5-1. A computer uses a memory unit with 256K words of 32 bits each. A binary instruction code is stored in one word of memory. The instruction has four parts: an indirect bit, an operation code, a register code part to specify one of 64 registers, and an address part.

a. How many bits are there in the operation code, the register code part, and the address part?

b. Draw the instruction word format and indicate the number of bits in each part.

c. How many bits are there in the data and address inputs of the memory?

5-2. What is the difference between a direct and an indirect address instruction? How many references to memory are needed for each type of instruction to bring an operand into a processor register?

5-3. The following control inputs are active in the bus system shown in Fig. 5-4. For each case, specify the register transfer that will be executed during the next clock transition.

	S_2	S_1	S_0	LD of register	Memory	Adder
a.	1	1	1	*IR*	Read	—
b.	1	1	0	*PC*	—	—
c.	1	0	0	*DR*	Write	—
d.	0	0	0	*AC*	—	Add

5-4. The following register transfers are to be executed in the system of Fig. 5-4. For each transfer, specify: (1) the binary value that must be applied to bus select inputs S_2, S_1, and S_0; (2) the register whose LD control input must be active (if any); (3) a memory read or write operation (if needed); and (4) the operation in the adder and logic circuit (if any).

a. $AR \leftarrow PC$

b. $IR \leftarrow M[AR]$

c. $M[AR] \leftarrow TR$

d. $AC \leftarrow DR$, $DR \leftarrow AC$ (done simultaneously)

5-5. Explain why each of the following microoperations cannot be executed during a single clock pulse in the system shown in Fig. 5-4. Specify a sequence of microoperations that will perform the operation.

a. $IR \leftarrow M[PC]$

b. $AC \leftarrow AC + TR$

c. $DR \leftarrow DR + AC$ (AC does not change)

5-6. Consider the instruction formats of the basic computer shown in Fig. 5-5 and the list of instructions given in Table 5-2. For each of the following 16-bit instructions, give the equivalent four-digit hexadecimal code and explain in your own words what it is that the instruction is going to perform.

a. 0001 0000 0010 0100

b. 1011 0001 0010 0100

c. 0111 0000 0010 0000

5-7. What are the two instructions needed in the basic computer in order to set the *E* flip-flop to 1?

5-8. Draw a timing diagram similar to Fig. 5-7 assuming that *SC* is cleared to 0 at time T_3 if control signal C_7 is active.

$$C_7T_3: \quad SC \leftarrow 0$$

C_7 is activated with the positive clock transition associated with T_1.

5-9. The content of *AC* in the basic computer is hexadecimal A937 and the initial value of *E* is 1. Determine the contents of *AC*, *E*, *PC*, *AR*, and *IR* in hexadecimal after the execution of the CLA instruction. Repeat 11 more times, starting from each one of the register-reference instructions. The initial value of *PC* is hexadecimal 021.

5-10. An instruction at address 021 in the basic computer has $I = 0$, an operation code of the AND instruction, and an address part equal to 083 (all numbers are in hexadecimal). The memory word at address 083 contains the operand B8F2 and the content of *AC* is A937. Go over the instruction cycle and determine the contents of the following registers at the end of the execute phase: *PC*, *AR*, *DR*, *AC*, and *IR*. Repeat the problem six more times starting with an operation code of another memory-reference instruction.

5-11. Show the contents in hexadecimal of registers *PC*, *AR*, *DR*, *IR*, and *SC* of the basic computer when an ISZ indirect instruction is fetched from memory and executed. The initial content of *PC* is 7FF. The content of memory at address 7FF is EA9F. The content of memory at address A9F is 0C35. The content of memory at address C35 is FFFF. Give the answer in a table with five columns, one for each register and a row for each timing signal. Show the contents of the registers after the positive transition of each clock pulse.

