04.06.2009

Name :
Student ID :

Digital Systems II, Final Exam
2008-09 Spring

SOLUTIONS

90 min.

Sequencing and Control

1. Design a binary multiplier that multiplies two unsigned binary numbers by the repeated addition method. For example, to multiply 5 by 4, the digital system adds the multiplicand four times: 5+5+5+5=20. The datapath of the multiplier is shown in Fig. 1. In the datapath, the multiplicand is in register BR; the multiplier is register AR; and the product will be register PR. An adder circuit adds the contents of BR to PR and store the sum in PR again, PR←PR+BR. AR is a down-counter. A zero-detection circuit Z checks when AR becomes zero after each time that is decremented.
   - **a.** In the binary multiplier, after signal START goes high, the multiplicand and multiplier are loaded from BUS into the corresponding registers in two successive clock cycles. By repeated addition method, the product is obtained in PR register. Adding BR to PR is stopped when the contents of AR is zero, Z=1.

     Find the ASM chart for the binary multiplier. [20 pts]

   - **b.** Design the control unit by the one flip-flop per state method. [20 pts]

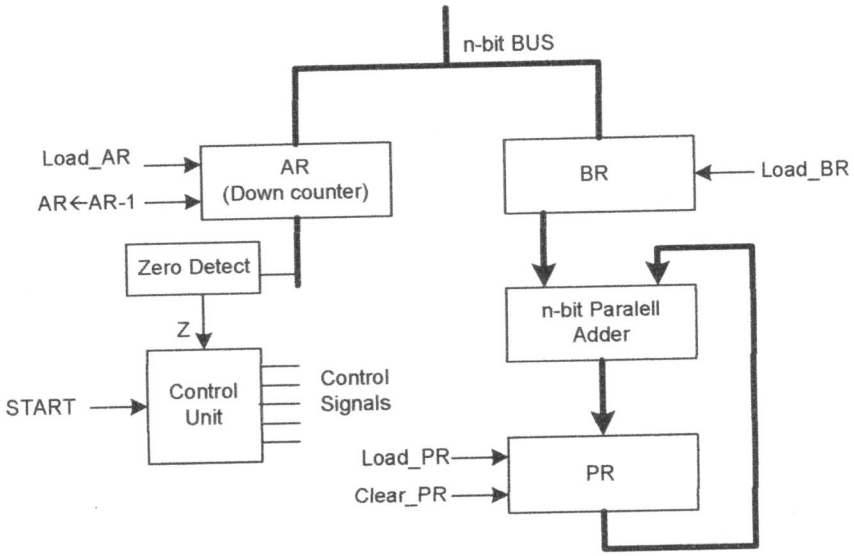
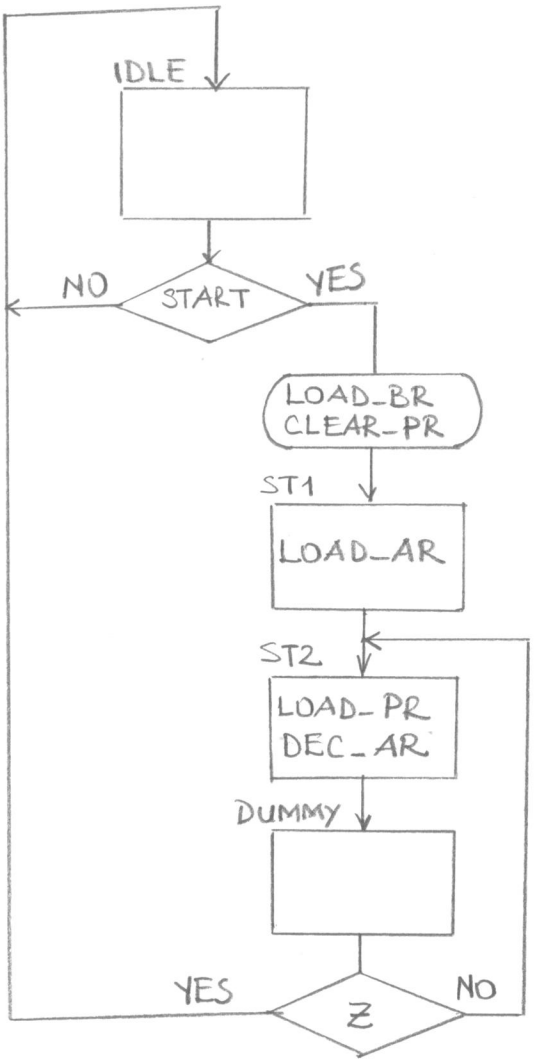
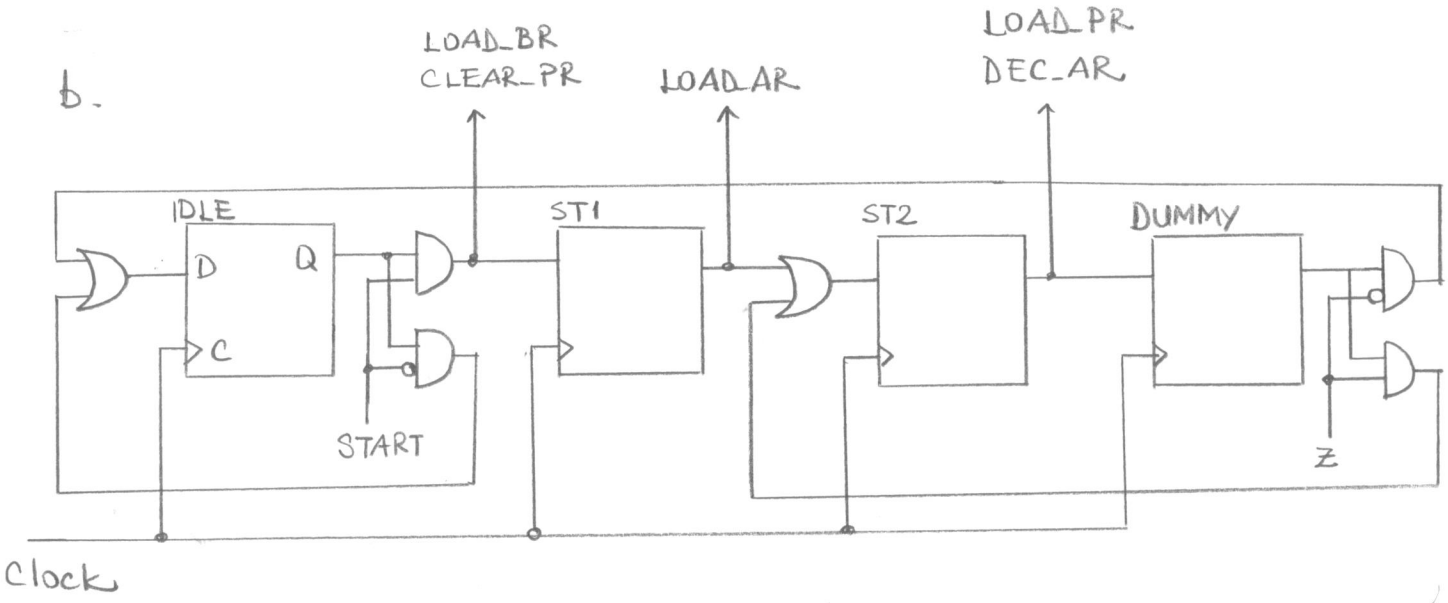


