

## THE SAM COMMUNICATIONS INTERFACE

When connected to the SAM Coupé computer, this interface allows you to attach parallel (centronics) or serial (RS232) external devices. The communications interface should allow any commercially-available printer to work with the Coupé, but note that other communications devices - modems, for example - are unlikely to work unless they have been designed specifically to be compatible with the Coupé.

### Connection

With the power to the computer off, connect the communications interface to the expansion port at the back of the Coupé. The serial number on the interface should be face downwards.

There are two sockets at the back of the communications interface. The larger 26-pin socket is for parallel/centronics connections, and a standard BBC-type printer cable should be used to connect the external device to the interface. The smaller 9-pin D-type socket is for serial/RS232 connections; it's IBM PC-AT compatible as shown in the connection diagram on page 4. Cables are usually provided with the external devices you purchase. But in case of any difficulty, please contact MGT on 0792 - 791100.

Both the parallel and the serial sockets can be in use at the same time. Note though, that if you wish to connect another external interface to the expansion connector - an external drive interface, for example - you'll need the SAM Card Cage, due for release later in 1990.

### Using a Parallel Printer

If your printer is Epson FX80 compatible - and most of them are these days - then you probably won't have to do anything. Just connect your printer to the parallel port (with all power off), and go. Follow the instructions in the main Coupé manual to LLIST or LPRINT - and try the spectacular screen-dumps using FLASH!

(If you're contemplating buying a printer, make sure you get one which is Epson FX80 compatible. Ask your dealer for guidance: most manufacturers make budget-priced printers to this standard.)

You may find that the printer wants to output everything on the same line, or that it jumps a line between each one that it prints. This is because printer manufacturers sometimes set automatic line-feed to ON, sometimes to OFF at the factory. You can correct the problem either by changing the dip-switch settings on your printer as instructed in the printer manual, or by changing one of the Coupé's system variables.

The key system variables are listed at the back of the main Coupé manual. You'll see that system variable (SVAR) 15 is concerned with printer automatic line feeds. If your printer is overprinting (in other words, not sending an automatic line feed), type in the instruction:

```
POKE SVAR 15,10          (RETURN)
```

If, on the other hand, your printer is sending an extra line, type:

```
POKE SVAR 15,0          (RETURN)
```

These settings will remain in operation until you reset or turn off the computer, or until you send the instruction NEW.

### Using the Serial/RS232 Port

Connect your external device to the communications interface with all power off.

Turn the power on, install the DOS (if you're using a disk drive) and type:

```
DEVICE T                (RETURN)
```

Press (F7) and then PLAY on the cassette recorder with the Communications Software cassette loaded. This will load the Basic loader and autorun the program. You'll see a number of settings displayed on screen with the option to change the settings. A little bit of background here....

Data can be transmitted between RS232/serial devices and the computer at varying rates, known as the baud rate. The documentation for the external device you are using should state the baud rate that needs to be set, and may also define criteria for stop bits and data bits. (Don't worry about definitions here; all that's important is that you set up the computer to receive and transmit data at the appropriate rates.)

When you first load in the program, you'll see that the baud rate is set to 9600 baud, stop bits are set at 1 bit, and data bits are set at 8 bits. If these are the settings that your external device requires, select option 1 - "Initialise using current data". But if you need to change the settings, select option 2 - "Edit serial parameters". You'll then be able to modify the settings for baud rate, stop bits and data bits.

When you've made the modifications, you then need to follow up by selecting option 1 - "Initialise using current driver" - which will re-initialise the driver.

Option 3 - "Back up comms software" - allows you to save a copy of the selections you have made. This will be particularly useful if you have two communications devices, each requiring different settings. By making a back-up, you will be able to load the driver which is suitable for the device straight away, instead of modifying the settings each time you use the device.

If you back up to disk, you are asked to enter a file-name, which will become the file-name in the disk directory. The filename you enter should only be 5 characters long - and the computer will automatically add a file-name extension - .BAS. The files saved are:

```
CSETUP.BAS      - Configuration program
CCODE.CDE      - Machine code driver
(file-name).BAS - Autorun file with name
```

In Basic instructions, the serial port is referred to as "b". Thus, when the driver is installed, you can input and output data to the serial port by:

```
LLIST          List a program
LPRINT a$      Print a$

OPEN #7;"b"    Open stream 7 and output to serial
PRINT #7;a$    port

OPEN #7;"b"    Open stream 7 and input from stream 7
INPUT #7;a$
LET a$ = INKEY$ #7
```

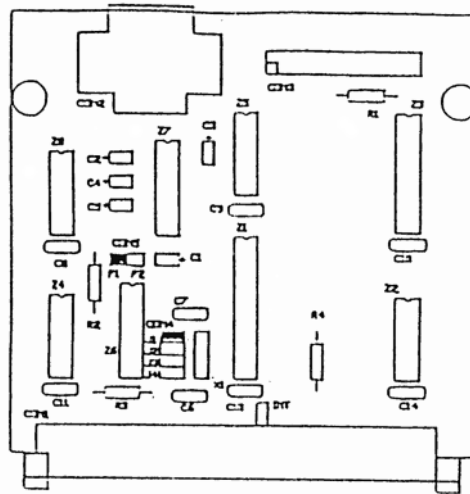
#### **NOTES:**

1) Using the communications interface, you are not permitted to output to devices simultaneously; nor can you input to two devices simultaneously. However, it is permitted to input data from one device and to output it to another simultaneously. For example, you could input from a modem and output to a parallel printer using the instructions:

```
5 GOSUB 100
10 CLOSE #6
20 OPEN #6;"b"
30 LET a$=INKEY$ #6
40 IF a$=" " THEN GOTO 30
50 LPRINT a$
60 GOTO 30

100 REM Set output to parallel
110 LET BCHAN=DPEEK SVAR 591+25
120 DPOKE BCHAN,&0181
130 RETURN
```

2) The comms driver is installed in the highest page of memory available. If the DOS is installed, it will be placed in the page below the DOS.



### JUMPERS SETTING

#### CON 4 JUMPER

<u>POSITION</u>	<u>FUNCTION</u>
TOP MOST (DEFAULT) (I/O PORT &EC)	MAP SERIAL PORT AS COM1
2ND FROM TOP (I/O PORT &ED)	MAP SERIAL PORT AS COM2
2ND FROM BOTTOM (I/O PORT &EE)	MAP SERIAL PORT AS COM3
BOTTOM MOST (I/O PORT &EF)	MAP SERIAL PORT AS COM4
REMOVED	DISABLE SERIAL PORT

#### CON 5 JUMPER

<u>POSITION</u>	<u>FUNCTION</u>
LEFT (DEFAULT) (I/O PORTS &E8, &E9)	MAP PARALLEL PORT AS LPT1
RIGHT (I/O PORTS &EA, &EB)	MAP PARALLEL PORT AS LPT2
REMOVED LPT1 AND LPT2	MAP PARALLEL PORT AS BOTH

#### CON 6 JUMPER

[next to C13]

<u>POSITION</u>	<u>FUNCTION</u>
OPEN	Serial port interrupt feature disabled.
CLOSE (DEFAULT)	Serial port can be configured to interrupt the Z80 CPU.

**Comms Interface PC Mouse modification instructions:**

- Open comms interface by unscrewing the two screws at the back
- remove PCB and turn it upside-down. The serial connector is located at the top-right side, just below the text 'MILES GORDON TECHNOLOGY'
- Solder a short wire (about 3 cm) to connect pins 4 and 7 of the serial connector (see Figure 1).
- Check your soldering for short-circuits and reassemble the comms interface.

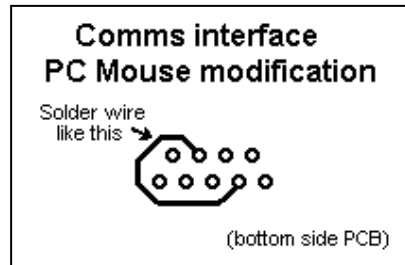
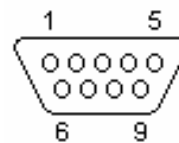
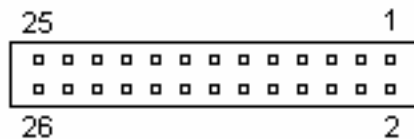
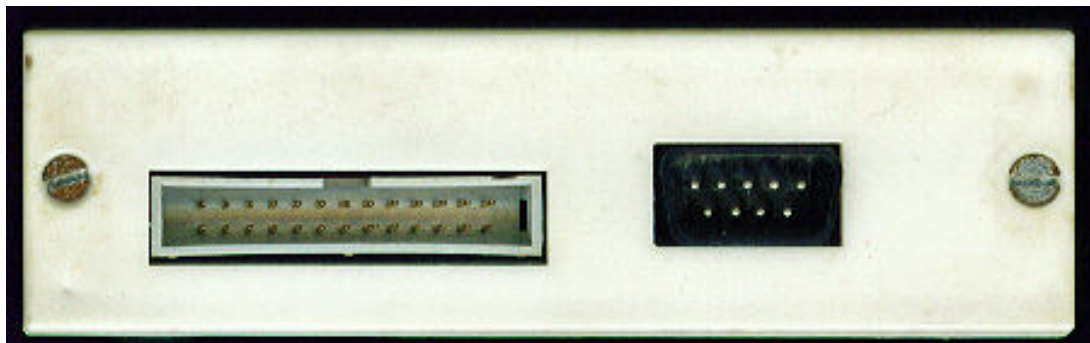


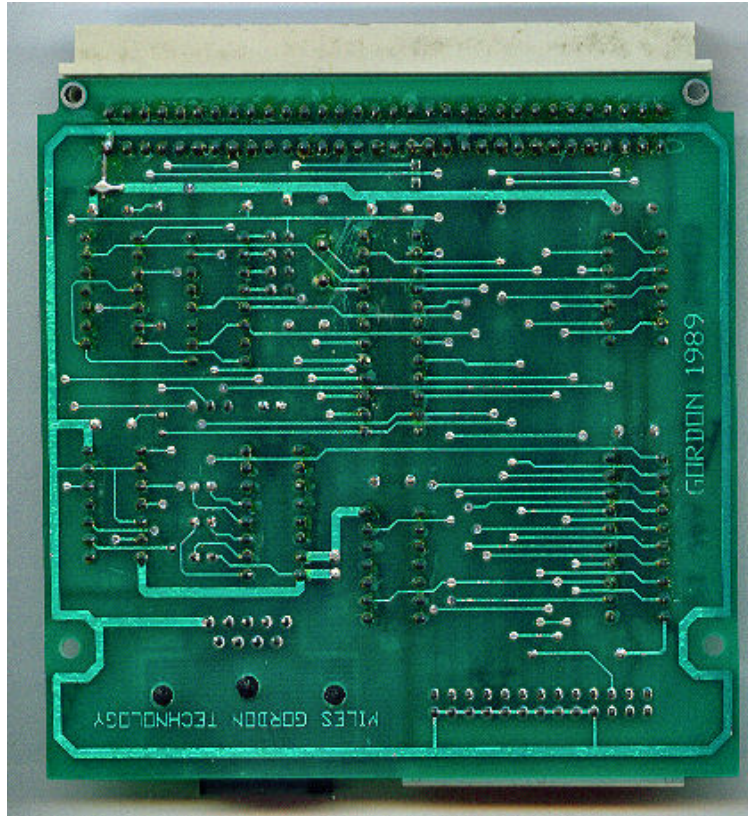
Figure 1

**Pin Outs**

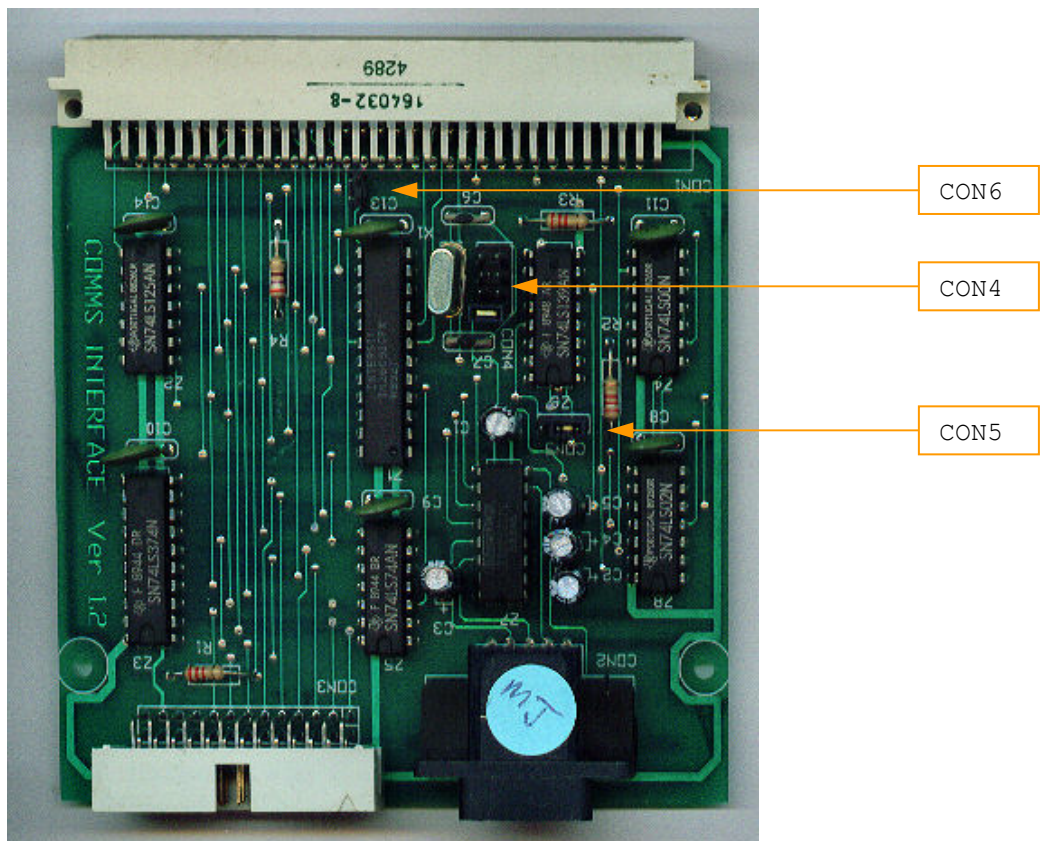


PIN	SIGNAL	TYPE	PIN	SIGNAL	TYPE
1	PRINTER STROBE	OUT	2	RXD	RECIEVE DATA
3	DATA BIT 0	OUT	3	TXD	TRANSMIT DATA
5	DATA BIT 1	OUT	4	DTR	DATA TERM. READY
7	DATA BIT 2	OUT	5	GND	GROUND
9	DATA BIT 3	OUT	8	CTS	CLEAR TO SEND
11	DATA BIT 4	OUT			
13	DATA BIT 5	OUT			
15	DATA BIT 6	OUT			
17	DATA BIT 7	OUT			
21	BUSY	IN			
2-20,23	GROUND PIN NUMBERS	0 V			
19,22,24,25,26	NOT CONNECTED				

Circuit Board Scans



Sam Coupé Comms Interface.



