

COL215_2018 Minor 2

SAKSHAM DHULL

TOTAL POINTS

9.5 / 10

QUESTION 1

1 Question 1 2 / 2

- + **0.5 pts** Correct Expression
- + **1 pts** Correct Expression with correct constant inputs
- + **1.5 pts** Schematic with out Correct Constants
- ✓ + **2 pts** Schematic with Correct Constants
- + **0 pts** Incorrect

QUESTION 2

2 Question 2 3.5 / 4

- + **2 pts** Controller & DataPath combined
- + **1.5 pts** Partially Correct [one of controller or datapath or merged]
- + **4 pts** Both controller and DataPath fully correct
- + **1 pts** Controller Partially Correct
- + **1 pts** Datapath Partially Correct
- + **1.5 pts** Controller Minor Errors
- + **1.5 pts** DataPath Minor Errors
- ✓ + **3.5 pts** Minor errors in overall Solution
- + **2 pts** One of Controller or datapath correctly implemented
- + **1 pts** Partially Correct [Combined Controller & datapath]
- + **0.5 pts** DataPath Entity
- + **0.5 pts** No credit can be awarded
- + **0 pts** Click here to replace this description.

QUESTION 3

3 Question 3 2 / 2

- ✓ + **0.25 pts** Left Circuit: a=0 b=0
- ✓ + **0.25 pts** Left Circuit: a=1 b=1
- ✓ + **0.25 pts** Left Circuit: a=0 b=1
- ✓ + **0.25 pts** Left Circuit: a=1 b=0
- ✓ + **0.25 pts** Right Circuit: a=0 b=0

- ✓ + **0.25 pts** Right Circuit: a=1 b=1
- ✓ + **0.25 pts** Right Circuit: a=0 b=1
- ✓ + **0.25 pts** Right Circuit: a=1 b=0
- + **0 pts** Incorrect
- **0.2 pts** Lacking Proper Explanation
- + **0 pts** Unattempted

QUESTION 4

4 Question 4 2 / 2

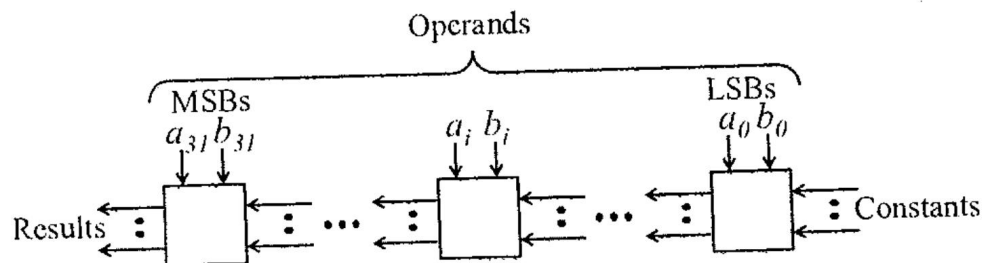
- ✓ + **2 pts** Correct
- + **1 pts** Part (b) is wrong,
- + **1.5 pts** Part (b) can be reduced
- + **0 pts** incorrect
- + **1 pts** Part (a) is wrong
- + **0.5 pts** Part (b) is correct but the reason is wrong
- + **0.5 pts** Part (b) reason is correct.
- + **0 pts** Not scanned properly.
- + **1.5 pts** Part (b) reason correct

6.10.2018

CSL215 Digital Logic and System Design : Minor Test 2

Max Marks : 10

1. A combinational circuit is to be designed for directly comparing (without use of an adder or a subtractor) two 32-bit unsigned integers. The circuit is to be structured as an array of 32 identical blocks as shown. Design the block and draw its schematic diagram. Also specify the values of the constant inputs.



l_i, g_i, e_i → input in i^{th} block.

[2]

