EENG2131 - Lab 6 – Digital Clock

Antonio Veguilla Hernandez

Student ID: P000166790

For this lab, we’ll be creating a Digital Clock using the Seven-Segment displays on the Basys3 FPGA dev board. We’ll have the left two digits represent the Hours and the right two digits represent the Minutes. We’ll use the right-most decimal point to flash every second.

|  |  |
| --- | --- |
| Part 1 –Clock Divider Bring back your Clock Divider module that produced 1 Hz output. Simplify the module to have just one output but change it to be a “once every minute” signal, or 1/60 Hz. Connect it to one of the LEDs on the FPGA to prove that it’s working.  **My Verilog module**  module Clock1min(clk, clk\_1min, clk\_1sec);  input clk; // Standard 100MHz clock  output reg clk\_1min = 0; //60Hz clock output = 1 minute  output reg clk\_1sec = 0; // 1Hz clock output = 1 second  reg [39:0] count10 = 0; //counter with a magnitude of 2^40  reg [28:0] count11 = 0; //counter with magnitude of 2^29.  always @(posedge clk)  begin  if(count11 == 50000000) //1second//  begin  clk\_1sec = ~clk\_1sec;  count11 <=0;  end  else  count11 <= count11 + 1;  if(count10 == 3000000000) //1minute//  begin  clk\_1min = ~clk\_1min;  count10 <= 0;  end  else  begin  count10 <= count10 + 1;  end  en  endmodule Part 2 – Enhance SevenSegmentFourDigits module to support DP Since we ignored the decimal point (DP) last time, we need to enhance our SevenSegmentFourDigits module to support the decimal points:   * Expand the **segN** signal to be 8-bit (that is, [7:0]) with the upper bit controlling the decimal point). | * Add a new 4-bit input called **DPs** that will indicate which decimal points should be lit. This should be an active-high signal, so you may need to invert the values before driving the **segN** signal. * Add the necessary logic to select the correct bit of **DPs** to be output as part of **segN**.   **My Verilog Module**  module SevenSegmentFourDigits(clk, bcd0, bcd1, bcd2, bcd3, AN, segN, DPs);  input clk; //Standard 100MHz clock  input [3:0]bcd0; //right most digit  input [3:0]bcd1;  input [3:0]bcd2;  input [3:0]bcd3; //left-most digit  input [3:0]DPs; //decimal point control, active high  output [3:0]AN; //digit enable active low  output [7:0]segN; //segment enable, active low  BCDtoSevenSegN dut(trueBCD, segN[6:0]);  DigitController dut1(clk, AN);  reg [3:0]trueBCD = 0;  reg segN7;  assign segN[7] = segN7;  always@(AN)  begin  if(AN[0] == 0)  begin  trueBCD = bcd0;  segN7 = ~DPs[0];  end  else if(AN[1] == 0)  begin  trueBCD = bcd1;  segN7 = ~DPs[1];  end  else if(AN[2] == 0)  begin  trueBCD = bcd2;  segN7 = ~DPs[2];  end  else if(AN[3] == 0)  begin  trueBCD = bcd3;  segN7 = ~DPs[3];  end  end  endmodule |

|  |  |
| --- | --- |
| Part 3 – BCD Counter BCD values typically range from 0-9, but with a clock, some of the digits need to “roll over” at 6! Let’s design a 4-bit BCD counter that has a parameter for the roll-over number.  **My Verilog Module**  module BCD\_Counter(clk, resetN, value, rollover);  parameter maxValue = 9;  parameter resetVal = 0;  input clk; //counter should increment at each clk posedge  input resetN; //reset, active low, synchronous to clk  output reg [3:0]value = 0; //counter value  output reg rollover = 0; //binary output to indicate roll-over  always @(posedge clk, negedge resetN)  begin  if(!resetN)  begin  value <= resetVal;  rollover <= 0;  end  else if(value == maxValue)  begin  rollover <= 1;  value <= 0;  end  else  begin  value <= value +1;  rollover <= 0;  end  end  endmodule  **My Verilog Tesbench Module**  module testbench\_BCD\_Counter;  reg clk, resetN;  wire [3:0]value;  wire rollover;  BCD\_Counter dut1\_1(clk, resetN, value, rollover);  initial begin  clk = 0;  resetN = 1;  #10  clk = 1; //1st posedge  #10  clk = 0;  #10  clk = 1; //2nd posedge  #10  clk = 0;  #10  clk = 1; //3rd posedge  #10  clk = 0;  #10  clk = 1; //4th posedge | #10  clk = 0;  #10  clk = 1; //5th posedge  #10  clk = 0;  #10  clk = 1; //6th posedge  #10  clk = 0;  #10  clk = 1; //7th posedge  #10  clk = 0;  #10  clk = 1; //8th posedge  #10  clk = 0;  #10  clk = 1; //9th posedge  #10  clk = 0;  #10  clk = 1; //10th posedge  #10  clk = 0;  #10  clk = 1; //11th posedge  #10  clk = 0;  #10  clk = 1; //12th posedge  #10  clk = 0;  #10  clk = 1; //13th posedge  #10  clk = 0;  #10  clk = 1; //14th posedge  #10  clk = 0;  #10  clk = 1; //15th posedge  #10  clk = 0;  #10  clk = 1; //16th posedge  #10  clk = 0;  #10  clk = 1; //17th posedge  #10  clk = 0;  #10  clk = 1; //18th posedge |

