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

XR68C681

CMOS Dual Channel
UART (DUART)

September 1999-2

FEATURES

- Two Full Duplex, Independent Channels
- Asynchronous Receiver and Transmitter
- Quadruple-Buffered Receiver, Dual Buffered Transmitter
- Programmable Stop Bits in 1/16 Bit Increments
- Internal Bit Rate Generators with 23 Different Bit Rates
- Independent Bit Rate Selection for Each Transmitter and Receiver
- External Clock Capability
- Maximum Bit Rate: 1X Clock - 1Mb/s, 16X Clock - 125Kb/s
- Normal, AUTOECHO, Local LOOPBACK and Remote LOOPBACK Modes
- Multi-function 16 Bit Counter/Timer
- Interrupt Output with Eight Maskable Interrupt Conditions
- Interrupt Vector Output on Acknowledge
- 8 General Purpose Outputs
- 6 General Purpose Inputs with Change of States Detectors on Inputs
- On-chip Oscillator for Crystal
- Standby Mode to Reduce Operating Power
- Compatible with the Motorola MC68681 and the Signetic SCC68692 Devices
- Advanced CMOS Low Power Technology

APPLICATIONS

- Multimedia Systems
- Serial to Parallel/Parallel to Serial Converter
- DTE for Modem Communication Systems

GENERAL DESCRIPTION

The EXAR Dual Universal Asynchronous Receiver and Transmitter (DUART) is a data communications device that provides two fully independent full duplex asynchronous communications channels in a single package. The DUART is designed for use in microprocessor based systems and may be used in a polled or interrupt driven environment.

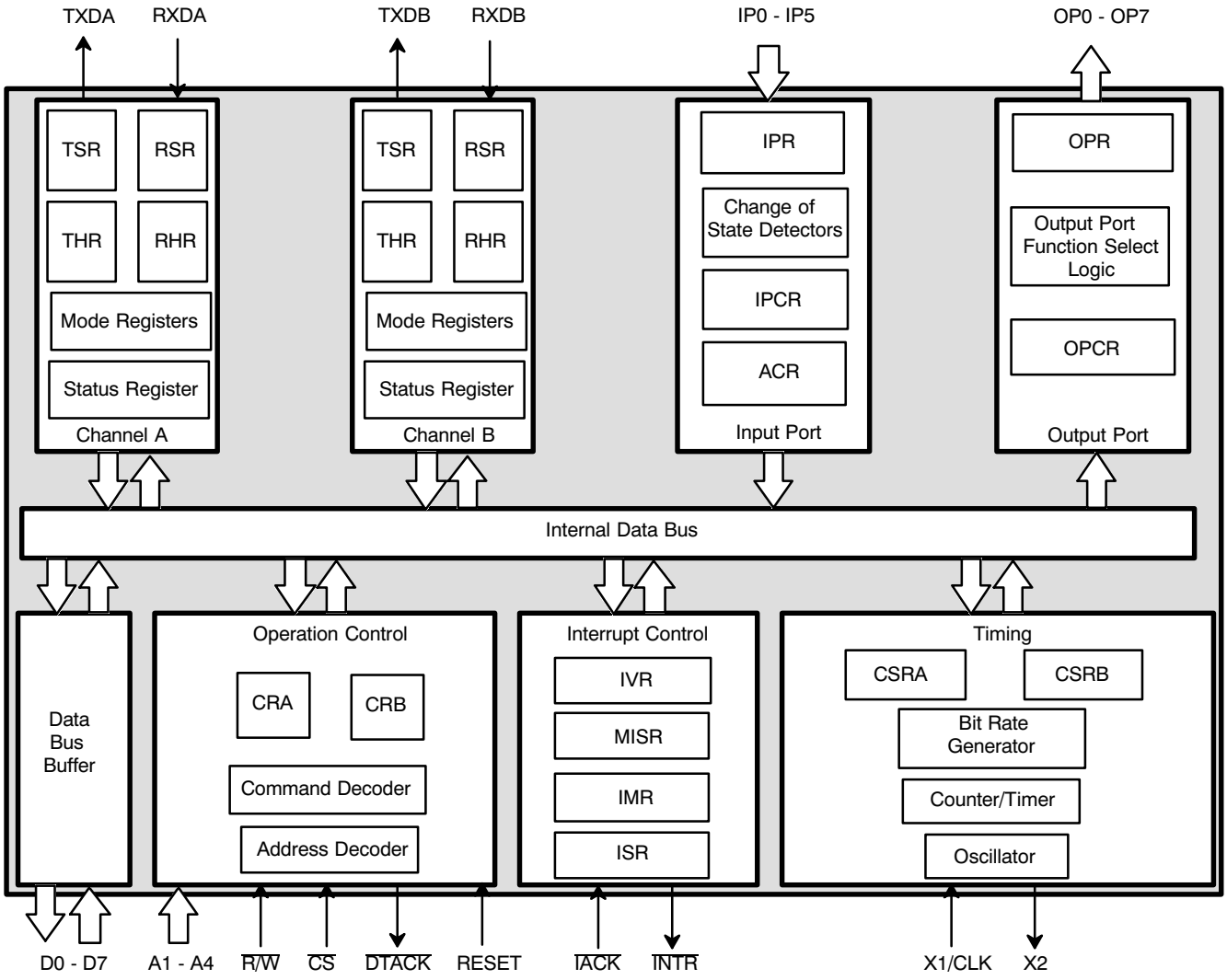
The XR68C681 device offers a single IC solution for the 68000 family of microprocessors

The DUART is fabricated using advanced two layer metal, with a high performance density EPI/CMOS 1.8 process to provide high performance and low power consumption, and is packaged in a 40 pin DIP or a 44 pin PLCC.

ORDERING INFORMATION

Part No.	Pin Package	Operating Temperature Range
XR68C681CJ	44 PLCC	0°C to +70°C
XR68C681J	44 PLCC	-40°C to +85°C

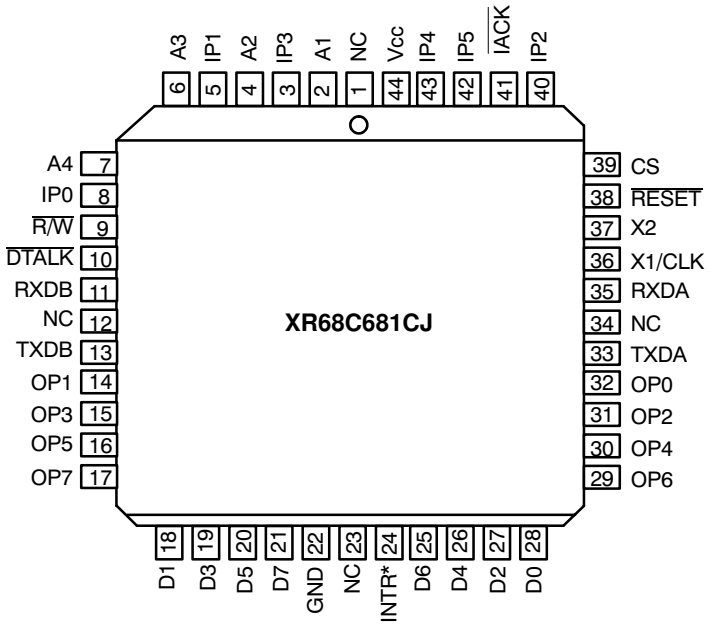
Part No.	Pin Package	Operating Temperature Range
XR68C681N	40 CDIP	-40°C to +85°C
XR68C681CP	40 PDIP	0°C to +70°C
XR68C681P	40 PDIP	-40°C to +85°C



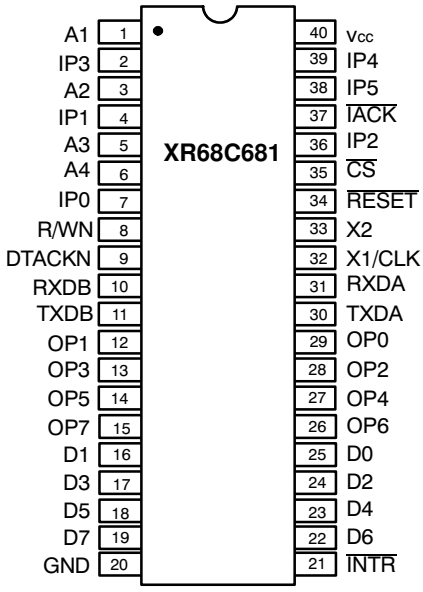
Overbar indicates Non-compliant MIL-STD-883 Product.

