1. Recall that the *control unit* of the processor drives the Fetch-Execute instruction cycle by generating an appropriate sequence of control signals. Consider the following simple computer.

FIGURE 7.1 Structure of a primitive computer in terms of buses and functional units

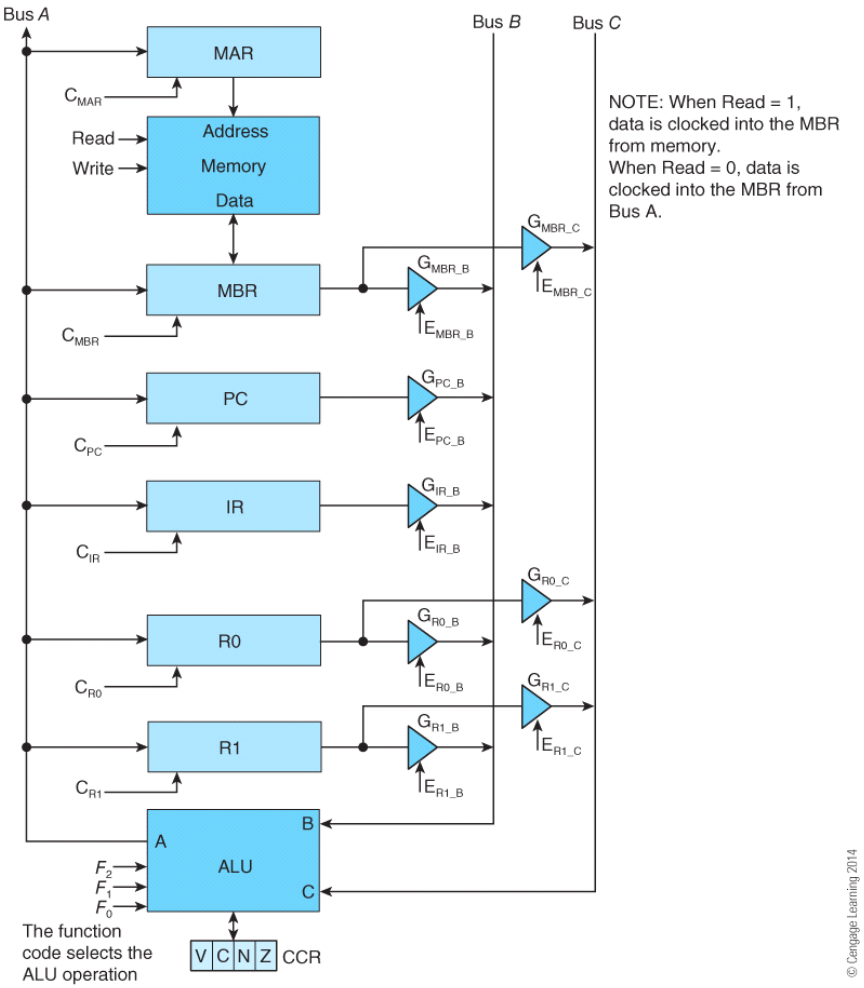

TABLE 7.1 ALU Function Codes for the Computer of Figure 7.1 (*A*, *B*, and *C* refer to the buses)

F_2	F_1	F_0	Operation	
0	0	0	Copy bus *B* to Bus *A*	$A = B$
0	0	1	Copy bus *C* to Bus *A*	$A = C$
0	1	0	Copy bus *B* + 1 to bus *A*	$A = B + 1$
0	1	1	Copy bus *C* + 1 to bus *A*	$A = C + 1$
1	0	0	Copy bus *B* − 1 to bus *A*	$A = B - 1$
1	0	1	Copy bus *C* − 1 to bus *A*	$A = C - 1$
1	1	0	Copy bus *B* + bus *C* to bus *A*	$A = B + C$
1	1	1	Copy bus *C* − bus *B* to bus *A*	$A = C - B$

