The bid level of an Xon/Xoff recognition event is controlled by the Bidding Control Register X, BCRx, of the channel. The interrupt status can be examined in ISR[4]. If cleared, no Xon/Xoff recognition event is interrupting. If set, an Xon or Xoff recognition event has been detected. The X Interrupt Status Register (XISR) can be read for details of the interrupt and to examine other, non-interrupting, status of the Xon/Xoff logic. Refer to Section 8.3.4 “Xon/Xoff Interrupt Status Register (XISR)”.

The character recognition function and the associated interrupt generation is disabled on hardware or software reset.

7.4.10.8 Multi-drop, Wake-up, or 9-bit mode

This mode is used to address a particular UART among a group connected to the same serial data source. Normally it is accomplished by redefining the meaning of the parity bit such that it indicates a character as address or data. While this method is fully supported in the SC28L201, it also supports recognition of the character itself. Upon recognition of its address the receiver will be enabled and data loaded onto the RxFIFO.

Further, the Address recognition has the ability, if so programmed, to disable (not reset) the receiver when an address is seen that is not recognized as its own. The particular features of ‘Auto Wake and Auto Doze’ are described in the detailed descriptions under Section 7.4.7 “Receiver operation”.

Remark: Care should be taken in the programming of the character recognition registers. Programming 0x00, for example, may result in a break condition being recognized as a control character. This will be further complicated when binary data is being processed.

7.5 Programming the host interface

The SC28L201 is designed for a very close compatibility with legacy software written for other Philips/Signetics 2 channel UARTs. The part will initialize to the SC28L91 function. This function is controlled in the lower 16 address positions.

A reset (both hardware and software) will return the part to this mode with the control registers set for 9600 baud, 8 bits, no parity and one stop bit. Interrupt will be set for Receiver Ready and transmitter Empty. Transmitters and receivers will not be enabled. Basic operation should be obtained by a single write of 0xE0 to the Command Register. That will enable the receiver and transmitter.

Addressing outside of the lower 16 address spaces will enable all the advanced features. In general, before calling legacy code, advanced features should be disabled (character stripping, for example).

Writing control words into the appropriate registers programs the operation of the UART. Operational feedback is provided via status registers that can be read by the CPU. The addressing of the registers is described in Section 9 “Register maps”.

The contents of certain control registers are initialized to zero on reset. Care should be exercised if the contents of channel specific register are changed during transmission and reception of data, since certain changes may cause data corruption.

For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped. This is far more a system requirement than a SC28L201 requirement.

The UART has 3 Mode Registers (MR0, MR1, MR2) which control the basic configuration of the channel. Mode, command, clock select, and status registers provide independent operation and control for receivers and transmitters. Refer to Section 8 “Register description and programming note” for register bit descriptions.

8. Register description and programming note

Programmers may use either of two register sets or mix the features of each. It is suggested that only the extended register set be used in new designs. However if a system needed to use a block of communications code written for an older system then that code could merely be called. This is similar to calling a DOS program in a Windows environment.

8.1 Registers that control global properties of the SC28L201

8.1.1 Global Configuration Control Register (GCCR)

Remark: This is a very important register. It should be the first register addressed during initialization.

Table 5: GCCR - Global Configuration Control Register (address 0x66) bit description

Bit	Symbol	Description
7:6		DACKN assertion. 00 = 2 SCLK to 3 SCLK 01 = 1 SCLK to 2 SCLK 10 = ½ SCLK to 1 SCLK 11 = reserved Motorola bus cycle time can be controlled by selecting a DACKN assertion time based on X1/SCLK speed. The time programmed should not be less than the minimum read or write pulse width. See examples below in Table 6 “DACKN assertion time”.
5:3	-	reserved; set to 0

Table 5: GCCR - Global Configuration Control Register (address 0x66) bit description

Bit	Symbol	Description
2:1	IVC	Interrupt Vector Control 00 = no interrupt vector 01 = IVR[7:0] 10 = IVR[7:1] + channel code 11 = IVR[7:5] + interrupt type + channel code The IVC field controls if and how the assertion of IACKN (the interrupt acknowledge pin) will form the interrupt vector for the UART. If 00, no vector will be presented during an IACKN cycle. The bus will be driven HIGH (0xFF). If the field contains a 01, the contents of the IVR, Interrupt Vector Register, will be presented as the interrupt vector without modification. If IVC = 10, the channel code will replace the LSB of the IVR; if IVC = 11 then a modified interrupt type and channel code replace the 3 LSBs of the IVR. **Remark:** The modified type field IVR[2:1] is: 10: Receiver without error 11: Receiver with error 01: Transmitter 00: all remaining sources
0		Interrupt status masking 0 = ISR unmasked 1 = ISR read masked by IMR This bit controls the readout mode of the Interrupt Status Register, ISR. If set, the ISR reads the current status masked by the IMR, that is, only interrupt sources enabled in the IMR can ever show a '1' in the ISR. If cleared, the ISR shows the current status of the interrupt source without regard to the Interrupt Mask setting.

Table 6: DACKN assertion time

X1/SCLK	Number of SCLK cycles	Delay
3.6864 MHz	½ to 1	136 ns to 272 ns
7.3728 MHz	½ to 1	68 ns to 136 ns
14.7456 MHz	½ to 1	34 ns to 68 ns
29.4912 MHz	1 to 2	34 ns to 68 ns
33.1776 MHz	2 to 3	60 ns to 90 ns
44.2368 MHz	2 to 3	46 ns to 68 ns

8.1.2 Special Feature and Status Register (SFSR A and B)

Table 7: SFSR - Special Feature and Status Register (address 0x45) bit description

Bit	Symbol	Description
7:4	-	reserved
3		Loopback error; status of loopback error check 0 = no 1 = yes (read only) A '1' indicates a loopback error occurred, which will be entered for interrupt arbitration. It can be cleared by the processor by a write to this register with D[3] equal to '1'.
2:1		Remote loopback error check 00 = disabled 01 = enabled, RxC ← TxC 10 = enabled, RxC ← TxCN Certification of returned data as Valid. (This feature implies the transmitted data is being returned by the remote receiver.) Sets automatic checking of returned data. This mode stores transmitted data and compares it to data returned from the remote receiver. It is used where relative short delay times are available, up to two characters in time. This mode will totally relieve the processor of this task where certainty of transmission and reception is required. The transmitted data is looped back by the remote station with a half-bit time delay. The local transmitted data is internally sent to the local receiver for comparison. An interrupt is generated in the case of an error (data mismatch, parity or framing). 00 = the checking is disabled 01 = return data is clocked in on rise of TxC 10 = return data is clocked on of rise of TxCN 00 = reserved
0	-	reserved

8.1.3 Test and Revision Register (TRR)

Table 8: TRR - Test and Revision Register (address 0x67) bit description

Bit	Symbol	Description
7	TRR[7]	Test 2 enable Bypass divide-by-16 counter in all TxC and RxC.
6:0	TRR[6:0]	Chip Revision code Indicates the revision of the chip. Initial code will be '000 0000'. The revision code bits [6:0] are hard-wired. The default setting of the test bits is all zeros.

8.1.4 Scan Test Control Register (STCR)

Table 9: STCR - Scan Test Control Register (address 0x77) bit description

Bit	Symbol	Description
7:3	-	
2		Memory test
1		Scan test
0		Iddq test. Turns off all pull-up devices on the I/O pins.

8.1.5 System Enable Status register A and B (SES)

This register reports the enabled status of the several sub-systems in the UART. These systems are sometimes controlled by the state machines of the receiver FIFOs.

Table 10: SES - System Enable Status register (address 0x40) bit description

Bit	Symbol	Description
7:6	-	reserved; set to 0
5		Transmitter enabled 0 = no 1 = yes
4		Receiver enabled 0 = no 1 = yes
3		Watchdog timer 0 = no 1 = yes
2		Address recognition 0 = no 1 = yes
1		Xon 0 = no 1 = yes
0		Xoff 0 = no 1 = yes

8.1.6 Enhanced Operation Status register (EOS)

This register reports the status of the Enhanced operation in several sub-systems in the UART.

Table 11: EOS - Enhanced Operation Status register (address 0x12) bit description

Bit	Symbol	Description
7	-	reserved
6		I/O port operation 0 = default 1 = enhanced
5		reserved
4		Counter/Timer 0 clock select 0 = default 1 = enhanced
3		not used; returns ‘0’ on read
2		Rx/Tx clock selection 0 = default 1 = enhanced
1		not used; returns ‘0’ on read
0		FIFO interrupt level control 0 = default 1 = enhanced

8.2 UART registers

Remark: These registers are generally concerned with formatting, transmitting and receiving data.

The user must exercise caution when changing the mode of running receivers, transmitters, PBRG or counter/timers. The selected mode will be activated immediately upon selection, even if this occurs during the reception or transmission of a character. It is also possible to disrupt internal controllers by changing modes at critical times, thus rendering later transmission or reception faulty or impossible.

An exception to this policy is switching from auto-echo or remote loopback modes to normal mode. If the deselecting occurs just after the receiver has sampled the stop bit (in most cases indicated by the assertion of the channels RxRDY bit) and the transmitter is enabled, the transmitter will remain in auto-echo mode until the end of the transmission of the stop bit.

8.2.1 Mode Register 0 (MR0)

MR0 can be accessed directly at H'20' and 0x20 in the Extended section of the address map, or by means of the 'MR Pointers' at the 0x00 used by legacy code.

Table 12: MR0 - Mode Register 0 (address 0x20) bit description

Bit	Symbol	Description
7	MR0[7] [1]	Rx Watchdog. Fixed length watchdog timer. 0 = disable 1 = enable This bit controls the receiver watchdog timer. 0 = disable, 1 = enable. When enabled, the watchdog timer will generate a receiver interrupt if the receiver FIFO has not been accessed within 64 bit times of the receiver 1× clock. This is used to alert the control processor that data is in the RxFIFO that has not been read. This situation may occur when the byte count of the last part of a message is not large enough to generate an interrupt.
6	MR0[6]	RxINT[2]. Bit 2 of receiver FIFO interrupt level. This bit along with Bit 6 of MR1 sets the fill level of the 8 byte FIFO that generates the receiver interrupt.
	MR0[6] and MR1[6]	Note that this control is split between MR0 and MR1. This is for backward compatibility to the SC2692 and SCC2681. For the receiver, these bits control the number of FIFO positions filled 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. See Table 13 and Table 14.
5:4	MR0[5:4]	TxINT[1:0]. Transmitter interrupt fill level. 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 completely empty. As soon as one byte is loaded, it is no longer empty and hence will withdraw its interrupt request. See Table 15 and Table 16.
3	MR0[3]	FIFO size. Selects between 8-byte or 256-byte FIFO structure. 0 = 8 bytes 1 = 256 bytes
2	MR0[2] [2]	Baud rate extended II. (Legacy baud rate group selection.) 0 = Normal mode 1 = Extended Mode II
1	MR0[1] [2]	reserved; set to 0. (Legacy baud rate group selection.)
0	MR0[0] [2]	Baud rate extended I. (Legacy baud rate group selection.) 0 = Normal mode 1 = Extended Mode I

[1] This bit control is duplicated at WCXER[7:6], the Watchdog, Character, Address and X Enable Register.

[2] These bits are used to select one of the six baud rate groups. Combinations of MR0[2:0] other than those shown below should not be used.

a) Normal Mode: MR0[2:0] = 000

b) Extended Mode I: MR0[2:0] = 001

c) Extended Mode II: MR0[2:0] = 100

Table 13: Receiver FIFO interrupt fill level MR0[3] = 0

MR0[6]	MR1[6]	Interrupt condition
0	0	1 or more bytes in FIFO (RxRDY)
0	1	3 or more bytes in FIFO
1	0	6 or more bytes in FIFO
1	1	8 bytes in FIFO (Rx full)

Table 14: Receiver FIFO interrupt fill level MR0[3] = 1

MR0[6]	MR1[6]	Interrupt condition
0	0	1 or more bytes in FIFO (RxRDY)
0	1	128 or more bytes in FIFO
1	0	192 or more bytes in FIFO
1	1	256 bytes in FIFO (Rx full)

Table 15: Transmitter FIFO interrupt fill level MR0[3] = 0

MR0[6]	MR1[6]	Interrupt condition
0	0	8 bytes empty (Tx EMPTY)
0	1	4 or more bytes empty
1	0	6 or more bytes empty
1	1	1 or more bytes empty (TxRDY)

Table 16: Transmitter FIFO interrupt fill level MR0[3] = 1

MR0[6]	MR1[6]	Interrupt condition
0	0	256 bytes empty (Tx EMPTY)
0	1	128 or more bytes empty
1	0	192 or more bytes empty
1	1	1 or more bytes empty (TxRDY)

8.2.2 Mode Register 1 (MR1)

MR1 can be accessed directly at 0x21 in the Extended section of the address map, or by means of the MR Pointers at 0x00 used by legacy code.

Table 17: MR1 - Mode Register 1 (address 0x21) bit description

Bit	Symbol	Description
7		RxRTS control. Receiver Request-to-Send (hardware flow control) 0 = off 1 = on This bit controls the deactivation of the RTSN output (I/O2) by the receiver. The I/O2 output is asserted and negated by commands applied via the command register or through the setting of the OPR register bits. MR1[7] = 1 enables the receiver state machine to control the state of the I/O2 (where the RTSN function is assigned) to be automatically negated (driven HIGH) upon receipt of a valid start bit if the receiver FIFO is 240 full or greater (for 8-byte mode the FIFO full signal is used). RTSN is reasserted when the FIFO fill level falls below 240 filled FIFO positions. This constitutes a change from previous members of Philips UART families where the RTSN function triggered on FIFO full. This behavior caused problems with PC UARTs that could not stop transmission at the proper time. **Remark:** When the FIFO is set to an 8-byte depth, the RTSN signaling is triggered on position 8 of the FIFO. The RTSN feature can be used to prevent overrun in the receiver by using the RTSN output signal to control the CTSN (see description of bit MR2[4] in Table 18 “MR2 - Mode Register 2 (address 0x22) bit description”) input of the transmitting device. It is not recommend to use the hardware flow control and the in-band (Xon/Xoff) flow control at the same time, although the UART hardware will allow it. To use the RTSN function: 1. Set MR1[7] to 1. 2. Set I/O0B as appropriate to logical 0. 3. Enable receiver.
6		Receiver interrupt control bit 1. See description under MR0[6]. (Writing to this register will reset the RxFIFO interrupt to the bit configuration of MR0 and MR1. Reading has no effect.) Refer to Table 13 and Table 14 in Section 8.2.1 “Mode Register 0 (MR0)”.
5		Error Mode. Error Mode Select and sub-modes. 0 = character 1 = block (entry or exit) This bit selects the operating mode of the three FIFOed status bits (FE, PE, and received break). In the character mode, status is provided on a character-by-character basis. The status applies only to the character at the output 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 output of the FIFO, since the last reset error command was issued. The Block Error mode has two-sub modes. These modes are controlled by the Command Register. The error is accumulated (as described above) at either the entry of the data in to the FIFO or on the exit (read of the FIFO). Of the two the setting of the error on the entry of the data into the FIFO gives the earliest warning of error data.

Table 17: MR1 - Mode Register 1 (address 0x21) bit description *…continued*

Bit	Symbol	Description
4:3		Parity Mode Select 00 = with parity 01 = force parity 10 = no parity 11 = multi-drop Special Mode 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.
2		Parity Type Select 0 = even 1 = odd This bit sets the parity type (odd or even) if the ‘with parity’ mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the ‘force parity mode is programmed it has no effect if the ‘no parity’ mode is programmed. In the special ‘Wake-up’ mode, it selects the polarity of the A/D bit. The parity bit is used to an address or data byte in the ‘Wake-up’ mode.
1:0		Bits per Character Select. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits This field selects the number of data bits per character to be transmitted and received. This number does **not** include the start, parity, or stop bits.

8.2.3 Mode Register 2 (MR2)

MR2 can be accessed directly at 0x22 in the Extended section of the address map, or by means of the MR Pointer at 0x00 used by legacy code.

The MR2 register provides basic channel setup control that may need more frequent updating.