|  |  |
| --- | --- |
| #10  clk = 0;  #10  clk = 1; //19th posedge  #10  clk = 0;  #10  clk = 1; //20th posedge  #10  //////Sequence to switch resetN/////////////////  clk = 0;  #10  clk = 1; //1st posedge  resetN = 0;  #10  clk = 0;  #10  clk = 1; //2nd posedge  #10  clk = 0;  #10  clk = 1; //3rd posedge  #10  clk = 0;  #10  clk = 1; //4th posedge  #10  clk = 0;  #10  clk = 1; //5th posedge  #10  clk = 0;  #10  clk = 1; //6th posedge  resetN = 1;  #10  clk = 0;  #10  clk = 1; //7th posedge  #10  clk = 0;  #10  clk = 1; //8th posedge  #10  clk = 0;  #10  clk = 1; //9th posedge  #10  clk = 0;  #10  clk = 1; //10th posedge  #10  clk = 0;  #10  clk = 1; //11th posedge  resetN = 0;  #10  clk = 0; | #10  clk = 0;  #10  clk = 1; //12th posedge  #10  clk = 0;  #10  clk = 1; //13th posedge  #10  clk = 0;  #10  clk = 1; //14th posedge  #10  clk = 0;  #10  clk = 1; //15th posedge  #10  clk = 0;  #10  clk = 1; //16th posedge  resetN =1;  #10  clk = 0;  #10  clk = 1; //17th posedge  #10  clk = 0;  #10  clk = 1; //18th posedge  #10  clk = 0;  #10  clk = 1; //19th posedge  #10  clk = 0;  #10  clk = 1; //20th posedge  #10  clk = 0;  #10  clk = 1; //21st posedge  resetN =1;  #10;  end  endmodule |

