TANDBERG

PROJECT 296 - TARTAR II

PRELIMINARY HARDWARE

DESCRIPTION

CONTENTS:

INTRODUCTION

1.1. Scope

The scope of this document is to serve as an aid for programmer. It is not ment to be a complete users manual or hardware description.

1.2. References

Reference is made to:

- "Intel 8080 Assembly Language Programming Manual"
- "Intel 8080 Microcomputer Systems User's Manual" (july -75)

1.3. General view of device codes

Available codes:	Nr.	Occupied: By.
00 → 0F	00 → 0F	Cluster intf. I
10 → 1F		
20 → 2F	20 → 27	Cassette intf.
30 → 3F	30 → 37	Floppy disk. intf.
40 → 4F	40-41-46-47	Syncron. intf.
50 → 5F	50-51-52	Card reader
60 → 6F	60 → 6F	Config. ports (synch. intf
70 → 7F	70 → 7F	" " "
80 → 8F	80 → 8F	Cluster intf. II
90 → 9F	-	
A0 → AF	-	
B0 → BF	-	
C0 → CF	-	
D0 → DF	-	
E0 → EF	E4-E5-E6-E7	Display logic
F0 → FF	F4-F5	CPU-card.
	F6-F7	Display logic

2. HARDWARE DESCRIPTION

2.1. General layout

A block diagram is shown in figure 2.1. The figure shows a maximum configuration. Each unit connected to the Generel Motherboard Bus are complete functional modules plugged in as required for the application.

The Display logic board requires two slots in the rack. All other units are on one single board.

The system can be a "teletype-replacement" unit, with only the units within the frame of broken lines in figure 2.1 connected.

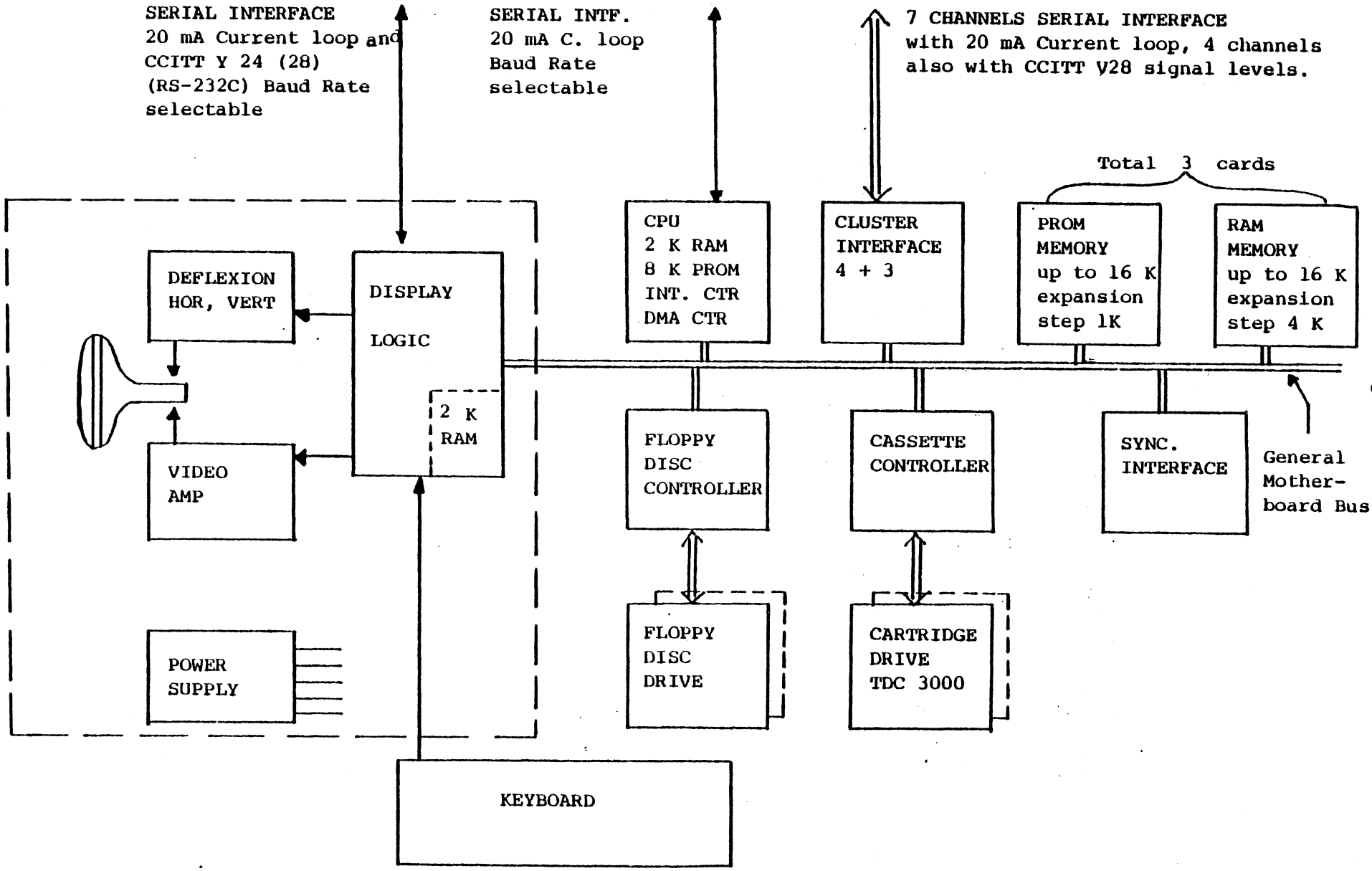


Figure 2.1 Block Diagram

2.2. Display_Logic_description

The Display logic boards contain all electronics necessary to drive a "teletype-replacement" data display. In addition it interfaces to the CPU on a parallell I/O basis.

The specifications for the "teletype replacement" data display is

2000 characters on a 25 line by 80 char. page.
95 displayable characters. (128 option).
7x9 dot matrix with underscore.

Baud rates; 110, 300, 600, 1200, 2400, 4800, 9600 and 19200.

Full Modem Interface (V.24 and RS232C) and current loop (20-60mA).

Video modification with underline (all characters visible) or with an attribute byte (attribute character displayed as space). The modification can be underline, blink, invisible and inverted video. Inverted video also switch selectable.

Cursor control characters and direct addressable cursor (from computer and/or keyboard).

Several keyboard layouts available (ECMA, TTY, Norwegian Standard and special system keyboards).

