Name :
Student ID :

28.04.2009

Digital Systems II, Midterm II
2008-09 Spring Semester

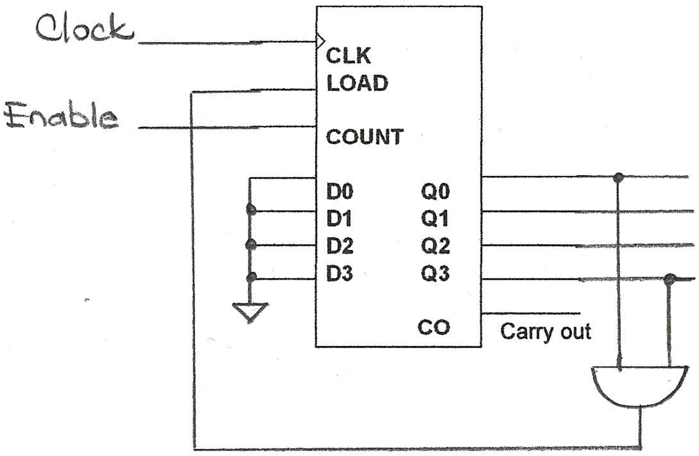
SOLUTIONS

90 minutes

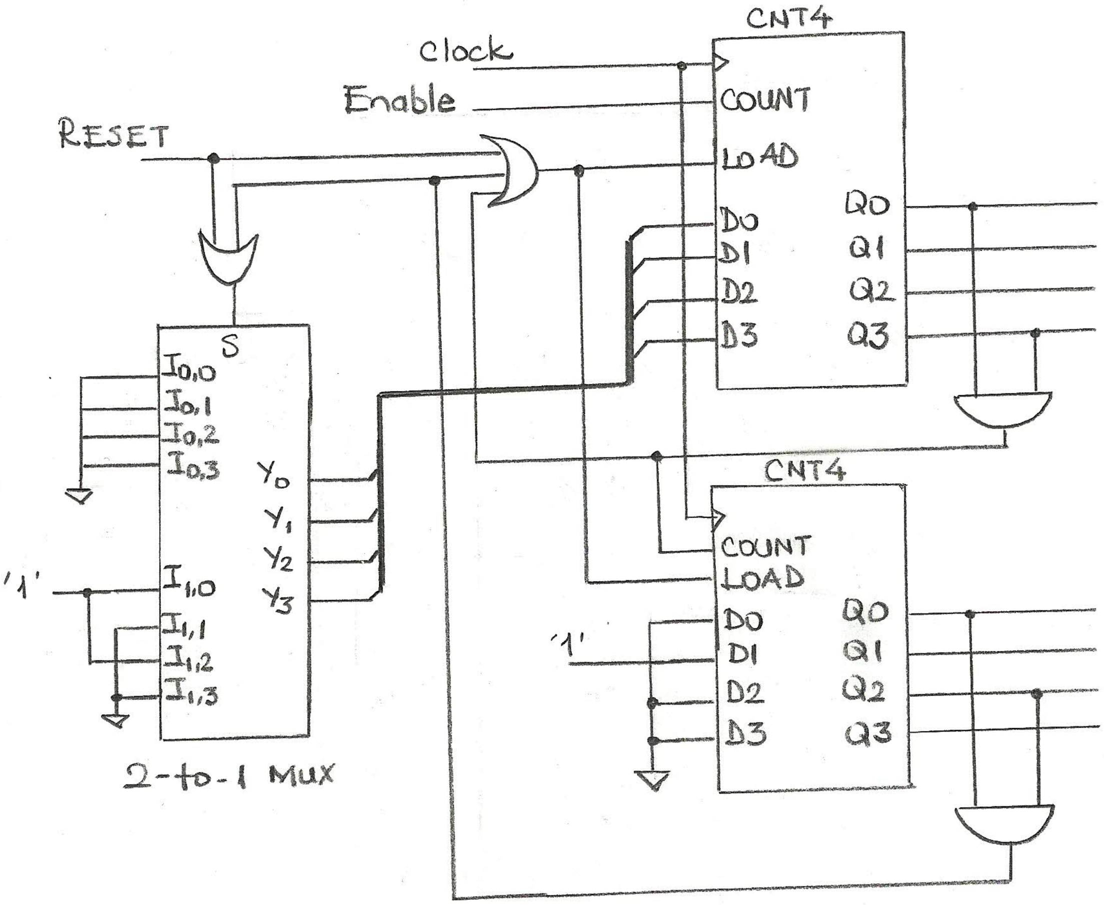
1.	
2.	
3.	
4.	

