2521/302 2602/302
2601/302 2603/302
MICROCONTROLLER TECHNOLOGY
AND MICROPROCESSOR SYSTEMS
June/July 2019
Time: 3 hours

THE KENYA NATIONAL EXAMINATIONS COUNCIL

DIPLOMA IN ELECTRICAL AND ELECTRONIC ENGINEERING
(POWER OPTION)
(TELECOMMUNICATION OPTION)
(INSTRUMENTATION OPTION)
MODULE III

MICROCONTROLLER TECHNOLOGY AND
MICROPROCESSOR SYSTEMS

3 hours

INSTRUCTIONS TO CANDIDATES

You should have the following for this examination:
Answer booklet;
Scientific calculator;
Intel 8080/8085 microprocessor instruction set;
8051 Microcontroller instruction set.
*The paper consists of **TWO** sections; **A** and **B**.*
*Answer any **THREE** questions from Section **A** and any **TWO** questions from Section **B** in the answer booklet provided.*
All questions carry equal marks.
Maximum marks for each part of a question are as indicated.
Candidates should answer all questions in English.

This paper consists of 10 printed pages.

Candidates should check the question paper to ascertain that all the pages are printed as indicated and that no questions are missing.

SECTION A: MICROPROCESSOR SYSTEMS

*Answer **THREE** questions from this section.*

1. (a) State **three** flags of a CPU status register and explain the function of each. (6 marks)

(b) Table 1 shows Intel 8085 machine code program.

Table 1

Machine code (Hex)
21
08
20
7E
23
32
10
30
70

(i) Decode the program into Intel 8085 mnemonics;

(ii) State the addressing mode of each instruction.

(8 marks)

(c) Write assembly language program segments to perform each of the following:

(i) 10100101_2 AND 23 H;
(ii) $(33 - 6) \times 2$.

(6 marks)

2. (a) Describe **two** modes of direct memory access. (4 marks)

(b) State **four** functions of interface modules of a microprocessor system. (4 marks)

(c) Figure 1 shows a schematic diagram of an Input/Output (I/O) port.

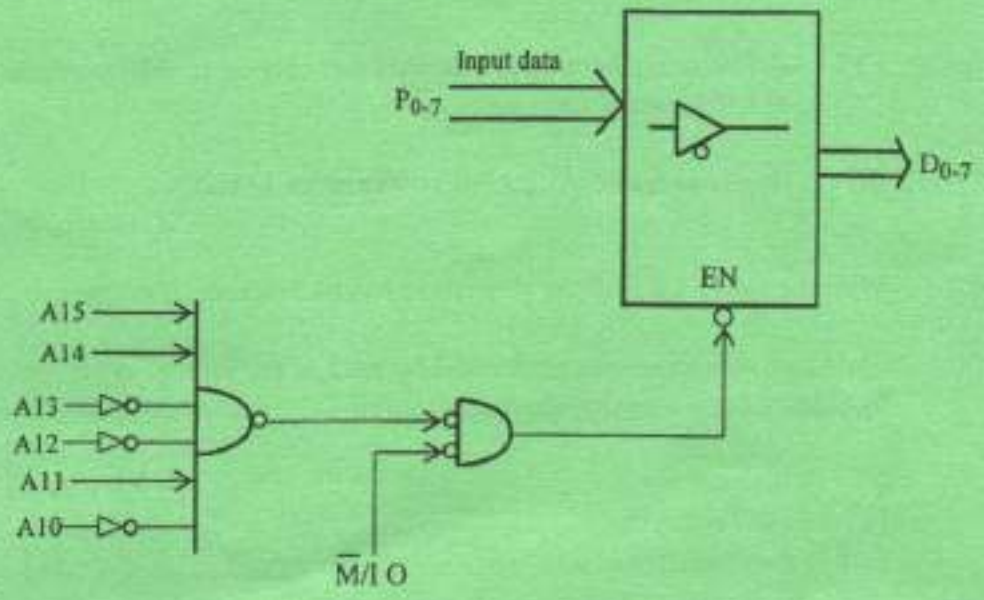

Fig. 1

(i) State with reasons, whether the port is I/O - mapped or memory-mapped.

(ii) Determine the range of addresses assigned to the port, in hexadecimal.

