NAZARBAYEV
UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

NAME. . . . .

EXAMINATION RULES

- The duration of this exam is 120 minutes.
- The exam consists of 14 pages including this one. Write your name to each page.
- The total points is 115, points above 100 are bonus.
- Students are required to follow all instructions given by the examiners.
- Talking is NOT allowed under any circumstances.
- Students MAY NOT bring any written or printed materials into the examination room except where explicitly allowed by the examiner.
- Mobile phones are strictly prohibited in the examination room.
- Students MAY NOT bring any electronic device into the examination room except where explicitly allowed by the examiner (e.g., calculators with specified capabilities).
- Students may raise their hand to ask the examiner a question. The examiner may decide not to answer the question: students are expected to know the requisite terminology and understand the examination questions.
- For examinations lasting two hours or less, students are NOT allowed to leave the examination room until ready to turn in their work.
- Once a student has seen the examination paper, the student is assumed to be in good health at the time of the examination.

I have read and understood the examination rules. I will not cheat, copy from other students, or use unauthorized materials or devices, and I have not brought such materials or devices into the examination room.

Signed: . . . .

Student Name:

2 May 2014

ROBT206 Microcontrollers with Lab – Final Exam

Please provide precise answers to all questions. You can use the back side of the sheet in order to answer the questions.

1. Fill the table for numbers in different bases: **(10 points)**

Decimal	Binary	Hexadecimal
21	0b10101	0x15
54	0b110110	0x36
210	0b11010010	0xD2
14.375	0b1110.011	0x E.6

2. Perform subtraction operation using 2s complement **(5 points)**

10101011 - 10111101

2s complement = 01000011

```
 10101011
 01000011
---------
 11101110
```

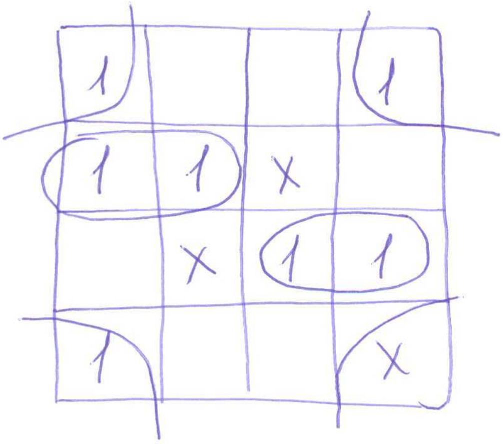
→ 2s comp. = 00010010

Result = - 010010

3. Optimize the following Boolean function ***F*** with don't-care conditions ***d***. **(5 points)**

$F(A,B,C,D)=\sum m(0,2,4,5,8,14,15)+d$ (A,B,C,D), d(A,B,C,D) = $\sum m(7,10.13)$

Find all prime implicants and apply the selection rule.



1			1
1	1	X	
	X	1	1
1			X

$F = \bar{B}\bar{D} + \bar{A}B\bar{C} + ABC$

↑

Essential Prime Impl.

Using Selection Rules we select only non overlapped prime implicants, which are:

$\bar{A}B\bar{C} + ABC$

4. (20 points total)

The truth table for a combinational circuit with three inputs (A,B,C) and one output (Z) is below.

Y is an external signal.

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	Y
0	1	1	1
1	0	0	Y
1	0	1	0
1	1	0	0
1	1	1	1