Figure 1. Block Diagram of the XR68C681 DUART Device

PIN CONFIGURATION



44 Pin PLCC



40 Pin PDIP, CDIP (0.600")

PIN DESCRIPTION

Pin Number (44 pin PLCC)	Pin Number (40 pin DIP)	Symbol	Type	Description
1		NC		**No Connect.**
2	1	A1	I	**LSB of Address Input.** This input, along with address inputs, A2 - A4 are used to select certain registers within the DUART device during read and write operations with the CPU.
3	2	IP3 (TXCA)	I	**Input Port 3.** General purpose input or the external clock input for Channel A Transmitter (TXCA)
4	3	A2	I	**Address Input.**
5	4	IP1 ($\overline{CTSB}$)	I	**Input Port 1.** General purpose input. This input can be configured to be the Active-low Channel B Clear-to-Send Input ($\overline{CTSB}$).
6	5	A3	I	**Address Input.**
7	6	A4	I	**MSB of Address Input.** This input, along with Address Inputs, A1 - A3 are used to select certain registers within the DUART device.

PIN DESCRIPTION (CONT'D)

Pin Number (44 pin PLCC)	Pin Number (40 pin DIP)	Symbol	Type	Description
8	7	IP0 ($\overline{CTSA}$)	I	**Input Port 0.** General purpose input. This input can also be configured to be the Active-low channel A Clear-to-Send Input ($\overline{CTSA}$).
9	8	$\overline{R/W}$	I	**Read/$\overline{Write}$ Input.** If this input is high while $\overline{CS}$ is low, then the CPU is performing a READ cycle with the DUART. If this input is low, while $\overline{CS}$ is low, then the CPU is performing a WRITE cycle with the DUART.
10	9	$\overline{DTACK}$	O	**Data Transfer Acknowledge.** Three State, active low: The DUART asserts $\overline{DTACK}$ in order to inform the CPU that the present READ or WRITE operation is nearly complete. The 68000 family of CPUs requires this signal from its peripheral devices, in order to quickly and properly complete a READ or WRITE cycle. If the DUART asserts $\overline{DTACK}$ during a READ operation, it indicates (to the CPU) that the requested data is on the data bus. If $\overline{DTACK}$ is asserted during an Interrupt Acknowledge cycle, the DUART is informing the CPU that the contents of the IVR (Interrupt Vector Register) are available on the data bus. If the DUART asserts the $\overline{DTACK}$ during a WRITE cycle, it is informing the CPU that the data, on the data bus, has been latched into the data bus buffer of the DUART device.
11	10	RXDB	I	**Receiver Serial Data Input- Channel B.** The least significant bit of the character is received first. If an external receiver clock is specified, the received data is sampled on the rising edge of this clock.
12		NC		**No Connect.**
13	11	TXDB	O	**Transmitter Serial Data Output - Channel B.** The least significant bit of the channel is transmitted first. This output is held in the marking (high) state when the transmitter is idle, disabled, or operating in the local LOOPBACK mode. If an external clock is specified, the transmitted data is shifted out of the TSR (Transmitter Shift Register) on the falling edge of this clock.
14	12	OP1 ($\overline{RTSB}$)	O	**Output 1.** A general purpose output. This output can also be configured as the active-low channel B Request-to-Send output ($\overline{RTSB}$).
15	13	OP3 (TXCB) (RXCB) $\overline{C/T\_RDY}$	O	**Output 3.** A general purpose output. This output pin can also be configured to be the channel B Transmitter 1X Clock output (TXCB), the channel B Receiver 1X Clock output (RXCB), or as an active-low, open-drain Counter/Timer Ready Output ($\overline{C/T\_RDY}$).
16	14	OP5 (RXRDY/ $\overline{FFULL\_B}$)	O	**Output 5.** A general purpose output. This output pin can also be configured to be the open-drain, active-low channel B RXRDY/FFULL output, active-low.

PIN DESCRIPTION (CONT'D)

Pin Number (44 pin PLCC)	Pin Number (40 pin DIP)	Symbol	Type	Description
17	15	OP7 ($\overline{\text{TXRDY\_A}}$)	O	**Output 7.** A general purpose output. This output pin can also be configured to be the open-drain active low channel A TXRDY output.
18	16	D1	I/O	**Three State Data Bus.**
19	17	D3	I/O	**Three State Data Bus.**
20	18	D5	I/O	**Three State Data Bus.**
21	19	D7	I/O	**MSB of Eight Bit Three State Data Bus.** All transfers between the CPU and DUART take place over the data bus (consists of pins D0 - D7). The bus is three-stated when the $\overline{\text{CS}}$ input is high, except during an $\overline{\text{IACK}}$ cycle.
22	20	GND		**Ground.** Reference
23		NC		**No Connect.**
24	21	$\overline{\text{INTR}}$	O	**Interrupt Request.** Active Low, Open-Drain. $\overline{\text{INTR}}$ is asserted upon the occurrence of one or more of the chip's maskable interrupting conditions. This signal will remain asserted throughout the interrupt service routine and will be negated once the condition(s) causing the interrupt request has been eliminated.
25	22	D6	I/O	**Three State Data Bus.**
26	23	D4	I/O	**Three State Data Bus.**
27	24	D2	I/O	**Three State Data Bus.**
28	25	D0	I/O	**LSB of the Eight Bit Three State Data Bus.** All transfers between the CPU and QUART take place over this bus. The bus is three-stated when the $\overline{\text{CS}}$ input is high, except during an $\overline{\text{IACK}}$ cycle.
29	26	OP6 ($\overline{\text{TXRDY\_A}}$)	O	**Output 6.** A general purpose output. This output pin can also be configured to be an active-low, open-drain channel A TXRDY output ($\overline{\text{TXRDY\_A}}$)
30	27	OP4 (RXRDY/ $\overline{\text{FFULL\_A}}$)	O	**Output 4.** A general purpose output. This output pin can also be configured to be an open-drain channel A RXRDY/FFULL output (active-low).
31	28	OP2 (TXCA_1X) (TXCA_16X)	O	**Output 2.** A general purpose output. This output pin can also be configured to be either 1X or 16X clock output for the channel A transmitter.
32	29	OP0 ($\overline{\text{RTSA}}$)	O	**Output 0.** A general purpose output. This output pin can also be configured to be the active-low Channel A Request-to-Send Output ($\overline{\text{RTSA}}$).

PIN DESCRIPTION (CONT'D)

Pin Number (44 pin PLCC)	Pin Number (40 pin DIP)	Symbol	Type	Description
33	30	TXDA	O	**Transmitter Serial Data Output.** Channel A. The least significant bit is transmitted first. This output is held in the marking (high) state when the transmitter is idle, disabled, or operating in the local LOOPBACK mode. If external clock is specified, the transmitted data is shifted out of the TSR (Transmitter Shift Register) on the falling edge of the clock.
34		NC		**No Connect.**
35	31	RXDA	I	**Receiver Serial Data Input.** Channel A. The least significant bit is received first. If an external receiver clock is specified, the received data is sampled on the rising edge of the clock.
36	32	X1/CLK	I or O	**Crystal Output or External Clock Input.** This pin is the connection for one side of the crystal and a capacitor to ground when the internal oscillator is used. If the oscillator is not used, an external clock signal must be supplied at this input. In order for the XR68C681 device to function properly, the user must supply a signal with frequencies between 2.0 MHz and 4.0 MHz. This requirement can be met by either a crystal oscillator or by the external TTL-compatible clock signal.
37	33	X2	I	**Crystal Input.** Connection for one side of the crystal (Opposite of X1/CLK). If the oscillator is used, a capacitor must also be connected from this pin to ground. This pin must be left open if an external clock is supplied at X1/CLK.
38	34	$\overline{\text{RESET}}$	I	**Master Reset.** A low on this pin clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), initializes the IVR to $0F_{16}$, stops the Counter/Timer, places the output port pins, OP0 - OP7 in the logic "high" state, and places both serial channels in the inactive state with the TXDA and TXDB output marking (high).
39	35	$\overline{\text{CS}}$	I	**Chip Select.** Active low. The data bus is three-stated when $\overline{\text{CS}}$ is high. Transfers between the CPU and the DUART via D0 - D7 are enabled when $\overline{\text{CS}}$ is low.
40	36	IP2 (C/T_EX) (RXCB)	I	**Input 2.** General purpose input. This input can also be configured to be C/T external clock input, or the channel B Receiver Clock Input (RXCB).
41	37	$\overline{\text{IACK}}$	I	**Interrupt Acknowledge.** Active Low. This input is the CPU's response to the interrupt request issued by the DUART device. When the CPU asserts this input, it indicates that the DUART's interrupt request is about to be serviced, and that the very next bus cycle will be an interrupt acknowledge cycle. The DUART will respond to the CPU's interrupt acknowledge by placing the contents of the Interrupt Vector Register (IVR) on the data bus (D0 - D7).

