Problems Part-6

-1. What is the difference between a microprocessor and a microprogram? Is it possible to design a microprocessor without a microprogram? Are all microprogrammed computers also microprocessors?

-2. Explain the difference between hardwired control and microprogrammed control. Is it possible to have a hardwired control associated with a control memory?

-3. Define the following: (a) microoperation; (b) microinstruction; (c) microprogram; (d) microcode.

.1

A microprocessor is a small size CPU (computer on a chip). Microprogram is a program for a sequence of microoperations. The control unit of a microprocessor can be hardwired or microprogrammed, depending on the specific design. A microprogrammed computer does not have to be a microprocessor.

.2

Hardwired control, by definition, does not contain a control memory.

.3

Micro operation - an elementary digital computer operation.
Micro instruction - an instruction stored in control memory.
Micro program - a sequence of microinstructions.
Micro code - same as microprogram.

-5. The system shown in Fig. 7-2 uses a control memory of 1024 words of 32 bits each. The microinstruction has three fields as shown in the diagram. The microoperations field has 16 bits.

a. How many bits are there in the branch address field and the select field?

b. If there are 16 status bits in the system, how many bits of the branch logic are used to select a status bit?

c. How many bits are left to select an input for the multiplexers?

Slide 16

Control memory = $2^{10} \times 32$

(a)

6	10	16
Select	Address	Micro operations

= 32 bits

(b) 4 bits

(c) 2 bits

6. The control memory in Fig. 7-2 has 4096 words of 24 bits each. Slide 16

a. How many bits are there in the control address register?

b. How many bits are there in each of the four inputs shown going into the multiplexers?

Slide 14

c. What are the number of inputs in each multiplexer and how many multiplexers are needed?

7. Using the mapping procedure described in Fig. 7-3, give the first microinstruction address for the following operation code: (a) 0010; (b) 1011; (c) 1111.

8. Formulate a mapping procedure that provides eight consecutive microinstructions for each routine. The operation code has six bits and the control memory has 2048 words.

.6

Control memory = $2^{12} \times 24$

(a) 12 bits

(b) 12 bits

(c) 12 multiplexers, each of size 4-to-1 line.

.7

(a) 0001000 = 8

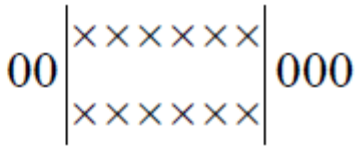
(b) 0101100 = 44

(c) 0111100 = 60

.8

opcode = 6 bits

control memory address = 11 bits



Problems Part-7

Slide 3

-1. A bus-organized CPU similar to Fig. 8-2 has 16 registers with 32 bits in each, an ALU, and a destination decoder.

a. How many multiplexers are there in the *A* bus, and what is the size of each multiplexer?
b. How many selection inputs are needed for MUX A and MUX B?
c. How many inputs and outputs are there in the decoder?
d. How many inputs and outputs are there in the ALU for data, including input and output carries?
e. Formulate a control word for the system assuming that the ALU has 35 operations.

-2. The bus system of Fig. 8-2 has the following propagation delay times: 30 ns for the signals to propagate through the multiplexers, 80 ns to perform the ADD operation in the ALU, 20 ns delay in the destination decoder, and 10 ns to clock the data into the destination register. What is the minimum cycle time that can be used for the clock?

.1

(a) 32 multiplexers, each of size 16 × 1.
(b) 4 inputs each, to select one of 16 registers.
(c) 4-to-16 – line decoder
(d) 32 + 32 + 1 = 65 data input lines
32 + 1 = 33 data output lines.

(e) 4 4 4 6 = 18 bits

SELA	SELB	SELD	OPR

2

30 + 80 + 10 = 120 n sec.

-5. Let *SP* = 000000 in the stack of Fig. 8-3. How many items are there in the stack if:

a. FULL = 1 and EMTY = 0?

b. FULL = 0 and EMTY = 1?

Slide 6

-6. A stack is organized such that *SP* always points at the next empty location on the stack. This means that *SP* can be initialized to 4000 in Fig. 8-4 and the first item in the stack is stored in location 4000. List the microoperations for the push and pop operations. Slide 7

.5

(a) Stack full with 64 items.