5-12. The content of *PC* in the basic computer is 3AF (all numbers are in hexadecimal). The content of *AC* is 7EC3. The content of memory at address 3AF is 932E. The content of memory at address 32E is 09AC. The content of memory at address 9AC is 8B9F.

a. What is the instruction that will be fetched and executed next?

b. Show the binary operation that will be performed in the *AC* when the instruction is executed.

c. Give the contents of registers *PC*, *AR*, *DR*, *AC*, and *IR* in hexadecimal and the values of *E*, *I*, and the sequence counter *SC* in binary at the end of the instruction cycle.

5.1

256 K = $2^8 \times 2^{10} = 2^{18}$

64 = 2^6

(a) Address: 18 bits

Register code: 6 bits

Indirect bit: 1 bit

25 32 – 25 = 7 bits for opcode.

(b) 1 7 6 18 = 32 bits

I	opcode	Register	Address

(c) Data; 32 bits; address: 18 bits.

5.2

A direct address instruction needs two references to memory: (1) Read instruction; (2) Read operand.

An indirect address instruction needs three references to memory: (1) Read instruction; (2) Read effective address; (3) Read operand.

5.3

(a) Memory read to bus and load to IR: IR ← M[AR]
(b) TR to bus and load to PC: PC ← TR
(c) AC to bus, write to memory, and load to DR:
DR ← AC, M[AR]← AC
(d) Add DR (or INPR) to AC: AC ← AC + DR

5.4

(1) $S_2S_1S_0$ Load(LD) Memory Adder
(2) (3) (4)
(a) AR ← PC 010 (PC) AR — — (b) IR ← M[AR] 111 (M) IR Read — (c) M[AR] ← TR 110 (TR) — Write —
(d) DR ← AC — Transfer DR to AC
AC ← DR

5.5

100 (AC) DR and AC

(a) IR ← M[PC] PC cannot provide address to memory. Address must be transferred to AR first
AR← PC
IR ← M[AR]
(b) AC ← AC + TR Add operation must be done with DR. Transfer TR to DR first.
DR ← TR
AC ← AC + DR

(c) DR ← DR + AC Result of addition is transferred to AC (not DR). To save value of AC its content must be stored temporary in DR (or TR).

AC ← DR, DR ← AC (See answer to Problem 5.4(d))

AC ← AC + DR
AC ← DR, DR ← AC

5.6

(a) 0001 0000 0010 0010 = $(1024)_{16}$
ADD $(024)_{16}$
ADD content of M[024] to AC ADD 024