PIN DESCRIPTION (CONT'D)

Pin Number (44 pin PLCC)	Pin Number (40 pin DIP)	Symbol	Type	Description
42	38	IP5 (TXCB)	I	**Input 5.** General purpose input. This input can also be configured as the channel B transmitter external clock input (TXCB).
43	39	IP4 (RXCA)	I	**Input 4.** General purpose input. This input can also be configured as the channel A Receiver External Clock Input (RXCA).
44	40	V_{CC}		

DC ELECTRICAL CHARACTERISTICS [1, 2, 3]

Test Conditions: T_A = 0 - 70°C, V_{CC} = 5V ±5% unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
V_{IL}	Input Low Voltage	0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{IH}	Input High Voltage (Military)	2.2			V	T_A = -55°C to 125°C
VIHX1	Input High Voltage (X1/CLK)	4.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2.4 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -400µA
I_{IL}	Input Leakage Current	-25		25	µA	V_{IN} = 0 to V_{CC}
I_{ILSEL}	Select Pin Leakage Current	-30		+30	µA	V_{IN} = 0 to V_{CC}
I_{X1L}	X1 Input Low Current		-20		µA	V_{IN} = 0
I_{X2L}	X2 Input Low Current		-7		mA	
I_{XIH}	X1 Input High Current		20		µA	V_{IN} = V_{CC}
I_{X2H}	X2 Input High Current		20		µA	V_{IN} = V_{CC}
I_{LL}	Data bus Tri-State Leakage Current	-10		10	µA	VO = 0 to V_{CC}
I_{OC}	Open Drain Output Leakage Current	-10		10	µA	VO = 0 to V_{CC}
I_{CCA}	Power Supply Current[4]		6	15	mA	Active Mode
I_{CCS}	Power Supply Current[4]		3	10	mA	Standby Mode

Notes

1. Parameters are valid over the specified temperature and operating supply ranges. Typical values are 25°C, V_{CC} = 5V and typical processing parameters.

2. All voltages are referenced to ground (GND). For testing, input signal levels are 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V as appropriate. See Figure 31.

3. For prime grade N, P, J, L, M, ML, V_{CC} = 5V ± 10%

4. Measured operating with a 3.6864MHz crystal and with all outputs open.

AC ELECTRICAL CHARACTERISTICS [1, 2, 3]

Test Conditions: T_A = 0 - 70°C, V_{CC} = 5.0V ±5% unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
Rest Timing (See *Figure 32*)						
t_{RES}	RESET Pulse Width	1.0			µs	
Read, Write and Interrupt Cycle Timing (*Figure 33, Figure 34, Figure 35*)						
t_{AS}	A1-A4 Setup Time to $\overline{CS}$ Low	10			ns	
t_{AH}	A1-A4 Hold Time from $\overline{CS}$ High	0			ns	
t_{RWS}	$\overline{R/W}$ Setup Time to $\overline{CS}$ Low	0			ns	
t_{RWH}	$\overline{R/W}$ Setup Time from $\overline{CS}$ High	0			ns	
t_{CSW}	$\overline{CS}$ High Pulse Width [4, 5]	90			ns	
t_{CSD}	$\overline{CS}$ or $\overline{IACK}$ High from $\overline{DTACK}$ Low	20			ns	
t_{DD}	Data Valid from CS or $\overline{IACK}$ Low[6]			175	ns	
t_{DF}	Data Bus Floating from $\overline{CS}$ or $\overline{IACK}$ High	10		100	ns	
t_{DS}	Data Setup Time to $\overline{CS}$ Low	0			ns	
t_{DH}	Data Hold Time from $\overline{CS}$ Low	125			ns	
t_{DAL}	$\overline{DTACK}$ Low from Read Data Valid	0			ns	
t_{DAH}	$\overline{DTACK}$ High from $\overline{CS}$ or $\overline{IACK}$ High			100	ns	
t_{DAT}	$\overline{DTACK}$ High Impedance from $\overline{CS}$ or $\overline{IACK}$ High			125	ns	
t_{CSC}	$\overline{CS}$ or $\overline{IACK}$ set up time to “High”[7]	80			ns	
Port Timing (*Figure 36*)						
t_{PS}	Port Input Setup Time to $\overline{CS}$ Low or $\overline{R/W}$ High	0			ns	
t_{PH}	Port Input Hold Time from $\overline{CS}$ High	0			ns	
t_{PD}	Port Output Valid from $\overline{R/W}$, $\overline{CS}$ High			400	ns	
Interrupt Output Timing (*Figure 37*)						
t_{IR}	$\overline{INTR}$ or OP3-OP7 when used as Interrupts High from:			300	ns	
	Clear of Interrupts Status Bits in ISR or IPCR			300	ns	
	Clear of Interrupt Mask in IMR					
Clock Timing (*Figure 38*)						
t_{CLK}	X1/CLK (External) High or Low Time	100			ns	
t_{CLK}	X1/CLK Crystal or External Frequency	2.0	3.684	7.372	MHz	
t_{CTC}	Counter/Timer External Clock High or Low Time (IP2)	100			ns	
t_{CTC}	Counter/Timer External Clock Frequency	0		7.372	MHz	

AC ELECTRICAL CHARACTERISTICS [1, 2, 3] (CONT'D)

Test Conditions: T_A = 0 - 70°C, V_{CC} = 5.0V ±5% unless otherwise specified.

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
Clock Timing (*Figure 38*)						
t_{RTX}	RXCn and TXCn (External) High or Low Time [8]	220			ns	
f_{RTX}	RXCn and TXCn (External) Frequency					
	16X	0		16.0	MHz	
	1X	0		1.0	MHz	
Transmitter Timing (*Figure 39*)						
t_{TXD}	TXD Output Delay - TXC (External) Low			350	ns	
t_{TCS}	TXD Output Delay - TXC (Internal) Output Low			150	ns	
Receiver Timing (*Figure 40*)						
t_{RXS}	RXD Data Setup Time to RXC (External) High	240			ns	
t_{RXH}	RXD Data Hold Time from RXC (External) High	200			ns	

Notes

1. Parameters are valid over the specified temperature and operating supply ranges. Typical values are 25°C, V_{CC} = 5V and typical processing parameters.

2. All voltages are referenced to ground (GND). For testing, input signal levels are 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V as appropriate. See Figure 31.

3. AC test conditions for outputs: CL = 50pF, RL = 2.7k• to V_{CC}.

4. Consecutive write operations to the same register require at least three edges of the X1 clock between writes.

5. This specification imposes a 6 MHz maximum 68000 clock frequency if a read or write cycle follows immediately after the previous read or write cycle. A higher 68000 clock can be used if this is not the case.

6. This specification imposes a lower bound on $\overline{CS}$ and $\overline{IACK}$ low, guaranteeing that they will be low for at least one CLK period.

7. This parameter is specified only to insure $\overline{DTACK}$ is asserted with respect to the rising edge of X1/CLK as shown in the timing diagram, not to guarantee operation of the part. If the specified setup time is violated, $\overline{DTACK}$ may be asserted as shown or may be asserted one clock cycle later.

8. The minimum high time must be at least 1.5 times the X1/CLK period and the minimum low time must be at least equal to the X1/CLK period if either channel's Receiver is operating in external 1X clock mode.

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS[1]

DC Supply Voltage 7V
Storage Temperature . . . . . -65°C to 150°C
All Voltages with respect to Ground[2] . . . -0.5V to +7V

1. Stresses above those listed under the 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 conditions above those indicated in the "Electrical Characteristics" section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

2. 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 voltage larger than the rated maximum.

SYSTEM DESCRIPTION

The XR68C681 consists of two independent, full-duplex communication channels; each consisting of their own Transmitter and Receiver. Each channel of the DUART may be independently programmed for operating mode and data format. The DUART is designed to interface with the 68000 Family of microprocessors with minimal external components. The operating speed of each receiver and transmitter may be selected from one of 23 internally generated fixed bit rates, from a clock derived from an internal Counter/Timer, or from an externally supplied 1x or 16x clock. The bit rate generator (the source of the 23 different fixed bit rates) can operate directly from a crystal connected across two pins or from an external clock. The ability to independently program the operating speed of the receiver and transmitter of each channel makes the DUART attractive for split speed channel applications such as clustered terminal systems.

Receiver data is quadrupled-buffered and the Transmitter data is dual-buffered via on-chip FIFOs in order to minimize the risk of receiver overrun and to reduce overhead in interrupt driven applications. The DUART also provides a flow control capability to inhibit transmission from a remote device when the buffer of the receiving DUART is full, thus preventing loss of data.

The DUART also provides a general purpose 16 bit Counter/Timer (which may also be used as programmable bit rate generators), a multi-purpose 6 bit input port and a multi-purpose 8 bit output ports.

PRINCIPLES OF OPERATION

Figure 1 presents an overall block diagram of the 68C681 DUART. As illustrated in the block diagram, the DUART consists of the following major functional blocks:

- Data Bus Buffer
- Interrupt Control
- Input Port
- Serial Communications Channels A and B
- Operation Control
- Timing
- Output Port

A. DATA BUS BUFFER

The data bus buffer provides the interface between the internal (within the chip) and external data buses. It is controlled by the operation control block to allow data transfers to take place between the host CPU and the DUART.

