ECE4514

Assignment Submission Procedure

PURPOSE

Automatic Evaluation and Grading of HDL Assignments

Evaluate Functional Correctness

Evaluate Compliance to Coding Standards

Coding Style

The loosely-typed nature of Verilog allows you to crank-out code quickly, yet it also does little to prevent you from writing nasty code.

Verilog was intended to be a *modeling language*, not a *synthesis language*

GETTING THE JOB DONE

VS

DOING THE JOB RIGHT

Why Coding Style?

- Design Consistency
- Design Reusabilty
- Design Portability
- Synthesizability
- Testability

COMPLIANCE TO CODING STANDARDS

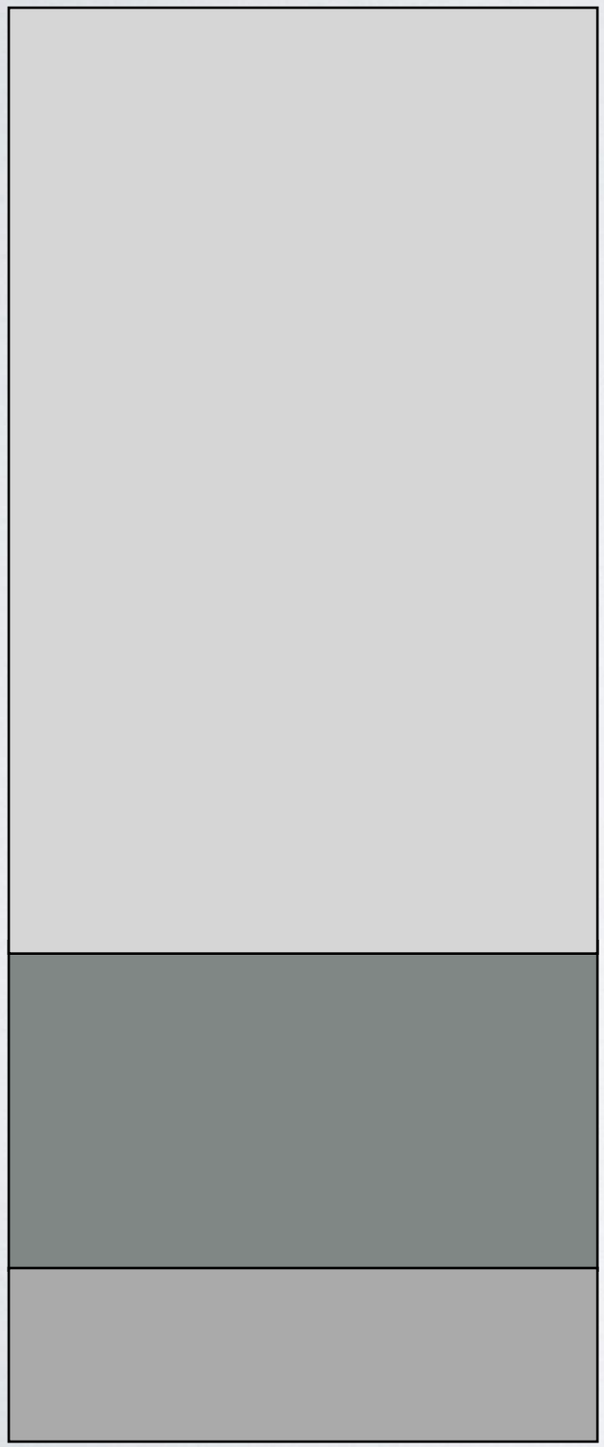
- Ability to communicate, maintain, and archive a design
  - Code headers with consistent fields
  - Proper constructs for synthesis
  - Choice of variable names, function names, etc.
  - Code appearance, tabbing, structure
  - Google: "Verilog coding standards guidelines"