The specifications for the parallell I/O interface to the internal CPU is:

Device Code	Input	Output
E4	UART DATA	UART DATA
E5	RAM READ	RAM WRITE (with CTRL-decoding)
E6	KEYBOARD DATA	RAM WRITE (without CTRL-decodi
E7	INTERFACE STATUS	INTERFACE COMMANDS
F6	INTERRUPT STATUS WORD	
F7	UART STATUS	

The bits of the UART STATUS byte are (all active 1):

bit	Meaning	
0	TBMT	Transmitter Buffer
1	DA	Received Date Available
2	TMBT-ENFF	TBMT - Interrupt Enable Flip Flop
3	PE	Parity Error in received byte
4	OR	Overrun Error, the byte before this was lost
5	FE	Framing Error, this byte did not have stop bit

The bits of the interrupt status word, all active 1:

bit	Meading	
0	EN-TBMT	Enabled TMBT - request
1	DA	Received Data Available
2	KBDA	Keyboard Data Available
3	BSYINT	Busy Interrupt, Display logic was not able to perform the write command.
4	MEMBSYINT	Memory Busy Interrupt. The RAM is busy. Read operation is not performed.
5	CI	Calling Indicator; from outside world.

The bits of the interface status word (All active 0):

bit	Meaning	
0	$\overline{\text{CON}}$	Connect data set to line
1	$\overline{\text{DSR}}$	Data set Ready
2	$\overline{\text{RQTS}}$	Request to Send
3	$\overline{\text{RFS}}$	Ready for sending
4	$\overline{\text{CAR}}$	Carrier
5	$\overline{\text{CI}}$	Calling indicator

The bits of the interface command (all active 1):

bit	Meaning	
0	CPU LINE	Switch the CON flip flop
1	CPU TRANS	Switch the RQTS " "
2	CPU CLEAR	CLEAR the indication LEDs L4 (ERROR), L5 (ENQ), L6 (ACK) and L7 (NAK)
3	TBMTENSW	Switch the TBMT - Interrupt Enable flip flop

The meaning of the keyboard data byte is the ASCII-code where bit 0 to bit 6 corresponds to bit 1 to bit 7 in the ASCII-code. Reference is made to the ASCII-table for TDV 2100.

When writing into RAM-memory in the data display one may use two different device codes. When using E5 the logic will decode the byte and either write into RAM (cursor counter as pointer) or perform the decoded function. When using E6 the logic will write data into RAM and perform no decoding (cursor as pointer). After having written the cursor counter will advance one position to the right, from 79. position to 0. position on the line below or when in last line (line 24) roll one line up with cursor in position 0.

When reading from the RAM we also use cursor counter as pointer to where to read. After having read the cursor counter advances one position to the right, from 79. position to 0. position on the line below or from 79. position on the last line to 0. position on line 0.

UART data uses bit 0 to bit 7 (bit 7 is MSB). Normally the UART will add parity bit and start/stop bits. Then bit 7 is not in use.

One of the functions the display logic perform is Cursor Counter load. This sequence is started by giving the code 10(H) to the logic (output port E5). Then the next byte out will be loaded into the cursor line counter and the second byte will be loaded into Cursor character counter. When loading cursor the CPU must not go outside the limits 0 to 24 for the line counter and 0 to 79 for the character counter.
The Erase Page function will erase from Home position, and then place the cursor in the Home position. This takes max. 1,4 ms.

The Erase Line function will erase the whole line and place the cursor in character position 0. This takes max. 100 µs.

An attribute byte (bit 7 = 1) is either written directly into RAM using port E6 or by setting the underline flip flop (forces bit 7 to a 1) first and then writing the attribute byte (the underline flip flop must be reset by the normal code, 0F).

The attribute uses bit 4, 5 and 6 in the memory, meaning:

bit 654	Meaning
000	NORMAL
001	NORMAL
010	LOW INTENSITY
011	BLINK
100	INVERTED VIDEO
101	UNDERLINE
110	INVISIBLE
111	NORMAL

The attribute byte will decide how to present the information on the CRT from the attribute byte to the end of the page.

The display logic may use two interrupts.

Interrupt level 1: Real time clock, one interrupt every 19.9857954 ms.

Interrupt level 3: General request for service. Read the interrupt status word (F6).

ASCII-table for TDV 2100

b_7					0	0	0	0	1	1	1	1
b_6					0	0	1	1	0	0	1	1
b_5					0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1	Row \ Column	0	1	2	3	4	5	6	7
0	0	0	0	0		CURSOR LOAD	(SPACE)	0	@	P	`	p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2	VIDEO OFF		"	2	B	R	b	r
0	0	1	1	3	VIDEO ON		£	3	C	S	c	s
0	1	0	0	4	ERASE LINE		$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	LAMP CLEAR	&	6	F	V	f	v
0	1	1	1	7	BEL	ROLL DOWN	'	7	G	W	g	w
1	0	0	0	8	←	→	(	8	H	X	h	x
1	0	0	1	9		ERASE PAGE	)	9	I	Y	i	y
1	0	1	0	10	LF		*	:	J	Z	j	z
1	0	1	1	11	↓		+	;	K	Æ	k	æ
1	1	0	0	12	ROLL UP	↑	,	<	L	Ø	l	ø
1	1	0	1	13	CR	↖	−	=	M	Å	m	å
1	1	1	0	14	Under-line		.	>	N	^	n	|
1	1	1	1	15	Normal		/	?	O	■	o	//

2.3. CPU Board

The CPU-Board contains, in addition to the 8080 CPU, the following:

- Sockets for 8 k byte of erasable PROM in 1 k byte circuits.
- 2 k byte of Random Access memory (RAM).
- Interrupt control circuits with 7 interrupt request inputs and corresponding acknowledge outputs.
- Direct memory access control circuits with 4 DMA request inputs and corresponding GRANT outputs.
- Asynchroneous serial interface for 20 mA Current loop intended for hard copy printer and test purposes. Referred to as TTY interface.

Memory locations are:

- 8 k byte PROM from 0000 to 1FFF (HEX)
- 2 k byte RAM from 2000 to 27FF (HEX)

The memory space from 2800 to 2FFF (HEX) is reserved.

The following Input/Output device codes apply:

Dev Code	Input	Output
F4	TTY DATA	TTY DATA
F5	TTY STATUS	RESERVED

TTY STATUS byte information is:

bit		Meaning
0	TBMT	Transmitter Buffer Empty
1	DA	Receive Data Available
2	NOT IN USE	
3	PE	Receive Parity Error
4	OR	Over-Run
5	FE	Framing Error
6	NOT IN USE	
7	BUSY (Drucker bereit ≙ 1 - 6324/27)	

Ba d Rate for the TTY or PRINTER interface is selected by switches on the board.

The INTERRUPT SYSTEM consists of a priority encoder receiving 7 interrupt request lines. The lower number has the higher priority. The circuits generates an interrupt senced by the CPU at the end of current instruction. When the CPU acknowledges the interrupt, a RESTART instruction corresponding to the active request line with highest priority is fetched and the corresponding interrupt acknowledge line is activated. The interrupt enable Flip-flop is automatically switched off to give time to reset the request.