B. OPERATION CONTROL BLOCK

The control logic of the operation control block receives operating commands from the CPU and generates proper signals to the various sections of the DUART. The operation control block functions as the user interface to the rest of the device. Specifically, it is responsible for DUART register address decoding, and command decoding. Therefore all commands to set baud rates, parity, other communication protocol parameters, start or stop the Counter/Timer or reading a “status register” to monitor data communication performance are processed via the operation control block.

The operation control block will control DUART performance based upon the following input signals:

- Address (Register Select) bits: A1 - A4
- $\overline{R/W}$ Input
- $\overline{CS}$ Input
- $\overline{RESET}$

The DUART includes a Data Transfer Acknowledge ($\overline{DTACK}$) output which is asserted during data transfer cycles in order to inform the CPU that the requested operation has been completed. An asserted $\overline{DTACK}$ signal indicates, to the CPU, that the input data has been latched, by the DUART data bus buffer, during a write cycle; that the requested data (from the DUART) is on the data bus and is valid during a read cycle, or that the interrupt vector is on the data bus during an interrupt acknowledge cycle.

When using the 6800 family processor, the XR-88C681 DUART data bus buffer should be used in lieu of this device. For information on how to interface a 6800 Family Processor to the XR-88C681 device, please see the XR-88C681 data sheet.

B.1 DUART REGISTER ADDRESSING

The addressing of the internal registers of the DUART is presented in *Table 1*. Please note that some of the registers are “Read Only” and others are “Write Only”. Each channel is provided with the following dedicated (addressable) registers.

- Command Register
- Mode Registers (MR1 and MR2)
- Status Register
- Clock Select Register
- Receiver Holding Register (RHR) and Transmit Holding Register (THR)

Additionally, the DUART contains the following registers that support/control both channel pairs.

- Interrupt Status Register (ISR)
- Interrupt Mask Register (IMR)
- Masked Interrupt Status Register (MISR)
- Interrupt Vector Register (IVR)
- Auxiliary Control Register (ACR)

Finally, the DUART contains additional registers that support functions other than serial data communication, such as the parallel ports and the counters/timers.

- Output Port Control Register (OPCR)
- Input Port Configuration Register (IPCR)
- Counter/Timer Upper Byte Register (CTUR)
- Counter/Timer Lower Byte Register (CTLR)
- Output Port Register (OPR)

Address (HEX)	Read Mode Registers		Write Mode Registers	
	Register Name	Symbol	Register Name	Symbol
00	Mode Register, channel A	MR1A, MR2A	Mode Register, channel A	MR1A, MR2A
01	Status Register, channel A	SRA	Clock Select Register, channel A	CSRA
02	Masked Interrupt Status Register	MISR	Command Register, channel A	CRA
03	Rx Holding Register, channel A	RHRA	Tx Holding Register, channel A	THRA
04	Input Port Change Register	IPCR	Auxiliary Control Register	ACR
05	Interrupt Status Register	ISR	Interrupt Mask Register	IMR
06	Counter/Timer Upper Byte Register	CTU	Counter/Timer Upper Byte Register	CTU
07	Counter/Timer Lower Byte Register	CTL	Counter/Timer Lower Byte Register	CTL
08	Mode Register, channel B	MR1B, MR2B	Mode Register, channel B	MR1B, MR2B
09	Status Register, channel B	SRB	Clock Select Register, channel B	CSRB
0A	RESERVED		Command Register, channel B	CRB
0B	Rx Holding Register, channel B	RHRB	Tx Holding Register, channel B	THRB
0C	Interrupt Vector Register	IVR	Interrupt Vector Register	IVR
0D	Input Port	IP	Output Port Configuration Register (OP0 - OP7)	OPCR
0E	Start Counter/Timer Command	SCC	Set Output Port Bits Command	SOPBC
0F	Stop Counter/Timer Command	STC	Clear Output Port Bits 1 Command	COPBC

Table 1. DUART Port And Register Addressing

Note:
The shaded blocks are not Read/Write registers but rather, "Address-Triggered" Commands.

Table 1 indicates that each channel is equipped with two "Mode Registers". Associated with each of these "Mode Register" pairs is a "Mode Register" pointer or MR pointer. Upon chip/system power up or RESET each MR pointer is "pointing to" the channel MR1n register. (Please note that the suffix "n" is used at the end of many of these register symbols in order to refer, generically, to either channels A or B). However, the contents of the MR pointer will shift from the address of the MR1n register to that of the MR2n register, immediately following any read or write access to the MR1n register. The MR pointer will continue to "point to" the MR2n register until a hardware reset occurs or until a "RESET MR POINTER" command has been invoked. The "RESET MR POINTER" command can be issued by writing the appropriate data to the appropriate channel's command register. Therefore, both mode registers, within a given channel, have the same logical address. The features and functions of the DUART that are controlled by the mode registers are discussed in detail in *Section G.3*.

B.2 COMMAND DECODING

Each channel is equipped with a command register. In general, the role of these command registers are to enable/disable the transmitter, enable/disable the receiver, along with facilitating a series of other miscellaneous commands. The bit format for each command register is presented below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Miscellaneous Commands				**Enable/Disable Transmitter**		**Enable/Disable Receiver**	
See Following Text				00 = No Change 01 = Enable Tx 10 = Disable Tx 11 = Not Valid (Do not use)		00 = No Change 01 = Enable Rx 10 = Disable Rx 11 = Not valid (do not use)	

Table 2. Command Register - CRA, CRB

The function of the lower nibble of the command registers is fairly straight-forward. This nibble is used to either enable or disable the transmitter and/or receiver.

The upper nibble of the command register is used to invoke a series of miscellaneous commands. *Table 3*

defines the commands associated with the upper nibble of the command registers. Please note that the upper nibble commands 1_{16} through B_{16} effects only the performance of command register's channel. However, commands C_{16} through F_{16} effects system (or chip) level operation.

Bit 7	Bit 6	Bit 5	Bit 4	Description
0	0	0	0	**Null Command:**
0	0	0	1	**Reset MRn Pointer:** Causes the channel's MRn pointer to point to MR1n.
0	0	1	0	**Reset Receiver:** Resets the individual channel receiver as if a hardware reset has been applied. The Receiver is disabled and the FIFO is flushed.
0	0	1	1	**Reset Transmitte**r: Resets the individual channel transmitter as if a hardware reset had been applied. The TXDn output is forced to a high level.
0	1	0	0	**Reset Error Status:** Clears the Received Break (RB), Parity Error (PE), Framing Error (FE) and Overrun Error (OE) status bits, SR[7:3]. Specifically, if the error mode, for a particular channel is set at "Block" error mode, this command will reset the all of the receiver error indicators in the status register. In the block error mode, once either a PE, FE, OE or RB occurs, this error will continue to be flagged in the channel status register, until this command is issued. If the Error Mode, for a particular channel is set at "Character" Error Mode, then the contents of the Status Register for PE, FE and RB are reflected on a character by character basic. In the "Character" Error Mode, the state of these indicators is based only upon the character that is at the top of the RHR. The OE indicator is always flagged as a "Block" Error Mode indicator, and requires this command to be reset.
0	1	0	1	**Reset Break Change Interrupt:** Clears the channel's break change interrupt status bit.
0	1	1	0	**Start Break:** Forces the TXDn output low. The transmitter must be enabled to start a break. If the transmitter is empty, the start of the break may be delayed up to two bit times. If the transmitter is active, the break begins when the transmission of those characters in the THR is completed, viz., TXEMP must be true before the break will begin.
0	1	1	1	**Stop Break:** The TXDn line will go high within two bit times. TXDn will remain high for one bit time before the next character, if any, is transmitted.
1	0	0	0	**Set Rx BRG Select Extend Bit:** Sets the channel's "Receiver BRG Select Extend Bit" to "1". (e.g. x=1)
1	0	0	1	**Clear Rx BRG Select Extend Bit:** Clears the channel's "Receiver BRG Select Extend Bit". (e.g. x=0)
1	0	1	0	**Set Tx BRG Select Extend Bit:** Sets the channel's "Transmitter BRG Select Extend Bit" to "1".
1	0	1	1	**Clear Tx BRG Select Extend Bit:** Clears the the channel's "Transmitter BRG Select Extend Bit" to "1".
1	1	0	0	**Set Standby Mode (Channel A):** When this command is invoked via the channel A command register, power is removed from each of the transmitters, receivers, Counter/Timer and additional circuits to place the DUART in the standby (or lower power) mode. Please note that this command effects the operation of the entire chip. Normal operation is restored by a hardware reset or by invoking the "SET ACTIVE MODE" command.
1	1	0	1	**Set Active Mode (Channel A):** When this command is invoked via the channel A command register, the DUART is removed from the standby mode and resumes normal operation.
1	1	1	0	**Reserved**
1	1	1	1	**Reserved**

Table 3. Miscellaneous Commands, Upper Nibble of all Command Registers, Unless Otherwise Specified.

