Homework #4 -

Problems 2.21 2.22 to 2.23 2.24 2.26 2.33 2.34

Engineering 378

Farnam Adelkhani
915815724

2.21

2.21 Draw the hardware obtained if the following code is synthesized:

```
module  reg3 (Q1,Q2,Q3,Q4, A,CLK);
input   A;
input   CLK;
output  Q1,Q2,Q3,Q4;

reg     Q1,Q2,Q3,Q4;

always @(posedge CLK)
begin
  Q4 = Q3;
  Q3 = Q2;
  Q2 = Q1;
  Q1 = A;
end

endmodule
```

% module name reg3

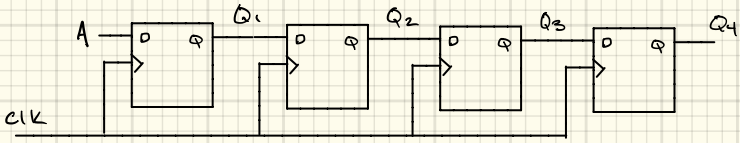
inputs: A, clk

outputs: Q_1, Q_2, Q_3, Q_4

- reg used for registers and counters, for data to be stored.
- Q_1, Q_2, Q_3, Q_4 are on Rt hand side of = ... they need to be stored till next assignment, so they are assigned to reg type.
- Always @ (posedge clk): sequential statements executed when pos edge of clock occurs.
- Blocking statements

$Q_4 = Q_3$;
$Q_3 = Q_2$; ⇒ only @ pos edge clk
$Q_2 = Q_1$;
$Q_1 = A$;

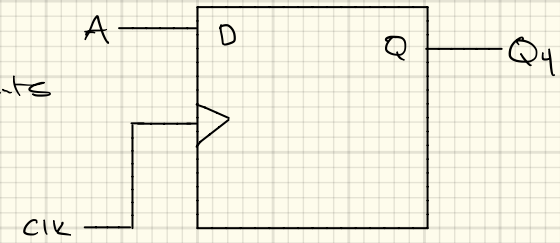
4-Bit shift register



2.22 Draw the hardware obtained if the following code is synthesized:

```
module  reg3 (Q1,Q2,Q3,Q4, A,CLK);
input   A;
input   CLK;
output  Q1,Q2,Q3,Q4;
reg     Q1,Q2,Q3,Q4;

always @(posedge CLK)
begin
  Q1 = A;
  Q2 = Q1;
  Q3 = Q2;
  Q4 = Q3;
end

endmodule
```

Blocking statements

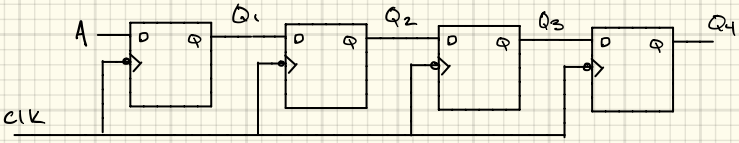


2.23 Draw the hardware obtained if the following two modules are synthesized and describe the differences.

```
module  reg3 (Q1,Q2,Q3,Q4,A,CLK);
input   A;
input   CLK;
output  Q1,Q2,Q3,Q4;

reg     Q1,Q2,Q3,Q4;

// first
always @(posedge CLK)
begin
  Q1 <= A;
  Q2 <= Q1;
  Q3 <= Q2;
  Q4 <= Q3;
end

endmodule


module  reg3 (Q1,Q2,Q3,Q4,A,CLK);
input   A;
input   CLK;
output  Q1,Q2,Q3,Q4;

reg     Q1,Q2,Q3,Q4;

// first -> second
always @(posedge CLK)
begin
  Q4 <= Q3;
  Q3 <= Q2;
  Q2 <= Q1;
  Q1 <= A;
end

endmodule
```

The sequential statements can be evaluated in any order, so the two modules are similar/identical

4-bit shift register



2.24 (a) Assume D_1=0, D_2=5, and D_1 changes to 1 at time=10ns. What are the values of D_1 and D_2 after the following code has been executed once? Do the values of D_1 and D_2 swap?

```
always @ (D1)
begin
    D2 <= D1;
    D1 <= D2;
End
```

Non-Blocking - simultaneous execution

(a) $D_1 = 0 \Rightarrow D_1 = 5 = 1$
$D_2 = 5 \quad D_2 = 1 \quad 5$

Values of D_1 & D_2 swap

(b) Assume D_1=0, D_2=5, and D_1 changes to 1 at time=10ns. What are the values of D_1 and D_2 after the following code has been executed once? Do the values of D_1 and D_2 swap?

```
always @ (D1)
begin
    D2 = D1;
    D1 = D2;
end
```

Blocking - sequential

(b) The statement is Blocking and thus, Sequential.

