(SOLUTIONS)

Name :
Student ID :

Digital Systems II
Final
2009-10 Spring

03.05.2010

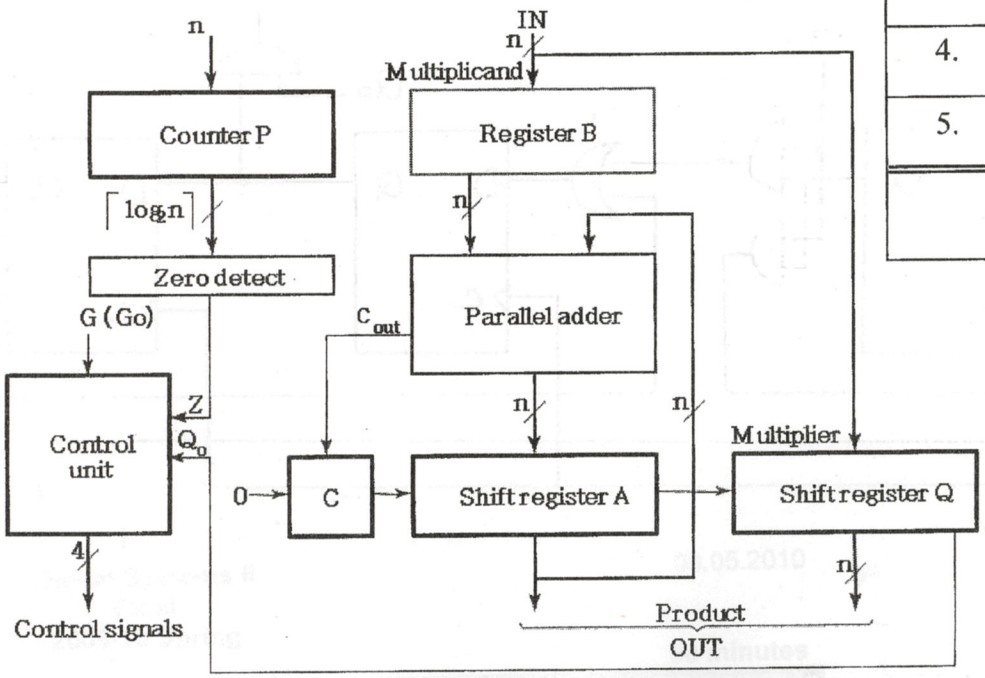
I/II... Öğretim

90 minutes

1.
2.
3.
4.
5.

Microprogrammed Control

1. Design a control unit using one flip-flop per state technique for the binary multiplier. The datapath and ASM chart of the binary multiplier are shown in Fig.1. (18pts)