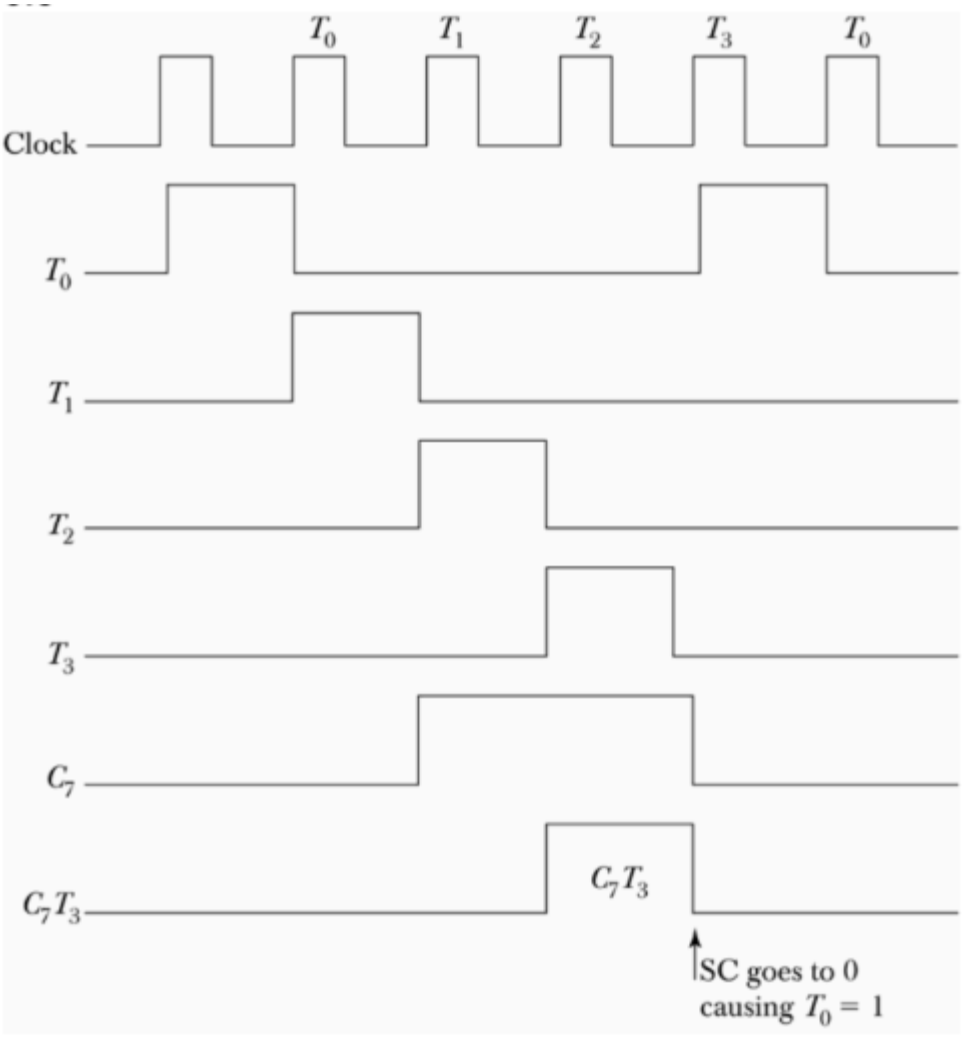
(b) 1 011 0001 0010 0100 = $(B124)_{16}$
I STA $(124)_6$
Store AC in M[M[124]] STA I 124

(c) 0111 0000 0010 0000 = $(7020)_{16}$
Register Increment AC INC

5.7

CLE Clear E
CME Complement E

5.8



5.9

	E	AC	PC	AR	IR
Initial	1	A937	021	—	—
CLA	1	0000	022	800	7800
CLE	0	A937	022	400	7400
CMA	1	56C8	022	200	7200
CME	0	A937	022	100	7100
CIR	1	D49B	022	080	7080
CIL	1	526F	022	040	7040

INC	1	A938	022	020	7020
SPA	1	A937	022	010	7010
SNA	1	A937	023	008	7008
SZA	1	A937	022	004	7004
SZE	1	A937	022	002	7002
HLT	1	A937	022	001	7001

5.10

	PC	AR	DR	AC	IR
Initial	021	—	—	A937	—
AND	022	083	B8F2	A832	0083
ADD	022	083	B8F2	6229	1083
LDA	022	083	B8F2	B8F2	2083
STA	022	083	—	A937	3083
BUN	083	083	—	A937	4083
BSA	084	084	—	A937	5083
ISZ	022	083	B8F3	A937	6083

5.11

PC AR DR IR SC

Initial	7FF	—	—	—	0
T_0	7FF	7FF	—	—	1
T_1	800	7FF	—	EA9F	2
T_2	800	A9F	—	EA9F	3
T_3	800	C35	—	EA9F	4
T_4	800	C35	FFFF	EA9F	5
T_5	800	C35	0000	EA9F	6

T_6	801	C35	0000	EA9F	0

5.12

(a) 9 = (1001)

1\001/

I=1 ADD ADD I 32E

Memory

3AF	932E
32E	09AC
9AC	8B9F

AC = 7EC3

(b)
AC = 7EC3 (ADD)
DR = 8B9F
0A62

E=1

(c) PC = 3AF + 1 = 3BO IR = 932E
AR = 7AC E = 1
DR = 8B9F I = 1
AC = 0A62 SC = 0000