UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang 1989/90

Mac/April 1990

ZAE 416/4 Pengantar Mikropemproses/Mikrokomputer

Masa : [3 jam]

Sila pastikan bahawa kertas peperiksaan ini mengandungi TIGA muka surat yang bercetak sebelum anda memulakan peperiksaan itu.
Setiap orang calon juga diberikan senaskah "Set Arahan untuk 8085" dan sepuluh keping "Formatted Answer Sheet".
Jawab KESEMUA EMPAT soalan.
Kesemuanya wajib dijawab di dalam Bahasa Malaysia.
Bagi Soalan 2, 3 dan 4 carta aliran harus diberi bagi tiap-tiap aturcara.

1. (a) Terangkan apa langkah yang dilaksanakan oleh MPU apabila ia disampukkan.

(20/100)

(b) Lakarkan senibina mikropemproses 8085 dan perihalkan fungsi bagi:-

(i) daftar-daftar sedia ada
(ii) ALU
(iii) penyah-kod dan daftar suruhan di dalam 8085 itu.

(40/100)

(c) Terangkan dengan ringkasnya kaedah pemindahan data I/O bagi:-

(i) pemindahan penyegerakan (synchronous transfer).
(ii) pemindahan tak segerak (asynchronous transfer).

Bincangkan kelemahan dan kebaikan kedua-dua kaedah I/O tersebut.

(40/100)

2. Tuliskan aturcara-aturcara untuk paparan/display nombor di LED tujuh-ruas (seven-segment LED) sedia ada di atas suatu komputer SDK-85.

...2/-

(a) Tuliskan satu aturcara untuk paparan nombor 7 di kanan medan data LED (paparan 5/display 5).

(20/100)

(b) Tambahkan satu gelung lengah (delay loop) untuk paparan selama 30 ms bagi nombor 7 tersebut di bahagian (a).

(20/100)

(c) Mengubahsuaikan aturcara paparan nombor 7 itu untuk menggerakkan paparan nombor tersebut dari kanan ke kiri, iaitu memaparkan nombor mula-mulanya di tempat paparan 5 dan selepas memindahkan paparan di tempat paparan 4 dan kemudian ke paparan 3, ... paparan 0 berturut-turut.

(60/100)

3. Tuliskan aturcara-aturcara untuk paparan LED bertujuh-ruas di atas SDK-85 hanya melalui penggunaan subroutin-subroutin monitor SDK-85 yang sedia ada.

(a) Tuliskan satu aturcara untuk paparan data di medan data LED hanya dengan penggunaan subroutin monitor tersebut. Data paparan terletak di alamat 2040H. Semasa paparan itu, kod syarat atau status word patut ditindankan (stacked). Paparan itu akan mengambil masa beberapa saat sahaja.

(30/100)

(b) Tuliskan satu aturcara untuk paparan alamat 2040H di medan alamat LED dengan penggunaan subroutin monitor.

(30/100)

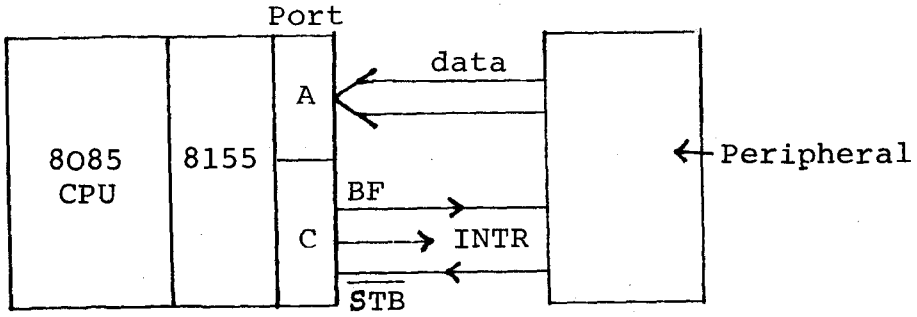
(c) Tuliskan satu aturcara lengkap untuk paparan kedua-dua alamat data dan data sama sekali dari alamat 2040H ke 2045H di medan alamat dan data masing-masing dengan paparan selama beberapa saat bagi setiap paparan.

(40/100)

4. Pertimbangkan liang/port A, B dan C bagi RAM 8155 digunakan untuk komunikasi/antaramuka dengan persisian peranti (peripheral device). Cara antara muka merupakan I/O teraturcara.

...3/-

(a) Peruntukkan liang A sebagai input data daripada suatu persisian dan liang C sebagai liang kawalan seperti berikut.



Tuliskan suatu aturcara dengan jabat-tangan yang sewajarnya untuk input data daripada liang A dan menentukan sama ada data input bit 2 sama dengan sifar atau tidak. Jika input bit 2 bukan sifar, liang A akan terus-menerus menerima data input baru sehingga bit 2 tersebut menjadi sifar.