(iii) With the aid of a flowchart, write an assembly language program segment to read the port and branch to location READY when bits P_0 and P_1 are both low, else branch to WAIT. (12 marks)

3. (a) With aid of a flowchart, describe a CPU instrument cycle. (6 marks)

(b) Write an assembly language program to perform the following:

- sum two BCD numbers located in memory locations 4001 H and 4002 H;
- subtract the BCD number 01000111 from the result;
- store the final BCD answer in memory location 4003 H;
- end.

(8 marks)

(c) Table 2 shows an Intel 8085 assembly language program listing. Draw trace table for the program. (6 marks)

Table 2

LX1 H, 4008 H
LX1 D, 3647 H
DAD D
SHLD 4002 H
HLT

4. (a) Draw a block diagram of a microprogrammed control unit and state the function of each block. (8 marks)

(b) (i) Differentiate between horizontal and vertical micro-operation field with respect to CPU control unit.

(ii) State **one** merit of each micro-operation in b(i).

(4 marks)

(c) With the aid of a flowchart, describe a microcomputer development cycle. (8 marks)

5. (a) State **two** measurements preformed by each of the following equipment while troubleshooting a microprocessor system:

(i) multimeter;
(ii) oscilloscope;
(iii) logic analyser.

(6 marks)

(b) Table 3 shows the contents of a ROM.

Table 3

Memory Address (Hex)	ROM contents (Hex)
3000	E7
3001	5D
3002	F0
3003	06
3004	C6

(i) Determine the checksum for the ROM;

(ii) Write an assembly language program to evaluate the checksum.

(10 marks)

(c) Explain **two** commands found in the monitor program of a microcomputer. (4 marks)

SECTION B: MICROCONTROLLER TECHNOLOGY

*Answer **TWO** questions from this section.*

6. (a) State **four** microcontroller on-chip components that are not found in a general purpose microprocessor. (4 marks)

(b) Table 4 shows an 8051 microcontroller assembly language program listing.

Table 4

MOV 66 H, #20 H
MOV A, #66 H
MOV RO, A
MOV A, @ RO

(i) State the addressing mode of each instruction.

(ii) Draw a trace table for the program.

(iii) State the contents of register A at the end of the program execution.

(9 marks)

(c) (i) State the equation of a proportional plus derivative (PD) controller.

(ii) Derive the transfer function of the controller in c(i).

(iii) State the effects of the PD controller on a process.

(7 marks)

7. (a) Figure 2 shows a diagram of a pneumatic time-delay actuator. Describe its operation. (5 marks)

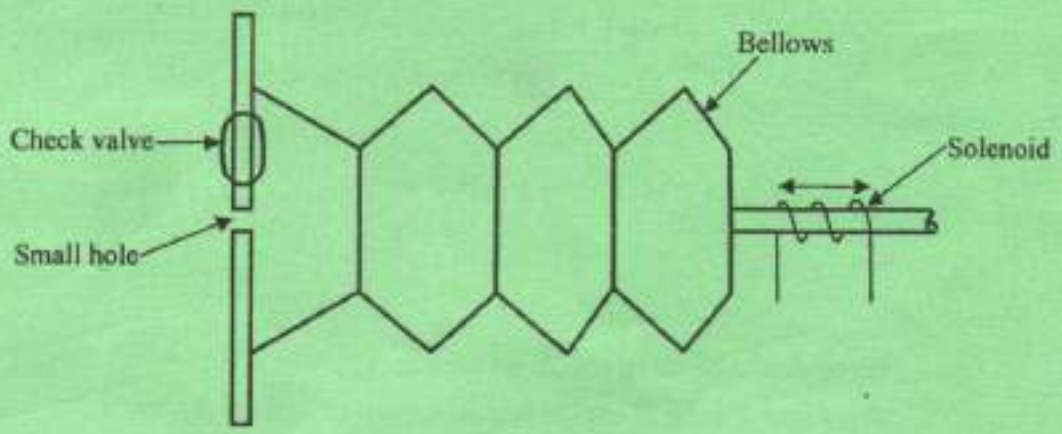

Fig. 2

(b) A data logging system monitors 15 analog loops. These loops are multiplexed into a microcomputer through an analog-to-digital converter (ADC). The computer requires 5 μS to perform each instruction. 100 instructions are required to address each multiplexer line, read in and process data in that line. The multiplexer requires 25 μS to select and capture the value of an input line. The ADC performs the conversion in 30 μS. Determine the:

(i) total time required to process all th 15 lines;

(ii) sampling rate of each line.

