CS3350 Computer Organization
Assignment 2
Mazen Baioumy
250924925

Answer to question 1:

CS 3350
Assignment 2
Mazen Baioumy

Answers to question 1:

$\rightarrow \overline{(\bar{P}+q)\cdot r}$

$= \overline{(\bar{P} \vee q)\wedge r}$

Let $x = \bar{P} \vee q$

$\therefore \overline{x \wedge r}$ $\rightarrow$ Negation of (AND ($\wedge$)) results in (NAND ($\uparrow$))

$\equiv x \uparrow r$

$\equiv (\bar{P} \vee q) \uparrow r$ $\rightarrow$ sub in $(\bar{P} \vee q)$ for x

$\equiv \neg(P \wedge \neg q) \uparrow r$ $\rightarrow$ Reverse De Morgan's law

$\equiv (P \uparrow \neg q) \uparrow r$ $\rightarrow$ using negation of AND

$\equiv \neg q = (q \uparrow q)$

$\therefore$ Therefore final answer $\boxed{(P \uparrow (q \uparrow q)) \uparrow r}$

Answer to question 2:
To prove that NOR is functionally complete, we must make sure that is equal and can be used to prove the AND, OR, or NOT gates.

NOR Truth Table

A	B	A V B	A↓B
0	0	0	1
0	1	1	0
1	0	1	0

1	1	1	0

NOT can be rewritten using only the NOR gate as shown below:
$\neg (A \vee A) \equiv (A \downarrow A) \equiv \neg A$

A	¬A	(A v A)	(A ↓ A)
0	1	0	1
1	0	1	0

Therefore, the NOR gate can be used to prove NOT.

AND can be rewritten using only the NOR gate as shown below:
$\neg (\neg A \vee \neg B) \equiv \neg (\neg A) \wedge \neg (\neg B) \equiv A \wedge B$

To prove the formula, we need to use $(A \downarrow A) \downarrow (B \downarrow B)$

A	B	A ∧ B	A ↓ A	B ↓ B	(A ↓ A) ↓ (B ↓ B)
0	0	0	1	1	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	0	0	1

Therefore, as shown in the truth table NOR can be used to prove AND it can be used to prove it.

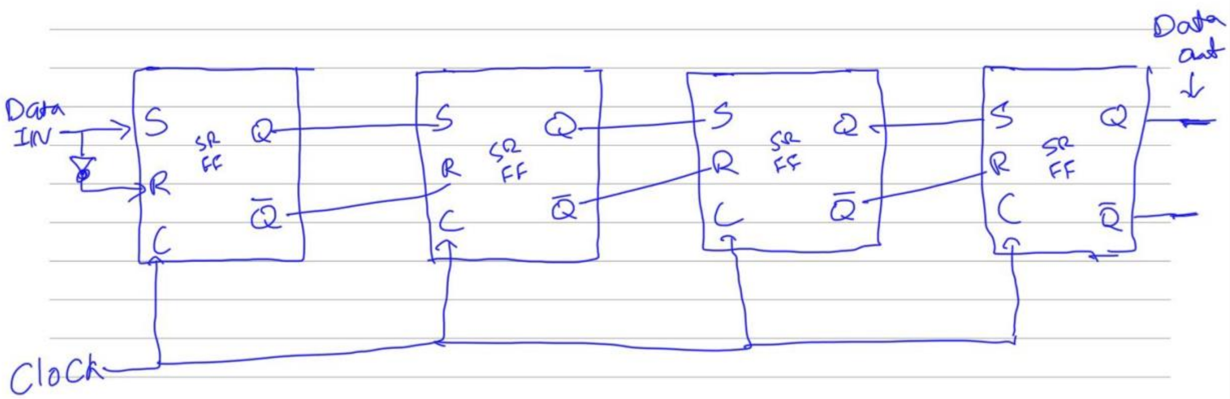
OR can be written using only NOR gate as shown below:
$(A \vee B) = (A \downarrow B) \downarrow (A \downarrow B)$

To prove we construct the truth table:

A	B	A v B	A ↓ B	A ↓ B	(A ↓ B) ↓ (A ↓ B)
0	0	0	1	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	0	1