(50/100)

(b) Peruntukkan liang B sebagai output data kepada suatu persisian dan liang C juga sebagai liang kawalan.
Tuliskan suatu aturcara dengan jabat-tanga yang sewajarnya. Data output tersebut terletak daripada 2040H hingga 204AH.

(50/100)

- oooOooo -

Soalan _____ Bahagian _____

ADRESS (HEX)	CONTENTS (HEX)	LABEL	MNEMONIC	OPERAND	COMMENTS

YBC/cm
14.10.1988

LAMPIRAN

ZAE 416

PUSAT PENGAJIAN SAINS FIZIK
UNIVERSITI SAINS MALAYSIA
11800 PULAU PINANG

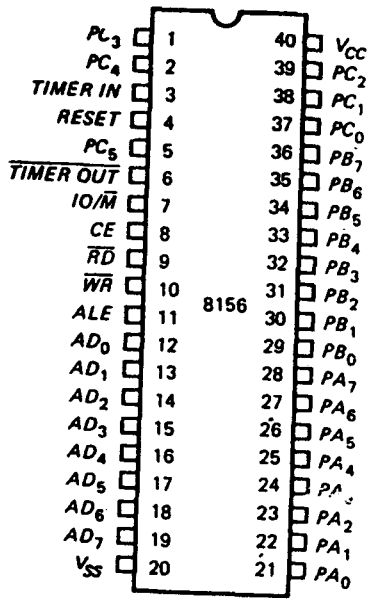
Instruction	Op Code	*T* states	Flags	Main Effect
ACI byte	CE	7	All	A ← A + CY + byte
ADC A	8F	4	All	A ← A + A + CY
ADC B	88	4	All	A ← A + B + CY
ADC C	89	4	All	A ← A + C + CY
ADC D	8A	4	All	A ← A + D + CY
ADC E	8B	4	All	A ← A + E + CY
ADC H	8C	4	All	A ← A + H + CY
ADC L	8D	4	All	A ← A + L + CY
ADC M	8E	7	All	A ← A + M_{HL} + CY
ADD A	87	4	All	A ← A + A
ADD B	80	4	All	A ← A + B
ADD C	81	4	All	A ← A + C
ADD D	82	4	All	A ← A + D
ADD E	83	4	All	A ← A + E
ADD H	84	4	All	A ← A + H
ADD L	85	4	All	A ← A + L
ADD M	86	7	All	A ← A + M_{HL}
ADI byte	C6	7	All	A ← A + byte
ANA A	A7	4	All	A ← A AND A
ANA B	A0	4	All	A ← A AND B
ANA C	A1	4	All	A ← A AND C
ANA D	A2	4	All	A ← A AND D
ANA E	A3	4	All	A ← A AND E
ANA H	A4	4	All	A ← A AND H
ANA L	A5	4	All	A ← A AND L
ANA M	A6	7	All	A ← A AND M_{HL}
ANI byte	E6	7	All	A ← A AND byte
CALL address	CD	18	None	PC ← address
CC address	DC	18/9	None	PC ← address if *CY* = 1
CM address	FC	18/9	None	PC ← address if *S* = 1
CMA	2F	4	None	A ← $\overline{A}$
CMC	3F	4	CY	CY ← $\overline{CY}$
CMP A	BF	4	All	Z ← 1 if A = A
CMP B	B8	4	All	Z ← 1 if A = B
CMP C	B9	4	All	Z ← 1 if A = C
CMP D	BA	4	All	Z ← 1 if A = D
CMP E	BB	4	All	Z ← 1 if A = E
CMP H	BC	4	All	Z ← 1 if A = H
CMP L	BD	4	All	Z ← 1 if A = L
CMP M	BE	7	All	Z ← 1 if A = M_{HL}
CNC address	D4	18/9	None	PC ← address if *CY* = 0
CNZ address	C4	18/9	None	PC ← address if *Z* = 0
CP address	F4	18/9	None	PC ← address if *S* = 0
CPE address	EC	18/9	None	PC ← address if *P* = 1
CPI byte	FE	7	All	Z ← 1 if *A* = byte
CPO address	E4	18/9	None	PC ← address if *P* = 0
CZ address	CC	18/9	None	PC ← address if *Z* = 1
DAA	27	4	All	A ← BCD number
DAD B	09	10	CY	HL ← HL + BC
DAD D	19	10	CY	HL ← HL + DE
DAD H	29	10	CY	HL ← HL + HL

