USN										

Dayananda Sagar College of Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Banashankari, Bangalore-560078, Karnataka
Tel : +91 80 26662226 26661104 Extn : 2731 Fax : +90 80 2666 0789
Web - http://www.dayanandasagar.edu Email : hod-ece@dayanandasagar.edu
(An Autonomous Institute Affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)
(Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade)

Department of Electronics & Communication Engg.

Continuous Internal Evaluation – I

Course Name : Digital System Design Using Verilog	Date :	06.10.2020
Course Code : 18EC5DCDSV	Day :	Tuesday
Semester : VI	Timings :	1 to 2.30 P.M
Max Marks : 50 M	Duration :	1½ Hrs.

No.		Question Description	M ks	CO & Levels
Q1	(a)	Read the following statements and choose which one is true. i) Net types are continuously updated whenever one of the signals on the right hand side of the assignment changes ii) Describing your identifier with type reg will give you a hardware register when you synthesize. iii) Register types and net types cannot be used together in assignments iv) Output and inout ports should be connected with reg data type only.	1	

	(b)	What does the following code segment implement?
		assign d = ~(c & b);
		assign c = ~(a & d);
		i) A 2-bit shift-register ii) Two NOR functions connected in cascade
		iii) A one-bit latch iv) A 2-bit comparator
			1	

	(c)	Which of these statements is not true? integer a; reg [2:0] b, c; wire [2:0]out; b = 3’b010; c = 3’b111 assign out = (a = = 3)?b:c ; i) When a is equal to 3, out gets the value of 010. ii) When a is equal to x, out gets the value of xxx. iii) When a is less than 3, out gets the value of 111. iv) When a is greater than 3, out gets the value of 111.	1	
	(d)	Read the statements and select which are correct**.** 1. Continuous assignments are made to net types only. 2. Continuous assignments are made using the keyword assign. 3. The right hand side of the continuous assignment is reevaluated each time an operand changes and reassigned to the left hand side. 4. Continuous assignments can assign to register types. i) 1,2 and 3. ii) 1 and 2. iii) 2 and 3. iv) 1,2,3 and 4.	1	

	(e)	Consider the following piece of Verilog code. If the procedure is called when the value of X is ‘0’,what do you think happens to the value of B?
		always@(X or A)
		begin
		    If (X)
		        B <= A;
		end
		i) B is assigned the value of A. ii) B is assigned ‘0’
			1	

		iii) B is assigned the value of X iv) B retains its existing value		
	(f)	Where are continuous assignments made? i) Anywhere in the code ii) Within a procedure iii) Outside a procedure iv) Within a procedure but outside of the begin-end block	1	
	(g)	What types are assigned within procedures? i) Net types and register types. ii) Register types only iii) Net types only iv) None of the above	1	
	(h)	Read the statements and select which one is true. i) Initial and always blocks are executed in turn in the order that they appear in the code. ii) Initial and always blocks are synthesizable iii) Only one initial block can be included per module iv) Initial blocks are not synthesizable but always blocks are	1	

	(i)	Consider the following code segment, the final value of variable “r” will be_____________
		integer p, q, r;
		initial
		begin
		p = 55; q = 10; r = 5;
		p = q * r;
		q = p – 25;
		r = p + q;
		end
		i) 75 ii) 65 iii) 80 iv) 40
			1	

	(j)	Consider the following code segment, the final value of variable “r” will be_____________
		integer p, q, r;
		initial
		begin
		p = 55; q = 10; r = 5;
		p <= q * r;
		q < = p – 25;
		r <= p + q;
		end
		i) 75 ii) 80 iii) 65 iv) 40
			1	

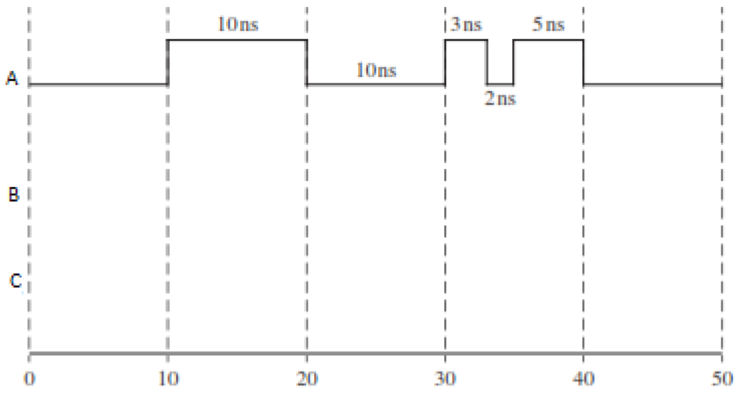
Q2	a)	Write a Verilog code for a full subtracter using logic equations and using this module as a component; Develop a Verilog code for 4 bit subtracter.	8	L4,CO2

	b)	For the following Verilog code segment:
		reg a;
		reg [2:0] b, c;
		reg [10:0] x;
		a = 3’b100; b = 1’b1; c = 3’b101;
		x = {{2{b}}, a,{2{c}}};
		What will be the value of x?
			2	L2,CO2

