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".

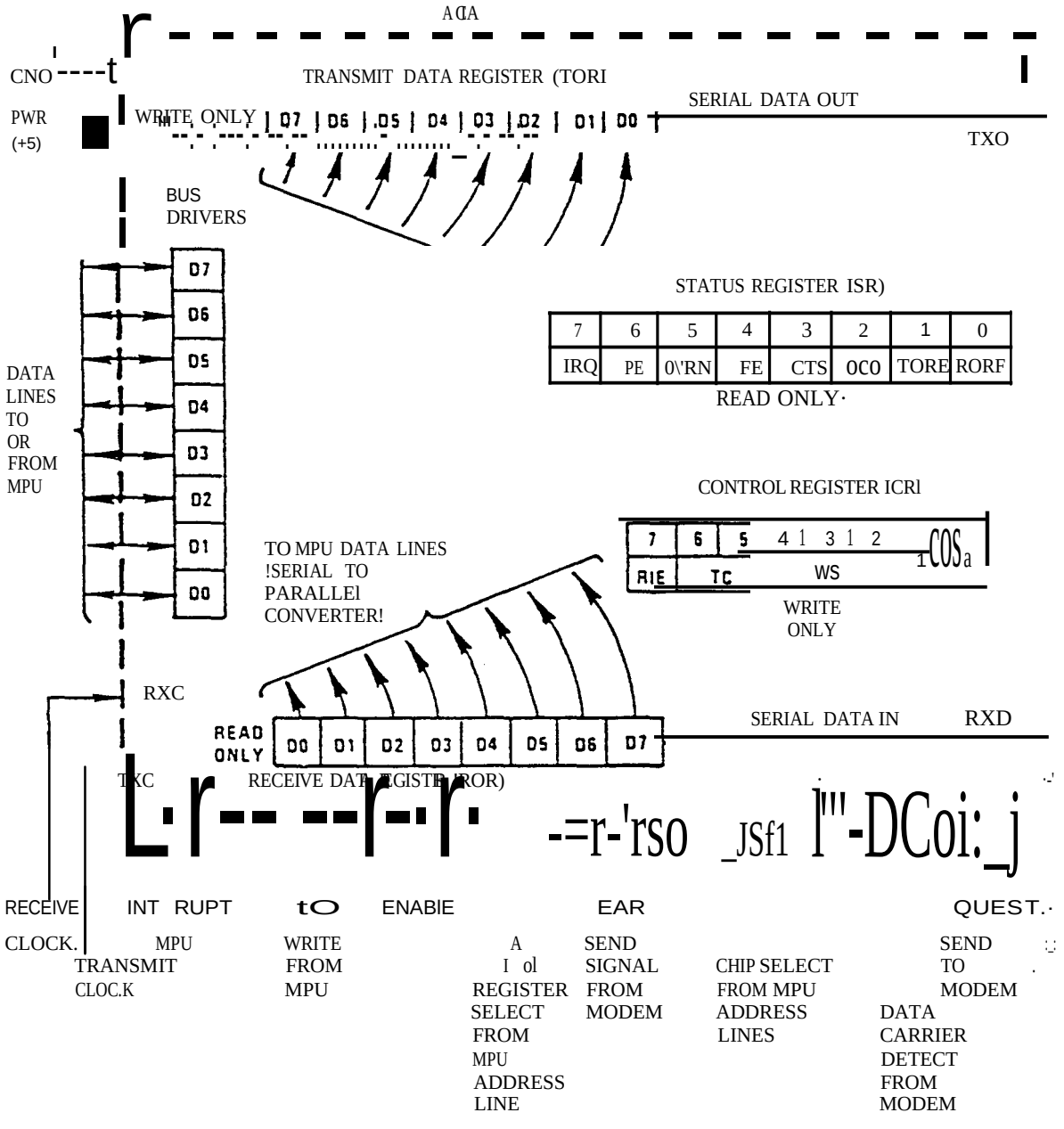
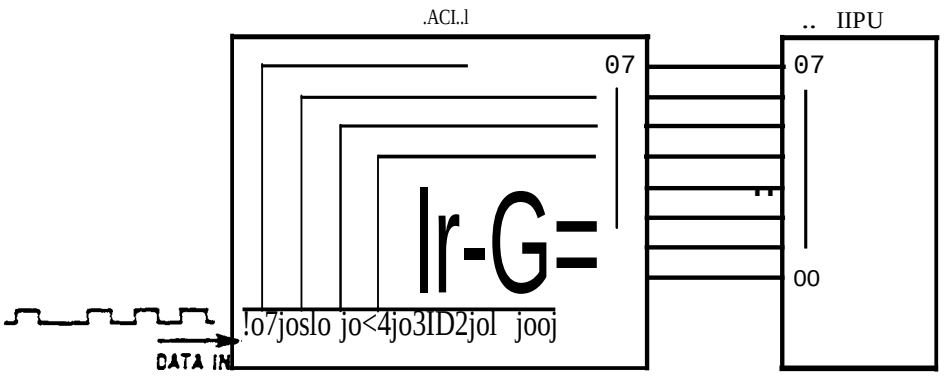
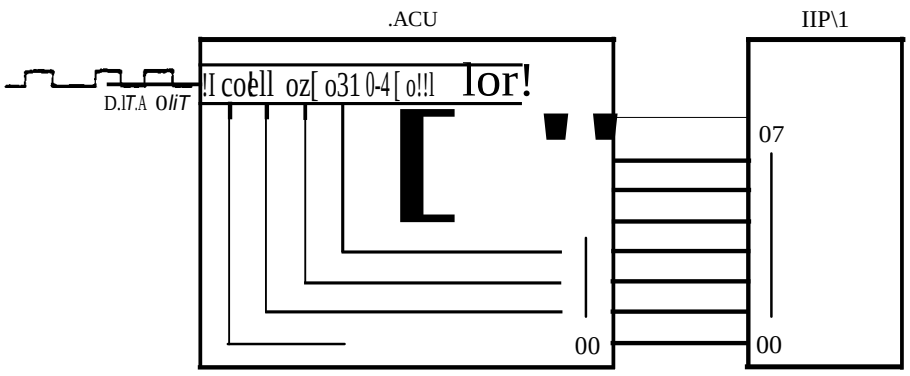