Instruction	Op Code	t states	Flags	Main Effect
DAD SP	39	10	CY	HL ← HL + SP
DCR A	3D	4	All but CY	A ← A − 1
DCR B	05	4	All but CY	B ← B − 1
DCR C	0D	4	All but CY	C ← C − 1
DCR D	15	4	All but CY	D ← D − 1
DCR E	1D	4	All but CY	E ← E − 1
DCR H	25	4	All but CY	H ← H − 1
DCR L	2D	4	All but CY	L ← L − 1
DCR M	35	10	All but CY	$M_{HL} \leftarrow M_{HL} - 1$
DCX B	0B	6	None	BC ← BC − 1
DCX D	1B	6	None	DE ← DE − 1
DCX H	2B	6	None	HL ← HL − 1
DCX SP	3B	6	None	SP ← SP − 1
DI	F3	4	None	Disable interrupts
EI	FB	4	None	Enable interrupts
HLT	76	5	None	Stop processing
IN byte	DB	10	None	A ← byte
INR A	3C	4	All but CY	A ← A + 1
INR B	04	4	All but CY	B ← B + 1
INR C	0C	4	All but CY	C ← C + 1
INR D	14	4	All but CY	D ← D + 1
INR E	1C	4	All but CY	E ← E + 1
INR H	24	4	All but CY	H ← H + 1
INR L	2C	4	All but CY	L ← L + 1
INR M	34	10	All but CY	$M_{HL} \leftarrow M_{HL} + 1$
INX B	03	6	None	BC ← BC + 1
INX D	13	6	None	DE ← DE + 1
INX H	23	6	None	HL ← HL + 1
INX SP	33	6	None	SP ← SP + 1
JC address	DA	10/7	None	PC ← address if $CY = 1$
JM address	FA	10/7	None	PC ← address if $S = 1$
JMP address	C3	10	None	PC ← address
JNC address	D2	10/7	None	PC ← address if $CY = 0$
JNZ address	C2	10/7	None	PC ← address if $Z = 0$
JP address	F2	10/7	None	PC ← address if $S = 0$
JPE address	EA	10/7	None	PC ← address if $P = 1$
JPO address	E2	10/7	None	PC ← address if $P = 0$
JZ address	CA	10/7	None	PC ← address if $Z = 1$
LDA address	3A	13	None	$A \leftarrow M_{adr}$
LDAX B	0A	7	None	$A \leftarrow M_{BC}$
LDAX D	1A	7	None	$A \leftarrow M_{DE}$
LHLD address	2A	16	None	$H \leftarrow M_{adr}$
LXI B, dble	01	10	None	BC ← dble
LXI D, dble	11	10	None	DE ← dble
LXI H, dble	21	10	None	HL ← dble
LXI SP, dble	31	10	None	SP ← dble
MOV A,A	7F	4	None	A ← A
MOV A,B	78	4	None	A ← B
MOV A,C	79	4	None	A ← C
MOV A,D	7A	4	None	A ← D
MOV A,E	7B	4	None	A ← E
MOV A,H	7C	4	None	A ← H
MOV A,L	7D	4	None	A ← L

Instruction	Op Code	*T* states	Flags	Main Effect
MOV A,M	7E	7	None	A ← M_{HL}
MOV B,A	47	4	None	B ← A
MOV B,B	40	4	None	B ← B
MOV B,C	41	4	None	B ← C
MOV B,D	42	4	None	B ← D
MOV B,E	43	4	None	B ← E
MOV B,H	44	4	None	B ← H
MOV B,L	45	4	None	B ← L
MOV B,M	46	7	None	B ← M_{HL}
MOV C,A	4F	4	None	C ← A
MOV C,B	48	4	None	C ← B
MOV C,C	49	4	None	C ← C
MOV C,D	4A	4	None	C ← D
MOV C,E	4B	4	None	C ← E
MOV C,H	4C	4	None	C ← H
MOV C,L	4D	4	None	C ← L
MOV C,M	4E	7	None	C ← M_{HL}
MOV D,A	57	4	None	D ← A
MOV D,B	50	4	None	D ← B
MOV D,C	51	4	None	D ← C
MOV D,D	52	4	None	D ← D
MOV D,E	53	4	None	D ← E
MOV D,H	54	4	None	D ← H
MOV D,L	55	4	None	D ← L
MOV D,M	56	7	None	D ← M_{HL}
MOV E,A	5F	4	None	E ← A
MOV E,B	58	4	None	E ← B
MOV E,C	59	4	None	E ← C
MOV E,D	5A	4	None	E ← D
MOV E,E	5B	4	None	E ← E
MOV E,H	5C	4	None	E ← H
MOV E,L	5D	4	None	E ← L
MOV E,M	5E	7	None	E ← M_{HL}
MOV H,A	67	4	None	H ← A
MOV H,B	60	4	None	H ← B
MOV H,C	61	4	None	H ← C
MOV H,D	62	4	None	H ← D
MOV H,E	63	4	None	H ← E
MOV H,H	64	4	None	H ← H
MOV H,L	65	4	None	H ← L
MOV H,M	66	7	None	H ← M_{HL}
MOV L,A	6F	4	None	L ← A
MOV L,B	68	4	None	L ← B
MOV L,C	69	4	None	L ← C
MOV L,D	6A	4	None	L ← D
MOV L,E	6B	4	None	L ← E
MOV L,H	6C	4	None	L ← H
MOV L,L	6D	4	None	L ← L
MOV L,M	6E	7	None	L ← M_{HL}
MOV M,A	77	7	None	M_{HL} ← A
MOV M,B	70	7	None	M_{HL} ← B

