3. TIMER, INTERRUPT AND COMMUNICATIONS CONTROL (TICC)

In addition to the general purpose I/O ports of the system connector, there is an 8-bit parallel input port and a separate 8-bit parallel latching output port available on the 25-pin DEVICE CONNECTOR (see Section 6.1.5). Data transfer between these ports and the 8080 CPU is under the control of the Timer, Interrupt and Communication Control, TICC; hence these ports are referred to as TICC I/O ports. An 8-bit word present at the TICC input port can be transferred into the accumulator with the "LDIN" instruction, while the content of the accumulator can be transferred to the TICC output port with the "STOUT" instruction.
The TICC output port is inverting.
The serial input-output, the interrupt processing and the interval timers are also under the control of the TICC. This section discusses the TICC features and provides the background information that will enable the programmer to use the DCE as a microcomputer with powerful real-time processing capability.

3.1 SERIAL INPUT-OUTPUT

Serial output data from the DCE pass through the SOPOS, SONEG output pair, and the input serial data to the DCE pass through the SIPOS ans SINEG input pair located at the DEVICE CONNECTOR. Both pairs are optically isolated from the DCE circuits and are compatible with 20mA current loop system. The serial input is conditioned by a schmidt trigger that switches on at 10mA and off at 5mA, providing thus a high degree of noise immunity. The serial output is capable of switching 30mA.

The serial I/O can be connected in a passive mode or in an active mode. The passive mode is used when the external device supplies the current. In this case connection is shown in Figure 3-1.

The active mode is used when the external device does not supply the current which is usually the case with the teletype. For this case the connection to the teletype is shown on Figure 3-2. Section 3. 1.2 describes the necessary teletype modifications.

The serial receiver register of the TICC detects the start and stop bits of an incoming character and places it into the receiver buffer. At this time the internal "receiver-buffer-loaded" flag goes high indicating to the CPU that the receiver is loaded with a new character. If the interrupt-mask-register does not mask this particuler input off, and if no higher priority interrupts exist at the time, this flag will interrupt the CPU and cause it to enter the "read-receiver-buffer" routine, at the vector address 0020H, which transfers the buffer contents into the accumulator and processes the data. The "receiver-buffer-loaded" flag is reset automatically when the receiver buffer is read.

Before serial transmission of output data, the program must send the control word to specify the BAUD rate and number of stop bits desired (see Section 4. 4. 3). The program provides this facility by transferring an 8-bit code from the accumulator to the TICC rate register by the instruction "STCRR". The code in the TICC register defines the BAUD rate and the number of stop bits. Once the BAUD rate is set, the instruction "STXMT" loads the character to be transmitted into the transmit buffer. The serial transmitter register, upon receiving the character from the buffer, generates the "start" and "stop" bits and shifts the data out at the programmed rate, independently of the CPU.

While the transmitter is shifting out the serial data, the CPU can be executing another program function. At the time that the transmit-buffer goes empty and the TICC is ready to receive the next character from the 8080 CPU, it will set the "transmit-buffer-empty" flag "high" and force an interrupt to occur. The program can then load the next character into the TICC transmit buffer and return to execution of the program function that was in progress before transmit-buffer-empty was signalled.

3.1.1 TICC I/O Conditions and Characteristics

PARAMETER	MIN	TYP	MAX	CONDITION
High level input voltage, V_{IH}	3.3v		6v	
Low level input voltage, V_{IL}	-1v		0.8v	
Input current. I_I			±10µA	V_I = 0 - 5v
High level output voltage, V_{OH}	3.7v			I_{OH} = 400µA
Low level output voltage, V_{OL}			0.45v	i_{OL} = 1.7mA
Interrupt pulse width, t_W	500ns			
Interrupt pulse spacing	500ns			
SERIAL OUT				
Breakdown voltage, $V(BR)_{CEO}$			30v	
Collector leakage, I_{CEO}			100nA	V_{CE} = 10v
Saturation voltage, $V_{CE}(SAT)$		1.2v		
Rise and fall times, t_R, t_F		80µsec		V_{CE} = 10v I_C = 10mA
SERIAL IN				
Forward current, $I_F(MAX)$			20mA	continuous
Reverse voltage, BV_R	3v			
Forward voltage, V_F			1.5v	I_F = 20mA
Turn on current, $I_F(ON)$		10mA	15mA	
Turn off current, $I_F(OFF)$	2mA	5mA		
Rise and fall times, t_R, t_F		400ns		