TABLE 7.3 Machine-Level Instructions for the CPU of Figure 7.1

Op-Code	Name	Operation (Defined in RTL)
0 0 0	LOAD **R0**,M	[R0] ← [M]
0 0 1	LOAD **R1**,M	[R1] ← [M]
0 1 0	STORE **M**,R0	[M] ← [R0]
0 1 1	STORE **M**,R1	[M] ← [R1]
1 0 0	ADD **R1**,R0	[R1] ← [R1] + [R0]
1 0 1	SUB **R1**,R0	[R1] ← [R1] − [R0]
1 1 0	BRA T	[PC] ← T
1 1 1	BEQ T	IF [Z] = 1 THEN [PC] ← T

The fetch of the next instruction pointed at by the PC needs the following steps (micro-operations):

[MAR] ← [PC]
[PC] ← [PC] + 1 (Increment PC)
[MBR] ← Memory read from MAR addr.
[IR] ← [MBR]

Once fetched into the IR, the LOAD R0, M instruction needs the following steps to execute:

[MAR] ← $[IR]_{Address}$
[MBR] ← Memory read from MAR addr.
[R0] ← [MBR]

a) What would be the steps needed to execute the STORE M, R1 instruction?

b) What would be the steps needed to execute the ADD R1, R0 instruction?

c) What would be the steps needed to execute the BRA T instruction?

d) What would be the steps needed to execute the BEQ T instruction?

2. Add "hardware" (gates, MUX, decoder, etc.) to Figure 7.1 to allow for:
a) BEQ T which has the micro-operation: IF [Z] = 1 THEN [PC] ← T

b) The MBR needs to be able receive data from two places: Memory and Bus A. Add hardware to allow this with a control signal MBR C_{MUX}.

3. (a) Complete the control signals for its micro-operations needed for the fetch-execute of the LOAD R0, M instruction:

Step	**RTN**	**Step #**	Mem Read	Mem Write	C_{MAR}	C_{MBR}	C_{PC}	C_{IR}	C_{R0}	C_{R1}	E_{MBR_B}	E_{PC_B}	E_{IR_B}	E_{R0_B}	E_{R1_B}	E_{MBR_C}	E_{R0_C}	E_{R1_C}	F_2	F_1	F_0	MBR C_{MUX}
Fetch	MAR ← PC	T_0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	PC ← PC + 1	T_1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
	MBR ← M[MAR]	T_2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	IR ← MBR	T_3																				
Execute	MAR ← $IR_{Address}$	T_4																				
	MBR ← M[MAR]	T_5																				
	R0 ← MBR	T_6																				
		T_7																				

b) Complete the control signals for its micro-operations needed for the fetch-execute of the STORE M, R1 instruction:

Step	**RTN**	**Step #**	Mem Read	Mem Write	C_{MAR}	C_{MBR}	C_{PC}	C_{IR}	C_{R0}	C_{R1}	E_{MBR_B}	E_{PC_B}	E_{IR_B}	E_{R0_B}	E_{R1_B}	E_{MBR_C}	E_{R0_C}	E_{R1_C}	F_2	F_1	F_0	MBR C_{MUX}
Fetch	MAR ← PC	T_0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	PC ← PC + 1	T_1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
	MBR ← M[MAR]	T_2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	IR ← MBR	T_3																				
Execute	MBR ← R1	T_4																				
	MAR ←$IR_{Address}$	T_5																				
	M[MAR] ← MBR	T_6																				
		T_7																				

Types of Control Unit

1) A hardwired control unit uses a circuit to repeatedly generates control signals to fetch-execute the next machine-language instruction of the program.
2) A microprogrammed control unit running a special purpose "microprogram" inside the control unit to generates control signals to fetch-execute the next machine-language instruction of the program.

- Microinstructions are fetched, decoded, and executed in the same manner as regular instructions. This extra level of instruction interpretation is what makes microprogrammed control slower than hardwired control.
- The advantages of microprogrammed control are that it can support very complicated instructions and only the microprogram might need to be changed if the instruction set changes (or an error is found).

FIGURE 7.4 The microprogrammed control unit