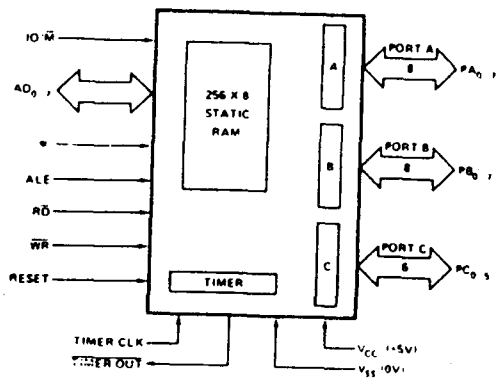
Instruction	Op Code	*T* states	Flags	Main Effect
MOV M,C	71	7	None	$M_{HL} \leftarrow C$
MOV M,D	72	7	None	$M_{HL} \leftarrow D$
MOV M,E	73	7	None	$M_{HL} \leftarrow E$
MOV M,H	74	7	None	$M_{HL} \leftarrow H$
MOV M,L	75	7	None	$M_{HL} \leftarrow L$
MVI A,byte	3E	7	None	A ← byte
MVI B,byte	06	7	None	B ← byte
MVI C,byte	0E	7	None	C ← byte
MVI D,byte	16	7	None	D ← byte
MVI E,byte	1E	7	None	E ← byte
MVI H,byte	26	7	None	H ← byte
MVI L,byte	2E	7	None	L ← byte
MVI M,byte	36	10	None	$M_{HL} \leftarrow$ byte
NOP	00	4	None	Delay
ORA A	B7	4	All	A ← A OR A
ORA B	B0	4	All	A ← A OR B
ORA C	B1	4	All	A ← A OR C
ORA D	B2	4	All	A ← A OR D
ORA E	B3	4	All	A ← A OR E
ORA H	B4	4	All	A ← A OR H
ORA L	B5	4	All	A ← A OR L
ORA M	B6	7	All	A ← A OR M_{HL}
ORI byte	F6	7	All	A ← A OR byte
OUT byte	D3	10	None	Port byte ← A
PCHL	E9	6	None	PC ← HL
POP B	C1	10	None	B ← M_{stk}
POP D	D1	10	None	D ← M_{stk}
POP H	E1	10	None	H ← M_{stk}
POP PSW	F1	10	None	F ← M_{stk}, A ← $M_{stk} - 1$
PUSH B	C5	12	None	$M_{stk} - 1 \leftarrow B$, $M_{stk} - 2 \leftarrow C$
PUSH D	D5	12	None	$M_{stk} - 1 \leftarrow D$, $M_{stk} - 2 \leftarrow E$
PUSH H	E5	12	None	$M_{stk} - 1 \leftarrow H$, $M_{stk} - 2 \leftarrow L$
PUSH PSW	F5	12	None	$M_{stk} - 1 \leftarrow A$, $M_{stk} - 2 \leftarrow F$
RAL	17	4	CY	Rotate all left
RAR	1F	4	CY	Rotate all right
RC	D8	12/6	None	PC ← return address if $CY = 1$
RET	C9	10	None	PC ← return address
RIM	20	4	None	A ← I
RLC	07	4	CY	Rotate left with carry
RM	F8	12/6	None	PC ← return address if $S = 1$
RNC	D0	12/6	None	PC ← return address if $CY = 0$
RNZ	C0	12/6	None	PC ← return address if $Z = 0$
RP	F0	12/6	None	PC ← return address if $S = 0$
RPE	E8	12/6	None	PC ← return address if $P = 1$
RPO	E0	12/6	None	PC ← return address if $P = 0$
RRC	0F	4	CY	Rotate right with carry
RST 0	C7	12	None	PC ← 0000H
RST 1	CF	12	None	PC ← 0008H
RST 2	D7	12	None	PC ← 0010H
RST 3	DF	12	None	PC ← 0018H
RST 4	E7	12	None	PC ← 0020H
RST 5	EF	12	None	PC ← 0028H
RST 6	F7	12	None	PC ← 0030H