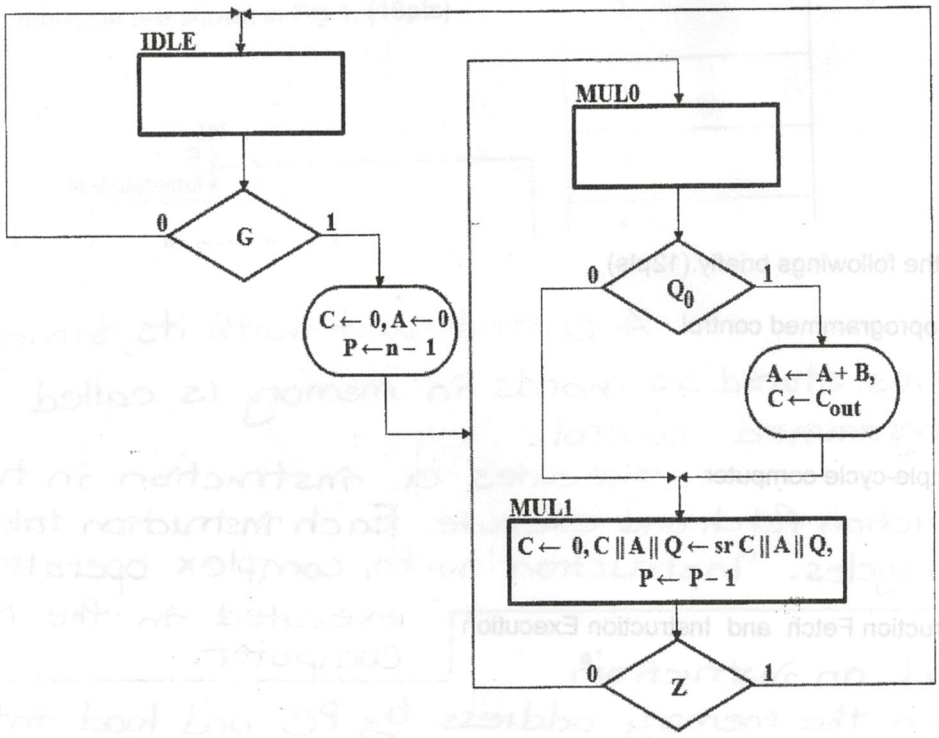
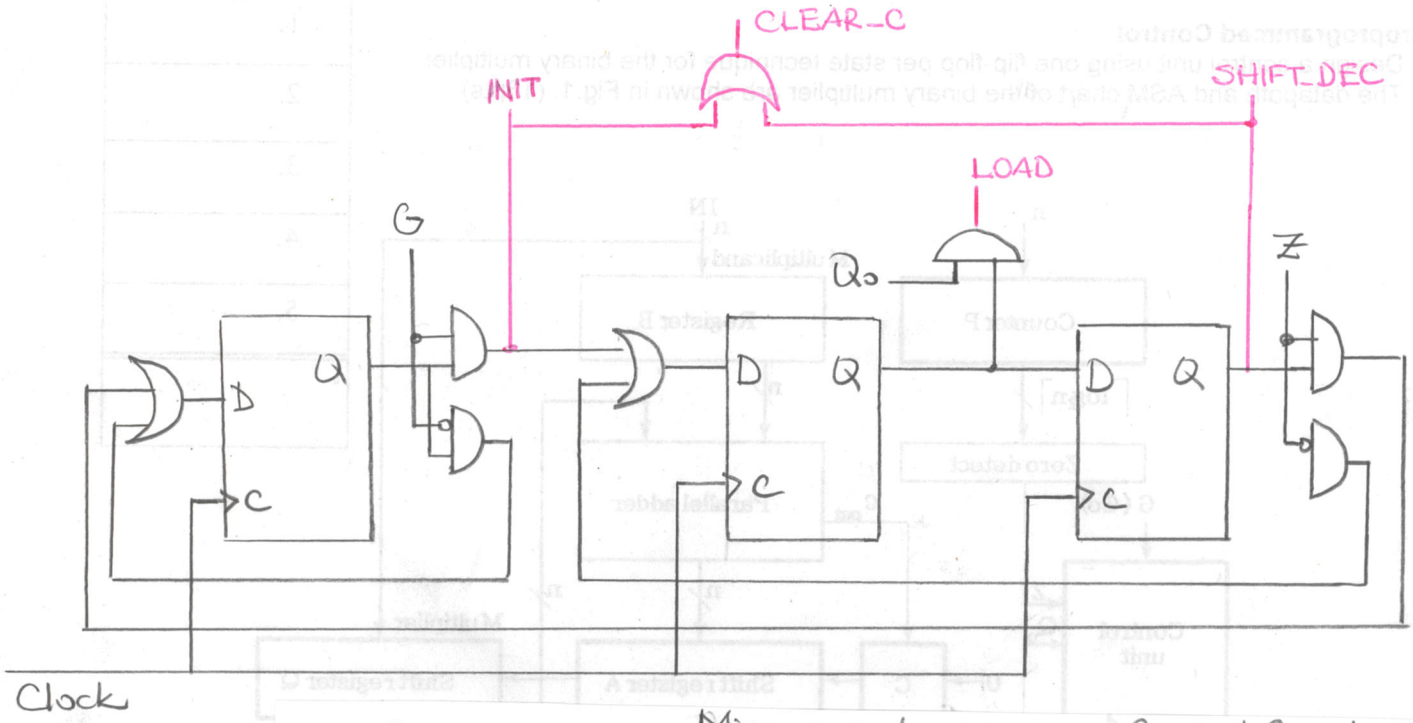


Fig 1. Datapath and ASM chart for the binary multiplier.



