EXPERIMENT #8

SERIAL COMMUNICATION USING THE ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

1.0 Purpose

The purpose of this experiment is to introduce the student to the following topics:

- the Asynchronous Communications Interface Adapter IC (ACIA, MC6850)
- the fundamentals of serial asynchronous data communications
- the RS-232-C Serial Communications Standard

2.0 Component Requirements

None.

3.0 Background

A. The Asynchronous Communications Interface Adapter (ACIA)

The ACIA (MC6850) provides the data formatting and control to interface serial asynchronous data communications systems to parallel bus systems.

When the CPU writes data to the ACIA in a parallel format, the ACIA performs a parallel-to-serial conversion before transmitting the data serially. Similarly, when the ACIA receives data in a serial format, it performs a serial-to-parallel conversion, which enables the CPU to read the data in a parallel format.

The parallel bus (host CPU) interface of the ACIA consists of the following signals:

Quantity	Description
3	Chip Select lines (*CS2, CS1, CS0)
1	Register Select line (RS)
1	Read/Write line (R/*W)
1	Clock line (E)
1	Interrupt Request line (*IRQ)
8	Data lines (D0 to D7)

The serial communications interface of the ACIA consists of the following signals:

Quantity	Description
1	Transmit Data (TX DATA)
1	Receive Data (RX DATA)
1	Clear-to-Send (*CTS)
1	Request-to-Send (*RTS)
1	Data Carrier Detect (*DCD)
1	Transmit Data Clock (TX CLOCK)
1	Receive Data Clock (RX CLOCK)

The top five serial communications signals are identical to the ones used to implement an asynchronous version of the RS-232-C Serial Communications Standard.

Registers

The ACIA has four internal registers that are accessible by the CPU. The selection of a certain register depends on the state of the Register Select (RS) and Read/*Write (R/*W) lines. The following chart illustrates how each register is selected.

RS	R/*W	Register Descriptions
0	0	Control Register (CR)
0	1	Status Register (SR)
1	0	Transmit Data Register (TDR)
1	1	Receive Data Register (RDR)

Note: 0 or 1 indicates the logic level of the signal.

Notice that the SR and the RDR are read-only registers, and the CR and the TDR are write-only registers.

The preceding discussion is meant to serve as a brief introduction to the ACIA. For more detailed information about this device refer to the course lecture notes and the following textbooks (included as ACIA&PIA.pdf in the ECE441 File Package):

a. Bishop, Ron. Basic Microprocessors and the 6800.
Rochelle Park, NJ: Hayden Book Company, 1979.
(Note: refer specifically to Chapter 9, section 9.6)

b. 8-Bit Microprocessor & Peripheral Data Book.
Series C. Austin, Texas: Motorola, Inc. 1983

B. Asynchronous Serial Communication

In serial communication systems, a byte of data is transmitted one bit at a time along the same physical wire. In asynchronous serial communications systems, "start" and "stop" bits are added before and after the data to inform the receiving device as to where the data begins and ends. When transmitting serial data asynchronously, the data packet must adhere to the following specific format.

The first bit transmitted is the Start Bit, and this bit indicates the beginning of a character word. This bit is always a logic 0.

Next, 7 or 8 Data Bits are transmitted, one bit at a time, starting with the Least Significant Bit or LSB (i.e. D0), and increasing towards the Most Significant Bit or MSB (i.e. D7).

The Parity Bit is sent next. It can be either logic 0 or 1 depending upon the data and the type of parity selected in the ACIA's Control Register.

Finally the Stop Bits are transmitted. These bits indicate the end of a character word. There can be either 1 or 2 Stop Bits, and they are always logic 1.

Refer to Figures 9.23 and 9.24 which illustrates the format for serially transmitting a character.

The preceding discussion is meant to serve as a brief introduction to asynchronous communications. For more detailed information about this device refer to the course lecture notes and the following textbook:

Bishop, Ron. Basic Microprocessors and the 6800.
Rochelle Park, NJ: Hayden Book Company, 1979.
(Note: refer specifically to Chapter 9, section 9.6)

C. RS-232-C Serial Communications Standard

RS-232-C is the name given to the hardware standard for the serial transmission of data from one computer to another computer or peripheral device. The voltage levels for a logic 0 are +3 to +15 Volts, and for a logic 1 are -3 to -15 Volts. For ACIA #1 and ACIA #2, the RS-232 signals are made available on two 25-pin connectors (plug type DB-25 Connectors) at the back of the SANPER-1 Educational Lab Unit.

In the SANPER-1 Unit, the TTL serial data being transmitted by the ACIA is first inverted, and then converted to RS-232-C type voltages by an integrated circuit known as a "TTL to RS-232-C Converter" (Motorola Part No.: MC1488). This device converts the TTL signals (0 or +5 Volts) to RS-232 signals (-3 to -15 Volts or +3 to +15 Volts). The RS-232 data is then sent to the receiving computer or peripheral device.

When RS-232 data is received, it is inverted and then converted to TTL level voltages by an integrated circuit known as a "RS-232-C to TTL Converter" (Motorola Part No.

MC1489). This device converts the RS-232 signals (-3 to -15 Volts or +3 to +15 Volts) to TTL signals (0 or +5 Volts). The TTL data is then input to the ACIA on the "RX DATA" pin.

4.0 Statement of the Problem

This experiment consists of two parts. In the first part, the student will implement serial asynchronous communications using the ACIA. The student will use ACIAs to establish a full-duplex communication channel with another lab unit. The student will write software routines to initialize, control, and monitor the operation of the ACIA, which will communicate with the other lab unit.

In the second part of the experiment, the student will modify the program from Experiment #4 (Code Conversion and Bit Manipulation) by replacing the TRAP #14 routines that interface to the terminal (i.e. PORT1IN), with their own terminal handling routines. These routines will control the ACIA, which inputs and outputs data to or from the terminal. These routines will be implemented using TRAP #15.

5.0 Preliminary Assignment

General Note: In sections A and B below, the student may use TRAP #14 routines for inputting and outputting data to/from their terminal. In section C, the student is prohibited from using TRAP #14 routines for terminal I/O.

A. Polling Implementation of Unit-to-Unit Communication

1. Write an initialization subroutine for ACIA #2 on the SANPER-1 Educational Lab Unit. Configure ACIA #2 to operate as follows:

   - transmitter and receiver clocks set to divide-by-16 mode
   - 8 data bits
   - no parity bit
   - 1 stop bit
   - *RTS pin low
   - transmitter and receiver interrupts disabled

2. Write a subroutine to transmit an ASCII character. First, determine from the appropriate ACIA status flags whether the ACIA is ready to accept a character for transmission. If it is, take the ASCII character pointed to by Address Register A0, store it in ACIA #2's TDR register, and then exit the subroutine.

3. Write a subroutine to continually monitor the status flags of ACIA #2 to determine if it has received an ASCII character. Once a character is received, again examine the ACIA's status flags to determine if the character was received error free.

If no errors occurred, place the character in Data Register D0, then exit the subroutine.

If an error occurred, print a message on the terminal indicating which type of error occurred and the value of the received data. Exit the subroutine.