Instruction	Op Code	*T* states	Flags	Main Effect
RST 7	FF	12	None	PC ← 0038H
RZ	C8	12/6	None	PC ← return address if Z = 1
SBB A	9F	4	All	A ← A − A − CY
SBB B	98	4	All	A ← A − B − CY
SBB C	99	4	All	A ← A − C − CY
SBB D	9A	4	All	A ← A − D − CY
SBB E	9B	4	All	A ← A − E − CY
SBB H	9C	4	All	A ← A − H − CY
SBB L	9D	4	All	A ← A − L − CY
SBB M	9E	7	All	A ← A − M − CY
SBI byte	DE	7	All	A ← A − byte − CY
SHLD address	22	16	None	M_{adr+1} ← H, M_{adr} ← L
SIM	30	4	None	I ← A
SPHL	F9	6	None	SP ← HL
STA address	32	13	None	M_{adr} ← A
STAX B	02	7	None	M_{BC} ← A
STAX D	12	7	None	M_{DE} ← A
STC	37	4	CY	CY ← 1
SUB A	97	4	All	A ← A − A
SUB B	90	4	All	A ← A − B
SUB C	91	4	All	A ← A − C
SUB D	92	4	All	A ← A − D
SUB E	93	4	All	A ← A − E
SUB H	94	4	All	A ← A − H
SUB L	95	4	All	A ← A − L
SUB M	96	7	All	A ← A − M
SUI byte	D6	7	All	A ← A − byte
XCHG	EB	4	None	HL ↔ DE
XRA A	AF	4	All	A ← A XOR A
XRA B	A8	4	All	A ← A XOR B
XRA C	A9	4	All	A ← A XOR C
XRA D	AA	4	All	A ← A XOR D
XRA E	AB	4	All	A ← A XOR E
XRA H	AC	4	All	A ← A XOR H
XRA L	AD	4	All	A ← A XOR L
XRA M	AE	7	All	A ← A XOR M
XRI byte	EE	7	All	A ← A XOR byte
XTHL	E3	16	None	HL ↔ stack



8156 pinout.

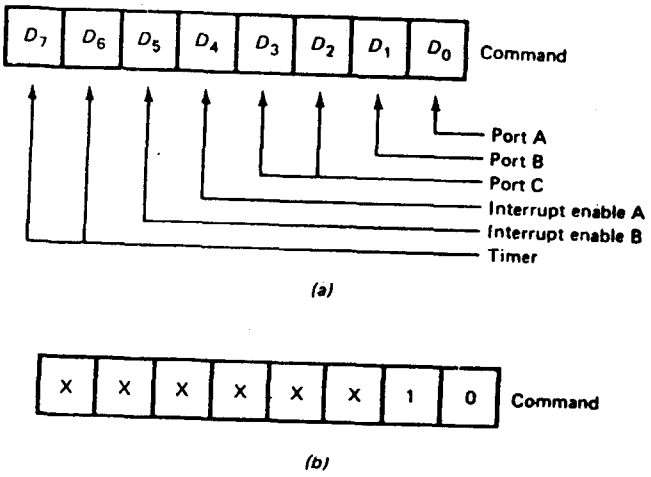
BLOCK DIAGRAM



* : 8155 = $\overline{CE}$, 8156 = CE

TABLE 15-2. 8156 PORT NUMBERS IN MINIMUM SYSTEM

Port Number	Selected Register
20H	Command-status register
21H	Port A
22H	Port B
23H	Port C
24H	Lower byte of timer count
25H	Timer mode and count (upper 6 bits)



(*a*) Command word; (*b*) B is output, A is input; (*c*) C and B are output, A is input.

TABLE 15-3. PORT A

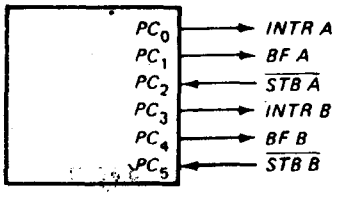
D_0	Effect
0	Input
1	Output

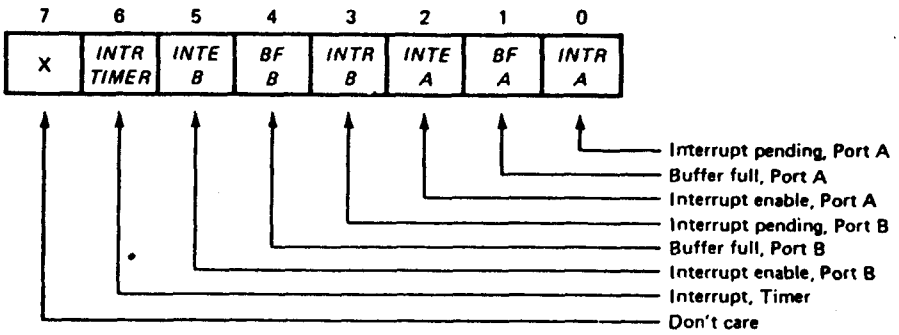
TABLE 15-4. PORT B

D_1	Effect
0	Input
1	Output

TABLE 15-5. PORT C

D_3	D_2	Mode
0	0	Input port (6 bits)
0	1	Port A handshaking and output (3 bits)
1	0	Ports A and B handshaking
1	1	Output port (6 bits)