	Microoperation	Control Signal Name
Register A	A ← 0	INIT
	A ← A+B	LOAD
	C||A || Q ← sr C||A||Q	SHIFT_DEC
Flip Flop C	C ← 0	CLR_C
	C ← Cout	LOAD
Register Q	C||A||Q ← sr C||A||Q	SHIFT_DEC
Counter P	P ← n-1	INIT
	P ← P-1	SHIFT_DEC

2. Explain the followings briefly.(12pts)

- Microprogrammed control A control unit with its binary control values stored as words in memory is called a micro-programmed control.
- Multiple-cycle computer executes a instruction in two steps: instruction fetch and execute. Each instruction takes different clock cycles. Instruction with complex operations can be executed in the multiple-cycle computer.
- Instruction Fetch and Instruction Execution
  - Read an instruction from the memory address by PC and load into IR.
  - Apply control words to the datapath in clock cyles according to the opcode of the instruction.
- Barrel Shifter

Barrel shifter shifts or rotates the input data bit by the number of bit positions specified by a binary value on the set of selection bits.

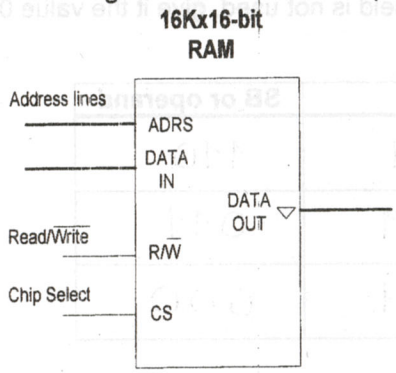
Name :
Student ID :