Sam Coupé Communications Interface



```
Serial Communications Driver

RS232 Parameters

Port Addr      236
Baud Rate     9600   Baud
Stop Bits     1      Bit(s)
Data Bits     8      Bit(s)

Driver Options

1. Initialise using current data
2. Edit serial parameters
3. Backup Comms Software
4. Exit to BASIC

Enter Option Number █
```

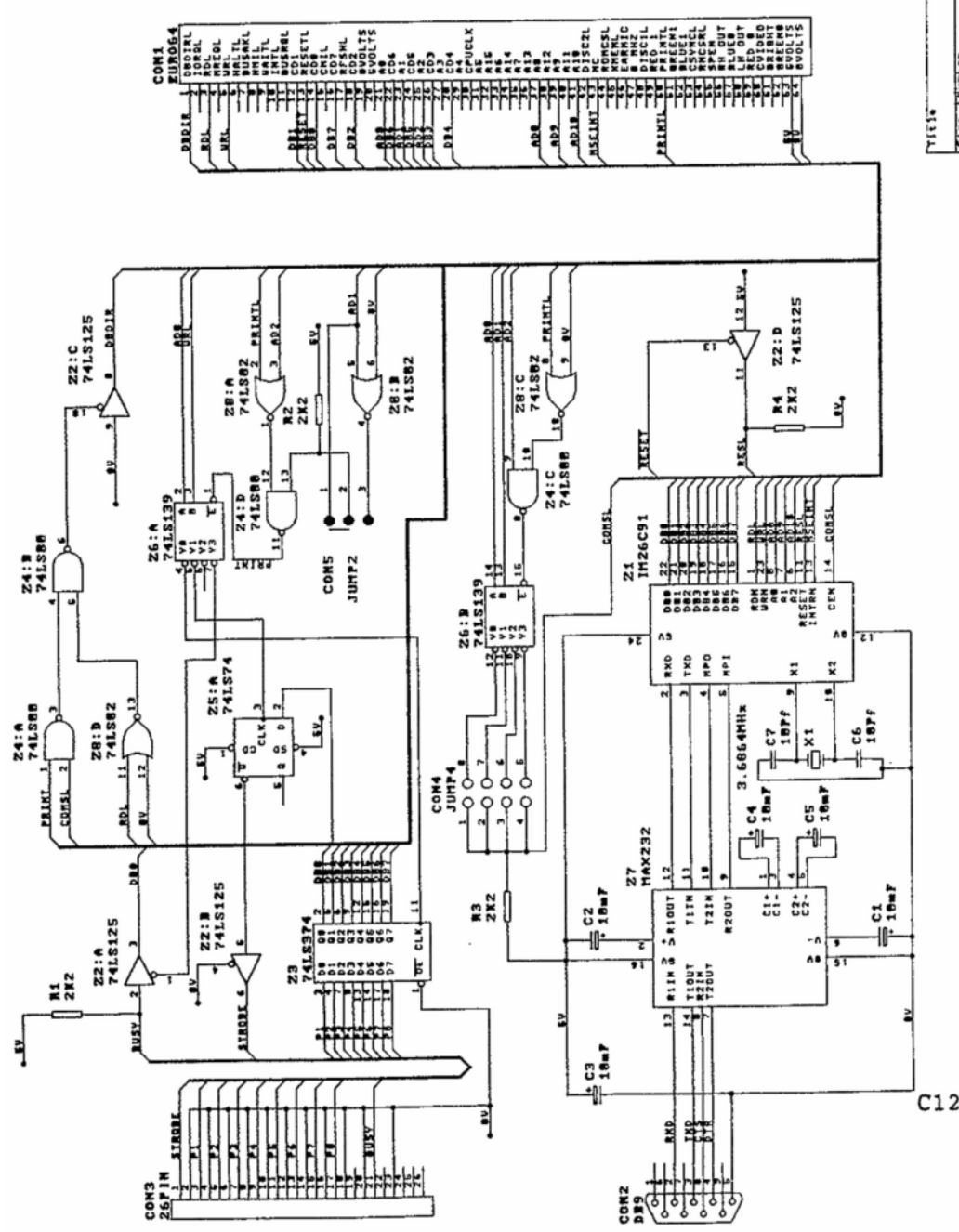
The initialise program in operation.  
[Running on Sim Coupe]



Port address	Decimal	Hexadecimal
	236	&EC
	237	&ED
	238	&EE
	239	&EF



# Comms Interface Schematic



T1C16

Size	Number	Revision
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30
31	31	31
32	32	32
33	33	33
34	34	34
35	35	35
36	36	36
37	37	37
38	38	38
39	39	39
40	40	40
41	41	41
42	42	42
43	43	43
44	44	44
45	45	45
46	46	46
47	47	47
48	48	48
49	49	49
50	50	50
51	51	51
52	52	52
53	53	53
54	54	54
55	55	55
56	56	56
57	57	57
58	58	58
59	59	59
60	60	60
61	61	61
62	62	62
63	63	63
64	64	64
65	65	65
66	66	66
67	67	67
68	68	68
69	69	69
70	70	70
71	71	71
72	72	72
73	73	73
74	74	74
75	75	75
76	76	76
77	77	77
78	78	78
79	79	79
80	80	80
81	81	81
82	82	82
83	83	83
84	84	84
85	85	85
86	86	86
87	87	87
88	88	88
89	89	89
90	90	90
91	91	91
92	92	92
93	93	93
94	94	94
95	95	95
96	96	96
97	97	97
98	98	98
99	99	99
100	100	100

This pdf was compiled by Steve Parry-Thomas 21 Jan 2005

For Sam Coupe Users everywhere

Thanks to the following for there help with this one.

Edwin Blink, Simon Owen & Colin Piggot.

# DATA SHEET

## **SCC2691**

Universal asynchronous  
receiver/transmitter (UART)

Product specification  
Supersedes data of 1995 May 01  
IC19 Data Handbook

1998 Sep 04

# Universal asynchronous receiver/transmitter (UART)

# SCC2691

## DESCRIPTION

The Philips Semiconductors SCC2691 Universal Asynchronous Receiver/Transmitter (UART) is a single-chip CMOS-LSI communications device that provides a full-duplex asynchronous receiver/transmitter. It is fabricated with Philips Semiconductors CMOS technology which combines the benefits of high density and low power consumption.

The operating speed of the receiver and transmitter can be selected independently as one of 18 fixed baud rates, a 16X clock derived from a programmable counter/timer, or an external 1X or 16X clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the UART particularly attractive for dual-speed channel applications such as clustered terminal systems.

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

The UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes.

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

## FEATURES

- Full-duplex asynchronous receiver/transmitter
- Quadruple buffered receiver data register
- Programmable data format:
  - 5 to 8 data bits plus parity
  - Odd, even, no parity or force parity
  - 1, 1.5 or 2 stop bits programmable in 1/16-bit increments
- 16-bit programmable Counter/Timer
- Baud rate for the receiver and transmitter selectable from:
  - 22 fixed rates: 50 to 115.2K baud
  - Non-standard rates to 115.2kb
  - Non-standard user-defined rate derived from programmable timer/ counter
  - External 1X or 16X clock
- Parity, framing, and overrun detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - Normal (full-duplex)
  - Automatic echo
  - Local loopback
  - Remote Loopback
- Multi-function programmable 16-bit counter/timer

## PIN CONFIGURATIONS

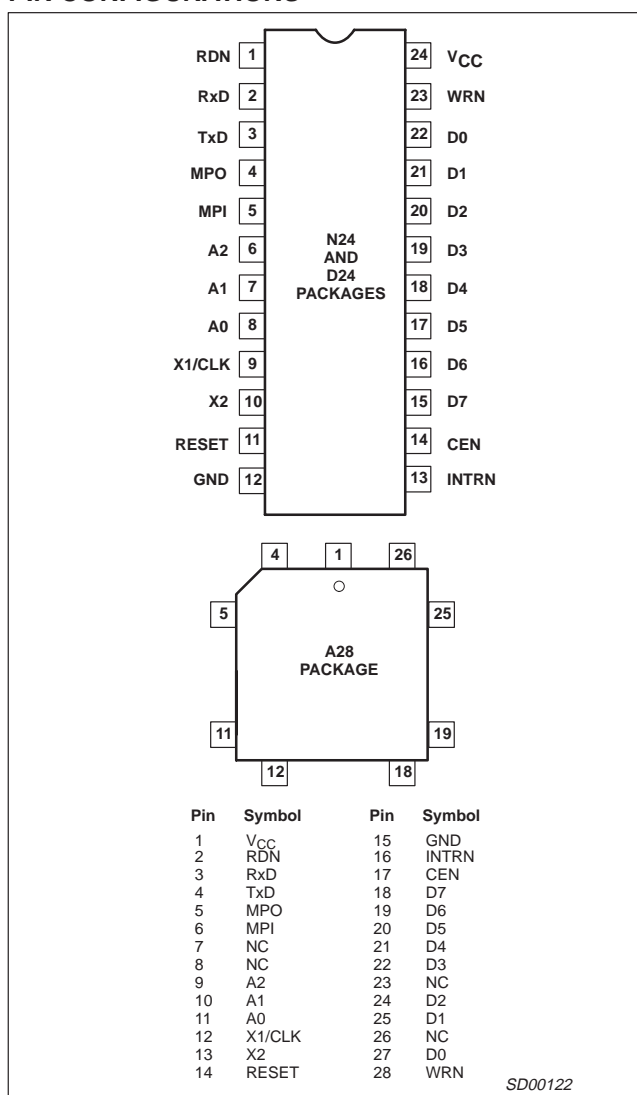


Figure 1. Pin Configurations

- Single interrupt output with seven maskable interrupting conditions
- On-chip crystal oscillator
- Low power mode
- TTL compatible
- Single +5V power supply
- Commercial (0°C to +70°C) and industrial (-40°C to +85°C) temperature versions available
- SOL, PLCC and 300 mil wide DIP packages available

# Universal asynchronous receiver/transmitter (UART)

SCC2691

## ORDERING INFORMATION

PACKAGES	COMMERCIAL	INDUSTRIAL	DWG #
	V <sub>CC</sub> = +5V ±10%, T <sub>A</sub> = 0°C to +70°C	V <sub>CC</sub> = +5V ±10%, T <sub>A</sub> = -40°C to +85°C	
24-Pin Plastic Dual In-Line Package (DIP)	SCC2691AC1N24	SCC2691AE1N24	SOT222-1
28-Pin Plastic Leaded Chip Carrier (PLCC) Package	SCC2691AC1A28	SCC2691AE1A28	SOT261-3
24-Pin Plastic Small Outline Large (SOL) Package	SCC2691AC1D24		SOT137-1

