CS M51A and EE M16 Spring 2015 Section 1

Logic Design of Digital Systems

Dr. Yutao He

Verilog Lab #2 - Design of Combinational Systems

Due: May 13rd, 2015

(1) Name:

Yu Dau-Po

Last First

Student ID: 504451468

Signature: Dau-Po Yu

(2) Name:

Lin Jennifer



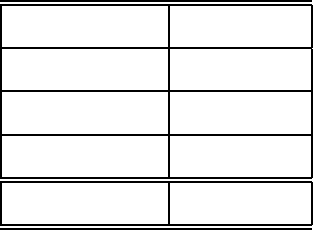
Last First

Student ID: 904299916

Signature: Jennifer Lin

Date: 5/13/2015

Result Correctness Creativity Report

Total Score

Verilog Lab #2 Project Requirement

Dr. Yutao He

Due: 5/13/2015

1 Objectives

The second project is to build a combinational circuit for a real-world appli- cation by applying systematically the concepts and techniques covered in the class, and walking through the whole cycle of designing digital systems, from the high-level specification to the final implementation. It basically consists of two steps:(1) the pencil-and-paper design, and (2) the implementation with Verilog by means of the CAD software Vivado.

2 Project Description

2.1 The High-Level Specification

The circuit to be built is a BCD-to-seven-segment display decoder that is used extensively to drive LED (Light-Emitting Diode)-based display devices such as billboards.

The structure of a seven-segment LED device and its interface with the decoder to be designed are shown in Figure 1. It is easily seen that any decimal number from zero to nine can be displayed with this device simply

by turning some segments on (shaded), while leaving others oﬀ (unshaded).

**a**

**f b**

**g 1**

**0**

**1 e c 0**

**d**

**b 0**

**a 1 a**

**1**

**0 f f b**

**1 1 g**

**BCD-to-7-Segment**

**Display Decoder**

**g**

**2**

**3 e 0 e c d 1**

**c 1 d**

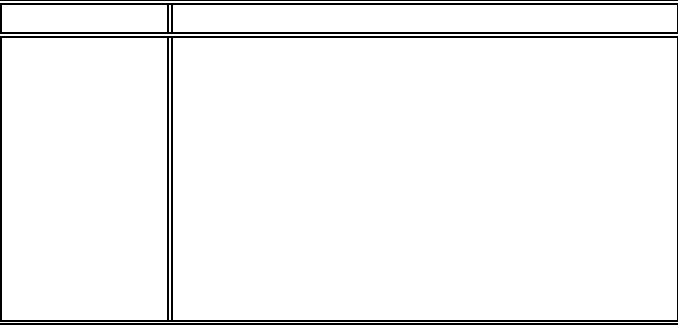
**unshaded shaded**

**The circuit to be designed**

**Display of the digit 5**

Figure 1: The Structure of a Seven-Segment Display Device

The complete display patterns for one-digit decimal numbers are speci- fied in Table 1.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal No.  0 | a  ON | b  ON | c  ON | d  ON | e  ON | f  ON | g  OFF |
| 1 | OFF | ON | ON | OFF | OFF | OFF | OFF |
| 2 | ON | ON | OFF | ON | ON | OFF | ON |
| 3 | ON | ON | ON | ON | OFF | OFF | ON |
| 4 | OFF | ON | ON | OFF | OFF | ON | ON |
| 5 | ON | OFF | ON | ON | OFF | ON | ON |
| 6 | ON | OFF | ON | ON | ON | ON | ON |
| 7 | ON | ON | ON | OFF | OFF | OFF | OFF |
| 8 | ON | ON | ON | ON | ON | ON | ON |
| 9 | ON | ON | ON | ON | OFF | ON | ON |

Table 1: The Seven-Segment Display Logic Specification

Given the above specification, in this project you are to build a circuit which takes a decimal digit, decodes it according to the specified logic, and generates the corresponding outputs. In addition, your circuit must also meet the following requirements:

1. Encoding scheme of inputs: You must use the BCD (Binary- Coded-Decimal) code to encode one decimal digit as specified in Ta- ble 2:

2. Encoding scheme of outputs: The on/oﬀ encoding scheme of out- puts is specified in Table 3.

3. Implementation: The system must be implemented with a two-level

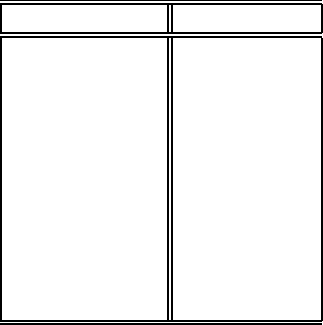
NAND-NAND network.