(c) How many latches will result when the following code is synthesized? Assume B is 3 bits long.

```
always @ (State) begin
      case(State)
      2'b00: B = 5;
      2'b01: B = 3;
      2'b10: B = 0;
      endcase
end
```

2'b11 not included

Circle the correct choice:

i. 1 latch because 1 case is missing
ii. 2 latches because state is 2 bits
iii. 3 latches because B is 3 bits
iv. 5 latches because B is 3 bits and state is 2 bits
v. None of the above. But it results in _____ latches.

(c) output is 3-bit wide (0-000, 5-101, 3-011)
8 latches

2.26 For the following Verilog code, assume that D changes to 1 at time 5 ns. Give the values of A, B, C, D, E, and F each time a change occurs. That is, give the values at time 5 ns, 5 + Δ, 5 + 2Δ, and so forth. Carry this out until 20 steps have occurred, until no further change occurs, or until a repetitive pattern emerges.

```
module prob(D);
inout D;
wire A, C;
reg B,E,F,temp_D;

initial
begin
    B = 1'b0;
    E = 1'b0;
    F = 1'b0;
    temp_D = 1'b0;
end

assign C = A;
assign A = (B & !E) | D;
assign D = temp_D;

always @(A)
begin
    B = A;
end

always
begin
    wait(A)
    E <= #5 B;
    temp_D <= 1'b0;
    F <= E;
end
endmodule
```

NS + Δ	A	B	C	D	E	F
0 0	0	0	0	0	0	0
5 0	0	0	0	1	0	0
5 1	1	0	0	1	0	0
5 2	1	1	1	1	0	0
6 0	1	1	1	0	0	0
10 0	1	1	1	0	1	0
10 1	0	1	1	0	1	1
10 2	0	0	0	0	1	1
11 0	0	0	0	0	1	1
15 0	0	0	0	0	0	1
15 1	0	0	0	0	0	0

2.33 If A = 101, B = 011, and C = 010, what are the values of the following statements? Assume A, B, and C are of reg type. Assume as many bits as necessary for the result.

(a) `{A,B} | {B,C}`
(b) `A >> 2`

(c) `A >>> 2`
(d) `{A,(~B)} == 111110`
(e) `A | B & C`

(a) concatenation $\{A,B\} = 101011$ $\{B,C\} = 011010$

OR operator is performed

111011

(b) $A >> 2 \rightarrow 001$

(c) $A >>> 2 \rightarrow 111$

(d) $\{A,(\sim B)\} == 111110$

$(\sim B) = 100$ ... $\{A,(\sim B)\} = 101100 \neq 111110$

Not equal so result is 0

(e) $A \mid B \& C$

$B \& C = (011) \& (010) = 010$

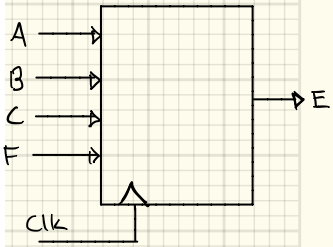
$A \mid B\&C = 101 \mid 010 = 111$

2.34 Consider the following Verilog code:

```
module Q3(A,B,C,F,Clk,E);
input A,B,C,F,Clk;
output reg E;

reg D,G;

initial
begin
    E = 1'b0;
    D = 1'b0;
    G = 1'b0;
end

always @(posedge Clk)
begin
    D <= A & B & C;
    G <= ~A & ~B;
    E <= D | G | F;
end
endmodule
```

(a) Draw a block diagram for the circuit (no gates and at block level only).
(b) Give the circuit generated by the preceding code (at the gate level).

(a)



(b)