The interrupt system is enabled by the instruction EI, and disabled by the instruction DI.

The interrupt levels are allocated as follows:

Level	Device	Restart Address	
0	RESET (Overiding)	0	
1	DISPLAY REFRESH LOGIC	8	(50 times/sec)
2	SYNC. INTERFACE	10	(HEX)
3	MASTER DISPLAY	18	"
4	CLUSTER INTERFACE	20	"
5	SPARE (CLUSTER INTF. II)	28	"
6	FLOPPY DISK. CTR	30	"
7	CASSETTE CTR	38	"

The DIRECT MEMORY ACCESS control circuits provides control for four devices connected to the general bus to transfer data to and from the systems memory, including the RAM on the CPU board. Address pointers and word counters must be located on the devices. The DMA control monitors for DMA REQUEST lines (1-4) number one has the highest priority. DMA transfer can take place in a memory fetch cycle only, and GRANT is not given before the CPU has read it's instruction. Then the appropriate GRANT line becomes active, and remains so for one microsecond. An exeption is DMA channel 1, where GRANT will remain active as long as the request is active. In addition to the GRANT line to each device, a signal DMA R/W is activated, to carry the information the GRANT has been given.

The DMA-channels are allocated as follows:

DMA-channel	Device
1	Test
2	Floppy Disc Controller
3	Cassette Controller
4	(Spare)

Maximum transfer rate for channel 2,3 and 4 (sum of rates) should be less than 50 k bytes/sec. For channel 1, maximum transfer rate is limited by the memory circuits.

2.4. KEYBOARD

Fig. 2.4.1 shows a keyboard configured to the maximum number of keys. Each key marked with a number can generate an ASCII code and those with text will generate a functionsignal for use only in the keyboard or in the display unit. Keys marked with text and numbers have both possibilities, depending on the type of switch in that position.

When specifying a keyboard a table must be given containing key-numbers in the left column and codes for different key-board modes in the right column. See example below.

KEY NO.	CODES (HEX) UNSHIFT	SHIFT	CTRL	CTRL+SHIFT
3	1B	1B	-	-
52	61	41	01	01
76	2E	3A	-	-
77	2D	5F	-	1F

A "———" in a column means that no code will be generated. This is possible only in the CTRL and CTRL + SHIFT mode. Codes generated in those two modes can not be freely chosen. Codes generated in these modes will be identical to codes generated in non-CTRL mode, except for bits 6 and 7, which are both zero.

When planning a code layout for a keyboard remember that the codes are electrically programmable and easy to change, but the key-layout is mecanically fixed and therefore much more expensive to change. Key positions not used must be covered by a top-cover, thus every new key-layout requires a different top-cover. The number of configuration requiring different mechanical layouts should therefore be kept to a minimum.

TAST NUMMER FOR TASTENE PÅ
NYTT TASTATUR

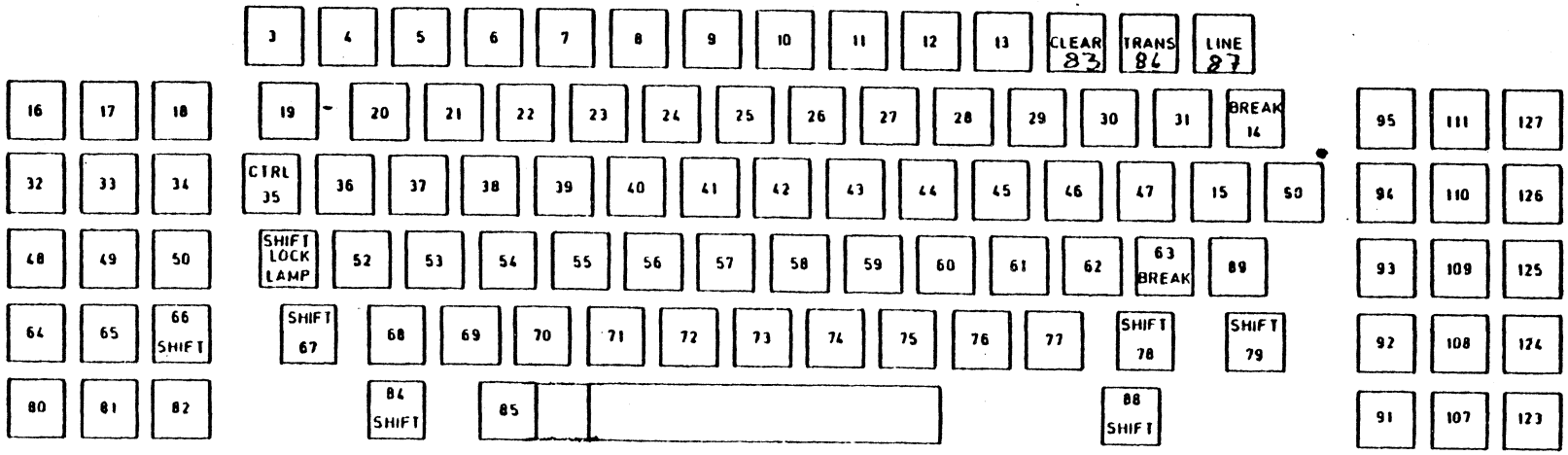
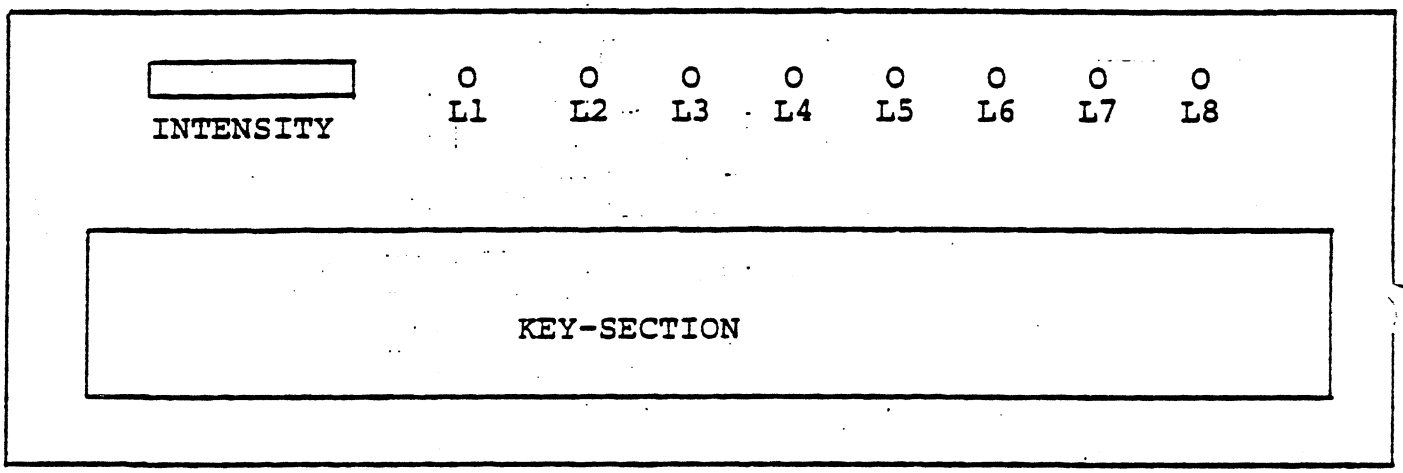


Fig. 2.4.1

In addition to the keys the keyboard comprises some status-indicators (LEDs) and an intensity pot.meter placed above the key section. The positions are indicated below.