$Z = C$

$Z = Y \cdot \bar{C} + C = Y + C$

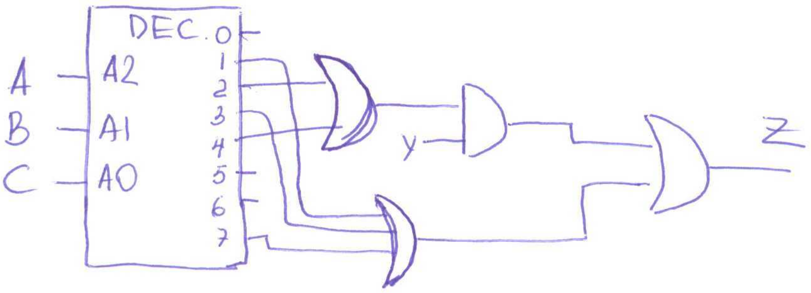
$Z = Y \cdot \bar{C}$

$Z = C$

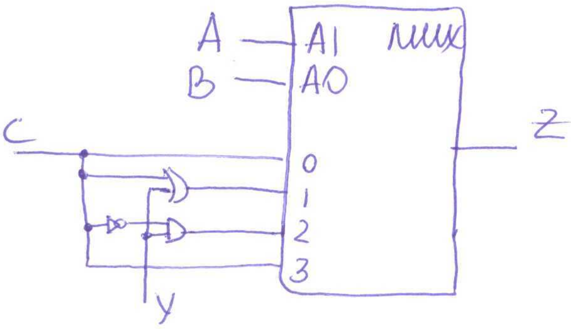
a) Implement the combinational circuit using a 3-to-8 decoder and additional gates. **(10 points)**

b) Implement the combinational circuit using a 4-to-1 line multiplexer. **(10 points)**

a)



b)



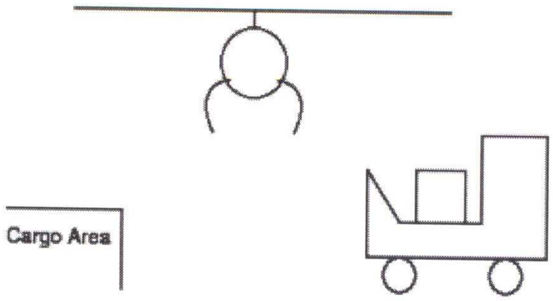
5. (**15 points**) A cargo crane: Create a controller for a crane that offloads cargo from trucks onto a cargo area. It has inputs S1 and S0 and a sensor S2. When S1S0=00 the crane goes down, if S1S0=11 the crane goes up. For S1S0=01, and S1S0=10 the crane goes right and left respectively.

There are additional rules.

- The crane always goes up after going down.
- It cannot go down if the cargo area is full, which is represented by the sensor (S2) having a value of '1'. If the crane is going left or right and the new command is to go down, the crane instead goes up.

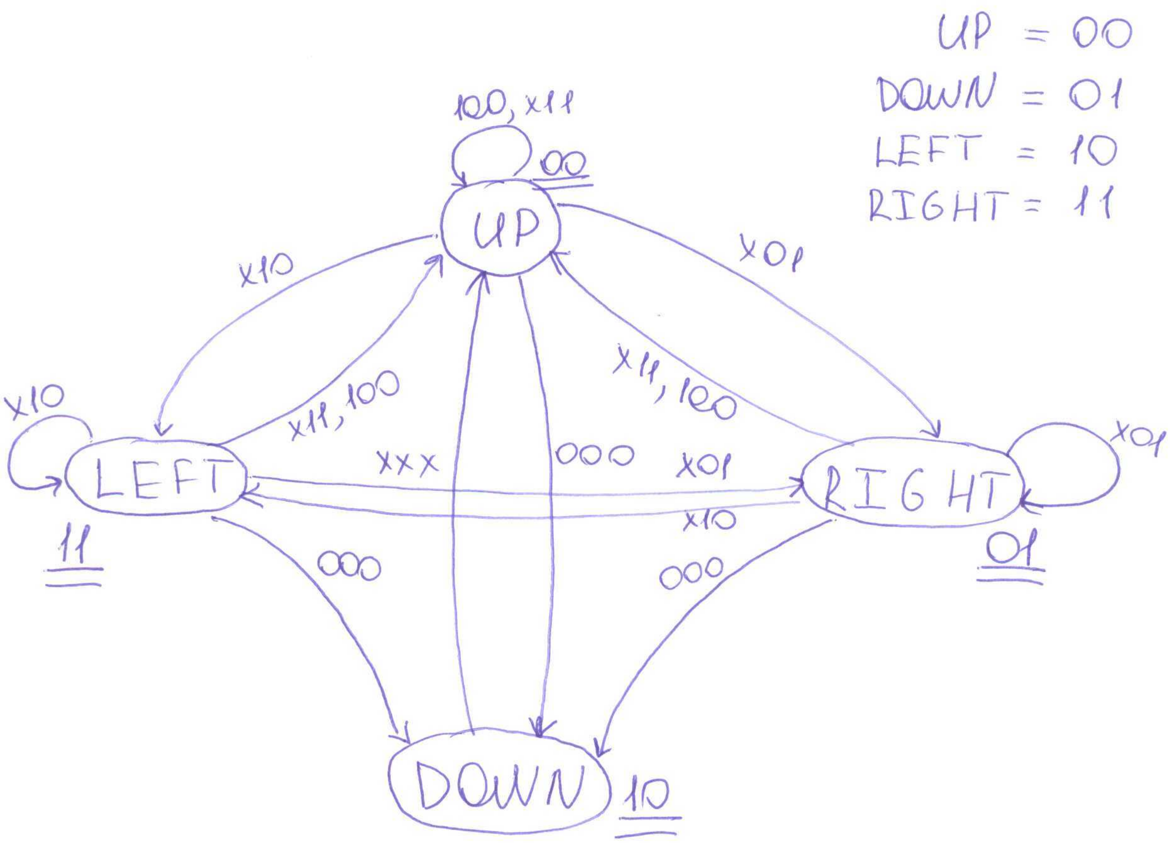
Do not worry about the outputs, only the state