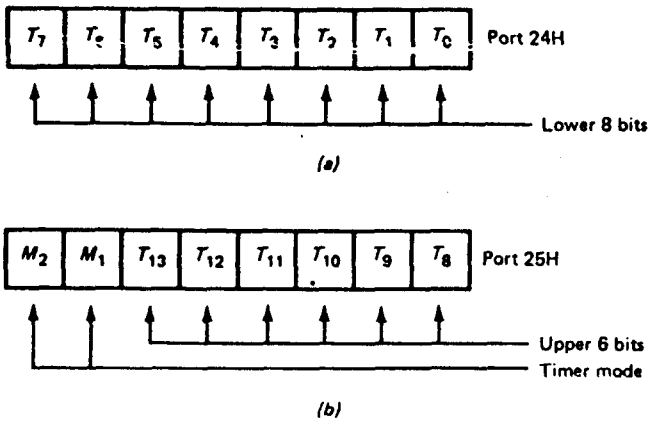




Status register.

TABLE 15-7. TIMER COMMAND

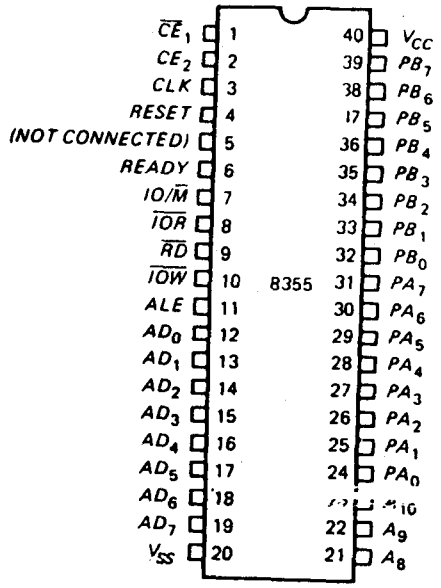
D_7	D_6	Effect
0	0	Nop
0	1	Stop immediately
1	0	Stop after TC is reached
1	1	Start



(*a*) Lower timer byte; (*b*) upper timer byte.

TABLE 15-6. TIMER MODE

M_2	M_1	Effect
0	0	Single square wave
0	1	Continuous square wave
1	0	Single pulse
1	1	Continuous pulse



8355 pinout.

TABLE 15-8. 8355 REGISTERS

AD_1	AD_0	Select
0	0	Port A
0	1	Port B
1	0	DDR A
1	1	DDR B

TABLE 15-9. 8355 PORT NUMBERS IN MINIMUM SYSTEM

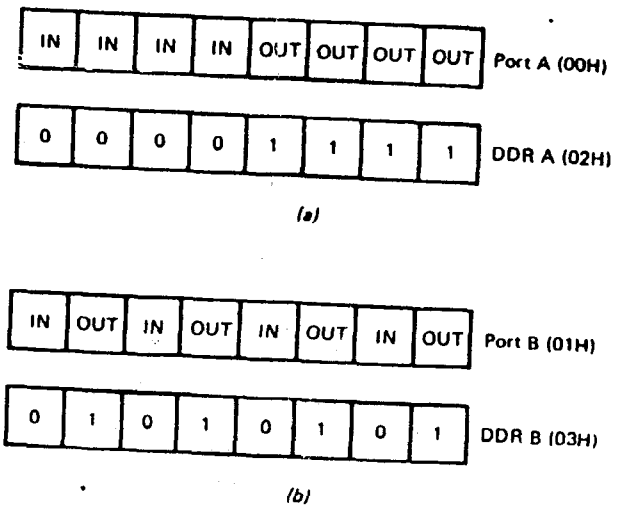
Port Number	Selected Register
00H	Port A
01H	Port B
02H	DDR A
03H	DDR B

TABLE 15-10. PORT A

DDR A bit	Port A pin
0	Input
1	Output

TABLE 15-11. PORT B

DDR B bit	Port B pin
0	Input
1	Output



Pin programming of ports A and B.

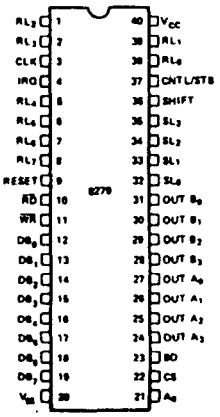
8279, 8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8 Character Keyboard FIFO
- 2 Key Lockout or N Key Rollover with Contact Debounce
- Dual 8 or 16 Numerical Display
- Single 16 Character Display
- Right or Left Entry 16 Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64 contact key matrix which can be expanded to 128. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the Hall effect and Ferrite variety. Key depressions can be 2 key lockout or N key rollover. Keyboard entries are debounced and strobed in an 8 character FIFO. If more than 8 characters are entered, over run status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has a 16 x 8 display RAM which can be organized into a dual 16 x 4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