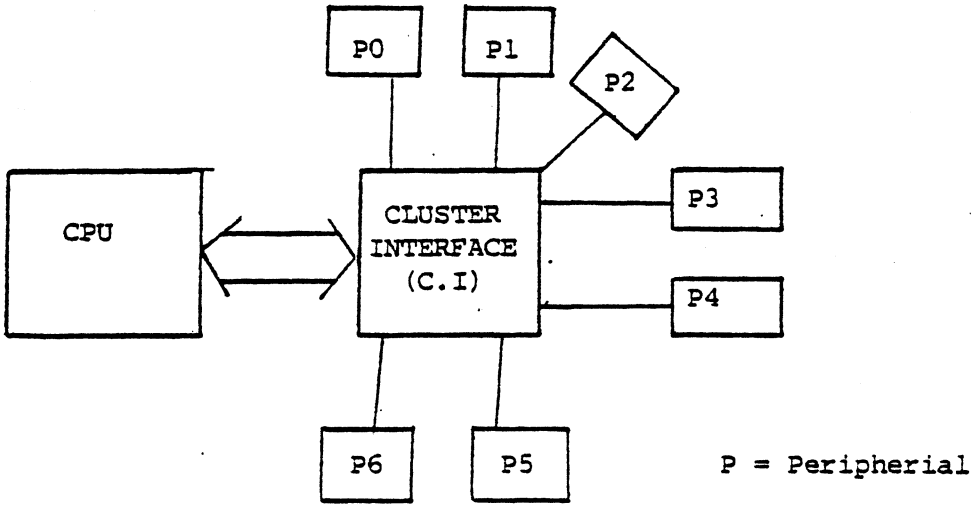
The LEDs will be identified with text written below each indicator.
Our TTY-replacement will be marked as follows:

L1	POWER ON
L2	ON LINE
L3	CARRIER
L4	ERROR
L5	ENQUIRY
L6	ACK
L7	NAK
L8	WAIT

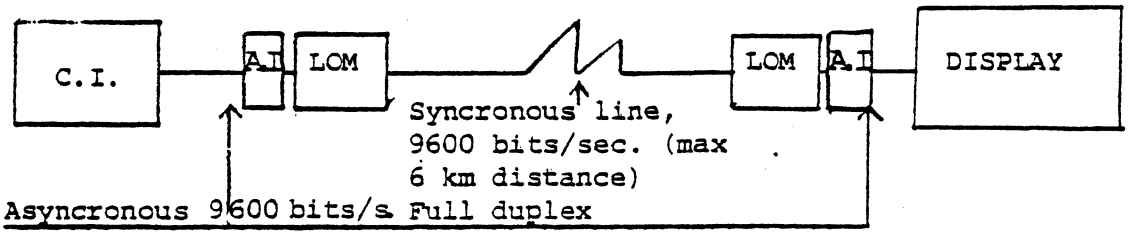
L5, L6 and L7 can be used and marked at the users discretion. L5, L6 and L7 will turn on when the display receives the codes 05, 06 and 15 (HEX) accordingly. All three lamps are turned off either with the CLEAR key or when the display receives the code 16 (strap selectable in the display).

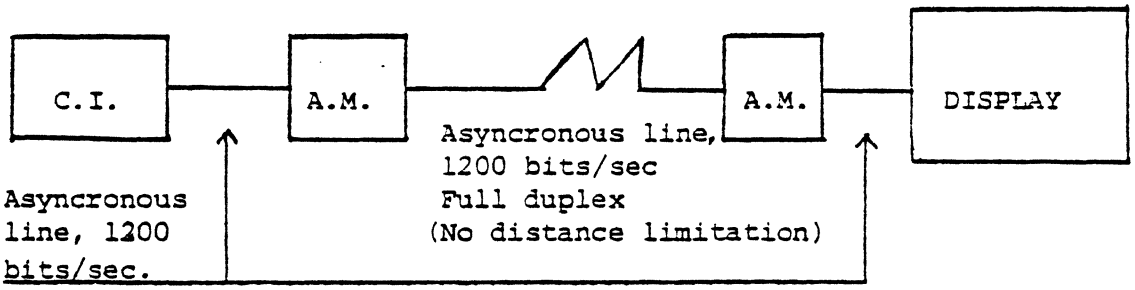
2.5. CLUSTER INTERFACE

The Cluster Interface is the link between the CPU, programmed I/O and up to seven slave displays or printers. The block diagram for a cluster system is shown below:



The communication between C.I. and each display or printer will be accomplished by a serial asyncronous full-duplex channel, normally based on a four wire 20 mA current loop. (Four of the channels may also use V.28 levels.) Transmission rate between C.I. and the peripherials will be adjusted to match the rate of the peripherial. We will normally use 19200 bits/sec. to and from a display provided the distance is less than 200 m (600 ft) and 9600 bits/sec. if the distance is between 200 m and 600 m. Beyond 600 m we will normally use a modem in each end of the line. Then we will have one of the two situations below:





(on V.28 levels, max 15 m)

LOM = Local Modem (syncronous)
A.M. = A syncronous Line Modem
A.I. = Asyncronous to Isocronous converter

Programming a Cluster Interface can be done two different ways, with or without use of the interrupt system. We shall first consider programming without interrupt. All exchange of data/status between CPU and C.I. is controlled by device codes (addresses). The table is shown below.

