ECE426/516 Lab 2

Professor: Chao Lu
Email: chaolu@siu.edu

ECE 426/516 Implementation of VLSI Systems with HDL
Lab 2 3-8 Decoder

Lab Description

In this lab section, you will design a BCD to decimal decoder (3-8 decoder) in both structural and behavioral Verilog codes, synthesis your designs and run simulations.

Lab Introduction

Table 1 is the truth table for BCD to decimal decoder

Table 1 Gray Code for each Binary Code Word

Enable	BCD Inputs			Outputs							
en	i_2	i_1	i_0	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1
0	X	X	X	0	0	0	0	0	0	0	0

Figure 1 is the block diagram of a 3:8 decoder with an enable pin.

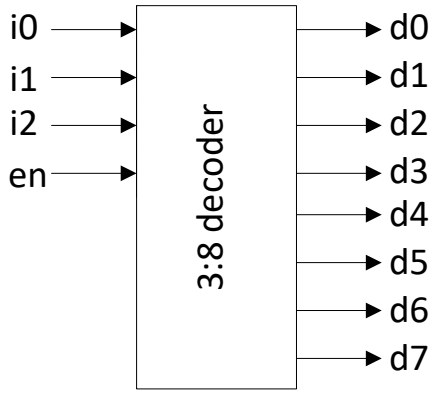


Figure 1 Block diagram of a 3:8 decoder

Lab Procedure

Please read and follow the underneath instructions to finish your lab.

1. Structural design of a 3:8 decoder with an enable signal (en), by instantiating AND, OR, and INV gates which are Verilog primitive gates. You should not use an always statement in your structural design. Use the following module template for your decoder design.

```
module decoder_3_8_structural (en, i, d);
input en;
input [2:0] i;
output [7:0] d;
wire [2:0] i_;

// start your code here

endmodule
```

2. Test your structural design by exhaustively run simulation of all possible input combinations (16 combinations in total).

3. Behaviorally design the same 3:8 decoder using a single always procedure. Hint: you can use an if-then-else statement to implement the enable functionality. Use the following module template for your decoder design.

```
module decoder_3_8_behavioral(en, i, d);

input [2:0] i;
input en;
output [7:0] d;
reg [7:0] d;

// start your code here

endmodule
```

4. Test your behavioral design by exhaustively simulating all possible input combinations (16 combinations in total). Ensure that your behavioral design behaves the same as your structural design.

We strongly recommend coding by yourself, although the code below is a reference.

Lab Guide

1. Create a new directory named ‘lab2’, using ‘mkdir’ command like this:

```
/home/grad/siu854024310/ECE426-45> mkdir lab2
```

Change your working directory to “lab2”. You can download 5 reference codes from D2L and upload them into this lab2 folder.

```
/home/staff/eeluchao/ECE426/lab2-53> ls
decoder_3_8_behavioral.v  run_dc_behavioral.tcl  testbench.v
decoder_3_8_structural.v  run_dc_structural.tcl
/home/staff/eeluchao/ECE426/lab2-54>
```

2. The structural code (decoder_3_8_structural.v) uses only primitive gates, and the behavioral code (decoder_3_8_behavioral.v) uses case-statements.

```
module decoder_3_8_structural (en, i, d);
input en;
input [2:0] i;
output [7:0] d;
wire [2:0] i_;
not N0 (i_[0], i[0]);
not N1 (i_[1], i[1]);
not N2 (i_[2], i[2]);
and A0 (d[0], en, i_[2], i_[1], i_[0]);
and A1 (d[1], en, i_[2], i_[1], i[0]);
and A2 (d[2], en, i_[2], i[1], i_[0]);
and A3 (d[3], en, i_[2], i[1], i[0]);
and A4 (d[4], en, i[2], i_[1], i_[0]);
and A5 (d[5], en, i[2], i_[1], i[0]);
and A6 (d[6], en, i[2], i[1], i_[0]);
and A7 (d[7], en, i[2], i[1], i[0]);
endmodule
```

```
module decoder_3_8_behavioral(en, i, d);

input [2:0] i;
input en;
output [7:0] d;
reg [7:0] d;
always @ (en or i)

begin
case({en, i})
4'b0XXX: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end
4'b1000: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 1; end
4'b1001: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 1; d[0] = 0; end
4'b1010: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 1; d[1] = 0; d[0] = 0; end
4'b1011: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 1; d[2] = 0; d[1] = 0; d[0] = 0; end
```

```
4'b1100: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 1; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end
4'b1101: begin d[7] = 0; d[6] = 0; d[5] = 1; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end
4'b1110: begin d[7] = 0; d[6] = 1; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end
4'b1111: begin d[7] = 1; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end

default: begin d[7] = 0; d[6] = 0; d[5] = 0; d[4] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0; end

endcase
end
endmodule
```

