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.

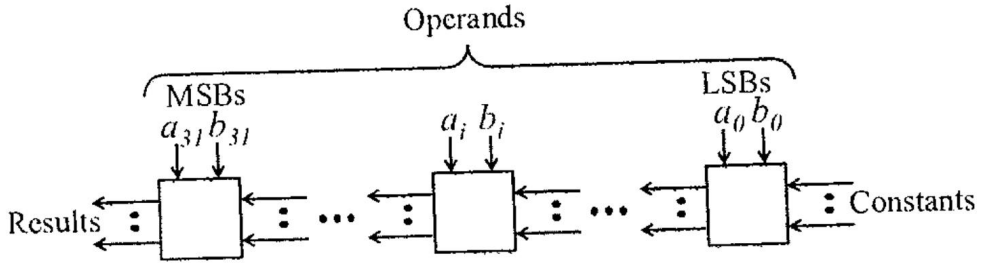
+ **0 pts** Not scanned properly.

+ **1.5 pts** Part (b) reason correct

NAME: SAKSHAM DHULL	ENTRY NO.: 2017CS10370

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.



[2]

l_i, g_i, e_i → input in ith block.

e_i →

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

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

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

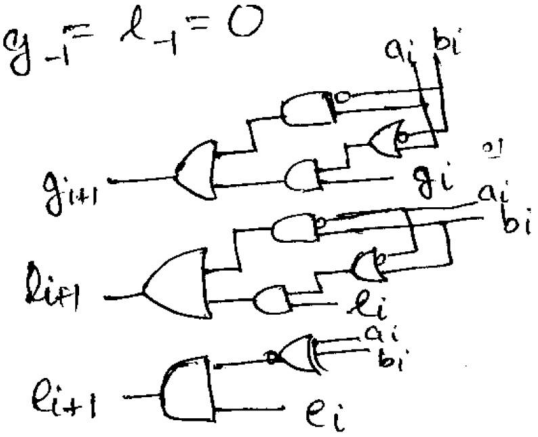
taking initial $e_{-1}=1$ & $g_{-1}=l_{-1}=0$

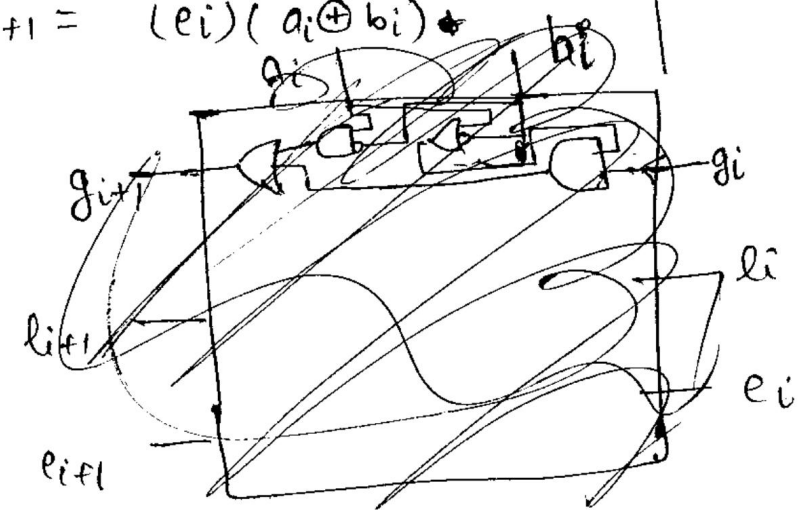
for each box i we have.

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

$l_{i+1} = (l_i)(\bar{a_i}+b_i) + \bar{a_i}b_i$

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





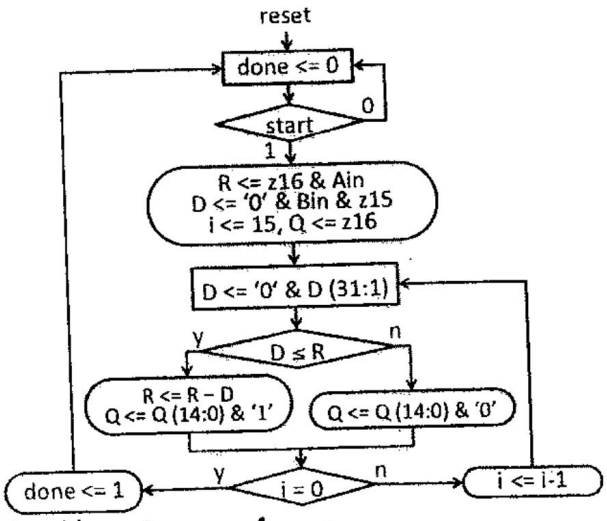
NAME: SAKSHAM DHULL	ENTRY NO.: 2017CS10370