Device code	Input	Output
00 (80)	DATA FROM PERIPH. 0	DATA TO PERIPH. 0
01 (81)	" " P1	" " P1
02 (82)	" " P2	" " P2
03 (83)	" " P3	" " P3
04 (84)	" " P4	" " P4
05 (85)	" " P5	" " P5
06 (86)	" " P6	" " P6
07 (87)	NOT IN USE	NOT IN USE
08 (88)	STATUS FROM P0	ONLY IN INTERRUPT PROGRAMMING
09 (89)	" " P1	" " " "
0A (8A)	" " P2	" " " "
0B (8B)	" " P3	" " " "
0C (8C)	" " P4	" " " "
0D (8D)	" " P5	" " " "
0E (8E)	" " P6	" " " "
0F (8F)	ONLY IN INTER-RUPT PROG.	NOT IN USE

(••) CLUSTER INTERFACE II.

Input Status byte information is:

BIT	Meaning
0	Transmitter Ready for a new byte
1	Receiver Ready with a new byte
2	NOT IN USE
3	The received byte has a parity error
4	The received byte before this was not read
5	The received byte was without a stop bit (BREAK)
6	NOT IN USE
7	NOT IN USE

All status bits is active when the bits are a 1.
Bits NOT IN USE will always be 1.

Only 7 bits of the data byte will be significant, bits 0 through 6. Bit 6 being the most significant bit. Bit 7 will be made a 0 in an input instruction, and is not significant in on output.
When programming, one has to look for a ready status bit and then, if active, exchange a data byte.

When using the interrupt system for data transfers another approach must be used. Before sending a byte to a peripherial output interrupts should be enabled by giving the C.I. a control byte. When the C.I. is ready it will interrupt on level 6. The interrupt routine must first read an Interrupt status byte using the device code 0F.

The Interrupt status byte information is:

BIT	Meaning
0	If 1, we have an interrupt pending, if 0, we have not
1	NOT IN USE will be logical 1
2	" " " " " " "
3	" " " " " " "
4	A, The least significant bit of the interrupt code
5	B, The second " " " " " "
6	C, The second most " " " " " "
7	D, The most " " " " " "

Interrupt code.

DCBA	Meaning		
0000	Interrupt from	P0	INPUT
0001	" "	P1	
0010	" "	P2	
0011	" "	P3	
0100	" "	P4	
0101	" "	P5	
0110	" "	P6	
0111	NOT IN USE		
1000	Interrupt from	P0	OUTPUT
1001	" "	P1	
1010	" "	P2	
1011	" "	P3	
1100	" "	P4	
1101	" "	P5	
1110	" "	P6	
1111	NOT IN USE		

Interrupt for output must be enabled by giving a control byte (only bit 0 and bit 1 is significant) to C.I. The table below will describe how:

Control byte.

Device code	bit 0	Meaning when bit 1 = 0
08 (88)	1	Enabling output interrupt possibility for P0
	0	Disabling " " " " "
09 (89)	1	Enabling " " " " P1
	0	Disabling " " " " "
0A (8A)	1	Enabling " " " " P2
	0	Disabling " " " " "
0B (8B)	1	Enabling " " " " P3
	0	Disabling " " " " "
0C (8C)	1	Enabling " " " " P4
	0	Disabling " " " " "
0D (8D)	1	Enabling " " " " P5
	0	Disabling " " " " "
0E (8E)	1	Enabling " " " " P6
	0	Disabling " " " " "

When bit 1 = 1 all output interrupt possibilities (for all 7 peripherials) will be disabled.

Input interrupt will always be let through to the CPU and must be handled before next byte has been received, one keystroke on the keyboard later. When handling it the status byte should be read first, checking any possible error in the received databyte.

After sending an interrupt on level 4(5), this interrupt request must be reset immediately after acknowledge to allow the CPU to enable interrupt again, thus enabling interrupt with higher priority to stop the normal sequence. However, there will not be sent another interrupt request on level 6 until there has been a period where no device (P0 to P6) has requested service. Therefore it must be checked whether a new request for service has occured while servicing the first one. This is accomplish by reading the interrupt status word (device code 0F) before leaving interrupt level 4(5). Then bit 0 should be checked for a 1. 1 will indicate that still more service is requested, and bit 7,6,5 and 4 will contain the interrupt code. When bit 0 is a 0 no further service is needed, and the C.I. will give a new interrupt request on level 4(5) when a new situation accures.

2.6. Floppy Disk Controller

The floppy disk system consists of a simple, low-priced disk d
and a controller/interface board. The floppy disk is a flexible
disk rotating 6 revolutions/sec. The disk is divided into 77 trac
numbered from 00 to 76. The data (serial) transfer rate is 250
kilobits/sec. or 31,25 kilobytes/sec. The maximum data storage
capacity on one disk is approx. 246 kilobytes when standard IBM37
format is employed.

Format:

Total number of tracks	Number of data tracks	Reserved track	Number of sectors on each track	Number of sectors on index track 00 with 128 bytes	Number o bytes or each sec
77	74	2(75-76)	26(1-26)	26(1-26)	128
77	74	2(75-76)	15(1-15)	26(1-26)	256 except track 00
77	74	2(75-76)	8(1-8)	26(1-26)	512 except track 00

What number of bytes on each sector is initialized on the diskette

Table 2.6.1 gives a summary of the instruction and commands set for the floppy disk controller. Basically, there are 6 output instructions, 1 input instruction, and 7 output commands (includir 1 which are optional).

The output instructions are transferred prior to any commands. These instructions contain information about drive address, track address, sector address, record length etc; information which is necessary before any command can be executed.

The output commands are used to execute a write/read or seek opera tion. When a operation is completed, the controller sets the necessary bits in the status register and the interrupt line is s true. The status register can then be read by an input instruction

After a command instruction has been transferred the next output- or command instruction must not be given until the interrup line is set true by the controller.

Data transfer to and from the controller takes place via the general DMA-system.

Read after write is not implemented.

Table 2.6.1

TARTAR

Summary of input/output instructions for the diskette drive controller.

Interrupt level: 6

DMA priority : 2

Output instructions (CPU → Controller)

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus (DB0-7)
DRIVA	Drive Address	xx30 *	7 … 0 0 0 0 0 D4 D3 D2 D1 A 1 in one of D1-D4 selects one of four drives. Must be given before each command. Positive logic. Only one drive can be selected at a time.
RELTA	Relative Track Address	xx31	7 … 0 R T6 T5 T4 T3 T2 T1 T0 The binary number T0 → T6 is the difference between the current and the new track address (T6 most significant bit). R indicates the direction of the movement. To move the head inward (towards a track with higher number), R is set to 1. Positive logic. Must be given before SEEK
SECTA	Sector Address	xx32	7 … 0 0 0 S5 S4 S3 S2 S1 S0 S0 - S5 is the binary sector number, with S5 most significant bit. Positive logic. An all zero number is not valid. Must be given before each REDAT, WRTDA or WTDEL commend.