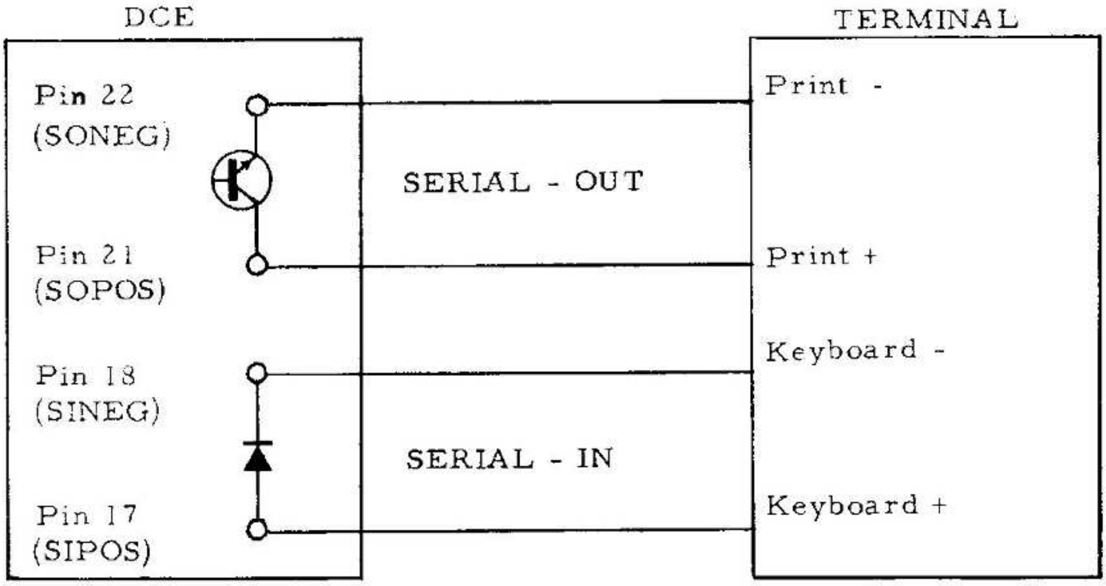

Figure 3-1 Terminal Connection, Passive Mode

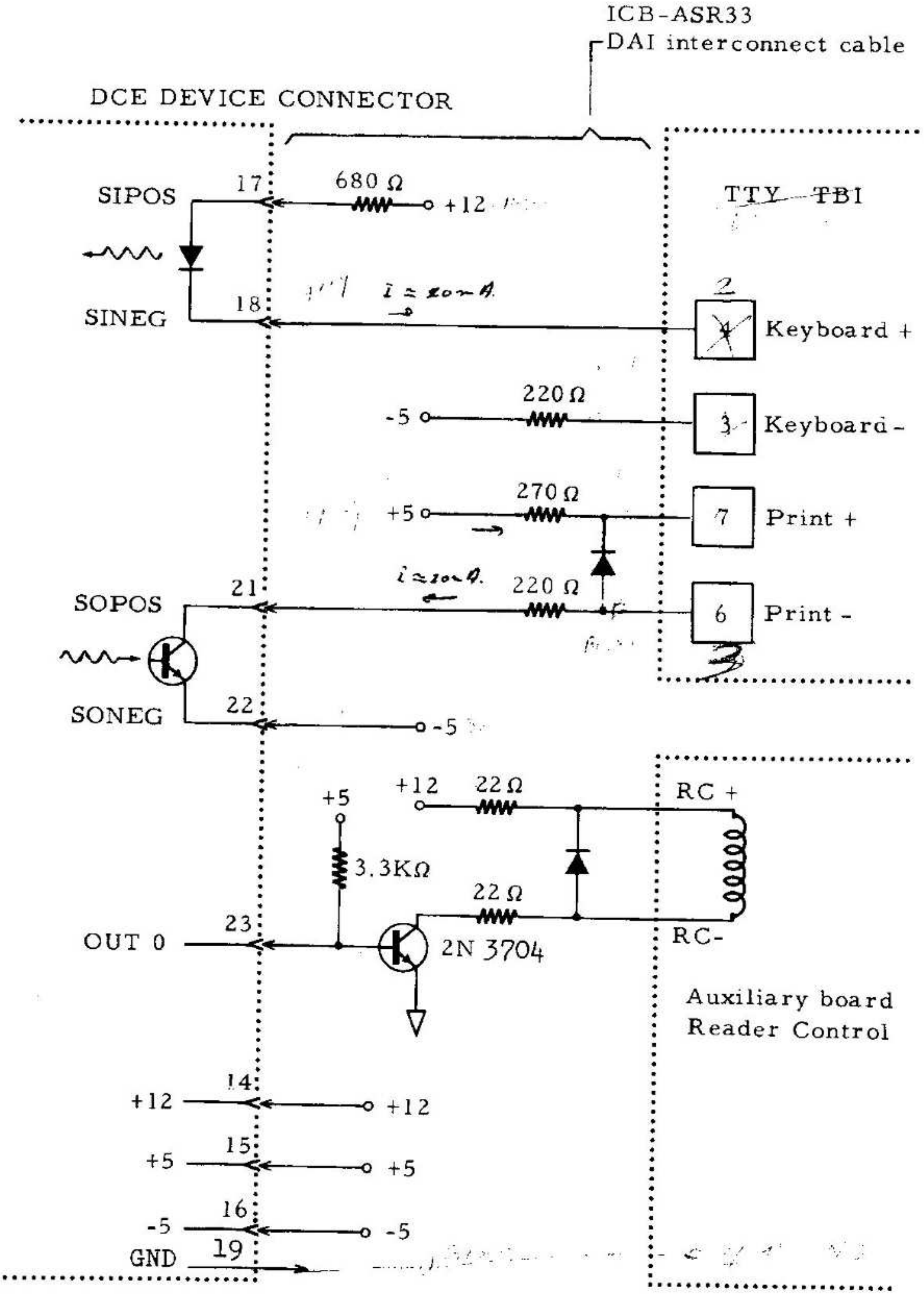

Figure 3-2 Teletype Connection, Active Mode

3.1.2 Teletype Modifications

The 20mA serial I/O of the DCE can be used directly as teletype interface. The user must connect the keyboard and the teleprinter to the DCE, if a 33ASR teletype is used,for communications with the microcomputer. For this connection a slight modification of the MODEL 33ASR is necessary. This modification procedure is detailed in the following two steps:

1. Change the factory-wired 60mA mode to 20mA mode by changing the resistor taps from the existing (factory tapped) at 750ohm position to the end (1450ohm) position in the electrical service unit. Push-on taps are provided, and no soldering is required. Refer to Figure 3-3 on page 3-7 for the location of the resistor. Figure 3-4 shows the location of the taps.

   In addition, a single wire on the barrier terminal strip must be moved at the rear of the electrical service unit. Unscrew the violet wire from terminal 8, and move it to terminal 9. Refer to figures 3-4 and 3-5 on page 3-7.

2. Change the factory-wired half duplex mode to full duplex mode by moving both the blue-white and the brown-yellow wire to terminal number 5. These leads are factory-wired to terminal 4 and 3 respectively. Refer to figures 3-5 and 3-6 on page 3-7.

Certain applications may require the DCE to control the tape reader drive. This can easily be accomplished by installing a simple relay circuit into the electrical service unit of the 33ASR. Realize the relay circuit on a small vector board with a Potter-Brumfield (No. JR-1005) relay, a carbon composition (470 ohm, 0.5w) resistor, and a (0.1μF, 200 vdc) capacitor as shown on figure 3-7 on page 3-8. Mount this auxiliary circuit on the vertical metal tab as shown on figure 3-8 on page 3-8.

Figures 3-7 and 3-9 show how to wire the auxiliary board into the electrical service unit. Splice into the brown wire and connect a lead from the splice to one side of the JR-1005 relay. Connect another wire from the splice to the LOCAL position of the MODE switch. Connect the LINE position of the mode switch to the other side of the JR-1005 relay. Identify the LOCAL and LINE terminals referring to figure 3-10. Figure 3-2 on page 3-4 shows how to interconnect the DCE to the modified teletype.