BCD Counter

1. a. Construct BCD counter using 4-bit binary counter with parallel load and external logic gate. [05pts]



b. Design two-digit BCD counter that counts from 25 to 50 decimal. Use the BCD counter that is constructed in part (a) and external gate(s). Add an additional input to the counter that initializes it synchronously to 25 decimal when the signal INIT is 1. [15pts]



Register Cell Design

2. Design a register cell for an 8-bit <u>register A</u> that has the following register transfer functions:

$$C_0 : A \leftarrow A+1 \quad \textit{Increment}$$
$$C_1 : A \leftarrow sl\ A \quad \textit{Shift left}$$

Find an optimum logic for the D input to the D flip-flop in the cell using simple approach method. <u>Use synchronous count-up binary counter to implement the register transfer A←A+1</u>. The control variables are mutually exclusive. [20 pts]

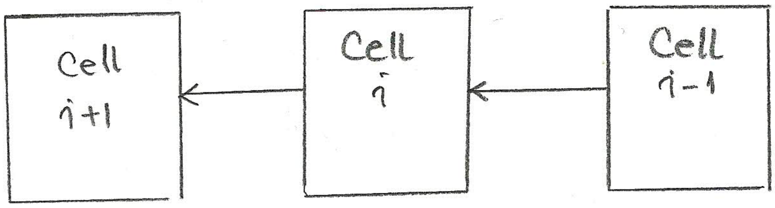
Synchronous binary counter

$Q_0 = Q_0 \oplus EN$

$Q_1 = Q_1 \oplus Q_0 EN$

$Q_2 = Q_2 \oplus Q_0 Q_1 EN$

$Q_i = Q_i \oplus Q_0 Q_1 .. Q_{i-1} . EN$



$C_0 \rightarrow INC$

$C_1 \rightarrow SHL$

$D_i = A_{i-1} . SHL + (A_i \oplus A_0 . A_1 .. A_{i-1} . INC)$

$LOAD = INC + SHL$

$D_{i,FF} = \overline{LOAD}\, A_i + LOAD . D_i$

Name :
Student ID :

Bus Transfer

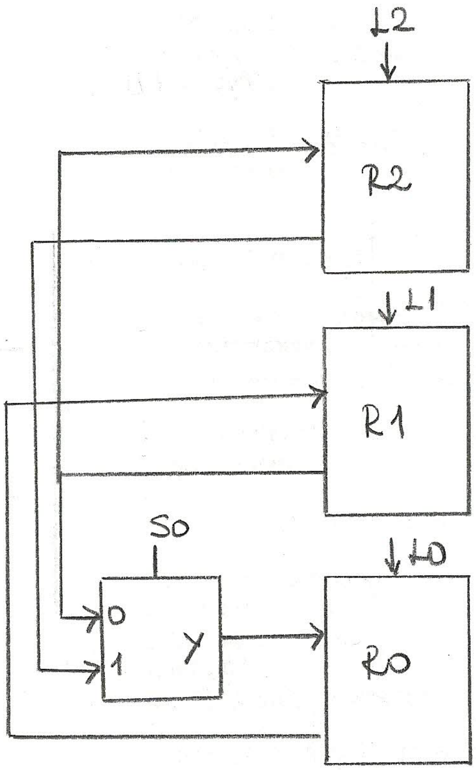
3. A system is to have the following set of register transfers, implemented using dedicated multiplexers:

$$C_0 : R1 \leftarrow R0,\ R2 \leftarrow R1$$
$$C_1 : R1 \leftarrow R0, R0 \leftarrow R1$$
$$C_2 : R1 \leftarrow R0, R0 \leftarrow R2$$

a) Using registers and dedicated multiplexers, draw the logic diagram of the system hardware that implements these register transfers. [15pts]

b) Design a simple combinational logic that generates the SELECT signals for the multiplexers and LOAD signals for the register using the control variables C_0, C_1, and C_2. [15pts]

The control variables are mutually exclusive.