In addition to the commands which are available through the command registers, the DUART also offers "Address-Triggered" commands. These commands are listed in *Table 1*, DUART PORT AND REGISTER ADDRESSING"; and are further identified by being "shaded" in . Specifically, these commands are:

- START COUNTER/TIMER COMMAND
- STOP COUNTER/TIMER COMMAND
- SET OUTPUT PORT BITS COMMAND
- CLEAR OUTPUT PORT BITS COMMAND

Each of these commands are invoked by either reading or writing data to their corresponding DUART addresses as specified in *Table 1*.

For example, the START COUNTER/TIMER COMMAND is invoked by the procedure of reading DUART address $0E_{16}$. Please note that this "Read Operation" will not result in placing the contents of a DUART register on the data bus. The only thing that will happen, in response to this procedure; is that the Counter/Timer will initiate counting. For a detailed discussion into the operation of the Counter/Timers, please see *Section D.2*.

Another example of an "Address-Triggered" commands is the "SET OUTPUT PORT BITS 1" Command. This command is invoked by performing a write of data to DUART address $0E_{16}$. When the user invokes this command , he/she is setting certain bits (to "1") within the OPR (Output Port Register). All other bits, within the OPR (not specified to be set), are not changed.

The state of the output port pins are complements of the individual bits within the OPR. Hence, if OPR[0] is set to "1", the state of the corresponding output port pin, OP0, is now set to a logic "0". Consequently, one can think of the "SET OUTPUT PORT BITS" command as the "CLEAR OUTPUT PORT PINS" command. For a more detailed discussion into the operation of the output ports, please see *Section F*.

C. INTERRUPT CONTROL BLOCK

The interrupt control block allows the user to apply the DUART in an "Interrupt Driven" environment. The DUART includes an Interrupt Request output signal ($\overline{INTR}$) is provided which may be programmed to be asserted upon the occurrence of any of the following events:

- Transmit Hold Register A or B ready
- Receive Hold Register A or B ready
- Receive FIFO A or B Full
- Start or End of Received Break in Channels A or B
- End of Counter/Timer Count Reached
- Change of State on input pins, IP0, IP1, IP2, IP3

The interrupt control block consists of an Interrupt Status Register (ISR), an Interrupt Mask Register (IMR), an Masked Interrupt Status Register (MISR) and an Interrupt Vector Register (IVR). *Table 4* lists these registers, and their address location (within the DUART).

Register	Description	Address Location (in DUART Address Space)
ISR	Interrupt Status Register	05_{16} (Read Only)
IMR	Interrupt Mask Register	05_{16}(Write Only)
MISR	Masked Interrupt Status Register	02_{16}(Read Only)
IVR	Interrupt Vector Register	$0C_{16}$

Table 4. Listing and Brief Description of Interrupt Control Block Registers

The role and purpose of each of these registers are defined as follows:

C.1 Interrupt Status Registers (ISR)

The contents of the ISR indicates the status of all potential interrupt conditions. If any bits within these registers are toggled "high", then the corresponding condition has or is occurring. In general, the contents of the ISR will indicate to the processor, the source or the reason for the interrupt request from the DUART. The bit-format of the ISR register is presented as follows.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input Port Change	Delta Break B	RXRDY/ FFULLB	TXRDYB	Counter Ready	Delta Break A	RXRDY/ FFULLA	TXRDYA
0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes

Table 5. Interrupt Status Register - (ISR) Bit Format

The meaning behind each of these bits is defined below.

ISR[7]: Input Port Change of State:

If this bit is at a logic "1", then a "change of state" was detected at the IP0 - IP3 pins. The user would service this interrupt by reading the IPCR (if ISR[7] = 1). ISR[7] is cleared when the CPU has read the IPCR. By reading the IPCR, the user will determine:

- The individual Input Port pin that changed state
- The final state of the monitored input ports, following the Change of State.

For a detailed description of the IPCR, please see *Section E*.

Please note that in order to enable this interrupt condition, the user must do two things:

1. Write the appropriate data to the lower nibble of the Auxiliary Control Register, ACR[3:0]. In this step, the user is specifying which input pins should trigger an "Input Port Change" interrupt request.
2. Write a logic "1" to IMR[7].

ISR[6] Delta Break Indicator - Channel B:

When this bit is set, it indicates that the channel B receiver has detected the beginning or end of a received break. This bit is cleared (or reset) when the CPU invokes a channel B "RESET BREAK CHANGE INTERRUPT" command. For more information into the DUART's response to a BREAK condition, please see *Section G.2*.

ISR[5] RXRDYB/FFULLB - Channel B Receiver Ready or FIFO Full

The function of this bit is selected by programming MR1B[6]. If programmed as the Receiver Ready indicator (RXRDYB), it indicates that at least one character of data is in the RHRB and is ready to be read by the CPU. This bit is set when a character is transferred from the received shift register to RHRB and is cleared when the CPU reads RHRB. If there are still more characters in RHRB after the read operation, the bit will be set again after RHRB is "popped".

If this bit is programmed as FIFO Full indicator (FFULLB), it is set when a character is transferred from the RSR to RHRB and the transfer causes RHRB to become full. This bit is cleared when the CPU reads RHRB; and thereby "popping" the FIFO, making room for the next character. If a character is waiting in the RSR because RHRB is full, this bit will be set again after the read operation, when that character is loaded into RHRB.

ISR[4] TXRDYB - Channel B Transmitter Ready

This bit, when set, indicates that THRB is empty and is ready to accept a character from the CPU. The bit is cleared when the CPU writes a new character to THRB; and is set again, when that character is transferred to the TSR. TXRDY is set when the transmitter is initially enabled and is cleared when the transmitter is disabled. Characters loaded into THRB while the transmitter is disabled will not be transmitted.

ISR[3] Counter Ready

In the TIMER mode, the C/T (Counter/Timer) will set ISR[3] once each cycle of the resultant square wave (available at the OP3 pin). ISR[3] will be cleared by invoking the "STOP COUNTER" command. Bear in mind, that in the TIMER mode, the "STOP COUNTER" command will not stop the C/T.

In the COUNTER mode, this bit is set when the C/T reaches the terminal count $(0000)_{16}$ and is cleared when the C/T is stopped by a "STOP COUNTER" command. When the Counter/Timer is in the COUNTER mode, this command will stop the counter.

ISR[2]: Delta Break Indicator - Channel A

Assertion of this bit indicates that the channel A receiver has detected the beginning or end of a Received Break (RB). This bit is cleared when the CPU invokes a channel A "RESET BREAK CHANGE INTERRUPT" command. For more information into the DUART's response to a BREAK condition, please see *Section G.2*.

ISR[1] RXRDYA/FFULLA - Channel A Receiver Ready or FIFO Full

The function of this bit is selected by programming MR1A[6]. If programmed as the Receiver Ready indicator (RXRDYA), this bit indicates that there is at least one character of data in RHRA, and is ready to be read by the CPU. This bit is set when a character is transferred from the RSR to RHRA and is cleared when the CPU reads (or "pops") RHRA. If there are still more characters in RHRA after the read operation, the bit will be set again after RHRA is "popped".

If this bit is programmed as the FIFO (RHR) full indicator (FFULLA), it is set when a character is transferred from the RSR to RHRA and the newly transferred character causes RHRA to become full. It is cleared when the CPU reads RHRA. If a character is waiting in the RSR because RHRA is full, this bit will be set again after the read operation, when that character is loaded into RHRA.

ISR[0]: Channel A Transmitter Ready

This bit, when set, indicates that THRA is empty and is ready to accept a character from the CPU. The bit is cleared when the CPU writes a new character to THRA; and is set again, when that character is transferred to the TSR. TXRDY is set when the transmitter is initially enabled and is cleared when the transmitter is disabled. Characters loaded into THRA while the transmitter is disabled will not be transmitted.

C.2 Interrupt Mask Register (IMR)

The interrupt mask register is a "Write Only" register which enables the user to select the conditions that will cause the DUART to issue an interrupt request to the processor. In other words, the use has the option of masking or blocking certain conditions from causing the DUART to issue an interrupt request. Therefore, the bit-format of the IMR is essentially the same as the ISR. However, for completeness, the bit format of the IMR is presented in the following table.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input Port Change	Delta Break B	RXRDY/ FFULLB	TXRDYB	Counter Ready	Delta Break A	RXRDY/ FFULLA	TXRDYA
0 = Off	0 = Off	0 = Off	0 = Off	0 = Off	0 = Off	0 = Off	0 = Off
1 = On	1 = On	1 = On	1 = On	1 = On	1 = On	1 = On	1 = On

Table 6. IMR Bit Format

If the user wishes to enable a certain interrupt, he/she should write a "1" to the bit, within the IMR, corresponding to that interrupt condition. Likewise, to disable or mask out a certain condition causing an interrupt, the user should write a "0" to the bit location corresponding to that condition. To enable all interrupts the user would write FFh (all "1"s) to this register.