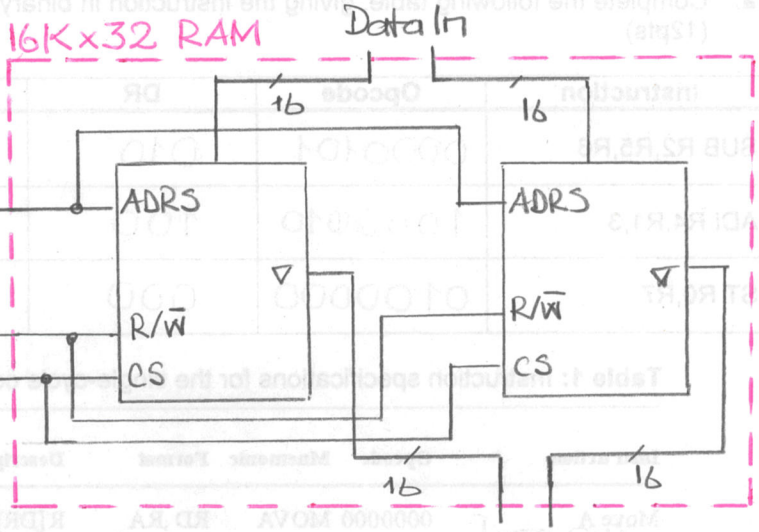
Digital Systems II
Final
2009-10 Spring

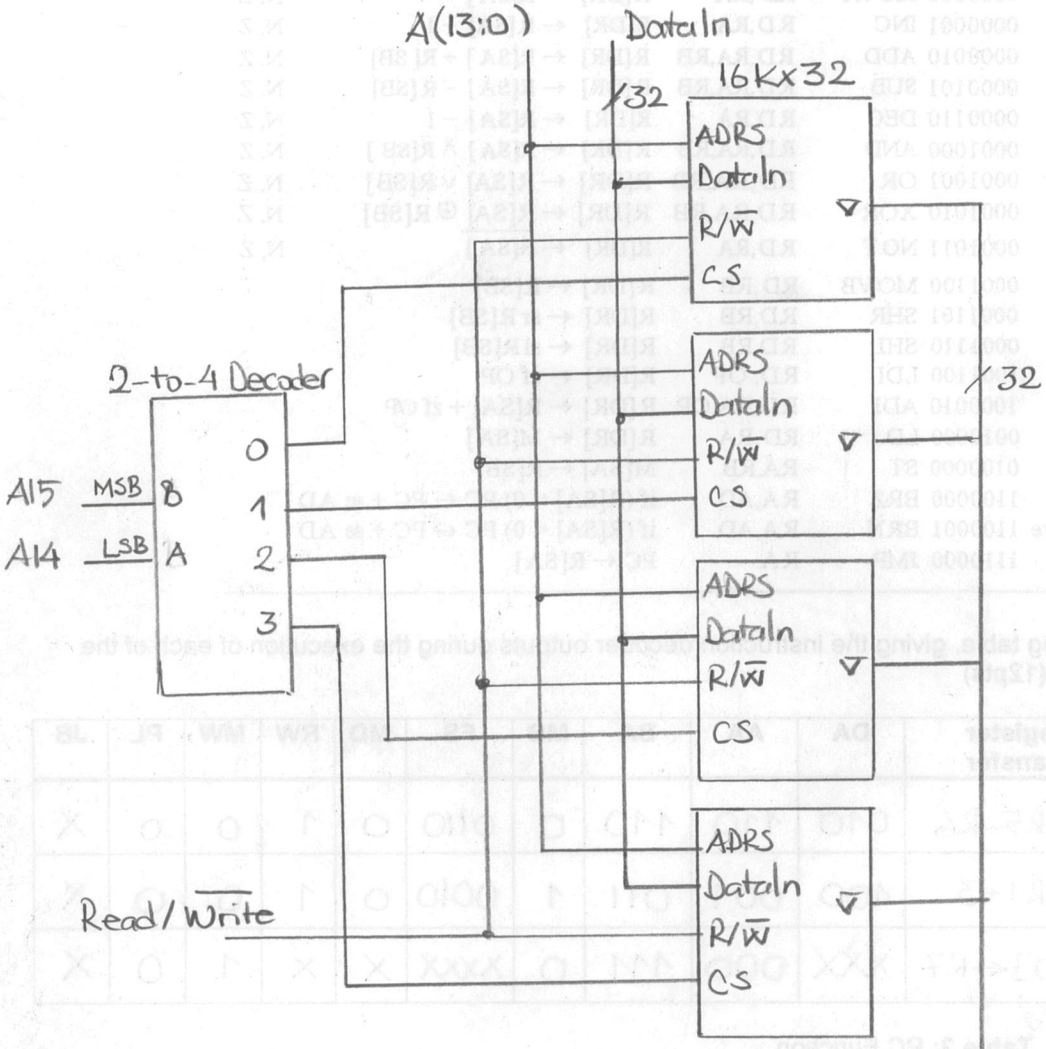
03.05.2010

Memory Basics

3. Using the 16Kx16 RAM chip and decoder, construct the block diagram for a 64Kx32 RAM. (20pts)







4. The single cycle computer in Fig 2. executes the following instructions :

SUB R2, R5, R6 Subtract
ADI R4, R5, 2 Add immediate
ST R0, R5 Store

a. Complete the following table, giving the instruction in binary (if any field is not used, give it the value 0). (12pts)

Instruction	Opcode	DR	SA	SB or operand
SUB R2,R5,R6	0000101	010	101	110
ADI R4,R1,3	1000010	100	001	011
ST R0,R7	0100000	000	111	000

Table 1: Instruction specifications for the single-cycle computer.