THE INTERNATIONAL OBFUSCATED C CODE CONTEST

```
X=1024; Y=768; A=3;

J=0;K=-10;L=-7;M=1296;N=36;O=255;P=9;_=1<<15;E;S;C;D;F(b){E="1""111886:6:??AAF"
"FHHMMOO55557799@@>>>BBBGGIIKK"[b]-64;C="C@=::C@@==@=:C@=:C@=:C5""31/513/5131/"
"31/531/53"[b ]-64;S=b<22?9:0;D=2;}I(x,Y,X){Y?(X^=Y,X*X>x?(X^=Y):0,  I (x,Y/2,X
)):(E=X);      }H(x){I(x,    _,0);}p;q(        c,x,y,z,k,l,m,a,          b){F(c
);x-=E*M     ;y-=S*M           ;z-=C*M         ;b=x*       x/M+          y*y/M+z
*z/M-D*D    *M;a=-x              *k/M     -y*l/M-z        *m/M;    p=((b=a*a/M-
b)>=0?(I    (b*M,_      ,0),b    =E,      a+(a>b      ?-b:b)):     -1.0);}Z;W;o
(c,x,y,     z,k,l,    m,a){Z=!    c?      -1:Z;c     <44?(q(c,x          ,y,z,k,
l,m,0,0     ),(p>      0&&c!=     a&&        (p<W         ||Z<0)          )?(W=
p,Z=c):     0,o(c+          1,    x,y,z,         k,l,          m,a)):0     ;}Q;T;
U;u;v;w    ;n(e,f,g,             h,i,j,d,a,    b,V){o(0      ,e,f,g,h,i,j,a);d>0
&&Z>=0? (e+=h*W/M,f+=i*W/M,g+=j*W/M,F(Z),u=e-E*M,v=f-S*M,w=g-C*M,b=(-2*u-2*v+w)
/3,H(u*u+v*v+w*w),b/=D,b*=b,b*=200,b/=(M*M),V=Z,E!=0?(u=-u*M/E,v=-v*M/E,w=-w*M/
E):0,E=(h*u+i*v+j*w)/M,h-=u*E/(M/2),i-=v*E/(M/2),j-=w*E/(M/2),n(e,f,g,h,i,j,d-1
,Z,0,0),Q/=2,T/=2,       U/=2,V=V<22?7:  (V<30?1:(V<38?2:(V<44?4:(V==44?6:3))))
,Q+=V&1?b:0,T                +=V&2?b        :0,U+=V    &4?b:0)     :(d==P?(g+=2
,j=g>0?g/8:g/     20):0,j    >0?(U=     j    *j/M,Q      =255-    250*U/M,T=255
-150*U/M,U=255    -100    *U/M):(U    =j*j     /M,U<M          /5?(Q=255-210*U
/M,T=255-435*U           /M,U=255    -720*      U/M):(U       -=M/5,Q=213-110*U
/M,T=168-113*U    /       M,U=111               -85*U/M)      ),d!=P?(Q/=2,T/=2
,U/=2):0);Q=Q<    0?0:      Q>O?     O:          Q;T=T<0?    0:T>O?O:T;U=U<0?0:
U>O?O:U;}R;G;B    ;t(x,y     ,a,    b){n(M*J+M    *40*(A*x   +a)/X/A-M*20,M*K,M
*L-M*30*(A*y+b)/Y/A+M*15,0,M,0,P,  -1,0,0);R+=Q    ;G+=T;B   +=U;++a<A?t(x,y,a,
b):(++b<A?t(x,y,0,b):0);}r(x,y){R=G=B=0;t(x,y,0,0);x<X?(printf("%c%c%c",R/A/A,G
/A/A,B/A/A),r(x+1,y)):0;}s(y){r(0,--y?s(y),y:y);}main(){printf("P6\n%i %i\n255"
"\n",X,Y);s(Y);}
```

ASSIGNMENT POINTS



Functional Correctness 75%

Compliance to coding guidelines 15%

Added Bonus for full Functional Correctness and Coding Style Compliance, 10%

PROCESS

ASSIGNMENTS IN THIS CLASS ARE TO BE COMPLETED INDIVIDUALLY

YOU

DEVELOP MODEL OF CIRCUIT INDIVIDUALLY

DOES IT FUNCTION PROPERLY? (SIMULATE)

EMAIL FILES TO EVALUATION SERVER

EVAL SERVER

EVALUATE IF MODEL FUNCTIONS CORRECTLY

EVALUATE CODING STYLE

RETURN RESULTS BY EMAIL

UNIQUE CHECKER

EVALUATE ORIGINALITY (MOSS)

REPORT TO INSTRUCTOR

MOSS

Moss (for a Measure Of Software Similarity) is an automatic system for determining the similarity of programs. To date, the main application of Moss has been in detecting plagiarism in programming classes: http://theory.stanford.edu/~aiken/moss/

Submissions flagged by MOSS will be forwarded to the Honor Court.

LANGUAGES:
C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Matlab, VHDL, **Verilog**, Spice, MIPS assembly, a8086 assembly, a8086 assembly, MIPS assembly, HCL2.

