**March 5, 2016**

# COMPUTER ENGINEERING DEPARTMENT

## COE 405

## DESIGN & MODELING OF DIGITAL SYSTEMS

**Midterm Exam**

**Second Semester (152)**

**Time: 7:30-10:00 PM**

Student Name : KEY_____

Student ID. : _____

| Question | Max Points | Score |
|---|---|---|
| Q1 | 12 | |
| Q2 | 12 | |
| Q3 | 8 | |
| Q4 | 8 | |
| Q5 | 7 | |
| Q6 | 18 | |
| Q7 | 10 | |
| Total | 75 | |

Dr. Aiman El-Maleh

**[12 Points]**

**(Q1)** Consider the function: $F(A,B,C,D) = ABCD + AB\overline{C}\,\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D}$

**(i)** Compute the expansion of $F$ using the **Orthonormal Basis** $\{\varnothing_1 = \overline{A}\,\overline{B}, \varnothing_2 = \overline{A}B, \varnothing_3 = A\overline{B}, \varnothing_4 = AB\}$.

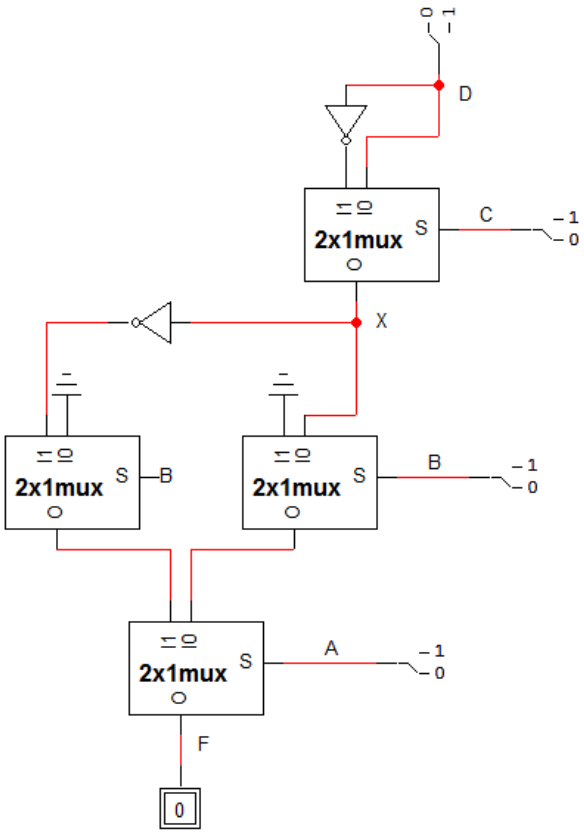
F = A'B' (C'D+CD') + A'B (0) + AB' (0) + AB (CD+C'D')

**(ii)** Compute the function $\overline{F}$ utilizing the orthonormal based expansion of the function.

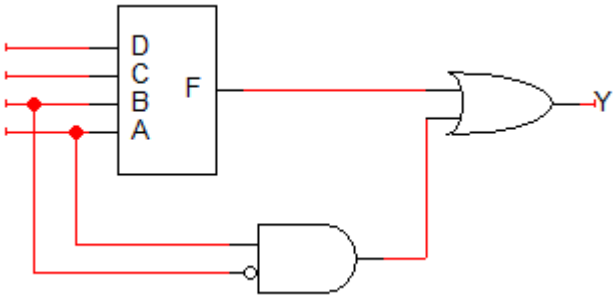
F' = A'B' (C'D'+CD) + A'B (1) + AB' (1) + AB (C'D+CD')

**(iii)** Implement the function F using <u>minimal</u> number of 2x1 MUXs and inverters.

F = A' [B' [C'D+CD']+B[0]] + A[B'[0]+B[CD+C'D']]
Let X = C'D+CD' = C'[D]+C[D']
F = A' [B' [X]+B[0]] + A[B'[0]+B[X']]



**(iv)** Suppose that the function F is part of a circuit whose output is Y as shown below. Simplify the equation of F into a minimal sum-of-products representation.