* x = Don't care

Mnemonic	Name	Address Bux (Hex.)	Data transfer Bus (DB0-7)
ADDHI	Address Register High	xx33	7 … 0 A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 This instruction is used to transfer the most significant byte of the memory start address to the controller. Must be given before each REDAT,WRTDA or WTDEL command. A15 is the most significant bit of the memory address. Positive logic
ADDLO	Address Register Low	xx34	7 … 0 A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 This instruction is used to transf the least significant byte of the memory address to the controller. Must be given before each REDAT, WRTDA or WTDEL command. A0 is the least significant bit of the memory address. Positive logic.
CLKBT	Clock Byte	xx85	7 … 0 C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 This instruction is used to transfer the clock bit pattern of the first byte to be written in a write format operation. Must be given before the WRTFO command.
DATBT	Data Byte	xx86	7 … 0 D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 This instruction is used to transfer the data bit pattern of the first byte to be written in a write format operation. Must be given before the WRTFO command.

I Output commands (CPU → controller)

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus (DB0-7)
WRTDA	Write Data field	xx87	7 0 1 1 1 1 0 0 0 1 Hex.: F1 Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, and SECTA instructions. Positive logic.
WTDEL	Write Deleted Data	xx87	7 0 1 1 1 1 0 0 1 0 Hex.: F2 With this command the controller writes a deleted ID mark for the data field. Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, and SECTA instructions. Positive logic.
WRTFO	Write format (optional)	xx87	7 0 1 1 1 1 1 0 1 1 Hex.: FB This command is employed to initialized an entire track. Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, CLKBT and DATBT instruction. In the execution of this command, the data and clock bit patterns for the whole track must be provided for by the main memory/CPU.

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus (DB0-7)
			The controller starts by writing the last 45 bytes of the pre-index gap. The clock bit pattern for the first of these bytes must be loaded into the S5 register (with a CLKBT operation) and the data bit pattern for the same byte must be loaded into the S6 register (with a DATBT operation) before the WRTFO command is given The clock bit pattern for the next byte must be in the first memory location (the location no. given by the ADDHI and ADDLO instruction and the data bit pattern for this byte must be in the next (+1) memory location. Then comes the next clock bit pattern and so on. Nominally, there is 5208 byte of clocks and 5208 bytes of data to be written into one track. The write operation stops 1 byte after the leading edge of the index strobe has been detected. Nominally, this is after the first 275 bytes of the pre-index gap. Due to speed tolerances, the main memory should contain at least 100 bytes of gap data and 100 bytes of gap clocks more, to compensate for worst case speed variation.

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus (DB0-7)
REDAT	Read Data field	xx37	7 0 1 1 1 0 0 0 0 0 Hex.: E0 Prior to the transfer of this command, it is necessary to carry out the ADDHI and ADDLO, and SECTA instructions. Positive logic.
REDID	Read ID	xx37	7 0 1 1 1 0 0 1 1 0 Hex.: E6 This command is used to read an ID-mark. 5 bytes are transferred to CPU: Byte Contents 1. ID Address Mark 2. Track address (binary) 3. Binary zero 4. Sector address (binary) 5. Record length (normally zero). Prior to the transfer of this command, it is necessary to carry out the ADDHI and ADDLO instruction Positive logic.
SEEKT	Seek Track	xx37	7 0 1 1 1 0 0 0 1 1 Hex.: E3 This command is used to move the head to a new track. Prior to this command, it is necessary to carry out the RELTA instructions. Positive logic.

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus (DB0-7)
RECAL	Recali-brate	xx37	7 … 0 1 1 1 0 1 1 0 0 Hex.: EC This command moves the head to track 00. Positive logic.

Input instruction (Controller → CPU)

Mnemonic	Name	Address Bus (Hex.)	Data transfer Bus
SENST	Sense Status	37	7 … 0

OPC	DR	NVC	NR	NA	CRE	SM	BSY

OPC = Operation completed
DR = Deleted Record ✱
NVC = Non-Valid command
NR = Drive not Ready
NA = No Address Mark
CRE = CRC Error
SM = Sector missing
BSY = Busy
NA/CRE = Timing error in read operat.
Comment: The drive is ready when the power is on, the floppy disk inserted and the drive is selected.

The Non-Valid command is set whenever the CPU transfer an irrecognizable instruction or the CPU gives a write command and the diskette is file protected.

The interrupt line is set true whenever one or more bits in the sense status register is set true, exce for the busy status. The SENST instruction can be given at any time. SENST must be read before every new output inst/commands because inst/com. must not be given when BSY is true.

✱ DR is set when a deleted address mark is detected. Data is transmitted as usual. After operation is finished both DR and OPC are true.

SM/CRE = Timing error during write data transfer operation.

2.7. Description of Synchronous Interface

The Synchronous Interface is used basically for communication between the terminal-controller or terminal-master and the host computer. This link should be synchronous to establish the most efficient way of communication. Also the synchronous communication equipment can communicate at a higher baud-rate.
If, however, the host computer does not have a synchronous port, the synchronous interface can be set up in an asynchronous mode. This is done partially by the program and partially by a hardware strap (the clock will then normally not be transmitted by the communication equipment, and must therefore be generated by the interface). The interface board also contains 32 8-bits input ports for use in a multi-terminal cluster system. These 32 ports will be programmed in such a way that an initial software program can supply all necessary information about the cluster configuration to the terminal-master.
Before going into detail about how to program the interface, the first matter we have to settle is how to select among the different ports on this board. The table below gives the specific device codes.

DEVICE CODE (HEX)	OUTPUT	INPUT
40	DATA TO HOST COMPUTER	DATA FROM HOST COMPUTER
41	COMMANDS FOR THE ABOVE TWO PORTS	STATUS FOR THE ABOVE TWO PORTS
46		STATUS FOR THE MODEM INTERFACE
47	INTERRUPT ENABLE/ DISABLE (DBO) FROM THE DATA OUTPUT PORT	THE INTERRUPT STATUS WORD
60→6F		CONFIG.PORT 00 - 15
70→7F		CONFIG.PORT 16 - 31

When initially starting up (after power on reset), the program will be interrupted on level 2 by the interface. This is to insure an initial reading of the status of the modem interface. After this the modem status port will interrupt whenever there has been a change since last modem status was read.

The vital part in the interface is an LSI-circuit, 8251, from Intel. This IC programs the transfer of data between the interface and the host computer. Therefore a detailed description of the operation of Intel 8251 is a part of this paper. The reader should be familiar with this circuit before going further.

In our system we will use internal SYNC DETECT, and the SYNDET-pin on 8251 is used to generate an interrupt when synchronization has been established. Further more both RxRDY and TxRDY is used to generate interrupt, TxRDY, however, only if the output interrupt enable flip-flop is set at the same time. This flip-flop is set by giving the device code 47 together with a "1" in DB0 out to the interface. A "0" in DB0 together with the device code 47 will reset the same flip-flop.