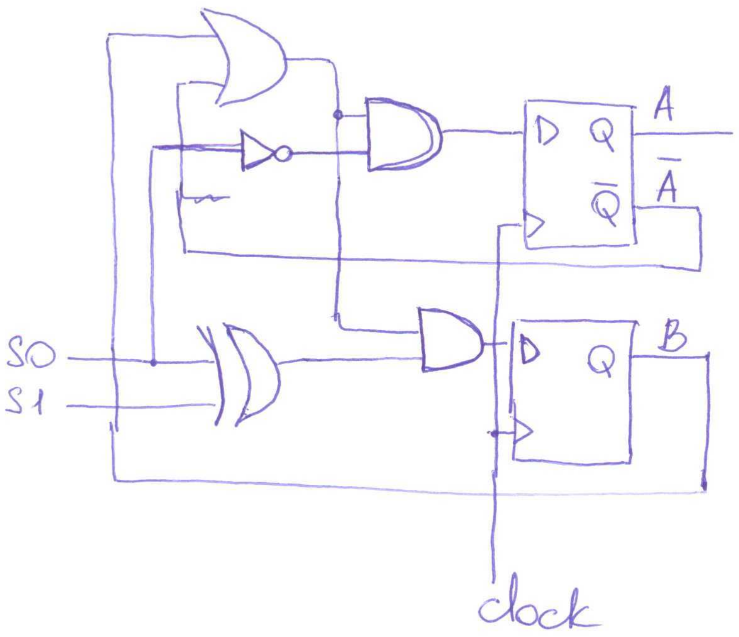
a)



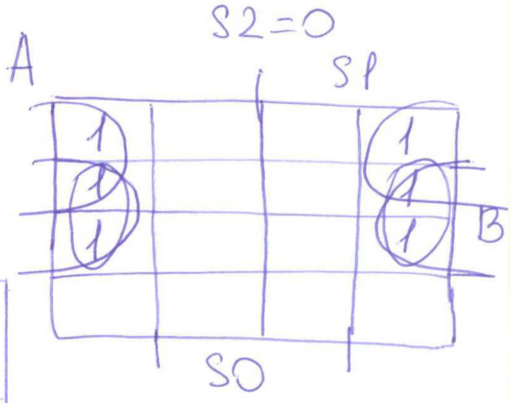
Find the state diagram and state table of the crane controller. Use UP, DOWN, LEFT and RIGHT as states, and combinations S2S1S0 as inputs **(5 point)**