We have in the example AB' as don't care condition. We simplify F using the don't care condition as follows:

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 0 | 1 1 | 0 3 | 1 2 |
| 01 | 0 4 | 0 5 | 0 7 | 0 6 |
| 11 | 1 12 | 0 13 | 1 15 | 0 14 |
| 10 | ? 8 | ? 9 | ? 11 | ? 10 |

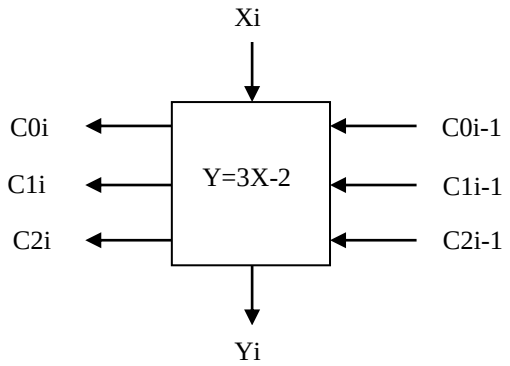
F = B'C'D+ B'CD'+ ACD+AC'D'

**[12 Points]**

**(Q2)** It is required to design an **<u>iterative</u>** combinational circuit that receives an **n-bit** signed 2's complement number X and computes the equation Y=3*X-2. The design should be based on a one-bit cell (i.e. processing one bit $X_i$) that can be copied n times to construct the n-bit design.

**(i)** Determine the inputs and outputs for a one-bit cell. Clearly explain the meaning of signal values for the interface signals between the cells.

We will model Y=3X-2 as Y=3X+(-1)+)(-1). Since -1 in 2's complement is 111…111, this implies adding 2 for every bit. Thus, we need to represent carry outs from 0 to 4. Thus, we need 3 signals to represent the carry outs in the interface between cells. The mean of the signal values is illustrated below:

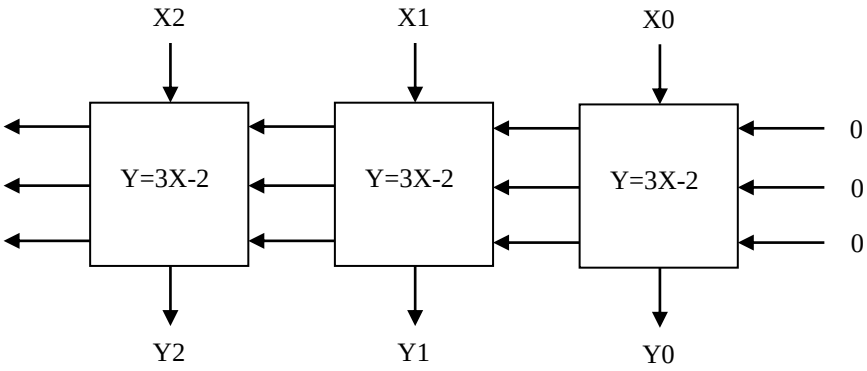



| C2 | C1 | C0 | Meaning |
|---|---|---|---|
| 0 | 0 | 0 | Carry=0 |
| 0 | 0 | 1 | Carry=1 |
| 0 | 1 | 0 | Carry=2 |
| 0 | 1 | 1 | Carry=3 |
| 1 | 0 | 0 | Carry=4 |

**(ii)** Derive the truth table for a one-bit cell.

| $C2_{i-1}$ | $C1_{i-1}$ | $C0_{i-1}$ | $X_i$ | $C2_i$ | $C1_i$ | $C0_i$ | $Y_i$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |

**(iii)** Show the block diagram for a 3-bit design.



**[8 Points]**

**(Q3)** Consider the given FSM that has 6 states, one input (X) and one output (Z), represented by the following state table:

| Present State | Next State, Z | |
|---|---|---|
| | **X=0** | **X=1** |
| S0 | S1, 0 | S2, 0 |
| S1 | S4, 0 | S3, 0 |
| S2 | S2, 0 | S5, 1 |
| S3 | S3, 0 | S2, 1 |
| S4 | S0, 0 | S5, 0 |
| S5 | S3, 0 | S5, 1 |