Please note that IMR is a "write-only" register, and therefore, cannot be read by the processor.

C.3 Masked Interrupt Status Register (MISR)

The contents of the MISR register is basically the results of ANDing the ISR and IMR together.

MISR Content = [ISR Contents]• [IMR Contents]

One limitation of interrupt routines that rely on reading the ISR is that the bits within the ISR can toggle "high" due to their corresponding conditions whether or not they are enabled by the IMR. Therefore, the user, following reading the interrupt status register, will have to make provisions for; and execute a "bit-by-bit" AND of the ISR and IMR contents. Since the IMR is a "Write Only" register and cannot be read by the processor, the contents of the IMR will have to be stored is system memory, for later recall. The additional hardware and software overhead required to suport this activity can be eliminated via use of the MISR.

C.4 Interrupt Vector Register, IVR

The 68000 family of microprocessors supports vectored-interrupt processing. Specifically, during interrupt servicing, the DUART will respond to the interrupt acknowledge signal, from the CPU, by placing the contents of the IVR on the data bus, to be read by the CPU. During normal operation, the contents of the IVR is related to a location in memory, where the appropriate interrupt service routine (for the interrupting DUART) resides.

Therefore, in vectored interrupt applications, the contents of the IVR accomplish two things:

1. Identify the peripheral components requesting the interrupt.
2. Allow the CPU to determine the location of; and branch program control to the location, in program memory, that contains the appropriate interrupt service routine for the interrupting DUART.

Consequently, during initialization of the DUART, the user will have to load the IVR with a hexadecimal numbers of values between 40_{16} $(64)_{10}$ through FF_{16} (255_{10}), inclusively. This is the range of the values, in the 680x0's exception vector table, that have been reserved for "User Interrupt Vector". The memory location of the "DUART" interrupt service routine can be found by multiplying the contents of the IVR by 4. Hence, the user should take care to make sure that the interrupt service routine starts at [Contents of IVR] • 4 in program memory.

The XR68C681, like many other 68000-series peripheral devices are designed such that the default contents of their IVR (following a RESET condition) is $0F_{16}$. Consequently, if, during an "interrupt acknowledge" cycle (see the next section) the CPU reads the value $0F_{16}$ from the DUART; and "Uninitialized Interrupt Vector" exception will be generated.

C.5 Limitations of the DUART Interrupt Structure

The interrupt structure offered by the DUART allows the user to program the DUART to generate interrupts in response to certain THR and RHR (FIFO) conditions; the "Counter/Timer Ready" condition, and to changes in the break condition (at the Receiver). However, aside from the "Delta Break Condition", the DUART's interrupt structure does not allow for interrupt requests due to receiver problems such as Parity Error (PE), receiver Overrun Error (OE), or Framing Error (FE). The DUART also does not offer the user the ability to configure one of the output ports to relay the occurrence of any of these conditions. The user is, therefore, recommended to "validate" the receive data by frequently reading the status register; and checking for any non-zero upper nibble values. This is especially the case if the user has set the error mode to "Character" (MR1n[5] = 0).

C.6 Servicing DUART Interrupts

The 68000 family of microprocessors supports vectored-interrupt processing. In vectored-interrupt processing, the peripheral device, responsible for requesting the interrupt, will identify itself to the microprocessor, during the "Interrupt Acknowledge" cycle. Once the microprocessor knows which peripheral device is requesting the interrupt, the microprocessor will determine the location of the appropriate interrupt service routine in memory, and branch program control to that location.

The advantage of using "Vectored-Interrupt" processing over "Polled-Interrupt" processing is significant in

time-critical applications using many peripherals devices. In “Polled-Interrupt” processing, upon the detection of the interrupt request, the microprocessor will have to go through and poll each and every peripheral device in order to determine the device causing the interrupt. Only after this polling procedure is completed can the microprocessor branch program control to the appropriate interrupt service routine. The time required to poll each of these peripheral devices adds to the interrupt latency period over and above that which would occur during vectored-interrupt processing.

Figure 2 presents a simple illustration of how to interface the DUART to a 68000 processor for interrupt service considerations.

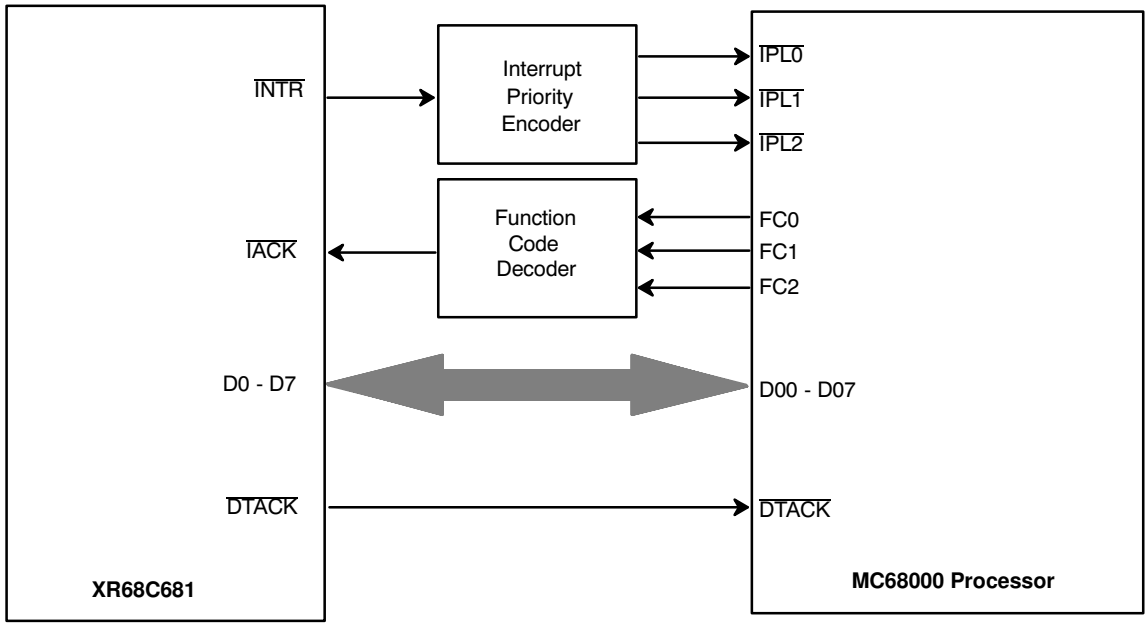


Figure 2. Simple Illustration Depicting the Interfacing of the XR68C681 DUART to a 68000 Processor

Figure 3 presents a more detailed schematic of the XR68C681 device interfacing to a 68000 microprocessor. This figure shows only the interrupt processing portion of the microprocessor/DUART interface. The address decoding circuitry for address bus bits A08 - A23 is not included. This circuit consists of an "Interrupt Priority Encoder" (SN74LS148), and two 3-line-to-8-line decoders (SN74LS138). For discussion purposes, one of these SN74LS138 devices are labeled "$\overline{\text{IACK}}$ Decoder" and the other is labeled "I/O Chip Select Decoder". In this figure, the DUART has an interrupt priority level of 4 (100_2).

A functional description of this circuit follows. If the DUART requires service from the CPU, it will assert the active-low, open-drain, output signal, $\overline{\text{INTR}}$. When this signal toggles "low" the interrupt priority encoder (SN74LS148) will generate the appropriate interrupt priority level and present this priority level to the CPU. In this case, the interrupt priority level is 4 ($\overline{\text{IPL2}}$ = 0, $\overline{\text{IPL1}}$ = 1, $\overline{\text{IPL0}}$ = 1). In response to the priority level 4 interrupt request, the CPU will check the interrupt mask bits (of its own internal status register) in order to determine the present interrupt priority level. If the present interrupt priority level is 4 or less; the CPU will acknowledge and begin service of this new DUART interrupt request. If the present interrupt priority level is 5 or greater, the DUART's interrupt request will not be serviced until completion of all higher priority interrupts. Once the microprocessor decides to service this particular interrupt request, it will do so by asserting all of the Function Code outputs (FC2 = 1, FC1 = 1, FC0 = 1), in order to indicate that this next bus cycle will be an interrupt acknowledge cycle. Additionally, whenever the 68000 CPU is interrupted, it will output on address bits A_{01}, A_{02}, and A_{03}, the interrupt priority level, while address bits A_{04} - A_{23} are all set to the logic one level. Therefore, the CPU will acknowledge this DUART interrupt request by setting A_{01} = 0, A_{02} = 0, A_{03} = 1, and A_{04}- A_{23} = 1. Once all of the Function Code outputs are set, the NAND gate (74LS10) will assert one of the enable inputs of the "$\overline{\text{IACK}}$ Decoder". Additionally, the Address Strobe output ($\overline{\text{AS}}$) will soon be asserted in order to start the next bus cycle. Once it is asserted, the other enable input of the $\overline{\text{IACK}}$ decoder will also be asserted. When the two enable inputs are asserted, the $\overline{\text{IACK}}$ decoder will assert the output labeled "$\overline{\text{IACK4}}$, thereby asserting the $\overline{\text{IACK}}$ input of the DUART. In parallel with the $\overline{\text{IACK4}}$ signal being asserted, the address bits, A_{08} - A_{23}, are routed through an address decoder (not shown). However, if all of these address bits are at logic "1" level, the "I/O Chip Select Decoder" will also be enabled. In this figure, the output labeled $\overline{\text{CS0}}$ will be asserted, thereby asserting the $\overline{\text{CS}}$ input of the DUART. Please note that the DUART does not require that it's $\overline{\text{CS}}$ input be asserted in order to respond to an "Interrupt Acknowledge" cycle. The DUART only requires that its "$\overline{\text{IACK}}$" input be asserted.

In response to the assertion of the $\overline{\text{IACK}}$ input, the DUART will place the contents of the IVR (Interrupt Vector Register) on the data bus (D0 - D7), where it can be read by the CPU. Once the DUART has placed the contents of the IVR on the data bus, it will assert the $\overline{\text{DTACK}}$ output in order to inform the CPU that data is ready to be read from the data bus. The CPU will then execute this "read" in a manner identical with any other read cycle. Once this "read" cycle is completed, the CPU will negate $\overline{\text{AS}}$ output (thereby negating the $\overline{\text{IACK}}$ input of the DUART); and the DUART will, in turn, negate the $\overline{\text{DTACK}}$ output to the CPU. Once the $\overline{\text{DTACK}}$ output has been negated, the interrupt cycle is completed, and the next several read and write cycles will likely be dedicated to servicing the DUART interrupt.

If the user had properly initialized the IVR, with values ranging between 64 (40_{16}) and 255 (FF_{16}), the CPU will multiply this value by 4, in order to determine the location, in memory, of the DUART interrupt service routine. Afterwards this address location will be loaded into the program counter (of the CPU) and the CPU will branch program control to this location. Obviously, the user must ensure that the appropriate interrupt service routine exists at the location in system level memory. If the user has failed to initialize the IVR, its contents will be (by default) $0F_{16}$. The XR68C681 device, like many other 68000 series peripherals are designed to have the default value of their interrupt vector registers to be $0F_{16}$. If, during the interrupt cycle, the CPU reads $0F_{16}$ from the DUART IVR, the CPU will multiply this value by 4, and will branch program control to the "Uninitialized Interrupt Vector" exception service routine, located at $03C_{16}$ in memory.

When the CPU has properly serviced the interrupt and the condition(s) causing the interrupt request(s) from the DUART have been eliminated, the $\overline{\text{INTR}}$ output from the DUART will be negated.

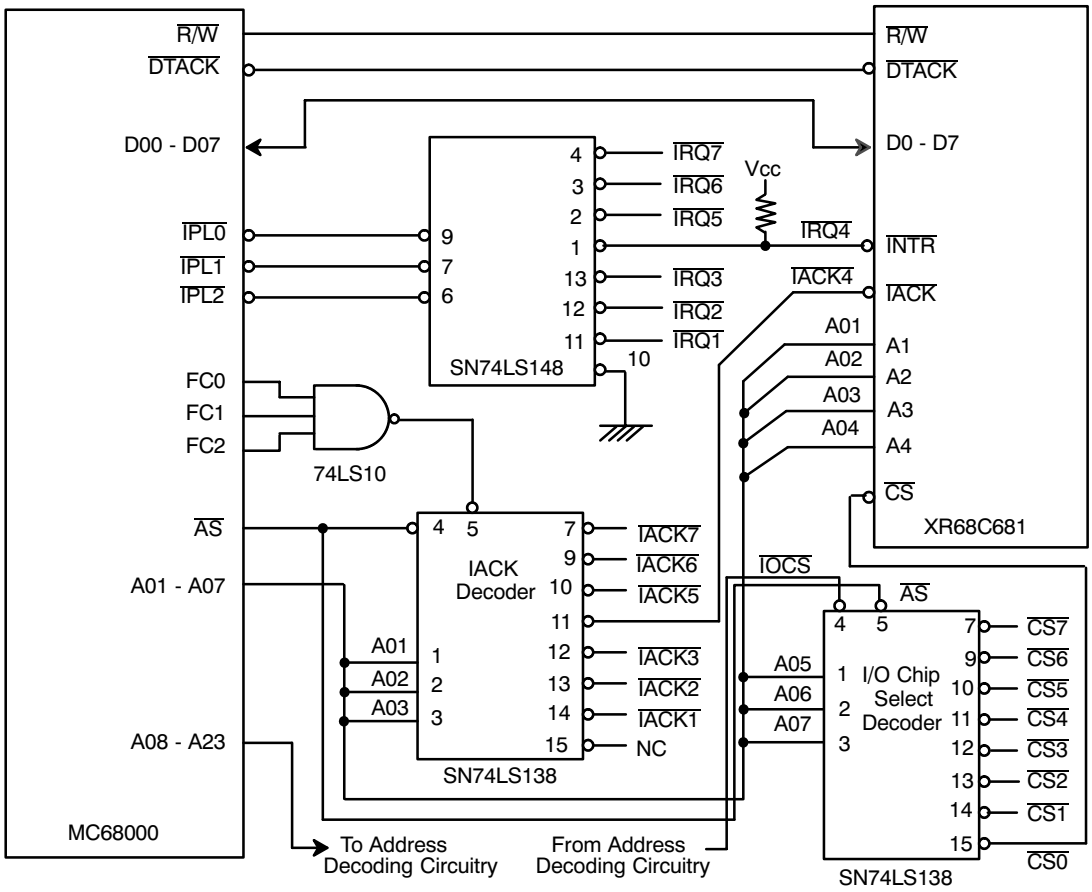


Figure 3. Detailed Schematics of the XR68C681 Interfacing to the 68000 Processor

Figure 4 presents a timing diagram depicting the sequence of events that will occur at the DUART/CPU interface, during an interrupt bus cycle.

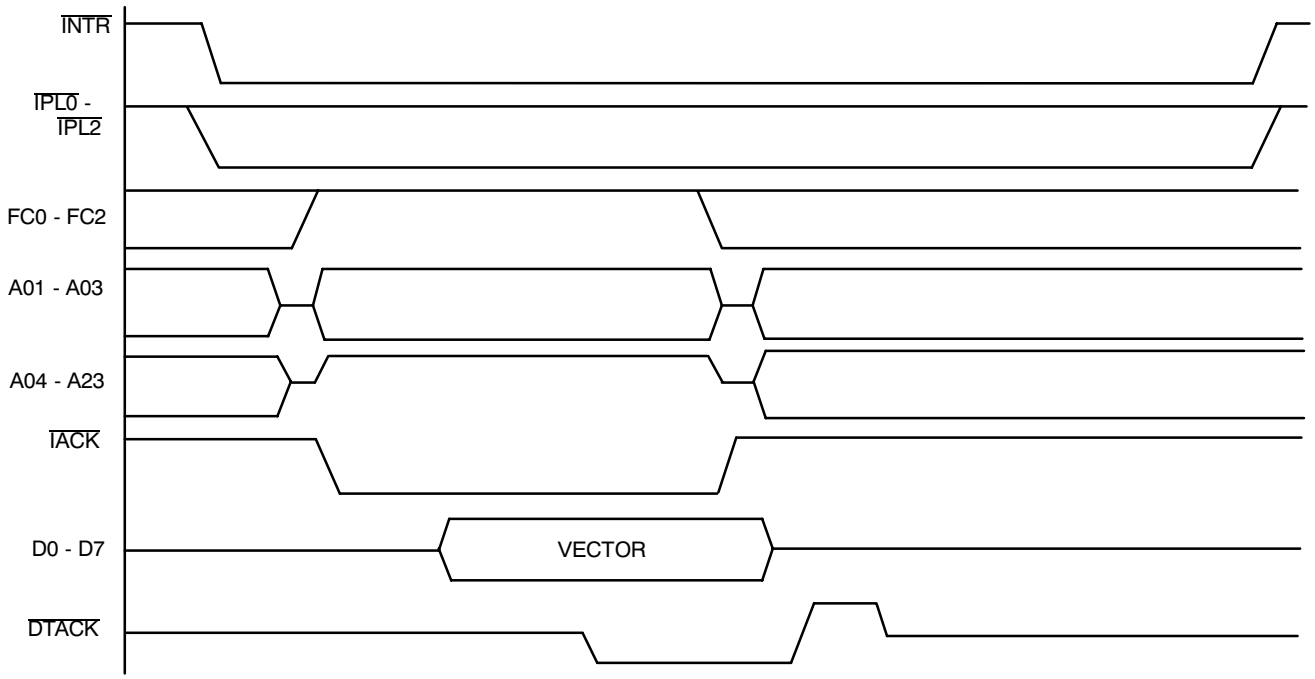


Figure 4. XR68C681/68000 CPU Interrupt Cycle Timing

Interrupt Service Routine

The objectives of the interrupt service routine are to :

1. Quickly identify the condition causing the interrupt request.
2. Quickly service the interrupt by eliminating the condition causing the Interrupt.