(7 marks)

(c) Figure 3 shows a pick-and-place robot that picks up parts from one conveyer belt and places them on another belt. When a part moving along the lower conveyer belt activates switch SW_1, a solenoid-powered gripper picks the part and carries it towards the upper conveyer belt. When the gripper reaches switch SW_2, it releases the part and moves back to pick another part. When the gripper reaches switch SW_3, it halts and waits for the next part to start the cycle all over again. Draw a relay logic ladder diagram to control this operation. (8 marks)

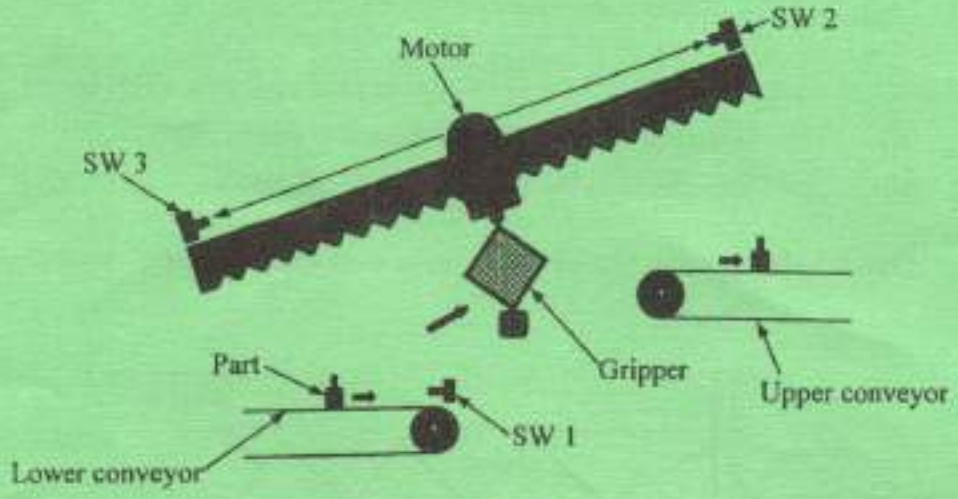

Fig. 3

8. (a) State:

(i) **three** merits of using robots in industries;

(ii) **three** specifications considered when selecting a robot.

(6 marks)

(b) Figure 4 shows a diagram of a robot arm. Identify the parts labelled W, X, Y and Z.
(4 marks)