**(i)** Determine the equivalent states.

| | | | | | |
|---|---|---|---|---|---|
| S1 | (1,4), (2,3) | | | | |
| S2 | | | | | |
| S3 | | | (2,5) | | |
| S4 | (0,1), (2,5) | (0,4), (3,5) | | | |
| S5 | | | (2,3) | (2,5) | |
| | S0 | S1 | S2 | S3 | S4 |

Equivalent state pairs: (S0,S1), (S0, S4), (S1, S4), (S2, S3), (S2,S5), (S3, S5)
Thus, equivalent states are: (S0, S1, S4) and (S2, S3, S5)

**(ii)** Reduce the state table into the minimum number of states and show the reduced state table.

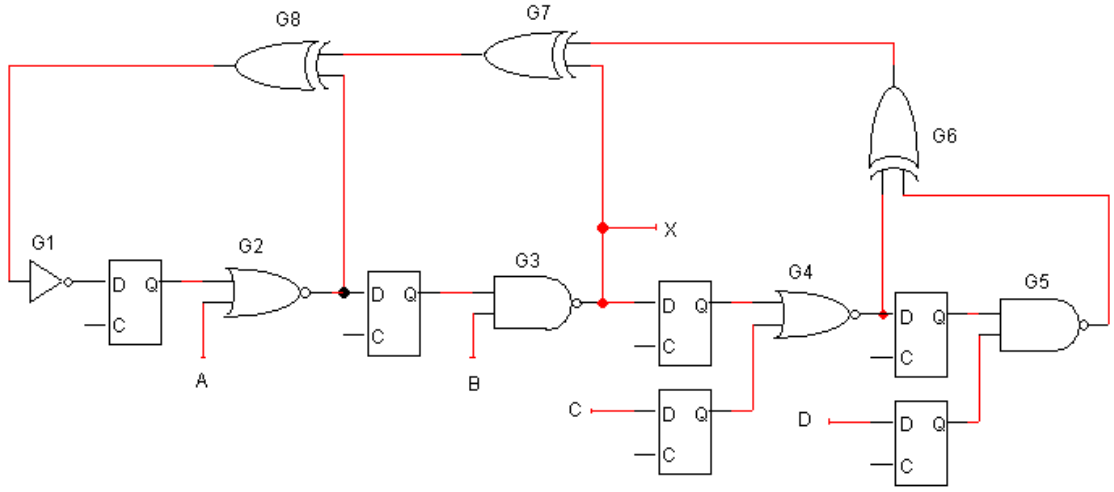
Reduced State Table:

We will rename the equivalent states (S0, S1, S4) as S0' and the equivalent states (S2, S3, S5) as S1'.

| Present State | Next State, Z | |
|---|---|---|
| | **X=0** | **X=1** |
| S0' | S0', 0 | S1', 0 |
| S1' | S1', 0 | S1', 1 |

**[8 Points]**

**(Q4)** Consider the sequential circuit given below having 4 inputs {A, B, C, D} and one output {X}. Assume that the delay of an inverter is 1 unit delay, the delay of a 2-input NAND gate is 2 unit delays, the delay of a 2-input NOR gate is 2 unit delays and the delay of a 2-input XOR gate is 3 unit delays.