|  |  |
| --- | --- |
| Part 4 – Hours:Minutes Clock Create a module to act as an Hours and Minutes clock display, with the following interface:  **My Verilog Module**  module ClockHoursMinutes(clk, reset, AN, segN);  input clk;  input reset; //Resets Active High  output [3:0]AN; //digit enable, active low  output [7:0] segN; //segment enable, active low  reg resetN;  wire [3:0]bcd0;  wire [3:0]bcd1;  wire [3:0]bcd2;  wire [3:0]bcd3;  /////////////////Instatiate////////////////////  SevenSegmentFourDigits(clk, bcd0, bcd1, bcd2, bcd3, AN, segN, DPs);  Clock1min dut\_clk(clk, clk\_1min, clk\_1sec);  //////////////////////////////////////////////  wire clk\_min;  wire clk\_1sec;  wire [3:0]DPs;  assign DPs[2] = clk\_1sec; //This will make the DP between HRs & mins to blink every second  BCD\_Counter #(.resetVal(0), .maxValue(9))cnt1(clk\_1min, resetN, bcd0, rollover0);  BCD\_Counter #(.resetVal(0), .maxValue(5))cnt2(rollover0, resetN, bcd1, rollover1);  BCD\_Counter #(.resetVal(1), .maxValue(9))cnt3(rollover1, resetN, bcd2, rollover3);  BCD\_Counter #(.resetVal(0), .maxValue(1))cnt4(rollover3, resetN, bcd3, rollover4);  ///////////////////////////////////////////////  always  begin  if(bcd3 == 1 & bcd2 == 2 & bcd1 == 5 & bcd0 == 9)  begin  resetN = 0;  end  else if(reset)  resetN = 0;  else  resetN = 1;  end  endmodule  **My Verilog FPGA Interface module**  module FPGA\_Interface(sw,an,seg,clk,led,btnC);  output [3:0]an;  input clk;  input [15:0]sw;  input btnC;  output [7:0]seg;  output [15:0]led;  ClockHoursMinutes(clk, btnC, an, seg);  endmodule | Constrain File  ## This file is a general .xdc for the Basys3 rev B board  ## To use it in a project:  ## - uncomment the lines corresponding to used pins  ## - rename the used ports (in each line, after get\_ports) according to the top level signal names in the project  ## Clock signal  set\_property PACKAGE\_PIN W5 [get\_ports clk]  set\_property IOSTANDARD LVCMOS33 [get\_ports clk]  create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]  ## Switches  set\_property PACKAGE\_PIN V17 [get\_ports {sw[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[0]}]  set\_property PACKAGE\_PIN V16 [get\_ports {sw[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[1]}]  set\_property PACKAGE\_PIN W16 [get\_ports {sw[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[2]}]  set\_property PACKAGE\_PIN W17 [get\_ports {sw[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[3]}]  set\_property PACKAGE\_PIN W15 [get\_ports {sw[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[4]}]  set\_property PACKAGE\_PIN V15 [get\_ports {sw[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[5]}]  set\_property PACKAGE\_PIN W14 [get\_ports {sw[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[6]}]  set\_property PACKAGE\_PIN W13 [get\_ports {sw[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[7]}]  set\_property PACKAGE\_PIN V2 [get\_ports {sw[8]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[8]}]  set\_property PACKAGE\_PIN T3 [get\_ports {sw[9]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[9]}]  set\_property PACKAGE\_PIN T2 [get\_ports {sw[10]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[10]}]  set\_property PACKAGE\_PIN R3 [get\_ports {sw[11]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[11]}]  set\_property PACKAGE\_PIN W2 [get\_ports {sw[12]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[12]}]  set\_property PACKAGE\_PIN U1 [get\_ports {sw[13]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[13]}]  set\_property PACKAGE\_PIN T1 [get\_ports {sw[14]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[14]}]  set\_property PACKAGE\_PIN R2 [get\_ports {sw[15]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sw[15]}]  ## LEDs  set\_property PACKAGE\_PIN U16 [get\_ports {led[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[0]}]  set\_property PACKAGE\_PIN E19 [get\_ports {led[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[1]}]  set\_property PACKAGE\_PIN U19 [get\_ports {led[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[2]}]  set\_property PACKAGE\_PIN V19 [get\_ports {led[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[3]}] |

|  |  |
| --- | --- |
| set\_property PACKAGE\_PIN W18 [get\_ports {led[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[4]}]  set\_property PACKAGE\_PIN U15 [get\_ports {led[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[5]}]  set\_property PACKAGE\_PIN U14 [get\_ports {led[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[6]}]  set\_property PACKAGE\_PIN V14 [get\_ports {led[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[7]}]  set\_property PACKAGE\_PIN V13 [get\_ports {led[8]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[8]}]  set\_property PACKAGE\_PIN V3 [get\_ports {led[9]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[9]}]  set\_property PACKAGE\_PIN W3 [get\_ports {led[10]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[10]}]  set\_property PACKAGE\_PIN U3 [get\_ports {led[11]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[11]}]  set\_property PACKAGE\_PIN P3 [get\_ports {led[12]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[12]}]  set\_property PACKAGE\_PIN N3 [get\_ports {led[13]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[13]}]  set\_property PACKAGE\_PIN P1 [get\_ports {led[14]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[14]}]  set\_property PACKAGE\_PIN L1 [get\_ports {led[15]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {led[15]}]  ##7 segment display  set\_property PACKAGE\_PIN W7 [get\_ports {seg[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[0]}]  set\_property PACKAGE\_PIN W6 [get\_ports {seg[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[1]}]  set\_property PACKAGE\_PIN U8 [get\_ports {seg[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[2]}]  set\_property PACKAGE\_PIN V8 [get\_ports {seg[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[3]}]  set\_property PACKAGE\_PIN U5 [get\_ports {seg[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[4]}] | set\_property PACKAGE\_PIN V5 [get\_ports {seg[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[5]}]  set\_property PACKAGE\_PIN U7 [get\_ports {seg[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[6]}]  set\_property PACKAGE\_PIN V7 [get\_ports seg[7]]  set\_property IOSTANDARD LVCMOS33 [get\_ports seg[7]]  set\_property PACKAGE\_PIN U2 [get\_ports {an[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[0]}]  set\_property PACKAGE\_PIN U4 [get\_ports {an[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[1]}]  set\_property PACKAGE\_PIN V4 [get\_ports {an[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[2]}]  set\_property PACKAGE\_PIN W4 [get\_ports {an[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[3]}]  ##Buttons  set\_property PACKAGE\_PIN U18 [get\_ports btnC]  set\_property IOSTANDARD LVCMOS33 [get\_ports btnC]  #set\_property PACKAGE\_PIN T18 [get\_ports btnU]  #set\_property IOSTANDARD LVCMOS33 [get\_ports btnU]  #set\_property PACKAGE\_PIN W19 [get\_ports btnL]  #set\_property IOSTANDARD LVCMOS33 [get\_ports btnL]  #set\_property PACKAGE\_PIN T17 [get\_ports btnR]  #set\_property IOSTANDARD LVCMOS33 [get\_ports btnR]  #set\_property PACKAGE\_PIN U17 [get\_ports btnD]  #set\_property IOSTANDARD LVCMOS33 [get\_ports btnD]  ##Pmod Header JA  ##Sch name = JA1  #set\_property PACKAGE\_PIN J1 [get\_ports {JA[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[0]}] |