Instruction	Opcode	Mnemonic	Format	Description	Status Bits
Move A	0000000	MOVA	RD ,RA	R[DR] ← R[SA]	N, Z
Increment	0000001	INC	RD,RA	R[DR] ← R[SA] + 1	N, Z
Add	0000010	ADD	RD,RA,RB	R[DR] ← R[SA] + R[SB]	N, Z
Subtract	0000101	SUB	RD,RA,RB	R[DR] ← R[SA] − R[SB]	N, Z
Decrement	0000110	DEC	RD,RA	R[DR] ← R[SA] − 1	N, Z
AND	0001000	AND	RD,RA,RB	R[DR] ← R[SA] ∧ R[SB]	N, Z
OR	0001001	OR	RD,RA,RB	R[DR] ← R[SA] ∨ R[SB]	N, Z
Exclusive OR	0001010	XOR	RD,RA,RB	R[DR] ← R[SA] ⊕ R[SB]	N, Z
NOT	0001011	NOT	RD,RA	R[DR] ← $\overline{R[SA]}$	N, Z
Move B	0001100	MOVB	RD,RB	R[DR] ← R[SB]	
Shift Right	0001101	SHR	RD,RB	R[DR] ← sr R[SB]	
Shift Left	0001110	SHL	RD,RB	R[DR] ← sl R[SB]	
Load Immediate	1001100	LDI	RD, OP	R[DR] ← zf OP	
Add Immediate	1000010	ADI	RD,RA,OP	R[DR] ← R[SA] + zf OP	
Load	0010000	LD	RD,RA	R[DR] ← M[SA]	
Store	0100000	ST	RA,RB	M[SA] ← R[SB]	
Branch on Zero	1100000	BRZ	RA,AD	if (R[SA] = 0) PC ← PC + se AD	
Branch on Negative	1100001	BRN	RA,AD	if (R[SA] < 0) PC ← PC + se AD	
Jump	1110000	JMP	RA	PC ← R[SA]	

b. Complete the following table, giving the instruction decoder outputs during the execution of each of the instruction in part(a). (12pts)

Instruction	Register Transfer	DA	AA	BA	MB	FS	MD	RW	MW	PL	JB
SUB R2,R5,R6	R2←R5−R6	010	110	110	0	0110	0	1	0	0	X
ADI R4,R1,3	R4←R1+3	100	001	011	1	0010	0	1	0	0	X
ST R0, R7	M[R0]←R7	XXX	000	111	0	XXXX	X	X	1	0	X

Table 2: PC Function

PC Operation	PL	JB	BC
Count Up	0	X	X
Jump	1	1	X
Branch on Negative (else Count Up)	1	0	1
Branch on Zero (else Count Up)	1	0	0

Name :
Student ID :

Digital Systems II
Final
2009-10 Spring

03.05.2010

Table 3: Encoding of control word for the datapath in the single-cycle computer.

DA, AA, BA		MB		FS		MD		RW	
Function	**Code**	**Function**	**Code**	**Function**	**Code**	**Function**	**Code**	**Function**	**Code**
$R0$	000	Register	0	$F \leftarrow A$	0000	Function	0	No write	0
$R1$	001	Constant	1	$F \leftarrow A+1$	0001	Data In	1	Write	1
$R2$	010			$F \leftarrow A+B$	0010				
$R3$	011			$F \leftarrow A+\overline{B}+1$	0011				
$R4$	100			$F \leftarrow A+\overline{B}$	0100				
$R5$	101			$F \leftarrow A+\overline{B}+1$	0101				
$R6$	110			$F \leftarrow A-1$	0110				
$R7$	111			$F \leftarrow A$	0111				
				$F \leftarrow A \wedge B$	1000				
				$F \leftarrow A \vee B$	1001				
				$F \leftarrow A \oplus B$	1010				
				$F \leftarrow \overline{A}$	1011				
				$F \leftarrow B$	1100				
				$F \leftarrow \text{sr } B$	1101				
				$F \leftarrow \text{sl } B$	1110				

c. Assume that the propagation delay of the components are as follows:

PC	1 ns
Instruction Memory	4 ns
Register File	3 ns
MUX B, MUX D	1 ns
Function Unit	4 ns
Data Memory	4 ns
Register File	3 ns
Zero Fill	0 ns
Sign Extension	0 ns