4. Cost: Let us define the cost of the system as the number of NAND

gates in the circuit. The system must be designed with minimal cost.

3 Report Outline

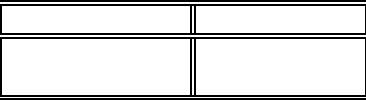
Each team is required to submit only one report that provides complete documentation of your project including the detailed design worksheets. As in all technical writing, its purpose is to communicate your work with your



|  |  |
| --- | --- |
| Decimal No.  0 | BCD Code  0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

Table 2: The Input Encoding Scheme

Segment State Binary Code

ON 1

OFF 0

Table 3: The Output Encoding Scheme

colleagues in an eﬃcient and professional way so that your design can be continuously upgraded and maintained even if you are no longer around. As a result, the report should be clear, concise, and complete and should contain the following parts:

(1) Title Page

It is provided and you just need to fill in your information in the blanks.

(2) Abstract

This is the brief high-level description of the project. (3) The Switching Functions of the Circuit

It is part of the design work for you to obtain the binary-level specifica-

tion for the function of the circuit in the form of switching expressions and the schematic. This section should present both minimal switching expressions in NAND-NAND form and the schematic of the circuit.

(4) The Verilog Code of the Circuit

The Verilog file (with extension .v) you write is the implementation of the circuit. You should include its printout in your report with your names and student IDs on it.

(5) The Simulation Result

You have to demonstrate that your implementation behaves as speci- fied by showing the correct simulation result. In particular, the result (in the forms of Timing Diagram) should include the displays for a complete set of decimal numbers (from zero to nine). Include in your report the Timing Diagram with enough information on it so that one of your colleagues who does not know anything about your project could understand easily which function you are trying to implement by just reading the Timing Diagram.

(6) The Design Review

This section summarizes your experiences and lessons throughout the project. It should be no more than one page and may include topics such as what you have learned, problem encountered during the im- plementation and the solutions you came up with, the approach you used, the most important aspects of the project for you, where you spent most of your time, and suggestions you would like to make.

(7) Team Member Contributions

Teamwork requires that each member be responsible and assume an relatively equal share of workload. In this section, a detailed descrip- tion on each member’s responsibility and contribution should be pre-

sented clearly, including an estimate of percentage of eﬀorts on the

project and a summary list of each member in the project.

Each member requires to review and sign the final report on the cover page.

(8) Appendix - The Detailed Design Worksheet

This section must include the complete set of worksheets from the pencil-and-paper design. Please scan and combine them with other parts of your report. Hand-written form is perfectly fine yet it must include:

8.1 inputs and outputs of the system.

8.2 encoding schemes of inputs and output.

8.3 truth tables.

8.4 minimization procedure by means of either K-map or Quine- McCluskey method.

8.5 transformation procedure.

8.6 final minimal expressions of the output functions in terms of both switching expressions and the schematic.

4 Project Submission

You should submit one zipped file named with ”Txx.zip” via the on-line submission link that will be avalable by 5/10. The zipped file should consist of three separate files:

1. The pdf file of your report. It must be named as ”Txx.pdf” where xx is your Team ID assigned to you, that is, your report should be called Txx.pdf;

2. The Verilog file of your circuit implementation. It should be named as csm51a proj2.v.

3. The testbench file. It should be named as csm51a proj2 tb.v.

5 Project Deadline

The report is due by midnight (11:59:59pm) on May 13rd (Wednesday),

2015. The deadline must be observed strictly and late submissions will be subject to penalty.

**Abstract**

For this project, the circuit we implemented is a BCD-to-seven-segment display decoder that could display numbers zero through nine by lightening up certain segments of the device. The circuit takes in four inputs that encode decimal numbers zero through nine using BCD code, and outputs a through g uses the on/off encoding scheme specified in Table 3. The system is implemented with a two-level NAND-NAND network.