b) Design the circuit using two D flip flops **(10 point)**





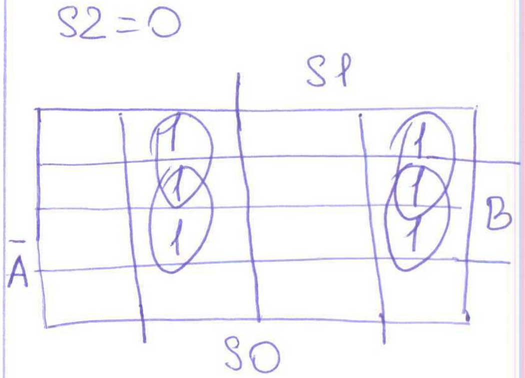
A	B	S2	S1	S0	A(t+1)	B(t+1)
0	0	X	0	1	0	1
0	0	0	0	0	1	0
0	0	X	1	0	1	1
0	0	1	0	0	0	0
0	0	X	1	1	0	0
0	1	X	0	1	0	1
0	1	X	1	1	0	0
0	1	1	0	0	0	0
0	1	0	0	0	1	0
0	1	X	1	0	1	1
Φ	0	X	X	X	0	0
1	1	X	1	0	1	1
1	1	X	1	1	0	0
1	1	1	0	0	0	0
1	1	0	0	0	1	0
1	1	X	0	1	0	1



1			1
1			1
1			1

$$A(t+1) = \bar{A}\,\overline{S0} + B\,\overline{S0}$$

$$= (A+B)\,\overline{S0}$$



	1		1
	1		1
	1		1

$$B(t+1) = \bar{A}\,\overline{S1}\,S0 + \bar{A}\,S1\,\overline{S0} +$$

$$+ B\,\overline{S1}\,S0 + B\,S1\,\overline{S0} =$$

$$= (\bar{A}+B)(\overline{S1}\,S0 + S1\,\overline{S0}) =$$

$$= (\bar{A}+B)\,(S1 \oplus S0)$$

6. Using D flip flops and logic gates design a binary counter to repeat sequence of states 1, 2, 3, 6 **(15 points)**

A	B	C	DA	DB	DC
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	1	0
1	1	0	0	0	1