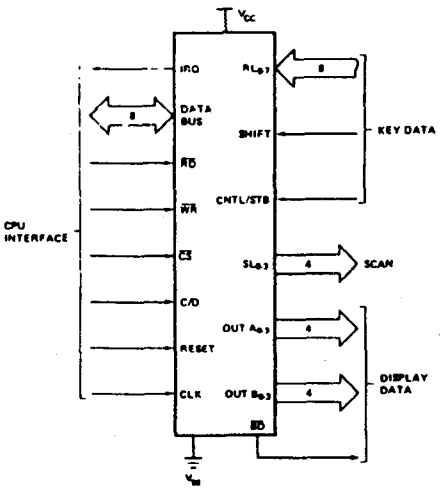
PIN CONFIGURATION



PIN NAMES

NAME	I/O	FUNCTION
DB_{0-7}	I/O	DATA BUS (BI-DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
$\overline{CS}$	I	CHIP SELECT
$\overline{RD}$	I	READ INPUT
$\overline{WR}$	I	WRITE INPUT
A_0	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL_{0-3}	O	SCAN LINES
RL_{0-7}	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
$OUT A_{0-3}$	O	DISPLAY (A) OUTPUTS
$OUT B_{0-3}$	O	DISPLAY (B) OUTPUTS
$\overline{BD}$	O	BLANK DISPLAY OUTPUT

LOGIC SYMBOL



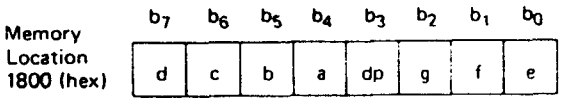


FIGURE 5-3. Segment connections for the SDK-85 displays.

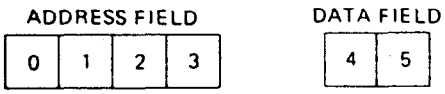


FIGURE 5-4. Numbering of the SDK-85 displays.

Table 5-1

NUMBERING OF THE SDK-85 DISPLAYS

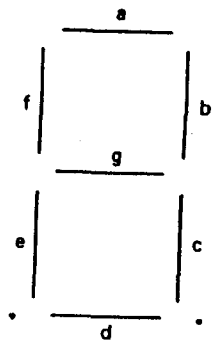
DISPLAY	NUMBER
Address digit 1	0
Address digit 2	1
Address digit 3	2
Address digit 4	3
Data digit 1	4
Data digit 2	5

Table 5-5

HEXADECIMAL-TO-SEVEN SEGMENT CONVERSION TABLE FOR SDK-85 DISPLAYS

HEXADECIMAL DIGIT	SEVEN-SEGMENT CODE (HEX)
0	0C
1	9F
2	4A
3	0B
4	99
5	29
6	28
7	8F
8	08
9	89
A	88
b (lower case)	38
C	6C
d (lower case)	1A
E	68
F	E8

324



A seven-segment display.

The seven-segment displays on the SDK-85 board are connected to the CPU through a complex LSI interface chip, the 8279 Keyboard/ Display Interface. The *MCS-85 User's Manual* describes this chip. We will only note that the following instructions are necessary to send data to one of the displays. The 8279 device is handled as if it were part of the memory rather than part of the input/output section.

```
MVI     A,NUM + 80H     ;ACTIVATE DISPLAY "NUM"
STA     1900H
MVI     A,DATA          ;SEND IT DATA
STA     1800H
```

Table A-1

SDK-85 MONITOR SUBROUTINES

NAME	CALLING ADDRESS (HEX)	PURPOSE
CI (CONSOLE INPUT)	07FD or 0590	Waits for an ASCII character to be received from the teletype and returns its value in the A register.
CLEAR (CLEAR DISPLAY)	01D7	Blanks the entire display. Puts a dot at the right edge of the address field if (B) = 1.
CNVBN (CONVERT TO BINARY)	05BB	Converts an ASCII hex digit in C to a binary value in A.
CO (CONSOLE OUTPUT)	07FA or 05C4	Transmits an ASCII character in register C to the teletype.
CROUT (CARRIAGE RETURN, LINE FEED)	05EB	Sends carriage return and line feed characters to the teletype.
DELAY (TIME DELAY)	05F1	Provides a time delay by counting register pair DE down to zero.
DISPC (DISPLAY PROGRAM COUNTER)	0200	Displays the user program counter and the contents of that address on the display.
ECHO (ECHO TO TERMINAL)	05F8	Sends the character in A to the teletype or terminal. Echoes a carriage return as a carriage return followed by a line feed and an ESCAPE character (1B hex) as a $ (24 hex).
HILO (COMPARE 16-BIT INTEGERS)	06A0	Compares the 16-bit integers in HL and DE. The CARRY is set if the contents of HL are larger than or equal to the contents of DE; the CARRY is cleared if the contents of HL are less than the contents of DE.
NMOUT (HEX NUMBER PRINTER)	06C7	Prints the contents of A as two hex digits on the teletype.
OUTPT (OUTPUT CHARACTERS TO DISPLAY)	02B7	Sends characters to display. Reg A = 0 = use address field = 1 = use data field Reg B = 0 = decimal point off = 1 = decimal point at right edge of field Reg HL = starting address of characters to be sent
PRVAL (CONVERT HEX TO ASCII)	06E2	Converts a hexadecimal digit in register C to an ASCII character in register C.
UPDAD (UPDATE ADDRESS DISPLAY)	0362	Displays the current value of registers H and L as four hex digits in the address field of the display. Puts a dot at the right edge if (B) = 1.
UPDDT (UPDATE DATA)	036E	Displays the contents of A as two hex digits in the data field of the display.
VALDG (CHECK FOR VALID HEX DIGIT)	075E	Checks if A contains a valid hex digit. CARRY is set if A does, cleared if it does not.