Fig. 9.30 Register select line



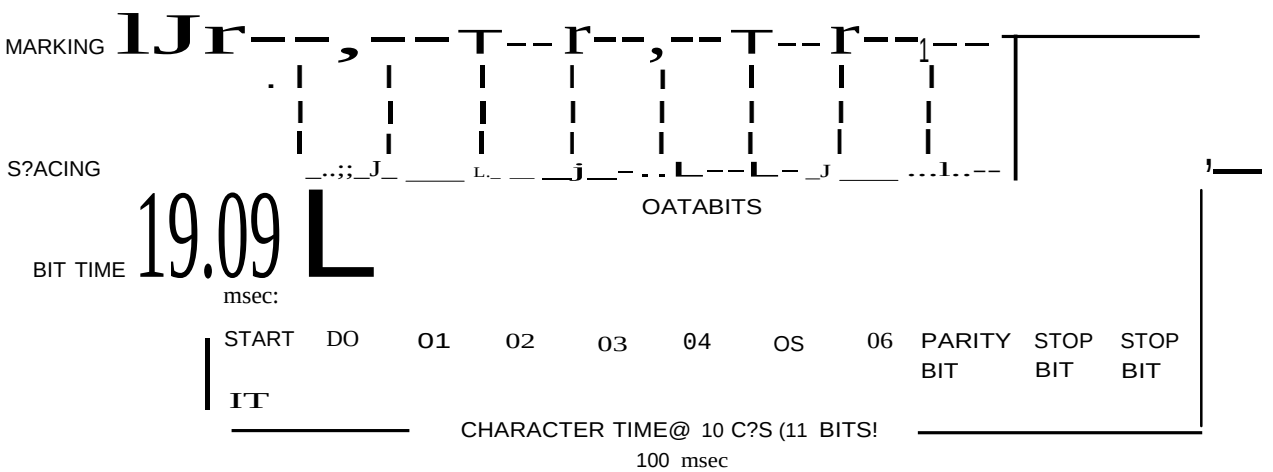
(A) SERIAL TO PARAL.l...EL



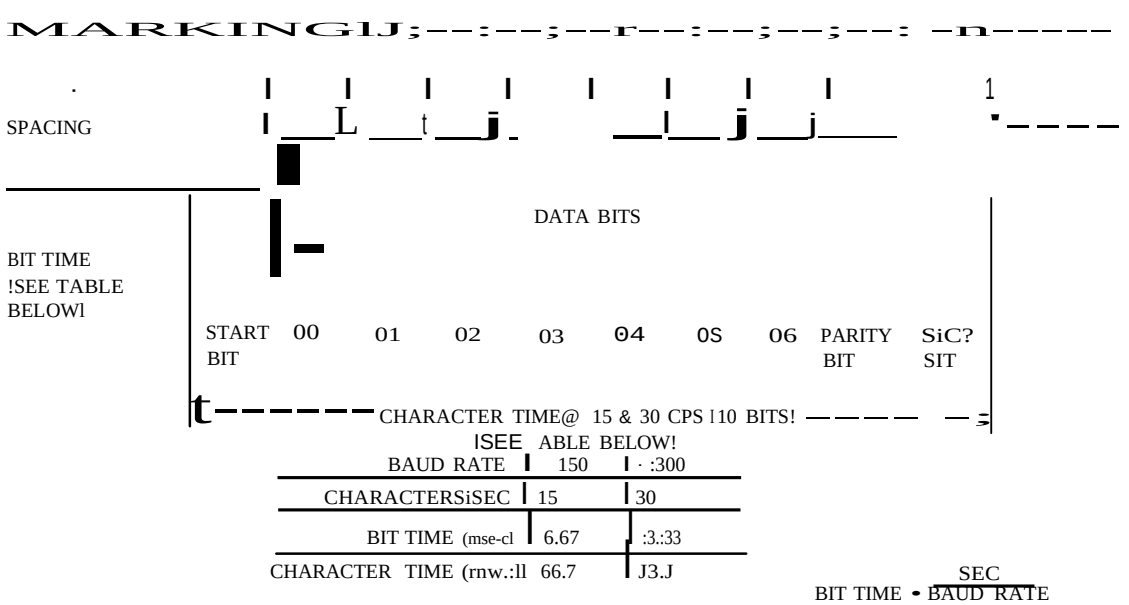
(8) PARALL!!..TO S!:RW.

Fig. 9.22 Conversion functionof the ACIA

110 BAUD
SERIAL ASCII DATA TIMING



TR1159



BAUD RATE	150	300
CHARACTERSiSEC	15	30
BIT TIME (mse-cl	6.67	:3.:33
CHARACTER TIME (rnw.:ll	66.7	J3.J

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

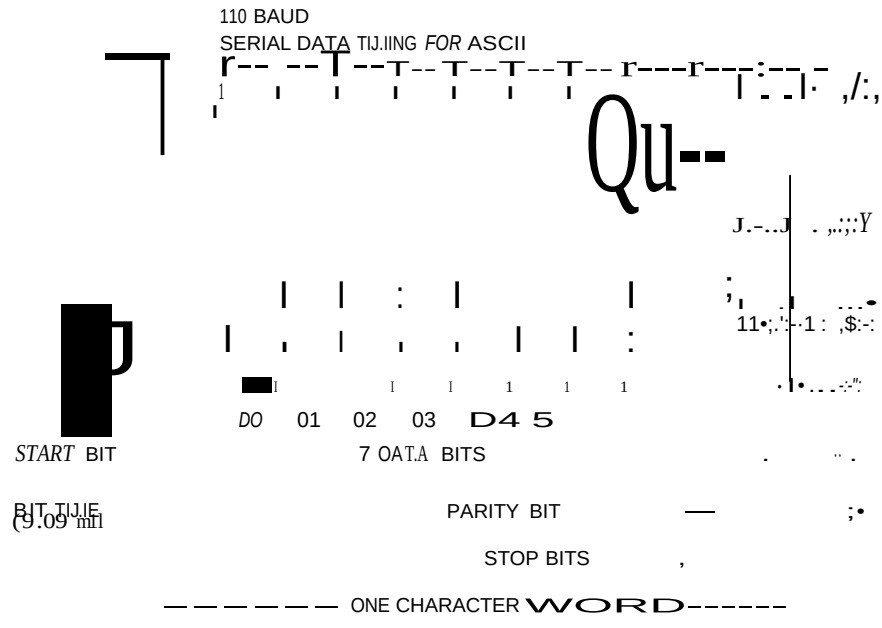


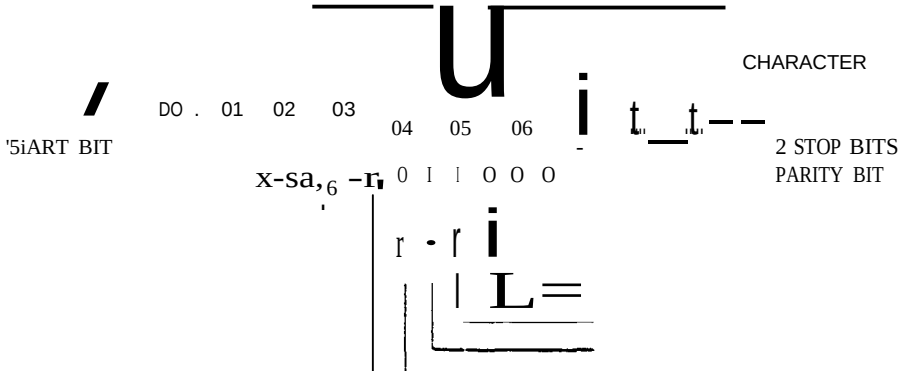
Fig. 9.23 Role of stop, start, and parity bits

Table 9.6 Baud **Rate Data**

Saud 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) X (bit time)
Characters/sec — 1/character time
Data bits/sec = 8 X characters/sec



