CSE 209:   LOGIC DESIGN LAB

III SEM B.E.(CS &E)

(2013)

Prepared by	Approved by
(Hemalatha S)	(H.O.D)

DEPT. OF COMPUTER SCIENCE & ENGG.

M.I.T., MANIPAL

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.

2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).

3. Students should bring the observation book, and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.

4. Once the experiment(s) get executed, they should show the results to the instructors and copy the same in their observation book.

5. Their observation book should be complete with proper logical diagrams, truth tables and any other information related to the experiment they perform.

6. Questions for lab exam need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

PROCEDURE FOR EVALUATION

The entire lab course consists of 100 marks. The marking scheme is as follows

Continuous Evaluation	60 marks
End Sem Lab Exam	40 marks
Total	100 marks

CONTENTS

WEEK 1

Introduction to MAXPlus II, Verification of logic gates and Boolean algebra.

1. Verification of logic gates

Write VHDL code to implement the following and simulate.

i) Two input AND, OR, EX-OR gates.

ii) Three input NAND using only 2 input NAND gates

iii) Three input NOR gates using only 2 input NOR gates

2. Verification of Boolean algebra

Prove the following

X (Y + Z) = XY + XZ; X + YZ = (X + Y) (X + Z); (A’)’ = A;

(if both LHS and RHS contain expressions write VHDL code for both sides and show that LHS=RHS)

WEEK 2

Application of Boolean algebra

1. Write VHDL code to simulate

   i) F=AB′+A′C+BC

   ii) F=AB′+CDB+A′CD

   iii) (A + B + C + ……………)′ = $\overline{A}$ • $\overline{B}$ • $\overline{C}$ • …………..

   iv) (A • B • C • …………….)′ = $\overline{A}$ + B + C + ………..

2. Given the logical expression Y=(A+BC)(B+CA). Convert it to SOP and simulate using only

   i) AND and OR gates. ii) NAND gates

3. Simulate F(x,y,z) = ∑(2,3,4,6,7) using

   i) NOR gates only ii) NAND gates only

WEEK 3

Simplification of expressions using K-map.

1. Write VHDL code to implement the function f(A,B,C,D) = ∑ (0,2,4,5,6,7,8,10,13,15)
2. Write VHDL code to implement the function f(A,B,C,D) = ∑ (0,6,8,13,14) + d(2,4,10)
3. Write VHDL code to implement the function f(A,B,C,D) = ∏(0,2,3,7,8,9,10,11,13)

4. Simulate a circuit, which has 3 inputs and one output. The output is high when majorities of the inputs are high.

WEEK 4

Multilevel NAND and NOR circuits

1. Minimize the following expressions using K-Map and simulate using NOR gates only.

   f1(A,B,C,D) = ∑ (0,1,2,3,4,8,9,12)

   f2(A,B,C,D) = ∑ (0,1,2,3,7,8,10) + d(5,6,11,15)

2 Minimize the following expression using K-Maps and simulate using NAND gates only.

   f1(A,B,C,D) = ∏(1,3,5,8,9,11,15)+d(2,13)

   f2(A,B,C,D)= ∏ (1,3,5,7,13,15)

3. Apply functional decomposition for the following function to obtain a circuit and write and simulate VHDL code to implement it.

   F(a.b.c.d) = a'bc+ab'c+abd+a'b'd

WEEK 5

Arithmetic Circuits

Write VHDL code to implement the following and simulate (behavioral and mixed styles)

1. Half adder and a full adder
2. Four bit adder
3. Four bit adder/subtractor
4. 2 Bit Multiplier
5. Single digit BCD adder

WEEK 6

Multiplexers

1. Write behavioral VHDL code for 8 to 1 multiplexer. Use case statement.

2. Write behavioral VHDL code for 4 to 1 multiplexer. Use With …. Select…. Statement. Use this to write hierarchical code for 16 to 1 multiplexer.