Figure 1 Datapath of the binary multiplier

1.a. ASM chart



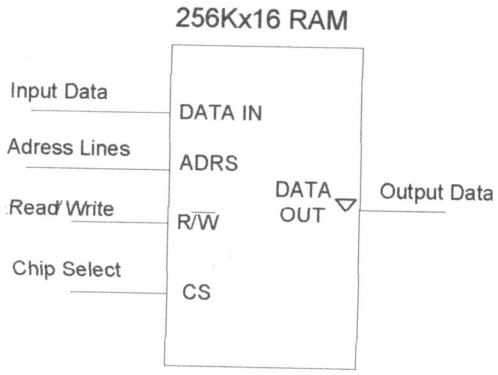
b.



Name:
Student ID:

Memory Basics

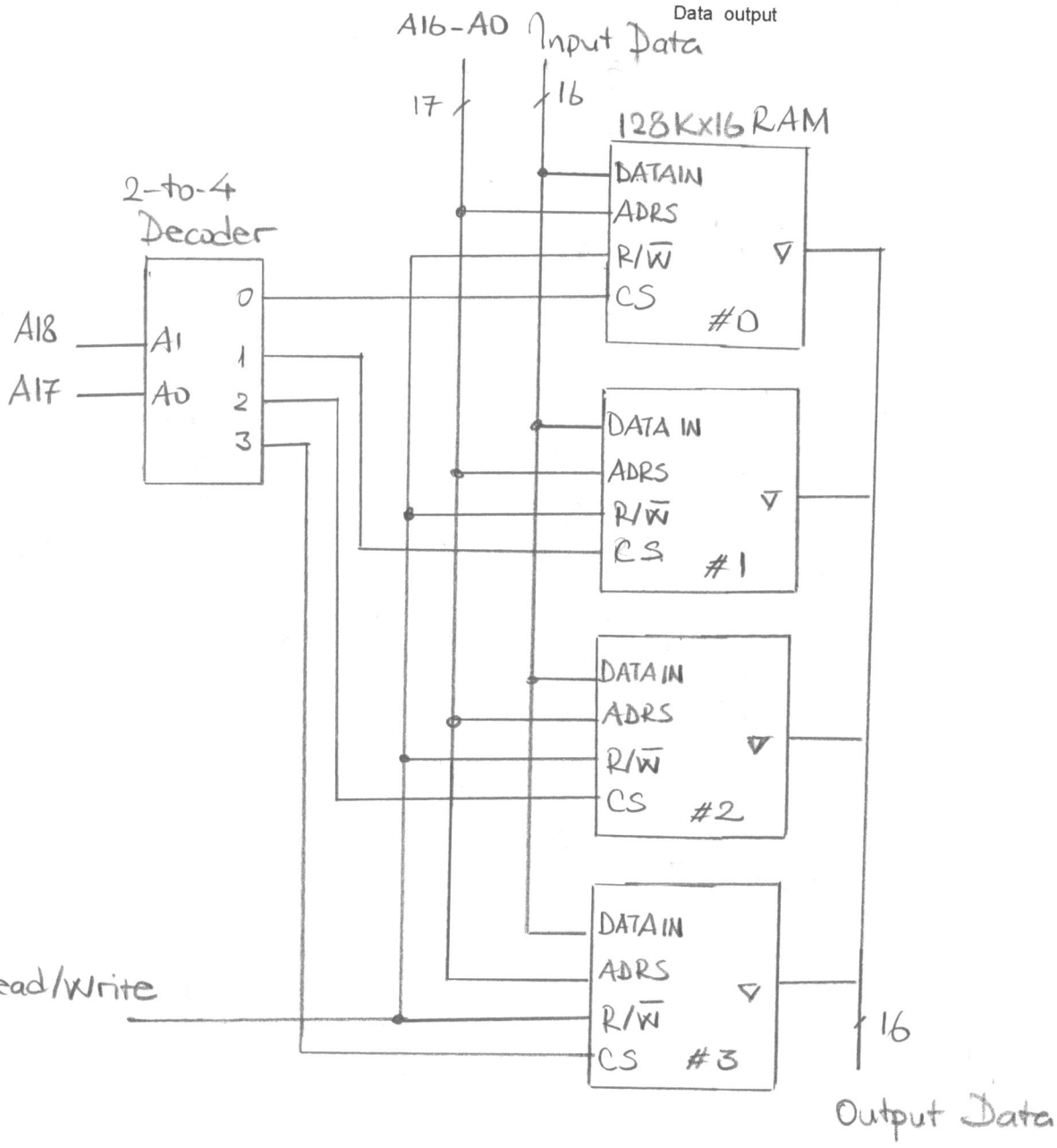
2. Using four 128Kx16 RAM chips and a decoder, construct the block diagram for a 512Kx16 RAM. Find the address range of each RAM. [20pts]



$512K = 2^9 \cdot 2^{10} = 2^{19}$

19 address lines.