Q3	a)	An AB flip-flop responds to the falling clock edge as follows: If *A* = *B* = 0, the flip-flop changes state. If *A* = 0 and *B* = 1, the flip-flop output is set to 1. If *A* = 1 and *B* = 0, the flip-flop output is set to 0. If *A* = *B* = 1, no change of flip-flop state occurs. The flip-flop is cleared asynchronously if *CLRn* = 0. Write a complete Verilog module that implements an AB flip-flop.	6	L3,CO2
	b)	Consider the following Verilog code segment, if the initial value of IR is 12345678 (in hexadecimal).	4	L3,CO2

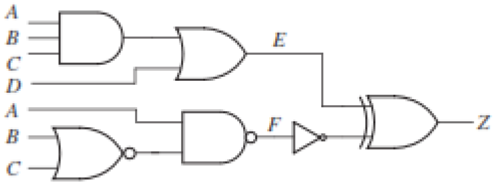
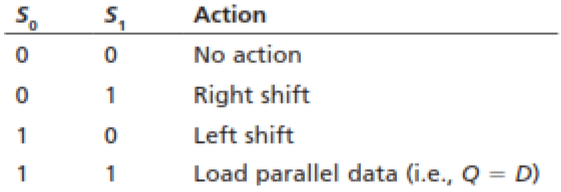
USN										

		wire [31:0] IR;
		wire [3:0] data;
		wire [15:0] d1;
		wire [31:16] d2;
		assign d1 = IR[31:16];
		assign d2 = IR[15:0];
		assign data = d1[7:4] + d2[19:16] + d2[31:28];
		the value of “data” in decimal will be _____
		(Note that “data” is a 4-bit variable **Note answer should be shown step by step**)		

Q4	(a)	Given the following timing waveform for A, draw B and C.
		wire #3 B;
		assign #5 B = A;
		wire #5 C;
		assign #3 C = A;
		
		 A
		B
		C
		10ns
		10ns
		3ns
		2ns
		5ns
		0
		10
		20
		30
		40
		50
			4	L3,CO1

	(b)	Draw the hardware circuit represented by the following Verilog process:
		always @(clk,clr)
		begin
		if(clr == 1'b1)
		Q <= 1'b0;
		else if(clk == 1'b0 && CE == 1'b1)
		begin
		if(C == 1'b0)
		Q <= A & B;
		else
		Q <= A | B;
		end
		end
			6	L3,CO5

		OR		
5	a)	Explain in detail Compilation, simulation and synthesis of Verilog Code with a neat block diagram	7	L2,CO1
	b)	Write Verilog modules using conditional operator that is equivalent to the following code: A = B1 when C = 1 else B2 when C = 2 else B3 when C = 3 else 0;	3	L2,CO1
6	a)	Consider the fallowing Verilog code: **module** bitwise_op ( );	3	L2,CO2

<table>
<tr><td></td><td></td><td>**reg** [2:0] a, b, c, x, y, z;
initial
begin
a = 5;b = 3’b111; c =3’d x;
x = a & b;
y = a & c;
z = b &amp; 1;
$display(“x = %b, y = %b, z = %b”,
x,y,z);
end
endmodule
Write the final result for x,y and z ?</td><td></td><td></td></tr>
<tr><td></td><td>b)</td><td>**Given**
reg signed [7:0] B = 8’hC7;
i)B>> 4
ii)B >>> 4
iii)B << 4
iv)B <<< 4</td><td>2</td><td>L2,CO1</td></tr>
<tr><td></td><td>c)</td><td>Write a Verilog description of the following combinational circuit using concurrent statements. Each gate has a 5-ns delay, excluding the inverter, which has a 2-ns delay.

 </td><td>5</td><td>L3,CO2</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>**OR**</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>a)</td><td>Write the Verilog code for the following 8-bit bidirectional synchronous shift register with parallel load capability. The notation used to represent the input/output pins is explained as follows:
CLR Asynchronous Clear, which overrides all other inputs
Q(7:0) 8-bit output
D(7:0) 8-bit input
S0,S1 mode control inputs
LSI serial input for left shift
RSI serial input for right shift
The mode control inputs work as follows:
<table><tr><th>S_0</th><th>S_1</th><th>Action</th></tr><tr><td>0</td><td>0</td><td>No action</td></tr><tr><td>0</td><td>1</td><td>Right shift</td></tr><tr><td>1</td><td>0</td><td>Left shift</td></tr><tr><td>1</td><td>1</td><td>Load parallel data (i.e., *Q* = *D*)</td></tr></table></td><td>6</td><td>L3,CO2</td></tr>
<tr><td></td><td>b)</td><td>**Given**
wire a = 1'b1;
wire [1:0] b = 2'b10;
wire [2:0] c = 3'b101;
evaluate
i. {a, b[0], c[1]}
ii. {a, b, c, 2'b01}</td><td>2</td><td>L2,CO1</td></tr>
</table>

USN										

		Given:
		reg signed [3:0] C = 4'b1101;
		reg signed [2:0] D = 3'b111;
		reg [3:0] A = 4'b1101;
		reg [2:0] B = 3'b111;
		reg signed [7:0] S;
		
		Evaluate
		S = C + D;
		S = A + B;
			2	L2,CO1

Dayananda Sagar College of Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Banashankari, Bangalore-560078, Karnataka
Tel : +91 80 26662226 26661104 Extn : 2731 Fax : +90 80 2666 0789
Web - http://www.dayanandasagar.edu   Email : hod-ece@dayanandasagar.edu
(An Autonomous Institute Affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)
(Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade)

Date of test : 06.10.2020	Digital System Design Using Verilog	Max Marks : 50.
Day : Tuesday		Sub Mentor : Dr. Dinesh P
Branch : E and C	Sub code: 18EC5DCDSV	Sub Mentor Sign:
Semester : 5		Staff i/c of sec : Dr. Rajagopal A
Section : A,B,C and D	Internal Test	Staffs i/c sign :
Timings : 1.00 to 2.30	CIE-I	HOD Name : Dr. TCM
Test Duration : 1½ Hrs.	Test Solutions	HOD’s sign :

Q. No.	Test question paper solutions with steps	Marks Allocation

1.a i) Net types are continuously updated whenever one of the signals on the right hand side of the assignment changes
1.b iii) A one-bit latch
1.c ii) When a is equal to x, out gets the value of xxx.
1.d i) 1,2 and 3.
1.e iv) B retains its existing value
1.f iii) Outside a procedure
1.g ii) Register types only
1.h iv) Initial blocks are not synthesizable but always blocks are
1.i i) 75
1.j iii) 65

2. a

```
module Fullsub(x, y, bin, diff, bout);
input x, y, bin;
output diff, bout;
assign diff = x ^ y ^ bin;
assign bout = (~x & bin) | (~x & y) | (bin & y);
endmodule ->3M

module Sub4bit(a, b, bin, d, bout);
input[3:0] a, b;
input bin;
output[3:0] d;
output bout; wire[3:0] bo;
Fullsub s1(a[0], b[0], bin, d[0], bo[1]);
Fullsub s2(a[1], b[1], bo[1], d[1], bo[2]);
Fullsub s3(a[2], b[2], bo[2], d[2], bo[3]);
Fullsub s4(a[3], b[3], bo[3], d[3], bout);
endmodule ->5M
```

3+5->8M

```
reg a;
  reg [2:0] b, c;
  reg [10:0] x;
    a = 3’b100; b = 1’b1; c = 3’b101;
     x = {{2{b}}, a,{2{c}}};
```

x=11100101101->2M

2.b

3.a

```
module ABFF(A, B, CLK, CLRn, Q, Qn);
input A, B, CLK, CLRn;
output reg Q;
output Qn; ->2M
initial
begin
Q <= 0;
end
 always @(CLK, CLRn)
 begin if(CLRn == 1'b0)
 begin Q <= 0;
end
else if(CLK == 0)
//( negedge clk) can be used
 begin if(A == 0 && B == 0) Q <= ~Q;
 else if(A == 0 && B == 1) Q <= 1;
else if(A == 1 && B == 0) Q <= 0;
else if(A == 1 && B == 1) Q <= Q;
end
end
assign Qn = ~Q;
end module  ->4M
```

2+4->6M

3.b

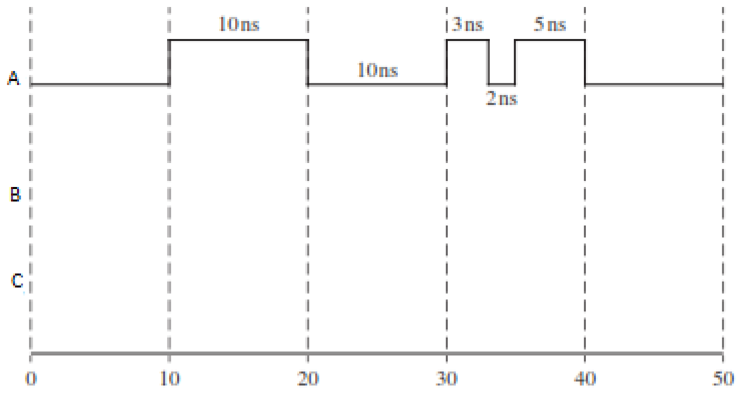
d1=0001 0010 0011 0100 ->1M

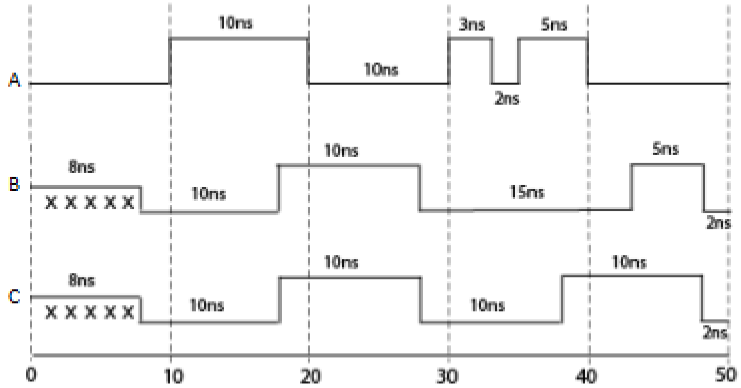
d2=0101 0110 0111 1000 ->1M

data=0011+1000+0101=0000 with 1 carry-1M

data=0000=0->1M

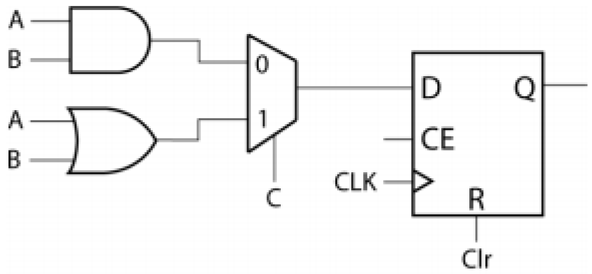
4.a





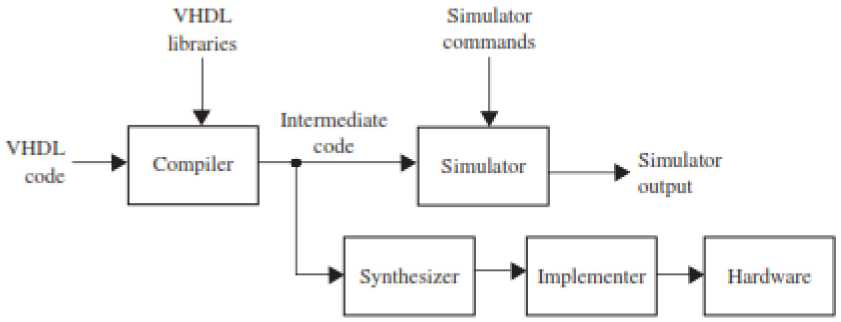
Waveform B and C ->2+2->4M

4.b)



1x6=6M

5.a



block diagram-4M

explanation-3M->4+3->7M

5.b

assign A = (C==1)? B1 : ((C==2)? B2 : ((C==3)? B3 : 0)); ->3M

6.a

x = 101,

y = x0x,

z = 001 ->1+1+1->3M

6.b

B >> 4 == 2’h0C

B >>> 4 == 2’hFC

B << 4 == 8’h70

B <<< 4 == 8’h70-> 4 x 0.5- >2M

6.c

```
module gate(A, B, C, D, Z);
input A, B, C, D;
output Z;
 wire E, F, G, H, I;
 assign #5 H = A & B & C;
 assign #5 E = H | D;
 assign #5 G = ~(B | C);
assign #5 F = ~(G & A);
assign #2 I = ~F;
assign #5 Z = E ^ I;
endmodule
```

7.a

```
module shift8(Q, D, CLR, CLK, S0, S1, LSI, RSI);
 input[7:0] D;
 output reg[7:0] Q;
input CLR, CLK, S0, S1, LSI, RSI;->2M
 initial
 begin Q = 0;
end
 always @(CLK, CLR)
 begin if(CLR == 1) Q <= 0;
 else if(CLK == 1)
 begin if(S0 == 1 && S1 ==1)
Q <= D;
 else if(S0 == 0 && S1 == 1)
Q <= {RSI, Q[7:1]};
else if(S0 == 1 && S1 == 0)
 Q <= {Q[6:0], LSI};
 else Q <= Q;
end
end
endmodule ->3M 2+3->5M
```

7.b

i) 8’b11010101

ii) 3’b100 1x2->2M

7.c

S=C+D=8’b11111100

S=A+B= 8’b00010100