Table 18: MR2 - Mode Register 2 (address 0x22) bit description

Bit	Symbol	Description
7:6		Channel Mode Select. The UART can operate in one of the following four modes: 00 = Normal mode (default) 01 = Automatic Echo mode 10 = Local loopback mode 11 = Remote loopback mode Normal mode: the transmitter and receiver operating independently. Automatic Echo mode: this mode automatically retransmits the received data. The following conditions are true while in Automatic Echo mode: • Received data is re-clocked and re-transmitted on the TXD output. • The receiver clock is used for the transmitted data. • The receiver must be enabled, but the transmitter need not be enabled. • The TxRDY and Tx Idle status bits are inactive. • The received parity is checked, but is not regenerated for transmission, that is, transmitted parity bit is as received. • Character framing is checked, but the stop bits are retransmitted as received. Rx data is sent to RxFIFO. • A received break is echoed as received until the next valid start bit is detected. • CPU to receiver communication continues normally, but the CPU to transmitter link is disabled. Local loopback mode. In this mode: • The transmitter output is internally connected to the receiver input. • The transmitter's 1× clock is used for the receiver. • The TXD output is held HIGH. • The RXD input is ignored. • The transmitter must be enabled, but the receiver need not be enabled. • CPU to transmitter and receiver communications continue normally. Remote Loopback diagnostic mode. In this mode: • Received data is re-clocked and re-transmitted on the TXD output. • The receiver 1× clock is used for the transmitted data. • Received data is not sent to the local CPU, and the error status conditions are inactive. • The received parity is not checked and is not regenerated for transmission, that is, the transmitted parity bit is as received. • The receiver must be enabled, but the transmitter need not be enabled. • Character framing is not checked, and the stop bits are retransmitted as received. • A received break is echoed as received until the next valid start bit is detected.

Table 18: MR2 - Mode Register 2 (address 0x22) bit description *...continued*

Bit	Symbol	Description
5		Tx Controls RTS. Transmitter Request-to-Send Control. 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 negates (drives to logical 1) RTSN 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 of a message as follows: • Program Auto Reset mode: MR2[5] = 1. • Enable transmitter. • Assert RTSN via command. • Send message. • Verify the next-to-last character of the message is being sent by waiting until transmitter ready is asserted. Disable transmitter after the last character is loaded into the TxFIFO. • The last character will be transmitted and RTSN will be reset one bit time after the last stop bit. **Remark:** When the transmitter controls the RTSN pin, the meaning of the pin is **completely** changed. It has nothing to do with the normal RTSN/CTSN 'handshaking'. It is usually used to mean 'end of message' and to 'turn the line around' in simplex communications. From a practical point of view, the simultaneous use of Tx control of RTSN and Rx control is mutually exclusive. However, if this is programmed, the UART performs as required.
4		CTS Enable Tx. Clear-to-Send Control. The state 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 state of CTSN each time it is ready to begin sending 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. Please see the CTS description in Section 7.4.6 "Transmitter" for a description of direct software control of this pin, thus giving software a direct control of the transmitter.
3:0		Stop Bit Length Select. This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of $^{9}/_{16}$ bit through 2 bits can be programmed. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1× clock is used for the transmitter, MR2[1] = 0 selects one stop bit and MR2[1] = 1 selects two stop bits to be transmitted.

8.2.4 Mode Register 3 (MR3)

Table 19: MR3 - Mode Register 3 (address 0x23) bit description

Bit	Symbol	Description
7		Xon/Xoff transparency [1]. Xon/Xoff character stripping. 0 = flow control characters received are loaded onto the RxFIFO 1 = flow control characters received are **not** loaded onto the RxFIFO Control the handling of recognized Xon/Xoff characters. If set, the character codes are placed on the RxFIFO along with their status bits just as ordinary characters are. If the character is not loaded onto the RxFIFO, its received status will be lost unless the receiver is operating in the block error mode, see MR1[5] and the general discussion on receiver error handling. Interrupt processing is not effected by the setting of these bits. See Section 6.1.9 "Character and address recognition".
6		Address recognition transparency [1]. Xon/Xoff character stripping. 0 = address characters received are loaded to RxFIFO 1 = address characters are **not** loaded onto the RxFIFO Control the handling of recognized Address characters. If set, the character codes are placed on the RxFIFO along with their status bits just as ordinary characters are. If the character is not loaded onto the RxFIFO, its received status will be lost unless the receiver is operating in the block error mode, see MR1[5] and the general discussion on receiver error handling. Interrupt processing is not effected by the setting of these bits. See Section 6.1.9 "Character and address recognition".
5:4	-	reserved
3:2		In-band flow control mode. Xon/Xoff processing. 00 = host mode; only the host CPU may initiate flow control actions through the CR. 01 = auto transmitter flow control 10 = auto receiver flow control 11 = auto Rx and Tx flow control Control the Xon/Xoff processing logic. Auto Transmitter flow control allows the gating of Transmitter activity by Xon/Xoff characters received by the Channel's receiver. Auto Receiver flow control causes the Transmitter to emit an Xoff character when the RxFIFO has loaded to a depth of 240 characters. Draining the RxFIFO to a level of 128 or less causes the Transmitter to emit an Xon character. All transmissions require no host involvement. A setting other than 00 in this field precludes the use of the command register to transmit Xon/Xoff characters. **Remark:** Interrupt generation in Xon/Xoff processing is controlled by the IMR (Interrupt Mask Register) of the individual channels. The interrupt may be cleared by a read of the XISR, the Xon/Xoff Interrupt Status Register. Receipt of a flow control character will always generate an interrupt if the IMR is so programmed. The MR0[3:2] bits have effect on the automatic aspects of flow control only, not the interrupt generation.

Table 19: MR3 - Mode Register 3 (address 0x23) bit description *...continued*

Bit	Symbol	Description
1:0		Address Recognition control. 00 = default 01 = Auto wake 10 = Auto doze 11 = Auto wake and Auto doze This field controls the operation of the Address recognition logic. If the device is not operating in the special or Wake-up mode, this hardware may be used as a general-purpose character detector by choosing any combination except 00. Interrupt generation is controlled by the channel IMR. The interrupt may be cleared by a read of the XISR, the Xon/Xoff Interrupt Status Register. See further description in Section 7.4.7.6 “Wake-up mode”.

[1] If this bit is not ‘0’, the characters will be stripped regardless of bits [3:2] or [1:0].

8.2.5 Receiver Clock Select Register (RxCSR) and Transmitter Clock Select Register (TxCSR)

Table 20: RxCSR - Receiver Clock Select Register (address 0x30) and TxCSR - Transmitter Clock Select Register (address 0x31) bit description

Bit	Symbol	Description
7:6	-	reserved
5:0	-	Transmitter/Receiver Clock Select code; see Table 21.

Both registers consist of single 6-bit field that selects the clock source for the receiver and transmitter respectively. During a read the unused bits in this register read ‘000’. The ‘BRG’ baud rates (fixed BRG rates) shown in Table 21 are based on the SCLK crystal frequency of 14.7456 MHz. The baud rates shown in Table 21 will vary as the SCLK crystal clock varies. For example, if the SCLK rate is changed to 7.3728 MHz all the rates below will reduce by $\frac{1}{2}$.

Table 21: Rx and Tx Clock Select

SCLK maximum rate is 50 MHz. Data clock rates will follow exactly the ratio of the X1/SCLK to 14.7456 MHz

TX clock select code	Clock selection, SCLK = 14.7456 MHz	TX clock select code	Clock selection, SCLK = 14.7456 MHz
00 0000	BRG - 50	01 0000	BRG - 75
00 0001	BRG - 110	01 0001	BRG - 150
00 0010	BRG - 134.5	01 0010	BRG - 450
00 0011	BRG - 200	01 0011	BRG - 1800
00 0100	BRG - 300	01 0100	BRG - 2000
00 0101	BRG - 600	01 0101	BRG - 14.4 k
00 0110	BRG - 1200	01 0110	BRG - 19.2 k
00 0111	BRG - 1050	01 0111	BRG - 28.8 k
00 1000	BRG - 2400	01 1000	BRG - 57.6 k
00 1001	BRG - 4800	01 1001	BRG - 115.2 k
00 1010	BRG - 7200	01 1010	BRG - 230.4 k
00 1011	BRG - 9600	01 1011	BRG - 460.8 k
00 1100	BRG - 38.4 k	01 1100	BRG - 921.6 k
00 1101	Timer 0	01 1101	Timer 1
00 1110	I/O[3]A transmitter - 16× external (refer to Table 22)	01 1110	PBRG 0
00 1111	I/O[3]A transmitter - 1× external (refer to Table 22)	01 1111	MIDI rate 31.25 kHz; 1.66 % error
		11 0000 to 11 1101	reserved
		11 1110	PBRG 1
		11 1111	reserved

Table 22: External clock pin and external clock mode assignment

Tx/Rx CSR[5:0]	RxC (receiver external clock) channel
00 1110 I/O[3]A	I/O[4]A 16×
00 1111 I/O[3]A	I/O[4]A 1×

8.2.6 Command Register Extension (CRx)

CR is used to write commands to the UART.

Table 23: CRx - Command Register Extension (address 0x12) bit description

Bit	Symbol	Description
7		Lock Tx and Rx enables. 0 = lock Rx and Tx state. Prevents changing transmitter and receiver enable bits while writing to the lower 5 bits of the command register. Bits CR[6:5] are not changed. 1 = change Rx and Tx state. Allows the receiver and transmitter enable bits to be changed while issuing a command to the Command Register. **Remark:** Receiver or transmitter disable is not the same as receiver or transmitter reset. **Writes to the lower 5 bits of the CR would usually have CR[7] at '0'** in order to maintain the enable/disable condition of the receiver and transmitter. The bit provides a mechanism for writing commands to a channel, via CR[4:0], without the necessity of keeping track of or reading the current enable status of the receiver and transmitter.
6		Enable Tx. Enable transmitter. 0 = disable 1 = enable A one written to this bit enables operation of the transmitter. The TxRDY status bit will be asserted. When disabled by writing a zero to this bit, the command terminates transmitter operation and resets the TxRDY and Tx Idle status bits returning the transmitter to its idle state. However, if a character is being transmitted or if characters are loaded in the TxFIFO when the transmitter is disabled, the transmission of the all character(s) is completed before assuming the inactive state.
5		Enable Rx. Enable receiver. 0 = disable 1 = enable A one written to this bit enables operation of the receiver. The receiver immediately begins the search for and the verification the start bit. If a zero is written, this command terminates operation of the receiver immediately-a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. The data in the RxFIFO will be retained and may be read. If the receiver is re-enabled subsequent data will be appended to that already in the RxFIFO. If the special Wake-up mode is programmed, the receiver operates even if it is disabled (see Section 7.4.7.6 "Wake-up mode").

Table 23: CRx - Command Register Extension (address 0x12) bit description *...continued*

Bit	Symbol	Description
4:0		Command Register codes The encoded value of this field can be used to specify a single command as follows: 0 0000: no command 0 0001: reserved 0 0010: Reset receiver. Immediately resets the receiver as if hardware reset had been applied. The receiver is reset and the FIFO pointer is reset to the first location effectively discarding all unread characters in the FIFO. 0 0011: Reset transmitter. Immediately resets the transmitter as if a hardware reset had been applied. The transmitter is reset and the FIFO pointer is reset to the first location effectively discarding all un-transmitted characters in the FIFO. 0 0100: Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the Status Register (SR[7:4]). It is used in either character or block mode. In block mode it would normally be used after the block is read. 0 0101: Reset break change interrupt. Causes the break detect change bit in the Interrupt Status Register (ISR[2]) to be cleared to zero. 0 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 and the TxFIFO is empty then the break begins when transmission of the current character is completed. If there are characters in the TxFIFO, the start of break is delayed until all characters presently in the TxFIFO and any subsequent characters loaded have been transmitted. (Tx Idle must be true before break begins). The transmitter must be enabled to start a break. 0 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 is transmitted. 0 1000: Assert RTSN. Causes the RTSN output to be asserted (LOW). 0 1001: Negate RTSN. Causes the RTSN output to be negated (HIGH). **Remark:** The two commands above actually reset and set, respectively, the I/O0B (Channel A) pin associated with the OPR register. (See Section 8.6.10 “Set the Output Port Bits OPR A and OPR B (SOPR A and SOPR B)” and Section 8.6.11 “Reset Output Port bits OPR A and OPR B (ROPR A, ROPR B)”). 0 1010: Set C/T Receiver time-out mode on. 0 1011: Set MR Pointer to 0. 0 1100: Set C/T Receiver time-out mode off. 0 1101: Block error status accumulation on FIFO entry. Allows the ‘received break’, ‘framing error’, and ‘parity error’ bits to be set as the received character is loaded to the RxFIFO (normally these bits are set on reading of the data from the RxFIFO). Setting this mode can give information about error data up to 256 bytes earlier than the normal mode. However, it clouds the ability to know precisely which byte(s) are in error. 0 1110: Power-down mode on. 0 1111: Disable Power-down mode.

Table 23: CRx - Command Register Extension (address 0x12) bit description *...continued*

Bit	Symbol	Description
4:0 (cont.)		1 0000: Transmit an Xon character. 1 0001: Transmit an Xoff character. 1 0010: C/T start sets the counter timer to the value of the Counter/Timer Preset Register and starts the counter. 1 0011: C/T stop. Effectively stops the counter/timer, captures the last count value and resets the counter ready status bit in the ISR. 1 0100: reserved 1 0101: reserved 1 0110: Transmitter resume command (this command is not active in 'Auto-Transmit mode'). A command to cancel a previous Host Xoff command. Upon receipt, the channels transmitter will transfer a character, if any, from the TxFIFO and begin transmission. 1 0111: Host Xoff (or transmitter pause) command (CRTXoff). This command allows tight host CPU control of the flow control of the channel transmitter. When interrupted for receipt of an Xoff character by the receiver, the host may stop transmission of further characters by the channel transmitter by issuing the Host Xoff command. Any character that has been transferred to the TXD shift register will complete its transmission, including the stop bit before the transmitter pauses. Even though the transmitter is paused it is still able to send Xon/Xoff by the request of its associated receiver. 1 1000: Cancel Host transmit flow control command. Issuing this command will cancel a previous command to transmit a flow control character if the flow control character is not yet loaded into the TXD Shift Register. If there is no character waiting for transmission, or if its transmission has already begun, then this command has no effect and the character will be sent. 1 1001: reserved 1 1010: reserved 1 1011: Reset Address Recognition Status. This command clears the interrupt status that was set when an address character was recognized by a disabled receiver operating in the special mode. 1 1100: reserved 1 1101: Block error status accumulates on FIFO read (default state). 1 1110: reserved 1 1111: Executes a chip wide reset. Executing this command in Channel A is equivalent to a hardware reset with the RESET(N) pin.

8.2.7 Channel Status Register (SR)

Table 24: SR - Channel Status Register (address 0x01) bit description

Bit	Symbol	Description
7		Received Break. 0 = no 1 = yes This bit indicates that an all zero character (including parity, if used) of the programmed length has been received with a stop bit at a logical zero. A single FIFO position is loaded with 0x00 when a break is received; further entries to the FIFO are inhibited until the RXD line returns to the marking state for at least one half bit time (two successive edges of the internal or external 1× clock). When this bit is set, the change in break bit in the ISR (ISR[2]) is set. ISR[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.
6	FE	Framing Error. 0 = no 1 = yes This bit indicates that a stop bit was not detected when an otherwise non-zeros data character (including parity, if enabled) was received. The stop bit check is made in the middle of the first stop bit position.
5	PE	Parity Error. 0 = no 1 = yes This bit is set when the ‘with parity’ or ‘force parity’ mode is programmed and the corresponding character in the FIFO was received with incorrect parity. In the special Wake-up mode, the parity error bit stores the received A/D bit.
4	OE	Overrun Error. 0 = no 1 = yes This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of the start bit of a new character when the RxFIFO is full and a character is already in the receive shift register (257 valid characters in the receiver) 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.
3	Tx Idle	Transmitter Idle. 0 = no 1 = yes This bit is set when the transmitter underruns, that is, both the TxFIFO and the Transmit Shift Register are empty. It is set after transmission of the last stop bit of a character, if no character is in the TxFIFO awaiting transmission. It is negated when the TxFIFO is loaded by the CPU, or when the transmitter is disabled or reset. This bit is concerned with the transmitter transmitting data and it essentially shows ‘transmitter underrun’. If, while it is underrun it is commanded to send an Xon/Xoff character it will remain at the zero state. If it is underrun and while sending an Xon/Xoff character the TxFIFO is loaded then the bit will go LOW.

Table 24: SR - Channel Status Register (address 0x01) bit description *…continued*

Bit	Symbol	Description
2	TxRDY	Transmitter Ready. 0 = no 1 = yes This bit, when set, indicates that the TxFIFO is ready to be loaded with at least one more character. This bit is cleared when the TxFIFO is full or is above its interrupt threshold level set in the MR registers or TxFIFO interrupt Fill Level register (TxFIL). Characters loaded in the TxFIFO while the transmitter is disabled will not be transmitted.
1	RxFULL	RxFIFO Full. 0 = no 1 = yes 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, that is, all 256 RxFIFO positions are occupied. It is reset when the CPU reads the RxFIFO and that read leaves one or more empty byte position(s). If a character is waiting in the receive shift register because the RxFIFO is full, RxFULL is not reset until the second read of the RxFIFO since the waiting character is immediately loaded to the RxFIFO.
0	RxRDY	Receiver Ready. 0 = no 1 = yes This bit indicates that a character has been received and is waiting in the RxFIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the RxFIFO and reset when the CPU reads the RxFIFO, and no more characters are in the RxFIFO.