$DA = \bar{A}BC$

$DB = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC$

$DC = \bar{A}B\bar{C}$

DB =

			B	
	X	1	1	1
A	X	X	X	0
		C	C	

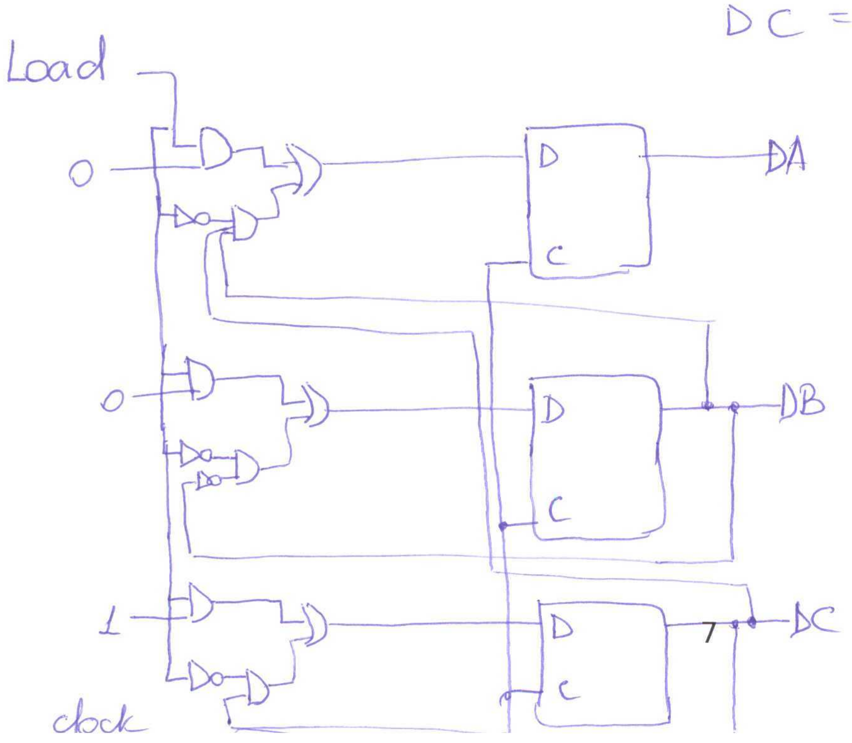
DC =