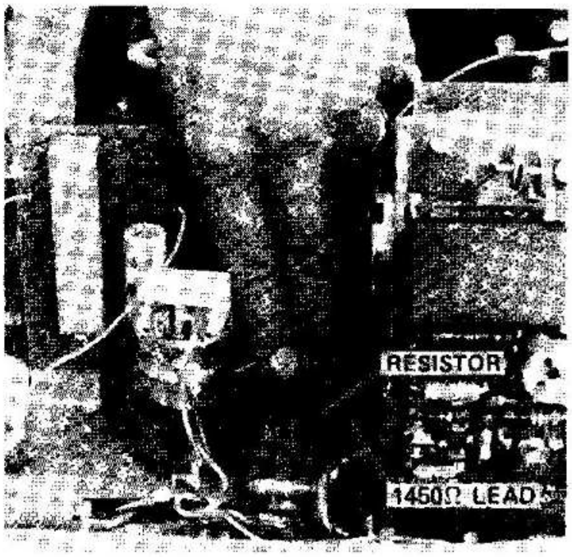

Figure 3-3 Current Source Resistor

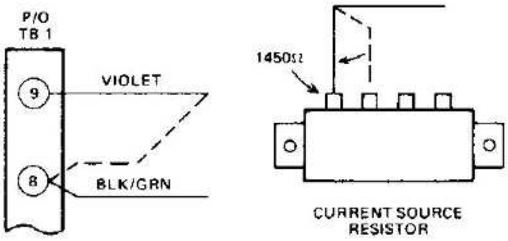

Figure 3-4 TTY: 20mA Wiring

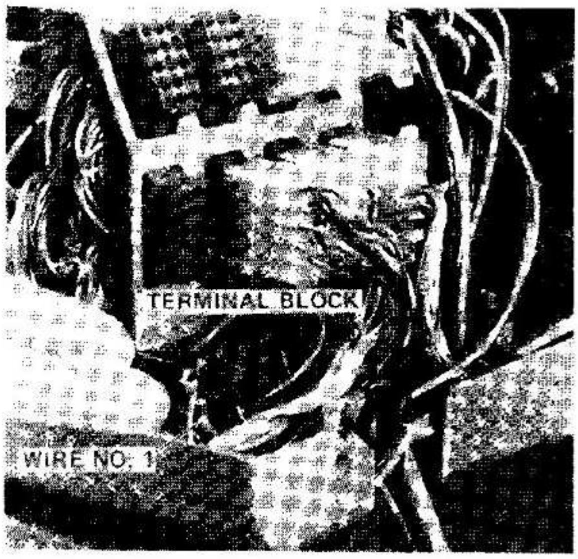

Figure 3-5 Terminal Block Location

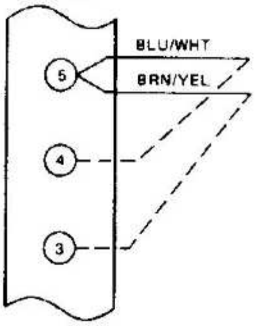

Figure 3-6 TTY: Full-Duplex Wiring

Pictures reprinted from Intel's Intellec 4 reference manual for reference.