$128K = 2^7 \cdot 2^{10} = 2^{17}$



Address Lines

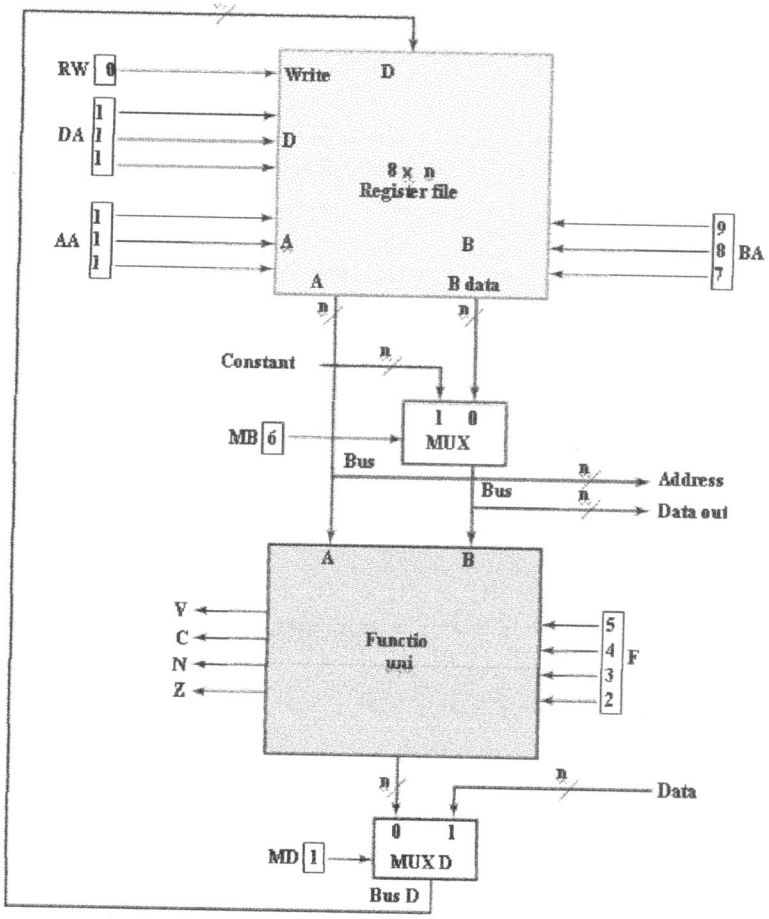
18 17	16 15 ... 2 1 0		
0 0	0 0 ... 0 0 0	00000	RAM #0
0 0	1 1 ... 1 1 1	1FFFF	
0 1	0 0 ... 0 0 0	20000	RAM #1
0 1	1 1 ... 1 1 1	3FFFF	
1 0	0 0 ... 0 0 0	40000	RAM #2
1 0	1 0 ... 1 1 1	5FFFF	
1 1	0 0 ... 0 0 0	60000	RAM #3
1 1	1 1 ... 1 1 1	7FFFF	

Selects one of four 128K×16 RAM

17 address lines for each 128K×16 RAM

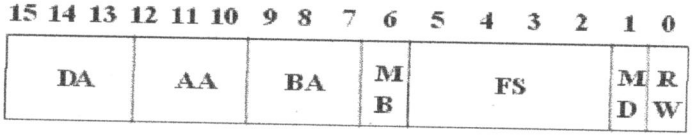
3. Specify the 16-bit control word that must be applied to the datapath of Fig. 2 to implement the following microoperations: [20 pts]

- R5←sr R6
- R1←R2+R3
- R1←R7+1
- R2←0
- R2←Data In



FS(3:0)	Microoperation
0000	$F \leftarrow A$
0001	$F \leftarrow A + 1$
0010	$F \leftarrow A + B$
0011	$F \leftarrow A + B + 1$
0100	$F \leftarrow A + \overline{B}$
0101	$F \leftarrow A + \overline{B} + 1$
0110	$F \leftarrow A - 1$
0111	$F \leftarrow A$
1000	$F \leftarrow A \wedge B$
1001	$F \leftarrow A \vee B$
1010	$F \leftarrow A \oplus B$
1011	$F \leftarrow \overline{A}$
1100	$F \leftarrow B$
1101	$F \leftarrow$ sr B
1110	$F \leftarrow$ sl B

(a) Block diagram



(b) Control word

Figure 2 Datapath with control variables

R5 ← sr R6

101 000 110 0 1101 01 = A335 hex

R1 ← R2+R3

Name:
Student ID:

R1 ← R7+1

001 111 000 0 0001 0 1 = 3C05 hex

R2 ← 0 or R2 ← R3 ⊕ R3

010 011 011 0 1010 01 = 4DA9 hex

R2 ← Data In

010 000 000 0 0000 11 = 4003 hex

Single-Cycle Computer

4. Write an assembly-language program using the instructions of the single-cycle computer that performs the arithmetic operation $2 \cdot N_1 - N_2 + 6$. The operands N_1 and N_2 are stored in the memory locations 20 and 21(decimal) respectively. Assume that register R2 contains the value of 20(decimal). The result is to be placed in 25(decimal). [20pts]

Give a comment for every instruction in the assembly program.

INC R1 Increment R1 or R1←R1+1

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]	

R2 : pointer register that holds the address of the operand N1.

```
LD  R3,R2     Load R3 with operand N1
              in memory
SHL R3,R3     Multiply contents of R3 by 2
INC R2,R2     Increment R2 (R2=21)
LD  R4,R2     Load R4 with operand N2
SUB R3,R3,R4  Subtract R4 from R3
ADI R3,R3,6   Add 6 to R3
ADI R2,R2,4   Add 4 to R2 (R2=25)
ST  R2,R3     Store R3 in memory location 25.
```