Answer to question 3:

4-bit, Serial In, serial out, shift register:



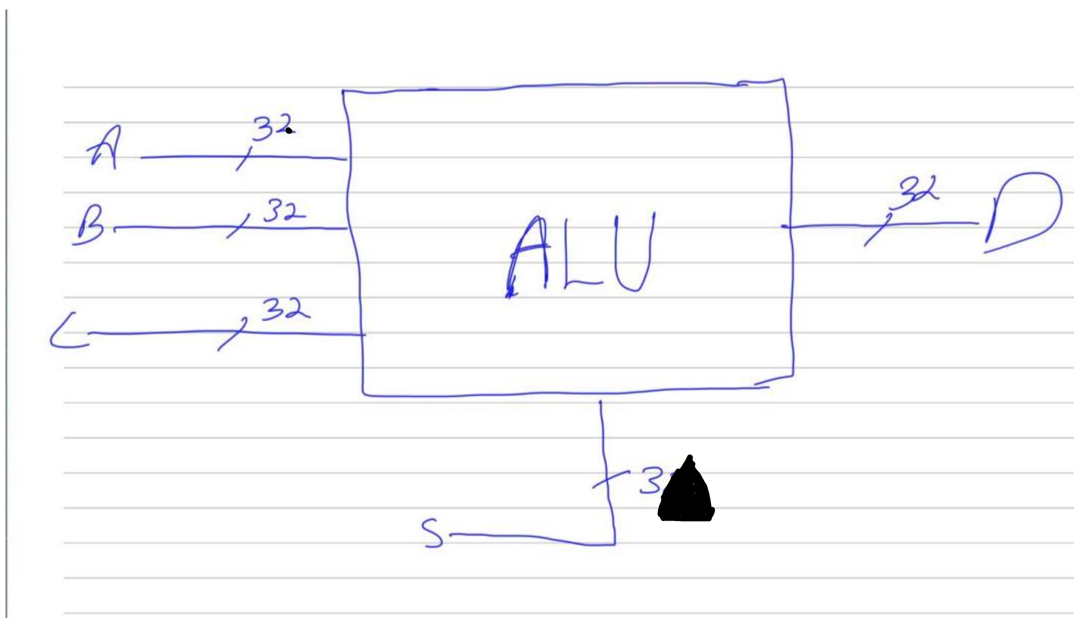
SISO Register consist of cascade of flip flops with common clock signal. The input signal is applied to first flip flop, it then passes to next flip flop on rising edge of clock signal. This process continues for as many flip-flops as there are in register. The output of last flip-flop in the chain is output of register

Answers to Question 4:

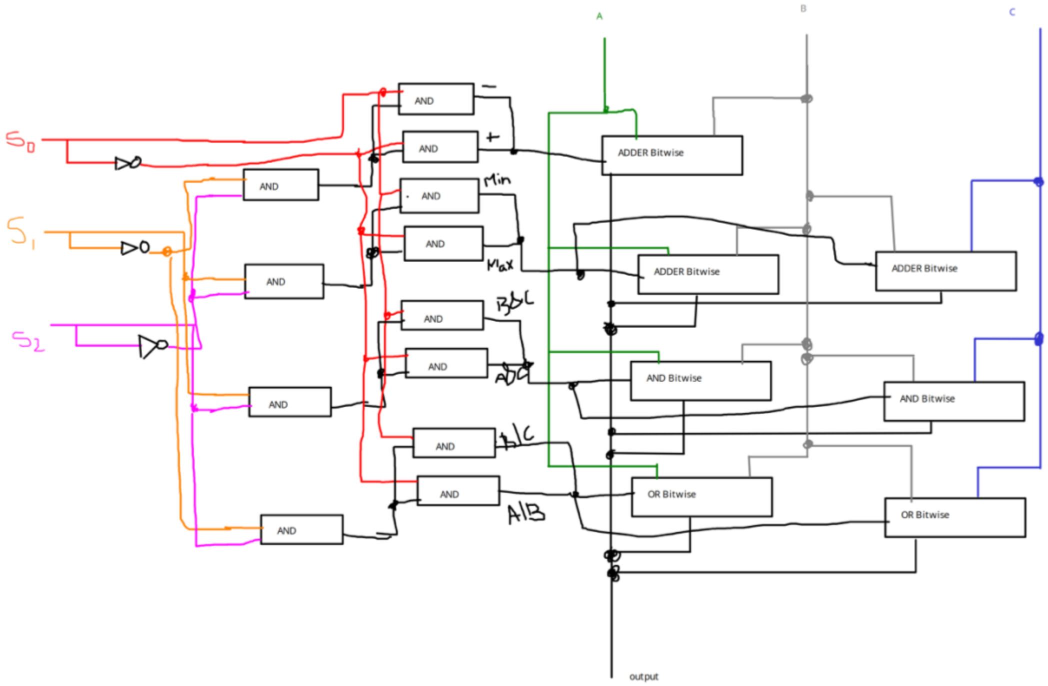
A) **i)** The ALU requires 4 control signals to select one of the 8 possible operations. We can use a 2-bit signal to select one of four possible operations on the first two inputs (A and B), and another bit signal to select one of four possible operations and the third input (C).