**(i)** Determine the critical path of this circuit and the maximum propagation delay.

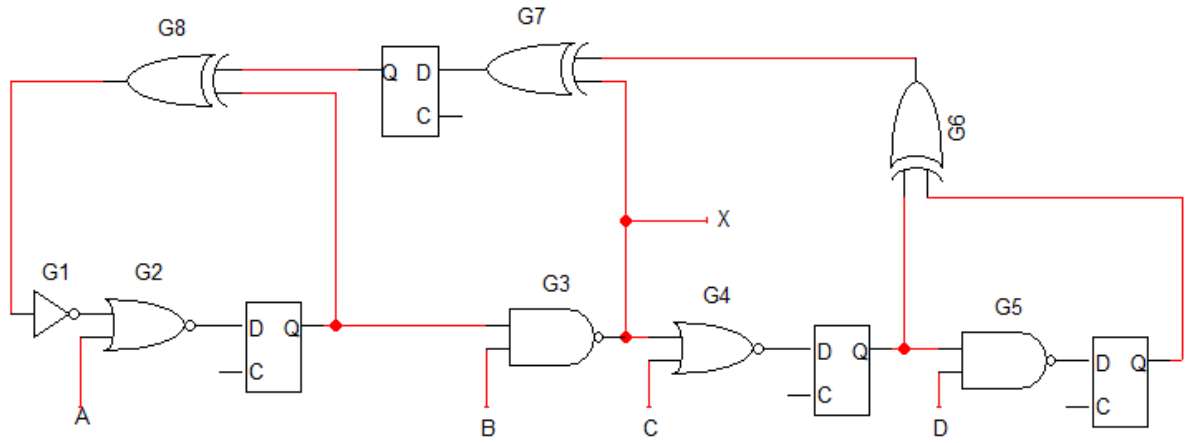
The maximum propagation delay is 12 and there are two critical paths as follows: {G5, G6, G7, G8, G1}, {G4, G6, G7, G8, G1},

**(ii)** Using only the **Retiming** transformation, minimize the critical path of this circuit with the minimum number of flip-flops possible. Determine the maximum propagation delay after retiming.

We can apply the following retiming transformations to reduce the critical path:

- Retime G5 by -1 (forward retiming)
- Retime G4 by -1 (forward retiming)
- Retime G1 by +1 (backward retiming)
- Retime G8 by +1 (backward retiming)
- Retime the stem on fanout of G2 by +1 (backward retiming)

The resulting retiming circuit is as follows which has a maximum propagation delay of 6 with only 4 FFs.



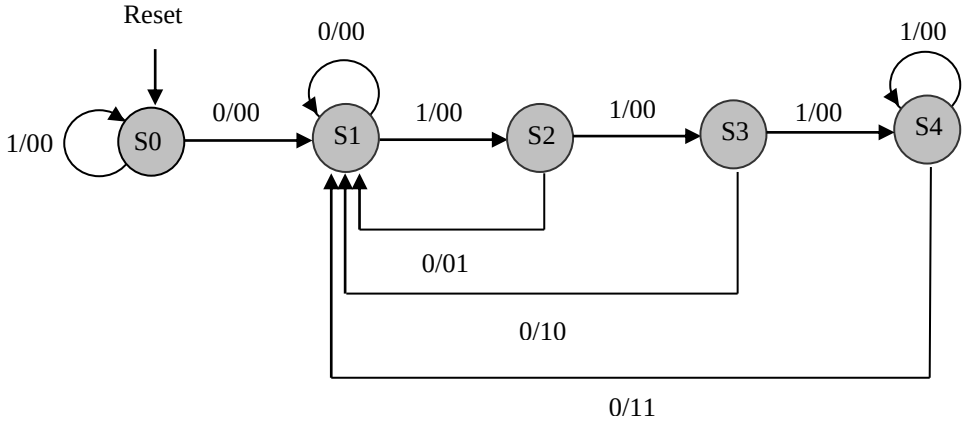

**[7 Points]**

**(Q5)** It is required to design a synchronous sequential circuit that receives a stream of data serially through input **X** and produces output values through the two outputs Z1Z0. The circuit produces an output value of 01 if it receives a 0-1-0 pulse of one cycle length, produces an output value of 10 if it receives a 0-1-0 pulse of two cycles length and produces and output value of 11 if it receives a 0-1-0 pulse of length of three cycles or more. Otherwise, the circuit produces an output value of 00. Assume the existence of an asynchronous reset input to reset the machine to a reset state. Draw the state diagram of the circuit assuming a **Mealy** model with **minimum** number of states. *You are not required to derive the equations and the circuit*. The following is an example of input and output stream:

Example:

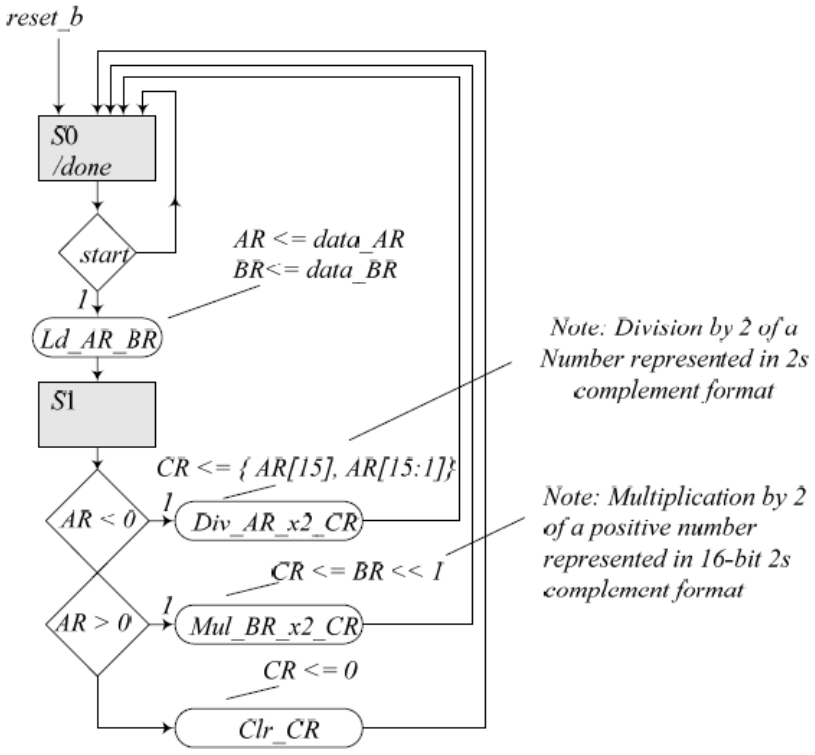
| | | |
|---|---|---|
| Input | **X** | 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 |
| Output | **Z0** | 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 |
| | **Z1** | 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 |

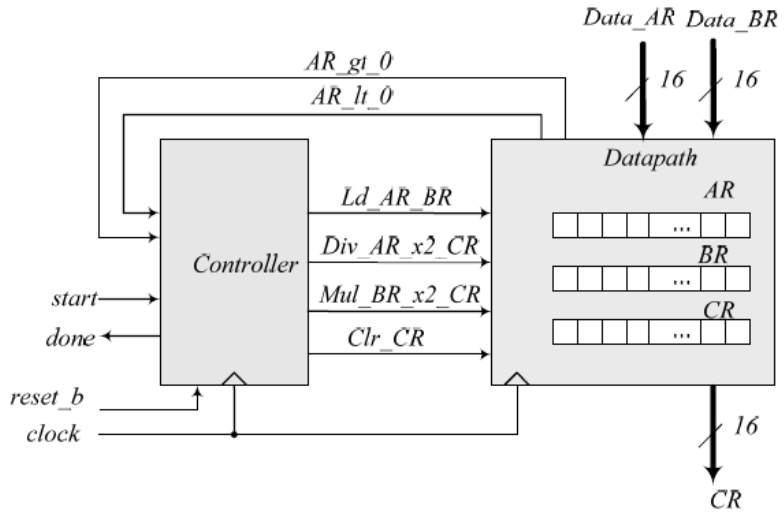
State Diagram:



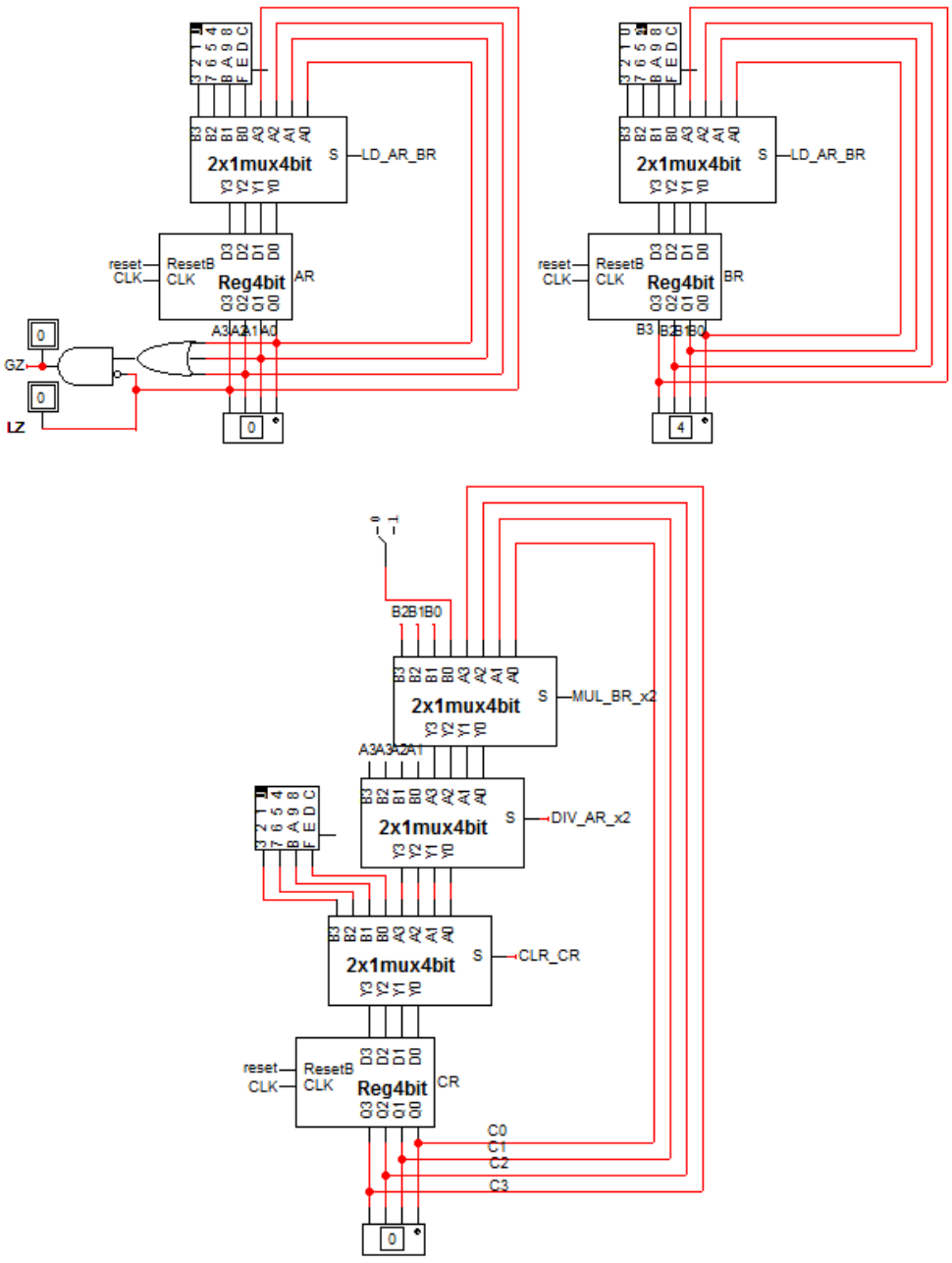
**[18 Points]**

**(Q6)** The ASMD chart and the block diagram given below show the datapath and controller for a machine that transfers two 4-bit signed numbers in 2's complement representation into registers AR and BR, divides the number in AR by 2 and transfers the result to register CR if the number in AR is negative, multiplies the number in BR by 2 and transfers the result to register CR if the number in AR is positive but non-zero, and if the number in AR is zero, clears register CR to 0.