8.2.8 Interrupt Status Register (ISR)

This register provides the status of all potential interrupt sources for a UART channel. When generating an interrupt arbitration value, 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', interrupt arbitration for this source will begin. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR can have no affect on the IRQN output. Note that the IMR may or may not mask the reading of the ISR as determined by GCCR[06]. If GCCR[0] is cleared, the reset and power on default, the ISR is read without modification. If GCCR[0] is set, the read of the ISR gives a value of the ISR ANDed with the IMR.

Table 25: ISR - Interrupt Status Register (address 0x25) bit description

Bit	Symbol	Description
7		I/O port change-of-state. Input change of state. This bit is set when a change of state occurs at the I/O1 or I/O0 input pins. It is reset when the CPU reads the Input Port Register, IPR.
6		Receiver watchdog time-out. Fixed watchdog time-out. This bit is set when the receiver's watchdog timer has counted more than 64 bit times since the last RxFIFO event. RxFIFO events are a read of the RxFIFO or GRxFIFO, or the load of a received character into the FIFO. The interrupt will be cleared automatically when the RxFIFO or GRxFIFO is read. The receiver watchdog timer is included to allow detection of the very last characters of a received message that may be waiting in the RxFIFO, but are too few in number to successfully initiate an interrupt. Refer to Section 7.4.7.8 "Receiver watchdog timer" for details of how the interrupt system works after a watchdog time-out.
5		Address recognition event. Address recognition status change. This bit is set when a change in receiver state has occurred due to an Address character being received from an external source and matches the reference address in ARCR. The bit and interrupt is negated by a write to the CR with command x1 1011, Reset Address Recognition Status.
4		Xon/Xoff event. Xon/Xoff status change. This bit is set when an Xon/Xoff character being received from an external source. The bit is negated by a read of the channel Xon/Xoff Interrupt Status Register, XISR.
3		C/T ready. Counter/Timer status. The C/T has timed-out or the count passed through 0. This bit is cleared by issuing the 'stop C/T' command.
2		Break change-of-state. Change in channel break status. 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 via the CR.

Table 25: ISR - Interrupt Status Register (address 0x25) bit description *...continued*

Bit	Symbol	Description
1	RxINT	Receiver entered the arbitration process. (Also Rx DMA hand-shake at I/O pins.) The general function of this bit is to indicate that the RxFIFO has data available and that it has entered the arbitration process. The particular meaning of this bit is programmed by RxFIL register. If programmed as receiver ready (MR2[3:2] = 00), it indicates that at least one character has been received and is waiting in the RxFIFO to be read by the host CPU. It is set when the character is transferred from the receive shift register to the RxFIFO and reset when the CPU reads the last character from the RxFIFO. If RxFIL is programmed as FIFO full, ISR[1] is set when a character is transferred from the receive holding register to the RxFIFO and the transfer causes the RxFIFO to become full, that is, all 256 FIFO positions are occupied. It is reset whenever RxFIFO is not full. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO. The other two conditions of these bits, ¾ and ½ full operate in a similar manner. The ISR[1] bit is set when the RxFIFO fill level meets or exceeds the value; it is reset when the fill level is less. See the description of the MR2 register. **Remark:** This bit must be at a one (1) for the receiver to enter the arbitration process. It is the fact that this bit is zero (0) when the RxFIFO is empty that stops an empty FIFO from entering the interrupt arbitration. Also note that the meaning if this bit is not quite the same as the similar bit in the Status Register (SR).
0	TxINT	Transmitter entered the arbitration process. (Also Tx DMA hand-shake at I/O pins.) The general function of this bit is to indicate that the TxFIFO has an at least one empty space for data. The particular meaning of the bit is controlled by MR0 [5:4] indicates the TxFIFO may be loaded with one or more characters. If MR0[5:4] = 00 (the default condition) this bit will not set until the TxFIFO is empty: 256 bytes available. If the fill level of the TxFIFO is below the trigger level programmed by the TxINT field of the Mode Register 0, this bit will be set. A one in this position indicates that at least one character can be sent to the TxFIFO. It is turned off as the TxFIFO is filled above the level programmed by MR0[5:4]. This bit turns on as the FIFO empties. (Note that the RxFIFO bit turns on as the FIFO fills.) This often a point of confusion in programming interrupt functions for the receiver and transmitter FIFOs. **Remark:** This bit must be at a one (1) for the transmitter to enter the arbitration process. It is the fact that this bit is zero (0) when the TxFIFO is full that stops a full TxFIFO from entering the interrupt arbitration. Also note that the meaning if this bit is not quite the same as the similar bit in the status register (SR).

8.2.9 Interrupt Mask Register (IMR)

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a ‘1’ and the corresponding bit in the IMR is a ‘1’, the interrupt source is presented to the internal interrupt arbitration circuits, eventually resulting in the IRQN output being asserted (LOW). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no affect on the IRQN output.

Table 26: IMR - Interrupt Mask Register (address 0x25) bit description

Bit	Symbol	Description
7		I/O port change-of-state. COS enable. Allows a change of state in the inputs equipped with input change detectors to cause an interrupt.
6		Rx watchdog time-out. Fixed watchdog enable. Controls the generation of an interrupt watchdog timer event, If set, a count of 64 idle bit times in the receiver will begin interrupt arbitration.
5		Address recognition event. Address recognition enable. Enables the generation of an interrupt in response to changes in the Address Recognition circuitry of the Special Mode (multi-drop or Wake-up mode).
4		Xon/Xoff event. Xon/Xoff enable. Enables the generation of an interrupt in response to recognition of an in-band flow control character.
3		C/T Ready. Counter/Timer Enable. Enable the C/T interrupt when the C/T reaches 0 count.
2		Break Change Of State. Enables the generation of an interrupt when a Break condition has been detected by the channel receiver.
1		RxRDY interrupt. Receiver (Rx) Enable. Enables the generation of an interrupt when servicing for the RxFIFO is desired.
0		TxRDY interrupt. Transmitter (Tx) Enable. Enables the generation of an interrupt when servicing for the TxFIFO is desired.

8.2.10 Receiver FIFO (RxFIFO)

The FIFO for the receiver is 11 bits wide and 256 words deep. The status of each byte received is stored with that byte and is moved along with the byte as the characters are read from the FIFO. The upper three bits are presented in the Status Register and they change in the Status Register each time a data byte is read from the FIFO. Therefore, the Status Register should be read **before** the byte is read from the RxFIFO if one wishes to ascertain the quality of the byte.

The foregoing applies to the ‘character error mode’ of status reporting. See MR1[5] in Table 17 and RxFIFO descriptions for ‘block error’ status reporting. Briefly, Block Error gives the accumulated error of all bytes received by the RxFIFO since the last Reset Error command was issued (CR = 0x04).

Table 27: RxFIFO - Receiver FIFO register (address 0x03) bit description

Bit	Symbol	Description
10 [1]		break received status
9 [1]		framing error status
8 [1]		parity error status
7:0		This is the data byte sent to the data bus on RxFIFO read. 8 data bits. MSBs = 0 for 7, 6, 5 bit data.

[1] These bits are sent to the Status Register.

8.2.11 Transmitter FIFO (TxFIFO)

The FIFO for the transmitter is 8 bits wide by 256 bytes deep. For character lengths less than 8 bits, the upper bits will be ignored by the transmitter state machine and thus are effectively discarded.

Table 28: TxFIFO - Transmitter FIFO register (address 0x03) bit description

Bit	Symbol	Description
7:0		8 data bits. MSBs are ignored to 0 for 7, 6, 5 bit data.

8.2.12 Receiver FIFO Interrupt Level (RxFIL)

The position in the RxFIFO that causes the receiver will enter the interrupt arbitration process. This register is used to offset the effect of the arbitration threshold. It use may yield moderate improvements in the interrupt service. It will also ‘equalize’ interrupt latency and allow for larger aggregate block transfers between fast and slow channels. Writing to this register removes the interrupt control established in MR0 and MR1.

Table 29: RxFIL - Receiver FIFO Interrupt Level register (address 0x46) bit description

Bit	Symbol	Description
7:0		any one of 256 FIFO fill positions

8.2.13 Receiver FIFO Fill Level register (RxFL)

The number of bytes filled in the Receiver FIFO.

Table 30: RxFL - Receiver FIFO Fill Level register (address 0x10) bit description

Bit	Symbol	Description
7:0		Channel byte count code **(1) = implied '1'. 0000 0001 = 1 0000 0010 = 2 to 1111 1111 = 255 **(1) 0000 0000 = 256 if RxRDY status bit is set.

8.2.14 Transmitter FIFO Interrupt Level (TxFIL)

The position in the Tx FIFO caused the transmitter to enter the interrupt arbitration process. This register is used to offset the effect of the arbitration threshold. Its use may yield moderate improvements in the interrupt service. It will also 'equalize' interrupt latency and allow for larger aggregate block transfers between fast and slow channels. Writing to this register removes the interrupt control established in MR0 and MR1.

Table 31: TxFIL - Transmitter FIFO Interrupt Level register (address 0x47) bit description

Bit	Symbol	Description
7:0		Any one of 256 FIFO empty positions.

8.2.15 Transmitter FIFO Empty Level register (TxEL)

The number of empty bytes in the Transmitter FIFO.

Table 32: TxEL - Transmitter FIFO Empty Level register (address 0x11) bit description

Bit	Symbol	Description
7:0		Channel byte count code **(1) = implied '1'. 0000 0001 = 1 0000 0010 = 2 to 1111 1111 = 255 **(1) 0000 0000 = 256 if TxRDY status bit is set.

8.3 Registers for character recognition

Please note that, although the names of the registers imply a particular function, there is not any hardware function directly attached to them. They are just three characters that may be used for any function requiring recognition or simple character stripping.

It is only when other internal logic is enabled that the reception of a recognized character will trigger particular chip functions and/or interrupts.

8.3.1 Xon Character Register (XonCR)

An 8-bit character register that contains the compare value for an Xon character.

Table 33: XonCR - Xon Character Register (address 0x41) bit description

Bit	Symbol	Description
7:0		8 bits of the Xon Character Recognition (resets to 0x11).

8.3.2 Xoff Character Register (XoffCR)

An 8-bit character register that contains the compare value for an Xoff character.

Table 34: XoffCR - Xoff Character Register (address 0x42) bit description

Bit	Symbol	Description
7:0		8 bits of the Xoff Character Recognition (resets to 0x13).

8.3.3 Address Recognition Character Register (ARCR)

An 8-bit character register that contains the compare value for the wake-up address character.

Table 35: ARCR - Address Recognition Character Register (address 0x43) bit description

Bit	Symbol	Description
7:0		8 bits of the Multi-drop Address Character Recognition (resets to 0x00).

8.3.4 Xon/Xoff Interrupt Status Register (XISR)

Reading this register clears XISR[7:4].

Table 36: XISR - Xon/Xoff Interrupt Status Register (address 0x44) bit description

Bit	Symbol	Description
7:6		Received X Character Status. 00 = none 01 = Xoff received 10 = Xon received 11 = both received This field can be read to determine if the receiver has encountered an Xon or Xoff character in the incoming data stream. These bits are maintained until a read of the XISR. The field is updated by X character reception regardless of the state of MR3[7] and MR3[3:2] or IMR[4]. The field can therefore be used as a character detector for the bit patterns stored in the Xon and Xoff Character Registers.
5:4		Automatic X Character transmission status. 00 = none 01 = Xon transmitted 10 = Xoff transmitted 11 = both transmitted This field indicates the last flow control character sent in the Auto Receiver flow control mode. If Auto Receiver mode has not been enabled, this field will always read 00. It will likewise reset to ‘00’ if MR0[3] is reset. If the Auto Receiver mode is exited while this field reads ‘10’, it is the user’s responsibility to transmit an Xon, when appropriate.
3:2		TXD flow status. TXD condition of the automatic flow control status. This field tracks the transmitter’s flow status as follows: 00 = normal transmission. Transmitter is not affected by Xon or Xoff. 01 = TXD halt pending. After the current character finishes the transmitter will stop. The status will then change to ‘11’. 10 = re-enabled. The transmitter had been halted and has been re-started. It is sending (or is prepared to send) data characters. After a read of the XISR, it will return to ‘normal’ status. 11 = flow halted. The Transmitter is stopped due to an Xoff character being received from its associated receiver. The transmitter is ‘flow controlled’.
1:0		TXD X Character Status. 00 = normal TXD data. 01 = Idle wait for FIFO data. The character is waiting for a data character to transfer from the TxFIFO. This condition will only occur for a bit time after an Xon or Xoff character transmission unless the TxFIFO is empty 10 = Xoff is pending. A command to send an Xoff character is pending. 11 = Xon is pending. A command to send an Xon character is pending. Conditions ‘10’ and ‘11’ will not exist for more than a character time.

8.3.5 Watchdog, Character, Address and X Enable Register (WCXER)

This register enables the UARTs Character Recognition, Address Recognition and Receiver watchdog timer. If both enable and disable are active, a disable results. This register is used to enable the general-purpose character recognition feature **without** causing any Xon/Xoff or Wake-up mode activities to occur. The recognition event is reported in the ISR register.

Table 37: WCXER - Watchdog, Character, Address and X Enable Register (address 0x40) bit description

Bit	Symbol	Description
7 [1]		Watchdog enable. 1 = disable watchdog 0 = no action
6		Watchdog disable. 1 = enable watchdog 0 = no action
5		Address recognition disable. 1 = disable Address recognition 0 = no action
4		Address recognition enable. 1 = enable Address recognition 0 = no action
3		Xon recognition disable. 1 = disable Xon 0 = no action
2		Xon recognition enable. 1 = enable Xon 0 = no action
1		Xoff recognition disable. 1 = disable Xoff 0 = no action
0		Xoff recognition enable. 1 = enable Xoff 0 = no action

[1] This bit control is duplicated at MR0[7].

8.4 Programmable counters, timers, and baud rate generators

8.4.1 Programmable BRG Timer Reload Registers, Upper 0 and Upper 1 (PBRGPU)

This is the upper byte of the 16-bit value used by the BRG timer in generating a baud rate clock.

Table 38: PBRGPU - Programmable BRG Timer Reload Registers, Upper 0 and Upper 1 (address 0x27) bit description

Bit	Symbol	Description
7:0		8 MSBs of the BRG Timer divisor.

8.4.2 Programmable BRG Timer Reload Registers, Lower 0 and Lower 1 (PBRGPL)

This is the lower byte of the 16-bit value used by the BRG timer in generating a baud rate clock.

Table 39: PBRGPL - Programmable BRG Timer Reload Registers, Lower 0 and Lower 1 (address 0x26) bit description

Bit	Symbol	Description
7:0		8 LSBs of the BRG Timer divisor.

8.4.3 Counter/Timer clock source (CTCS0 and CTCS1)

Remark: Writing to this register removes the control established in the counter/timer portion of the ACR in the default register map.

Table 40: CTCS0 and CTCS1 - Counter/Timer clock source registers (address 0x24, 0x2C) bit description

Bit	Symbol	Description
7:6		reserved
5:4		Mode control 00 = selects Counter mode. Generates a timing edge. 01 = selects Timer mode. Generates a square wave. 10 = reserved 11 = selects Timer Pulse mode. Generates periodic pulses twice the frequency as in Timer mode. Pulse width is one cycle of the clock as it is delivered to the C/T, that is, after any prescale.
3:0		Clock selection 0000 = external I/O2A (for CT 0); I/O7A (for CT 1) 0001 = external I/O2A/16 (for CT 0); I/O7A/16 (for CT 1) 0010 = SCLK 0011 = SCLK/2 0100 = SCLK/16 0101 = SCLK/32 0110 = SCLK/64 0111 = SCLK/128 1000 = TxCA1X 1001 = reserved 1010 = reserved 1011 = reserved 1100 = Rx Character Count (Ch A) Clock is RxFIFO A load pulse 1101 = reserved 1110 = reserved 1111 = reserved

8.4.4 Counter Timer Value Registers, Upper 0 and Upper 1 (CTVU0, CTVU1)

Reading this register gives the value of the upper 8 bits of the Counter/Timer.

Table 41: CTVU0, CTVU1 - Counter Timer Value registers, Upper 0 and Upper 1 (address 0x16, 0x1E) bit description

Bit	Symbol	Description
7:0		8 MSBs of the Counter/Timer preset value

8.4.5 Counter Timer Value Registers, Lower 0 and Lower 1 (CTVL0, CTVL1)

Reading this register gives the value of the lower 8 bits of the Counter/Timer.

Table 42: CTVL0, CTVL1 - Counter Timer Value registers, Lower 0 and Lower 1 (address 0x17, 0x1F) bit description

Bit	Symbol	Description
7:0		8 LSBs of the Counter/Timer preset value

Remark: The Counter/Timer should be stopped before reading. Usually the clock of the Counter/Timer is not synchronized with the read of the C/T. It is therefore possible to capture changing data during the read. Depending on the clock speed with respect to the read cycle, this could be made worse or completely eliminated. If the 'Stop counter' command is issued and following that the C/T is read, there will be no uncertainty to its value. If it is necessary to read the C/T 'on the fly', then reading it twice and comparing the values will correct the problem. The double read will not be effective if the counter timer clock is faster than a read cycle.