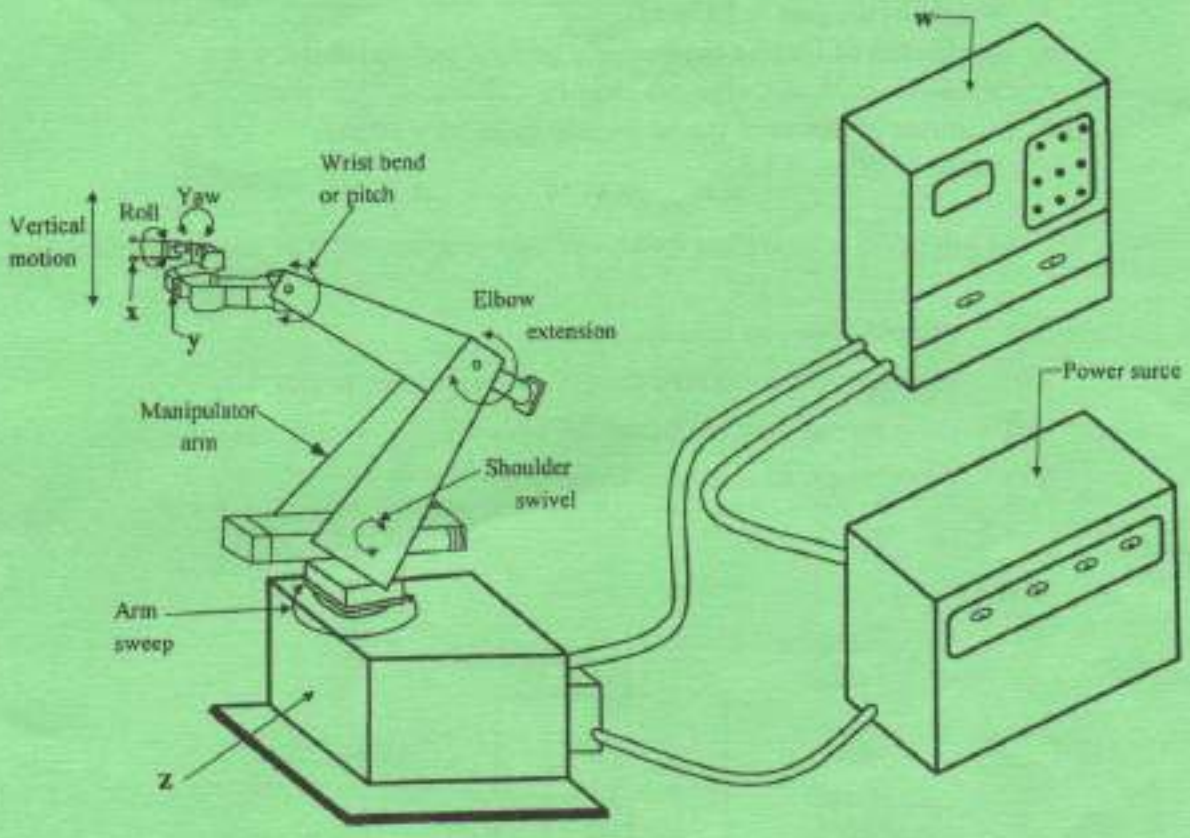

Fig. 4

(c) Figure 5 shows a schematic diagram of a robot gripper mechanism, using friction to grasp an object. The following data relate to the gripper:

- weight of the part = 40 N
- coeffieicnt of friction between the gripper pad and object = 0.4
- lengths l_1 = 75 mm, l_2 = 50 mm, l_3 = 20 mm, l_4 = 56 mm.
- diameter of piston of the pneumatic cylinder = 80 mm
- factor of safety = 1.5

If the gripper is accelerating down with acceleration, a = 9.81 m/s^2, determine the:

(i) gripping force to hold the part;

(ii) actuation force required to obtain this gripping force;

(iii) pressure required to operate the piston;

(iv) power required if the discharge is 0.015 m^3/s.

(10 marks)

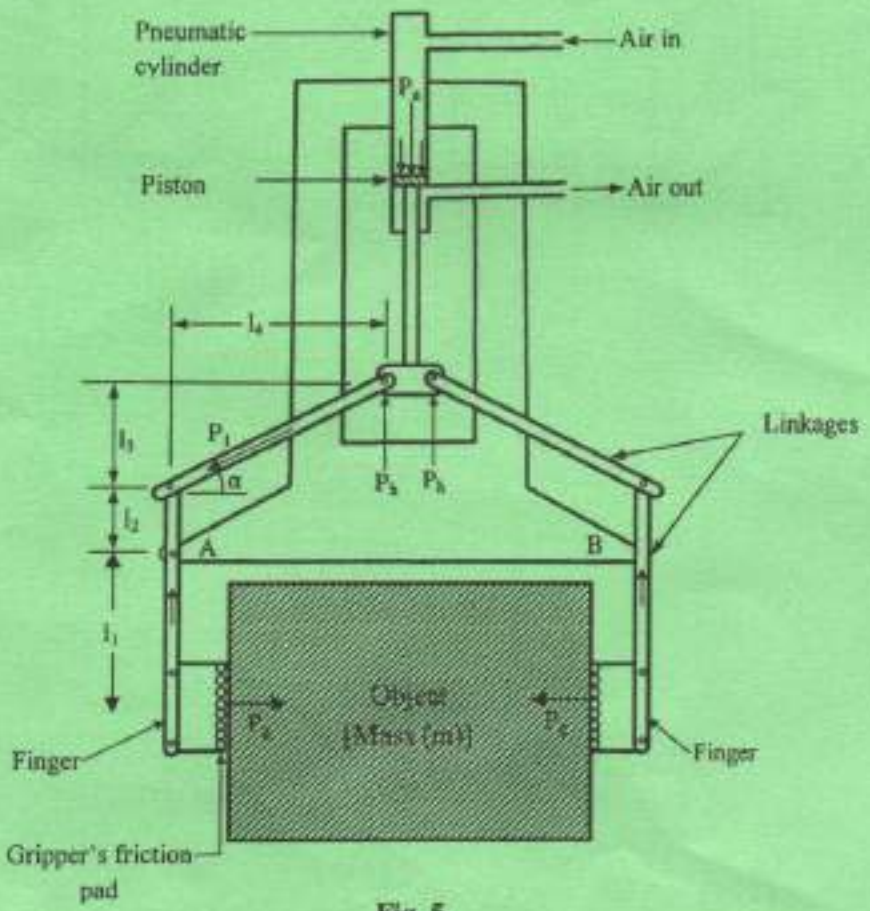

Fig. 5

Instruction set of

8080/8085

OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC
00	NOP	28	DCX H	56	MOV D,M	81	ADD C	AC	XRA H	D7	RST 2
01	LXI B,D16	2C	INR L	57	MOV D,A	82	ADD D	AD	XRA L	D8	RC
02	STAX B	2D	DCR L	58	MOV E,B	83	ADD E	AE	XRA M	D9	–
03	INX B	2E	MVI L,D8	59	MOV E,C	84	ADD H	AF	XRA A	DA	JC Adr
04	INR B	2F	CMA	5A	MOV E,D	85	ADD L	B0	ORA B	DB	IN D8
05	DCR B	30	SIM	5B	MOV E,E	86	ADD M	B1	ORA C	DC	CC Adr
06	MVI B,D8	31	LXI SP,D16	5C	MOV E,H	87	ADD A	B2	ORA D	DD	–
07	RLC	32	STA Adr	5D	MOV E,L	88	ADC B	B3	ORA E	DE	SBI D8
08	–	33	INX SP	5E	MOV E,M	89	ADC C	B4	ORA H	DF	RST 3
09	DAD B	34	INR M	5F	MOV E,A	8A	ADC D	B5	ORA L	E0	RPO
0A	LDAX B	35	DCR M	60	MOV H,B	8B	ADC E	B6	ORA M	E1	POP H
0B	DCX B	36	MVI M,D8	61	MOV H,C	8C	ADC H	B7	ORA A	E2	JPO Adr
0C	INR C	37	STC	62	MOV H,D	8D	ADC L	B8	CMP B	E3	XTHL
0D	DCR C	38	–	63	MOV H,E	8E	ADC M	B9	CMP C	E4	CPO Adr
0E	MVI C,D8	39	DAD SP	64	MOV H,H	8F	ADC A	BA	CMP D	E5	PUSH H
0F	RRC	3A	LDA Adr	65	MOV H,L	90	SUB B	BB	CMP E	E6	ANI D8
10	–	3B	DCX SP	66	MOV H,M	91	SUB C	BC	CMP H	E7	RST 4
11	LXI D,D16	3C	INR A	67	MOV H,A	92	SUB D	BD	CMP L	E8	RPE
12	STAX D	3D	DCR A	68	MOV L,B	93	SUB E	BE	CMP M	E9	PCHL
13	INX D	3E	MVI A,D8	69	MOV L,C	94	SUB H	BF	CMP A	EA	JPE Adr
14	INR D	3F	CMC	6A	MOV L,D	95	SUB L	C0	RNZ	EB	XCHG
15	DCR D	40	MOV B,B	6B	MOV L,E	96	SUB M	C1	POP B	EC	CPE Adr
16	MVI D,D8	41	MOV B,C	6C	MOV L,H	97	SUB A	C2	JNZ Adr	ED	–
17	RAL	42	MOV B,D	6D	MOV L,L	98	SBB B	C3	JMP Adr	EE	XRI D8
18	–	43	MOV B,E	6E	MOV L,M	99	SBB C	C4	CNZ Adr	EF	RST 5
19	DAD D	44	MOV B,H	6F	MOV L,A	9A	SBB D	C5	PUSH B	F0	RP
1A	LDAX D	45	MOV B,L	70	MOV M,B	9B	SBB E	C6	ADI D8	F1	POP PSW
1B	DCX D	46	MOV B,M	71	MOV M,C	9C	SBB H	C7	RST 0	F2	JP Adr
1C	INR E	47	MOV B,A	72	MOV M,D	9D	SBB L	C8	RZ	F3	DI
1D	DRC E	48	MOV C,B	73	MOV M,E	9E	SBB M	C9	RET Adr	F4	CP Adr
1E	MVI E,D8	49	MOV C,C	74	MOV M,H	9F	SBB A	CA	JZ	F5	PUSH PSW
1F	RAR	4A	MOV C,D	75	MOV M,L	A0	ANA B	CB	–	F6	ORI D8
20	RIM	4B	MOV C,E	76	HLT	A1	ANA C	CC	CZ Adr	F7	RST 6
21	LXI H,D16	4C	MOV C,H	77	MOV M,A	A2	ANA D	CD	CALL Adr	F8	RM
22	SHLD Adr	4D	MOV C,L	78	MOV A,B	A3	ANA E	CE	ACI D8	F9	SPHL
23	INX H	4E	MOV C,M	79	MOV A,C	A4	ANA H	CF	RST 1	FA	JM Adr
24	INR H	4F	MOV C,A	7A	MOV A,D	A5	ANA L	D0	RNC	FB	EI
25	DCR H	50	MOV D,B	7B	MOV A,E	A6	ANA M	D1	POP D	FC	CM Adr
26	MVI H,D8	51	MOV D,C	7C	MOV A,H	A7	ANA A	D2	JNC Adr	FD	–
27	DAA	52	MOV D,D	7D	MOV A,L	A8	XRA B	D3	OUT D8	FE	CPI D8
28	–	53	MOV D,E	7E	MOV A,M	A9	XRA C	D4	CNC Adr	FF	RST 7
29	DAD H	54	MOV D,H	7F	MOV A,A	AA	XRA D	D5	PUSH D		
2A	LHLD Adr	55	MOV D,L	80	ADD B	AB	XRA E	D6	SUI D8		