|  |  |
| --- | --- |
| ##Sch name = JA2  #set\_property PACKAGE\_PIN L2 [get\_ports {JA[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[1]}]  ##Sch name = JA3  #set\_property PACKAGE\_PIN J2 [get\_ports {JA[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[2]}]  ##Sch name = JA4  #set\_property PACKAGE\_PIN G2 [get\_ports {JA[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[3]}]  ##Sch name = JA7  #set\_property PACKAGE\_PIN H1 [get\_ports {JA[4]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[4]}]  ##Sch name = JA8  #set\_property PACKAGE\_PIN K2 [get\_ports {JA[5]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[5]}]  ##Sch name = JA9  #set\_property PACKAGE\_PIN H2 [get\_ports {JA[6]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[6]}]  ##Sch name = JA10  #set\_property PACKAGE\_PIN G3 [get\_ports {JA[7]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[7]}]  ##Pmod Header JB  ##Sch name = JB1  #set\_property PACKAGE\_PIN A14 [get\_ports {JB[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[0]}]  ##Sch name = JB2  #set\_property PACKAGE\_PIN A16 [get\_ports {JB[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[1]}]  ##Sch name = JB3  #set\_property PACKAGE\_PIN B15 [get\_ports {JB[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[2]}]  ##Sch name = JB4 | #set\_property PACKAGE\_PIN B16 [get\_ports {JB[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[3]}]  ##Sch name = JB7  #set\_property PACKAGE\_PIN A15 [get\_ports {JB[4]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[4]}]  ##Sch name = JB8  #set\_property PACKAGE\_PIN A17 [get\_ports {JB[5]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[5]}]  ##Sch name = JB9  #set\_property PACKAGE\_PIN C15 [get\_ports {JB[6]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[6]}]  ##Sch name = JB10  #set\_property PACKAGE\_PIN C16 [get\_ports {JB[7]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[7]}]  ##Pmod Header JC  ##Sch name = JC1  #set\_property PACKAGE\_PIN K17 [get\_ports {JC[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[0]}]  ##Sch name = JC2  #set\_property PACKAGE\_PIN M18 [get\_ports {JC[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[1]}]  ##Sch name = JC3  #set\_property PACKAGE\_PIN N17 [get\_ports {JC[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[2]}]  ##Sch name = JC4  #set\_property PACKAGE\_PIN P18 [get\_ports {JC[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[3]}]  ##Sch name = JC7  #set\_property PACKAGE\_PIN L17 [get\_ports {JC[4]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[4]}]  ##Sch name = JC8  #set\_property PACKAGE\_PIN M19 [get\_ports {JC[5]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[5]}] |