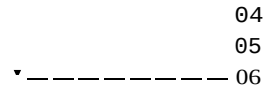
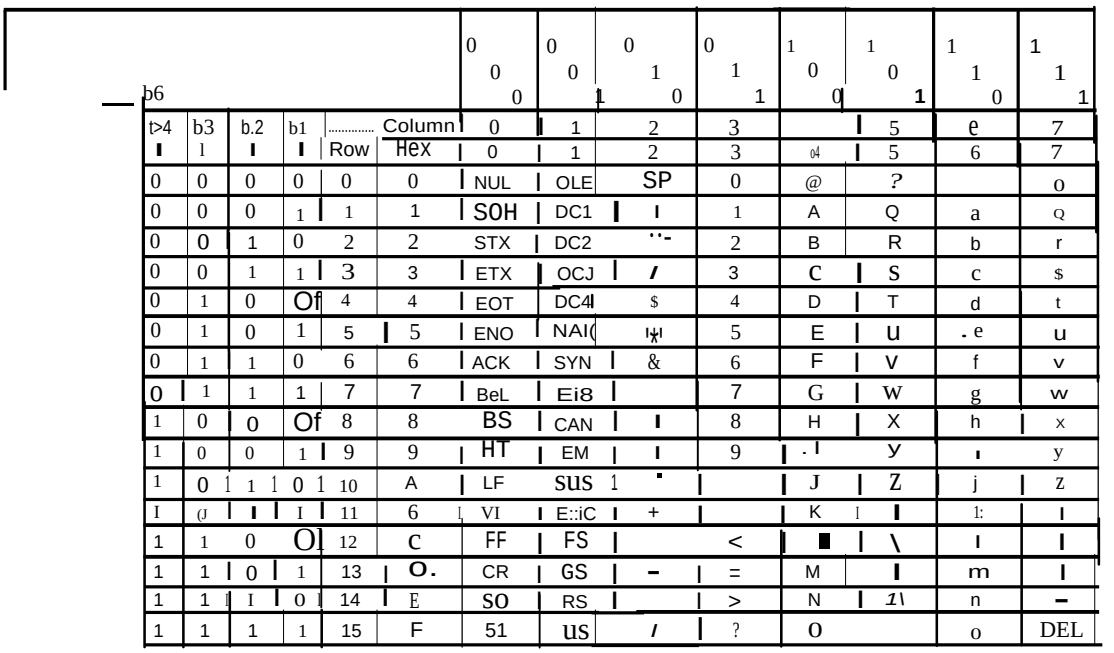


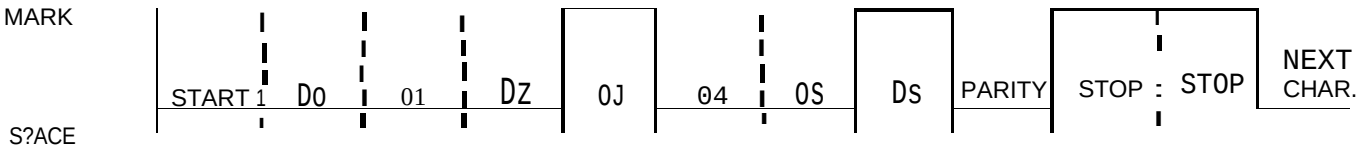
Fig. 9.24 Pulse *train*



b6						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	Hex	0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	OLE	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. 5C!ICharacter Set

SEND A 7 BIT ASCII CHAR. "H"
EVEN PARITY- 2 STOP BITS
H = 481s 1001 00Cb



ACIA CONTROL REGISTER FORMAT

97 RIE	66 TC2	65 TC1	8-1 WS3	63 WS2	62 WS1	61 C2	60 Cl

81	80	FUNCTION ITx. Rxl	MAX DATA CLOCK RATE
0	0	-i-1	500KHz
0	,	.;. 16	800KHz
,	0	-i-64	800KHz
,	,	MASTER RESET	

84	83	82	WORD LENGTH +	PARITY +	STOP BITS
0	0	0	7	EVEN	2
0	0	1	7	ODD	2
0	,	0	7	EVEN	1
0	1	,	7	ODD	,
	0	0	8	NONE	2
1	0	1	8	NONE	1
	1	0	8	EVEN	1
	,	1	8	ODD	1

BITS CR5 AND CR5 HAVE THE FOLLOWING SYSTEM A?P!..!C..\7i0N:

CRS	CRS	
0	0	THE RTS PIN IS LOW AND TRANSM!T INTERRUPTS ARE INHIBITED. THIS IS THE CODE USED WHEN REQt;ESTiNG THAT THE COMMUNICATIONS CHANNEL BE SET-UP. IT IS NOT C!..:AR TO SEND DATA YET.
0		THE RTS PIN IS LOW AND THE COMMUNICAT10NS CHANNEL HAS SEEN SET UP. THEREFORE. THIS CODE IS USED TO GENERATE IRQ'S VIA THE TORE SIT IN THE STATUS REGISTER.
	0	THE RTS ?IN IS HIGH AND TRANSMIT INTERRUPTS ARE INHIBITED. THIS CODE c.:..N SE USED TO "KNOCK-OOWN" THE COMMUNIC.:..TIONS CHANNEL
		THE RTS ?IN IS LOW (KE:::? UP COMMUNICATIONS CHANNEL), A SREAK SIGNAL (LO'N LEVEL ON TRANSMIT DATA OUT LINE) IS Ti=IANSMITTED. THIS IS USED TO INTERRUPT THE RE OTE SYSTE .

BT 7.- RECEIVER INTERRUPT ENABLE (RIE)

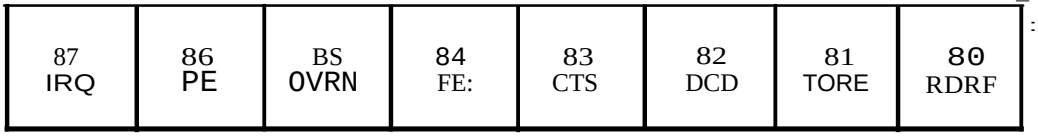
'T' E ASLES INTERRUPTS c;.us;c BY

A) RECEIVER DATA REGISTER FULL GOING HIGH

il) A LOW TO HIGH TR..\ SITION ON THE DATA C:.RRIEn DETECT SIGNAL LINE

"0" – INHIBITS I ITERRUPTS :JUE TO RECEIVE DATA REGISTER FULL OR LOSS OF RECEIVE *D;.U* CARRIER.

ACIA STATUS REGISTER FORMAT



STATIJS REGISTEr\

lnforma;:on on :ne s:a:us of :ne ACLO:. is ava1lable :o the MPU by reaaing the ACI;.. Status Re;:s:er. This reac·only register is selec:ee when F.S is lew ana *?Jw* IS hign. lnicr·r:"\ation ere-:! in tn1s r rster incicates tl'".e status of me T:ansmit Cat.J Register.!:":Receive Ca:a Megister anc error lcgic,·ar.c :ne cenpnerallr.:odem status inouts cf the ACIA.

Receive Data i'\egister Full !RCi'\Fl, Bit 0 — Rece1ve Oa:a Reg1s:er F\:11 indiC3tes that rec!:ved ::ata has en trans·lerred to :r.e rteceiVe Da:a ?.e<;is:er. ROF.F is c:earea ai:er an MPU reaa of ::'\e Rece1ve Data i'ie;•ster or eva master res t. The Clearea or e:r.::v s:a:e incic:ates t:'\a::-:e contents at tl':e P.eceive Ca:a eg1ster are r.o: C\Jtrent. Oata Carr:er Cetect t:e1ng ni;:-: also causes R D = :a indicate emctv.

Trens'"it Oats i'\eo;;io:ter :,.,o:V !TDi'\El, Bit 1 — Tt1e iransmit Cata rtegister ::-::tv bn oeu·::: set hic:n inC1:a•es : at :he 7:ar.smr: Data =e-;:s:er c:= :er:-:.s ave-=n ::a s· 1errea anc t!'"lat new Cata .'"';ay be erne e-:. ihe !cw sta!e 1:-1.. cu:ates t::at roe !"t'g!Ster :s it..:t!anc: :.-,at :-ar.s:-r:rsstcn of a new C arac!e!" aS r:::t t:e-;un S!ro:Ce te las:wn:e ca:a c:r.-:mar:d.