Table A-2

SDK-85 DISPLAY OUTPUT TABLE

INPUT	CHARACTER DISPLAYED
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	A
B	b
C	C
D	d
E	E
F	F
10	H
11	L
12	P
13	l
14	r
15	Blank

Table C-1

RESTART INSTRUCTIONS AND INPUTS

INSTRUCTION OR INPUT	CODE (HEX)	VECTOR ADDRESS (HEX)
RST 0	C7	0000
RST 1	CF	0008
RST 2	D7	0010
RST 3	DF	0018
RST 4	E7	0020
TRAP	HARDWARE INPUT	0024
RST 5	EF	0028
RST 5.5	HARDWARE INPUT	002C
RST 6	F7	0030
RST 6.5	HARDWARE INPUT	0034
RST 7	FF	0038
RST 7.5	HARDWARE INPUT	003C

Table C-2

SDK-85 INTERRUPT ALLOCATION

INPUT	FUNCTION	LOCATION OF SERVICE ROUTINE*
RST 5.5	Keyboard interrupt	Monitor ROM
RST 6.5	User interrupt	20C8 or 20CE
RST 7.5	Vectored interrupt key	20CE or 20D4
TRAP	8155 timer interrupt	Monitor ROM
INTR	User interrupt	20C2 or 20C8 (RST 5)
		20C5 or 20CB (RST 6)
		20CB or 20D1 (RST 7)

*The alternative service addresses refer to SDK-85 monitor versions 2.1 (lower address) and 1.2 (higher address).

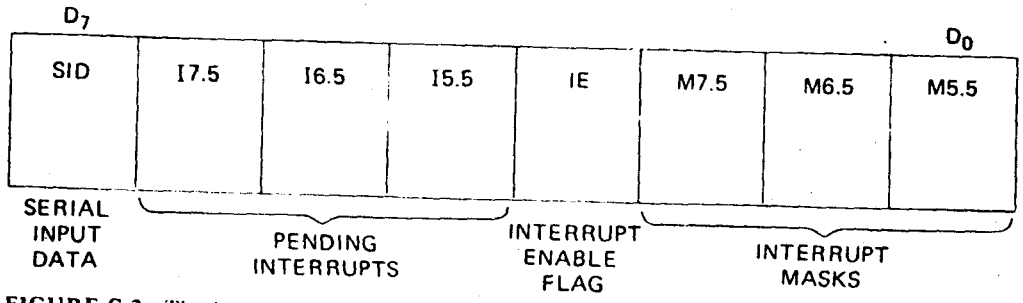


FIGURE C-2. The interrupt mask (I) register (as read by a RIM instruction).

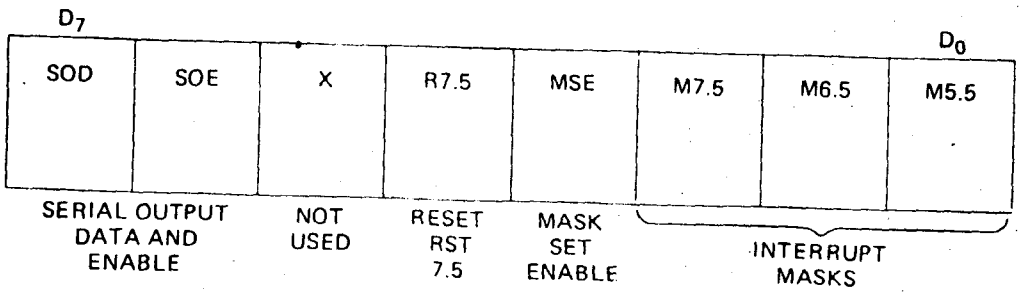


FIGURE C-3. The interrupt mask (I) register (as set by a SIM instruction).