4. Write a subroutine that calls the subroutine of Prelim #3 and then displays the received character on the terminal.

5. Write a subroutine to prompt the user to enter ten ASCII characters at the terminal. An example of the input format is: ABCDE12345 <CR>. The routine will then input these characters from the terminal, and then store them in a table in memory (locations $900 to $909). When the table is full, transmit each character out of ACIA #2 using the subroutine of Prelim #2, then exit the subroutine.

6. Write a subroutine to receive ASCII characters from another lab unit (via ACIA #2) using the subroutine of Prelim #3. As each character is received, its ASCII code should be stored in a table in memory (locations $910 to $919). When the table is full, display the received characters on the terminal, then exit this subroutine.

B. <u>Interrupt Implementation of Unit-to-Unit Communications</u>

7. Write an initialization subroutine for ACIA #2. Configure the ACIA to operate as follows:

   - transmitter and receiver clocks set to divide-by-16 mode
   - 8 data bits
   - no parity bit
   - 1 stop bit
   - *RTS pin low
   - transmitter interrupts disabled
   - receiver interrupts enabled

8. Write an Interrupt Service Routine that queries ACIA #2 to determine the cause of the interrupt request.

   If the receiver section caused the interrupt, examine the error condition bits of the Status Register.

   If no receiver errors occurred, read the data from the RDR and store it into a 10 byte long table (locations $910 to $919) using Address Register A0 as the pointer into the table. After storing the data, determine the condition of the table. If the table is full, display its contents on the terminal, reinitialize the pointer to the

starting address of the table, and then exit the routine. If the table is not full, increment the pointer, then exit the interrupt service routine.

If an error occurred, print a message on the terminal indicating which type of error occurred and the value of the received data. Exit the service routine.

Note: If the interrupt was generated by some condition other than received data, disregard the interrupt.

9. Assemble each of the subroutines created above into the following program format.

```
PROC1       MOVE.L #$TBD,A7     * TRANSMIT ONLY routine
            JSR SUBRT1          * ACIA init. - polling
LOOP1       JSR SUBRT2          * transmit a character.
            BRA LOOP1           * transmit continuously.

PROC2       MOVE.L #$TBD,A7     * RECEIVE ONLY routine
            JSR SUBRT1          * ACIA init. - polling
LOOP2       JSR SUBRT4          * display received char
            BRA LOOP2           * receive continuously

PROC3       MOVE.L #$TBD,A7     * TX and RX - POLLING
            JSR SUBRT1          * ACIA init. - polling
LOOP3       JSR SUBRT5          * transmit char block
            JSR SUBRT6          * receive char block.
            BRA LOOP3

PROC4       MOVE.L #$TBD,A7     * TX and RX - INTERRUPTS
            JSR SUBRT7          * ACIA init. - interrupts
LOOP4       JSR SUBRT5          * transmit char block
            BRA LOOP4
```

Notes:

a. The label "SUBRT1" is the name of the subroutine from Prelim #1, label "SUBRT2" is the name of the subroutine from Prelim #2, and so on.
b. 'TBD' means 'to be determined' by user.

C. Terminal I/O Routines using the TRAP #15 Handler

10. Write two subroutines to control ACIA#1 on the SANPER-1 Educational Lab Unit. This ACIA provides an input and output interface to the terminal. The routines should be similar to those of TRAP 14 Handler Functions 241 and 243. The proper registers must be initialized before the TRAP calling sequence is invoked. Note that entering a Return <CR> terminates data entry at the terminal.

11. The above routines can only be accessed by executing a TRAP #15 instruction in your program. Modify the original source code of Experiment #4 by replacing all TRAP #14 instructions with TRAP #15 instructions.

6.0 Procedure

Note: Bring graph paper to the lab for the purpose of recording waveforms.

A. Unit-to-Unit Communications

1. Use the Memory Modify (MM) command to store the character "i" into location $900.

2. Initialize address register A0 to point to location $900.

3. Run the PROC1 program. The same ASCII character should be transmitted continuously. Using an oscilloscope, observe the transmitted data on pin #3 (TX DATA) of the DB-25 connector that connects to ACIA #2 on the SANPER-1 Educational Lab Unit. Also, connect a wire from pin #4 to pin #5 on the DB-25 connector to tie *CTS to *RTS. Record this waveform in its RS-232-C format, and indicate the start, data, parity, and stop bits for this bit stream. From this waveform, draw the corresponding TTL waveform, which is the actual output of ACIA #2.

4. Modify PROC1 so that the transmitted character is now "S". Run PROC1 again. Record the RS-232-C waveform and indicate all four groups of bits. Draw the TTL waveform. Repeat this procedure three additional times for the following characters: "$", "ESC", "a".

5. Ask your Lab Instructor to connect your lab unit to an adjacent lab unit. Execute PROC1 on one on the lab units and PROC2 on the other. One of the terminals should display the characters received from the other lab unit. Verify that you have received the correct value and number of characters. Reverse the execution of the programs on the lab units, and again verify that the other lab unit is receiving and displaying characters properly.

6. Demonstrate to your Lab Instructor that Procedure Step #5 is working properly.

7. Leave your lab unit connected to an adjacent lab unit. Execute PROC3 on each lab unit. Each group should transmit a block of 10 characters. Your terminal should display the characters received from the other lab group. Through your terminal, your lab group should be able to pass data back and forth to the other lab group.

8. Demonstrate to your Lab Instructor that your polling routine (Procedure Step #7) is working properly.

9. Set the appropriate exception vector for ACIA #2 interrupt requests to point to the starting address of the interrupt service routine, SUBRT8.

10. Leave your lab unit connected to an adjacent lab unit. Execute PROC4. Each lab group should begin transmitting characters and your terminal should correctly display the characters received from the other lab group.

11. Demonstrate to your Lab Instructor that your interrupt routine (Procedure Step #10) is working properly.

B. Terminal I/O Routines

12. Set the exception vector of TRAP #15 to point to the starting address of SUBRT10.

13. Execute the revised logic translator program. Your program should be able to accept data from the terminal, perform the logic translation, and output data to both the terminal and the User Display of the SANPER-1 Educational Lab Unit.

14. Enter the test data and verify that your program is working properly. Debug your program using software breakpoints, software tracing, and the hardware single-step mode.

15. Demonstrate to your Lab Instructor that the revised logic translator program works properly.

7.0 Discussion

Submit the following to your Lab Instructor as a Final Report:

1. Listing files of all your programs and subroutines which include both global and local comments.

2. Drawings of the RS-232-C and TTL waveforms of the five different character words from Procedures A.3 and A.4.

3. In Procedure step #7, one of the lab units did not receive one block of characters. Why? How can this problem be solved?

4. Describe the advantages and disadvantages of implementing polling vs. interrupts.

5. List and explain which bits in the Status Register can cause an interrupt to occur.

6. The ACIA's Status Register contains the value $A3. What is the status of the ACIA?

7. What are the characteristics of a communications system if the ACIA's Control Register contains $C2?

8. If the ACIA's Control Register reads $81, determine what the parity bit must be when transmitting each of the following characters: "!", "7", "N", "P".

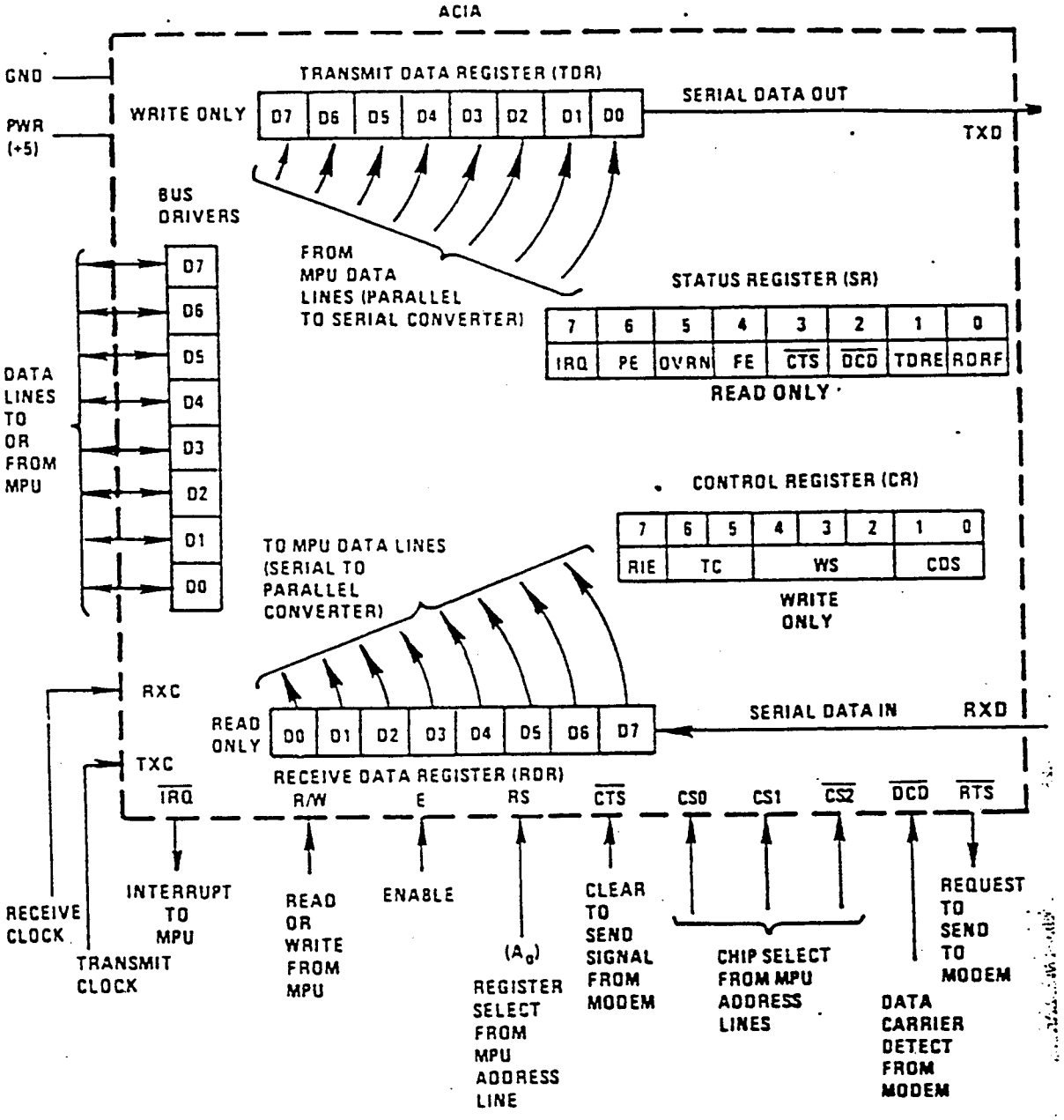


Fig. 9.30 Register select line

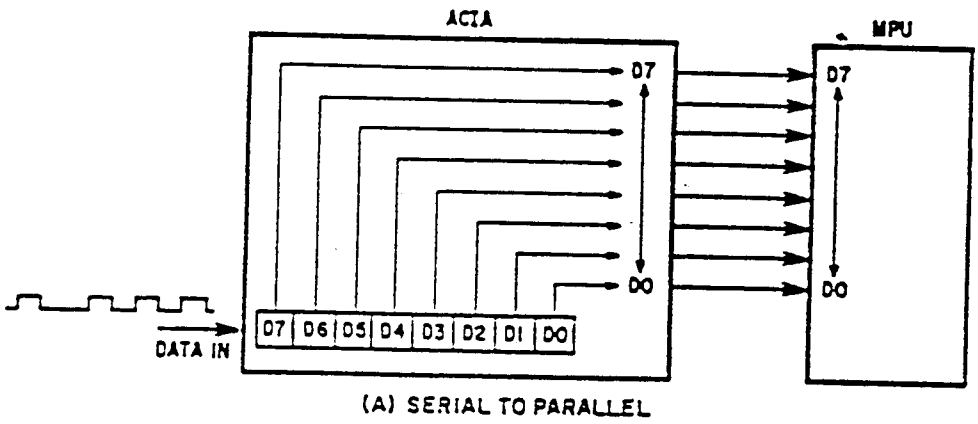
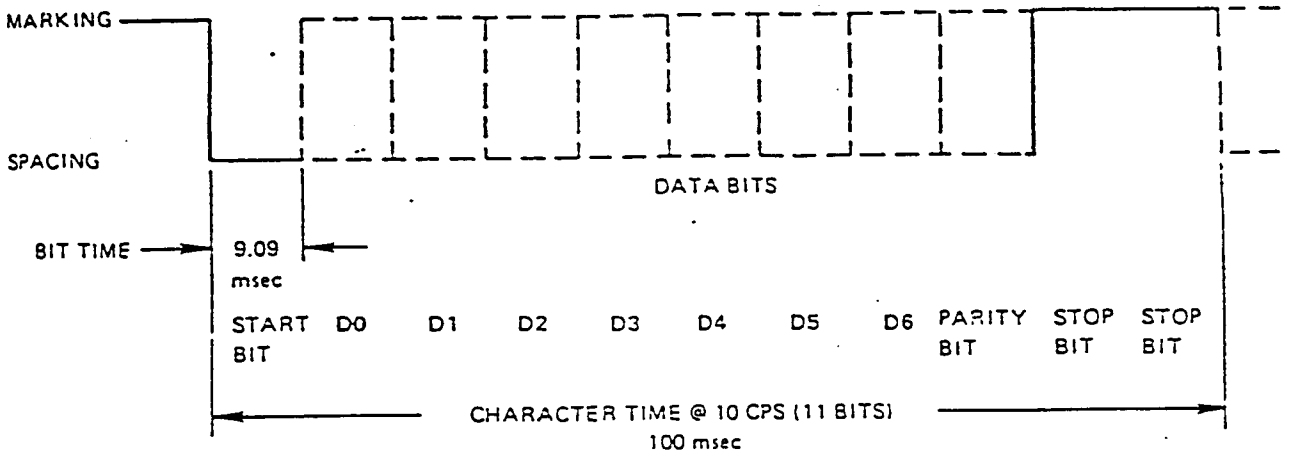
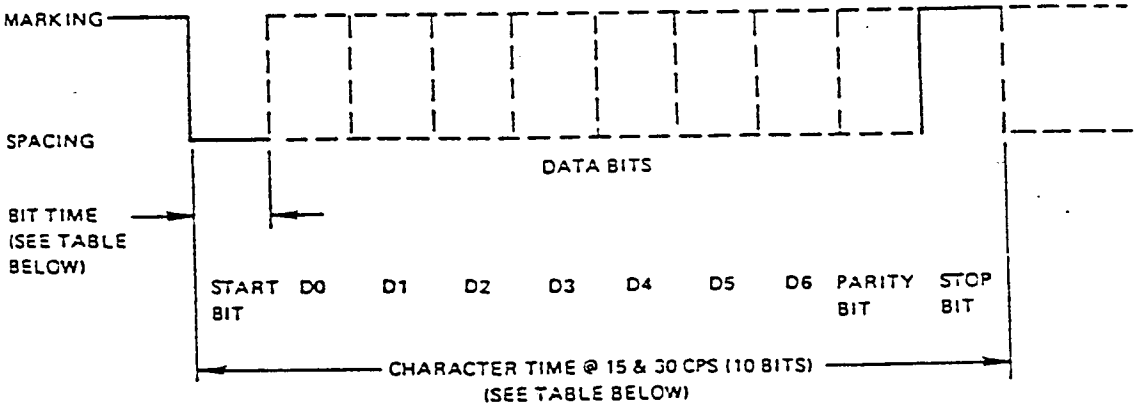