Data Carrier Ostec: !C5l, Sit 2 — ine Data Carner C!tec! :1 wti; e nrc;n wl'"ethe .:-.:'..:; trorn a ccem :-.as ;one u;=" :::lrt'IC:cae tnat a earner :s r.c::esen:.. 'ihis Cit going :":;;;'"\ causes an lnte ruot Me-: es: :c be cene!"ated .,. en :-:e...:. ecet_vlr'"te n.::: ::-ao•e rs se. t: re!'T1 tns !'"ign a••er :ne •new:.. tS re:ur:'1ec lew !.;-:::1 c:eare-o v irst reaaar:; ::-:e 5:.atus Rtl9:s:.er anc ! e:e Oa:a egts:.e:- :r u :ii a rr.as:er !"eset occ-..:!"S. !f t:"le QC:J tr:ot.t :-err.atr.s ig:"l a er reac s:a:..s ar:c :-eae :ata or mas:er reset nas ccc ::-ea tne inter,...;:t:s :::e!ree, tile 5'COsta:us ::: rema1ns r.i;n an w111 fellow ::'\e inc:. t.

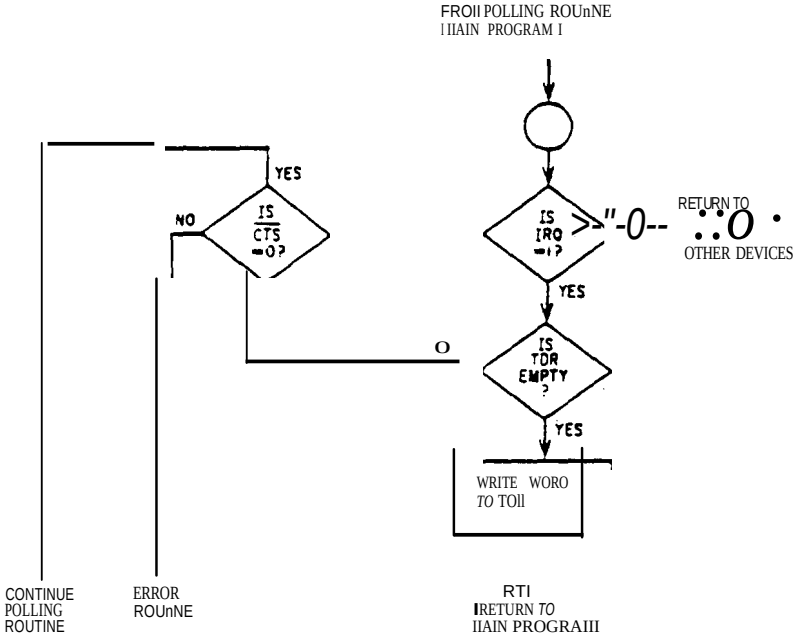
C!ear-:o-Send lffil. 6it 3 — Tt-e C:ear·tc·Ser:d ::it in·C:':;ates ::::;s:a:e o!the C:ear·to·SenC: 1n::ut !rem a mccem. '"" •Ow C • S .nc1c.lts :r:at ::'"!ere tS a \..:ear-:c.. Send trc:"":'! :r.e mocem. :n :!":e nu;state. tr.e Trans 1t Ca:a F\egts:er :. ;tv tat 1s inniOitana t:'\e c:aar-to·Ser.c s:a::.;s ::!t Wtil :'.-e nic!1. Master reset eoes not at e::he C!ear..!o-Send s atus ::r;.

Framing Error !F:J. Sit 4 — !=ra.,.,1n;error 1notc3:es that ::oe recetvec =..,arac:e' •S ;.,cro: errv trarf'ed ::Jv a s:art and a s:oo b1t ar:1S c :ec:ea ::v tne acsence of tne first s:o:l t:at. T:-.ts errcr : c;cates 3 svnc:o:..ronz:3uon err:r. fauatv transmrs·s1cn, cr 1 breat co"C1t10n. ihe tram1n9 error f!ag 1S set cr reset cunnt; :ne recetve aa:a transfer :tme. iheretcre. : 15 er.. ror ineic3:or is 'esent :nrou;!'lo-.t ::-:e uma :nat tne assocatea cnarac:er 1s available.

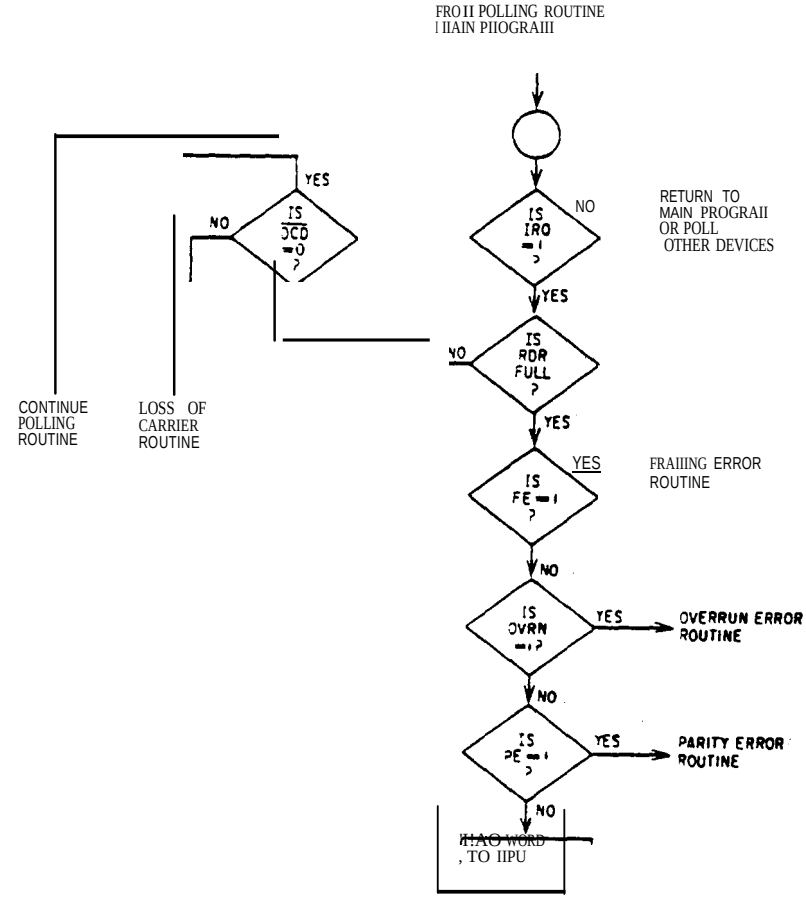
Rec:e er Ovem:n IOVRNl, Bit 5 — Cverrun is an error llag :na11nd:cates :nat one or more Cr'larac::ers in the data stream were lest. That is, a cl':arac::er or a numoer of c..,arao:ers ":.ere received but not read l:cm the Rece1ve Oa:a Re;ister !nDRl pnor :o su:se<;uent ::t:arac:ers :;,ein; rece1ved. ine overrun condition oe<;ins at t.e m1Cpoint of :ne last l:it of tile second cl!arac:r 'ec:eived in suc::ession wi:hout a read of tne ROR having cc:urred. Tne Cverrun coes not oc:ur in :ne S:atus Re;ister un:.l tl':e valid c..,a,ac:er ;:nor :o Overrun nas t:ee'n read. The RCi'IF bit rerr.air.s set unul :ne Overrun is reset. C.'larao:er svnehrcn1zation is main:ained cluring tt:e Overrun c::nditicn. The Overrun indication is reset after the reading of data from tne Rece.ve Data i'le;ister or by a Master Reset.

