Sign Extender

- Emily Harrison
- Jordan Rodecap
- Holland Whitley
- Taylor Perry

What is it?

- The process of increasing the number of bits to 64 bits
- If the first value is 0, it will add 0's at the beginning to keep the number positive
- If the first value is 1, it will add 1's at the beginning to keep the number negative

Why Need it?

- Memory does not always give out a 64-bit integer
- Add or subtract two integers when they could possibly have a different number of bits

Test Code

```
hawhitley@ubuntu:~/csci-330-spring-2020/homework/Gate2/SignExt$ make
iverilog *.v && ./a.out
SignExtTest.v:23: warning: extra digits given for sized binary constant.
NEGATIVE
Unextended = 1100011111111111
Extended = 1111111111111111111111111111111111111111111111111100011111111111
POSITIVE
Unextended = 0000000011111111
Extended = 0000000000000000000000000000000000000000000000000000011111111
TOO MANY
Unextended = 0000000001111111
Extended = 0000000000000000000000000000000000000000000000000000001111111
NOT ENOUGH
Unextended = 0000000000111000
Extended = 0000000000000000000000000000000000000000000000000000000111000
```

```
module SignExtTest();

reg [15:0] Unext;
wire [63:0] Ext;

SignExt DUT(Unext, Ext);

initial
begin

        Unext <= 16'b1100011111111111;
        #4
        $display("NEGATIVE");
        $display("Unextended = %b", Unext);
        $display("Extended = %b", Ext);

        Unext <= 16'b0000000011111111;
        #4
        $display("POSITIVE");
        $display("Unextended = %b", Unext);
        $display("Extended = %b", Ext);

        Unext <= 16'b0000000001111111;
        #4
        $display("TOO MANY");
        $display("Unextended = %b", Unext);
        $display("Extended = %b", Ext);

        Unext <= 16'b111000;
        #4
        $display("NOT ENOUGH");
        $display("Unextended = %b", Unext);
        $display("Extended = %b", Ext);

"SignExtTest.v" 37L, 691C
```

Problems Encountered