EMAIL SUBMISSIONS

VirginiaTech
WebMail Compose Message

Added an attachment.

Send Message | Save Draft | Cancel Message

From: athanas@vt.edu (Default Identity)

To: vtevaluator@autograde.ece.vt.edu

CC:

BCC:

Subject: hw1prob2

Attachments: Browse... Attach

prob2b.v (1.71 KB)

Remove Selected Attachments Total Size: 1.71 KB

Check Spelling | Special Characters

I have neither given nor received unauthorized assistance on this assignment.

1. Send submission to this address

2. Problem name (must be exact for the assignment)

3. Attach one or more files

4. Honor pledge

EMAIL SUBMISSIONS

NOTE:

The email must be from your official VT PID.

The FROM and REPLY fields must be your official PID or it will be rejected by the server.

EMAIL RESPONSE (GOOD)

```
Dear Athanas,

The results from the analysis of your submission are complete.  Your submission was analyzed
for functional correctness and for compliance to contemporary Verilog coding style
guidelines.  You can examine the detailed results of your compliance to coding guidelines can
be found here:
     http://moby.ece.vt.edu:8800/4514/submissions/Athanas_read_eeprom_R378781515

If the above link is broken, it could be that something bad happened during analysis.  To
examine the analysis log, examine:
     http://moby.ece.vt.edu:8800/4514/submissions/Athanas_read_eeprom_R378781515/alint.log
     http://moby.ece.vt.edu:8800/4514/submissions/Athanas_read_eeprom_R378781515/results.log

As a reference, the list of the enforced Verilog guidelines can be found at
     http://moby.ece.vt.edu:8800/4514/starc_vlog
     http://moby.ece.vt.edu:8800/4514/aldec_basic
     http://moby.ece.vt.edu:8800/4514/Verilog_Reference

These files will be archived off of the server in 3 days.

Happy coding!

==============  C I R C U I T   V E R I F I C A T I O N   R E S U L T S  ==============

ncsim> source /opt/cadence/tools/inca/files/ncsimrc
ncsim> run
Seed is now
EEPROM MEM[          0] = 0x00
EEPROM MEM[          1] = 0xfe
EEPROM MEM[          2] = 0xf6
```

EMAIL RESPONSE (CONT'D)

```
============ V E R I L O G   C O D I N G   S T Y L E   A N A L Y S I S ============

A total of 1 style warnings were were identified in your design.  Refer to the style analysis
link above for more information. Below are a summary of the identified problems:


================================================================================
    STARC_VLOG.2.10.3.1 - Clearly match the bit widths of the right side and the left side of
relational operators
--------------------------------------------------------------------------------
Help:                   | http://moby.ece.vt.edu:8800/4514/starc_vlog/starc_vlog.2.10.3.1.htm
================================================================================
                        |
Rule:                   | STARC_VLOG.2.10.3.1
Violation:              | Right side bit width 32 of relational operator == does not match to
bit width 8 of the left side. Use equal bit widths to avoid confusion.
File:                   | Athanas_read_eeprom_R378781515/read_eeprom.v
Line number:            | 151
Design unit:            | read_eeprom
Instance:               | Instance read_eeprom
Violation type:         | Relevant
Severity:               | Warning
Rule level:             | Recommendation 2
                        |
================================================================================
```

EMAIL RESPONSE (BAD)

```
============  C I R C U I T   V E R I F I C A T I O N   R E S U L T S  ============

Evaluation of homework assignment hw1prob2.


************************ BEGIN Athanas_hw1prob2_R511909040 TEST ************************
                500: MISMATCH for {c, b, a, d1, d0} = 000101, z out=1 (should be 0).
                600: MISMATCH for {c, b, a, d1, d0} = 000110, z out=0 (should be 1).
                700: MISMATCH for {c, b, a, d1, d0} = 000111, z out=0 (should be 1).
                800: MISMATCH for {c, b, a, d1, d0} = 001000, z out=0 (should be 1).
               2100: MISMATCH for {c, b, a, d1, d0} = 010101, z out=0 (should be 1).
               2200: MISMATCH for {c, b, a, d1, d0} = 010110, z out=1 (should be 0).
               2300: MISMATCH for {c, b, a, d1, d0} = 010111, z out=1 (should be 0).
               2400: MISMATCH for {c, b, a, d1, d0} = 011000, z out=1 (should be 0).
               2500: MISMATCH for {c, b, a, d1, d0} = 011001, z out=1 (should be 0).
               2600: MISMATCH for {c, b, a, d1, d0} = 011010, z out=1 (should be 0).
               2700: MISMATCH for {c, b, a, d1, d0} = 011011, z out=1 (should be 0).
               2800: MISMATCH for {c, b, a, d1, d0} = 011100, z out=0 (should be 1).
************************** END Athanas_hw1prob2_R511909040 TEST **************************
FAILURE -- simulation complete, yet mismatches occurred during testing.
Score:          12 mismatches out of          32 test steps.

Simulation complete via $finish(1) at time 3200 NS + 0
  o
  o
  o
  o
```