Parity Error !PEl. Bit 6 — The ::aritv error flag indicates that the numt:er of highs lonesl in the c. arac:er cces not agree w1th tne oreseiec:ed odd or even ::arity. Odd ::antv is defined :o t:e wnen :r:e total n ,;ml:er of ones is odd. ihe parity error lndiC3t10n wm t:e ::re.sent as !eng as :.e cata c:-:arac:er is in :.'"e RCR. If no ::lri:V is selec:ed. :r:en botn the transmitter ;)aritV generator eu::ut ar.d :::e receiver partly c. eck: resuJts are inn•bited.

tntem:ot IR—tiRO!, Sit 7 — The InC :m indi03tes :r.e s ate of tne tnU cutcut. Any in!erNCt c::r.c1 :cn w•tn its ac· :icable enable ...il be inci03te<:: on :r.is s tus it. Anyn.,.,e tl':e ii'iQ out::ut is low the ifilj' t:1t woil ::e r:i;n :a inOi03te tl':e in:erruot cr ser.ice re<;uest status. iRU is c:eared bv a read ocerat1on :o :e ece.ve Data Me<;1s:er cr a wnte ccera:,on :o :r.e irans.,•t Oata Reo;ister.



Flowcnart of transmit sequence



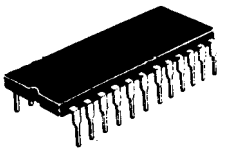
'HI
.RfTURII ro
Alii PROGRAIII

\.O MOTOROLA
Sen iconduc'fors

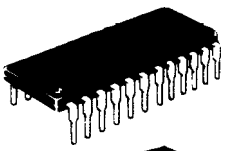
MC6850
(1.0 MHz}
MC68A50
(1.5 MHz)
MC68B50
(2.0 MHz)

MOS
(N-CHANNEL, SILICON-GATE)

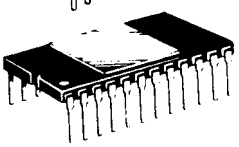
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



S SUFFIX CtfiOIP PAC"AGt CA::,E 623



P SUFFIX PLASTIC PACKAGE CASE700



L SUFFIX CERAMIC PACKAGE CASE 716

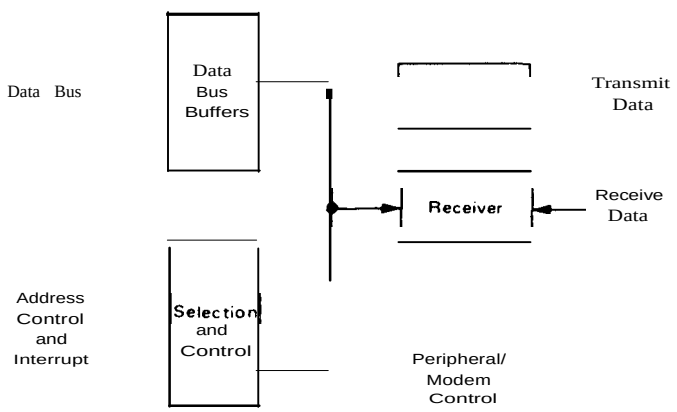
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIAl

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.

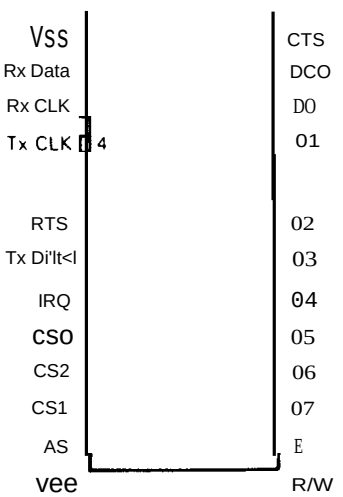
The bus interface of the MC6850 includes select, enable, read/write, Interrupt and bus interface logic to allow data transfer over an 8 bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional conf1guration 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, rece1ve control, and interrupt control. For peripheral or modem operat1on, 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 Transm1ssion
- 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 Funct1ons
- Double Buffered
- One- or Two-Stop Bit Operation

MC6860 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	Vee	-0.3 to + 7.0	v
Input Voltage	Vin	-0.3 to + 7.0	v
Operatrng Temperature Range MeOO .MCOOA .MCOOB MC C. MeOOA C. MCOOB e	TA	TL to TH 0 to 70 -40 to +85	0C
Storage Temperature Range	Tstq	-55to+1	0C

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 Vss or Vee!.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plasttc Ceramtc Cerdip	8JA	 120 60 65	°C/W

POWER CONSIDERATIONS

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

$$TJ=TA+(Po•8JAI \quad (1)$$

Where:

TA ➖ Ambient Temperature, oc

8JA•Package Thermal Resistance, Junction-to-Ambient, °C/W

Po ➖ PiNT+ PPORT

PiNT•Icc x Vee. Watts – Chip Internal Power

PPORT ➖ Port Power Dissipation, Watts – User Determined

For most applications PPORT -C PiNT and can be neglected. PpQRT may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between Po and TJ (if PPQRT is neglected) is:

$$Po=K+(TJ+273°C) \quad (2)$$

Solving equat1ons 1 and 2 for K gives:

$$K = Po•(TA+ 273°Cl + 8JA•Po^2 \quad (3)$$

Where K IS a constant pertaining to the part1cular part. K can be determined from equation 3 by measuring Po !at equilibrium) for a known T A Using this value of K the values of Po and T J can be obtained by solving equations (1) and (2) iteratively for any value ofT A-

DC ELECTRICAL CHARACTERISTICS IVee= 5.0 Vdc ± 5%. Vss = 0, TA= TL to TH unless otherwtse noted I

Chl.tra<::teristic		Symbol	Min	Typ	Max	Unit
Input Htgh Voltage		VtH	Vss + 2.0	–	vee	v
Input Low Voltage		V1L	vss-03	–	Vss+O.B	v
Input Leakage Current IV,n = 0 to 5.25 VI	R/W, CSO, CS1, Cs-2", Enable RS. Rx D. Rx C. CTS, DCD	I,n	–	1.0	25	p.A
Three-State IOff State) Input Current Vin = 0.4 to 2.4 VI	00-07	ITSI	–	2.0	10	p.A
Output Htgh Voltage IILoad= -205 p.A, Enable Pulse Width < 25 p.sl IILoad= – 1OO ,.A. Enable Pulse Wtdth< 25 tsl	00-07 Tx Data, RTS	VOH	 Vss+2.4 Vss+2.4	 – –	 – –	v
Output Low Voltage IILoad= 1.6 mA. Enable Pulse Width< 25 p.sl		VOL	–	–	Vss+0.4	v
Output Leakage Current !Off State! IVoH = 2.4 VI	IRQ	ILOH	–	1.0	10	p.A
IntPrnal Power Otsstpatron !Measured at TA= TLI		PiNT	–	300	525	mW
Internal Input Capacrtance IVrn=O, TA= 25°C, f= 1 OMHzl E. Tx CLK, Rx CLK. R/W, RS. Rx Data, CSO, CS1, CS2, ffi. DCD	 DO-D7. 	Cin	 – –	 10 7.0	 12.5 7.5	pF
Output Capacitance IV,n = 0, TA= 25°C, f = 1.0 MHzl	RTS, Tx Data TAO	Cout	– –	– –	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 + 1 Mode	PWcL	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Pulse Width, High (See Figure 2)	16.... 64 Modes ... 1 Mode	PWcH	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Frequency	+ 16. + 64 Modes + 1 Mode	fc	– –	0.8 500	– –	1.0 750	– –	1.5 1CXXl	MHz kHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)		tTDD	–	600	–	540	–	460	ns
Rece1ve Data Setup T1me (See Figure 41	+ 1 Mode	'ADS	250	–	100	–	30	–	ns
Receive Data Hold Time (See F1gure 51	... 1 Mode	tRDH	250	–	100	–	30	–	ns
Interrupt Request Release T1me (See Figure 6)		tiR	–	1.2	–	0.9	–	0.7	JLS
Request-to-Send Delay T1me (See Figure 61		tATS	–	560	–	480	–	400	ns
Input R1se and Fall T1mes (or 10% of the pulse width if smaller)		tr. If	–	1.0	–	0.5	–	0 25	JLS