			B	
	X	0	0	1
A	X	X	X	1
		C	C	

$\therefore DA = BC$

$DB = \bar{A}$

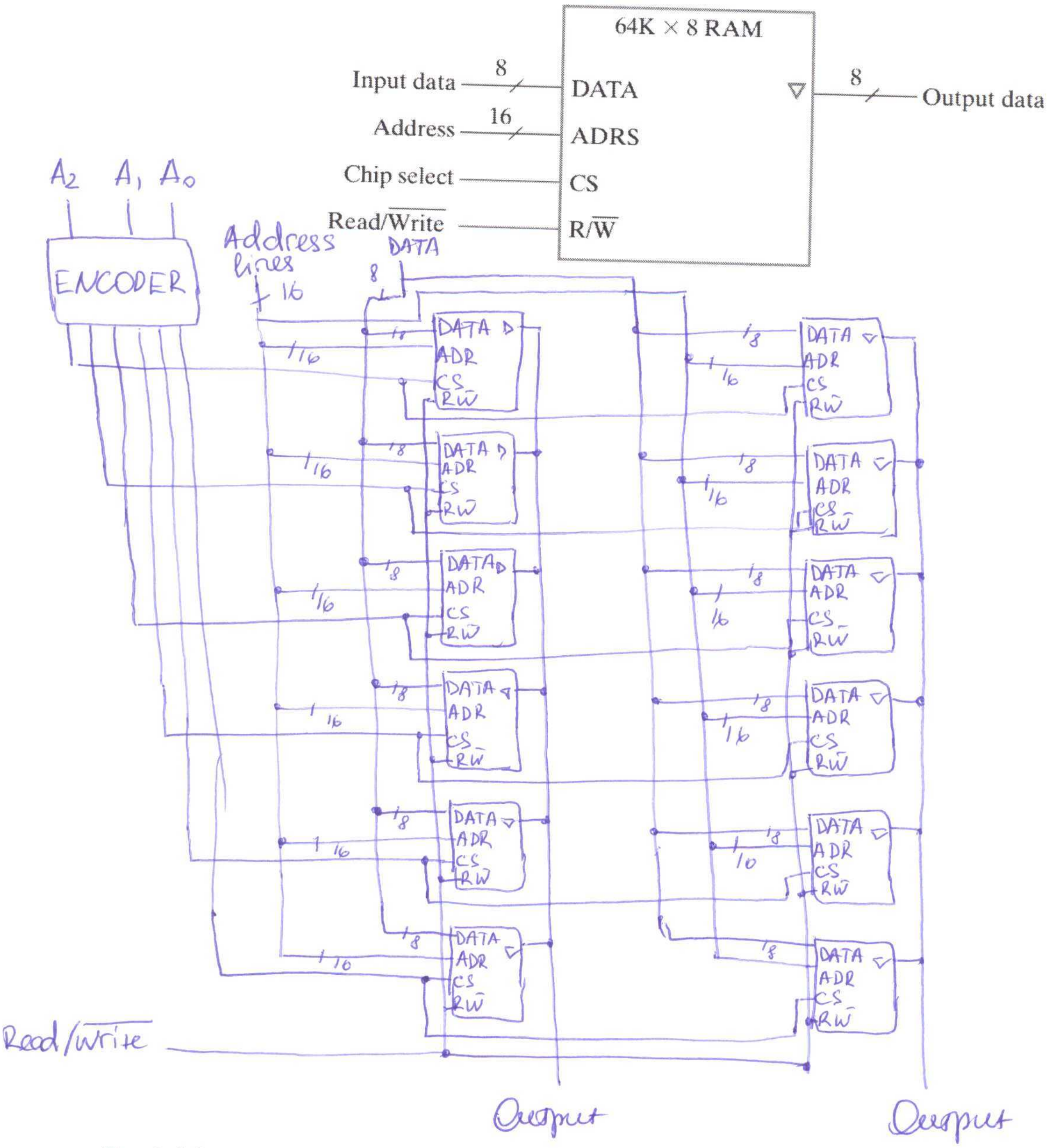
$DC = \bar{C}$



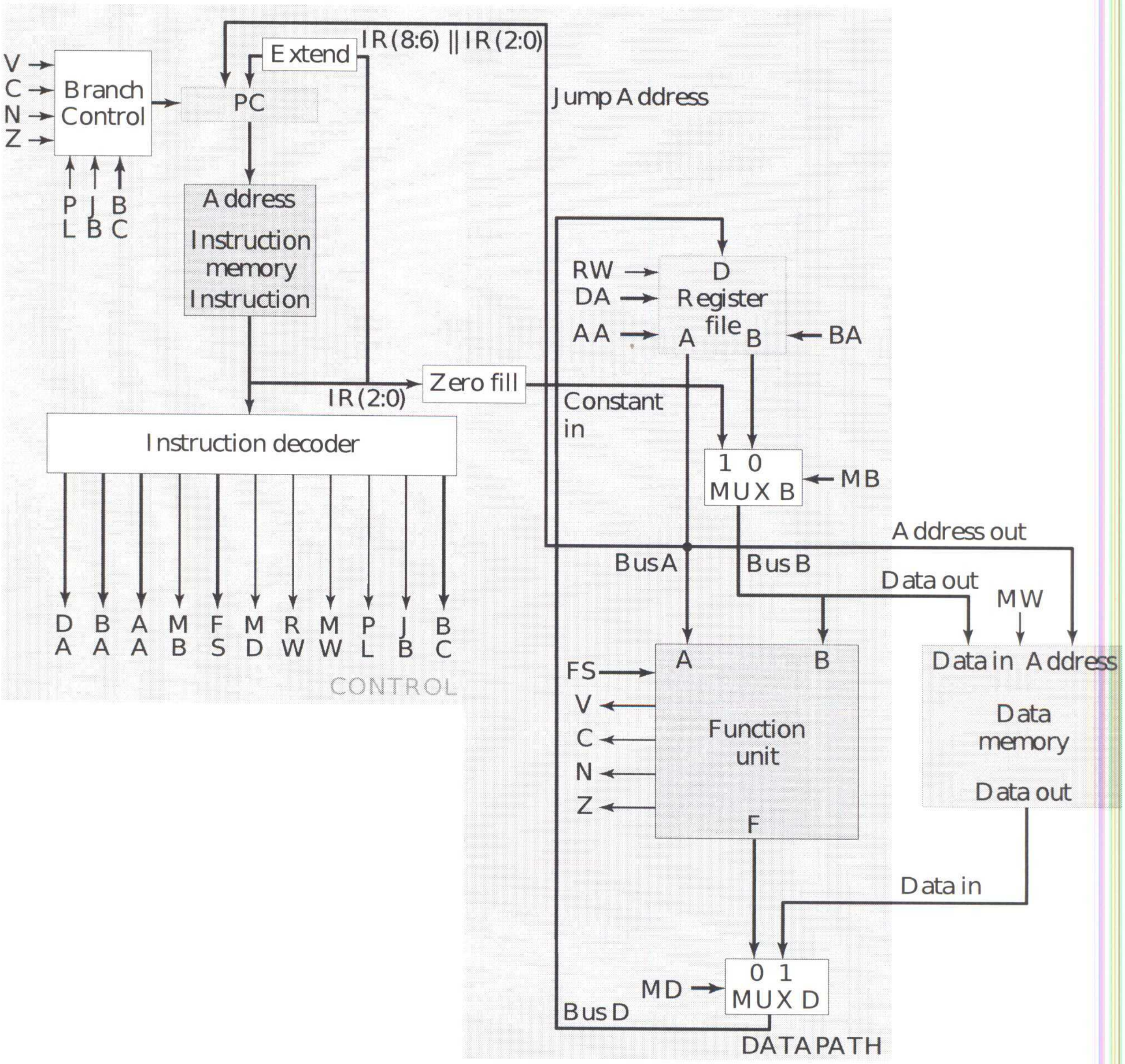
Student Name:

2 May 2014

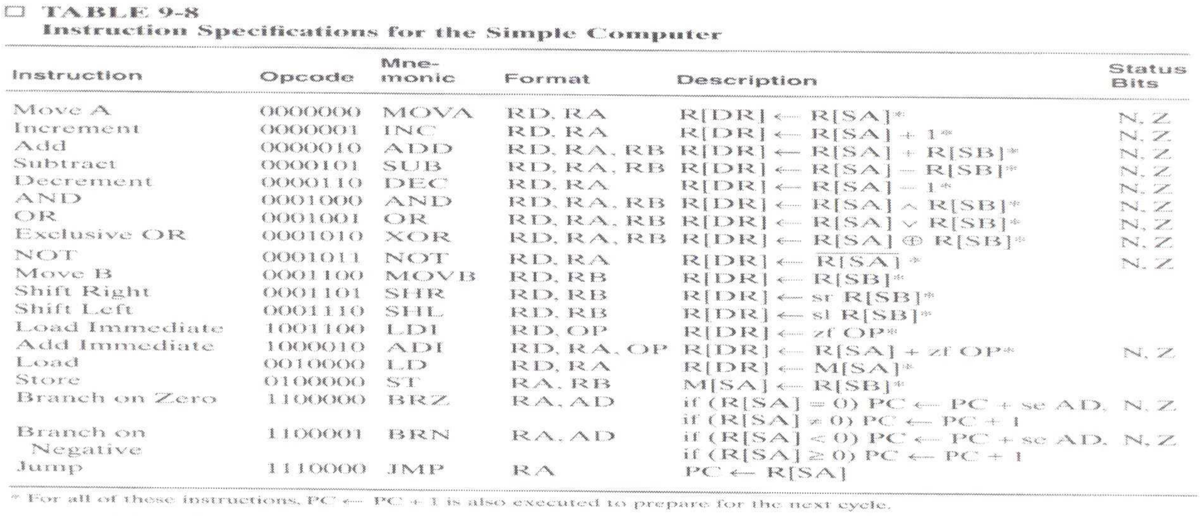
7. Using a 64K x 8 bit RAM chips as shown below design a 386K x 16-bit memory **(10 points)**



8. A block diagram of a single cycle computer with 8 registers, the instruction set and a PC control combination table are given below. Complete the table below for executing instructions
(15 points)



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



□ TABLE 9-8
Instruction Specifications for the Simple Computer

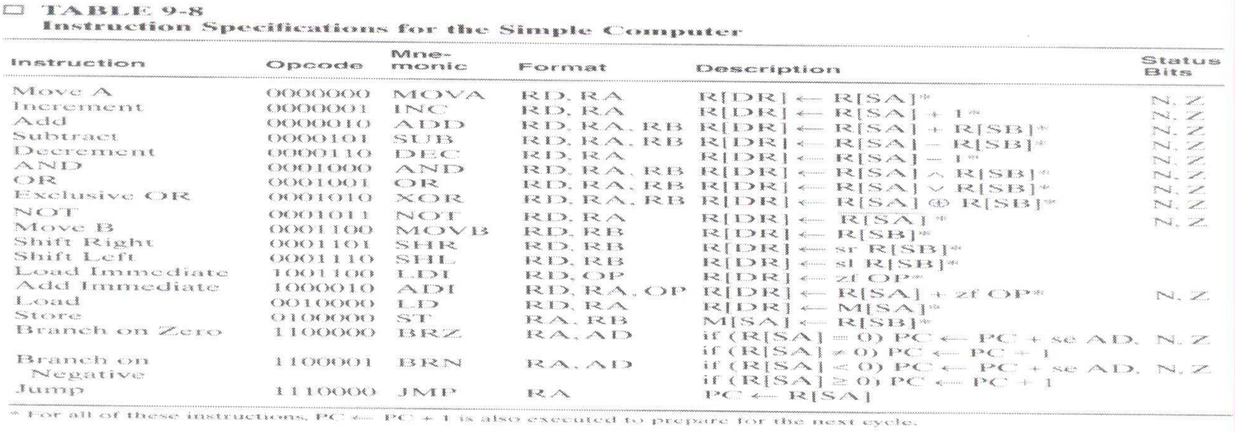
Instruction	Opcode	Mne-monic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	R[DR] ← R[SA]*	N, Z
Increment	0000001	INC	RD, RA	R[DR] ← R[SA] + 1*	N, Z
Add	0000010	ADD	RD, RA, RB	R[DR] ← R[SA] + R[SB]*	N, Z
Subtract	0000101	SUB	RD, RA, RB	R[DR] ← R[SA] − R[SB]*	N, Z
Decrement	0000110	DEC	RD, 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*	N, Z
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,
				if (R[SA] ≠ 0) PC ← PC + 1
					N, Z

Branch on Negative	1100001	BRN	RA, AD	if (R[SA] < 0) PC ← PC + se AD,
				if (R[SA] ≥ 0) PC ← PC + 1
					N, Z

Jump	1110000	JMP	RA	PC ← R[SA]	

* For all of these instructions, PC ← PC + 1 is also executed to prepare for the next cycle.