**Switching Functions**

Switching functions of outputs a to g:

a= [x3’x1’(x2x0)’(x2’x0’)’]’

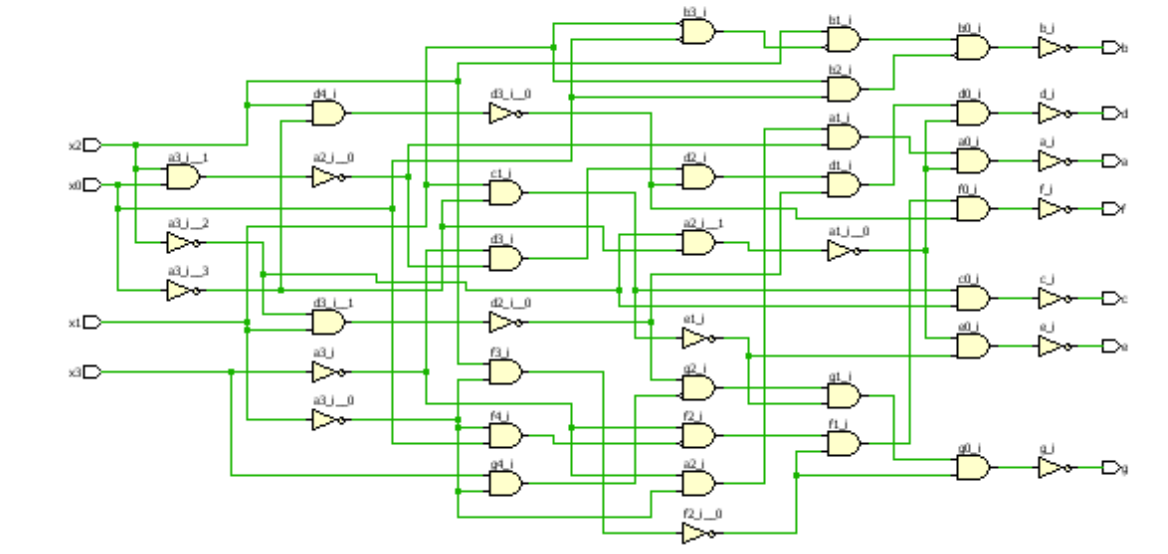
b= [x2(x1’x0’)’(x1x0)’]’

c= (x2’x1x0’)’

d= [x3’(x2x1’x0)’(x2x1x0’)’(x2’x1)’(x2’x0’)’]’

e= [(x2’x0’)’(x1x0’)’]’

f= [x3’(x1’x0’)’(x2x1’)’(x2x0’)’]’

g= [(x3x1’)’(x2’x1)’(x2x0’)’(x2x1’)’]’

**Verilog Code**

module csm51a\_proj2(x3, x2, x1, x0, a, b, c, d, e, f, g);

input x3, x2, x1, x0;

output a, b, c, d, e, f, g;

assign a= ~( (~x3) & (~x1) & ~(x2 & x0) & ~(~x2 & ~x0) );

assign b= ~( (x2) & ~(~x1 & ~x0)& ~(x1 & x0) );

assign c= ~( (x1) & (~x0) & (~x2) );

assign d= ~( (~x3) & ~(x2 & ~x1 & x0) & ~(x2 & x1 & ~x0) & ~(~x2 & x1) & ~(~x2 & ~x0) );

assign e= ~(~(~x2 & ~x0) & ~(x1 & ~x0));

assign f= ~( (~x3) & ~(~x1 & ~x0) & ~(x2 & ~x1) & ~(x2 & ~x0));

assign g= ~(~(x3 & ~x1) & ~(~x2 & x1) & ~(x1 & ~x0) & ~(x2 & ~x1));

endmodule

**Testbench**

module csm51a\_proj2\_tb;