FIGURE 1 – CLOCK PULSE WIDTH, LOW-STATE

FIGURE 2– CLOCK PULSE WIDTH, HIGH-STATE

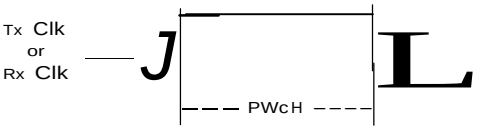


FIGURE 3– TRANSMIT DATA OUTPUT DELAY

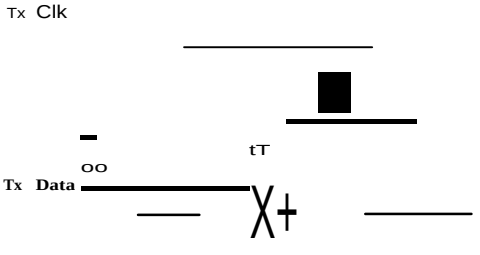


FIGURE 4– RECEIVE DATA SETUP TIME
(+ 1 Model

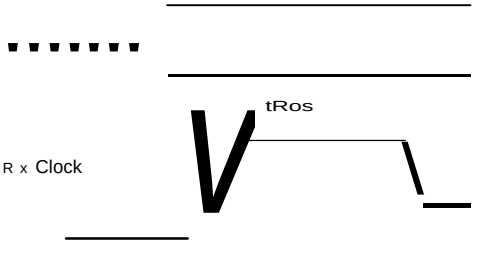


FIGURE 5– RECEIVE DATA HOLD TIME
(+ 1 Model

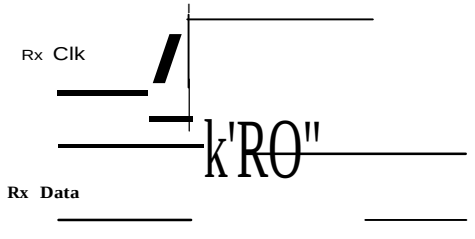
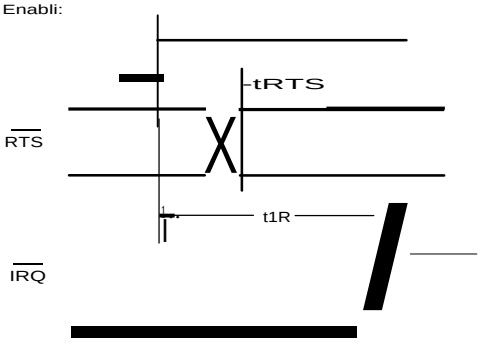


FIGURE 6 – REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES



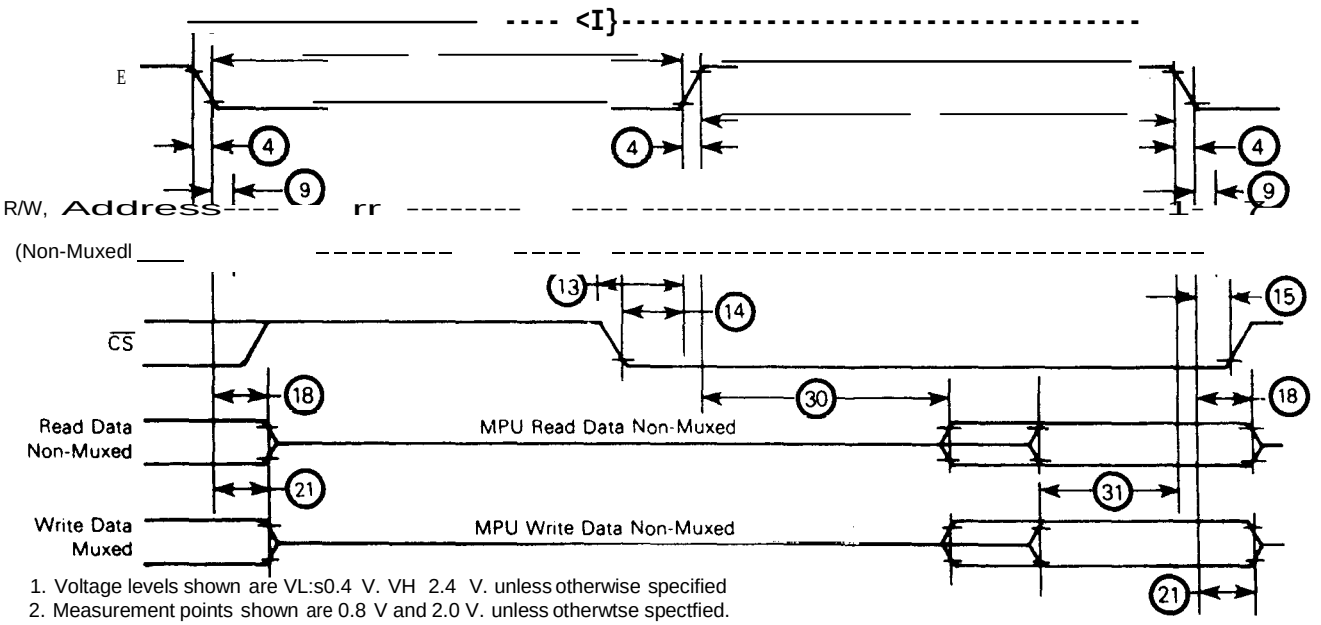
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage ol 2.0 volts, unless otherwise noted

BUS TIMING CHARACTERISTICS !See Notes 1 and 2 and Ftgure 7l

Ident. Number	Characteristic	Symbol	MC5860		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	'eye	1.0	10	0.67	10	0.5	10	jlS
2	Pulse Width, E Low	PWEL	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PWEH	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	tr. If	–	25	–	25	–	20	ns
9	Address Hold Time	'AH	10	–	10	–	10	–	ns
13	Address Setup Time Before E	'AS	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	tcs	80	–	60	–	40	–	ns
15	Chip Select ·Hold Time	'cH	10	–	10	–	10	–	ns
18	Read Data Hold Time	'DHR	20	50.	20	50.	20	50.	ns
21	Write Data Hold Time	IDHW	10	–	10	–	10	–	ns
30	Output Data Delay Time	IDDR	–	290	–	180	–	150	ns
31	Input Data Setup Time	'Dsw	165	–	80	–	60	–	ns

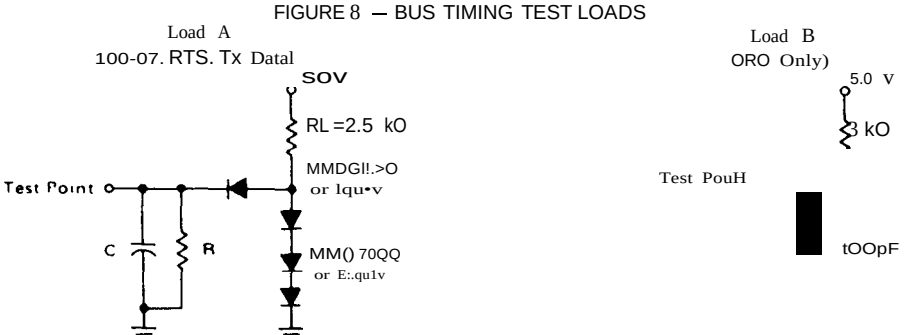
·The data bus output buffers are no longer sourcing or sinking current by tDHRmax (High Impedance!.