(b) stack empty

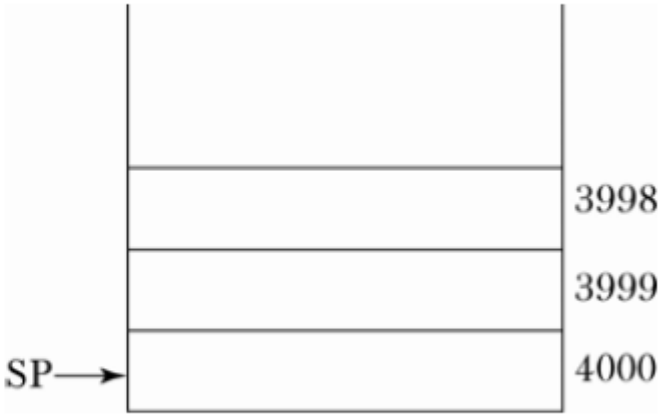
.6

PUSH : $M[SP] \leftarrow DR$

$SP \leftarrow SP - 1$

POP : $SP \leftarrow SP + 1$

$DR \leftarrow M[SP]$



-7. Convert the following arithmetic expressions from infix to reverse Polish notation.

a. $A * B + C * D + E * F$

b. $A * B + A * (B * D + C * E)$

c. $A + B * [C * D + E * (F + G)]$

d. $\dfrac{A * [B + C * (D + E)]}{F * (G + H)}$

(a) AB * CD * EF * + +

(b) AB * ABD * CE * + * +

(c) FG + E * CD * + B * A +

(d) ABCDE + * + * FGH + */

-8. Convert the following arithmetic expressions from reverse Polish notation to infix notation.

a. *A B C D E + * − /*

b. *A B C D E * / − +*

c. *A B C * / D − E F / +*

d. *A B C D E F G + * + * + *

(a) $\dfrac{A}{B-(D+E)*C}$

(b) $A+B\dfrac{C}{D*E}$

(c) $\dfrac{A}{B*C}-D+\dfrac{E}{F}$

(d) (((F + G) * E + D) * C + B) * A

-9. Convert the following numerical arithmetic expression into reverse Polish notation and show the stack operations for evaluating the numerical result.

$$(3 + 4)[10(2 + 6) + 8]$$

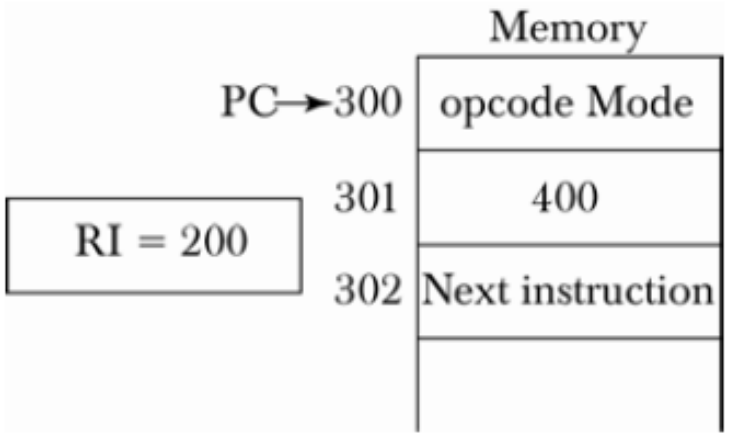
(3 + 4) [10 (2 + 6) + 8] = 616
RPN : 34 + 2 6 + 10 * 8 + *

				6		10		8		
	4		2	2	8	8	80	80	88	
3	3	7	7	7	7	7	7	7	7	616
3	4	+	2	6	+	10	*	8	+	*

-18. An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register $R1$ contains the number 200. Evaluate the effective address if the addressing mode of the instruction is (a) direct; (b) immediate; (c) relative; (d) register indirect; (e) index with $R1$ as the index register.

Effective address

(a) Direct: 400
(b) Immediate: 301
(c) Relative: 302 + 400 = 702
(d) Reg. Indirect: 200
(e) Indexed: 200 + 400 = 600



-33. What are the basic differences between a branch instruction, a call subroutine instruction, and program interrupt?

Branch instruction – Branch without being able to return.
Subroutine call – Branch to subroutine and then return to calling program.
Program interrupt – Hardware initiated branch with possibility to return.