e_i →

$$l_i \equiv (a_{i-1} \dots 0 < b_{i-1} \dots 0)$$

$$g_i \equiv (a_{i-1} \dots 0 > b_{i-1} \dots 0)$$

$$e_i \equiv (a_{i-1} \dots 0 = b_{i-1} \dots 0)$$

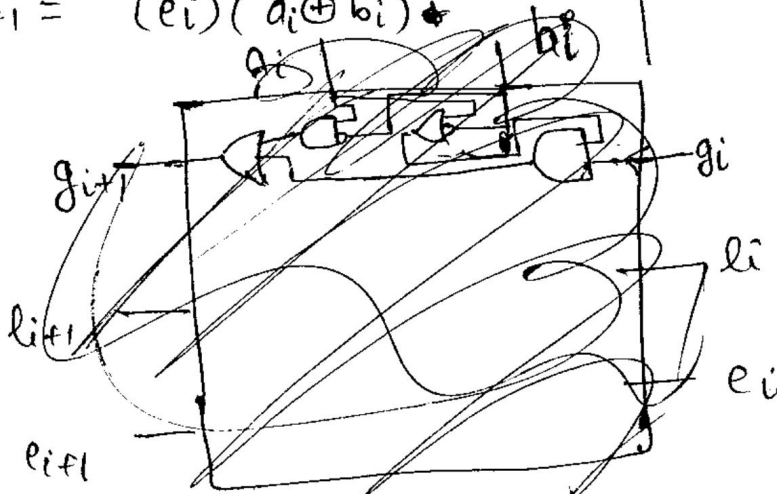
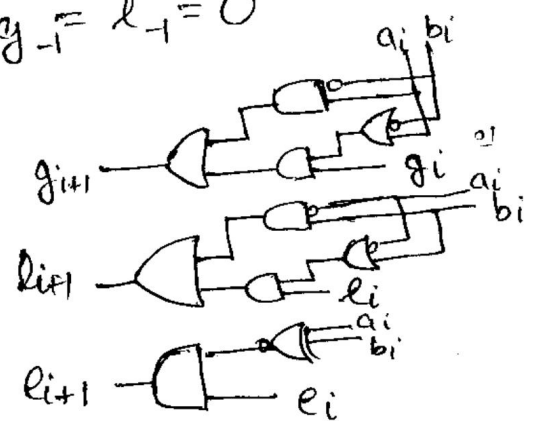
taking ~~initial~~ initial $e_{-1} = 1$ & $g_{-1} = l_{-1} = 0$

for each box i we have.

$$g_{i+1} = (g_i) \cdot (a_i + \bar{b}_i) + a_i \bar{b}_i$$

$$l_{i+1} = (l_i) \cdot (\bar{a}_i + b_i) + \bar{a}_i b_i$$

$$e_{i+1} = (e_i) \cdot (a_i \oplus b_i)$$



6.10.2018

CSL215 Digital Logic and System Design : Minor Test 2

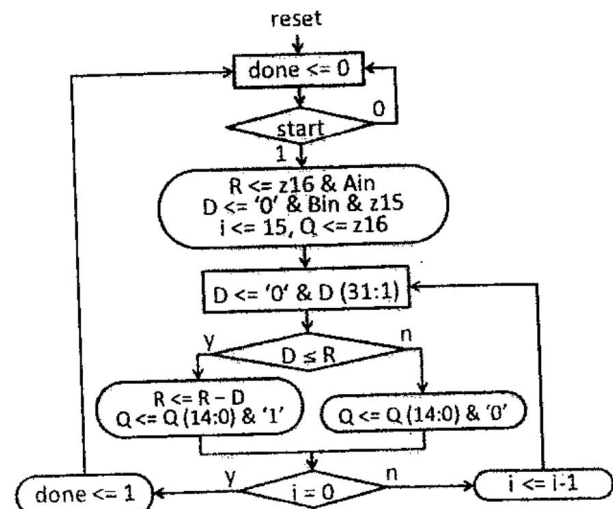
Max Marks : 10

2. What does the ASM chart shown here do?

The inputs are - start (1 bit), Ain (16 bits) and Bin (16 bits). Outputs are - done (1 bit), lower 16 bits of R and Q (16 bits). R and D are 32 bit vectors. There are 2 constants - z15 and z16 which denote vectors of 15 and 16 zeroes, respectively.

Write an equivalent VHDL description, keeping datapath and controller as two separate entities.

The given asm performs the division algorithm.



[4]

Entity Data-en is

PORT (~~start~~ in bit; Ain : in bit-vector (15 downto 0); Bin : in bit-vector (15 downto 0); p, z : out bit; outR, outQ : out bit-vector (15 downto 0));

Architecture of Datapath of Data-en is

Signal D : bit-vector (32 downto 0);
Signal tR : bit-vector (32 downto 0);
Signal tQ : bit-vector (15 downto 0);
Signal i : int;

Begin

Process (Clock),

if rising-edge (Clock) then

if (ldR = 1) then tR <= z16 & Ain end if;

if (ldD = 1) then D <= '0' & Bin & z15 end if;

if (ldi = 1) then i <= 15 end if;

if (ldQ = 1) then tQ <= z16

if (shiftD = 1) then D <= '0' & D(31:1)

if (D <= R) then z <= 1 end if;

if (shiftD <= 1) then tQ <= tQ(14:0) & '0' end if;

if (shiftD <= 1) then tQ <= tQ(14:0) & '1' end if;

if (decR <= 1) then tR <= tR - D end if;

if (i = 0) then p <= 1 end if;

if (outR <= 1) then Q <= tQ, R <= tR(15 downto 0) end if;

if (decI <= 1) then i <= i - 1 end if;

end if;

end process;

end architecture Datapath;

Entity Control is

PORT (p : in bit; z : in bit; outR, outD, outQ, shiftD, shiftQ, shiftR, decR, decQ, decI : out bit);

Architecture Cont of Control

Type state type : (S0, S1)

Signal state type : state : State type

Begin

Process (Clock)

if rising-edge (Clock) then

if (start = 1) then

state <= S0, done <= 1, outR <= 1

end if;

when S1 : shiftD <= 1

if (R = '1') then

decR <= 1, shiftQ <= 1

if (p = 1) then

state <= S0, done <= 1, outR <= 1

end if;

end if;

when S0 :

if (start = 1) then

ldR <= 1, ldi <= 1, ldD <= 1, ldQ <= 1

state <= S1

end if;

end case;

end process

end architecture Cont.