8.4.6 Programmable BRG Clock Source, 0 and 1 (PBRGCS)

Start/Stop control and Clock Select register for the two BRG counters. The clock selection is for the input to the counters. It is that clock divided by the number represented by the PBRGPU and PBRGPL then will be used as the 16× clock for the receivers and transmitters. When the BRG timer Clock is selected for the receiver(s) or transmitter(s) the receivers and transmitters will consider it as a 16× clock and further divide it by 16. In other words, the receivers and transmitters will always be in the 16× mode of operation when the internal BRG timer is selected for their clock.

Table 43: PBRGCS - Programmable BRG Clock Source 0 and 1 register (address 0x33) bit description

Bit	Symbol	Description
7		PBRG 1 register control 0 = resets PBRG 1 and holds it stopped 1 = allows PBRG 1 to run
6:4		PBRG 1 clock selection 000 = SCLK 001 = SCLK/2 010 = SCLK/6 011 = SCLK/32 100 = SCLK/64 101 = SCLK/128 110 = I/O4A 111 = reserved

Table 43: PBRGCS - Programmable BRG Clock Source 0 and 1 register (address 0x33) bit description *...continued*

Bit	Symbol	Description
3		PBRG 0 register control 0 = resets PBRG 0 and holds it stopped 1 = allows PBRG 0 to run
2:0		PBRG 0 clock selection 000 = SCLK 001 = SCLK/2 010 = SCLK/16 011 = SCLK/32 100 = SCLK/64 101 = SCLK/128 110 = I/O3A 111 = reserved

8.4.7 Counter/Timer Preset Upper and Counter/Timer Preset Lower (CTPU, CTPL)

Table 44: CTPU - Counter/Timer Preset Upper 0 and 1 register (address 0x16, 0x1E) bit description

Bit	Symbol	Description
7:0		The lower eight (8) bits for the 16-bit Counter/Timer Preset register

Table 45: CTPL - Counter/Timer Preset Lower 0 and 1 register (address 0x17, 0x1F) bit description

Bit	Symbol	Description
7:0		The upper eight (8) bits for the 16-bit Counter/Timer Preset register

The CTPU and CTPL 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 that may be loaded into the CTPU/CTPL registers is 0x0000. Note that these registers are write-only and cannot be read by the CPU.

In the timer mode, the C/T generates a square wave whose period is twice the value (in C/T clock periods) of the CTPU and CTPL. The waveform so generated is often used for a data clock. The formula for calculating the divisor 'n' to load to the CTPU and CTPL for a particular 1× data clock is shown in Equation 2.

$$n = \frac{\text{C/T clock input frequency}}{(2 \times 16 \times (\text{baud rate desired}))} \quad (2)$$

Remark: The 2 in the denominator is for the square wave generation. For the Pulse mode, change the 2 to a 1. This doubles the C/T output speeds for any given input clock.

Often this division will result in a non-integer value: 26.3 for example. One may only program integer numbers to a digital divider. Therefore, 26 (0x001A) would be chosen. If 26.7 were the result of the division, then 27 (0x0017) would be chosen. This gives a baud rate error of 0.3/26.3 or 0.3/26.7 that yields a percentage error of 1.14 % or 1.12 %, respectively, well within the ability of the asynchronous mode of operation. Higher input frequency to the counter reduces the error effect of the fractional division.

If the value in CTPU and CTPL 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 from the Command Register (or a read at address A6 to A0 = 0x0E in the lower 16 position address space). After this, while in timer mode, the C/T will run continuously. Receipt of a start counter command causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTPU and CTPL.

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 from the command register (or a read with A6 to A0 = 0x0F in the lower 16 position address space). The command however, does not stop the C/T. the generated square wave is output on I/O3 if it is programmed to be the C/T output. In the counter mode, the value C/T loaded into CTPU and CTPL by the CPU is counted down to 0. Counting begins upon receipt of a start counter command. Upon reaching terminal count 0x0000, the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If I/O3 is programmed to be the output of the C/T, the output remains HIGH until terminal count is reached; at which time it goes LOW. The output returns to the HIGH state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTPU and CTPL at any time, but the new count becomes effective only on the next start counter commands. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTPU, CTPL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems that may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTPU and CTPL. When the C/T clock divided by 16 is selected, the maximum divisor becomes 1,048,575.

8.4.8 The CTS, RTS, CTS Enable Tx signals

CTS (Clear To Send) is usually meant to be a signal to the transmitter meaning that it may transmit data to the receiver. The CTS input is on pin I/O0A for Tx A. The CTS signal is active LOW, thus it is called CTSN A for Tx A. RTS is usually meant to be a signal from the receiver indicating that the receiver is ready to receive data. It is also active LOW and is called RTSN A for Rx A. RTSN A is on pin I/O0B. A receiver's RTSN output will usually be connected to the CTS input of the associated transmitter. Therefore, one could say that RTS and CTS are different ends of the same wire!

MR2[4] is the bit that allows the transmitter to be controlled by the CTS pin (I/O0A or I/O1A). When this bit is set to one **and** the CTS input is driven HIGH, the transmitter will stop sending data at the end of the present character being serialized. It is usually the RTS output of the receiver that will be connected to the transmitter's CTS input. The receiver will set RTS HIGH when the receiver FIFO is full **and** the start bit of the ninth character is sensed. Transmission then stops with nine valid characters in the receiver. When MR2[4] is set to one, CTSN must be at zero for the transmitter to operate. If MR2[4] is set to zero, the I/O pin will have no effect on the operation of the transmitter. MR1[7] is the bit that allows the receiver to control I/O0B. When the receiver controls I/O0B, the meaning of that pin will be the RTSN function.

8.5 Registers of the Arbitrating Interrupt system and bidding control

8.5.1 Interrupt Control Register (ICR)

Table 46: ICR - Interrupt Control Register (address 0x60) bit description

Bit	Symbol	Description
7:0		Upper 8 bits of the Arbitration Threshold

This register provides a single 8-bit field called the interrupt threshold for use by the interrupt arbiter. The field is interpreted as a single unsigned integer. The interrupt arbiter will not generate an external interrupt request, by asserting IRQN, unless the value of the highest priority interrupt exceeds the value of the interrupt threshold. If the highest bidder in the interrupt arbitration is lower than the threshold level set by the ICR, the Current Interrupt Register, CIR, will contain 0x00.

Remark: While a Watchdog Timer interrupt is pending, the ICR is not used and only receiver codes are presented for interrupt arbitration. This allows receivers with very low count values (perhaps below the threshold value) to win interrupt arbitration without requiring the user to explicitly lower the threshold level in the ICR.

8.5.2 Update CIR (UCIR): the most important command for polled or interrupt service

Table 47: UCIR - Update CIR (address 0x61) bit description

Bit	Symbol	Description
7:0		data not important

A write to address 0x61 data is not important. A command based upon a decode of a write to address 0x61 (UCIR is not a register). A write (the write data is not important; a ‘Don’t care’) to this ‘register’ causes the Current Interrupt Register to be updated with the value that is winning interrupt arbitration. The register would be used in systems that poll the interrupt status registers rather than wait for interrupts. Alternatively, the CIR is normally updated during an Interrupt Acknowledge Bus (IACKN) cycle in interrupt-driven systems.

8.5.3 Current Interrupt Register (CIR)

Table 48: CIR - Current Interrupt Register (address 0x61) bit description

Bit	Symbol	Description
7:6		Type 00 = type other than transmit or Receiver 01 = transmit 10 = receive without errors 11 = receive with errors
5:1		Current byte count/type. When CIR[7:6] = 00: 0 0000 = no interrupt 0 0001 = change-of-state 0 0010 = address recognition 0 0011 = Xon/Xoff status 0 0100 = Receiver watchdog 0 0101 = break change 0 0110 = Counter/Timer 0 0111 = Rx Loopback error Current count code. When CIR[7:6] = 01, 10, or 11: 0 0000 => at least 1 character 0 0001 => at least 16 characters 0 0010 => at least 24 characters … 1 1101 => at least 240 characters 1 1110 => at least 248 characters 1 1111 => 256 (see also Section 8.5.6 “Global Interrupting Byte Count Register (GIBCR)”)
0		Channel number or C/T number 0 = Channel A or C/T 0 1 = C/T 1

The Current Interrupt Register is provided to speed up the specification of the interrupting condition in the UART. The CIR is updated at the beginning of an interrupt acknowledge bus cycle or in response to an Update CIR command (see Section 8.5.2). Although interrupt arbitration continues in the background, the current interrupt information remains frozen in the CIR until another IACKN cycle or Update CIR command occurs. The LSBs of the CIR provide part of the addressing for various Global Interrupt registers including the GIBCR, GICR, GITR and the Global RxFIFO and TxFIFO FIFO. The host CPU need not generate individual addresses for this information since the interrupt context will remain stable at the fixed addresses of the Global Interrupt registers until the CIR is updated. For most interrupting sources, the data available in the CIR alone will be sufficient to set up a service routine.

The CIR may be processed as follows:

If CIR[7] = 1, then a receiver interrupt is pending and the count is CIR[5:1], channel is CIR[0]

Else if CIR[6] = 1 then a transmitter interrupt is pending and the count is CIR[5:1], channel is CIR[0]

Else the interrupt is another type, specified in CIR[5:1].

Remark: The GIBCR, Global Interrupting Byte Count Register, may be read to determine an exact character count.

8.5.4 Interrupt Vector Register (IVR)

The IVR contains the byte that will be placed on the data bus during an IACKN cycle when the GCCR bits [2:1] are set to binary '01'. This is the unmodified form of the interrupt vector.

Table 49: IVR - Interrupt Vector Register (address 0x64) bit description

Bit	Symbol	Description
7:0		8 data bits of the Interrupt Vector (IVR)

8.5.5 Global Interrupting Channel Register (GICR)

A register associated with the interrupting channel as defined in the CIR. It contains the channel number for the interrupting channel.

Table 50: GICR - Global Interrupting Channel Register (address 0x70) bit description

Bit	Symbol	Description
7:1		reserved
0		Channel code 0 = a 1 = reserved

8.5.6 Global Interrupting Byte Count Register (GIBCR)

A register associated with the interrupting channel as defined in the CIR. Its numerical value equals TxEL or RxFL at the time IACKN or 'Update CIR' command was issued. The true number of bytes ready for transfer to the transmitter or transfer from the receiver. It is undefined for other types of interrupts.

Table 51: GIBCR - Global Interrupting Byte Count Register (address 0x71) bit description

Bit	Symbol	Description
7:0		Channel byte count code 0000 0001 = 1 0000 0010 = 2 … 1111 1111 = 255 0000 0000 = 256

8.5.7 Global Interrupting Type Register (GITR)

A register associated with the interrupting channel as defined in the CIR. It contains the type of interrupt code for all interrupts.

Table 52: GITR - Global Interrupting Type Register (address 0x72) bit description

Bit	Symbol	Description
7:6		Receiver interrupt 0x = not receiver 10 = with receive errors 11 = without receive errors
5		Transmitter interrupt 0 = not transmitter 1 = transmitter interrupt
4:3		Reserved. Read 0x00.
2:0		Other types 000 = not ‘other’ type 001 = change of state 010 = address recognition event 011 = Xon/Xoff status 100 = Rx watchdog 101 = Break change 110 = Counter/Timer 111 = Rx Loopback Error

8.5.8 Global RxFIFO register (GRxFIFO)

The RxFIFO of the channel indicated in the CIR channel field. Undefined when the CIR interrupt context is not a receiver interrupt.

Table 53: GRxFIFO - Global RxFIFO register (address 0x73) bit description

Bit	Symbol	Description
7:0		8 data bits of RxFIFO. MSBs set to ‘0’ for 7, 6, 5 bit data.

8.5.9 Global TxFIFO register (GTxFIFO)

The TxFIFO of the channel indicated in the CIR channel field. Undefined when the CIR interrupt context is not a transmitter interrupt. Writing to the GTxFIFO when the current interrupt is not a transmitter event may result in the characters being transmitted on a different channel than intended.

Table 54: GTxFIFO - Global TxFIFO register (address 0x73) bit description

Bit	Symbol	Description
7:0		8 data bits of TxFIFO. MSBs not used for 7, 6, 5 bit data.

8.5.10 Bidding Control Register, Break Change (BCRBRK)

This register provides the 8 MSBs of the Interrupt Arbitration number for a break change interrupt.

Table 55: BCRBRK - Bidding Control Register, Break change (address 0x50) bit description

Bit	Symbol	Description
7:0		MSBs of break change interrupt bid.

8.5.11 Bidding Control Register, Change-Of-State (BCRCOS)

This register provides the 8 MSBs of the Interrupt Arbitration number for a Change-Of-State, COS, interrupt.

Table 56: BCRCOS - Bidding Control Register, Change-Of-State (address 0x51) bit description

Bit	Symbol	Description
7:0		MSBs of Change-Of-State detectors (COS) interrupt bid.

8.5.12 Bidding Control Register, Xon/Xoff (BCRx)

This register provides the 8 MSBs of the Interrupt Arbitration number for an Xon/Xoff interrupt.

Table 57: BCRx - Bidding Control Register, Xon/Xoff (address 0x53) bit description

Bit	Symbol	Description
7:0		MSBs of an Xon/Xoff interrupt bid.

8.5.13 Bidding Control Register, Address (BCRA)

This register provides the 8 MSBs of the Interrupt Arbitration number for an address recognition event interrupt.

Table 58: BCRA - Bidding Control Register, Address (address 0x54) bit description

Bit	Symbol	Description
7:0		MSBs of an address recognition event interrupt bid.

8.5.14 Bidding Control Register, C/T 0 and C/T 1 (BCR C/T)

This register provides the 8 MSBs of the Interrupt Arbitration number for a counter/timer event interrupt.

Table 59: BCRA - Bidding Control Register, Address (address 0x52) bit description

Bit	Symbol	Description
7:0		MSBs of a Counter/Timer event interrupt bid.

8.5.15 Bidding Control Register, Received Loopback Error (BCRLBE)

This register provides the 8 MSBs of the Interrupt Arbitration number for the received loopback error interrupt.

Table 60: BCRLBE - Bidding Control Register, Address (address 0x55) bit description

Bit	Symbol	Description
7:0		MSBs of a received loopback error event interrupt bid.

8.6 Registers of the I/O ports

8.6.1 Input Port Change Register Lower Nibble, A (IPCRL)

This register may be read to determine the current logical level of the I/O pins and examine the output of the change detectors assigned to each pin. If the change detection is not enabled or if the pin is configured as an output, the associated change field will read '0'.

Table 61: IPCRL - Input Port Change Register Lower Nibble, A (address 0x14) bit description

Bit	Symbol	Description
7		ΔI/O3A change. 0 = no change 1 = change
6		ΔI/O2A change. 0 = no change 1 = change
5		ΔI/O1A change. 0 = no change 1 = change
4		ΔI/O0A change. 0 = no change 1 = change
3		I/O3A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
2		I/O2A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
1		I/O1A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
0		I/O0A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level

8.6.2 Input Port Change Register Lower Nibble, B (IPCRL)

This register may be read to determine the current logical level of the I/O pins and examine the output of the change detectors assigned to each pin. If the change detection is not enabled or if the pin is configured as an output, the associated change field will read '0'.

Table 62: IPCRL - Input Port Change Register Lower Nibble, B (address 0x1C) bit description

Bit	Symbol	Description
7		ΔI/O3B change. 0 = no change 1 = change
6		ΔI/O2B change. 0 = no change 1 = change
5		reserved
4		ΔI/O0B change. 0 = no change 1 = change
3		I/O3B state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
2		I/O2B state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
1		reserved
0		I/O0B state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level

8.6.3 Input Port Change Register Upper Nibble, A (IPCRU)

This register may be read to determine the current logical level of the I/O pins and examine the output of the change detectors assigned to each pin. If the change detection is not enabled or if the pin is configured as an output, the associated change field will read '0'.

Table 63: IPCRU - Input Port Change Register Upper Nibble, A (address 0x13) bit description

Bit	Symbol	Description
7		ΔI/O7A change. 0 = no change 1 = change
6		ΔI/O6A change. 0 = no change 1 = change
5		ΔI/O5A change. 0 = no change 1 = change
4		ΔI/O4A change. 0 = no change 1 = change
3		I/O7A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
2		I/O6A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
1		I/O5A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
0		I/O4A state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level

8.6.4 Input Port Change Register Upper Nibble, B (IPCRU)

This register may be read to determine the current logical level of the I/O pins and examine the output of the change detectors assigned to each pin. If the change detection is not enabled or if the pin is configured as an output, the associated change field will read '0'.

Table 64: IPCRU - Input Port Change Register Upper Nibble, B (address 0x1B) bit description

Bit	Symbol	Description
7		reserved
6		ΔI/O6B change. 0 = no change 1 = change
5		reserved
4		ΔI/O4B change. 0 = no change 1 = change
3		reserved
2		I/O6B state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level
1		reserved
0		I/O4B state. Reads the actual logic level at the pin. 1 = HIGH level 0 = LOW level

8.6.5 Input Port Register, A (IPR)

Table 65: IPR - Input Port Register, A (address 0x15) bit description

Bit	Symbol	Description
7:0		logical levels of I/O[7:0]A

8.6.6 Input Port Register, B (UPT)

Table 66: UPT - Input Port Register, B (address 0x1D) bit description

Bit	Symbol	Description
7:0		logical levels of I/O[6]B, I/O[4:2]B, I/O[0]B

8.6.7 Input Change Detect Enable, A (IPCE)

IPCE[7:0] bits activate the input change of state detectors. If a pin is configured as an output, the change of state detectors, if enabled, continue to be active and will show a change of state as the I/P port changes.

Table 67: IPCE - Input Change Detect Enable register, A (address 0x32) bit description

Bit	Symbol	Description
7		ΔI/O7A enable. 0 = disable 1 = enable
6		ΔI/O6A enable. 0 = disable 1 = enable
5		ΔI/O5A enable. 0 = disable 1 = enable
4		ΔI/O4A enable. 0 = disable 1 = enable
3		ΔI/O3A enable. 0 = disable 1 = enable
2		ΔI/O2A enable. 0 = disable 1 = enable
1		ΔI/O1A enable. 0 = disable 1 = enable
0		ΔI/O0A enable. 0 = disable 1 = enable

8.6.8 Input Change Detect Enable, B (IPCE)

IPCE[7:0] bits activate the input change of state detectors. If a pin is configured as an output, the change of state detectors, if enabled, continue to be active and will show a change of state as the I/P port changes.

Table 68: IPCE - Input Change Detect Enable register, A (address 0x32) bit description

Bit	Symbol	Description
7		reserved
6		ΔI/O6B enable. 0 = disable 1 = enable
5		reserved
4		ΔI/O4B enable. 0 = disable 1 = enable
3		ΔI/O3B enable. 0 = disable 1 = enable
2		ΔI/O2B enable. 0 = disable 1 = enable
1		reserved
0		ΔI/O0B enable. 0 = disable 1 = enable

8.6.9 I/O Port Configuration Registers (I/OPCR0, I/OPCR1, I/OPCR2, I/OPCR3)

These four registers contain 4 2-bit fields that set the direction and source for each of the I/O pins associated with the channel. The I/O0B output may be RTSN if MR1[7] is set. It may also signal end of transmission if MR2[5] is set. (Please see the descriptions of these functions under Section 8.2.2 “Mode Register 1 (MR1)” and Section 8.2.3 “Mode Register 2 (MR2)”).

The binary settings of the binary 00 combination always configures the I/O pins as ‘inputs’. However, the input circuit of the I/O pins are **always** active. In actuality, the binary 00 condition only disables the output driver of the pin. Since the input circuit and the associated change-of-state detector is always active, the output signal may generate interrupts or drive counters.

This register resets to 0x00 on reset, effectively configuring all I/O pins as inputs. Inputs may be used as RxC, TxC inputs or CTSN and General Purpose Inputs simultaneously. All inputs are equipped with change detectors that may be used to generate interrupts or can be polled, as required.

Remark: Both I/O Port A and B default to input upon a hardware reset to avoid hardware conflicts with I/O direction.

Table 69: I/OPCR0 - I/O Port Configuration Register (address 0x13) bit description

Bit	Symbol	Description
7:6		I/O3A control 00 = GPI/TxC A 01 = OPR[3] A 10 = TxC A (16×) output 11 = reserved
5:4		I/O2A control 00 = GPI/CT 0 clock input 01 = OPR[2] A/DTRN A 10 = TxC A (1×) output 11 = reserved
3:2		I/O1A control 00 = GPI 01 = OPR[1] A 10 = reserved 11 = reserved
1:0		I/O0A control 00 = GPI/CTSN A 01 = OPR[0] A 10 = reserved 11 = reserved

Table 70: I/OPCR1 - I/O Port Configuration Register (address 0x14) bit description

Bit	Symbol	Description
7:6		I/O7A control 00 = GPI/CT 1 Clock input 01 = OPR[7] A 10 = TxC B (1×) output 11 = reserved
5:4		I/O6A control 00 = GPI/PBRG 1 clock input 01 = OPR[6] A 10 = reserved 11 = reserved
3:2		I/O5A control 00 = GPI 01 = OPR[5] A 10 = reserved 11 = reserved
1:0		I/O4A control 00 = GPI/RxC A/PBRG 0 clock input 01 = OPR[4] A 10 = RxC A (16×) output 11 = reserved

Table 71: I/OPCR2 - I/O Port Configuration Register (address 0x1B) bit description

Bit	Symbol	Description
7:6		I/O3B control 00 = GPI 01 = OPR[3] B 10 = reserved 11 = C/T 0 output (open-drain)
5:4		I/O2B control 00 = GPI/DSRN A 01 = OPR[2] B 10 = RxC A (1×) output 11 = CT 1 output (open-drain)
3:2		reserved
1:0		I/O0B control 00 = GPI 01 = OPR[0] B 10, 11 = reserved

Table 72: I/OPCR3 - I/O Port Configuration Register (address 0x1C) bit description

Bit	Symbol	Description
7:6		returns ‘0’ on a read cycle
5:4		I/O6B control 00 = GPI/RIN A 01 = OPR[6] B 10 = TxINTN A (open-drain) 11 = reserved
3:2		returns ‘0’ on a read cycle
1:0		I/O4B control 00 = GPI/DCDN A 01 = OPR[4] B 10 = RxINTN A (open-drain) 11 = reserved

8.6.10 Set the Output Port Bits OPR A and OPR B (SOPR A and SOPR B)

Ones in the byte written to this register will cause the corresponding bit positions in the OPR to set to '1'. Zeros have no effect. This allows software to set individual bits with our keeping a copy of the OPR bit configuration. One register for each channel.

Table 73: SOPR A, SOPR B - Set the Output Port Bits OPR A and OPR B register (address 0x10, 0x18) bit description

Bit	Symbol	Description
7		OPR 7 1 = set bit 0 = no change
6		OPR 6 1 = set bit 0 = no change
5		OPR 5 1 = set bit 0 = no change
4		OPR 4 1 = set bit 0 = no change
3		OPR 3 1 = set bit 0 = no change
2		OPR 2 1 = set bit 0 = no change
1		OPR 1 1 = set bit 0 = no change
0		OPR 0 1 = set bit 0 = no change

8.6.11 Reset Output Port bits OPR A and OPR B (ROPR A, ROPR B)

Ones in the byte written to the ROPR will cause the corresponding bit positions in the OPR to set to '0'. Zeros have no effect. This allows software to reset individual bits with our keeping a copy of the OPR bit configuration. One register for each channel.

Table 74: ROPR A, ROPR B - Reset Output Port Bits OPR A and OPR B register (address 0x11, 0x19) bit description

Bit	Symbol	Description
7		OPR 7 1 = reset bit 0 = no change
6		OPR 6 1 = reset bit 0 = no change
5		OPR 5 1 = reset bit 0 = no change
4		OPR 4 1 = reset bit 0 = no change
3		OPR 3 1 = set bit 0 = no change
2		OPR 2 1 = reset bit 0 = no change
1		OPR 1 1 = reset bit 0 = no change
0		OPR 0 1 = reset bit 0 = no change

8.6.12 Output Port Register A and B (OPR)

The I/O pins drive the logical inverse of the data in this register.

Table 75: OPR A, OPR B - Output Port Register A and B bit description
n = A for A, n = B for B

Bit	Symbol	Description
7		I/O7 n 0 = pin HIGH 1 = pin LOW
6		I/O6 n 0 = pin HIGH 1 = pin LOW
5		I/O5 n 0 = pin HIGH 1 = pin LOW
4		I/O4 n 0 = pin HIGH 1 = pin LOW
3		I/O3 n 0 = pin HIGH 1 = pin LOW
2		I/O2 n 0 = pin HIGH 1 = pin LOW
1		I/O1 n 0 = pin HIGH 1 = pin LOW
0		I/O0 n 0 = pin HIGH 1 = pin LOW

The I/O pins, when their control code in the IOPCR is set to code 01, drive the inverse level of the data in the OPR register. This register is **not readable** and its value is controlled by the SOPR and ROPR described in Section 8.6.10 and Section 8.6.11.

8.7 BRG characteristics

Table 76: Baud Rate Generator (BRG) characteristics

Crystal or clock = 14.7456 MHz; duty cycle of 16× clock is 50 % ± 1 %.

Normal rate (baud)	Actual 16× 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
19.2 k	307.2	0
38.4 k	614.4	0
14.4 k	230.4	0
28.8 k	460.8	0
57.6 k	921.6	0
115.2 k	1843.2	0
230.4 k	3686.4	0

9. Register maps

The registers of the SC28L201 are loosely partitioned into two groups: those used in controlling data channels, and those used in handling the actual data flow and status. Below is shown the general configuration of all the registers addressed. The Table 78 "Register map summary" shows the configuration of the address.

Any programming using the SC28L201 as it is intended would always use the address space from 0x10 through 0x7F.

Table 77: Register map detail

A[6:0]	Read	Write
001 0000 (0x10)	Receiver FIFO Fill Level (RxFL A)	Set Output Port Register (SOPR A)
001 0001 (0x11)	Transmitter FIFO Empty level (TxEL A)	Reset Output Port Register (ROPR A)
001 0010 (0x12)	Enhanced Operation Status (EOS) [1]	Command Register Extension (CRx A)
001 0011 (0x13)	Input Port Change Register Upper (IPCRU A)	I/O Port Configuration Register 0 (I/OPCR 0)
001 0100 (0x14)	Input Port Change Register Lower (IPCRL A)	I/O Port Configuration Register 1 (I/OPCR 1)
001 0101 (0x15)	Input Port Register (IPR A)	
001 0110 (0x16)	Counter Timer Value Register Upper (CTVU 0)	Counter Timer Preset Register Upper (CTPU 0)
001 0111 (0x17)	Counter Timer Value Register Lower (CTVL 0)	Counter Timer Preset Register Lower (CTPL 0)
…		
001 1000 (0x18)		Set Output Port Register (SOPR B)
001 1001 (0x19)		Reset Output Port Register (ROPR B)
001 1010 (0x1A)		Command Register Extension (CRx B)
001 1011 (0x1B)	Input Port Change Register Upper (IPCRU B)	I/O Port Configuration Register 2 (I/OPCR 2)
001 1100 (0x1C)	Input Port Change Register Lower (IPCRL B)	I/O Port Configuration Register 3 (I/OPCR 3)
001 1101 (0x1D)	Input Port Register (IPR B)	
001 1110 (0x1E)	Counter Timer Value Register Upper (CTVU 1)	Counter Timer Preset Register Upper (CTPU 1)
001 1111 (0x1F)	Counter Timer Value Register Lower (CTVL 1)	Counter Timer Preset Register Lower (CTPL 1)

[1] This register configures the whole chip.

Table 78: Register map summary

The register map for channel A (UART A).

A[6:0]	Read	Write
010 0000 (0x20)	Mode Register 0 (MR0 A)	Mode Register 0 (MR0 A)
010 0001 (0x21)	Mode Register 1 (MR1 A)	Mode Register 1 (MR1 A)
010 0010 (0x22)	Mode Register 2 (MR2 A)	Mode Register 2 (MR2 A)
010 0011 (0x23)	Mode Register 3 (MR3 A)	Mode Register 3 (MR3 A)
010 0100 (0x24)	Counter/Timer Clock Source (CTCS 0)	Counter/Timer Clock Source (CTCS 0)
010 0101 (0x25)	Interrupt Status Register (ISR A)	Interrupt Mask Register (IMR A)
010 0110 (0x26)	Programmable BRG Preset Lower (PBRGPL 0)	Programmable BRG Preset Lower (PBRGPL 0)
010 0111 (0x27)	Programmable BRG Preset Upper (PBRGPU 0)	Programmable BRG Preset Upper (PBRGPU 0)
…		
010 1000 (0x28)		
010 1001 (0x29)		
010 1010 (0x2A)		
010 1011 (0x2B)		
010 1100 (0x2C)	Counter/Timer Clock Source (CTCS 1)	Counter/Timer Clock Source (CTCS 1)
010 1101 (0x2D)		
010 1110 (0x2E)		
010 1111 (0x2F)		
…		
011 0000 (0x30)	Receiver Clock Select Register (RxCSR A)	Receiver Clock Select Register (RxCSR A)
011 0001 (0x31)	Transmitter Clock Select Register (TxCSR A)	Transmitter Clock Select Register (TxCSR A)
011 0010 (0x32)	Input Port Change Interrupt Enable (IPCE A)	Input Port Change Interrupt Enable (IPCE A)
011 0011 (0x33)	Programmable BRG Clock Source (PBRGCS)	Programmable BRG Clock Source (PBRGCS)
011 0100 (0x34)		
011 0101 (0x35)		
011 0110 (0x36)	Programmable BRG Preset Lower (PBRGPL 1)	Programmable BRG Preset Lower (PBRGPL 1)
011 0111 (0x37)	Programmable BRG Preset Upper (PBRGPU 1)	Programmable BRG Preset Upper (PBRGPU 1)
…		
011 1000 (0x38)		
011 1001 (0x39)		
011 1010 (0x3A)	Input Port Change Interrupt Enable (IPCE B)	Input Port Change Interrupt Enable (IPCE B)
011 1011 (0x3B)		
011 1100 (0x3C)		
011 1101 (0x3D)		
011 1110 (0x3E)		
011 1111 (0x3F)		
…		

Table 78: Register map summary *…continued*

The register map for channel A (UART A).

A[6:0]	Read	Write
100 0000 (0x40)	System Enable Status (SES A)	Watchdog, Character and X Enable (WCXER A)
100 0001 (0x41)	Xon Character Register (XonCR A)	Xon Character Register (XonCR A)
100 0010 (0x42)	Xoff Character Register (XoffCR A)	Xoff Character Register (XoffCR A)
100 0011 (0x43)	Address Recognition Character (ARCR A)	Address Recognition Character (ARCR A)
100 0100 (0x44)	Xon/Xoff Interrupt Status Register (XISR A)	
100 0101 (0x45)	Special Function Register (SFR A)	Special Function Register (SFR A)
100 0110 (0x46)	Receiver FIFO Interrupt Level (RxFIL A)	Receiver FIFO Interrupt Level (RxFIL A)
100 0111 (0x47)	Transmitter FIFO Interrupt Level (TxFIL A)	Transmitter FIFO Interrupt Level (TxFIL A)
…		
101 0000 (0x50)	Bidding Control Register, Break Change (BCRBRK A)	Bidding Control Register, Break Change (BCRBRK A)
101 0001 (0x51)	Bidding Control Register, Change-Of-State (BCRCOS A)	Bidding Control Register, Change-Of-State (BCRCOS A)
101 0010 (0x52)	Bidding Control Register, Counter/Timer (BCRCT A)	Bidding Control Register, Counter/Timer (BCRCT A)
101 0011 (0x53)	Bidding Control Register, Xon (BCRx A)	Bidding Control Register, Xon (BCRx A)
101 0100 (0x54)	Bidding Control Register, Address (BCRA A)	Bidding Control Register, Address (BCRA A)
101 0101 (0x55)	Bidding Control Register, Loop Back Error (BCRLBE A)	Bidding Control Register, Loop Back Error (BCRLBE A)
101 0110 (0x56)		
101 0111 (0x57)		
…		
101 1010 (0x5A)	Bidding Control Register, Counter/Timer (BCRCT B)	Bidding Control Register, Counter/Timer (BCRCT B)
101 1110 (0x5E)		
101 1111 (0x5F)		

Table 78: Register map summary *…continued*

The register map for channel A (UART A).

A[6:0]	Read	Write
Global		
110 0000 (0x60)	Interrupt Control Register (ICR) [1]	Interrupt Control Register (ICR) [1]
110 0001 (0x61)	Current Interrupt Register (CIR) [1]	Update Current Interrupt Register (UCIR) [1]
110 0010 (0x62)		
110 0011 (0x63)		
110 0100 (0x64)	Interrupt Vector Register (IVR) [1]	Interrupt Vector Register (IVR) [1]
110 0101 (0x65)		
110 0110 (0x66)	Global Chip Configuration Register (GCCR) [1]	Global Chip Configuration Register (GCCR) [1]
110 0111 (0x67)	Test and Revision Register (TRR) [1]	Test and Revision Register (TRR) [1]
…		
110 1000 (0x68)		
110 1001 (0x69)		
110 1010 (0x6A)		
110 1011 (0x6B)		
110 1100 (0x6C)		
110 1101 (0x6D)		
110 1110 (0x6E)		
110 1111 (0x6F)		
…		
111 0000 (0x70)	Global Interrupt Channel Register (GICR) [1]	
111 0001 (0x71)	Global Interrupt Byte Count Register (GIBCR) [1]	
111 0010 (0x72)	Global Interrupt Type Register (GITR) [1]	
111 0011 (0x73)	Global RxFIFO Register (GRxFIFO) [1]	Global TxFIFO Register (GTxFIFO) [1]
111 0100 (0x74)		
111 0101 (0x75)		
111 0110 (0x76)		
111 0111 (0x77)	Scan Test Control Register (STCR) [1]	Scan Test Control Register (STCR) [1]
…		
111 1000 (0x78)		
111 1001 (0x79)		
111 1010 (0x7A)		
111 1011 (0x7B)		
111 1100 (0x7C)		
111 1101 (0x7D)		
111 1110 (0x7E)		
111 1111 (0x7F)		

[1] This register configures the whole chip.