In order to identify the cause of the interrupt, the CPU must read the interrupt status register (or Masked Interrupts Status Register) of the DUART. The contents of the ISR identifies the condition(s) causing the interrupt request. *Section C.1* defines the bit format of the ISR and discusses how to clear each of the bits within the ISR.

D. TIMING CONTROL BLOCK

The timing control block allows the user to specify the bit rates that he/she wishes to transmit and receive data at each channel. The timing control block consists of the following elements:

- Oscillator Circuit
- Bit Rate Generator
- 16 bit Counter/Timer
- 4 - External Input Pins (to clock the Transmitters and Receivers, directly)
- Two Clock Select Registers (32:1 MUXs)

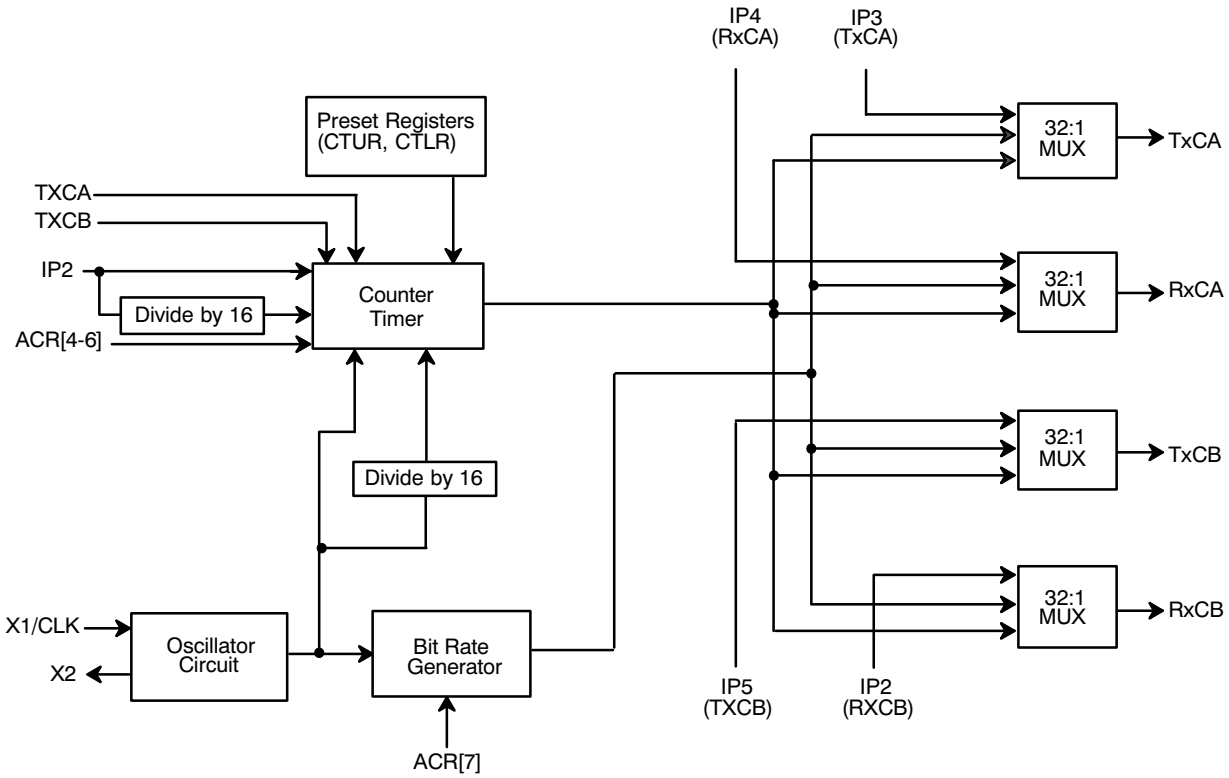


Figure 5. Block Diagram of DUART Timing Control Block

Figure 5 presents a block diagram of the timing control block for the XR68C681 device.

Each element of the timing control block is discussed in the following section.

D.1 Oscillator Circuit:

A crystal oscillator is typically connected externally across the X1/CLK and X2 pins. The oscillator circuit (within the chip) functions as the load for the resonant (crystal) oscillator, and buffers the resulting oscillating signal, for use by the bit rate generator, and Counter/Timer. A crystal or TTL signal frequency of between 2 MHz and 4 MHz is required for proper operation of the DUART. However, a crystal or TTL signal frequency of 3.6864 MHz is required for the generation of standard bit rates by the bit rate generator (See *Table 3*). *Figure 6* presents a recommended schematic for the XTAL oscillator circuitry. If the user desires to run numerous DUARTs from a single crystal oscillator, *Figure 7* presents an approach and the necessary circuitry to accomplish this objective.

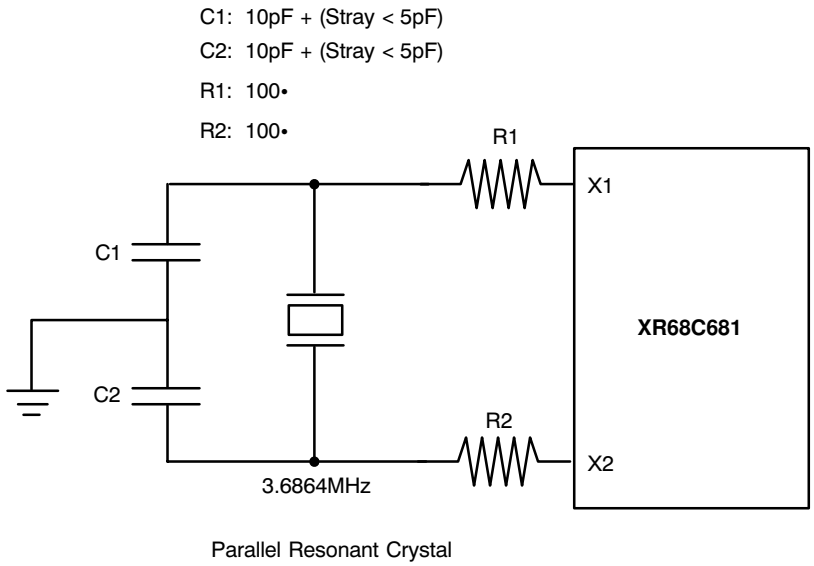


Figure 6. A Recommended Schematic for the XTAL Oscillator Circuitry

Note:
The user also has an option to drive the oscillator circuit with a TTL input signal, in lieu of using a crystal oscillator. If this approach is used, the TTL must be driven into the X1/CLK pin, and the X2 pin must be left floating.

If the user desires to run numerous DUARTs from a single crystal oscillator, *Figure 7* presents an approach and the necessary circuitry to accomplish this objective.

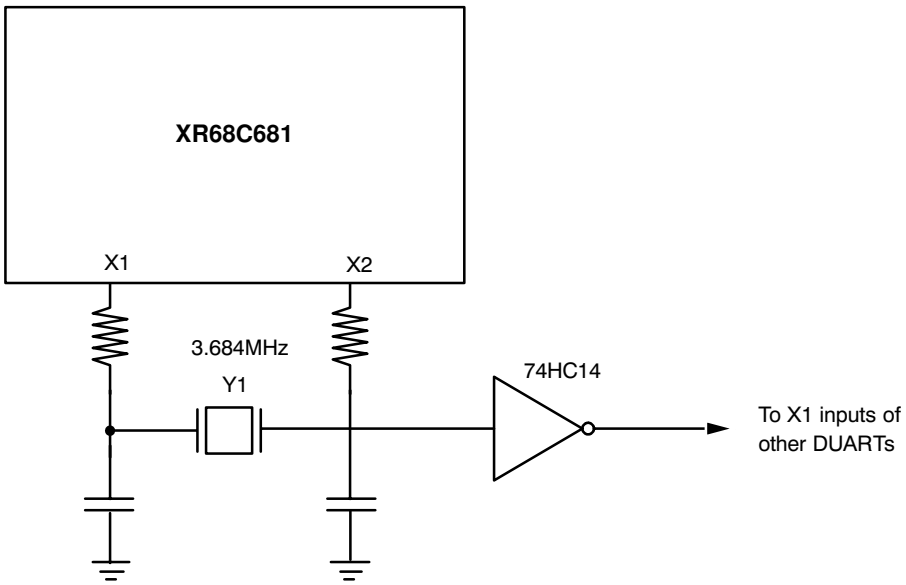


Figure 7. A Recommended Schematic to Drive Multiple DUARTs From the Same Crystal Oscillator