3. Write testbench.v to run simulation.

```
`include "decoder_3_8_structural.v"
`include "decoder_3_8_behavioral.v"

module testbench();

reg [2:0] inputs;
reg enable;
wire [7:0] o_structural;
wire [7:0] o_behavioral;

initial

begin: apply_stimulus

reg [4:0] invect;

for (invect = 0; invect <= 16; invect = invect + 1)
  begin
        {enable, inputs[2], inputs[1], inputs[0]} = invect[4:0];
        #10 $monitor ("#structural# enable = %b, inputs inputs[2:0]=%b, outputs
d[7:0]=%b", enable, inputs[2:0], o_structural[7:0]);
        #10 $monitor ("#behavioral# enable = %b, inputs inputs[2:0]=%b, outputs
d[7:0]=%b", enable, inputs[2:0], o_behavioral[7:0]);
   end
end

decoder_3_8_structural inst_s (
.en(enable),
.i(inputs),
.d(o_structural)
);

decoder_3_8_behavioral inst_b (
.en(enable),
.i(inputs),
.d(o_behavioral)
);

endmodule
```

4. Type command ‘synopsys’ to enter Synopsys shell. This may take a few minutes.

To check if you are in the synopsys shell environment, you can check using command ‘which dc_shell’, if the return of such command popped up is like below, then you have successfully entered synopsys shell.

```
/home/staff/eeluchao/ECE426/lab2-55> synopsys
/home/staff/eeluchao/ECE426/lab2-56> which dc_shell
/synopsys/Design_Compiler/syn/S-2021.06-SP5-4/bin/dc_shell
/home/staff/eeluchao/ECE426/lab2-57>
```

Vlogan the design: type “vlogan decoder_3_8_behavioral.v”

```
/home/staff/eeluchao/ECE426/lab2-58> vlogan decoder_3_8_behavioral.v
                        Chronologic VCS (TM)
          Version T-2022.06-SP1 -- Wed Jan 25 18:57:05 2023

                    Copyright (c) 1991 - 2022 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                    infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
            Inclusivity and Diversity" (Refer to article 000036315 at
                        https://solvnetplus.synopsys.com)

Parsing design file 'decoder_3_8_behavioral.v'
CPU time: .427 seconds to compile
/home/staff/eeluchao/ECE426/lab2-59>
```

Vlogan the design: type “vlogan decoder_3_8_structural.v”

```
/home/staff/eeluchao/ECE426/lab2-59> vlogan decoder_3_8_structural.v
                        Chronologic VCS (TM)
          Version T-2022.06-SP1 -- Wed Jan 25 18:57:37 2023

                    Copyright (c) 1991 - 2022 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                    infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
            Inclusivity and Diversity" (Refer to article 000036315 at
                        https://solvnetplus.synopsys.com)

Parsing design file 'decoder_3_8_structural.v'
CPU time: .142 seconds to compile
/home/staff/eeluchao/ECE426/lab2-60>
```

Vlogan the testbench: type “vlogan testbench.v”

```
/home/staff/eeluchao/ECE426/lab2-60> vlogan testbench.v
                        Chronologic VCS (TM)
          Version T-2022.06-SP1 -- Wed Jan 25 18:58:53 2023

                    Copyright (c) 1991 - 2022 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                    infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
            Inclusivity and Diversity" (Refer to article 000036315 at
                        https://solvnetplus.synopsys.com)

Parsing design file 'testbench.v'
CPU time: .125 seconds to compile
/home/staff/eeluchao/ECE426/lab2-61>
```