Control Signal	Operation
000	A + B
001	A - B
010	MAX (A, B, C)
011	MIN (A, B, C)
100	A & B
101	B & C
110	A | B
111	B | C

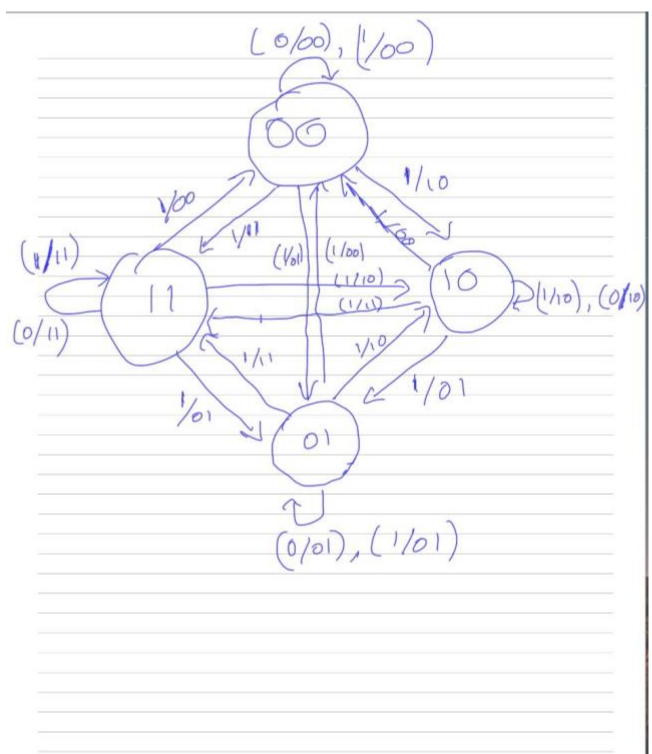
ii)



B)



Answer to Question 5:
Part A:



Part B:

X_0	X_1	(RC)	R	C	I	Output
1	0	00	0	0	1	10
1	1	01	0	1	1	11
1	0	10	1	0	1	10
1	1	11	1	1	1	11
0	0	00	0	0	1	00
0	1	01	0	1	1	01
0	0	10	1	0	1	00
0	1	11	1	1	1	01

Part C:

$$00 \equiv \overline{X_0 X_1 I} + X_0 X_1 \overline{I} \equiv (\overline{X_0 X_1} + X_0 X_1)\overline{I} \equiv \boxed{\overline{I}}$$

$$01 \equiv \overline{X_0}\,\cancel{X_1}\,\overline{I} + X_0 \overline{X_1 I} \equiv \boxed{(\overline{X_0} X_1 + X_0 \overline{X_1})\overline{I}}$$

$$10 \equiv \overline{X_0 X_1}\, I + X_0 \overline{X_1 I} \equiv \boxed{(\overline{X_0} I + X_0 \overline{I})\overline{\cancel{X}}_1}$$

$$11 \equiv \overline{X_0} X_1 I + X_0 X_1 I \equiv (\overline{X_0} + X_0) X_1 I \equiv \boxed{X_1 I}$$

Part D: