DAILY ASSESSMENTFORMAT

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 06-06-2020 | Name: | mamatha |
| Course: | DIGITAL DESIGN USING  HDL | USN: | 4al16ec035 |
| Topic: | * FPGA Basics: Architecture, ApplicationsandUsesVerilog HDL Basics by Intel Verilog Testbenchcodetoverifythe   design under test (DUT) | Semester & Section: | 6th sem B sec |
| Github Repository: | Mamatha-m |  |  |

|  |
| --- |
| FORENOON SESSION DETAILS |
|  |
| **Report –**  FPGA Basics: Architecture, Applications and Uses:   * AbasicFPGAarchitecture(Figure1)consistsofthousandsoffundamentalelements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects,calledafabric,thatroutessignalsbetweenCLBs.Input/output(I/O)blocks interfacebetweentheFPGAandexternaldevices. * Dependingonthemanufacturer,theCLBmayalsobereferredtoasalogicblock(LB),a logicelement(LE)oralogiccell(LC).   Application:   * Manyapplicationsrelyontheparallelexecutionofidenticaloperations;theabilityto configuretheFPGA’sCLBsintohundredsorthousandsofidenticalprocessingblockshas applicationsinimageprocessing,artificialintelligence(AI),datacenterhardware accelerators,enterprisenetworkingandautomotiveadvanceddriverassistancesystems (ADAS). * Manyoftheseapplicationareasarechangingveryquicklyasrequirementsevolveand newprotocolsandstandardsareadopted.FPGAsenablemanufacturerstoimplement systemsthatcanbeupdatedwhennecessary. * AgoodexampleofFPGAuseishigh-speedsearch:MicrosoftisusingFPGAsinitsdata |

|  |
| --- |
| centerstorunBingsearchalgorithms.TheFPGAcanchangetosupportnewalgorithms astheyarecreated.Ifneedschange,thedesigncanberepurposedtorunsimulationor modelingroutinesinanHPCapplication.Thisflexibilityisdifficultorimpossibleto achieve with anASIC.   * OtherFPGAusesincludeaerospaceanddefense,medicalelectronics,digitaltelevision, consumerelectronics,industrialmotorcontrol,scientificinstruments,cybersecurity systems and wirelesscommunications.   Verilog HDL Basics byIntel:   * + VerilogisaHARDWAREDESCRIPTIONLANGUAGE(HDL).Itisalanguageused fordescribingadigitalsystemlikeanetworkswitchoramicroprocessorora memory or aflip−flop.   + Itmeans,byusingaHDLwecandescribeanydigitalhardwareatanylevel. Designs,whicharedescribedinHDLareindependentoftechnology,veryeasyfor designinganddebugging,andarenormallymoreusefulthanschematics, particularly for largecircuits.   + Behaviorallevel   + Register-transferlevel   + Gatelevel   + LexicalTokens   + Numbers   + Identifiers   + Operators   + DataTypes   + Operators   + Operands   + Modules |

Verilog Test bench code to verify the design under test (DUT):

TASK:

Implementa4:1MUXandwritethetestbenchcodetoverifythemodule: Multiplexer(4:1)

Verilog design:

module mux41(

inputi0,i1,i2,i3,sel0,sel1,outputregy);always @(\*) begin

case({sel0,sel1})2'b00:y=i0;2'b01:y=i1;2'b10:y=i2;2'b11:y=i3;endcase

endendmodule TestBench:

module tb\_mux41;

regI0,I1,I2,I3,SEL0,SEL1;wireY;

mux41MUX(.i0(I0),.i1(I1),.i2(I2),.i3(I3),.sel0(SEL0),.sel1(SEL1),.y(Y));

initial begin I0 =1'b0; I1= 1'b0; I2 =1'b0;

I3 =1'b0; SEL0 =1'b0; SEL1 =1'b0; #45 $finish;

end

always#2I0=~I0;always#4I1=~I1;always#6I2=~I1;always#8I3=~I1;always#3SEL0=~SEL0;always#3SEL1=~SEL1;

always @(Y)

$display("time=%0tINPUTVALUES:\tI0=%bI1=%bI2=%bI3=%bSEL0=%b

SEL1=%b\toutputvalueY=%b",$time,I0,I1,I2,I3,SEL0,SEL1,Y);endmodule

output:

time=0INPUTVALUES:outputvalueY=0 time=2INPUTVALUES:outputvalueY=1

time=3INPUTVALUES:outputvalueY=0 time=6INPUTVALUES:outputvalueY=1

time=8INPUTVALUES:outputvalueY=0 time=14INPUTVALUES:outputvalueY=1

time=15INPUTVALUES:outputvalueY=0

I0=0I1=0I2=0I3=0SEL0=0SEL1=0 I0=1I1=0I2=0I3=0SEL0=0SEL1=0 I0=1I1=0I2=0I3=0SEL0=1SEL1=1 I0=1I1=1I2=0I3=0SEL0=0SEL1=0 I0=0I1=0I2=0I3=0SEL0=0SEL1=0

I0=1I1=1I2=1I3=0SEL0=0SEL1=0I0=1I1=1I2=1I3=0SEL0=1SEL1=1