Fig. 9.22 Conversion functions of the ACIA

110 BAUD
SERIAL ASCII DATA TIMING



TR1159



BAUD RATE	150	300
CHARACTERS/SEC	15	30
BIT TIME (msec)	6.67	3.33
CHARACTER TIME (msec)	66.7	33.3

$$\text{BIT TIME} = \frac{\text{SEC}}{\text{BAUD RATE}}$$

150 & 300 BAUD SERIAL ASCII DATA TIMING

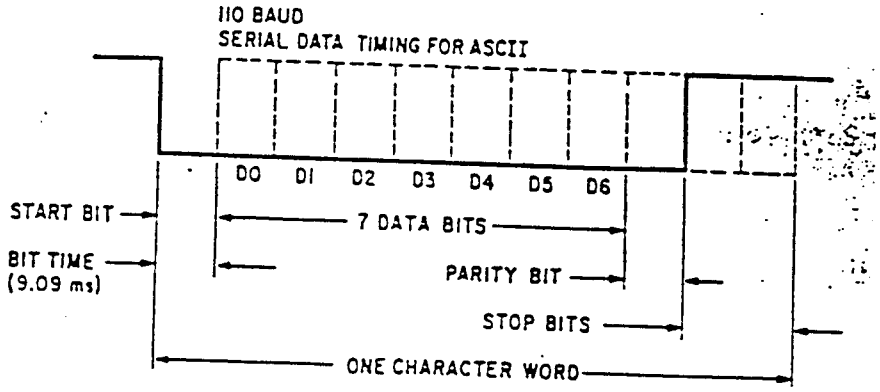


Fig. 9.23 Role of stop, start, and parity bits

Table 9.6 Baud Rate Data

Baud rate	110	150	300	1200
Bit time(msec)	9.09	6.66	3.33	.833
*Character time	.1 sec	0.73 sec	.0366 sec	.0092 sec
Characters/sec	10	13.7	27.32	108.7
Data bits/sec	80	110	218.6	870

*Assume one start bit, eight data bits (including parity), and two stop bits, or eleven bits per character.

Bit time = 1/baud rate
Character time = (total number of bits in word) × (bit time)
Characters/sec = 1/character time
Data bits/sec = 8 × characters/sec

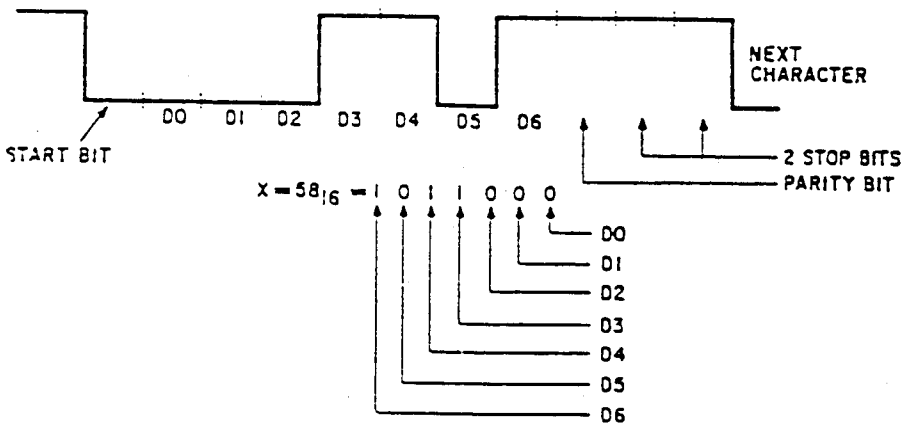
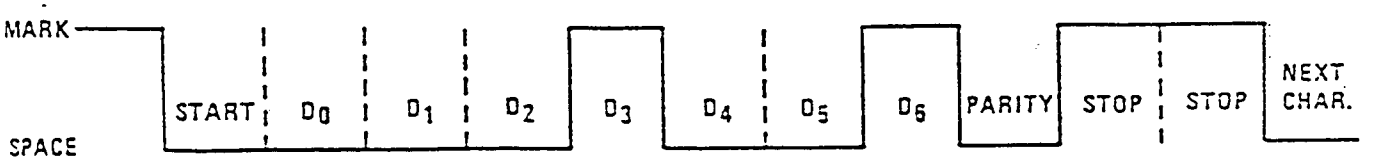


Fig. 9.24 Pulse train

b7 b6 b5 / Bits						0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4	b3	b2	b1	Row	Column / Hex	0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	B	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	C	FF	FS	,	<	L	\	l	|
1	1	0	1	13	D	CR	GS	-	=	M	]	m	}
1	1	1	0	14	E	SO	RS	.	>	N	^	n	~
1	1	1	1	15	F	SI	US	/	?	O	_	o	DEL

FIGURE 4-2. ASCII Character Set

SEND A 7 BIT ASCII CHAR. "H"
EVEN PARITY — 2 STOP BITS
$H = 48_{16} = 1001000_2$



ACIA CONTROL REGISTER FORMAT

B7 RIE	B6 TC2	B5 TC1	B4 WS3	B3 WS2	B2 WS1	B1 C2	B0 C1

B1	B0	FUNCTION (Tx. Rx)	MAX DATA CLOCK RATE
0	0	÷ 1	500 KHz
0	1	÷ 16	800 KHz
1	0	÷ 64	800 KHz
1	1	MASTER RESET	

B4	B3	B2	WORD LENGTH	+ PARITY	+ STOP BITS
0	0	0	7	EVEN	2
0	0	1	7	ODD	2
0	1	0	7	EVEN	1
0	1	1	7	ODD	1
1	0	0	8	NONE	2
1	0	1	8	NONE	1
1	1	0	8	EVEN	1
1	1	1	8	ODD	1