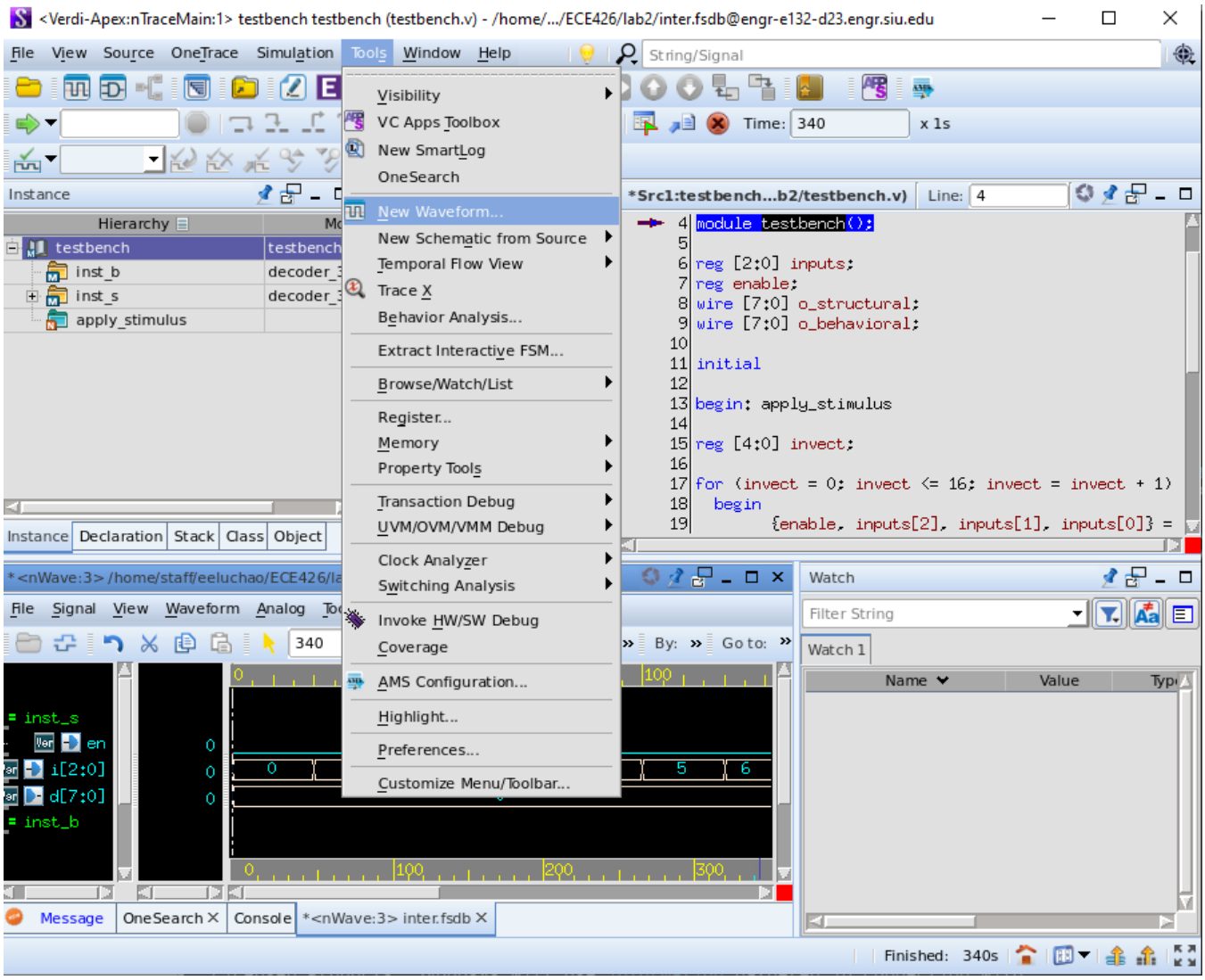
Type ‘vcs testbench.v -debug_acc+all -debug_region+cell+encrypt -kdb”

```
/home/staff/eeluchao/ECE426/lab2-76> vcs testbench.v -debug_acc+all -debug_region+cell+encrypt -kdb
*** Using c compiler gcc instead of cc ...
                        Chronologic VCS (TM)
         Version T-2022.06-SP1 -- Wed Jan 25 19:09:45 2023

                    Copyright (c) 1991 - 2022 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
           Inclusivity and Diversity" (Refer to article 000036315 at
                       https://solvnetplus.synopsys.com)

Parsing design file 'testbench.v'
Parsing included file 'decoder_3_8_structural.v'.
Back to file 'testbench.v'.
Parsing included file 'decoder_3_8_behavioral.v'.
Back to file 'testbench.v'.
Top Level Modules:
       testbench
No TimeScale specified
Starting vcs inline pass...

1 module and 0 UDP read.
recompiling module testbench
rm -f _cuarc*.so _csrc*.so pre_vcsobj_*.so share_vcsobj_*.so
if [ -x ../simv ]; then chmod a-x ../simv; fi
g++  -o ../simv      -L/opt/net/lib:/opt/net/usr/lib:/usr/lib:/usr/ucblib:/lib32:/usr/lib32 -m32 -m32 -rdynamic  -Wl,-rpath='$ORIGIN'/simv.daidir -Wl,-rpath=./simv.daidir
-Wl,-rpath=/synopsys/VCS_2022/vcs/T-2022.06-SP1/linux/lib -L/synopsys/VCS_2022/vcs/T-2022.06-SP1/linux/lib  -Wl,-rpath-link=./   objs/amcQw_d.o   _32663_archive_1.so  SIM
l.o       rmapats_mop.o rmapats.o rmar.o rmar_nd.o  rmar_llvm_0_1.o rmar_llvm_0_0.o           -lvirsim -lerrorinf -lsnpsmalloc -lvfs    -lvcsnew -lsimprofile -luclinativ
 /synopsys/VCS_2022/vcs/T-2022.06-SP1/linux/lib/vcs_tls.o   -Wl,-whole-archive  -lvcsucli    -Wl,-no-whole-archive        _vcs_pli_stub_.o   /synopsys/VCS_2022/vcs/T-2022
06-SP1/linux/lib/vcs_save_restore_new.o /synopsys/VERDI_2022/verdi/T-2022.06-SP2-1//share/PLI/VCS/LINUX/pli.a /synopsys/VCS_2022/vcs/T-2022.06-SP1/linux/lib/ctype-stubs_3
.a -ldl  -lc -lm -lpthread -ldl
../simv up to date
CPU time: .696 seconds to compile + .342 seconds to elab + .696 seconds to link
Verdi KDB elaboration done and the database successfully generated: 0 error(s), 0 warning(s)
```

