Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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

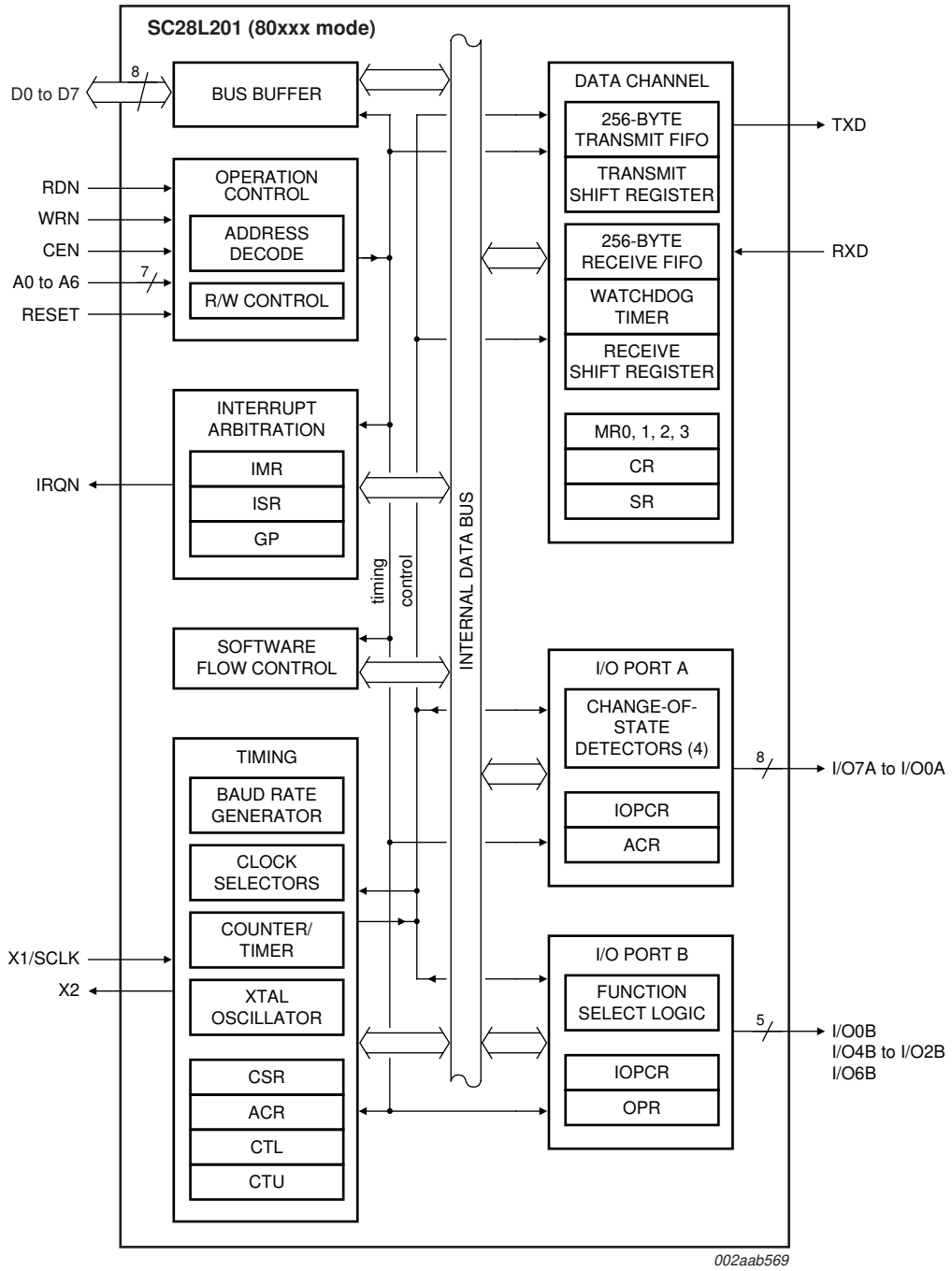

Fig 1. Block diagram of SC28L201 (80xxx mode)

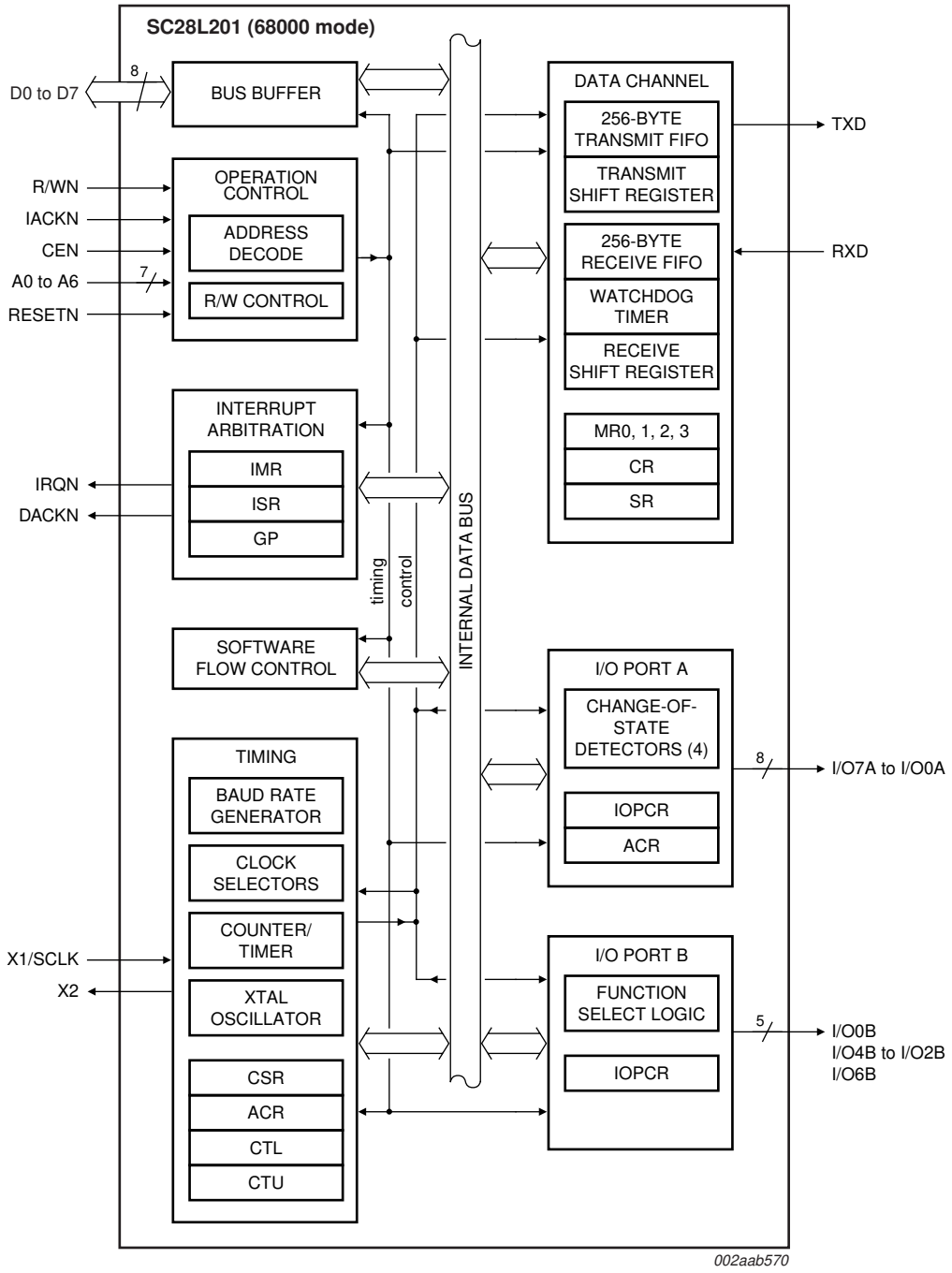

Fig 2. Block diagram of SC28L201 (68000 mode)