Control On next page 3p.T.O. ->

Q-2

Entity Control is

```
PORT
(in bit p, z, start: in bit; Clock: in bit; read: in bit;
ldR, ldD, ldi, ldQ, shiftD0, shiftQ1, shiftD, decR, decI, done, outRq =: out bit);
end Control.
```

Architecture Cont of Control

```
TYPE state-type (S0; S1);
Signal state: state-type;
```

Begin

Process (Clock)

if (risingedge (Clock)) then

ldR ≤ 0, ldD ≤ 0, ldi ≤ 0, ldQ ≤ 0, shiftD₀ ≤ 0, shiftQ₁ ≤ 0, shiftD ≤ 0, decR ≤ 0, decI ≤ 0,
done ≤ 0, outRq ≤ 0;

case state is

when S₁:

shiftD ≤ 1;

if (z = 1) then decR ≤ 1, ~~dec~~ shiftD₀ ≤ 1; else shiftD₀ ≤ 1 end if

if (p = 1) then state ≤ S₀, done ≤ 1, outRq ≤ 1 else decI ≤ 1 end if

when S₀:

if (start = 1) then ldR ≤ 1, ldD ≤ 1, ldQ ≤ 1, ldi ≤ 1, state ≤ S₁ end if;

end case;

~~end if~~ if (reset = 1) then State ≤ S₀, done ≤ 0 end if;

~~end if~~

end if;

end process;

end Architecture Cont.

NAME: SAKSHAM DHULL

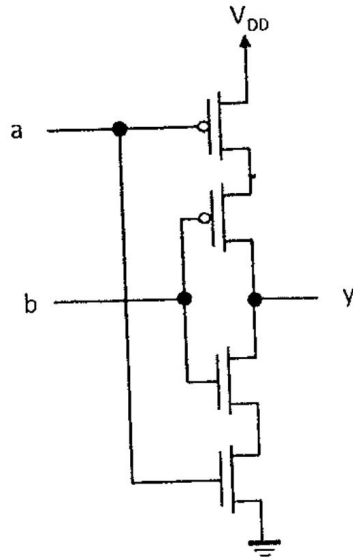
ENTRY NO.: 2019C310370

6.10.2018

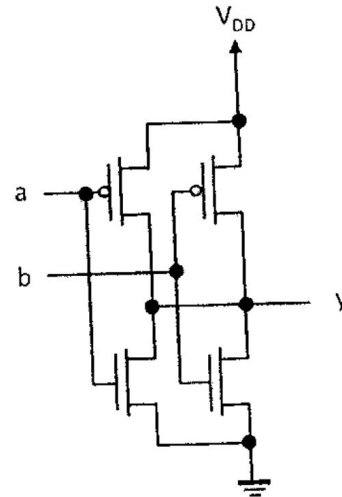
CSL215 Digital Logic and System Design : Minor Test 2

Max Marks : 10

3. Each of the two circuits shown here has two inputs a, b and an output y. What do these circuits do?



a	b	y
0	0	1
0	1	0
1	0	0
1	1	1



a	b	y
0	0	1
0	1	0
1	0	0
1	1	1

[2]

~~Handwritten scribbles and markings at the bottom of the page.~~

NAME: SAKSHAM DHULL

ENTRY NO.: 2017CS10370

6.10.2018

CSL215 Digital Logic and System Design : Minor Test 2

Max Marks : 10

4. Consider a PLA with 3 inputs, 3 outputs and 4 product terms as a configurable module to implement 3-input, 3-output combinational circuits. (a) Give an example of a 3-input, 3-output combinational circuit that can be implemented using this PLA. (b) Give an example of a 3-input, 3-output combinational circuit that cannot be implemented using this PLA. These examples should be in the form of Boolean expressions.

[2]

(a)

$$\begin{aligned}x &= a' + b \\y &= b'c \\z &= abc'\end{aligned}$$

{ a, b, c \rightarrow inputs }
{ four different product terms }
thus could be formed easily

(b).

$$\begin{aligned}x &= a'bc \\y &= ab'c + a'b'c' \\z &= abc' + a'\end{aligned}$$

{ it requires 5 diff product terms hence not possible }.

101.

$$a_i + \bar{b}_i$$