3. Write behavioral VHDL code for 4 to 1 multiplexer. Use if then else statement.

4. Write behavioral VHDL code for 4 to 1 multiplexer. Use conditional signal assignment statement.

5. Write behavioral VHDL code for 16 to 1 multiplexer using generate statement.

WEEK 7

Multiplexer Applications

1. Design a binary to gray code converter using 8:1 MUXes. Write VHDL code to implement this.

2. Realize the following using MUXes and implement in VHDL.

   i) f1 (A,B,C,D)= ∑ (1,3,5,6,9,11) ii) f2 (A,B,C)= ∑ (0,2,3,6)

3. Realize the following using MUXes and implement in VHDL.

   i) f1 (A,B,C) = B’ + A’C’ + AC ii) f2 (A,B,C) = A’B’ + B’C’ + ABC

4. Using a 4 to 1 multiplexer design a full adder and simulate it by writing VHDL code

5. Design a BCD to 2421 code converter using multiplexer. Write VHDL code to implement this.

WEEK 8

Decoders and encoders

1. Write behavioral VHDL code for 3 to 8 decoder with active high enable input and active low output. Use With …. Select…. Statement.

2. Write behavioral VHDL code for 8 to 3 priority encoder. Use conditional signal assignment statement.

3. Write behavioral VHDL code for 8 to 3 priority encoder. Using if then else statement.

4. Design 4 to 16 decoder using two 3 to 8 decoders. Write VHDL code to implement this.

5. Construct a decoder tree to implement 5 to 32 decoder using four 3 to 8 decoders with enable and one 2 to 4 decoder. Write VHDL code to implement this.

WEEK 9

Applications of Decoders and encoders

1. Write VHDL code to implement f (A, B, C) = ∑ (0, 2, 3, 4, 5, 7) using 3 to 8 binary decoder and an OR gate. Use selected signal assignment statement

2. A combinational circuit is specified by the following three Functions.

   F1= X’Y’Z’ + XZ, F2=XY’Z’ + X’Y, F3=X’Y’Z + XY

   Design the circuit with a decoder and external gates. Write VHDL code to implement 3 to 8 binary decoder using if then else statement. Use this entity to implement the above circuit

3. A combinational circuit is specified by the following three Functions.

   F1 (A,B,C)= ∑ (2,4,7) F2 (A,B,C)= ∑ (0,3) F3 (A,B,C)= ∑ (0,2,3,4,7)

   Design the circuit with a decoder and external gates. Use the 3 to 8 decoder entity of the previous question to write VHDL code for this circuit and simulate the same.

4. Design and implement full adder using decoder and other gates. Write VHDL code to implement 3 to 8 binary decoder using case statement. Use this entity to implement full adder.

5. Implement 4 to 1 multiplexer using 2 to 4 decoder and external gates.

WEEK 10

Comparators and code converters

1. Write VHDL code and simulate 1 bit equality comparator.

2. Write VHDL code and simulate 4 bit comparator.

3. Write behavioral VHDL code to convert an N bit binary number into an equivalent grey code. Use for loop.

4. Write behavioral VHDL code to convert an N bit grey code into an equivalent binary code. Use for loop.

5. Write VHDL code to convert BCD to excess-3 code.

WEEK 11

FF's, and Registers

1. Write behavioral VHDL code for negative edge triggered D FF with asynchronous reset

2. Write behavioral VHDL code for negative edge triggered T FF with synchronous reset

3. Write behavioral VHDL code for positive edge triggered JK FF with reset facility

4. Write VHDL code for 8 bit register

5. Write VHDL code for N bit register

6. Write VHDL code for 4 bit shift register

7. Write VHDL code for N bit shift register

WEEK 12

Counters

1. Write VHDL code for ring counter
2. Write VHDL code for johnson counter
3. Write VHDL code for the following counters

    i) 4 bit asynchronous up counter

    ii) 4 bit synchronous up counter

    iii) 4 bit synchronous up/down counter with a control input up/$\overline{\text{down}}$. If up/$\overline{\text{down}}$ = 1, then the circuit should behave as an up counter. If up/$\overline{\text{down}}$ = 0, then the circuit should behave as down counter.

    iv) 4 bit asynchronous up/down counter with a control input up/down. If up/$\overline{\text{down}}$ = 1, then the circuit should behave as an up counter. If up/$\overline{\text{down}}$ = 0, then the circuit should behave as down counter.

WEEK 13 & 14

TEST

References:

1. Stephen Brown and Zvonko Vranesic, "Fundamentals of digital logic with VHDL design", Tata MGH 2000.
2. J. Bhasker, "A VHDL Primer", PHI Pvt. Ltd., 3rd Edition, 2005.