**(i)** Show the design of the data-path unit.




**(ii)** Show the design of the control unit using the following state assignment: *S0=0, S1 =1.*

**<u>Control Unit</u>**:

| C.S. | Input | | | N.S. | Output | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | start | GZ(AR_gt_0) | LZ(AR_lt_0) | | Done | LD_AR_BR | Div_AR_x2 | Mul_AR_x2 | Clr_CR |
| S0 | 0 | x | x | S0 | 1 | 0 | 0 | 0 | 0 |
| S0 | 1 | x | x | S1 | 1 | 1 | 0 | 0 | 0 |
| S1 | x | x | 1 | S0 | 0 | 0 | 1 | 0 | 0 |
| S1 | x | 1 | 0 | S0 | 0 | 0 | 0 | 1 | 0 |
| S1 | x | 0 | 0 | S0 | 0 | 0 | 0 | 0 | 1 |

We assume the state assignment S0=0 and S1=1.

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 0 | 0 1 | ? 3 | 0 2 |
| 01 | 1 4 | 1 5 | ? 7 | 1 6 |
| 11 | 0 12 | 0 13 | ? 15 | 0 14 |
| 10 | 0 8 | 0 9 | ? 11 | 0 10 |

F+ = F' Start

Done = F'
LD_AR_BR = F' Start

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 0 | 0 1 | ? 3 | 0 2 |
| 01 | 0 4 | 0 5 | ? 7 | 0 6 |
| 11 | 0 12 | 1 13 | ? 15 | 0 14 |
| 10 | 0 8 | 1 9 | ? 11 | 0 10 |

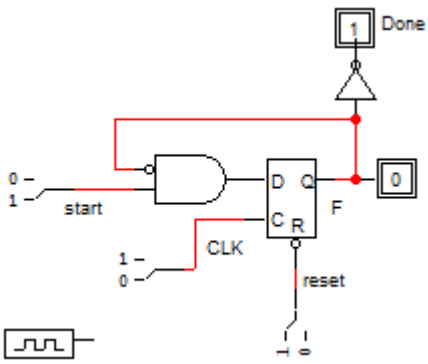
DIV_AR_x2 = F LT

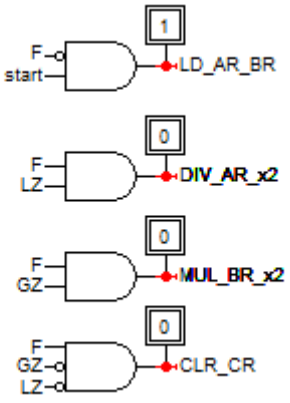
| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 0 | 0 1 | ? 3 | 0 2 |
| 01 | 0 4 | 0 5 | ? 7 | 0 6 |
| 11 | 0 12 | 0 13 | ? 15 | 1 14 |
| 10 | 0 8 | 0 9 | ? 11 | 1 10 |

MUL_BR_x2 = F GT

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 0 | 0 1 | ? 3 | 0 2 |
| 01 | 0 4 | 0 5 | ? 7 | 0 6 |
| 11 | 1 12 | 0 13 | ? 15 | 0 14 |
| 10 | 1 8 | 0 9 | ? 11 | 0 10 |

CLR_CR = F GT' LT'



**[10 Points]**

**(Q7)** It is required to design a circuit that counts the number of data transitions (i.e. 0→1and 1→0 data changes) through a stream of 128 bit data. The data is applied serially through an input *X* once the user presses a *Start* button, where the first bit is transmitted in the same cycle the *Start* button is asserted. Once the computation is finsihed the machine asserts a *Done* signal which remains asserted until the user presses the Start button again or resets the machine. Assume that the machine has Asynchronous *Reset* input. Show the ASMD chart for this machine.