## BLOCK DIAGRAM

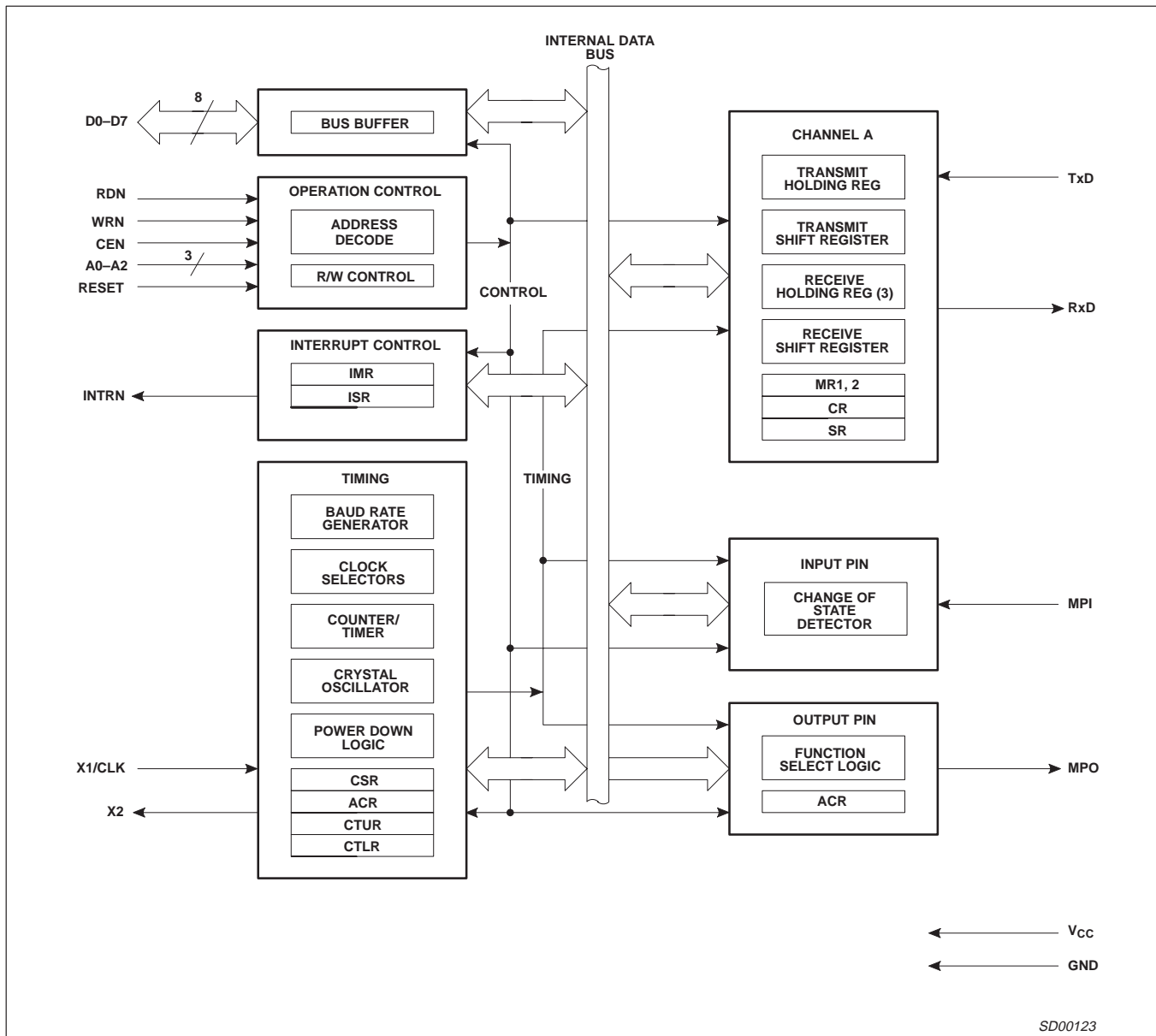


Figure 2. Block Diagram

## Universal asynchronous receiver/transmitter (UART)

SCC2691

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0–D7	22–15	27, 25, 24, 22–18	I	<b>Data Bus:</b> Active-high 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is low. When the CEN input is high, the data bus is in the 3-State condition.
CEN	14	17	I	<b>Chip Enable:</b> Active-low input. When low, data transfers between the CPU and the UART are enabled on D0–D7 as controlled by the WRN, RDN and A0–A2 inputs. When CEN is high, the UART is effectively isolated from the data bus and D0–D7 are placed in the 3-State condition.
WRN	23	28	I	<b>Write Strobe:</b> Active-low input. A low on this pin while CEN is low causes the contents of the data bus to be transferred to the register selected by A0–A2. The transfer occurs on the trailing (rising) edge of the signal.
RDN	1	2	I	<b>Read Strobe:</b> Active-low input. A low on this pin while CEN is low causes the contents of the register selected by A0–A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0–A2	8–6	11–9	I	<b>Address Inputs:</b> Active-high address inputs to select the UART registers for read/write operations.
RESET	11	14	I	<b>Reset:</b> Master reset. A high on this pin clears the status register (SR), the interrupt mask register (IMR), and the interrupt status register (ISR), sets the mode register pointer to MR1, and places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (high) state. Clears Test modes.
INTRN	13	16	O	<b>Interrupt Request:</b> This active-low output is asserted upon occurrence of one or more of seven maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s). This open-drain output requires a pull-up resistor.
X1/CLK	9	12	I	<b>Crystal 1:</b> Crystal connection or an external clock input. A crystal of a clock the appropriate frequency (nominally 3.6864 MHz) must be supplied at all times. For crystal connections see Figure 7, Clock Timing.
X2	10	13	I	<b>Crystal 2:</b> Crystal connection. See Figure 7. If a crystal is not used it is best to keep this pin not connected although it is permissible to ground it.
RxD	2	3	I	<b>Receiver Serial Data Input:</b> The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock.
TxD	3	4	O	<b>Transmitter Serial Data Output:</b> The least significant bit is transmitted first. This output is held in the marking (high) condition when the transmitter is idle or disabled and when the UART is operating in local loopback mode. If external transmitter is specified, the data is shifted on the falling edge of the transmitter clock.
MPO	4	5	O	<b>Multi-Purpose Output:</b> One of the following functions can be selected for this output pin by programming the auxiliary control register: <b>RTSN</b> – Request to send active-low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. <b>C/TO</b> – The counter/timer output. <b>TxC1X</b> – The 1X clock for the transmitter. <b>TxC16X</b> – The 16X clock for the transmitter. <b>RxC1X</b> – The 1X clock for the receiver. <b>RxC16X</b> – The 16X clock for the receiver. <b>TxRDY</b> – The transmitter holding register empty signal. Active-low output. (Open drain) <b>RxRDY/FFULL</b> – The receiver FIFO not empty/full signal. Active-low output. (Open drain)
MPI	5	6	I	<b>Multi-Purpose Input:</b> This pin can serve as an input for one of the following functions: <b>GPI</b> – General purpose input. The current state of the pin can be determined by reading the ISR. <b>CTSN</b> – Clear-to-send active-low input. <b>CTCLK</b> – Counter/timer external clock input. <b>RTCLK</b> – Receiver and/or transmitter external clock input. This may be a 1X or 16X clock as programmed by CSR[3:0] or CSR[7:4]. Pin has an internal V <sub>CC</sub> pull-up device supplying 1 to 4 μA of current.
V <sub>CC</sub>	24	1	I	<b>Power Supply:</b> +5V supply input.
GND	12	15	I	<b>Ground</b>

## Universal asynchronous receiver/transmitter (UART)

SCC2691

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING	UNIT
$T_A$	Operating ambient temperature range <sup>2</sup>	Note 4	°C
$T_{STG}$	Storage temperature range	-65 to +150	°C
$V_{CC}$	Voltage from $V_{CC}$ to GND <sup>3</sup>	-0.5 to +7.0	V
$V_S$	Voltage from any pin to ground <sup>3</sup>	-0.5 to $V_{CC} \pm 10\%$	V
$P_D$	Power Dissipation	300	mW

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
- For operating at elevated temperature, the device must be derated based on +150°C maximum junction temperature.
- 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.
- Parameters are valid over specified temperature range. See Ordering Information table for applicable operating temperature and  $V_{CC}$  supply range.

DC ELECTRICAL CHARACTERISTICS<sup>1, 2, 3</sup>

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
$V_{IL}$ $V_{IH}$	Input low voltage Input high voltage  All except X1/CLK X1/CLK				0.8	V V V
			2 $0.8V_{CC}$		$V_{CC}$	
$V_{OL}$ $V_{OH}$ <sup>4</sup>	Output low voltage Output high voltage (except open drain outputs)	$I_{OL} = 2.4\text{mA}$  $I_{OH} = -400\mu\text{A}$			0.4	V V
			2.4			
$I_{IL}$	Input leakage current	$V_{IN} = 0$ to $V_{CC}$	-10		10	$\mu\text{A}$
$I_{LL}$	Data bus 3-State leakage current	$V_O = 0.4$ to $V_{CC}$	-10		10	$\mu\text{A}$
$I_{OD}$	Open-drain output leakage current	$V_O = 0.4$ to $V_{CC}$	-10		10	$\mu\text{A}$
$I_{XIL}$	X1/CLK low input current	$V_{IN} = 0$ , X2 floated	-100	-30	0	$\mu\text{A}$
$I_{XIH}$	X1/CLK high input current	$V_{IN} = V_{CC}$ , X2 floated	0	30	100	$\mu\text{A}$
$I_{X2L}$	X2 low output current	$V_{OUT} = 0$ , X1/CLK = $V_{CC}$	-100			$\mu\text{A}$
$I_{X2H}$	X2 high output current	$V_{OUT} = V_{CC}$ , X1/CLK = 0V			100	$\mu\text{A}$
$I_{CCA}$	Power supply current, active	0°C to +70°C		0.8	2.0	mA
$I_{CCD}$	Power down current <sup>5</sup>	-40°C to +85°C		1.0	2.5 500	mA $\mu\text{A}$

## NOTES:

- Parameters are valid over specified temperature range. See Ordering Information table for applicable operating temperature and  $V_{CC}$  supply range.
- All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0V and 3.0V with a transition time of 20ns max. For X1/CLK, this swing is between 0.4V and 4.0V. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages of 0.8V and 2V as appropriate.
- Typical values are at +25°C, typical supply voltages, and typical processing parameters.
- Test condition for outputs:  $C_L = 150\text{pF}$ , except interrupt outputs. Test conditions for interrupt outputs:  $C_L = 50\text{pF}$ ,  $R_L = 2.7\text{k}\Omega$  to  $V_{CC}$ .
- For power down current levels in the 1 $\mu\text{A}$  region see the UART application note.

## Universal asynchronous receiver/transmitter (UART)

SCC2691

AC ELECTRICAL CHARACTERISTICS<sup>1, 2, 3, 4</sup>

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Typ	Max	
<b>Reset timing (Figure 3)</b>					
t <sub>RES</sub>	Reset pulse width	100			ns
<b>Bus timing (Figure 4)<sup>5</sup></b>					
t <sub>AS</sub>	A0–A2 setup time to RDN, WRN low	10			ns
t <sub>AH</sub>	A0–A2 hold time from RDN, WRN low	100			ns
t <sub>CS</sub>	CEN setup time to RDN, WRN low	0			ns
t <sub>CH</sub>	CEN hold time from RDN, WRN high	0			ns
t <sub>RW</sub>	WRN, RDN pulse width	150			ns
t <sub>DD</sub>	Data valid after RDN low			125	ns
t <sub>DF</sub>	Data bus floating after RDN high			110	ns
t <sub>DS</sub>	Data setup time before WRN high	50			ns
t <sub>DH</sub>	Data hold time after WRN high	30			ns
t <sub>RWD</sub>	Time between reads and/or writes <sup>6, 7</sup>	150			ns
<b>MPI and MPO timing (Figure 5)<sup>5</sup></b>					
t <sub>PS</sub>	MPI input setup time before RDN low	30			ns
t <sub>PH</sub>	MI input hold time after RDN low	30			ns
t <sub>PD</sub>	MPO output valid after WRN high			370	ns
<b>Interrupt timing (Figure 6)</b>					
t <sub>IR</sub>	INTRN negated				
	Read RHR (RxRDY/FFULL interrupt)			370	ns
	Write THR (TxRDY, TxEMT interrupt)			370	ns
	Reset command (break change interrupt)			370	ns
	Reset command (MPI change interrupt)			370	ns
	Stop C/T command (counter interrupt)			370	ns
	Write IMR (clear of interrupt mask bit)			270	ns
<b>Clock timing (Figure 7)</b>					
t <sub>CLK</sub>	X1/CLK high or low time	100			ns
f <sub>CLK</sub> <sup>9</sup>	X1/CLK frequency	0		4.0	MHz
t <sub>CTC</sub>	Counter/timer clock high or low time	100			ns
f <sub>CTC</sub> <sup>8</sup>	Counter/timer clock frequency	0		4.0	MHz
t <sub>RX</sub>	RxC high or low time	220			ns
f <sub>RX</sub> <sup>8</sup>	RxC frequency (16X)	0	3.6864	2.0	MHz
	RxC frequency (1X)	0		1.0	MHz
t <sub>TX</sub>	TxC high or low time	220			ns
f <sub>TX</sub> <sup>8</sup>	TxC frequency (16X)	0		2.0	MHz
	TxC frequency (1X)	0		1.0	MHz
<b>Transmitter timing (Figure 8)</b>					
t <sub>TXD</sub>	TxD output delay from TxC external clock input on IP pin			350	ns
t <sub>TCS</sub>	Output delay from TxC low at OP pin to TxD data output	0		150	ns
<b>Receiver timing (Figure 9)</b>					
t <sub>RXS</sub>	RxD data setup time before RxC high at external clock input on IP pin	100			ns
t <sub>RXH</sub>	RxD data hold time after RxC high at external clock input on IP pin	100			ns

## NOTES:

- Parameters are valid over specified temp. range. See Ordering Information table for applicable operating temp. and V<sub>CC</sub> supply range.
- All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0V and 3.0V with a transition time of 20ns max. For X1/CLK, this swing is between 0.4V and 4.0V. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages of 0.8V and 2V as appropriate.
- Typical values are at +25°C, typical supply voltages, and typical processing parameters.
- Test condition for outputs: C<sub>L</sub> = 150pF, except interrupt outputs. Test conditions for interrupt outputs: C<sub>L</sub> = 50pF, R<sub>L</sub> = 2.7kΩ to V<sub>CC</sub>.
- Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are ORed internally. As a consequence, this signal asserted last initiates the cycle and the signal negated first terminates the cycle.
- If CEN is used as the 'strobing' input, this parameter defines the minimum high time between one CEN and the next. The RDN signal must be negated for t<sub>RWD</sub> guarantee that any status register changes are valid.
- Consecutive write operations to the command register require at least three rising edges of the X1 clock between writes.
- These parameters are guaranteed by design, but are not 100% tested in production.
- Operation to 0MHz is assured by design. Minimum test frequency is 2MHz.