When getting an interrupt from the interface (level 2) the program should read the interrupt status word to see what caused the interrupt.

The contents of this status word (device code 47, input) is

DB0	RxRDY	All active ONE
DB1	Output Interrupt	
DB2	SYNCHRONIZATION DETECTED	
DB3	MODEM STATUS INTERRUPT	

DB4 - DB7 NOT IN USE (will be 1 • s)

If one interrupt occurs while another is pending but after the interrupt status word has been read, this will not cause another interrupt. Therefore the program must check for this before leaving the interrupt level. This is done by reading the interrupt status word one more time, checking that no interrupts are active.

As mentioned before, the modem status word will give a modem status interrupt if anything changes in the modem controlsignal-interface. Getting such an interrupt, the program should read the modem status word and examine the new situation. The modem status word (device code 46, input) consists of the following signals.

DB0	DATA TERMINAL READY, CT 108	All active one
DB1	DATA SET READY, CT 107	
DB2	REQUEST TO SEND, CT105	
DB3	CLEAR TO SEND, CT 106	
DB4	CARRIER INDICATION, CT 109	
DB5	CALLING INDICATOR, CT 125	

DB6, DB7 NOT IN USE, WILL BE "1".

As seen by the description of 8251 the signal DATA TERMINAL READY, DTR, also is a part of the status word from this device (device code 41).

One thing the user should be aware of is that the signal CALLING INDICATOR, CT125, is a temporary signal following the ringing signal in an ordinary telephone. After causing an interrupt, it will go passive again after approximately 1 second. Therefore the modem status word must be read before this, if CT125 is of vital interest.

The remaining 32 ports on this board are called configuration ports. The content is decided at installation time by programming a PROM-IC (IM 5610).

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize the 8251 to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD PARITY etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251 is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251. On the other hand, the 8251 receives serial data from the MODEM or I/O device, upon receiving an entire character the RxRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TxEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TxD output will be held in the marking state upon Reset.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

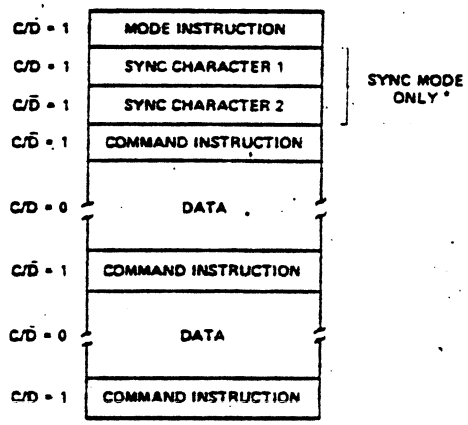
This format defines the general operational characteristics of the 8251. It must follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, SYNC characters or Command instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



*The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

Typical Data Block

Mode Instruction Definition

The 8251 can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251 the designer can best view the device as two separate components sharing the same package. One Asynchronous the other Synchronous. The format definition can be changed "on the fly" but for explanation purposes the two formats will be isolated.

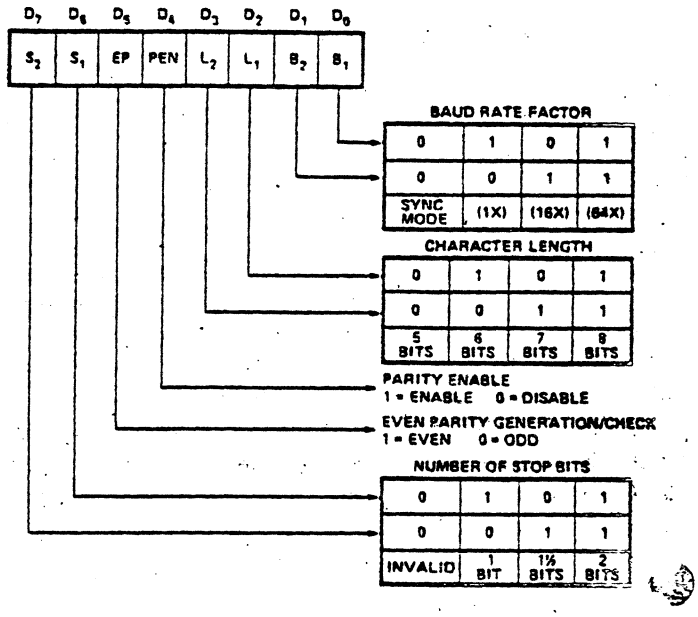
Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

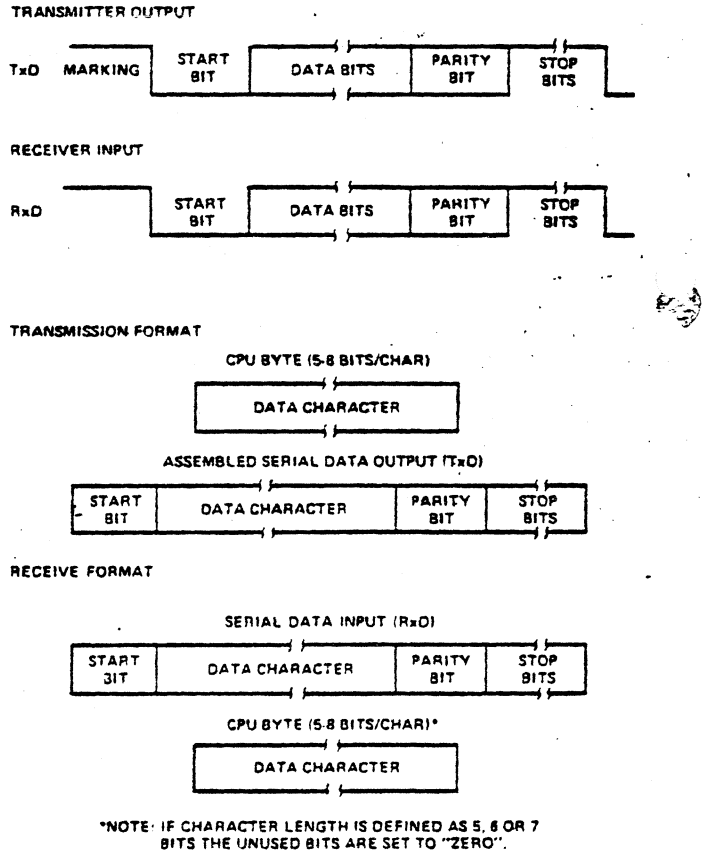
When no data characters have loaded into the 8251 the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.



Mode Instruction Format, Asynchronous Mode



*NOTE: IF CHARACTER LENGTH IS DEFINED AS 5, 6 OR 7 BITS THE UNUSED BITS ARE SET TO "ZERO".

Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251 is empty and SYNC characters are being sent out. The TxEMPTY pin is internally reset by the next character being written into the 8251.

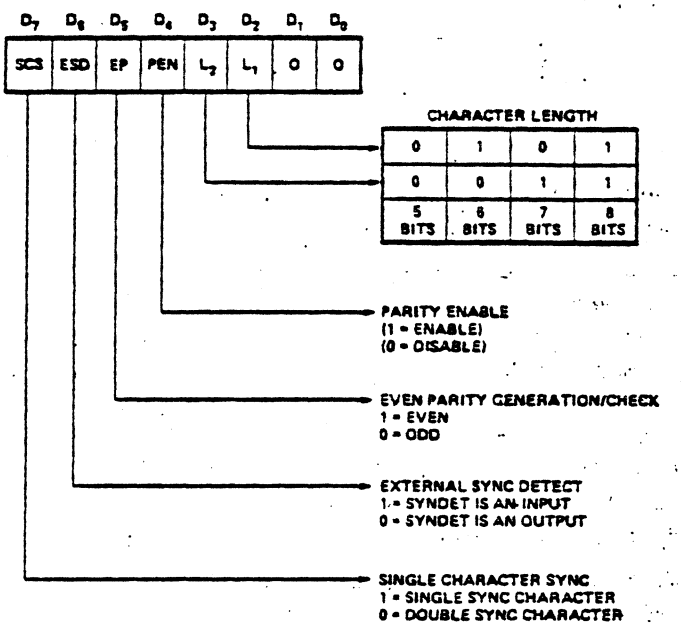
Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RxD pin is then sampled in on the rising edge of $\overline{RxC}$. The content of the Rx buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

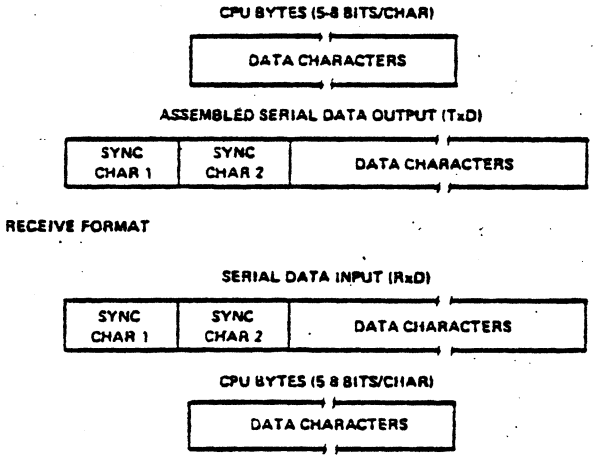
In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin. The high level can be removed after one $\overline{RxC}$ cycle.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.



Mode Instruction Format, Synchronous Mode

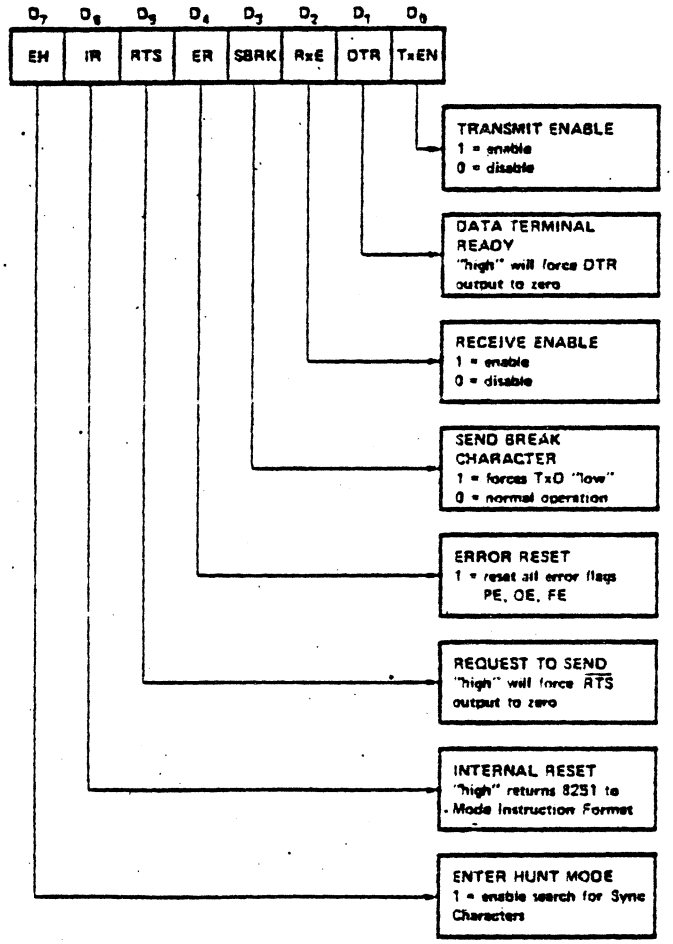


Synchronous Mode, Transmission Format

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" (C/$\overline{D}$ = 1) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.



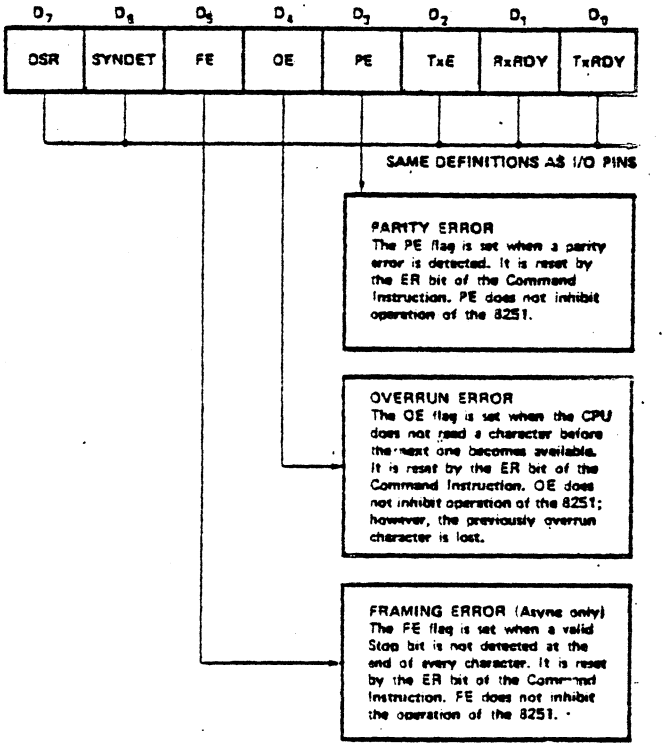
Command Instruction Format

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.

A normal "read" command is issued by the CPU with the C/D input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment.



Status Read Format