□ TABLE 9-8
Instruction Specifications for the Simple Computer

Instruction	Opcode	Mne-monic	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*	N, Z
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,
				if (R[SA] ≠ 0) PC ← PC + 1
					N, Z

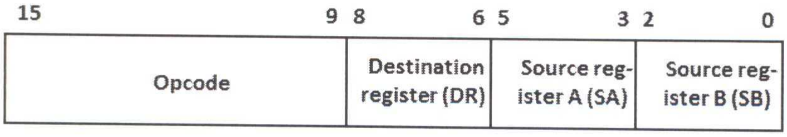
Branch on Negative	1100001	BRN	RA, AD	if (R[SA] < 0) PC ← PC + se AD,
				if (R[SA] ≥ 0) PC ← PC + 1
					N, Z

Jump	1110000	JMP	RA	PC ← R[SA]	

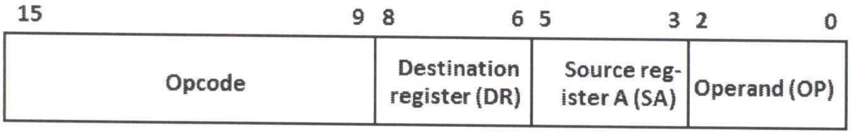
* For all of these instructions, PC ← PC + 1 is also executed to prepare for the next cycle.

Instruction-Register transfer	DA	AA	BA	MB	Opcode	MD	RW	MW	PL	JB	BC
M[R2] □ R5	xxx	010	101	0	0100000	x	0	1	0	x	x
R1□R6 - R3	001	110	011	0	0000101	0	1	0	0	x	x
if (R6<0) PC□PC + sePC else PC□PC + 1	xxx	110	xxx	x	1100001	x	0	0	1	0	1

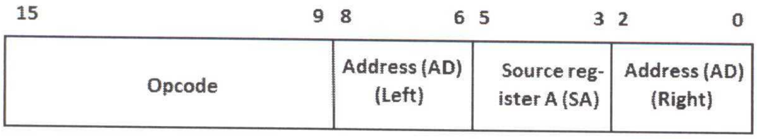
9. The format of different types of instructions for a simple computer with 8 registers each holding 16 bits is given below. The instruction set architecture is given in the previous question
(20 points).



(a) Register



(b) Immediate



(c) Jump and Branch

Two values are stored in the memory locations 31 and 32. Load them to registers R0 and R1 and perform the following operations.

1. R2 = R0 + R1
2. R3 = R2 – 5
3. R3 = R3 + 1

The final value of R3 is to be stored in the memory location 33.

Write a mnemonic program for the simple computer based on that information and fill the memory content as much as you can (assume the memory contains instructions and data). Explain your reasoning. Assume that for executing your program PC starts from value 14. Initial values of the register file registers are given on the left.

Register	Decimal Value
R0	0
R1	0
R2	0
R3	0
R4	0
R5	31
R6	0
R7	0

```
LD    R0, R5
INC   R5, R5
LD    R1, R5
ADD   R2, R0, R1
LDI   R4, #5
SUB   R3, R2, R4
INC   R3
INC   R5
ST    R5, R3
```

Decimal Memory Address	Memory Content
14	0010000 000 101 xxx
15	0000001 101 101 xxx
16	0010000 001 101 xxx
17	0000010 010 000 001
18	1001100 100 xxx 101
19	0000101 011 010 100
20	0000001 011 011 xxx
21	0000001 101 101 xxx
22	0100000 xxx 101 011
23	
24	
25	
26	
27	
28	
29	
30	
31	0000 0000 0001 0101
32	0000 0000 0001 0110
33	0000000000100111
34	