10. Limiting values

Table 79: Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
T_{amb}	ambient temperature	operating in free air	[1] [2]	–40	+85	°C
T_{stg}	storage temperature			–65	+150	°C
V_{DD}	supply voltage		[3]	–0.5	+7.0	V
V_{SS}	ground supply voltage		[3]	–0.5	V_{DD} + 0.5	V
P	power dissipation			-	2.4	W
$R_{th(j-c)}$	thermal resistance from junction to case			-	20	K/W
$R_{th(j-a)}$	thermal resistance from junction to ambient			-	83	K/W

[1] For operation at elevated temperatures, the device must be derated based on +150 °C maximum junction temperature.

[2] Parameters are valid over specified temperature range.

[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.

11. Static characteristics

Table 80: Static characteristics, nominal 5 V operation

V_{DD} = 5 V ± 10 %; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	LOW-state input voltage		[2]	-	-	0.8	V
V_{IH}	HIGH-state input voltage	except pin X1/SCLK	[2]	2.4	-	-	V
		pin X1/SCLK	[2]	$0.8V_{DD}$	-	-	V
V_{OL}	LOW-state output voltage	I_{OL} = 4 mA	[2]	-	-	0.4	V
V_{OH}	HIGH-state output voltage	except open-drain outputs; I_{OH} = −400 µA	[2] [3]	V_{DD} − 0.5	-	-	V
$I_{I(PD)(X1/SCLK)}$	Power-down mode input current on pin X1/SCLK	V_I = 0 V to V_{DD}		−1	-	1	µA
$I_{IL(X1/SCLK)}$	LOW-state input current on pin X1/SCLK	operating mode; V_I = 0 V		−30	-	0	µA
$I_{IH(X1/SCLK)}$	HIGH-state input current on pin X1/SCLK	operating mode; V_I = V_{DD}		0	-	30	µA
I_{LI}	input leakage current	V_I = 0 V to V_{DD}					
		input port and IACKN pins	[4]	−10	-	1	µA
		all other pins	[4]	−1	-	1	µA
I_{OZH}	HIGH off-state output current	3-state data bus; V_I = V_{DD}		-	-	5	µA
I_{OZL}	LOW off-state output current	3-state data bus; V_I = 0 V		−5	-	-	µA
I_{ODL}	LOW off-state open-drain output current	V_I = 0 V		−10	-	-	µA
I_{ODH}	HIGH off-state open-drain output current	V_I = V_{DD}		-	-	1	µA
I_{DD}	supply current	CMOS input levels	[5]				
		operating mode; f = 10 MHz		-	9	20	mA
		Power-down mode; f = 0 MHz		-	200	500	µA

[1] Typical values are at +25 °C, typical supply voltages, and typical processing parameters.

[2] All voltage measurements are referenced to ground. For testing, all inputs swing between 0.4 V and 3.0 V with a transition time of 5 ns maximum. For X1/SCLK this swing is between 0.4 V and 4.4 V. All time measurements are referenced at input voltages of 0.8 V and 2.0 V, and output voltages of 0.8 V and 2.0 V, as appropriate.

[3] Test conditions for outputs: C_L = 85 pF, except interrupt outputs. Test conditions for interrupt outputs: C_L = 85 pF, R_L = 2.7 kΩ to V_{DD}.

[4] I/O port and IACKN pins have active pull-up transistors that will source a typical 2 µA from V_{DD} when they are at V_{SS}. At V_{DD} they source 0.0 µA.

[5] All outputs are disconnected. Inputs are switching between CMOS levels of V_{DD} − 0.2 V and V_{SS} + 0.2 V.

Table 81: Static characteristics, nominal 3.3 V operation

V_{DD} = 3.3 V ± 10 %; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	LOW-state input voltage		[2]	-	-	$0.2V_{DD}$	V
V_{IH}	HIGH-state input voltage	except pin X1/SCLK	[2]	2.4	-	-	V
		pin X1/SCLK	[2]	$0.8V_{DD}$	-	-	V
V_{OL}	LOW-state output voltage	I_{OL} = 4 mA	[2]	-	-	0.4	V
V_{OH}	HIGH-state output voltage	except open-drain outputs; I_{OH} = −400 µA	[2] [3]	V_{DD} − 0.5	-	-	V
$I_{I(PD)(X1/SCLK)}$	Power-down mode input current on pin X1/SCLK	V_I = 0 V to V_{DD}		−1	-	1	µA
$I_{IL(X1/SCLK)}$	LOW-state input current on pin X1/SCLK	operating mode; V_I = 0 V		−30	-	0	µA
$I_{IH(X1/SCLK)}$	HIGH-state input current on pin X1/SCLK	operating mode; V_I = V_{DD}		0	-	30	µA
I_{LI}	input leakage current	V_I = 0 V to V_{DD}					
		input port and IACKN pins	[4]	−10	-	1	µA
		all other pins	[4]	−1	-	1	µA
I_{OZH}	HIGH off-state output current	3-state data bus; V_I = V_{DD}		0	-	5	µA
I_{OZL}	LOW off-state output current	3-state data bus; V_I = 0 V		−5	-	-	µA
I_{ODL}	LOW off-state open-drain output current	V_I = 0 V		−10	-	-	µA
I_{ODH}	HIGH off-state open-drain output current	V_I = V_{DD}		-	-	1	µA
I_{DD}	supply current	CMOS input levels	[5]				
		operating mode; f = 10 MHz		-	9	20	mA
		Power-down mode; f = 0 MHz		-	200	500	µA

[1] Typical values are at +25 °C, typical supply voltages, and typical processing parameters.

[2] All voltage measurements are referenced to ground. For testing, all inputs swing between 0.4 V and 3.0 V with a transition time of 5 ns maximum. For X1/SCLK this swing is between 0.4 V and 4.4 V. All time measurements are referenced at input voltages of 0.8 V and 2.0 V, and output voltages of 0.8 V and 2.0 V, as appropriate.

[3] Test conditions for outputs: C_L = 85 pF, except interrupt outputs. Test conditions for interrupt outputs: C_L = 85 pF, R_L = 2.7 kΩ to V_{DD}.

[4] I/O port pins have active pull-up transistors that will source a typical 2 µA from V_{DD} when they are at V_{SS}. At V_{DD} they source 0.0 µA.

[5] All outputs are disconnected. Inputs are switching between CMOS levels of V_{DD} − 0.2 V and V_{SS} + 0.2 V.

12. Dynamic characteristics

Table 82: Dynamic characteristics

T_{amb} = −40 °C to +85 °C; voltage tolerance ± 10 %; unless otherwise specified. [1]

Symbol	Parameter	Conditions		V_{DD} = 3.3 V		V_{DD} = 5 V		Unit
				Min	Max	Min	Max	
Reset timing (see Figure 6 and Figure 7)								
$t_{w(RESET)}$	pulse width on pin RESET			100	-	100	-	ns
Bus timing [2] (see Figure 8)								
$t_{su(An-RWL)}$	An to RDN and WRN LOW setup time			10	-	10	-	ns
$t_{h(RWL-An)}$	RDN and WRN LOW to An hold time			10	-	10	-	ns
$t_{su(RWL-CEN)}$	CEN to RDN and WRN LOW setup time			0	-	0	-	ns
$t_{h(RWH-CEN)}$	RDN and WRN HIGH to CEN hold time			0	-	0	-	ns
$t_{w(RWL)}$	RDN and WRN LOW pulse width			40	-	30	-	ns
$t_{Dv(RL)}$	RDN LOW to data valid time	85 pF load		-	-	-	30	ns
		125 pF load		-	40	-	-	ns
$t_{d(RL-D)(act)}$	RDN LOW to data bus active delay time		[3]	0	-	0	-	ns
$t_{d(D)(float)}$	RDN or CEN HIGH to data bus floating delay time			-	15	-	15	ns
$t_{d(RH,CENH-DX)}$	RDN or CEN HIGH to data bus invalid delay time		[4]	-	-	0	-	ns
$t_{su(D)}$	data bus to WRN or CEN HIGH setup time	write cycle		15	-	15	-	ns
$t_{h(D)}$	data hold time after WRN HIGH			0	-	0	-	ns
$t_{d(RWD)}$	read and write cycle HIGH time		[2] [4]	10	-	10	-	ns
Port timing [2] (see Figure 12)								
$t_{su(port)}$	port to RDN LOW setup time	read IP ports cycle		0	-	0	-	ns
$t_{h(port)}$	RDN HIGH to input port hold time			0	-	0	-	ns
$t_{v(port)}$	WRN or CEN HIGH to output port valid time	OPR write cycle		-	40	-	30	ns

Table 82: Dynamic characteristics *...continued*

T_{amb} = −40 °C to +85 °C; voltage tolerance ± 10 %; unless otherwise specified. [1]

Symbol	Parameter	Conditions		V_{DD} = 3.3 V		V_{DD} = 5 V		Unit
				Min	Max	Min	Max	
Interrupt timing (see Figure 13)								
$t_{d(int)}$	interrupt delay time	IRQN (or I/O[7:3]B when used as interrupts) negated from						
		read RxFIFO (RxRDY/FFULL interrupt)		-	40	-	30	ns
		write TxFIFO (TxRDY interrupt)		-	40	-	30	ns
		reset command (delta break change interrupt)		-	40	-	30	ns
		stop C/T command (Counter/Timer interrupt)		-	40	-	30	ns
		read IPCR (delta input port change interrupt)		-	40	-	30	ns
		write IMR (clear of change interrupt mask bit(s))		-	40	-	30	ns
Clock timing (see Figure 14, Figure 15, and Figure 16)								
t_{CLK}	X1/SCLK HIGH or LOW time			10		8		ns
$f_{X1/SCLK}$	clock frequency on pin X1/SCLK	7.0 MHz to 16.2 MHz with crystal		1	34	1	50	MHz
t_{CTC}	Counter/Timer clock HIGH or LOW time	IP2; C/T external clock input		10	-	10	-	ns
f_{CTC}	Counter/Timer clock frequency	IP2	[5]	0	8	0	20	MHz
t_{RX}	RxC HIGH or LOW time (16X)			10	-	8	-	ns
$f_{RX(16)}$	RxC frequency (16X)			0	24	0	50	MHz
$f_{RX(1)}$	RxC frequency (1X)		[5] [6]	0	1.5	0	3	MHz
t_{TX}	TxC HIGH or LOW time			10	-	8	-	ns
$f_{TX(16)}$	TxC frequency (16X)			0	24	-	50	MHz
$f_{TX(1)}$	TxC frequency (1X)		[5] [6]	0	1.5	0	3	MHz
Transmitter timing (see Figure 15 and Figure 17)								
$t_{d(o)(TXD)}$	TxC LOW to TXD output delay time	TxC input pin		-	40	-	30	ns
t_{TCS}	TxC output pin LOW to TXD data output delay time			-	40	-	30	ns
Receiver timing (see Figure 16 and Figure 18)								
$t_{su(RXD-RXCH)(D)}$	RXD to RxC HIGH data setup time			20	-	20	-	ns
$t_{h(RXCH-RXD)(D)}$	RxC HIGH to RXD data hold time			20	-	20	-	ns
68000 (Motorola) bus timing (see Figure 9, Figure 10, and Figure 11)								
$t_{su(RWL-CEN)(mot)}$	CEN LOW to RWN setup time			5	-	5	-	ns
$t_{su(D)(mot)}$	data bus to X1/SCLK HIGH setup time		[7]	10	-	10	-	ns

Table 82: Dynamic characteristics *…continued*

T_{amb} = −40 °C to +85 °C; voltage tolerance ± 10 %; unless otherwise specified. [1]

Symbol	Parameter	Conditions	V_{DD} = 3.3 V		V_{DD} = 5 V		Unit
			Min	Max	Min	Max	
$t_{h(D)(mot)}$	CEN HIGH to data hold time		0	-	0	-	ns
$t_{su(An-CENL)(mot)}$	address to CEN LOW setup time		10	-	10	-	ns
$t_{h(CENL-An)(mot)}$	CEN LOW to address hold time		10	-	10	-	ns
$t_{Dv(CENL)}$	CEN LOW to data valid time		-	45	-	35	ns
$t_{d(RWD)(mot)}$	read and write cycle HIGH time	[2] [4]	10	-	10	-	ns
t_{DCR}	X1/SCLK HIGH to DACKN LOW time (read cycle)		-	35	-	35	ns
t_{DCW}	X1/SCLK HIGH to DACKN LOW time (write cycle)		-	30	-	25	ns
t_{DAT}	CEN to IACKN HIGH DACKN high-impedance time		-	15	-	15	ns
t_{CSC}	CEN or IACKN to X1/SCLK HIGH setup time	for minimum DACKN cycle	10	-	10	-	ns

[1] Test conditions for outputs: C_L = 85 pF, except interrupt outputs. Test conditions for interrupt outputs: C_L = 85 pF, R_L = 2.7 kΩ to V_{DD}.

[2] Timing is illustrated and referenced to the WRN and RDN inputs. Also, CEN may be the ‘strobing’ input. CEN and RDN (also CEN and WRN) are ORed internally. The signal asserted last initiates the cycle and the signal negated first terminates the cycle.

[3] Guaranteed by characterization of sample units.

[4] If CEN is used as the ‘strobing’ input, the parameter defines the minimum HIGH times between one CEN and the next. The RDN signal must be negated for $t_{d(RWD)}$ time to guarantee that any Status Register changes are valid.

[5] Minimum frequencies are not tested, but are guaranteed by design.

[6] Clocks for 1× mode should be reasonably symmetrical.

[7] Data is usually setup with respect to CEN going LOW (the leading edge of CEN). This mode strongly implies the use of DACKN. (Its use is not strictly required.) DACKN is derived from the X1/SCLK input. It is seldom that the system clocks that ultimately drive the CEN, address, and R/WN signals are synchronous to the X1/SCLK. If address, data, R/WN are setup **before** CEN goes LOW and hold through DACKN, the timing parameters above will be guaranteed.

12.1 Timing diagrams

The active time of read or write cycle exists only when CEN is LOW and RDN or CEN is also LOW.

Write = CEN and WRN LOW.

Read = CEN and RDN LOW.

For the 68000 mode:

Write = CEN LOW and R/WN LOW and DACKN HIGH.

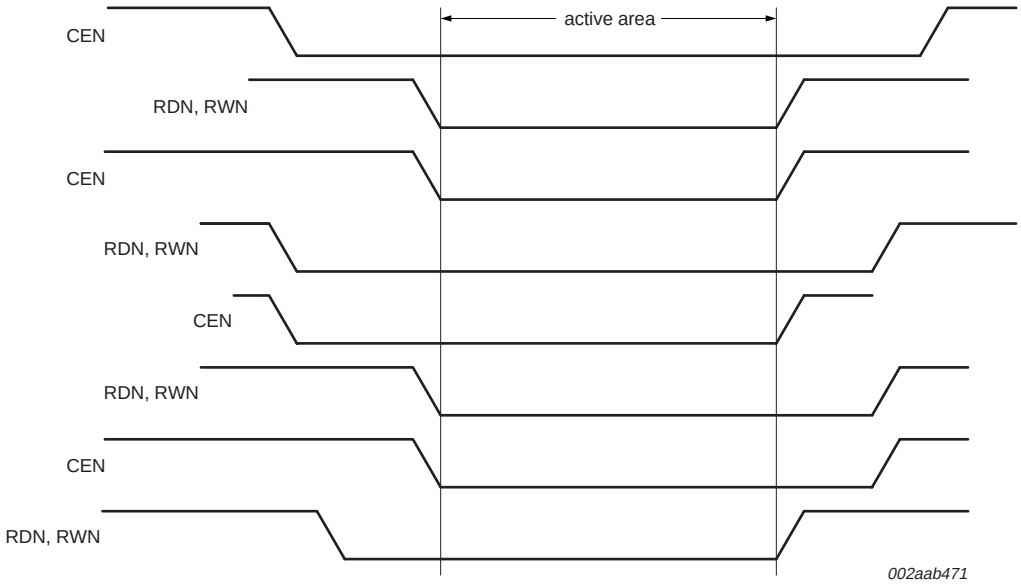
Read = CEN LOW and R/WN HIGH.