Destination	Source
R0	R1, R2
R1	R0
R2	R1

Control			Select	Load		
C_0	C_1	C_2	S_0	L_0	L_1	L_2
0	0	0	X	0	0	0
1	0	0	X	0	1	1
0	1	0	0	1	1	0
0	0	1	1	1	1	0

$S_0 = C_2$

$L_0 = C_1 + C_2$

$L_1 = C_0 + C_1 + C_2$

$L_2 = C_0$

Name :
Student ID :

Sequencing and Control

4. Implement the ASM using sequence register and decoder. [30pts]

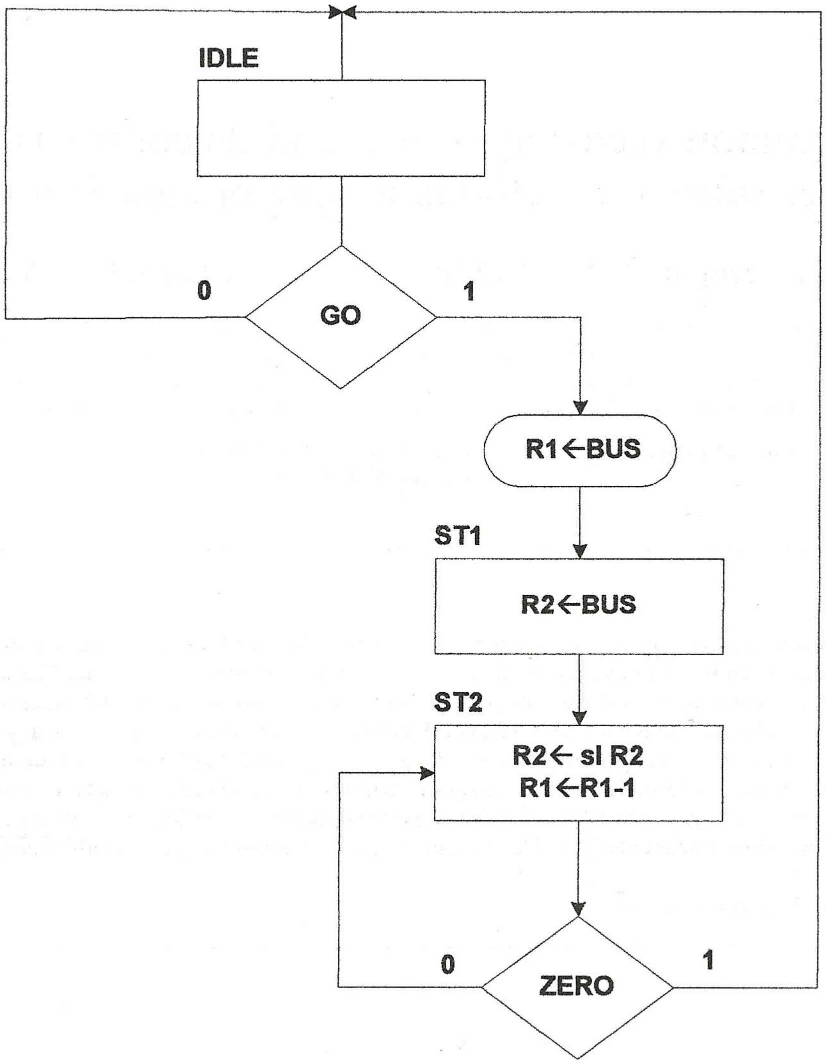


Fig.1 ASM chart

Control Signal Table

Register	Microoperation	Control Signal Name	Control Expression
R1	R1 ← BUS	RB1	IDLE . GO
	R1 ← R1-1	SUB1	ST2
R2	R2 ← BUS	RB2	ST1
	R2 ← sl R2	SHL2	ST2

State Table

	Present State		Inputs		Next State		Decoder Output		
	M1	M0	GO	ZERO	M1(t+1)	M0(t+1)	IDLE	ST1	ST2
IDLE	0	0	0	X	0	0	1		
	0	0	1	X	0	1	1		
ST1	0	1	X	X	1	0		1	
ST2	1	0	X	0	1	0			1
	1	0	X	1	0	0			1

$$D_{M1} = M1(t+1) = ST1 + ST2 \cdot \overline{ZERO}$$

$$D_{M0} = M0(t+1) = IDLE \cdot GO$$