# Universal asynchronous receiver/transmitter (UART)

# SCC2691

## BLOCK DIAGRAM

As shown in the block diagram, the UART consists of: data bus buffer, interrupt control, operation control, timing, receiver and transmitter.

### Data Bus Buffer

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

### Interrupt Control

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

- Transmit holding register ready
- Transmit shift register empty
- Receive holding register ready or FIFO full
- Change in break received status
- Counter reached terminal count
- Change in MPI input
- Assertion of MPI input

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

### Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

**Table 1. Register Addressing**

A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	MR1, MR2	MR1, MR2
0	0	1	SR	CSR
0	1	0	BRG Test	CR
0	1	1	RHR	THR
1	0	0	1X/16X Test	ACR
1	0	1	ISR	IMR
1	1	0	CTU	CTUR
1	1	1	CTL	CTLR

**NOTE;**

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

ACR = Auxiliary control register

CR = Command register

CSR = Clock select register

CTL = Counter/timer lower output register

CTLR = Counter/timer lower preset register

CTU = Counter/timer upper output register

CTUR = Counter/timer upper preset register

MR = Mode register A

SR = Status register

THR = Tx holding register

\* See Table 6 for BRG Test frequencies in this data sheet, and "Extended baud rates for SCN2681, SCN68681, SCC2691,

SCC2692, SCC68681 and SCC2698B" Philips Semiconductors ICs for Data Communications, IC-19, 1994.

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

### Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and two clock selectors.

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external clock is used instead of a crystal, X1/CLK is driven using a configuration similar to the one in Figure 7. In this case, the input high-voltage must be capable of attaining the voltage specified in the DC Electrical Characteristics. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied if the internal BRG is not used.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection by the receiver and transmitter of any of these baud rates or an external timing signal.

### Counter/Timer (C/T)

The C/T operation is programmed by ACR[6:4]. One of eight timing sources can be used as the input to the C/T. The output of the C/T is available to the clock selectors and can be programmed by ACR[2:0] to be output on the MPO pin.

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

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start C/T command. Upon reaching terminal count, the counter ready bit in the ISR is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and the counter ready bit is cleared when the counter is stopped by a stop counter command. the CPU may change the

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command following a stop counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower eight bits of the counter may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower eight bits to the upper eight bits occurs between the times that both halves of the counter are read. However, a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR. See further description in CTUR/CTLR section.

### Receiver and Transmitter

The UART is a full-duplex asynchronous receiver/transmitter. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter/timer, or from an external input. Registers associated with the communications channel are: the mode registers (MR1 and MR2), the clock select register (CSR), the command register (CR), the status register (SR), the transmit holding register (THR), and the receive holding register (RHR).

#### Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the SR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character in the THR. In the 16X clock mode, this also resynchronizes the internal 1X transmitter clock so that transmission of the new character begins with minimum delay.

The transmitter can be forced to send a break (continuous low condition) by issuing a start break command via the CR. The break is terminated by a stop break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted and the character in the THR, if any, are completely sent out. Characters cannot be loaded in the THR while the transmitter is disabled.

#### Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The data is then transferred to the RHR and the RxRDY bit in the SR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for

one-half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error and overrun error (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set.

If a break condition is detected (RxD is low for the entire character including the stop bit), only one character consisting of all zeros will be loaded in the FIFO and the received SR break bit is set to 1. The RxD input must return to high for two (2) clock edges of the X1 crystal clock for the receiver to recognize the end of the break condition and begin the search for a start bit. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.**

### RECEIVER FIFO

The RHR consists of a first-in-first-out (FIFO) queue with a capacity of three characters. Data is loaded from the receive shift register into the top-most empty position of the FIFO. The RxRDY bit in the status register (SR) is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three queue positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits are 'popped' thus emptying a FIFO position for new data.

### Receiver Status Bits

In addition to the data word, three status bits (parity error, framing error, and received break) are appended to each data character in the FIFO. Status can be provided in two ways, as programmed by the error mode control bit in mode register 1. In the character mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last reset error command was issued. In either mode, reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore, the SR should be read prior to reading the corresponding data character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

### Receiver Reset and Disable

Receiver disable stops the receiver immediately – data being assembled if 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. A 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 has the appearance of "clearing or flushing" the receiver FIFO. In fact, the FIFO is NEVER cleared! The data in the FIFO remains valid until overwritten by another received character. Because of this, erroneous reading or extra reads of the receiver FIFO will miss-align the FIFO pointers and result in the reading of previously read data. A receiver reset will re-align the pointers.

# Universal asynchronous receiver/transmitter (UART)

# SCC2691

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

In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, whose receivers are normally disabled, examine the received data stream and 'wake-up' the CPU [by setting RxRDY] only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

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

While in this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character in the RHR FIFO if the received A/D bit is a one, but discards the received character if the received A/D bit is a zero. If enabled, all received characters are then transferred to the CPU via the RHR. In either case, the data bits are loaded in the data FIFO while the A/D bit is loaded in the status FIFO position normally used for parity error (SR[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

## MULTI-PURPOSE INPUT PIN

The MPI pin can be programmed as an input to one of several UART circuits. The function of the pin is selected by programming the appropriate control register (MR2[4]), ACR[6:4], CSR [7:4, 3:0]). Only one of the functions may be selected at any given time. If CTS or GPI is selected, a change of state detector provided with the pin is activated. A high-to-low or low-to-high transition of the inputs lasting longer than 25–50µs sets the MPI change-of-state bit in the interrupt status register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU by setting the corresponding bit in the interrupt mask register.

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This produces a sampling period of slightly more than 25µs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25µs if the transition occurs coincident with the first sample pulse. The 50µs time refers to the condition where the change of state is just missed and the first change of state is not detected until after an additional 25µs. The MPI pin has a small pull-up device that will source 1 to 4 µA of current from V<sub>CC</sub>. This pin does not require pull-up devices or V<sub>CC</sub> connection if it is not used.

## MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1X or 16X transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see ACR[2:0] – MPO Output Select). Please note that this pin drives both high and low. HOWEVER when it is programmed to represent interrupt type functions (such as receiver ready, transmitter ready or counter/timer ready) it will be switched to an open drain configuration in which case an external pull-up device would be required.

## REGISTERS

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

The contents of certain control registers are initialized to zero on reset (see RESET pin description). Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. The contents of the MR, the CSR, and the ACR should only be changed while the receiver and transmitter are disabled, and certain changes to the ACR should only be made while the C/T is stopped. The bit formats of the UART are shown in Table 2.

### MR1 – Mode Register 1

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

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

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

#### MR1[6] – Receiver Interrupt Select

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

#### MR1[5] – Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). In the character mode, status is provided on a character-by-character basis. The status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last reset error command was issued.

#### MR1[4:3] – Parity Mode Select

If with parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

#### MR1[2] – Parity Type Select

This bit selects the parity type (odd or even) if the with parity mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the force parity mode is programmed. It has no effect if the no

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

parity mode is programmed. In the special wake-up mode, it selects the polarity of the transmitted A/D bit.

### MR1[1:0] – Bits Per Character Select

This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

### MR2 – Mode Register 2

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

#### MR2[7:6] – Mode Select

The UART can operate in one of four modes. MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[7:6] = 01 places the channel in the automatic echo mode, which automatically re-transmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is re-clocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU-to-receiver communication continues normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2[7:6] = 11. In this mode:

1. Received data is re-clocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

When switching in and out of the various modes, the selected mode is activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode

immediately. An exception to this is switching out of auto-echo or remote loopback modes; if the deselection occurs just after the receiver has sampled the stop bit (indicated in auto-echo by assertion of RxRDY), and the transmitter is enabled, the transmitter is enabled, the transmitter will remain in auto-echo mode until one full stop bit has been retransmitted.

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

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

This bit allows deactivation of the RTSN output by the transmitter. This output is manually asserted and negated by the appropriate commands issued via the command register. MR2[5] set to 1 caused the RTSN to be reset automatically one bit time after the character(s) in the transmit shift register and in the THR (if any) are completely transmitted (including the programmed number of stop bits) if a previously issued transmitter disable is pending. This feature can be used to automatically terminate the transmission as follows:

1. Program the auto-reset mode: MR2[5]=1
2. Enable transmitter, if not already enabled
3. Assert RTSN via command
4. Send message
5. After the last character of the message is loaded to the THR, disable the transmitter. (If the transmitter is underrun, a special case exists. See note below.)
6. The last character will be transmitted and the RTSN will be reset one bit time after the last stop bit is sent.

NOTE: The transmitter is in an underrun condition when both the TxRDY and the TxEMT bits are set. This condition also exists immediately after the transmitter is enabled from the disabled or reset state. When using the above procedure with the transmitter in the underrun condition, the issuing of the transmitter disable must be delayed from the loading of a single, or last, character until the TxRDY becomes active again after the character is loaded.

#### MR2[4] – Clear-to-Send Control

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

#### MR2[3:0] – Stop Bit Length Select

This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1–9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1–1/16 to 2 stop bits can be programmed in increments of 1/16 bit. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1X clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.



# Universal asynchronous receiver/transmitter (UART)

SCC2691

**Table 2. Register Bit Formats**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MR1 (Mode Register 1)</b>							
<b>RxRTS Control</b>	<b>RxINT Select</b>	<b>Error Mode*</b>	<b>Parity Mode</b>		<b>Parity Type</b>	<b>Bits per Character</b>	
0 = no 1 = yes	0 = RxRDY 1 = FFULL	0 = Char 1 = Block	00 = With parity 01 = Force parity 10 = No parity 11 = Special mode		0 = Even 1 = Odd	00 = 5 01 = 6 10 = 7 11 = 8	
<b>NOTE:</b> *In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.							
<b>MR2 (Mode Register 2)</b>							
<b>Channel Mode</b>		<b>TxRTS Control</b>	<b>CTS Enable Tx</b>	<b>Stop Bit Length*</b>			
00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = No 1 = Yes	0 = No 1 = Yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000
<b>NOTE:</b> *Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/character.							
<b>CSR (Clock Select Register)</b>							
<b>Receiver Clock Select</b>				<b>Transmitter Clock Select</b>			
See Text				See Text			
See Table 6 for BRG Test frequencies in this data sheet, and "Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B" Philips Semiconductors ICs for Data Communications, IC-19, 1994.							
<b>CR (Command Register)</b>							
<b>Miscellaneous Commands</b>				<b>Disable Tx</b>	<b>Enable Tx</b>	<b>Disable Rx</b>	<b>Enable Rx</b>
See Text				0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes
<b>NOTE:</b> Access to the miscellaneous commands should be separated by 3 X1 clock edges. A disabled transmitter cannot be loaded.							
<b>SR (Channel Status Register)</b>							
<b>Received Break</b>	<b>Framing Error</b>	<b>Parity Error</b>	<b>Overrun Error</b>	<b>TxE<sub>MT</sub></b>	<b>TxRDY</b>	<b>FFULL</b>	<b>RxRDY</b>
0 = No 1 = Yes *	0 = No 1 = Yes *	0 = No 1 = Yes *	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes
<b>NOTE:</b> *These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits [7:5] from the top of the FIFO together with bits [4;0]. These bits are cleared by a reset error status command. In character mode they are reset when the corresponding data character is read from the FIFO. In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.							
<b>ACR (Auxiliary Control Register)</b>							
<b>BRG Set Select</b>	<b>Counter/Timer Mode and Source</b>			<b>Power-Down Mode</b>	<b>MPO Pin Function Select</b>		
0 = Set 1 1 = Set 2	See Text			0 = On PWRDN Active 1 = Off Normal	000 = RTSN 001 = C/TO 010 = Tx <sub>C</sub> (1X) 011 = Tx <sub>C</sub> (16X)	100 = Rx <sub>C</sub> (1X) 101 = Rx <sub>C</sub> (16X) 110 = Tx <sub>RDY</sub> 111 = Rx <sub>RDY</sub> /FFULL	
<b>ISR (Interrupt Status Register)</b>							
<b>MPI Pin Change</b>	<b>MPI Pin Current State</b>	<b>Not used</b>	<b>Counter Ready</b>	<b>Delta Break</b>	<b>RxRDY/FFULL</b>	<b>TxE<sub>MT</sub></b>	<b>TxRDY</b>
0 = No 1 = Yes	0 = Low 1 = High		0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes
<b>IMR (Interrupt Mask Register)</b>							
<b>MPI Change Interrupt</b>	<b>MPI Level Interrupt</b>	<b>Not used</b>	<b>Counter Ready Int</b>	<b>Delta Break Interrupt</b>	<b>RxRDY/FFULL Interrupt</b>	<b>TxE<sub>MT</sub> Interrupt</b>	<b>TxRDY Interrupt</b>
0 = Off 1 = On	0 = Off 1 = On		0 = Off 1 = On	0 = Off 1 = On	0 = Off 1 = On	0 = Off 1 = On	0 = Off 1 = On

# Universal asynchronous receiver/transmitter (UART)

# SCC2691

**Table 2. Register Bit Formats (Continued)**

CTUR (Counter/Timer Upper Register)							
C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CTLR (Counter/Timer Lower Register)							
C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

**CSR – Clock Select Register (see Table 6. also)**

**Table 3. Baud Rate Selection**

CSR[3:0]/ [7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	MPI – 16X	MPI – 16X
1 1 1 1	MPI–1X	MPI–1X

The receiver clock is always a 16X clock, except for CSR[7:4] = 1111.

See “Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B” in application notes elsewhere in this publication

**CSR[7:4] – Receiver Clock Select**

This field selects the baud rate clock for the receiver as shown in Table 3. The baud rates listed are for a 3.6864MHz crystal or external clock.

**CSR[3:0] – Transmitter Clock Select**

This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3.

**CR – Command Register**

CR is used to write commands to the UART. Multiple commands can be specified in a single write to CR as long as the commands are non-conflicting, e.g., the enable transmitter and reset transmitter commands cannot be specified in a single command word.

**CR[7:4] – Miscellaneous Commands**

The encoded value of this field may be used to specify a single command as follows:

NOTE: Access to the upper four bits of the command register should be separated by three (3) edges of the X1 clock.

- 0000 No command.
- 0001 Reset MR pointer. Causes the MR pointer to point to MR1.
- 0010 Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disable and the FIFO is flushed.

- 0011 Reset transmitter. Resets the transmitter as if a hardware reset had been applied
- 0100 Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared), and in block mode to clear all error status after a block of data has been received.
- 0101 Reset break change interrupt. Causes the break detect change bit in the interrupt status register (ISR[3]) to be cleared to zero.
- 0110 Start break. Forces the TxD output low (spacing). If the transmitter is empty, the start of the break condition will be delayed up to two bit times. If the transmitter is active, the break begins when transmission of the character is completed. If a character is in the THR, the start of break is delayed until that character or any others loaded after it have been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break
- 0111 Stop break. The TxD line will go high (marking) within two bit times. TxD will remain high for one bit time before the next character, if any, is transmitted.
- 1000 Start C/T. In counter or timer modes, causes the contents of CTUR/CTLR to be preset into the counter/timer and starts the counting cycle. In timer mode, any counting cycle in progress when the command is issued is terminated. In counter mode, has no effect unless a stop C/T command was issued previously.
- 1001 Stop counter. In counter mode, stops operation of the counter/timer, resets the counter ready bit in the ISR, and forces the MPO output high if it is programmed to be the output of the C/T. In timer mode, resets the counter ready bit in the ISR but has no effect on the counter/timer itself or on the MPO output.
- 1010 Assert RTSN. Causes the RTSN output (MPO) to be asserted (low).
- 1011 Negate RTSN. Causes the RTSN output (MPO) to be negated (high).
- 1100 Reset MPI change interrupt. Causes the MPI change bit in the interrupt status register (ISR[7]) to be cleared to zero.
- 1100 Reserved.
- 111x Reserved.

**CR[3] – Disable Transmitter**

This command terminates operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state. A disabled transmitter cannot be loaded.

**CR[2] – Enable Transmitter**

Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

### CR[1] – Disable Receiver

This command terminates operation of the receiver immediately; a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special wake-up mode is programmed, the receiver operates even if it is disabled (see Wake-up Mode).

### CR[0] – Enable Receiver

Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

### SR – Channel Status Register

The status register is updated while RDN is negated. Therefore, the bus interface used with this device must not use a static RDN line. The RDN line must be pulsed to allow status register updates.

### SR[7] – Received Break

This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received; further entries to the FIFO are inhibited until the RxD line returns to the marking state for at least one half bit time two successive edges of the internal or external 1X clock. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.**

When this bit is set, the change in break bit in the ISR (ISR[3]) is set. ISR[3] 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 time in order for it to be detected.

### SR[6] – Framing Error (FE)

This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

### SR[5]– Parity Error (PE)

This bit is set when the with parity or force parity mode is programmed and the corresponding character in the FIFO was received with incorrect parity. In special wake-up mode, the parity error bit stores the received A/D bit.

### SR[4] – Overrun Error (OE)

This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost. This bit is cleared by a reset error status command.

### SR[3] – Transmitter Empty (TxEMT)

This bit will be set when the transmitter underruns, i.e., both the TxEMT and TxRDY bits are set. This bit and TxRDY are set when the transmitter is first enabled and at any time it is re-enabled after either (a) reset, or (b) the transmitter has assumed the disabled state. It is always set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission.

It is reset when the THR is loaded by the CPU, a pending transmitter disable is executed, the transmitter is reset, or the transmitter is disabled while in the underrun condition.

### SR[2] – Transmitter Ready (TxRDY)

This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

### SR[1] – FIFO Full (FFULL)

This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL will be reset by the CPU read and then set by the transfer of the character to the FIFO, which causes all three FIFO positions to be occupied.

### SR[0] – Receiver Ready (RxRDY)

This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, and no more characters are in the FIFO.

### ACR – Auxiliary Control Register

#### ACR[7] – Baud Rate Generator Set Select

This bit selects one of two sets of baud rates generated by the BRG.

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the receiver and transmitter. See Table 3 for characteristics of the BRG.

#### ACR[6:4] – Counter/Timer Mode and Clock Source Select

This field selects the operating mode of the counter/timer and its clock source as follows:

**Table 4. ACR[6:4] Operating Mode**

ACR [6:4]	Mode	Clock Source
0 0 0	Counter	MPI pin
0 0 1	Counter	MPI pin divided by 16
0 1 0	Counter	TxC-1X clock of the transmitter
0 1 1	Counter	Crystal or X1/CLK divided by 16
1 0 0	Timer (square wave)	MPI pin
1 0 1	Timer (square wave)	MPI pin divided by 16
1 1 0	Timer (square wave)	Crystal or external clock (X1/CLK)
1 1 1	Timer (square wave)	Crystal or X1/CLK divided by 16

The timer mode generates a squarewave.

#### ACR[3] – Power-Down Mode Select

This bit, when set to zero, selects the power-down mode. In this mode, the SCC2691 oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior

# Universal asynchronous receiver/transmitter (UART)

# SCC2691

to placing the SCC2691 in this mode. Note that this bit must be set to a logic 1 after reset.

When the power-down mode is enabled, internal circuitry forces the X1/CLK pin to the low state and the X2 pin to the high state. If an external clock is being used to drive the device, it is recommended that the clock source be three-stated or forced low while the UART is in power-down mode in order to prevent the clock driver from being short circuited.

**Table 5. BRG Characteristics**

Nom Rate (Baud)	Actual 16X Clock (kHz)	Error (%)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
14.4K	230.4	0
19.2k	307.2	0
28.8K	460.8	0
38.4k	614.4	0
57.6K	921.6	0
115.2K	1843.2K	0

Duty cycle of 16X clock is 50% ±1%. Crystal or Clock = 3.6864MHz

Asynchronous UART communications can tolerate frequency error of 4.1% to 6.7% in a “clean” communications channel. The percent of error changes as the character length changes. The above percentages range from 5 bits not parity to 8 bits with parity and one stop bit. The error with 8 bits no parity and one stop bit is 4.6%. If a stop bit length of 9/16 is used, the error tolerance will approach 0 due to a variable error of up to 1/16 bit time in receiver clock phase alignment to the start bit.

**ACR[2:0] – MPO Output Select**

This field programs the MPO output pin to provide one of the following:

- 000 Request-to-send active-low output (RTSN). This output is asserted and negated via the command register. RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.
- 001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTUR and CTLR. In the counter mode, the output remains high until the terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command.
- 010 The 1X clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1X clock is output.

- 011 The 16X clock for the transmitter. This is the clock selected by CSR[3:0] = 1111.
- 100 The 1X clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1X clock is output.
- 101 The 16X clock for the receiver. This is the clock selected by CSR[7:4], and is a 1X clock if CSR[7:4] = 1111.
- 110 The transmitter register empty signal, which is the complement of SR[2]. Active low output.
- 111 The receiver ready or FIFO full signal (complement of ISR[2]). Active-low output.

**ISR – Interrupt Status Register**

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a ‘1’ and the corresponding bit in the IMR is also a ‘1’, the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR; the true status is provided regardless of the contents of the IMR. This register is cleared when the device is reset.

**ISR[7] – MPI Change-of-State**

This bit is set when a change-of-state occurs at the MPI input pin. It is reset by a reset change interrupt command.

**ISR[6] – MPI Current State**

This bit provides the current state of the MPI pin. This information is latched and reflects the state of the pin at the leading edge of the ISR ready cycle.

**ISR[4] – Counter Ready**

In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

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

**ISR[3] – Change in Break**

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

**ISR[2] – Receiver Ready or FIFO Full**

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the FIFO is read and there is no character in the receive shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

**ISR[1] – Transmitter Empty**

This bit is a duplicate of TxEMT (SR[3]).

**ISR[0] – Transmitter Ready**

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



# Universal asynchronous receiver/transmitter (UART)

# SCC2691

## IMR – Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is a '1', the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask reading of the ISR. NOTE: When IMR[6] is a 1, a 1 on the MPI pin causes an interrupt.

## CTUR and CTLR – Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded is H'0002'.

In the timer (programmable divider) mode, the C/T generates a square wave whose period is twice the value (in clock periods) of the CTUR and CTLR. The waveform so generated is often used for a data clock. The formula for calculating the divisor n to load to the CTUR and CTLR for a particular 1X data clock is shown below:

$$n = \frac{\text{C/T Clock Frequency}}{2 \times 16 \times \text{Baud rate desired}}$$

Often this division will result in a non-integer number; 26.3, for example. One can only program integer numbers in a digital divider. Therefore, 26 would be chosen. This gives a baud rate error of

0.3/26.3 which is 1.14%; well within the ability asynchronous mode of operation.

If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half-periods will be.

The counter ready status bit (ISR[4]) is set once each cycle of the square wave. The bit is reset by a stop counter command. The command, however, does not stop the C/T. The generated square wave is output on MPO if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded in CTUR and CTLR. Counting begins upon receipt of a start C/T command. Upon reaching the terminal count, the counter ready interrupt bit (ISR[4]) is set. the counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until the terminal count is reached, at which time it goes low.

The output returns to the high state and ISR[4] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous values are preserved and used for the next count cycle.



Figure 3. Reset Timing

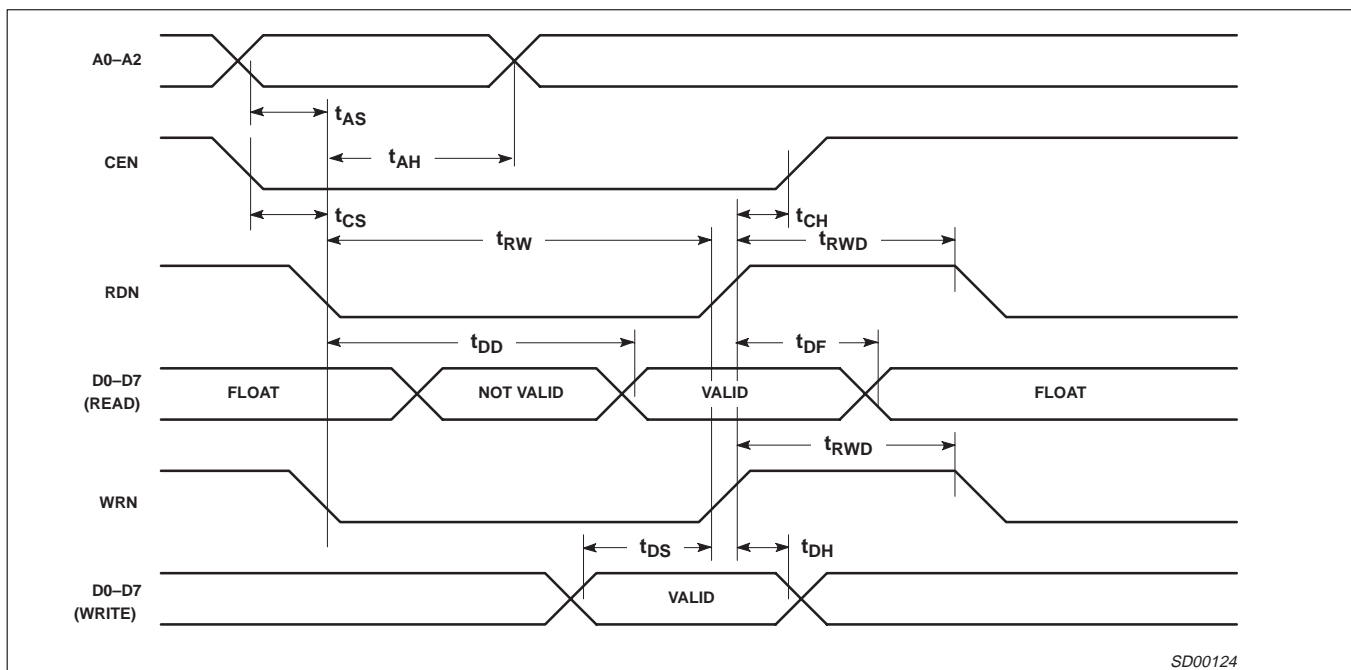


Figure 4. Bus Timing

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

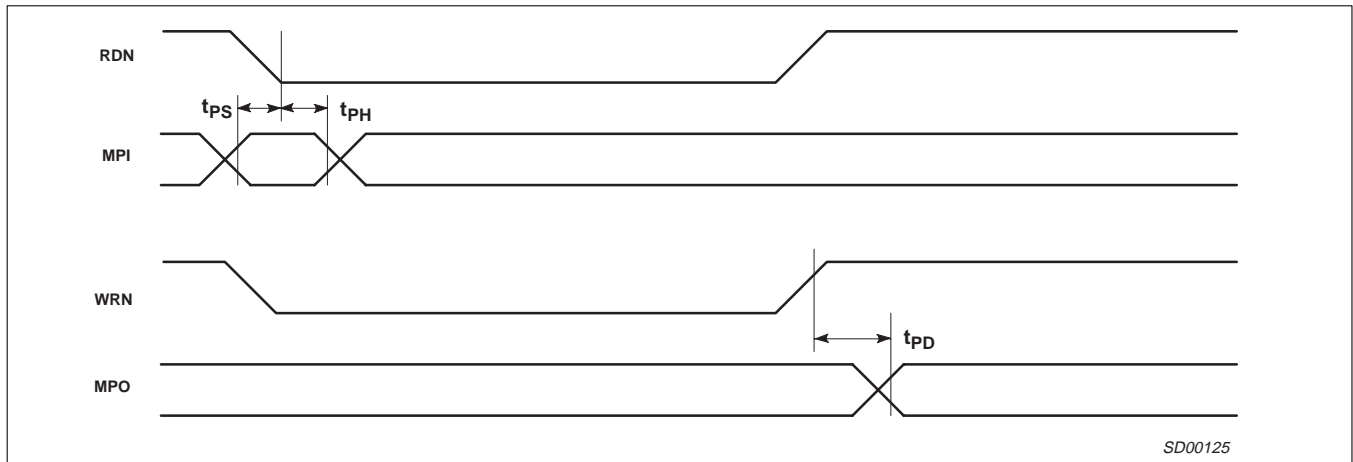
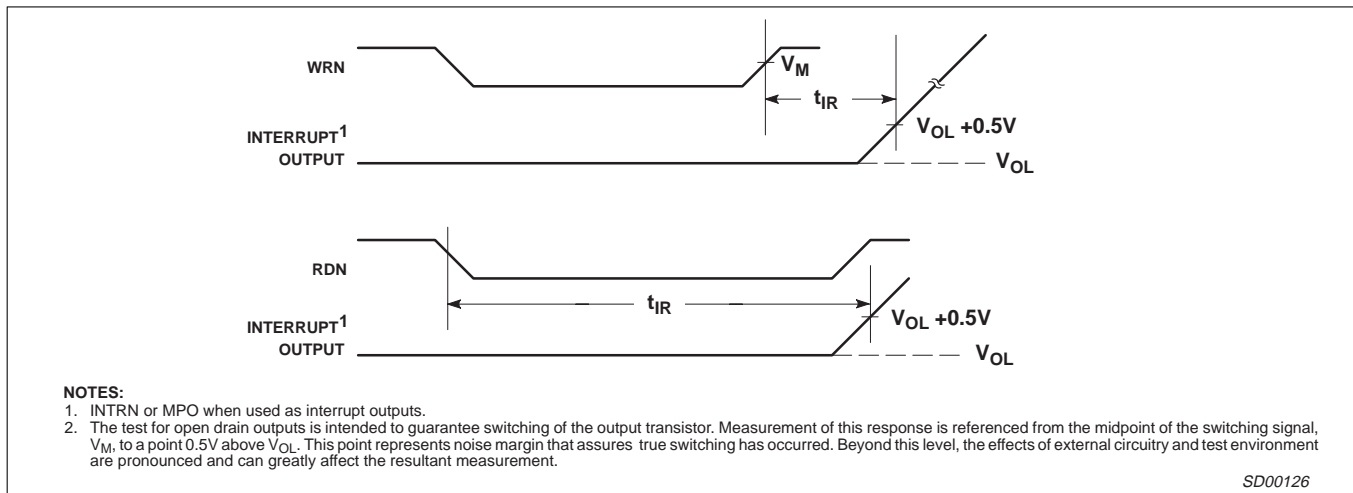


Figure 5. I/O Timing



**NOTES:**

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

SD00126

Figure 6. Interrupt Timing

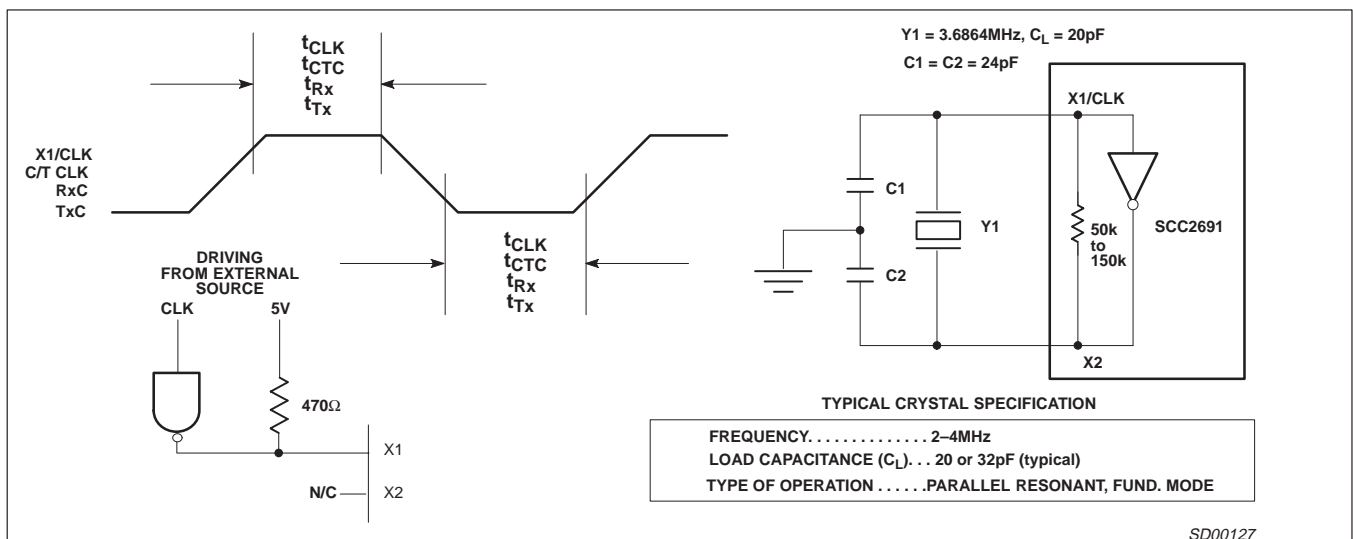


Figure 7. Clock Timing

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

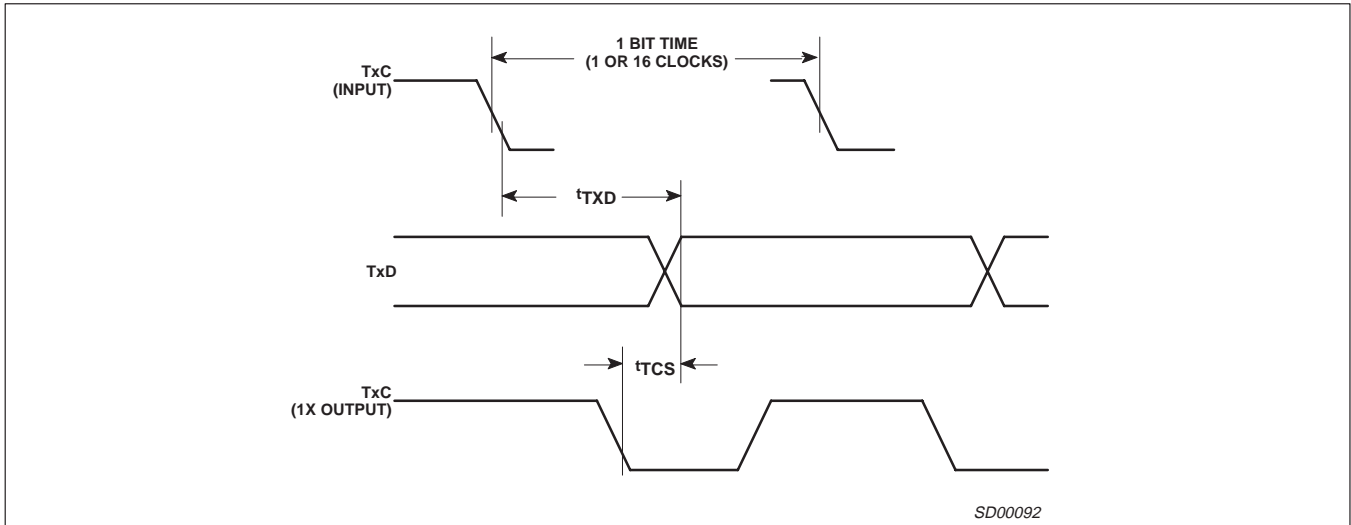


Figure 8. Transmit Timing

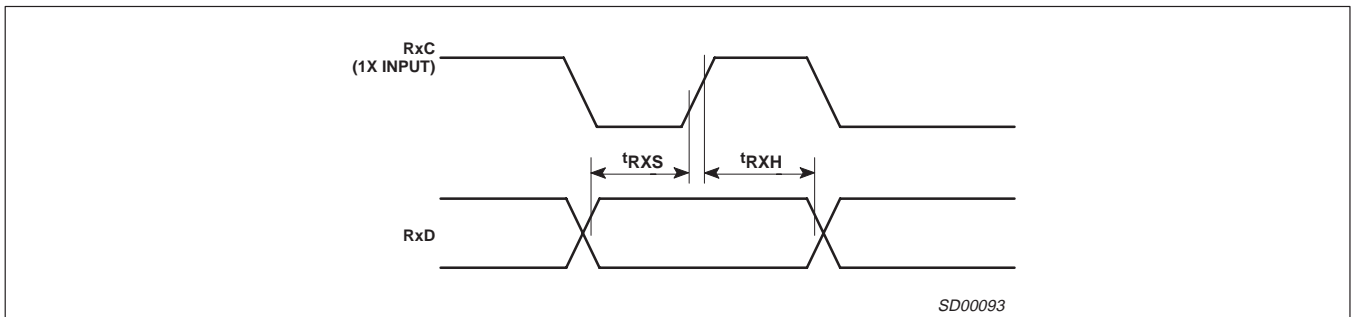


Figure 9. Receive Timing

# Universal asynchronous receiver/transmitter (UART)

SCC2691

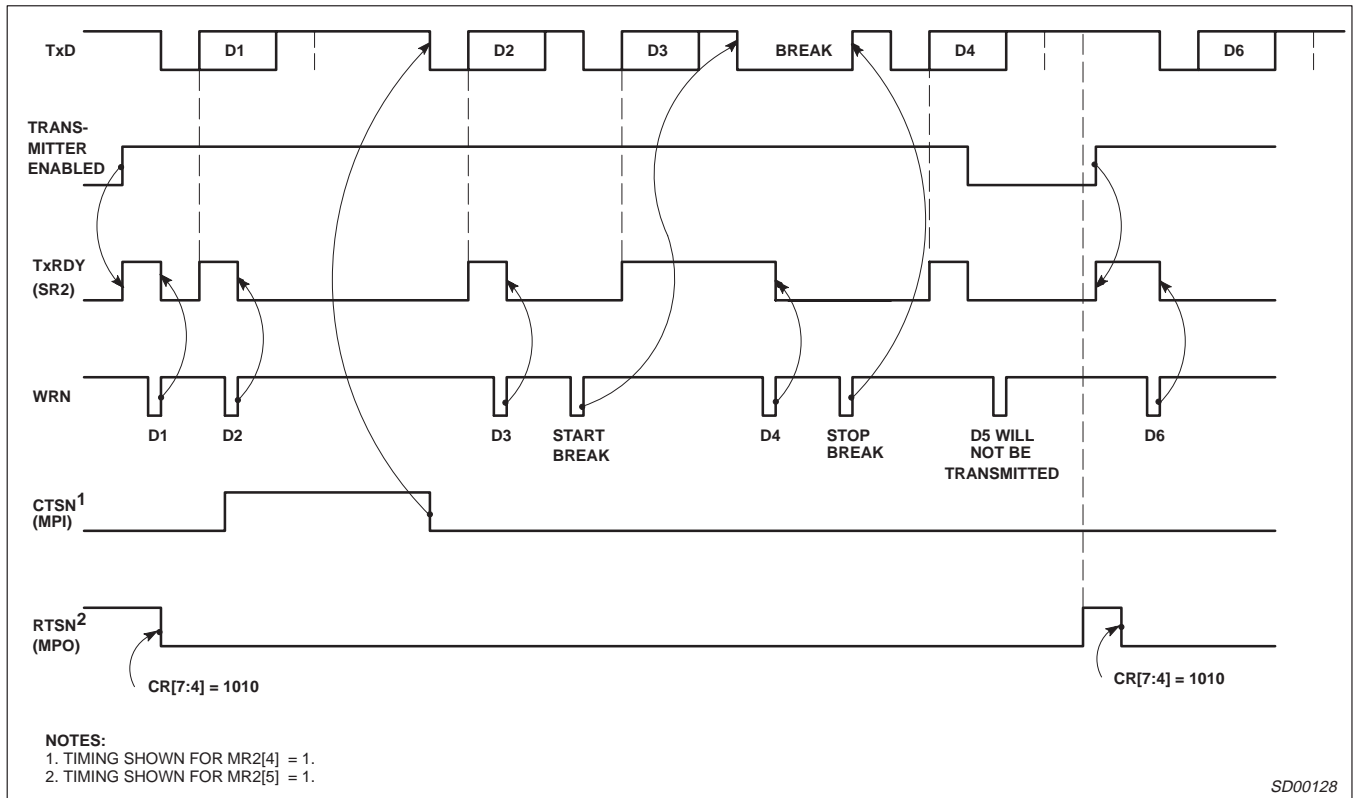


Figure 10. Transmitter Timing

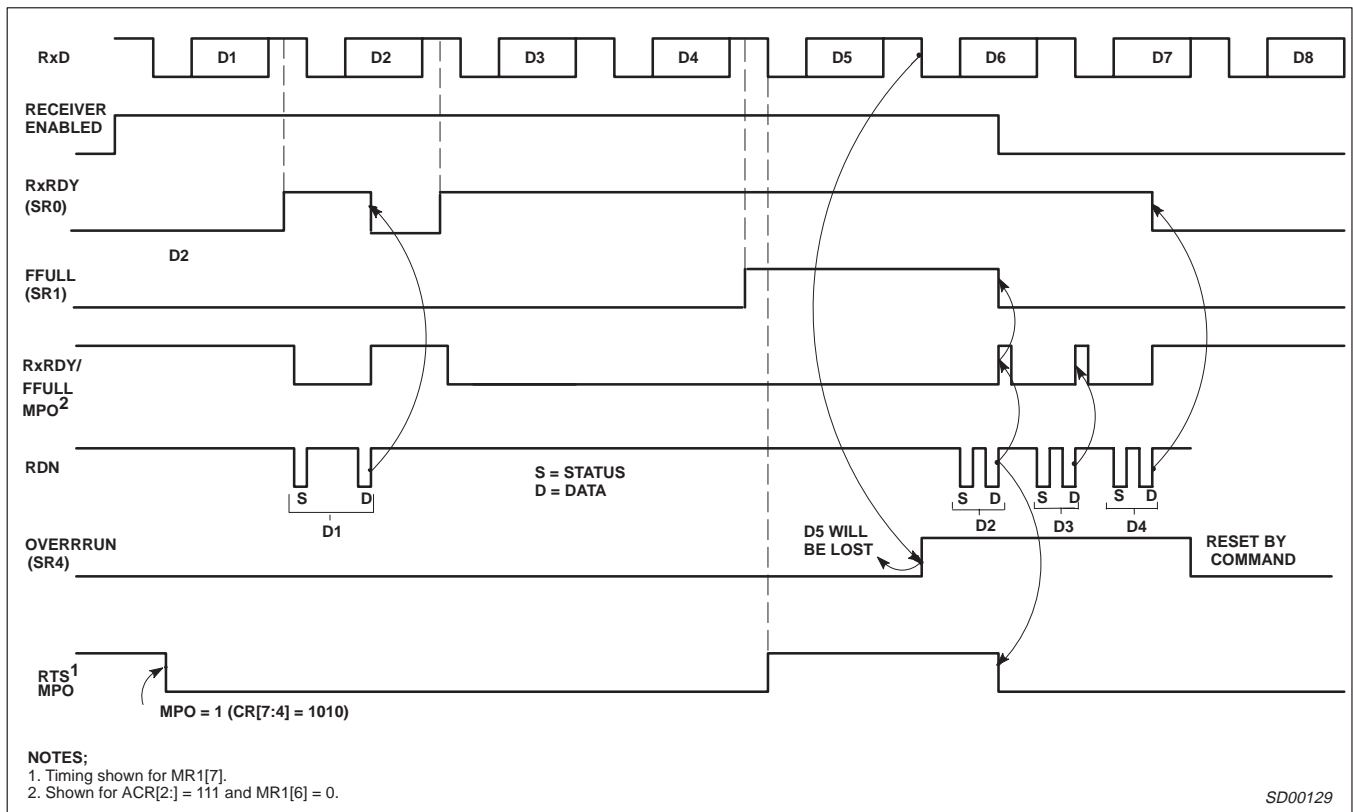


Figure 11. Receiver Timing

# Universal asynchronous receiver/transmitter (UART)

SCC2691

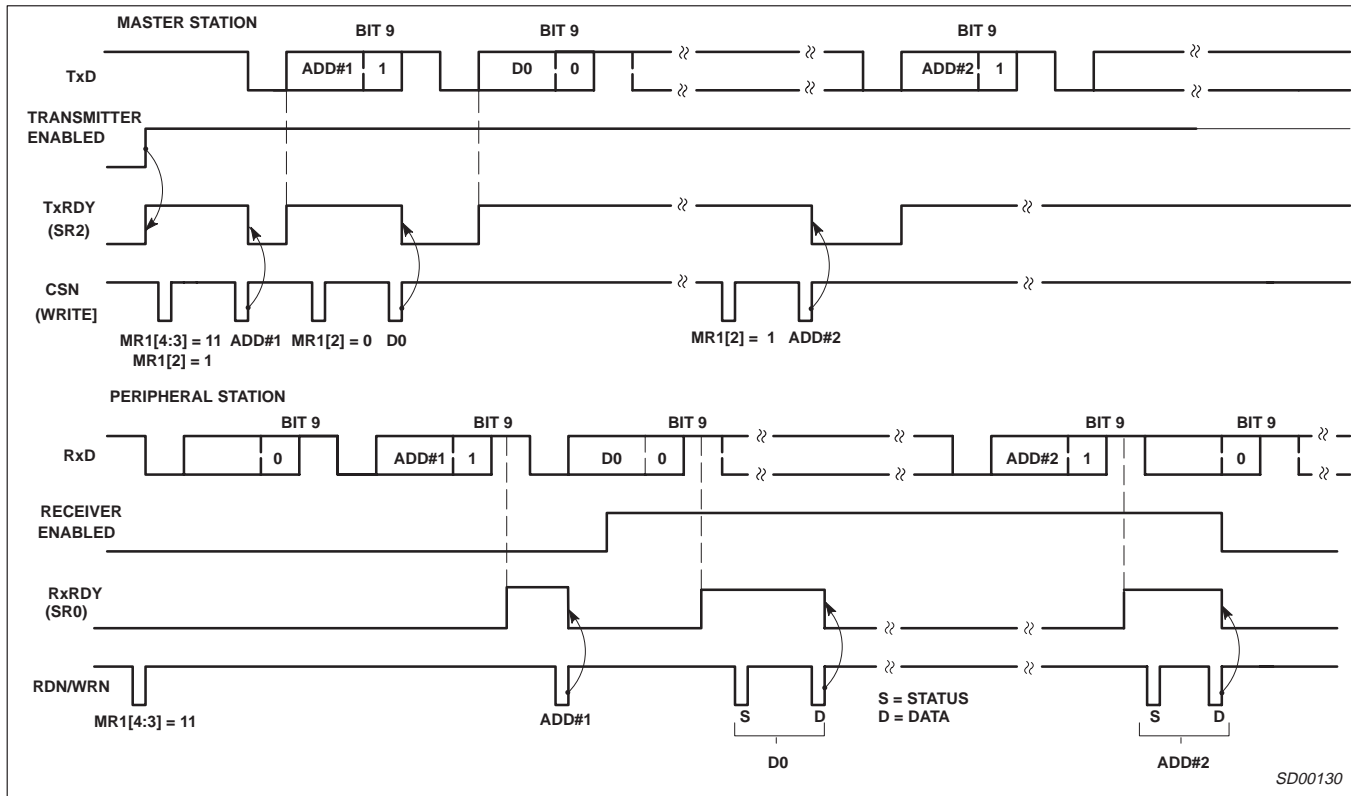


Figure 12. Wake-Up Mode

## 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 MPI. The CTS signal is active low; thus, it is called CTSN.

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, thus, called RTSN. RTSN is on pin MP0. A receiver's RTS 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 (MPI). 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 fourth character is sensed. Transmission then stops with four 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 MP pin will have no effect on the operation of the transmitter.

MR1(7) is the bit that allows the receiver to control MP0. When MP0 is controlled by the receiver, the meaning of that pin will be RTS. However, a point of confusion arises in that MP0 may also be controlled by the transmitter. When the transmitter is controlling this pin, its meaning is not RTS at all. It is, rather, that the transmitter has finished sending its last data byte. Programming the MP0 pin to be controlled by the receiver and the transmitter at the same time is allowed, but would usually be incompatible.

RTS can also be controlled by the commands 1010 and 1011 in the command register. RTS is expressed at the MP0 pin which is still an output port. Therefore, the state of MP0 should be set low (by commands to the CR register) for the receiver to generate the proper RTS signal. The logic at the output is basically a NAND of the MP0 bit register and the RTS signal as generated by the receiver. When the RTS flow control is selected via the MR(7) bit the state of the MP0 register is not changed. Terminating the use of "Flow Control" (via the MR registers) will return the MP0 pin to the control of the MP0 register.

## Transmitter Disable Note

The sequence of instructions enable transmitter — load transmit holding register — disable transmitter will result in nothing being sent if the time between the end of loading the transmit holding register and the disable command is less than 3/16 bit time in the 16x mode or one bit time in the 1x mode. Also, if the transmitter, while in the enabled state and underrun condition, is immediately disabled after a single character is loaded to the transmit holding register, that character will not be sent.

In general, when it is desired to disable the transmitter before the last character is sent AND the TxEMT bit is set in the status register (TxEMT is always set if the transmitter has underrun or has just been enabled), be sure the TxRDY bit is active immediately before issuing the transmitter disable instruction. TxRDY sets at the end of the "start bit" time. It is during the start bit that the data in the transmit holding register is transferred to the transmit shift register.

Non-standard baud rates are available as shown in Table 6 below, via the BRG Test function.

## Universal asynchronous receiver/transmitter (UART)

SCC2691

Table 6. Baud Rates Extended

CSR[7:4]	Normal BRG		BRG Test	
	ACR[7] = 0	ACR[7] = 1	ACR[7] = 0	ACR[7] = 1
0000	50	75	4,800	7,200
0001	110	110	880	880
0010	134.5	134.5	1,076	1,076
0011	200	150	19.2K	14.4K
0100	300	300	28.8K	28.8K
0101	600	600	57.6K	57.6K
0110	1,200	1,200	115.2K	115.2K
0111	1,050	2,000	1,050	2,000
1000	2,400	2,400	57.6K	57.6K
1001	4,800	4,800	4,800	4,800
1010	7,200	1,800	57.6K	14.4K
1011	9,600	9,600	9,600	9,600
1100	38.4K	19.2K	38.4K	19.2K
1101	Timer	Timer	Timer	Timer
1110	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X	I/O2 – 16X
1111	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X	I/O2 – 1X

## NOTE:

Each read on address H'2' will toggle the baud rate test mode. When in the BRG test mode, the baud rates change as shown to the left. This change affects all receivers and transmitters on the DUART. See "Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B" in application notes elsewhere in this publication.

The test mode at address H'A' changes all transmitters and receivers to the 1x mode and connects the output ports to some internal nodes.

A condition that occurs infrequently has been observed where the receiver will ignore all data. It is caused by a corruption of the start bit generally due to noise. When this occurs the receiver will appear to be asleep or locked up. The receiver must be reset for the UART to continue to function properly.

**Reset in the Normal Mode (Receiver Enabled)**

Recovery can be accomplished easily by issuing a receiver software reset followed by a receiver enable. All receiver data, status and programming will be preserved and available before reset. The reset will NOT affect the programming.

**Reset in the Wake-Up Mode (MR1[4:3] = 11)**

Recovery can also be accomplished easily by first exiting the wake-up mode (MR1[4:3] = 00 or 01 or 10), then issuing a receiver software reset followed by a wake-up re-entry (MR1[4:3] = 11). All receiver data, status and programming will be preserved and available before reset. The reset will NOT affect the programming.

The receiver has a digital filter designed to reject "noisy" data transitions and the receiver state machine was designed to reject noisy start bits or noise that might be considered a start bit. In spite of these precautions, corruption of the start bit can occur in 15ns window approximately 100ns prior to the rising edge of the data clock. The probability of this occurring is less than  $10^{-5}$  at 9600 baud.

A corrupted start bit may have some deleterious effects in ASYNC operation if it occurs within a normal data block. The receiver will tend to align its data clock to the next '0' bit in the data stream, thus potentially corrupting the remainder of the data block. A good design practice, in environments where start bit corruption is possible, is to monitor data quality (framing error, parity error, break change and received break) and "data stopped" time out periods. Time out periods can be enabled using the counter/timer in the SCC2691, SCC2692, SCC2698B and SC68692 products. This monitoring can indicate a potential start bit corruption problem.

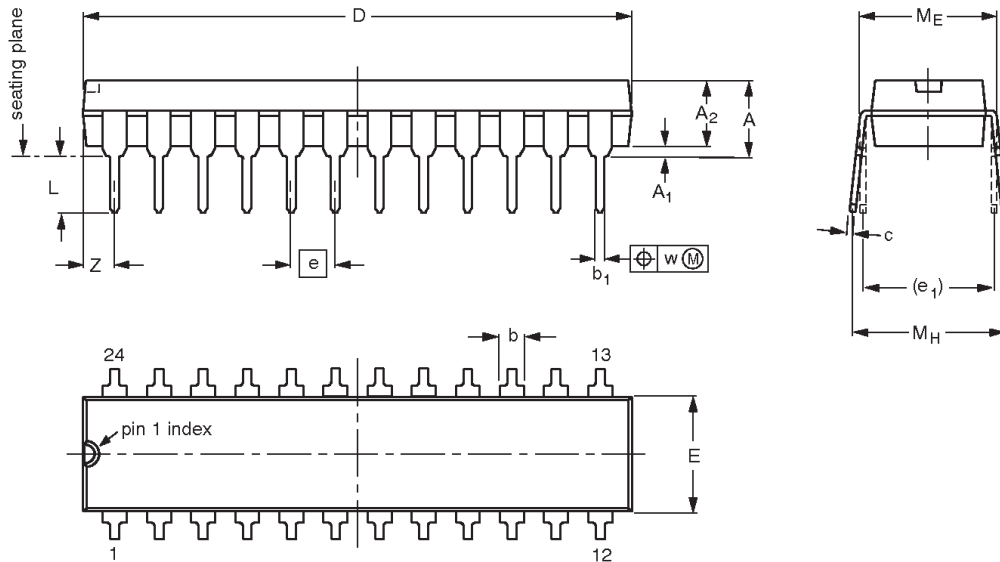
SD00097

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

DIP24: plastic dual in-line package; 24 leads (300 mil)

SOT222-1



**DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)**

UNIT	A max.	A <sub>1</sub> min.	A <sub>2</sub> max.	b	b <sub>1</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	e <sub>1</sub>	L	M <sub>E</sub>	M <sub>H</sub>	w	Z <sup>(1)</sup> max.
mm	4.70	0.38	3.94	1.63 1.14	0.56 0.43	0.36 0.25	31.9 31.5	6.73 6.48	2.54	7.62	3.51 3.05	8.13 7.62	10.03 7.62	0.25	2.05
inches	0.185	0.015	0.155	0.064 0.045	0.022 0.017	0.014 0.010	1.256 1.240	0.265 0.255	0.100	0.300	0.138 0.120	0.32 0.30	0.395 0.300	0.01	0.081

**Note**

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

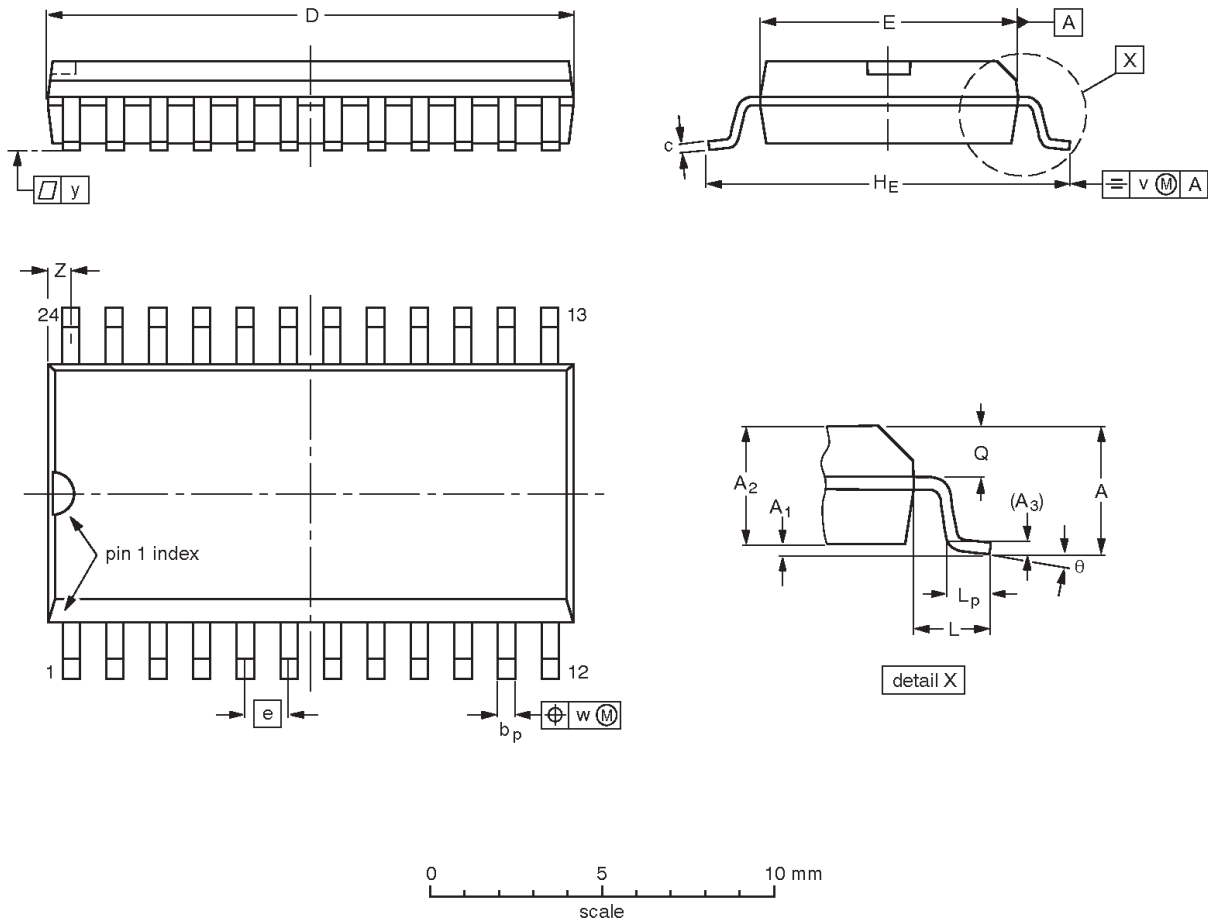
OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT222-1		MS-001AF				95-03-11

# Universal asynchronous receiver/transmitter (UART)

## SCC2691

**SO24:** plastic small outline package; 24 leads; body width 7.5 mm

**SOT137-1**



**DIMENSIONS (inch dimensions are derived from the original mm dimensions)**

UNIT	A max.	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	b <sub>p</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	H <sub>E</sub>	L	L <sub>p</sub>	Q	v	w	y	z <sup>(1)</sup>	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	15.6 15.2	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.61 0.60	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

**Note**

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

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT137-1	075E05	MS-013AD				95-01-24 97-05-22

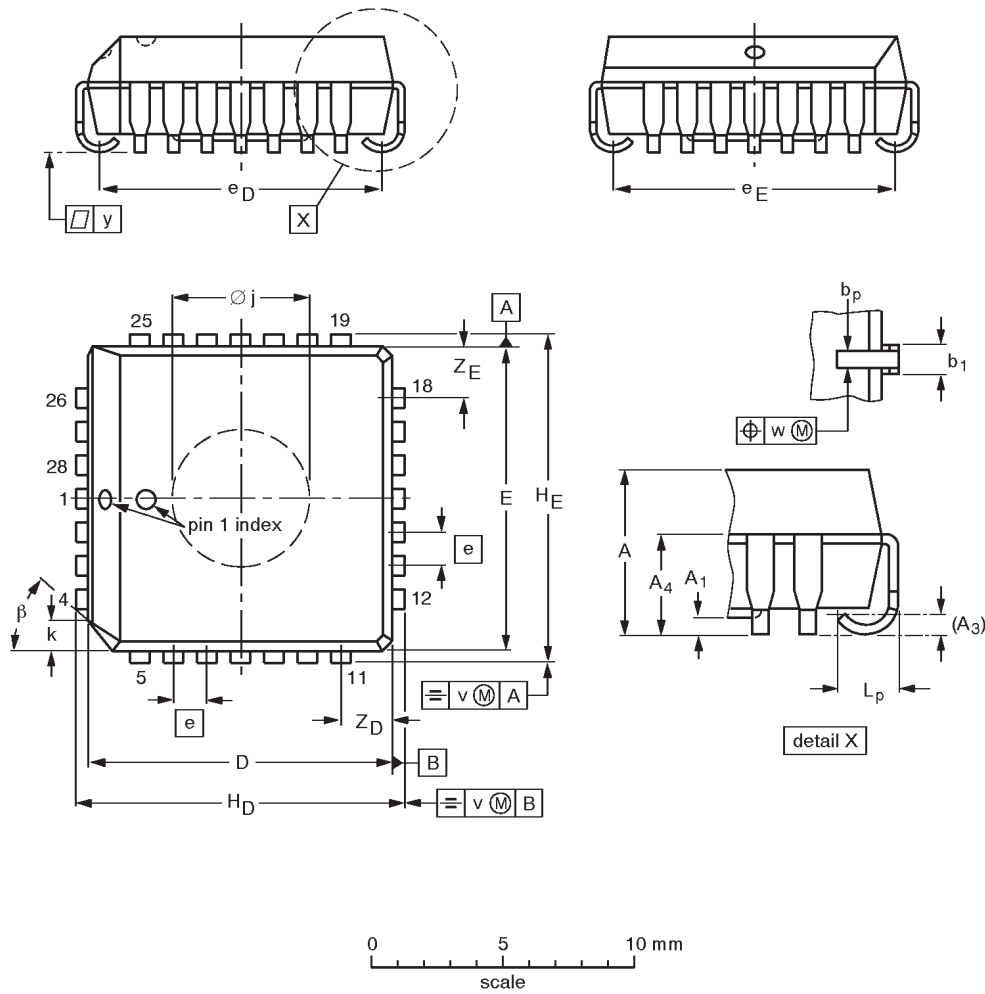


# Universal asynchronous receiver/transmitter (UART)

## SCC2691

PLCC28: plastic leaded chip carrier; 28 leads; pedestal

SOT261-3



**DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)**

UNIT	A	A <sub>1</sub> min.	A <sub>3</sub>	A <sub>4</sub> max.	b <sub>p</sub>	b <sub>1</sub>	D <sup>(1)</sup>	E <sup>(1)</sup>	e	e <sub>D</sub>	e <sub>E</sub>	H <sub>D</sub>	H <sub>E</sub>	k	∅ <sub>j</sub>	L <sub>p</sub>	v	w	y	Z <sub>D</sub> <sup>(1)</sup> max.	Z <sub>E</sub> <sup>(1)</sup> max.	β
mm	4.57 4.19	0.13	0.25	3.05	0.53 0.33	0.81 0.66	11.58 11.43	11.58 11.43	1.27	10.92 9.91	10.92 9.91	12.57 12.32	12.57 12.32	1.22 1.07	5.69 5.54	1.44 1.02	0.18	0.18	0.10	2.06	2.06	45°
inches	0.180 0.165	0.005	0.01	0.12	0.021 0.013	0.032 0.026	0.456 0.450	0.456 0.450	0.05	0.430 0.390	0.430 0.390	0.495 0.485	0.495 0.485	0.048 0.042	0.224 0.218	0.057 0.040	0.007	0.007	0.004	0.081	0.081	

**Note**

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT261-3		MO-047AB			95-02-25 97-12-16

## Universal asynchronous receiver/transmitter (UART)

SCC2691

## Data sheet status

Data sheet status	Product status	Definition [1]
Objective specification	Development	This data sheet contains the design target or goal specifications for product development. Specification may change in any manner without notice.
Preliminary specification	Qualification	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product specification	Production	This data sheet contains final specifications. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

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

## 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 134). 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.

## 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, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. 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.

Philips Semiconductors  
811 East Arques Avenue  
P.O. Box 3409  
Sunnyvale, California 94088-3409  
Telephone 800-234-7381

© Copyright Philips Electronics North America Corporation 1998  
All rights reserved. Printed in U.S.A.

Date of release: 09-98

Document order number:

9397 750 04358

*Let's make things better.*