5. Pinning information

5.1 Pinning

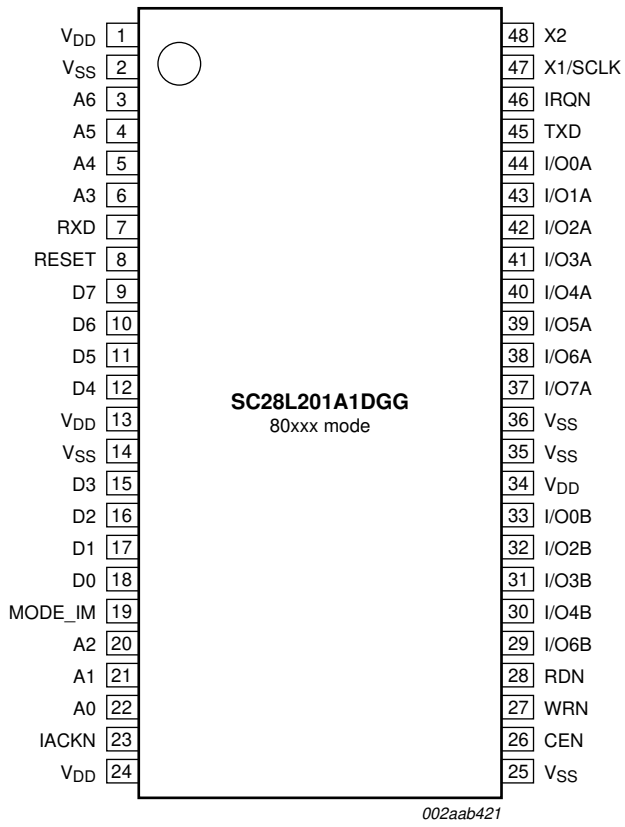

Fig 3. Pin configuration for TSSOP48; 80xxx mode

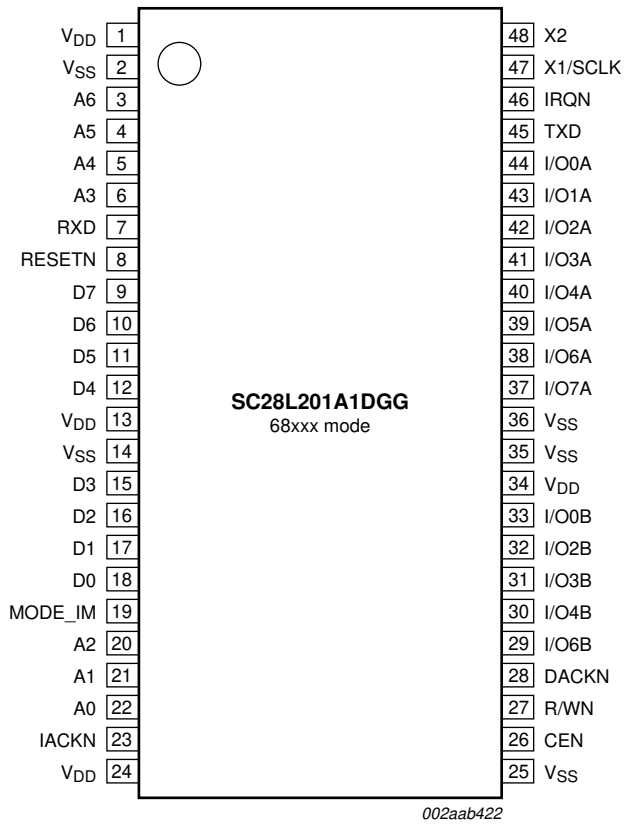

Fig 4. Pin configuration for TSSOP48; 68000 mode

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 ½ 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 ½ 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; ½-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