FIGURE 7 — BUS TIMING CHARACTERISTICS



1. Voltage levels shown are VL:s0.4 V. VH 2.4 V. unless otherwise specified
2. Measurement points shown are 0.8 V and 2.0 V. unless otherwtse spectfied.

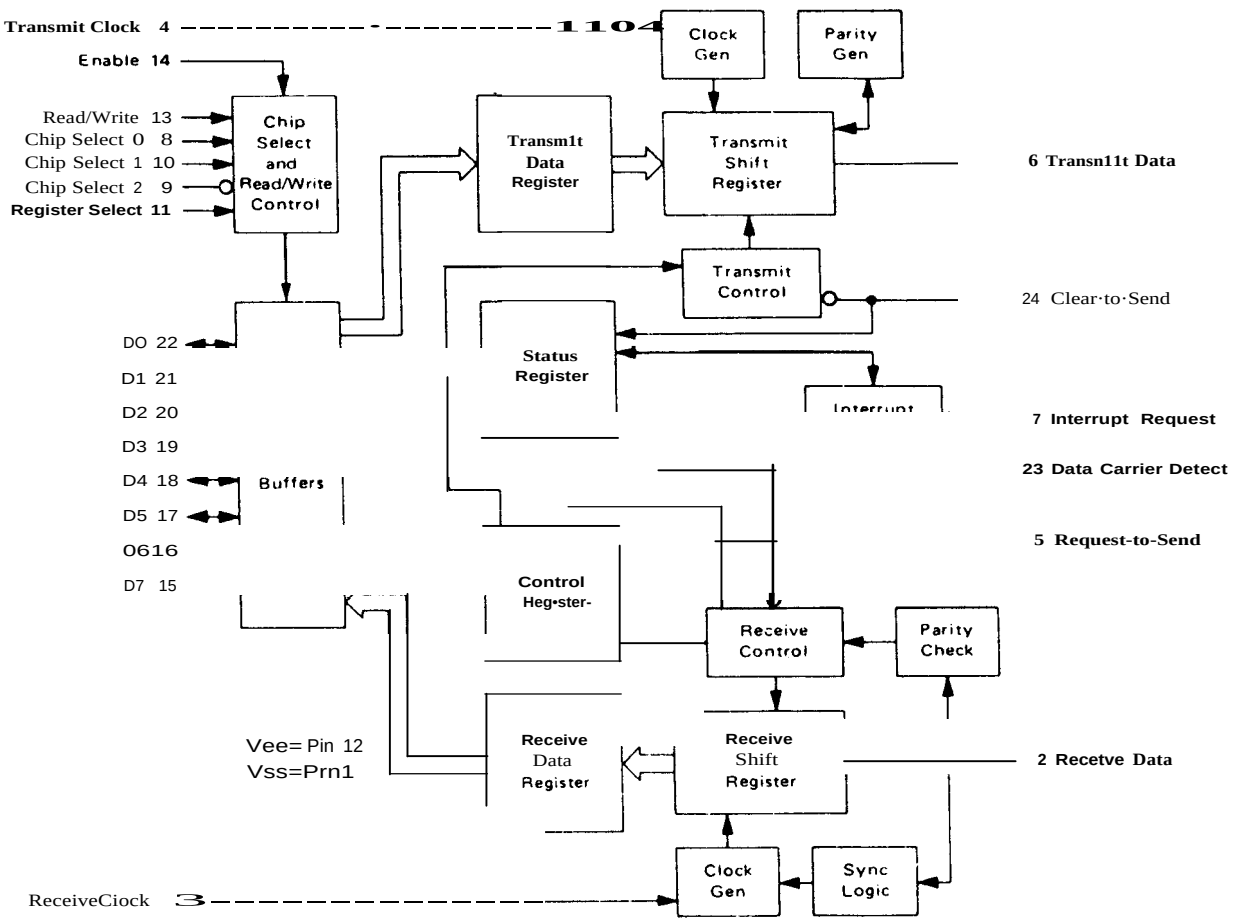
FIGURE 8 — BUS TIMING TEST LOADS



C - IJO pF to• DO 01
- 30 pF lor R"TS """ T' Dat.1

R a 11.7 kH for 00-07
- 24 kfl for RTs and Tx Data

FIGURE 9 — EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus mterface, the A CIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two wnte-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface cons1sts of serial input and output lines w1th independent clocks. and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset ICRO, CRII should be set during system rnrtialization to insure the reset conditron and prepare tor pro·g:amm1ng the ACIA functional configuration when the com·munr•:ations channel is required. During the first master reset. the IRQ and RTS outputs are held at level 1 On all other master resets, the RTS output can be programmed hrgh :)r low w1th the IRQ output held high. Control bits CR5 and CR6 should also be programmed to define the state of RTS whenever master reset rs utilized. The ACIA also con-tarns rnternal power-on reset !ogre to detect \he power line turn-on transition and hold the chip in a reset state to pre-vent erroneous output transitions prior to initialization This crrcuitry 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 programmabie 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 Ieven, odd, or none). etc,

TRANSMIT

A typical transmitting sequence consrsts of readrng the ACIA Status Register either as a result of an interrupt or rn the ACIA's turn in a pollrng sequence. A character may be written into the Transmit Data Register ⌈1 :he sratus read operation has indicated that the Transmit Data Rr:grster is empty. This character is transferred to a Shift Regrs:M where it rs serialized and transmitted from the Transrrrt Data outp,Jt preceded by a start bit and followed by one or two stop trrs. Internal panty (odd or even) can be optronally added to •fore character and will occur between the I<Jst dolo bit and rt·:e frrst stop bit. After the first character is wntten in the Da·,a Register. the Status Regrster can be read agarn ro check for a Transmrt Data Register Empty condition and currP.nt peripheral status. If the regrster rs ernpty, another character can be loaded for transmrssion even tt>ough the first character is in the process of being transmit\8d !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 Rece1ve Data input. A divide-by-one clock rat1o is prov1ded for an externally synchronized clock (to 1ts data) while the divide-by-16 and 64 ratios are provided for internal synchroniz:.H1on. Bit ;ynchronization in the divide-by-16 and 64 modes is in-Itiated by the detection of 8 or 32 low samples on the receive line in the div1de-by- 16 and 64 modes respectively. False start b11 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 receiv1ng sequence, the Status Reg1ster is read to determme if a character has been rece1ved from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit A CIA 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 (07 = 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 lS available in the Receive Data RP-glster. The receiver is also double buffered so that a character can be read from the data register as another character is being rece1ved 1n the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The A CIA interfaces to the M6800 M PU w1th an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/wnte line, and enable line. 1 hese s1gnals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (00-07) – The bidirectional data lines (DQ-07) 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 *o* high-impedance TTL-compatible input that enables the bus in-put/ output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 .*P2* Clock or MC6809 E clock.

Read/Write (R/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 reg1ster is read. When it is low, the ACIA output drivers are turned off and the MPU w1ites into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers with1n the ACIA.

Chip Select (CSO, CS1, CS2) – These three high-Impedance TTL-compatible input lines are used to address the ACIA. The ACIA is selected when CSO and CS1 are high and 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 h1gh·impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low !eve! the Control/Status Reg1sters. The ReadiWrite signal line is used in conJunction with Register Select to select tile 1ead-only or write-only reg1ster in each register pair.

Interrupt Request (IRQ) – Interrupt Request is a TTL-compatible, open-drain (no internal pullupl, act1ve low output that is used to interrupt the MPU The 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 IRQ status bit, when high, indicates the IRQ output is 1n the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter sect1on causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5•CR6l, and tt1e Transmit Data Reg1ster Empty (TORE) status bit is high. The TORE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send (CTSl being high or the ACIA being maintained in the Reset condition. The interrupt lS cleared by wntmg data mto the Transmit Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of CTS which inhibits the TORE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRFl status bit is high, an Overrun has occurred, or Data Carrier Detect (DCDl 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 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 InterrupEnable.