BITS CR5 AND CR6 HAVE THE FOLLOWING SYSTEM APPLICATION:

CR6	CR5	
0	0	THE $\overline{\text{RTS}}$ PIN IS LOW AND TRANSMIT INTERRUPTS ARE INHIBITED. THIS IS THE CODE USED WHEN REQUESTING THAT THE COMMUNICATIONS CHANNEL BE SET-UP. IT IS NOT CLEAR TO SEND DATA YET.
0	1	THE $\overline{\text{RTS}}$ PIN IS LOW AND THE COMMUNICATIONS CHANNEL HAS BEEN SET UP. THEREFORE, THIS CODE IS USED TO GENERATE IRQ'S VIA THE TDRE BIT IN THE STATUS REGISTER.
1	0	THE $\overline{\text{RTS}}$ PIN IS HIGH AND TRANSMIT INTERRUPTS ARE INHIBITED. THIS CODE CAN BE USED TO "KNOCK-DOWN" THE COMMUNICATIONS CHANNEL.
1	1	THE $\overline{\text{RTS}}$ PIN IS LOW (KEEP UP COMMUNICATIONS CHANNEL), A BREAK SIGNAL (LOW LEVEL ON TRANSMIT DATA OUT LINE) IS TRANSMITTED. THIS IS USED TO INTERRUPT THE REMOTE SYSTEM.

BIT 7 — RECEIVER INTERRUPT ENABLE (RIE)

"1" – ENABLES INTERRUPTS CAUSED BY

A) RECEIVER DATA REGISTER FULL GOING HIGH

B) A LOW TO HIGH TRANSITION ON THE DATA CARRIER DETECT SIGNAL LINE

"0" – INHIBITS INTERRUPTS DUE TO RECEIVE DATA REGISTER FULL OR LOSS OF RECEIVE DATA CARRIER.

ACIA STATUS REGISTER FORMAT

B7	B6	B5	B4	B3	B2	B1	B0
IRQ	PE	OVRN	FE	$\overline{CTS}$	$\overline{DCD}$	TDRE	RDRF

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/$\overline{W}$ is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 — Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 — The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect ($\overline{DCD}$), Bit 2 — The Data Carrier Detect bit will be high when the $\overline{DCD}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{DCD}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains high and will follow the $\overline{DCD}$ input.

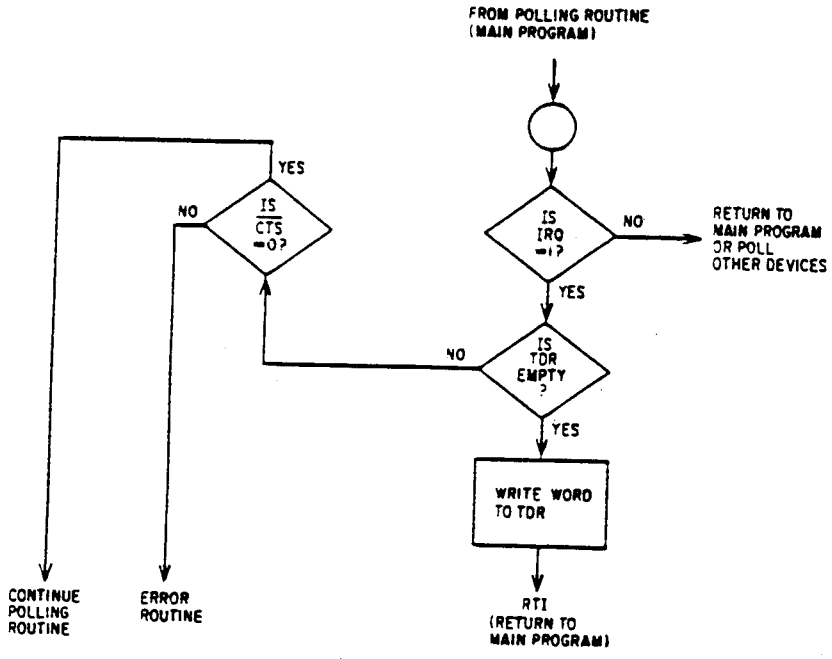
Clear-to-Send ($\overline{CTS}$), Bit 3 — The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low $\overline{CTS}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high. Master reset does not affect the Clear-to-Send status bit.

Framing Error (FE), Bit 4 — Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the first stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

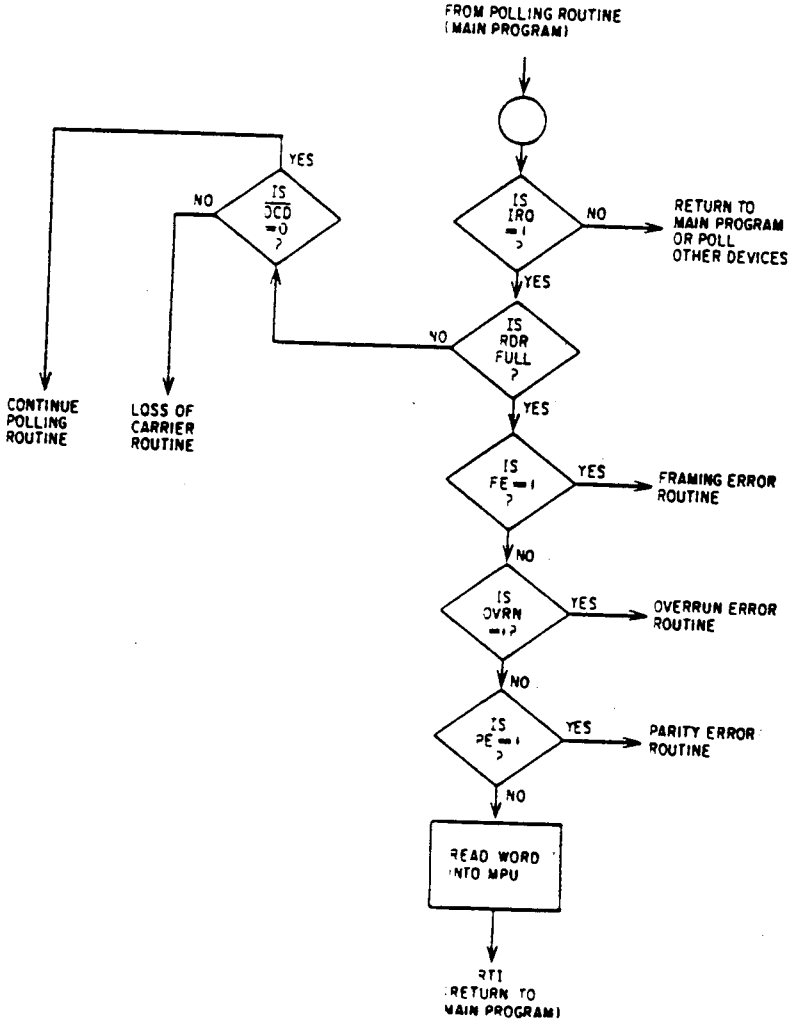
Receiver Overrun (OVRN), Bit 5 — Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6 — The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

Interrupt Request ($\overline{IRQ}$), Bit 7 — The $\overline{IRQ}$ bit indicates the state of the $\overline{IRQ}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{IRQ}$ output is low the $\overline{IRQ}$ bit will be high to indicate the interrupt or service request status. $\overline{IRQ}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.