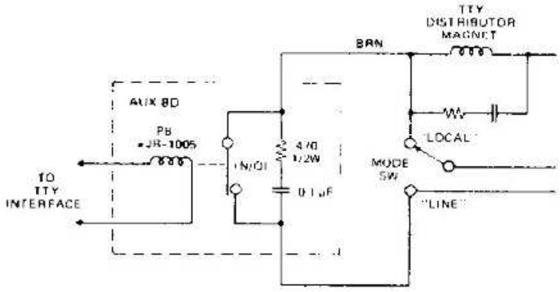

Figure 3-7 Reader Control Auxiliary Board Schematic

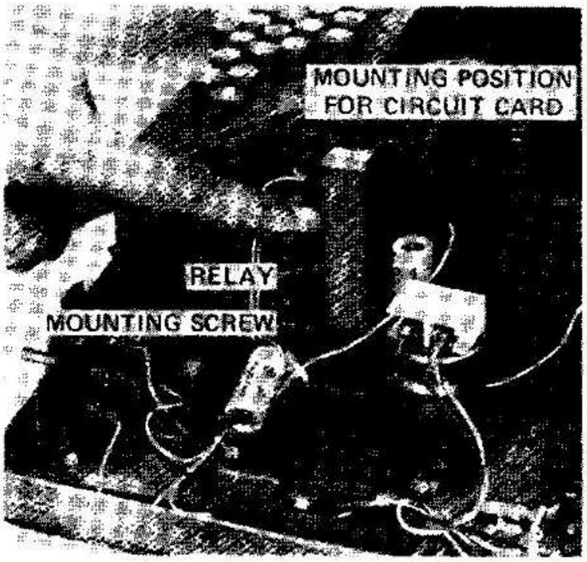

Figure 3-8 Relay Board Location

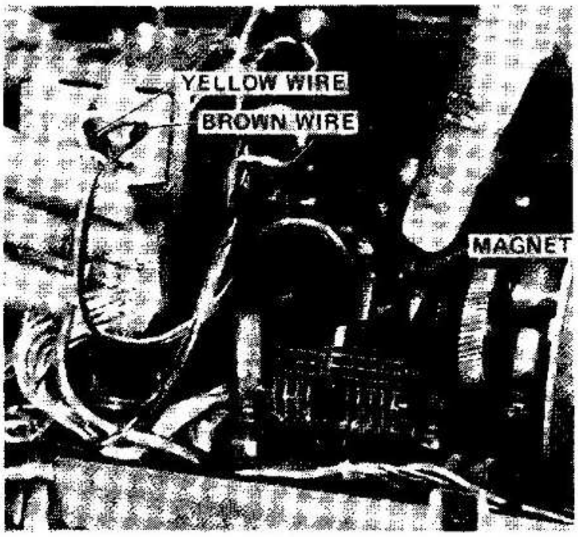

Figure 3-9 ESU Wiring

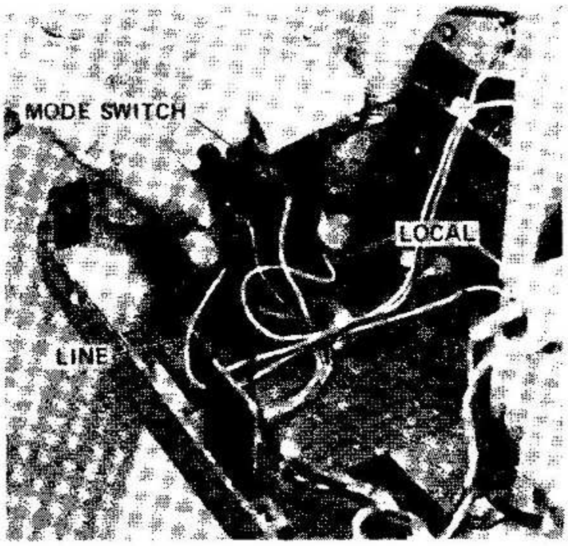

Figure 3-10 Teletype Mode Switch

Pictures reprinted from Intel's Intellec 4 reference manual for reference.

3.2 INTERRUPTS (8-bit DCE)

Interrupt processing ease is one of the most powerful features of the DCE that allows maximum utilization of the CPU. Complex interrupt structures can be established by software that tailors the response of the DCE to customized circumstances under the control of the microcomputer. For example, several peripheral sensors or devices having different priority levels can share DCE execution time. The programmer has the flexibility to use the interrupt control functions to establish priority, mask out undesired interrupts, or generate interrupt requests to deal with high priority conditions. For example, a vectored interrupt is generated when any of the five interval timers count down to zero. If any of the peripheral devices can accept data only at certain time intervals, the program can load one of the timers with the number corresponding to the time interval. While the CPU is processing the main program, the timer counts down. At zero, the CPU is made to branch to the routine that outputs the next data byte, and then it returns to the execution of the main program. Re-loading the timer can be part of the output routine to ensure continuous data output at the desired time intervals.

The programmer can selectively enable desired interrupts by loading a control word into the Interrupt Mask Register from the accumulator with the "STIMR" macro instruction. A logical ONE at bit position 'n' of this register allows interrupt 'n' pass through to interrupt the executing program and branch to the vector address specified in Section 3.2.3. The PROM (or RAM) memory at the address specified must contain the start of the interrupt service routine which saves the CPU registers (if necessary), performs the required processing functions, restores CPU registers, enables CPU interrupts and returns to the previous program. Interrupt requests disabled via the interrupt mask register are also latched in the TICC interrupt register. If the corresponding bit in the interrupt mask register is set subsequently, the interrupt request will become active. All the 8 interrupts are handled by the TICC.

The programmer can choose two different ways to configure the DCE for servicing an interrupt; he can set up an interrupt driven system or a polled interrupt system.

3.2.1 Interrupt Driven System (see Figure 6-1, 6-2)

Conditions: a) CPU interrupts are enabled
b) Bit 3 of TICC Command Register is set
c) Interrupt Mask Register corresponding to interrupt is set.

The following is the sequence of events when an interrupt occurs:

1. The Interrupt Register latches the interrupt condition and the corresponding bit remains set until the interrupt is serviced (i. e. the TICC transfers the corresponding RST instruction to the CPU).
2. TICC sets bit 5 of TICC Status Register high to indicate interrupt pending.
3. TICC sends an INT signal to CPU.
4. CPU acknowledges interrupt with an INTA signal.
5. TICC transfers the RST instruction of the highest pending interrupt to the CPU and clears the corresponding bit in the Interrupt Register.
6. CPU loads the program counter with the RST address and program execution starts at the interrupt service routine.

3.2.2 Polled Interrupt System (see Figure 6-1, 6-2)

Conditions: a) CPU interrupts are disabled
b) Bit 3 of TICC Command Register is cleared
c) Interrupt Mask Register bit corresponding to interrupt is set.

In this system an interrupt condition will not interrupt the CPU.

The following is the sequence of events:

1. The Interrupt Register latches the interrupt condition and the corresponding bit remains set until the CPU reads the interrupt address register.
2. The TICC sets bit 5 of TICC status register high to indicate interrupt pending.
3. TICC sends an INT signal to CPU. The CPU does not acknowledge the INT signal since its interrupt is disabled. (see note)*
4. When the program is ready to accept an interrupt, it must poll the interrupt pending bit of the TICC Status Register (bit 5).
5. If the interrupt pending bit is set, program must read the interrupt address register, where the address of the highest priority pending interrupt is stored. This reading clears the interrupt register bit for that interrupt. When the RST instruction in the interrupt address register is executed, the corresponding service routine is entered (can be done by loading program counter with 9802H via H, L registers).

*NOTE:

If the CPU interrupts are enabled in this mode the CPU will acknowledge the interrupt but the TICC will not release the contents of the interrupt address register. Since the CPU bus in this condition is not driven, it will float high, and the CPU sees an RST 7 (FF) instruction.

An important feature of the Interrupt Register is that it is edge triggered. A low to high transition on the external interrupt, for example, sets bit 2. Servicing the interrupt clears this bit even if the external interrupt input stays at logical ONE. It is, therefore, not necessary to employ external hardware to remove the request.

Regarding point 5 above, it must be noted that the CPU can only read the interrupt Address Register if bit 3 of the TICC Command Register is zero. If this bit is set, any attempt to read the Interrupt Address Register results in a reading of FFH, and pending interrupts are not cleared

Disable CPU interrupts whenever changing the contents of the TICC interrupt mask register to prevent spurious interrupts due to transients.

The lowest priority interrupt, number 7 (see Figure 6-1) can be generated by either the fifth interval timer or by the auxiliary interrupt IN7 (bit 7 of the TICC parallel input port). The programmer can select Timer 5 or IN7 by setting Bit 2 of the TICC Command Register to zero or one respectively.

The power-on reset and the Timer 1 interrupt both set the program counter to zero. When Timer 1 is being used, the program can determine the source of interrupt 0 by looking at an unused bit of GIC Port 2, for example Bit 4. If this bit is connected to +5v through a resistor, after power-on it will be read as a "one" since all GIC ports will go into the input mode. If the initialization program configures GIC Port 2 Bit 4 as output and sets it to zero, it will remain at zero until the next power-on reset. The program starting at address 0000 can therefore read P2B4 and jump to the initialization routine or Timer 1 service routine accordingly. (note: this technique cannot be used in multiprocessor systems since they use P2B4)

3.2.3 Interrupt Vector Assignments (see Figure 6-2)

INTERRUPT NUMBER	SOURCE	VECTOR ADDRESS (HEX)
0	Reset or Timer 1 has expired	0000
1	Timer 2 has expired	0008
2	EXINTR low to high detected	0010
3	Timer 3 has expired	0018
4	Serial receiver has received a full character	0020
5	Serial transmitter is ready to accept next character	0028
6	Timer 4 has expired	0030
7	Timer 5 has expired or IN7 low to high detected	0038

3.3 INTERVAL TIMERS

The five programmable timers of the DCE provide program controlled time intervals that are used to invoke a subroutine that performs a servicing function for a peripheral device or event that requires periodic recognition. These timers are 8-bit counters, which provide intervals that vary in duration from 64 to 16,320 microseconds. Much longer intervals can be generated by cascading the timers in software.

To start a time count using any of the five timers, the program can load the required interval by an instruction with general format "STTIM n". For example, the instruction "STTIM 2" loads the contents of the accumulator into Timer 2. Loading the timer activates it, and it starts counting down increments of 64 microseconds at the first TICC clock pulse. The first count period can be any length between 0 and 64 microseconds. When the count reaches zero, the bit in the interrupt register corresponding to the timer is set, and an interrupt is generated if CPU interrupts are enabled and that mask bit is set. The timer then remains inactive until a new interval is loaded. Loading an interval value of zero, causes an immediate interrupt. A new value loaded while the interval timer is counting overrides the previous value and the timer starts counting down the new value.