D8 = constant, or logical/arithmetic expression that evaluates to an 8-bit data quantity. D16 = constant, or logical/arithmetic expression that evaluates to a 16-bit data quantity. Adr = 16-bit address.

MCS-51™ Instruction Set Summary

ARITHMETIC OPERATIONS

Mnemonic		Description	Bytes	Cycles
ADD	A,Rn	Add register to A	1	1
ADD	A,direct	Add direct byte to A	2	1
ADD	A,@Ri	Add indirect RAM to A	1	1
ADD	A,#data	Add immediate data to A	2	1
ADDC	A,Rn	Add register to A with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry	1	1
ADDC	A,#data	Add immediate data to A with Carry	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immediate data from A with Borrow	2	1
INC	A	Increment A	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement A	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A & B (A = B => BA)	1	4
DIV	AB	Divide A by B (A/B => A + B)	1	4
DA	A	Decimal Adjust A	1	1

LOGICAL OPERATIONS

Mnemonic		Description	Bytes	Cycles
ANL	A,Rn	AND register to A	1	1
ANL	A,direct	AND direct byte to A	2	1
ANL	A,@Ri	AND indirect RAM to A	1	1
ANL	A,#data	AND immediate data to A	2	1
ANL	direct,A	AND A to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to A	1	1
ORL	A,direct	OR direct byte to A	2	1
ORL	A,@Ri	OR indirect RAM to A	1	1
ORL	A,#data	OR immediate data to A	2	1
ORL	direct,A	OR A to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to A	1	1
XRL	A,direct	Exclusive-OR direct byte to A	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR A to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2
CLR	A	Clear A	1	1
CPL	A	Complement A	1	1
RL	A	Rotate A Left	1	1
RLC	A	Rotate A Left through Carry	1	1
RR	A	Rotate A Right	1	1
RRC	A	Rotate A Right through Carry	1	1
SWAP	A	Swap nibbles within A	1	1

DATA TRANSFER

Mnemonic		Description	Bytes	Cycles
MOV	A,Rn	Move register to A	1	1
MOV	A,direct	Move direct byte to A	2	1
MOV	A,@Ri	Move indirect RAM to A	1	1
MOV	A,#data	Move immediate data to A	2	1
MOV	Rn,A	Move A to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move A to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move A to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load Data Pointer with 16-bit constant	3	1
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with A	1	1
XCH	A,direct	Exchange direct byte with A	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Bytes	Cycles
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry flag	2	2
ORL	C,bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry flag	2	2
MOV	C,bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Bytes	Cycles
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to DPTR	1	2
JZ	rel	Jump if A is Zero	2	2
JNZ	rel	Jump if A is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit,rel	Jump if direct Bit is set	3	2
JNB	bit,rel	Jump if direct Bit is Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Compare immediate to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Compare immed. to reg. & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Compare immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct byte & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes

- Rn — Working register R0-R7
- direct — 128 internal RAM locations, any I/O port, control or status register
- @Ri — indirect internal RAM location addressed by register R0 or R1
- #data — 8-bit constant included in instruction
- #data16 — 16-bit constant included in instruction
- bit — 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes

- addr16 — Destination address may be anywhere in 64-kByte program address space
- addr11 — Destination address will be within same 2-kByte page of program address space as first byte of the following instruction
- rel — 8-bit offset relative to first byte of following instruction (+127, -128)

Information from Intel Application Note AP-69

THIS IS THE LAST PRINTED PAGE.

2521/302 2602/302
2601/302 2603/302
June/July 2019