In general, it is convenient (but is not at all required) to think of the Read/Write signal to be active and then let the CEN be the ‘strobing’ or clocking control. However, some users have wired CEN LOW and allowed RDN or WRN to be the clocking or strobing input. While this is completely within the specified limits, it is not recommended since it will greatly increase the part’s sensitivity to noise ‘glitches’ on the RDN and WRN signals.

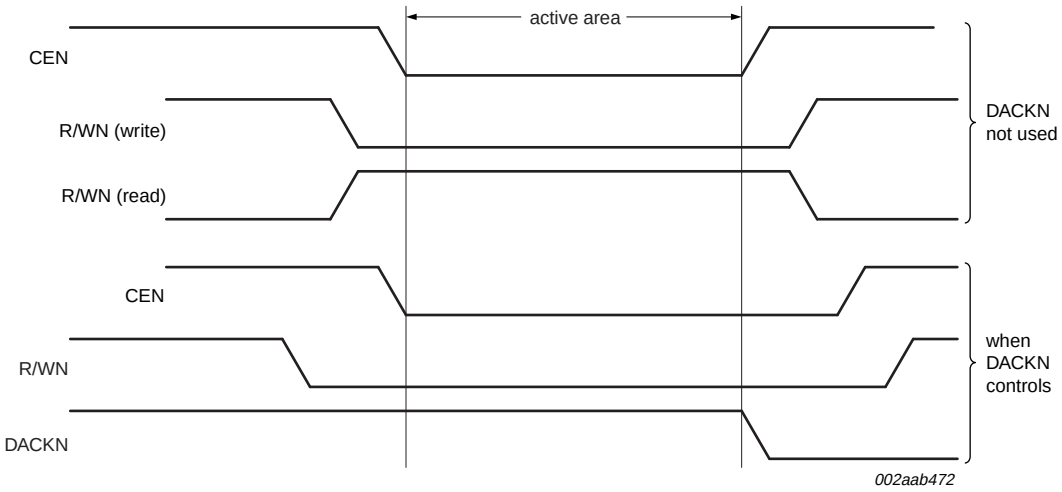
For the 68000 mode, the CEN is very much the clock or ‘strobing’ signal. The RDN and WRN signals have been combined into the R/WN signal. Therefore, the part is **always** prepared to do a write or read; it only needs CEN to enable.

In the 68000 mode design, care should be given to system drift over temperature, voltage, and age when R/WN and CEN change very close to each other. If R/WN switches shortly before CEN (due to system drift) it is possible to produce very short internal read or write pulses which could change internal controls, FIFO address pointers, for example.

Figure 5 loosely shows the timing conditions that may exist of the active area those signals will produce.



a. Intel-type bus



b. Motorola-type:

Remark: In the 68000 mode, the write occurs on the rise of CEN or fall of DACKN, whichever occurs **first**.

Fig 5. Active area in 68000 mode

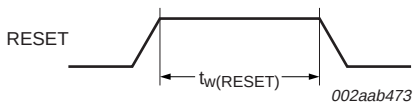


Fig 6. Reset timing (80xxx mode)

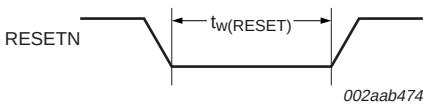


Fig 7. Reset timing (68000 mode)

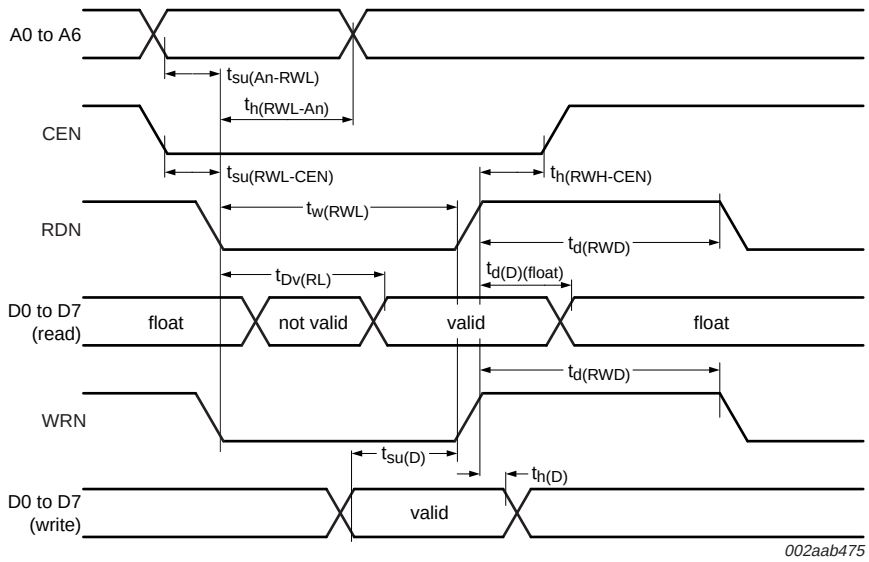
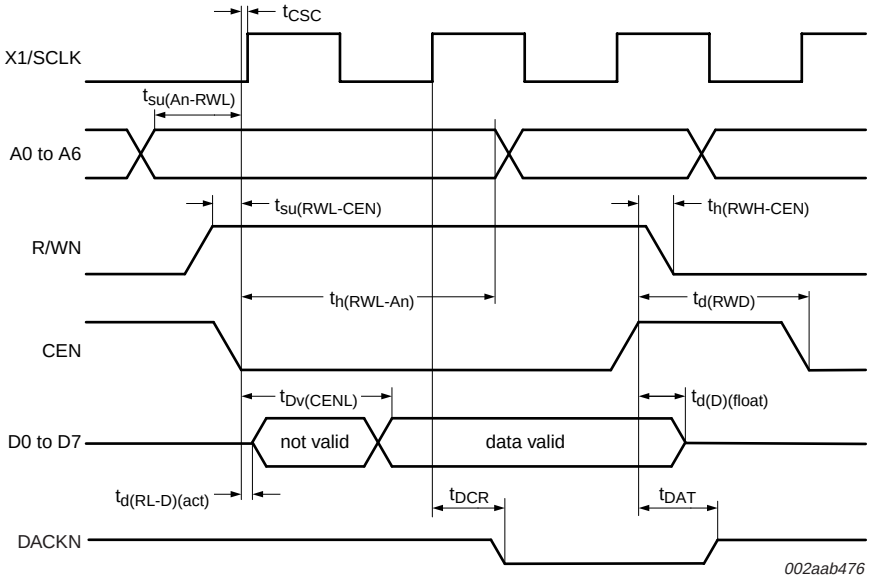
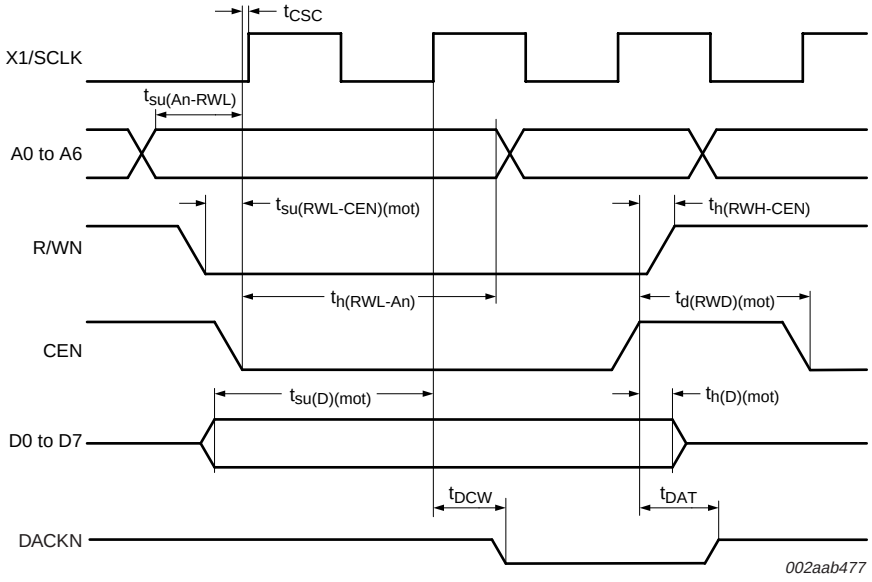


Fig 8. Bus timing (80xxx mode)



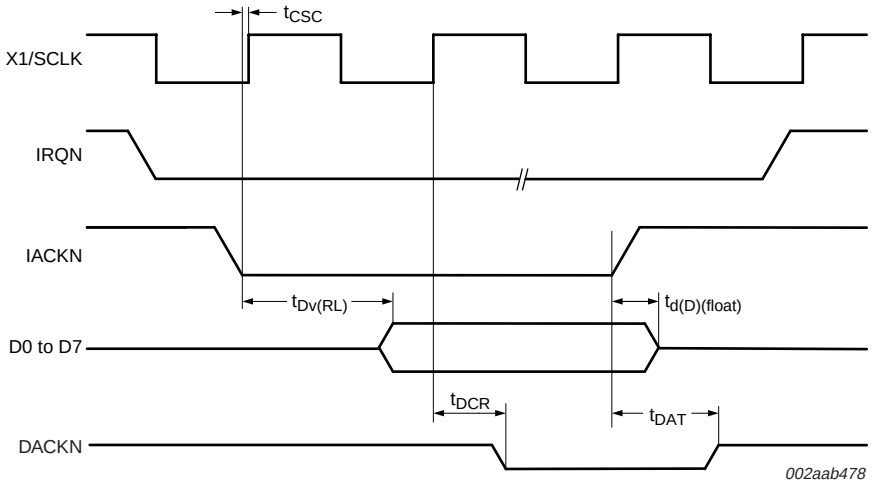
DACKN LOW requires two rising edges of X1 clock after CEN is LOW.

Fig 9. Bus timing, read cycle (68000 mode)



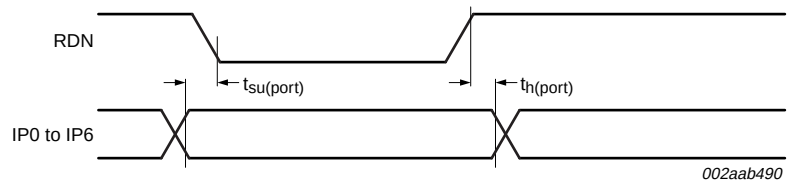
DACKN LOW requires two rising edges of X1 clock after CEN is LOW.

Fig 10. Bus timing, write cycle (68000 mode)

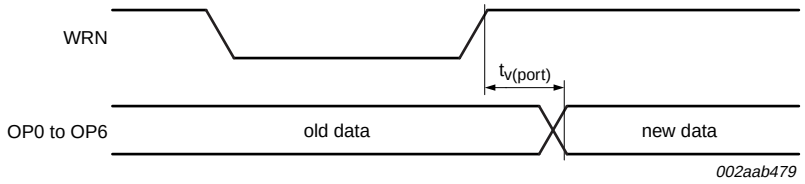


DACKN LOW requires two rising edges of X1 clock after CEN is LOW.

Fig 11. Interrupt cycle timing (68000 mode)

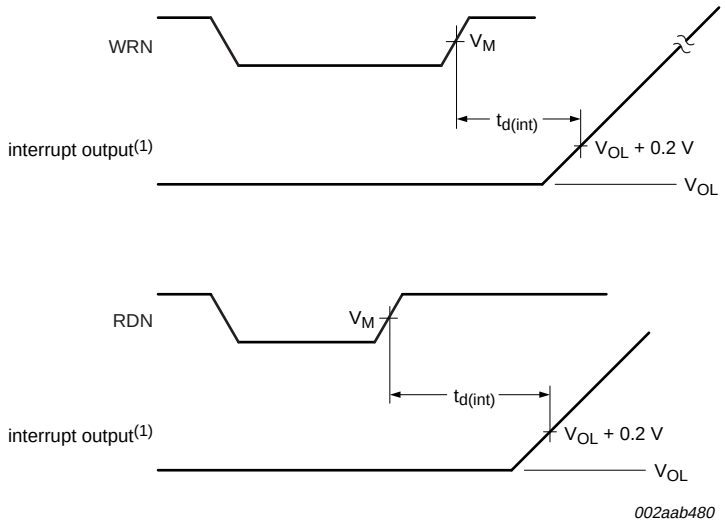


a. Input pins



b. Output pins

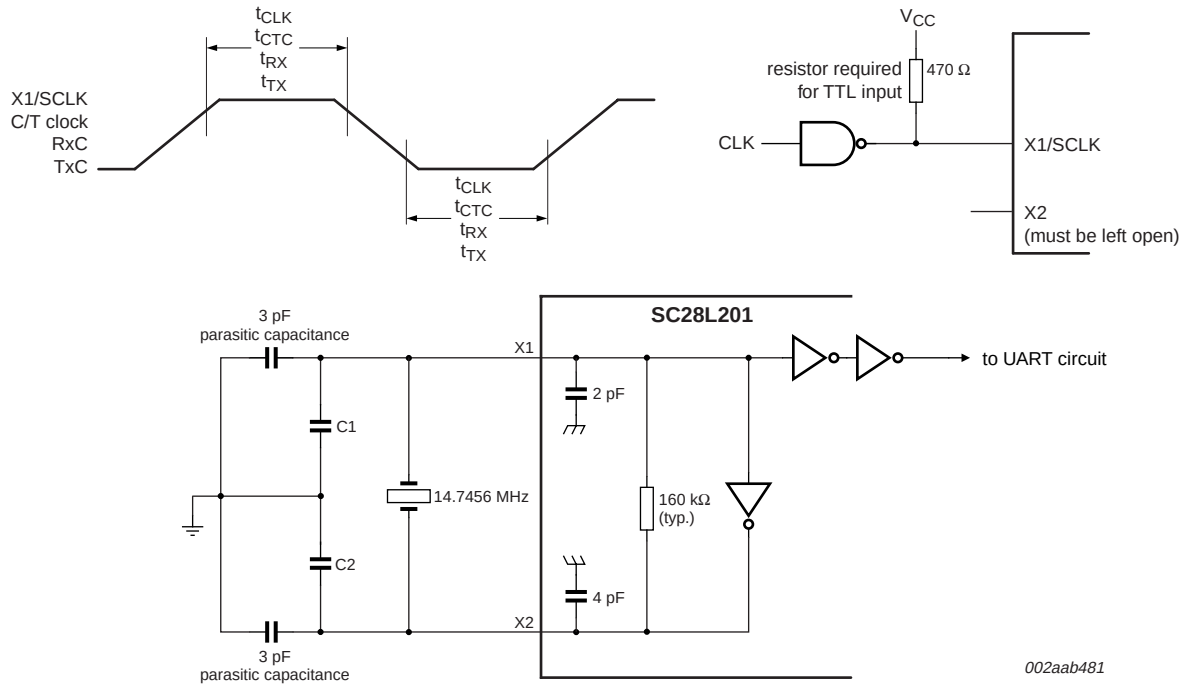
Fig 12. Port timing



(1) IRQN or OP3 to OP7 when used as interrupt outputs.

(2) The test for open-drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.2 V 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.

Fig 13. Interrupt timing (80xxx mode)



C1 = C2 ~ 24 pF for C_L = 13.5 pF. For the oscillator feedback loop, the capacitors C1 and C2 are in series.

C1 and C2 should be chosen according to the crystal manufacturer's specification.

C1 and C2 values will include any parasitic capacitance of the wiring and X1, X2 pins.

Package capacitance approximately 4 pF.