To maintain continued accuracy, a timer has to be reloaded within 64 microseconds after firing. If a series of timed interrupts with crystal accuracy is required, the timer interrupt service routine must reload that timer within 64 microseconds.

Note: None of the TICC command registers used for giving control information or data to the TICC (eg. interrupt mask register) can be read back. Therefore, copies of the contents of these registers should be maintained, if required.

3.4 TICC STATUS AND COMMANDS

The DCE allows the programmer to poll the status of the TICC with respect to external conditions and issue discrete commands to deal with specific circumstances. These actions can be executed via the TICC Status Register, the TICC Command Register, and the Communications Rate Register described in this section.

3.4.1 TICC Status Register (see Figure 6-2)

To query the status of the TICC, the program must transfer the contents of this register to the 8080 CPU and examine its contents bit by bit. The status conditions associated with each bit are described in the following paragraphs and summarized in Sections 3.5.

Bit 0, Framing Error

A logical ONE in this bit position indicates that one or both stop-bits were in error, which implies that the serial data received and held in the receiver buffer is probably erroneous. If this flag is high and the bits of the receiver-buffer-register are all logical ONE, then a break in the serial communications line has occurred. This flag will remain high until the next valid character is received.

Bit 1, Overrun Error

A logical ONE in this bit position indicates that a new character was loaded into the receiver buffer before a previous character was read out by the program. A new read-receiver-buffer command, "LDRCV" or a reset command clears this flag.

Bit 2, Serial Input, Direct Connection

This bit is directly reading the serial input line and permits bypassing the serial receiver register. This feature can be useful if the BAUD rate of the incoming character is not standard. In this case the programmer can handle the non-standard BAUD rate with software and one of the interval timers. Bit 2 can also be used for test purposes or detecting a break; it is normally "high" when no data is being received and line current is flowing.

Bit 3, Receiver Buffer Loaded

A logical ONE in this bit indicates that the receiver buffer is loaded with a new character. A read-receiver-buffer instruction, "LDRCV" or the reset function clears this flag.

Bit 4, Transmitter Buffer Empty

A logical ONE in this bit indicates that the transmitter-buffer is empty. This condition occurs as soon as the buffer register transferred the data to the serial transmitter (which may still be in process of shifting out a character at the time the CPU polls bit 4). The reset function sets this flag high.

Bit 5, Interrupt Pending

A logical ONE in this bit indicates that one or more of the enabled interrupt conditions has occurred. This is the bit polled by the CPU in the polled-interrupt method described in Section 3.2.2

Bit 6, Full Bit Detected

A logical ONE in this bit indicates that the first data bit of an incoming character has been detected in the serial receiver register. This bit, provided for test purposes, will remain high until the entire character has been received. A reset instruction clears this flag.

Bit 7, Start Bit Detected

A logical ONE in this bit indicates that the start bit of an incoming character has been detected in the serial receiver register. This bit, provided for test purposes, will remain high until the entire character has been received. A reset instruction clears this flag.

3.4.2 TICC Command Register (see Figure 6-2)

The programmer can issue discrete commands to the TICC through the Command Register. Desired conditions can be encoded in the accumulator and transferred into the Command Register with the "STTCM" instruction. Here it will remain latched, with the exception of bit 0, until a new command arrives. The conditions associated with each bit of the Command Register are described in the following paragraphs.

Bit 0, Reset

A logical ONE transferred to this bit position will cause the following:

1. Clear the receiver buffer, the receiver register, the receiver-buffer-loaded flag, the start-bit-detected flag, the full-bit-detected flag, and the overrun-error flag.
2. Set the transmitter-buffer-empty flag high, indicating that the transmitter buffer is ready to accept a character from the CPU.
3. Clear the interrupt register except for the bit corresponding to the transmitter-buffer-interrupt, which is set to high
4. Inhibit the interval timers
5. Set transmitter data output high (marking).

The TICC Command Register does not latch a logical ONE transferred to bit 0, and therefore this does not have to be explicitly cleared; it is normally at logical ZERO

The reset function has no effect on the output ports, the external inputs, the interrupt acknowledge enable, the mask register, the rate register, the transmitter buffer and the transmitter register

Bit 1, Break

A logical ONE in this bit sets the serial output to the high impedence or non-conducting state (spacing). The reset function of bit 0 overrides this function. Normally this should be at logical ZERO.

Bit 2, Interrupt 7 Select

A logical ONE in this bit selects the auxiliary interrupt input at bit 7 of the parallel input port of the DEVICE CONNECTOR as input source. A logical ZERO selects the Interval Timer 5.

