

EECS 270 W25

Discussion 5

February 6th, 2025
By: Mick Gordinier

Please download this zip
file in CAEN to participate
in today's discussion!



AGENDA

1. Discussion 4 Recap
2. ModelSim Demos!
3. Project 4 Discussion
4. (Bonus) Number Representation Practice

Discussion 4 Recap

Verilog's Many Levels of Abstraction

**“BEHAVIORAL
VERILOG”**

Behavioral Level

Project 3B - 7

Dataflow Level

**“STRUCTURAL
VERILOG”**

Gate Level

Projects 0 - 3A

Switch Level

**Because of Shannon,
we can ignore**

Behavioral Verilog Overview

- Gate-Level “Structural Verilog” Modelling
 - We can instantiate every primitive gate individually and connect them together
 - Helpful in building our knowledge of digital logic design
 - PROBLEM: Can get very difficult and tedious will more complex logic designs
- **Dataflow and Behavioral Level “Behavioral Verilog” Modelling**
 - Higher level abstraction of Verilog
 - **Allows us to focus more on the algorithmic approach than the individual gates**
- Logical Synthesis compiles our design to low-level Verilog to be understood by the board

ModelSim Demos!!

Post Discussion Walkthrough of Buggy Code: [Link to Doc](#)

Project 4 Discussion

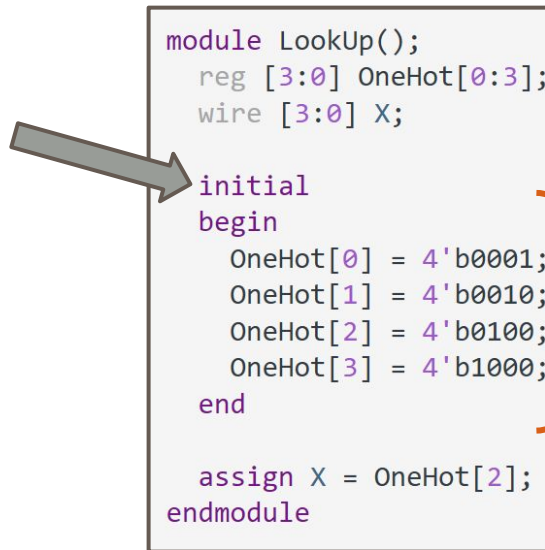
USE BEHAVIORAL VERILOG FOR Project 4!!!

- **You are allowed to use**
 - Continuous 'assign' statements
 - Ternary operators / Concatenation
- **Behavioral Operations NOT allowed for this project**
 - Addition (+), subtraction/integer negation (-)
 - Greater than (>), less than (<), or equality (==)
 - If statements

Procedural Statements and Blocks

- 2 procedural statements: **always** and **initial**
 - Each procedural statement represents a separate activity flow
 - All other behavioral modeling statements go inside the procedural blocks

Procedural Statement



```
module LookUp();  
    reg [3:0] OneHot[0:3];  
    wire [3:0] X;  
  
    initial  
    begin  
        OneHot[0] = 4'b0001;  
        OneHot[1] = 4'b0010;  
        OneHot[2] = 4'b0100;  
        OneHot[3] = 4'b1000;  
    end  
  
    assign X = OneHot[2];  
endmodule
```

Procedural Block

THANK YOU



**Please fill out form if you haven't
already done so!**
([Link Here As Well](#))

(Bonus) Number Representation Practice

Signed Magnitude

Unique Characteristics:

???

4-bit Range: (?, ?)

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

1's Complement

Unique Characteristics:

???

4-bit Range: (?, ?)

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

2's Complement

Unique Characteristics:

???

4-bit Range: (?, ?)

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

Signed Magnitude

Unique Characteristics:

MSB is an unweighted sign bit

0 → Positive, 1 → Negative

4-bit Range: **(-7, -7)**

Most positive 4-bit number
binary representation:

0111_{2C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1111_{2C} → **-7**₁₀

Multiple
Representations? : **Y**

If so, what?:

0000_{SM} → **0**₁₀
1000_{SM} → **-0**₁₀ = **0**₁₀

1's Complement

Unique Characteristics:

???

4-bit Range: **(?, ?)**

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

2's Complement

Unique Characteristics:

???

4-bit Range: **(?, ?)**

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

Signed Magnitude

Unique Characteristics:

MSB is an unweighted sign bit

0 → Positive, 1 → Negative

4-bit Range: **(-7, -7)**

Most positive 4-bit number
binary representation:

0111_{2C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1111_{2C} → **-7**₁₀

Multiple
Representations? : **Y**

If so, what?:

0000_{SM} → **0**₁₀

1000_{SM} → **-0**₁₀ = **0**₁₀

1's Complement

Unique Characteristics:

MSB is $-(2^{(n-1)} - 1)$

Neg. rep. → Just flip bits!

4-bit Range: **(-7, -7)**

Most positive 4-bit number
binary representation:

0111_{1C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1000_{1C} → **-7**₁₀

Multiple
Representations? : **Y**

If so, what?:

0000_{SM} → **0**₁₀

1111_{SM} → **0**₁₀

2's Complement

Unique Characteristics:

???

4-bit Range: **(?, ?)**

Most positive 4-bit number
binary representation:

????

Most negative 4-bit number
binary representation:

????

Multiple
Representations? : **Y / N**

If so, what?:

Signed Magnitude

Unique Characteristics:

MSB is an unweighted sign bit

0 → Positive, 1 → Negative

4-bit Range: **(-7, -7)**

Most positive 4-bit number
binary representation:

0111_{2C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1111_{2C} → **-7**₁₀

Multiple
Representations? : **Y**

If so, what?:

0000_{SM} → **0**₁₀

1000_{SM} → **-0**₁₀ = **0**₁₀

1's Complement

Unique Characteristics:

MSB is $-(2^{(n-1)} - 1)$

Neg. rep. → Just flip bits!

4-bit Range: **(-7, -7)**

Most positive 4-bit number
binary representation:

0111_{1C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1000_{1C} → **-7**₁₀

Multiple
Representations? : **Y**

If so, what?:

0000_{SM} → **0**₁₀

1111_{SM} → **0**₁₀

2's Complement

Unique Characteristics:

MSB is $-(2^{(n-1)} - 1)$

Neg. rep. → Flip bits, add 1

No repeat representation!

Easy hardware implementation!

4-bit Range: **(-7, +8)**

Most positive 4-bit number
binary representation:

0111_{2C} → **+7**₁₀

Most negative 4-bit number
binary representation:

1000_{2C} → **-8**₁₀

Multiple
Representations? : **N**

If so, what?: **N/A**