wire t\_1, t\_2, t\_3, t\_4, t\_5, t\_6, t\_7;

reg t\_a, t\_b, t\_c, t\_d;

csm51a\_proj2 my\_gate( .x3(t\_a), .x2(t\_b), .x1(t\_c), .x0(t\_d), .a(t\_1), .b(t\_2), .c(t\_3), .d(t\_4), .e(t\_5), .f(t\_6), .g(t\_7));

initial

begin

$monitor(t\_a, t\_b, t\_c, t\_d, t\_1, t\_2, t\_3, t\_4, t\_5, t\_6, t\_7);

t\_a = 1'b0;

t\_b = 1'b0;

t\_c = 1'b0;

t\_d = 1'b0;

#5

t\_a = 1'b0;

t\_b = 1'b0;

t\_c = 1'b0;

t\_d = 1'b1;

#5

t\_a = 1'b0;

t\_b = 1'b0;

t\_c = 1'b1;

t\_d = 1'b0;

#5

t\_a = 1'b0;

t\_b = 1'b0;

t\_c = 1'b1;

t\_d = 1'b1;

#5

t\_a = 1'b0;

t\_b = 1'b1;

t\_c = 1'b0;

t\_d = 1'b0;

#5

t\_a = 1'b0;

t\_b = 1'b1;

t\_c = 1'b0;

t\_d = 1'b1;

#5

t\_a = 1'b0;

t\_b = 1'b1;

t\_c = 1'b1;

t\_d = 1'b0;

#5

t\_a = 1'b0;

t\_b = 1'b1;

t\_c = 1'b1;

t\_d = 1'b1;

#5

t\_a = 1'b1;

t\_b = 1'b0;

t\_c = 1'b0;

t\_d = 1'b0;

#5

t\_a = 1'b1;

t\_b = 1'b0;

t\_c = 1'b0;

t\_d = 1'b1;

#5

t\_a = 1'b1;

t\_b = 1'b0;

t\_c = 1'b1;

t\_d = 1'b0;

#5

t\_a = 1'b1;

t\_b = 1'b0;

t\_c = 1'b1;

t\_d = 1'b1;

#5

t\_a = 1'b1;

t\_b = 1'b1;

t\_c = 1'b0;

t\_d = 1'b0;

#5

t\_a = 1'b1;

t\_b = 1'b1;

t\_c = 1'b0;

t\_d = 1'b1;

#5

t\_a = 1'b1;

t\_b = 1'b1;

t\_c = 1'b1;

t\_d = 1'b0;

#5

t\_a = 1'b1;

t\_b = 1'b1;