Flowchart of transmit sequence



MOTOROLA Semiconductors

MC6850 (1.0 MHz) MC68A50 (1.5 MHz) MC68B50 (2.0 MHz)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

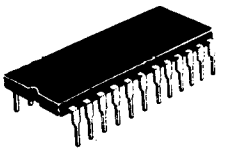
The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

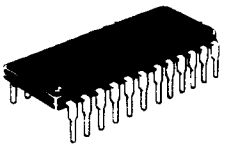
- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS
(N-CHANNEL, SILICON-GATE)

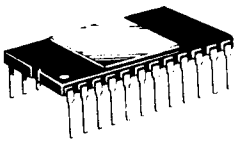
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



S SUFFIX
CERDIP PACKAGE
CASE 623

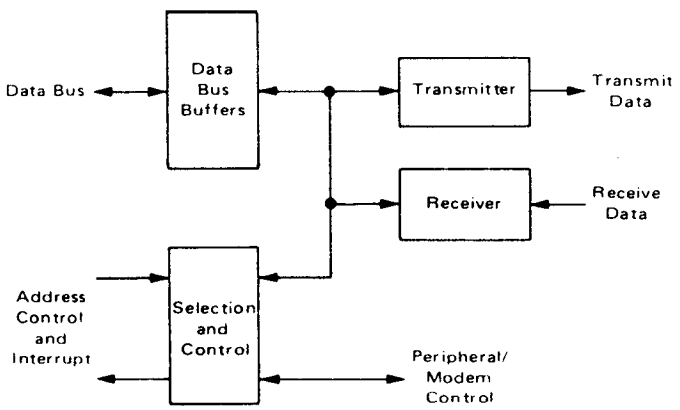


P SUFFIX
PLASTIC PACKAGE
CASE 709

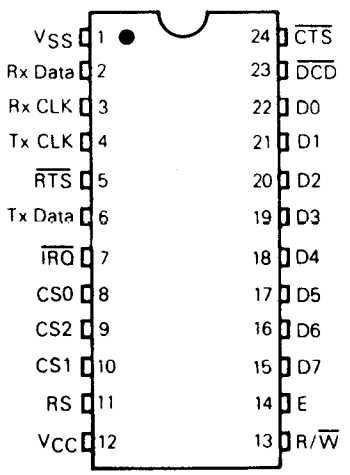


L SUFFIX
CERAMIC PACKAGE
CASE 716

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



©MOTOROLA INC., 1981

MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6850, MC68A50, MC68B50 MC6850C, MC68A50C, MC68B50C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip	θ_{JA}	 120 60 65	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, CS0, CS1, $\overline{CS2}$, Enable RS, Rx D, Rx C, $\overline{CTS}$, $\overline{DCD}$	I_{in}	–	1.0	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA, Enable Pulse Width < 25 μs) (I_{Load} = −100 μA, Enable Pulse Width < 25 μs)	D0-D7 Tx Data, $\overline{RTS}$	V_{OH}	V_{SS} + 2.4 V_{SS} + 2.4	– –	– –	V
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 μs)		V_{OL}	–	–	V_{SS} + 0.4	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	300	525	mW
Internal Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 E, Tx CLK, Rx CLK, R/W, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$	C_{in}	– –	10 7.0	12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	RTS, Tx Data $\overline{IRQ}$	C_{out}	– –	– –	10 5.0	pF

SERIAL DATA TIMING CHARACTERISTICS

Characteristic		Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low (See Figure 1)	÷ 16, ÷ 64 Modes	PW_{CL}	600	–	450	–	280	–	ns
	÷ 1 Mode		900	–	650	–	500	–	
Data Clock Pulse Width, High (See Figure 2)	÷ 16, ÷ 64 Modes	PW_{CH}	600	–	450	–	280	–	ns
	÷ 1 Mode		900	–	650	–	500	–	
Data Clock Frequency	÷ 16, ÷ 64 Modes	f_C	–	0.8	–	1.0	–	1.5	MHz
	÷ 1 Mode		–	500	–	750	–	1000	kHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)		t_{TDD}	–	600	–	540	–	460	ns
Receive Data Setup Time (See Figure 4)	÷ 1 Mode	t_{RDS}	250	–	100	–	30	–	ns
Receive Data Hold Time (See Figure 5)	÷ 1 Mode	t_{RDH}	250	–	100	–	30	–	ns
Interrupt Request Release Time (See Figure 6)		t_{IR}	–	1.2	–	0.9	–	0.7	μs
Request-to-Send Delay Time (See Figure 6)		t_{RTS}	–	560	–	480	–	400	ns
Input Rise and Fall Times (or 10% of the pulse width if smaller)		t_r, t_f	–	1.0	–	0.5	–	0.25	μs

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

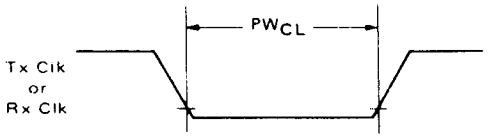


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

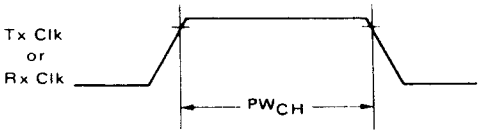


FIGURE 3 — TRANSMIT DATA OUTPUT DELAY

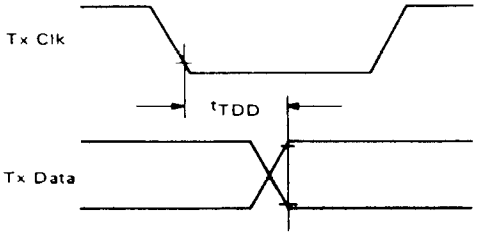


FIGURE 4 — RECEIVE DATA SETUP TIME (÷ 1 Mode)

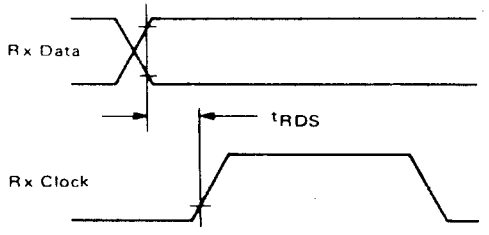


FIGURE 5 — RECEIVE DATA HOLD TIME (÷ 1 Mode)

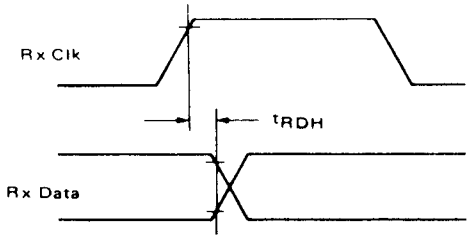
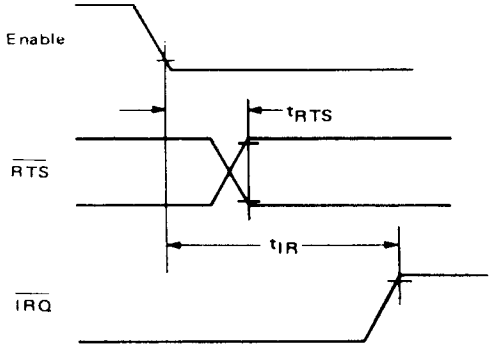


FIGURE 6 — REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES



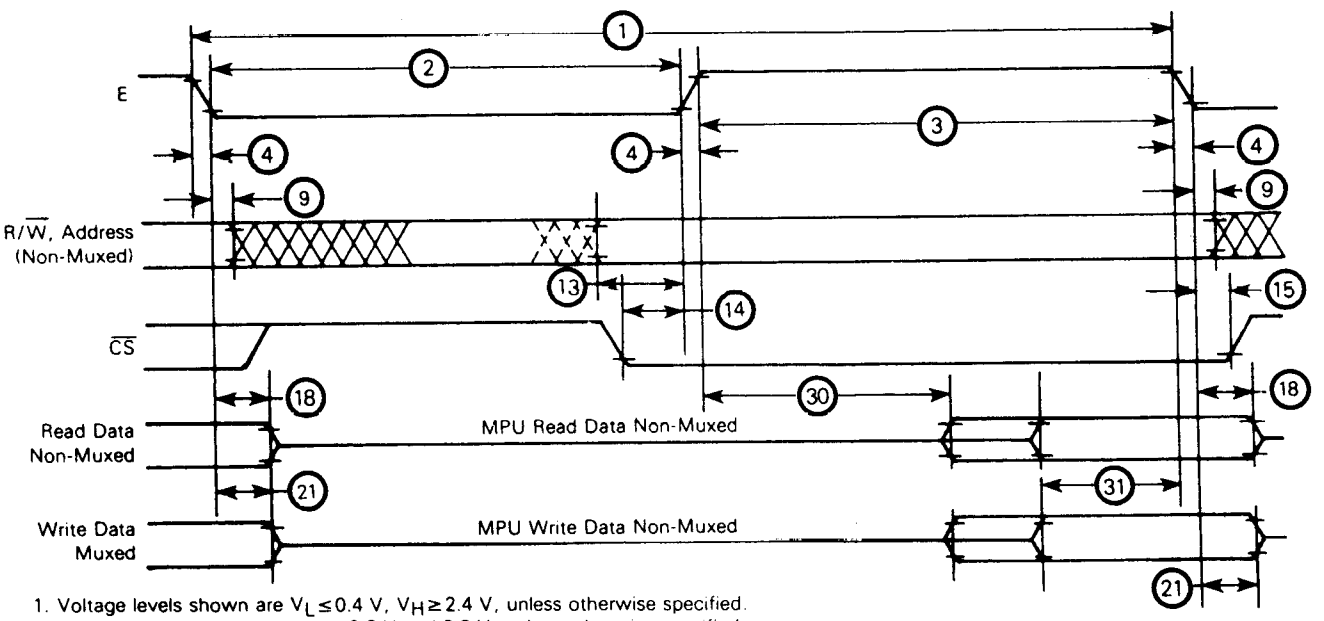
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2 and Figure 7)

Ident. Number	Characteristic	Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

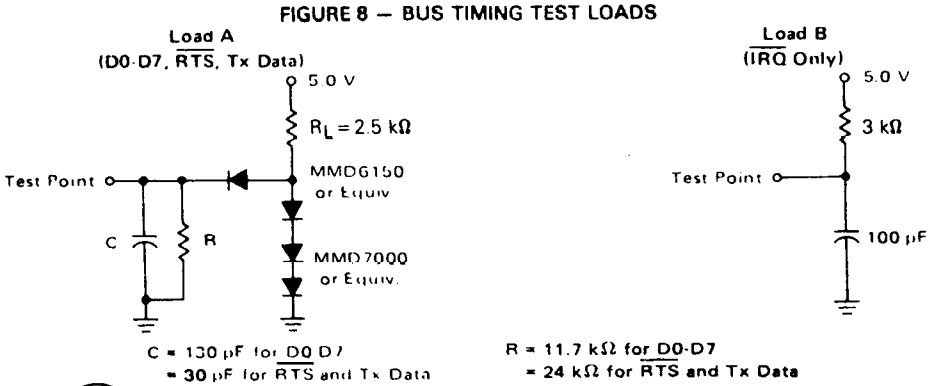
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 7 – BUS TIMING CHARACTERISTICS



1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

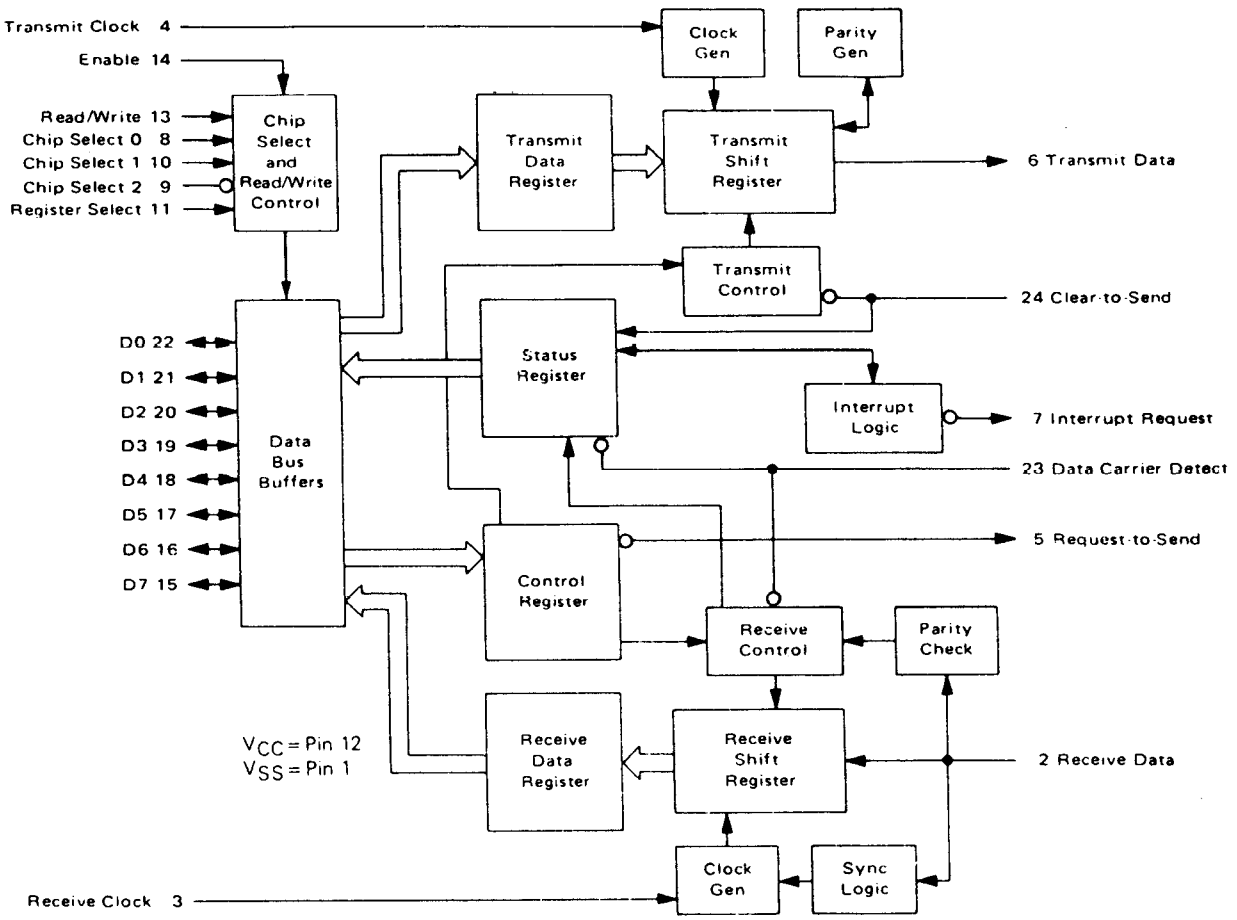
FIGURE 8 – BUS TIMING TEST LOADS



C = 130 pF for D0-D7
= 30 pF for RTS and Tx Data

R = 11.7 kΩ for D0-D7
= 24 kΩ for RTS and Tx Data

FIGURE 9 — EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset (CR0, CR1) should be set during system initialization to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. During the first master reset, the $\overline{IRQ}$ and $\overline{RTS}$ outputs are held at level 1. On all other master resets, the $\overline{RTS}$ output can be programmed high or low with the $\overline{IRQ}$ output held high. Control bits CR5 and CR6 should also be programmed to define the state of $\overline{RTS}$ whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits, parity (even, odd, or none), etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of

double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of 8 or 32 low samples on the receive line in the divide-by-16 and 64 modes respectively. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for a 7-bit word (7 bits plus parity), the receiver strips the parity bit (D7 = 0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the M6800 MPU with an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (D0-D7) — The bidirectional data lines (D0-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 $\phi 2$ Clock or MC6809 E clock.

Read/Write (R/$\overline{W}$) — The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, $\overline{CS2}$) — These three high-impedance TTL-compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and $\overline{CS2}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable Signal, Read/Write, and Register Select.

Register Select (RS) — The Register Select line is a high-impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request ($\overline{IRQ}$) — Interrupt Request is a TTL-compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The $\overline{IRQ}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The $\overline{IRQ}$ status bit, when high, indicates the $\overline{IRQ}$ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5•$\overline{CR6}$), and the Transmit Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send ($\overline{CTS}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmit Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of $\overline{CTS}$ which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect ($\overline{DCD}$) has gone high. An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of $\overline{DCD}$ are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16, or 64 times the data rate may be selected.

Transmit Clock (Tx CLK) — The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK) — The Receive Clock input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally.) The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) — The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used.

Transmit Data (Tx Data) — The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send ($\overline{CTS}$) — This high-impedance TTL-compatible input provides automatic control of the transmitting end of a communications link via the modem Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send ($\overline{RTS}$) — The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The RTS output corresponds to the state of the Control Register bits CR5 and CR6. When CR6 = 0 or both CR5 and CR6 = 1, the $\overline{RTS}$ output is low (the active state). This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect ($\overline{DCD}$) — This high-impedance TTL-compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output. The $\overline{DCD}$ input inhibits and initializes the receiver section of the ACIA when high. A low-to-high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set. The Rx CLK must be running for proper $\overline{DCD}$ operation.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed with RS high and R/$\overline{W}$ low. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 1-bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/$\overline{W}$ high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

TABLE 1 — DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	RS • R/$\overline{W}$ Transmit Data Register (Write Only)	RS • R/$\overline{W}$ Receive Data Register (Read Only)	$\overline{RS}$ • R/$\overline{W}$ Control Register (Write Only)	$\overline{RS}$ • R/$\overline{W}$ Status Register (Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send ($\overline{CTS}$)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request ($\overline{IRQ}$)

* Leading bit = LSB = Bit 0
** Data bit will be zero in 7 bit plus parity modes.
*** Data bit is "don't care" in 7 bit plus parity modes.

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/$\overline{W}$ are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1) – The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{CTS}$ and $\overline{DCD}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	÷1
0	1	÷16
1	0	÷64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) – The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{RTS}$ = low, Transmitting Interrupt Disabled.
0	1	$\overline{RTS}$ = low, Transmitting Interrupt Enabled.
1	0	$\overline{RTS}$ = high, Transmitting Interrupt Disabled.
1	1	$\overline{RTS}$ = low, Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7) – The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low-to-high transition on the Data Carrier Detect ($\overline{DCD}$) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/$\overline{W}$ is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 – Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 – The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect ($\overline{DCD}$), Bit 2 – The Data Carrier Detect bit will be high when the $\overline{DCD}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{DCD}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains high and will follow the $\overline{DCD}$ input.

Clear-to-Send ($\overline{CTS}$), Bit 3 – The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low $\overline{CTS}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high. Master reset does not affect the Clear-to-Send status bit.

Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the first stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

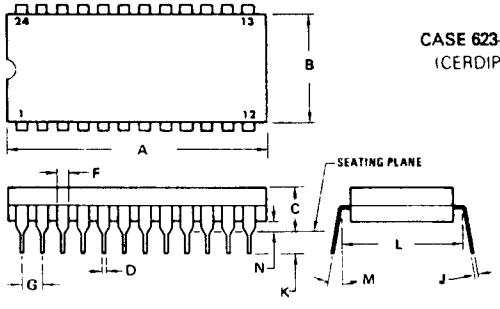
Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has

been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6 – The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver partiy check results are inhibited.

Interrupt Request ($\overline{IRQ}$), Bit 7 – The $\overline{IRQ}$ bit indicates the state of the $\overline{IRQ}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{IRQ}$ output is low the $\overline{IRQ}$ bit will be high to indicate the interrupt or service request status. $\overline{IRQ}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

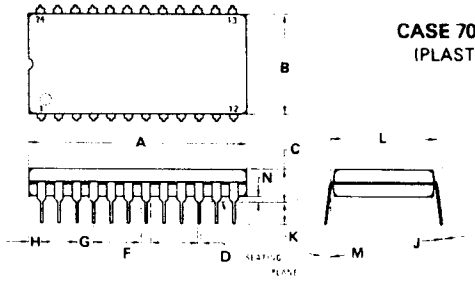
PACKAGE DIMENSIONS



CASE 623-03
(CERDIP)

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	31.24	32.77	1.230	1.290
B	12.70	15.49	0.500	0.610
C	4.06	5.59	0.160	0.220
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	2.29	4.06	0.090	0.160
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.27	0.020	0.050

NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL)



CASE 709-02
(PLASTIC)

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	31.37	32.13	1.235	1.265
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.03	0.065	0.080
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

Motorola reserves the right to make changes to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.