|  |  |
| --- | --- |
| ##Sch name = JC9  #set\_property PACKAGE\_PIN P17 [get\_ports {JC[6]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[6]}]  ##Sch name = JC10  #set\_property PACKAGE\_PIN R18 [get\_ports {JC[7]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[7]}]  ##Pmod Header JXADC  ##Sch name = XA1\_P  #set\_property PACKAGE\_PIN J3 [get\_ports {JXADC[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[0]}]  ##Sch name = XA2\_P  #set\_property PACKAGE\_PIN L3 [get\_ports {JXADC[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[1]}]  ##Sch name = XA3\_P  #set\_property PACKAGE\_PIN M2 [get\_ports {JXADC[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[2]}]  ##Sch name = XA4\_P  #set\_property PACKAGE\_PIN N2 [get\_ports {JXADC[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[3]}]  ##Sch name = XA1\_N  #set\_property PACKAGE\_PIN K3 [get\_ports {JXADC[4]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[4]}]  ##Sch name = XA2\_N  #set\_property PACKAGE\_PIN M3 [get\_ports {JXADC[5]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[5]}]  ##Sch name = XA3\_N  #set\_property PACKAGE\_PIN M1 [get\_ports {JXADC[6]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[6]}] | ##Sch name = XA4\_N  #set\_property PACKAGE\_PIN N1 [get\_ports {JXADC[7]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[7]}]  ##VGA Connector  #set\_property PACKAGE\_PIN G19 [get\_ports {vgaRed[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[0]}]  #set\_property PACKAGE\_PIN H19 [get\_ports {vgaRed[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[1]}]  #set\_property PACKAGE\_PIN J19 [get\_ports {vgaRed[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[2]}]  #set\_property PACKAGE\_PIN N19 [get\_ports {vgaRed[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[3]}]  #set\_property PACKAGE\_PIN N18 [get\_ports {vgaBlue[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[0]}]  #set\_property PACKAGE\_PIN L18 [get\_ports {vgaBlue[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[1]}]  #set\_property PACKAGE\_PIN K18 [get\_ports {vgaBlue[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[2]}]  #set\_property PACKAGE\_PIN J18 [get\_ports {vgaBlue[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[3]}]  #set\_property PACKAGE\_PIN J17 [get\_ports {vgaGreen[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[0]}]  #set\_property PACKAGE\_PIN H17 [get\_ports {vgaGreen[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[1]}]  #set\_property PACKAGE\_PIN G17 [get\_ports {vgaGreen[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[2]}] |

|  |  |
| --- | --- |
| #set\_property PACKAGE\_PIN D17 [get\_ports {vgaGreen[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[3]}]  #set\_property PACKAGE\_PIN P19 [get\_ports Hsync]  #set\_property IOSTANDARD LVCMOS33 [get\_ports Hsync]  #set\_property PACKAGE\_PIN R19 [get\_ports Vsync]  #set\_property IOSTANDARD LVCMOS33 [get\_ports Vsync]  ##USB-RS232 Interface  #set\_property PACKAGE\_PIN B18 [get\_ports RsRx]  #set\_property IOSTANDARD LVCMOS33 [get\_ports RsRx]  #set\_property PACKAGE\_PIN A18 [get\_ports RsTx]  #set\_property IOSTANDARD LVCMOS33 [get\_ports RsTx]  ##USB HID (PS/2)  #set\_property PACKAGE\_PIN C17 [get\_ports PS2Clk]  #set\_property IOSTANDARD LVCMOS33 [get\_ports PS2Clk]  #set\_property PULLUP true [get\_ports PS2Clk]  #set\_property PACKAGE\_PIN B17 [get\_ports PS2Data]  #set\_property IOSTANDARD LVCMOS33 [get\_ports PS2Data]  #set\_property PULLUP true [get\_ports PS2Data]  ##Quad SPI Flash  ##Note that CCLK\_0 cannot be placed in 7 series devices. You can access it using the  ##STARTUPE2 primitive.  #set\_property PACKAGE\_PIN D18 [get\_ports {QspiDB[0]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[0]}]  #set\_property PACKAGE\_PIN D19 [get\_ports {QspiDB[1]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[1]}]  #set\_property PACKAGE\_PIN G18 [get\_ports {QspiDB[2]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[2]}] | #set\_property PACKAGE\_PIN F18 [get\_ports {QspiDB[3]}]  #set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[3]}]  #set\_property PACKAGE\_PIN K19 [get\_ports QspiCSn]  #set\_property IOSTANDARD LVCMOS33 [get\_ports QspiCSn]  ## Configuration options, can be used for all designs  set\_property CONFIG\_VOLTAGE 3.3 [current\_design]  set\_property CFGBVS VCCO [current\_design]  Conclusion:  The laboratory was very challenging, but at the same time interesting and full of surprises. I did really enjoy seeing the 7-segment digit display transformed into a clock. It took a lot of effort to make it work, but at the end it was worth it.  An appendix section was included to put the screenshots of the testbench simulations. |

**BCD to Seven Segment Testbench Simulation**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Digit Controller Testbench Simulation**

**Graphical user interface, application

Description automatically generated**

**BCD Counter Testbench Simulation**

A screenshot of a computer

Description automatically generated

Graphical user interface, application

Description automatically generated