Fig 14. Clock timing

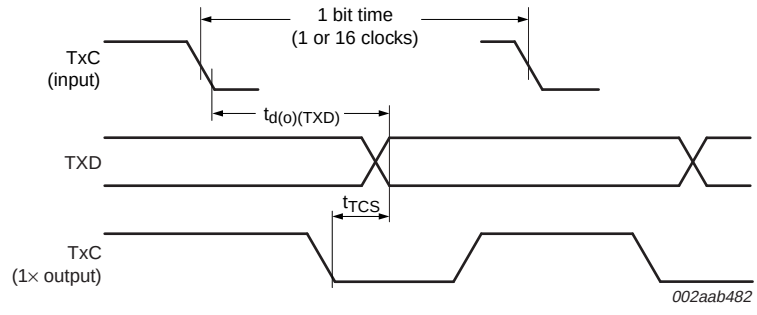


Fig 15. Transmitter external clocks

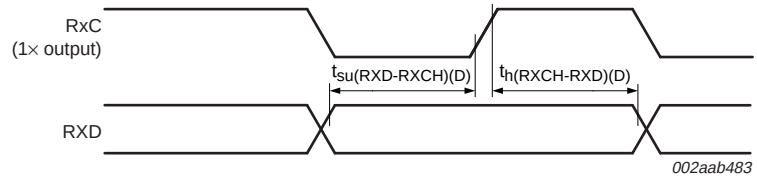
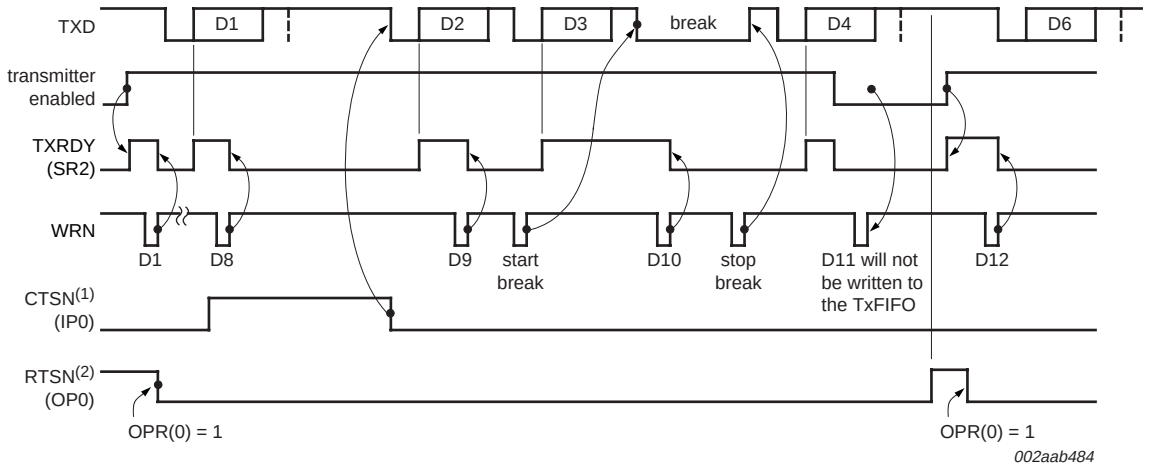


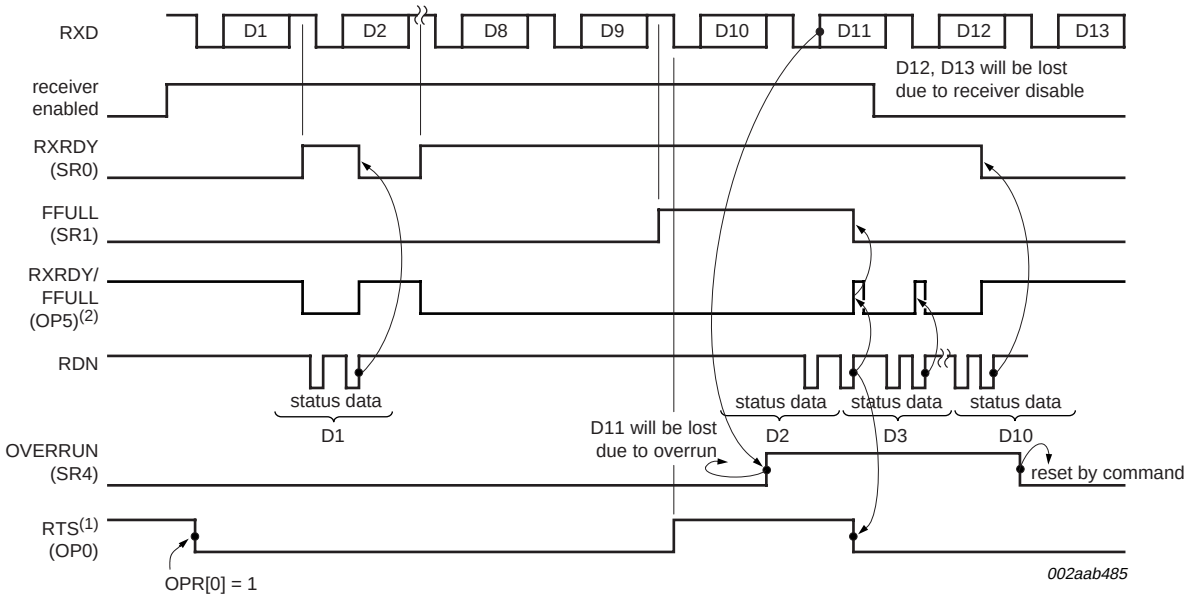
Fig 16. Receiver external clock



(1) Timing shown for MR2[4] = 1.

(2) Timing shown for MR2[5] = 1.

Fig 17. Transmitter timing



(1) Timing shown for MR1[7] = 1.

(2) Shown for OPCR[4] = 1 and MR[6] = 0.

Fig 18. Receiver timing

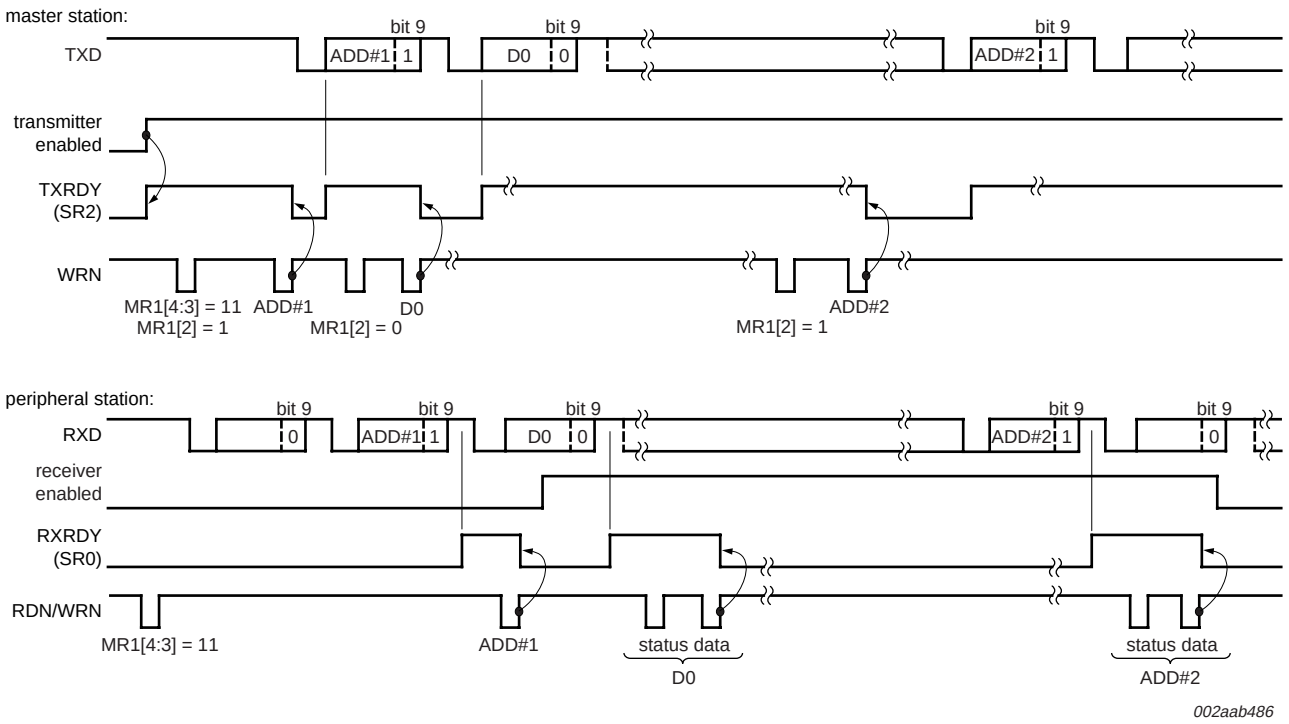


Fig 19. Wake-up mode timing

13. Test information

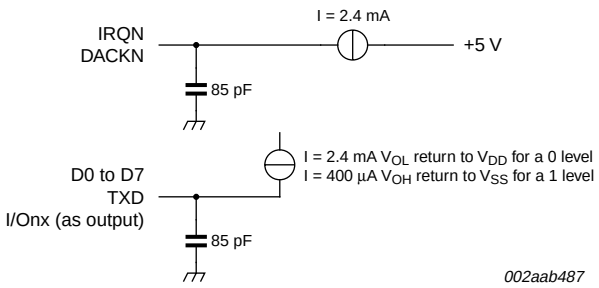
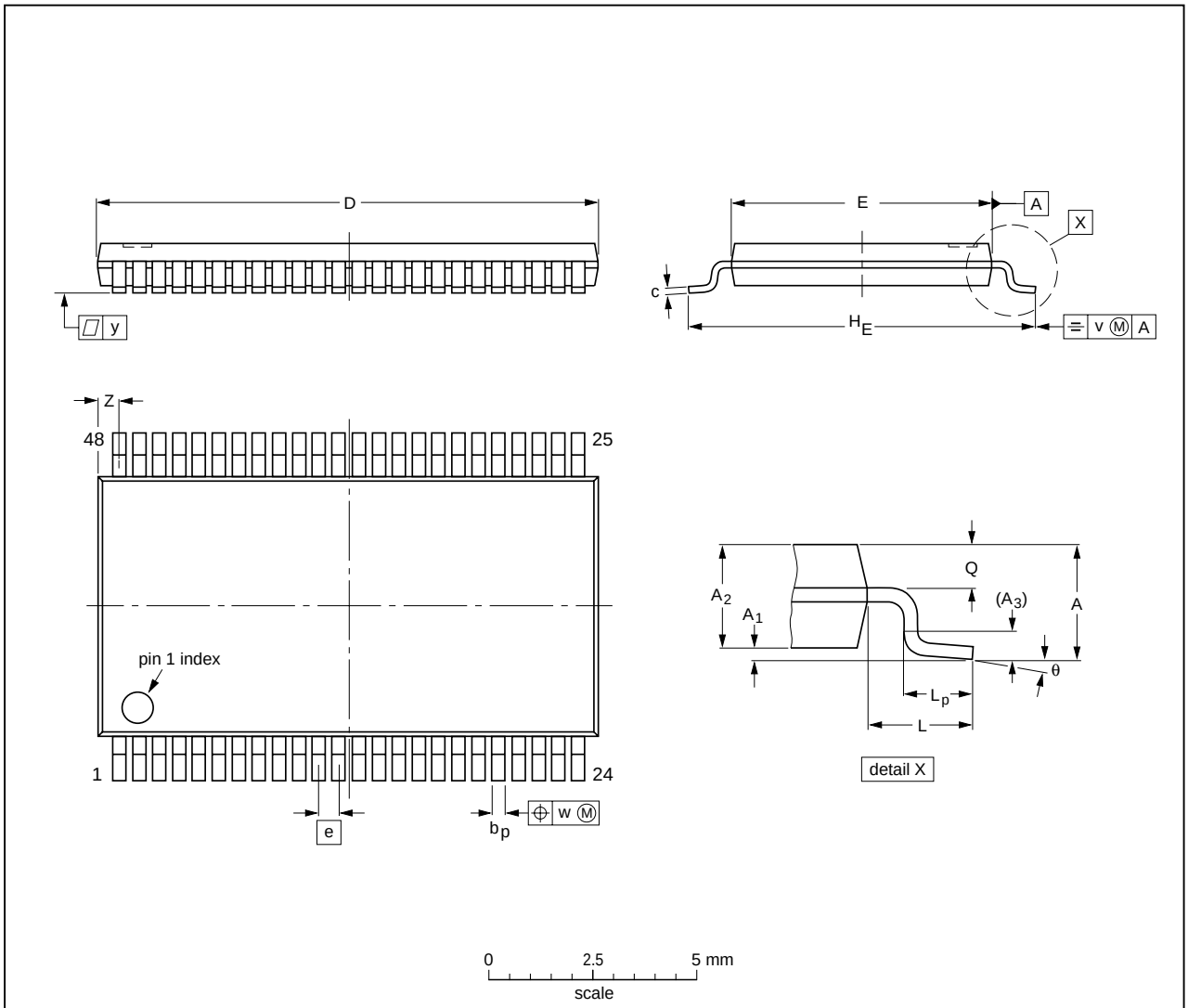


Fig 20. Test conditions on outputs

14. Package outline

TSSOP48: plastic thin shrink small outline package; 48 leads; body width 6.1 mm **SOT362-1**



DIMENSIONS (mm are the original dimensions).

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	Z	θ
mm	1.2	0.15 0.05	1.05 0.85	0.25	0.28 0.17	0.2 0.1	12.6 12.4	6.2 6.0	0.5	8.3 7.9	1	0.8 0.4	0.50 0.35	0.25	0.08	0.1	0.8 0.4	8^o 0^o

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT362-1		MO-153				~~99-12-27~~ 03-02-19

Fig 21. Package outline SOT362-1 (TSSOP48)

15. Soldering

15.1 Introduction to soldering surface mount packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *Data Handbook IC26; Integrated Circuit Packages* (document order number 9398 652 90011).

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

15.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement. Driven by legislation and environmental forces the worldwide use of lead-free solder pastes is increasing.

Several methods exist for reflowing; for example, convection or convection/infrared heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 seconds and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 °C to 270 °C depending on solder paste material. The top-surface temperature of the packages should preferably be kept:

- below 225 °C (SnPb process) or below 245 °C (Pb-free process)
  - for all BGA, HTSSON..T and SSOP..T packages
  - for packages with a thickness ≥ 2.5 mm
  - for packages with a thickness < 2.5 mm and a volume ≥ 350 mm^3 so called thick/large packages.
- below 240 °C (SnPb process) or below 260 °C (Pb-free process) for packages with a thickness < 2.5 mm and a volume < 350 mm^3 so called small/thin packages.

Moisture sensitivity precautions, as indicated on packing, must be respected at all times.

15.3 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;

– smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

The footprint must incorporate solder thieves at the downstream end.

- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time of the leads in the wave ranges from 3 seconds to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

15.4 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 seconds to 5 seconds between 270 °C and 320 °C.

15.5 Package related soldering information

Table 83: Suitability of surface mount IC packages for wave and reflow soldering methods

Package [1]	Soldering method	
	Wave	Reflow [2]
BGA, HTSSON..T [3], LBGA, LFBGA, SQFP, SSOP..T [3], TFBGA, VFBGA, XSON	not suitable	suitable
DHVQFN, HBCC, HBGA, HLQFP, HSO, HSOP, HSQFP, HSSON, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable [4]	suitable
PLCC [5], SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended [5] [6]	suitable
SSOP, TSSOP, VSO, VSSOP	not recommended [7]	suitable
CWQCCN..L [8], PMFP [9], WQCCN..L [8]	not suitable	not suitable

[1] For more detailed information on the BGA packages refer to the *(LF)BGA Application Note* (AN01026); order a copy from your Philips Semiconductors sales office.

[2] All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods*.

[3] These transparent plastic packages are extremely sensitive to reflow soldering conditions and must on no account be processed through more than one soldering cycle or subjected to infrared reflow soldering with peak temperature exceeding 217 °C ± 10 °C measured in the atmosphere of the reflow oven. The package body peak temperature must be kept as low as possible.

[4] These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.

[5] If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.

[6] Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.

[7] Wave soldering is suitable for SSOP, TSSOP, VSO and VSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

[8] Image sensor packages in principle should not be soldered. They are mounted in sockets or delivered pre-mounted on flex foil. However, the image sensor package can be mounted by the client on a flex foil by using a hot bar soldering process. The appropriate soldering profile can be provided on request.

[9] Hot bar soldering or manual soldering is suitable for PMFP packages.

16. Abbreviations

Table 84: Abbreviations

Acronym	Description
DMA	Direct Memory Access
UART	Universal Asynchronous Receiver/Transmitter
FIFO	First In/First Out
CPU	Central Processing Unit
COS	Change-Of-State
BRG	Baud Rate Generator
MIDI	Musical Instrument Digital Interface
C/T	Counter/Timer

17. Revision history

Table 85: Revision history

Document ID	Release date	Data sheet status	Change notice	Doc. number	Supersedes
SC28L201_1	20051031	Product data sheet		9397 750 13138	-

18. Data sheet status

Level	Data sheet status [1]	Product status [2] [3]	Definition
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

[1] Please consult the most recently issued data sheet before initiating or completing a design.

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

[3] For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

19. Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

20. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

21. Trademarks

Notice — All referenced brands, product names, service names and trademarks are the property of their respective owners.

22. Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, send an email to: **sales.addresses@www.semiconductors.philips.com**

23. Contents

continued >>

Date of release: 31 October 2005
Document number: 9397 750 13138

Published in The Netherlands

ООО «НИОКРсистемс» - это оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов. Реализуемая нашей компанией продукция насчитывает более полумиллиона наименований.

Благодаря этому наша компания предлагает к поставке практически не ограниченный ассортимент компонентов как оптовыми, мелкооптовыми партиями, так и в розницу.

Благодаря развитой сети поставщиков, помогаем в поиске и приобретении экзотичных или снятых с производства компонентов.

Наша компания это:

- Гарантия качества поставляемой продукции
- Широкий ассортимент
- Минимальные сроки поставок
- Техническая поддержка
- Подбор комплектации
- Индивидуальный подход
- Гибкое ценообразование
- Работаем по 275 ФЗ

ООО «НИОКР СИТЕМС»
Москва, 125362, Россия, ул. Вишневая, д. 9, к. 1, Блок 104 а и 104 в

Телефон: 8 (495) 268-14-82
Email: n@nsistems.ru

ИНН: 7735154786
ОГРН: 1167746717709