Bit 3, Interrupt Acknowledge Enable

A logical ONE in this bit enables the TICC to accept an interrut-acknowledge-decode from the CPU and transfer the RST instruction as a response. This condition sets up an interrupt driven system discussed in Section 3.2.1. A logical ZERO in bit 3 causes the TICC to ignore the interrupt-acknowledge-decode, a condition that results in a polled-interrupt-system, discussed in Section 3.2.2.

Bits 4 - 7

These bits are always ZERO, used only for DCE test procedures.

3.4.3 Communication Rate Register (see Figure 6-2)

This register allows the programmer to select any of the 7 standard BAUD rates for serial data transmission and reception. A logical ONE in any of the bits 0 through 6 of this register will select the BAUD rate for both the transmitter and the receiver as defined below:

Bit 0	110 BAUD
Bit 1	150 BAUD
Bit 2	300 BAUD
Bit 3	1200 BAUD
Bit 4	2400 BAUD
Bit 5	4800 BAUD
Bit 6	9600 BAUD

If more than one bit is high, the highest of the rates indicated will result. If all six bits are low, both the receiver and the transmitter will be inhibited.

Bit 7 allows the selection of stop bits to be generated by the transmitter register. Logical ONE in bit 7 selects one stop bit, while logical ZERO selects two stop bits.

The programmer can encode the desired BAUD rate and number of stop bits in the accumulator and transfer this code into the rate register with the "STCRR" instruction. This register holds the code until it receives a new command.

3.5 TICC REGISTER SUMMARY

REGISTER ADDRESS	REGISTER NAME	DESCRIPTION	MACRO INSTRUCTION	FUNCTION
9806	TRANSMIT BUFFER	Asynchronous serial transmitter buffer	STXMT	(TRANSMIT BUFFER)←(A)
9800	RECEIVE BUFFER	Asynchronous serial receive buffer	LDRCV	(A)←(RECEIVE BUFFER)
9807	OUT PORT	Parallel output port	STOUT	(OUT PORT)←(A)
9801	IN PORT	Parallel input port	LDIN	(A)←(IN PORT)
9809	TIMER 1	Interval timer 1	STTIM 1	(TIMER 1)←(A)
980A	TIMER 2	Interval timer 2	STTIM 2	(TIMER 2)←(A)
980B	TIMER 3	Interval timer 3	STTIM 3	(TIMER 3)←(A)
980C	TIMER 4	Interval timer 4	STTIM 4	(TIMER 4)←(A)
980D	TIMER 5	Interval timer 5	STTIM 5	(TIMER 5)←(A)
9808	INTR MASK	Interrupt mask register "1" in bit n enables interrupt n	STIMR	(INTR MASK)←(A)
9803	TICC STATUS	Status register:	LDSTA	(A)←(TICC STATUS)
	BIT 0:	1: Receiver framing error (stop bit(s) in error) 0: No framing error on last character		
	BIT 1:	1: Receiver overrun error (new character received before last one read) 0: No overrun error		
	BIT 2:	0/1: Serial input - direct connection		
	BIT 3:	1: Receive buffer loaded reset to 0 by LDRCV or system reset		
	BIT 4:	1: Transmit buffer empty reset to 0 by STXMT or system reset		
	BIT 5:	1: Some enabled interrupt is pending 0: No enabled interrupts pending		
	BITS 6, 7	Used for DCE test proceedings		
9804	TICC COMMAND	Command register	STTCM	(TICC COMMAND)←(A)
	BIT 0:	1 to reset TICC		
	BIT 1:	1 to send break on serial output-direct connection		
	BIT 2:	1 to select IN7 as source of interrupt 7 0 to select TIMER 5		
	BIT 3:	1 to enable TICC generation of 8080 interrupts		
	BIT 4 - 7	0 always (used for DCE test procedures)		
9805	RATE REGISTER	Communication rate register	STCRR	(RATE REGISTER)←(A)
	BITS 0 - 6	All zero: inhibit serial communications ONE on: select 110, 150, 300, 1200, 2400 4800, or 9600 BAUD by bit 0,1,2,3,4,5 or 6 on respectively		
	BIT 7	1 for one stop bit 0 for two stop bits		
9802	INTER PEND	Interrupt pending register	LDIRP	(A)←(INTR PENDING)
	BITS 5,4,3	Number of highest pending interrupt		
	BITS 7,6,2,1,0	Always 1		

3.6 ASCII CODE TABLE

USASCII (USA Standard Code for Information Interchange)

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