Find the components used in the execution of the SUB R2,R5,R6 instruction and calculate the execution time of the instruction. (06pts)

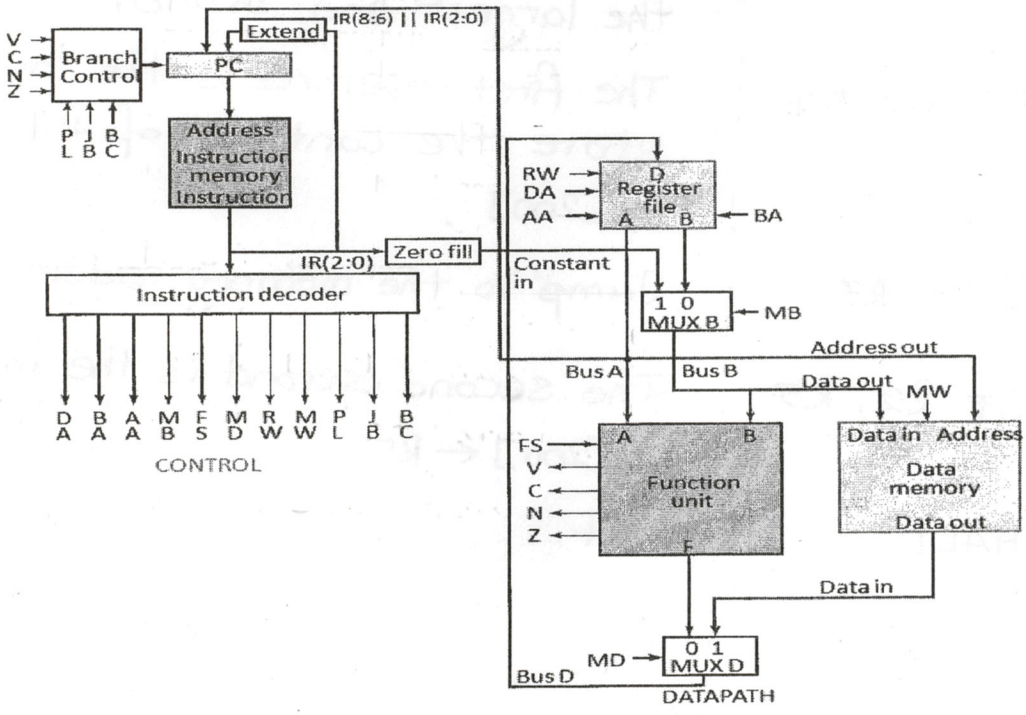


Fig. 2: Single-cycle computer

PC, Instruction Memory, Register File, MUX B, Function Unit, MUX D, Register File

1 + 4 + 3 + 1 + 4 + 1 + 3 = 17 ns

5. Write an assembly program that finds the largest value of two memory contents. The program compares the memory locations 100 and 101 and finds the largest of them. The result will be stored in the memory location 200. Assume that the R0 contains 100 and R1 contains 200. The instruction specifications of the single-cycle computer are given in Table 1. Use comments with instructions to make your program readable.(20pts)

Procedural steps:

- Read the operands from the memory.
- Subtract the first operand from the second.
- If the difference<0 then second operand is the largest, otherwise the first operand is largest.
- Store the largest value in M[200].
- Halt (not available in the instruction list).

Mem. Add.		
0	LDI R7, 7	Set the jump address
1	ADI R7, R7, 3	
2	LD R4, R1	Read the first operand R4 ← M[100]
3	INC R1, R1	Increment
4	LD R5, R1	Read the second operand R5 ← M[101]
5	SUB R6, R4, R5	Subtract the second operand from the first.
6	BRN R6, 3	If N=1, the second operand is the largest and branch.
7	ST R2, R4	The first operand is the largest, store the contents of R4 into M[200].
8	JMP R7	Jump to the memory address 10.
9	ST R2, R5	The second operand is the largest. M[200] ← R5
10	HALT	Halt.