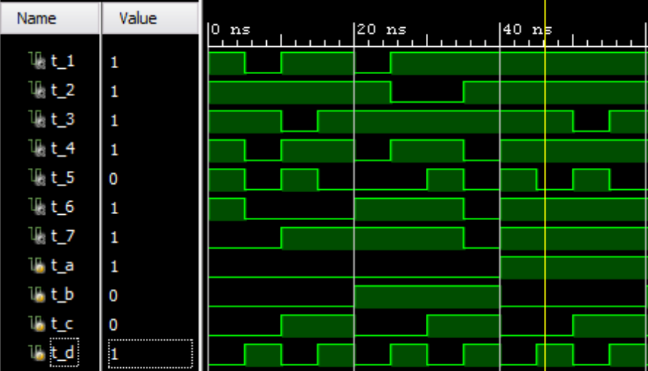
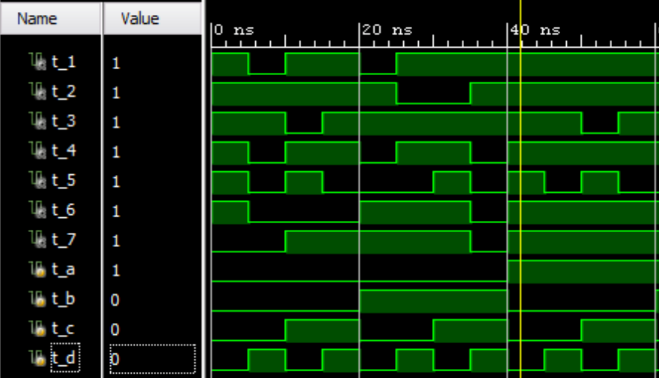
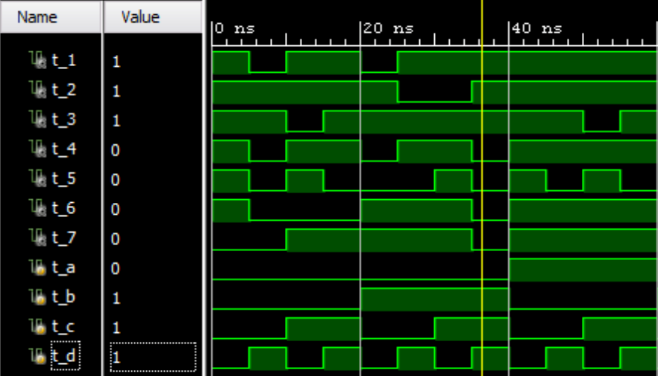
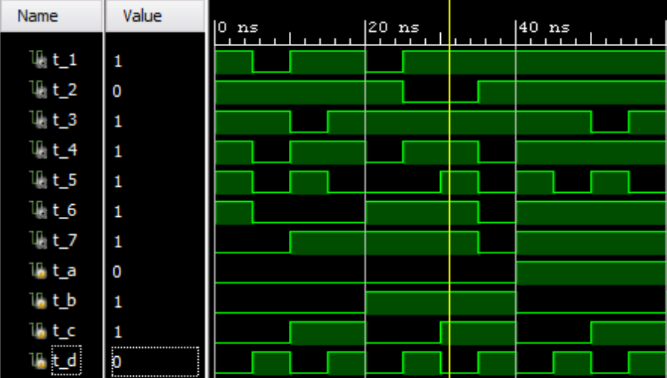
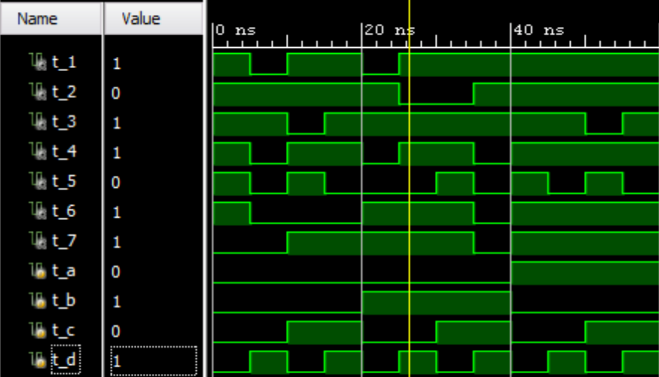
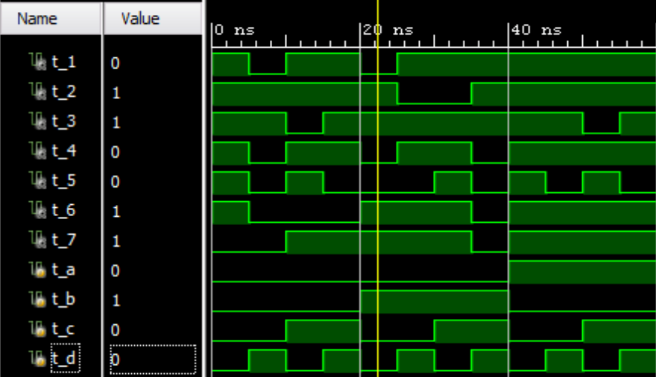
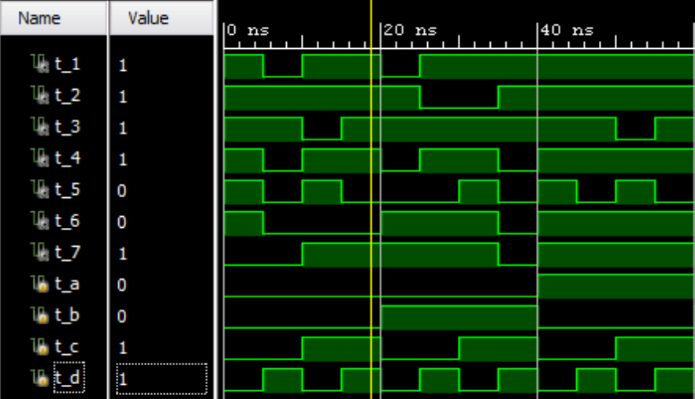
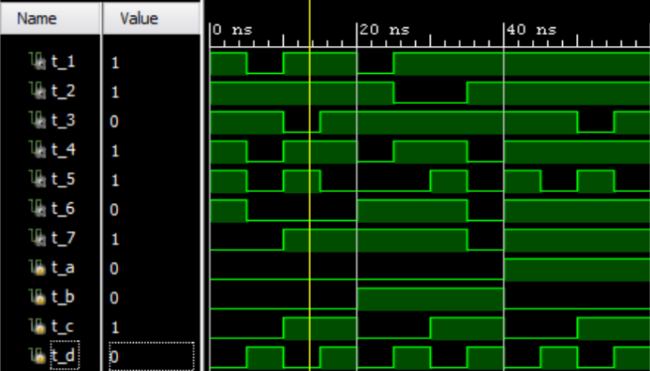
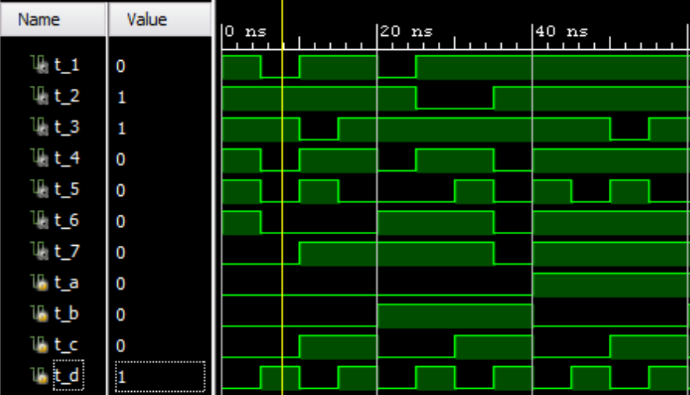
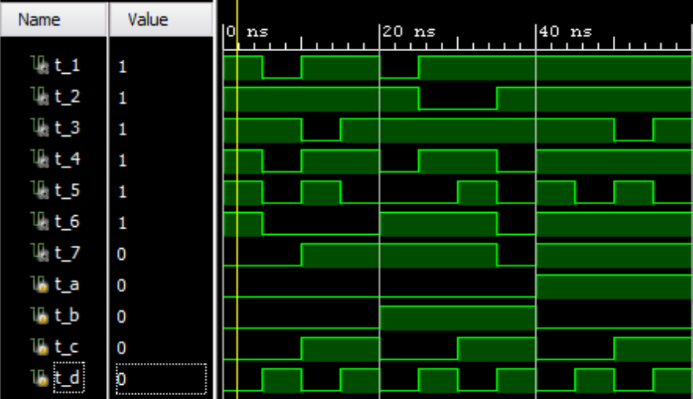
t\_c = 1'b1;

t\_d = 1'b1;

end

endmodule

**Simulation Result**



**Design Review**

This project enlightens me how to solve a real-world question. Before this project, I would not know how to translate and apply what we learn in class into real-world application. However, after this assignment, I understand how to approach a question in a different perspective, and further solve it with what we learn in class.

During this project, we spent some time writing, checking, and double-checking the minimum minterm expression. This step is extremely important, since we are required to minimize the cost, and the process after it would be in vain if this step is wrong. Additionally, testbench may not detect the errors here, so we were really careful about this step. Afterwards, we went smoothly in implementing and stimulating the modules.

**Team Member Contribution**

The workload of this project is equally shared between us. Rather than dividing and assigning who should do what part of this project, both of us did most of the work and checked on each other’s work to make sure that we’re on the right track.

**Appendix**