EMAIL RESPONSE (BAD)
(cont'd)

- o
- o
- o
- o

```
==============  G R A D E   R E S U L T S ==============

Simulation correctness points: 48 out of 75
Verilog coding style points:   14 out of 15  * subject to change after code review
Bonus for perfect score:        0 out of 10
                               -------
FINAL SCORE                    62 out of 100
```

VERILOG
CODING GUIDELINES

EXPECTED STRUCTURE

```
//  File     :    hw1_prob2b.v
//  Author   :    P. Athanas
//  Date     :    01/20/16
//  Version  :    1.0
//  Description : This file contains a possible solution to
//       HW1, Problem 2B.  The problem statement is as follows:
//       Consider a logic unit that that has 3 operational inputs {A, B, C},
//       two data inputs {D0, D1}, and a single output {Z}.  The logic unit is
//       defined as follows:  when ABC=000, Z=0; ABC=001, ABC=010, Z=D1 AND
//       D0; ABC=011, Z=D1 NAND D0; ABC=100, Z=D1 NOR D0; ABC=101, Z=D1 OR D0;
//       ABC=110,  Z=D1, ABC=111, Z=1.

module hw1_prob2b(z, d1, d0, a, b, c );
     output z;      // Computed function output
     input d0;      // Data input 0
     input d1;      // Data input 1
     input a;       // Function selector (MSB)
     input b;       // Function selector
     input c;       // Function selector (LSB)

     // Declare local wires
     wire func000 = 1'b0;     // 000 => constant0
     wire func001 = d0;  // 001 => d0
     wire func010 = d1;  // 010 => d1
     wire func011;       // 011 (below)
     wire func100;       // 100 (below)
     wire func101;       // 101 (below)
     wire func110 = d1;  // 110 => d1
     wire func111 = 1'b1;     // 111 => constant1

     // Create the individual functions:
     nand      u1 (func011, d0, d1);
     nor       u2 (func100, d0, d1);
     or        u3 (func101, d0, d1);

     // Instantiate the multiplexor
     mux8to1 u4 ( z, a, b, c,
          func111, func110, func101, func100,
          func011, func010, func001, func000 );

endmodule
```

REQUIRED HEADER COMPONENTS

```
//  File        :    hw1_prob2b.v
//  Author      :    P. Athanas
//  Date        :    01/20/16
//  Version     :    1.0
//  Description : This file contains a possible solution to
//        HW1, Problem 2B.  The problem statement is as follows:
//        Consider a logic unit that that has three operational inputs {A, B, C},
//        two data inputs {D0, D1}, and a single output {Z}.  The logic unit is
//        defined as follows:  when ABC=000, Z=0; ABC=001, Z=D0; ABC=010, Z=D1 AND
//        D0; ABC=011, Z=D1 NAND D0; ABC=100, Z=D1 NOR D0; ABC=101, Z=D1 OR D0;
//        ABC=110,  Z=D1, ABC=111, Z=1.

module hw1_prob2b;
```

Include one space between header

Using Aldec ALINT

< See this tutorial>

Sample Rules

- **RMM.VLOG.5.2.1.3**
- **RMM.VLOG.5.5.2.1**
- **ALDEC_BASIC.VLOG.2202**
- **STARC_VLOG.1.1.1.5**
- **STARC_VLOG.3.2.3.1**

PRACTICE

For some assignments, you will be allowed to submit as many times as you like up to the due date.

For some assignments, you will only be allowed to submit them during class.

For some assignments, you'll only get one chance to submit.

The End

Project #0

1. Resurrect your last project in ECE3544

2. Extract the Verilog source code. Delete all testbench files.

3. Using ALINT, compile a list of all errors and warnings. Repair all reported errors and warnings, preserving correct operation.

4. Verify that it still works.

THIS PROJECT IS WORTH 0 POINTS AND HAS NO DUE DATE

Graduate Students: Try ALINT on a recent Verilog/VHDL project that you completed.