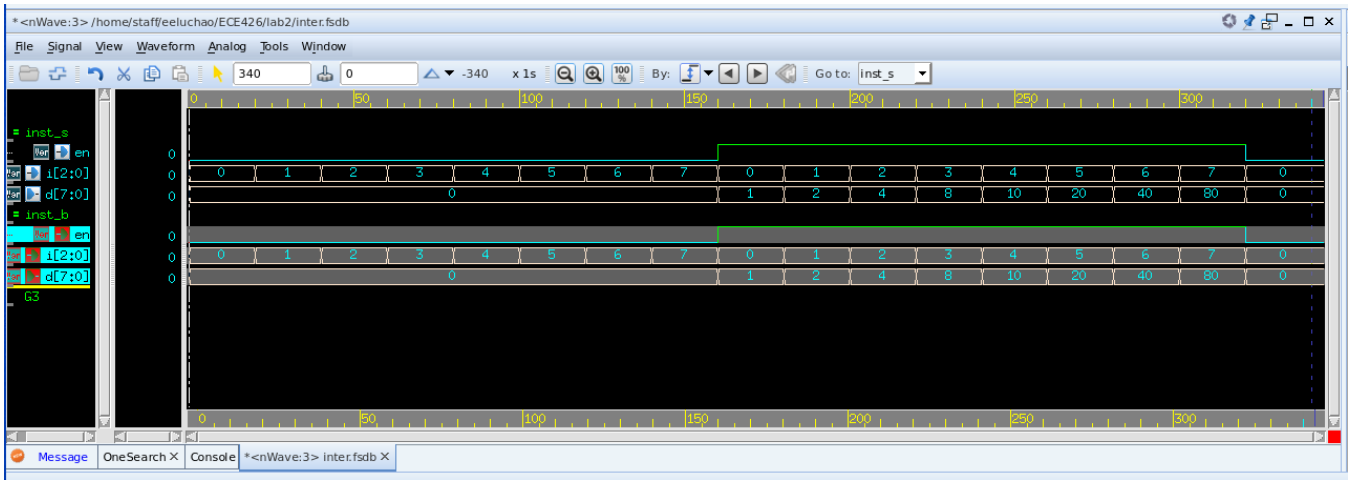
Run ‘./simv -gui’ to see the testbench output in a graphic environment tool verdi.

```
/home/staff/eeluchao/ECE426/lab2-77> ./simv -gui
logDir = /home/staff/eeluchao/ECE426/lab2/verdiLog


                                Verdi (R)

              Version T-2022.06-SP2-1 for linux - Jan 12, 2023

                    Copyright (c) 1999 - 2023 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
           Inclusivity and Diversity" (Refer to article 000036315 at
                       https://solvnetplus.synopsys.com)
rcfile = /synopsys/VERDI_2022/verdi/T-2022.06-SP2-1/etc/novas.rc
guiConfFile (read)= /home/staff/eeluchao/ECE426/lab2/novas.conf (working directory)
guiConfFile (write)= /home/staff/eeluchao/ECE426/lab2/novas.conf (working directory)
*WARN* Verdi-Elite (Verdi) license is not available. Try to check out Verdi-Apex (Verdi-Ultra) license.
```

5. See the waveforms of your design. To generate a waveform, click on "Tools > New Waveform".



6) Drag the instance that is being simulated (e.g. inst_b, inst_s) and drop it on the left black window/panel. Finally, to run the simulation, go to "Simulation >Run/Continue". The results (waveforms) can be observed in the larger (right) window/panel.



Demo

You should demo the following aspects of your design, and include the following 5 items in your report.

1. Verilog code for structural 3:8 decoder design.
2. Simulation waveforms demonstrating correct functionality of the structural 3:8 decoder design
3. Verilog code for behavioral 3:8 decoder design.
4. Simulation waveforms demonstrating correct functionality of the behavioral 3:8 decoder design
5. Synthesize one of your design and capture the reported performance (area and power).