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 mav be selected.

Transmit Clock (Tx CLK) – The Transmit Clock input is used for the clocking of transmitted data. The transmitter in-Itiates data on the negative transition of the clock.

Receive Clock (Rx CLK) – The Receive Clock mput 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 Datal — 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 (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 tTDREl status bit.

Request-to-Send (RTSI — 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 RTS output is low (the active state). This output can also be used for Data Terminal Ready (OTRI.

Data Carrier Detect (OCD) — 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 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 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 (TORI

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the A CIA has been addressed with RS high and R/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 (TORE) bit to indicate empty.

RECEIVE DATA REGISTER IRDRI

Data is automatically transferred to the empty Receive Data Register (RDRI from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRFI 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/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/W Transmit Data Register	RS • RIW Receive Data Register	RS • RlW Control Register	RS • R/W Status Register
	IWrite Only)	IRead Only)	IWrite Only)	IRead Only)
0	Data B•t 0•	Oata81tO	Counter 01v1cie Select 1 (CROl	Recetve Data Aegtster Full (RDRF)
1	Data B•t 1	Data Bit 1	Counter Div1de Select 2 1CR 1)	Transrn•t Data Reg,ster Empty tTOREl
2	Data Bit 2	Data B•t 2	Word Select 1 ICR2l	Data Carr•er Oetect IOCOl
3	Data B•t 3	Data B1t 3	Word Select 2 ICR3l	Clear to Send ICTSl
4	Data Bit 4	Data Btt 4	Word Select 3 ICR4i	Frarnmg Error IF E I
5	Oata B•l 5	Data 81\ 5	Transn111 Control 1 ICR5l	R@ce•ver Overrun IOVRNI
6	Data Btt 6	Data 81t 6	Transmtt Control 2 ICR6l	Partty Error (PEl
7	Data B•t 7•••	Oata81t7''''''	Rece•ve Interrupt E nable l C A 7)	Interrupt Request iiROl

• Leadtng btt LSB Btl 0
•• Data bit wtll Ue *zero* .n 7 btt jjlus panty modes
••• Data bit is "don't care·· tn 7 btt plus panty rnoUes.

CONTROL REGISTER

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

Counter Divide Select Bits (CRO and CR1l – The Counter Divide Select Bits !CRO and CR1l determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these btts are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on CTS and DCD) and initializes both the receiver and transrrutter. 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 btts provide for the following clock divide ratios:

CR1	CRO	Function
0	0	+1
0	1	+16
1	0	+64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4l – 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 B1ts + Even Parity+ 2 Stop Bits
0	0	1	7 B1ts +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 B1ts + 2 Stop Bits
1	0	1	8 Bits+ 1 Stop Bit
1	1	0	8 Bits+ Even paritv+ 1 Stop Bit
1	1	1	8 Bits+ Odd Parity+ 1 Stop B1t

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

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control btts provide for the control of the tnterrupt from the Transmit Data Regtster Empty condition, the Request-to Send (RTSl output, and the transmission of a Break level !space) The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{\text{RTS}}$ =!ow, Transmitting Interrupt D1sabled.
0	1	$\overline{\text{RTS}}$ 1 <N,, Transmitting Interrupl Enabled.
1	0	$\overline{\text{ATS}}$= f11qh. Transm;ttmg Interrupt D1sabled.
1	1	$\overline{\text{RTS}}$ =-low, Transm1ts a Break leve: on the Transmit Data Output Transmitting Interrupt D1sabted.

Receive InterPJpt Enable Bit (CR7l The !ollowmg interrupts Will be enabled by a high level in btt positton 7 of the Control Register (CR7': Receive Data Register Full, Overrun. ;,Jr a low to-hih transition ')n the Data Carrier Detect (DCDl sigroall111e.

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/W is high. Information stored in this register tndicates 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 (RDRFl, Bit 0 – Receive Data Regtster 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 (TORE), Bit 1 – The Transmit Data Regtster Empty bit being set high tndtcates that the Tran.smit Data Regtster 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 (DCD), Bit 2 – The Data Carrier Detect bit will be high when the BCi:5 mput 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 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 DCD tnput remains high after read status and read data or master reset has occurred, the interrupt is cleared, the DCD status bit remains htgh and will follow the DCD input.

Clear-to-Send (CTS), Bit 3 – The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low 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 transmis-Sion, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, thts error indicator is present throughout the time that the associated character is available.

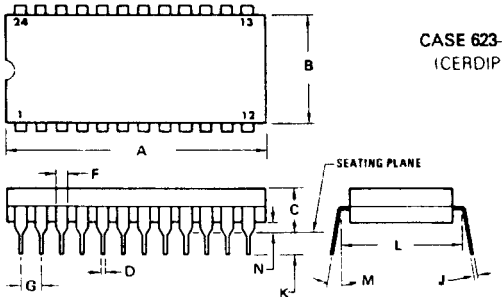
Receiver Overrun IOVRN). Bit 5 – Overrun ts an error flag that indicates that one or more characters tn the data stream were lost. That is. a character or a number of characters were received but not read from the Receive Data Register (RDRl prior to subsequent characters being received The '-'verrun condttton begtns at the midpoint of the last bit of the second character recetved in succession without a read of the RDR havtng occurred The Overrun does not occur tn 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 Re;,.::t.

Parity Error IPE), Bit 6 – The parity error flag indicates that the number of highs Iones) 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 indicatioP 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 ORO), Bit 7 – The IRQ b1t indicates the state of the IRO output Any inier:upt condition with its applicable enable will be indicated in this status b1t. Anytime the IRQ output is low the IRQ b1t will be high to indicate the Interrupt or service request status. IRQ IS cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

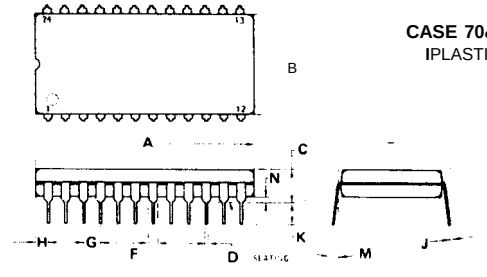
PACKAGE DIMENSIONS



CASE 623-03
(CERDIP)

	MILLIMEHRS		INCHES	
	041	051		
f	1.27	1.52	0.050	
G	1.548SC		O.IOOBSC	
J	0.10	0.30	0.008	0.012
K	2.29	4.06	0.090	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

NOTES
1. OIM T" TO CENTER Of LEADS WHEN FORMED PARALLEL.
2 LEADSIVITHINO.IJmm 10.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. IWHEN FORMED PARALLEL!



CASE 70&-02
IPLASTIC)

	J,IMETERS		IN_CH	
	MIN	MAX	TMN	
C			0.155,	0.200
0	036	056	1001	4100211
f	1.01	1.52	0 040	0.060
G	1.54 BSC		0 100 BSC	
H	1.65	1.03	0.065	
J	0.1Q	0.38	0.008	0 015
K	5 24		609	
N	051	101		

NOTES
1 POSITIONAL TOLERANCE Of LEADS 101. SHALL BE WIIHIN 0 15'<'m 100101 AT **MAXIMUM MATERIAL CONDITION IN** RELATION TO SEATING PLANE AND EACH OTHER
2 DIMENSION L TO CENHR Of LEADS WHEN fORMED PARALLEL
3 DIMENSION 8 DOES NOT INCLUDE MOLO f ASH