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 performes the division algorithm. [4]

```
entity Data_en is
PORT ( ldR, ldD, ldi, ldQ, shiftD, shiftQ0, shiftQ, decR, deci, inbit
       Ain : in bit_vector(15 downto 0);
       Bin : in bit_vector(15 downto 0);
       z, p : out bit; outRQ : in bit
       Q, R : out bit_vector(15 downto 0)
     );

Architecture Datapath of Data_en is
Signal D : bit_vector(32 downto 0);
Signal tR : bit_vector(32 downto 0);
Signal tQ0 : 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 tQ0 <= z16
  if (shiftD = 1) then D <= '0' & D(31:1)
  if (D ≤ R) then z <= 1 & end if;
  if (shiftQ0 = 1) then tQ0 <= tQ(14:0) & '0' end if;
  if (shiftQ = 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 (outRQ = 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, start, inbit
       ldR, ldD, ldi, ldQ, shiftD, shiftQ0, shiftQ
       decR, deci : out bit );
       done, outRQ

Architecture Cont of Control is
 Type state_type : (S0, S1)
 Signal state : State_type
Begin
 if rising_edge(Clock) then
Process (Clock)
 ldR, ldD, ldi, ldQ, shiftD, shiftQ0, shift
 done, shiftQ, decR, deci <= 0
Case state is
  when S1 : shiftD <= 1
    if (z = '1') then
      decR <= 1, shiftQ0 <= 1
      if (p = 1) then
        state <= S0, done <= 1, outRQ <= 1
      else
        deci <= 1
      end if
  when S0 :
    if (start = '1') then
      ldR <= 1, ldi <= 1, ldD <= 1, ldQ <= 1
      state <= S1
    end if;
 end case;
 end if;
end process
end architecture Cont;
```

else shiftQ <= 1 end if;

(Control on next page) P.T.O. →

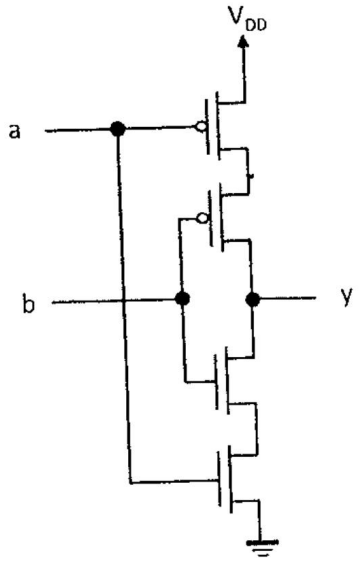
Q-2

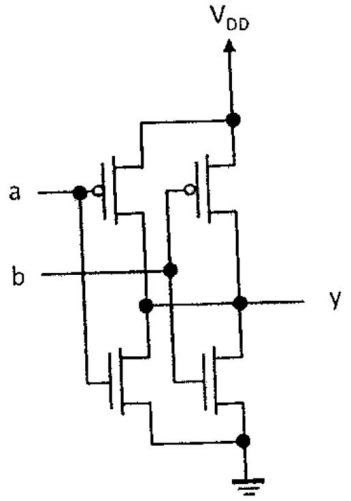
```
entity Control is
PORT
( ~~p: in bit~~ p, z, Start : in bit; Clock: in bit; reset : in bit;
ldR, ldD, ldi, ldQ, shiftQ0, shiftQ1, shiftD, decR, deci, done, outRq =: out bit );
end Control.

Architecture Cont of Control

TYPE state_type (S0, S1);
Signal state: state_type;

Begin
Process (Clock)
if (rising_edge (Clock) then
   ldR ⇐ 0, ldD ⇐ 0, ldi ⇐ 0, ldQ ⇐ 0, shiftQ0 ⇐ 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~~ shiftQ1 ⇐ 1; else shift Q0 ⇐ 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~~ ~~case~~ ~~if reset~~ if (reset = 1) then State ⇐ S0, done ⇐ 0 end if;
        end if;
    end process;
    end Architecture Cont.
```

NAME: SAKSHAM DHULL	ENTRY NO.: 2019CS10370

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	z
1	0	z
1	1	0

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

[2]

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) $x = a' + b$

$y = b'c$

$z = abc'$

{ a, b, c → inputs }

{four different product terms} thus could be formed easily

(b). $x = a'bc$

$y = ab'c + a'b'c'$

$z = ab\cdot c' + \cdot a'$

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

$ai+\bar{b}i$

10.