Lecture 7

System Verilog Note: updated from Verilog

Common Hardware Blocks Note: what we've looked so far is "Combinational Logic" - Current input used to calculate output - **Nothing is stored**

Modules: - A block of hardware with inputs and outputs

```
module basic\_logic(input logic a, b, output logic w, x, y, z)
    assign w = a & b;
    assign x = a | b;
    assign y = ~a;
    assign z = a ^ b;
endmodule
```

- assign: describes combinatorial logic
- logic: type that indicates Boolean variables
- actual piece of hardware: cannot call itself, no recursion
- Multiplexers (mux)
  - Design a circuit that controls a light f from either switches x,y
  - s is a switch that decides which switch controls f
  - if s == 0, f is controlled by x, else y
  - Called 2-to-1 mux: selects 1 of 2 possible inputs
- Multi-bit signals, a.k.a. a bus
  - Mux with 2 bit inputs x, y
- Verilog code

```
module mux2to1-2bit(input logic[1:0]x,y, inut logic s, output logic[1:0] f)
    assign f[1] = (~s & x[1]) | (s & y[1]);
    assign f[0] = (~s & x[0]) | (s & y[0]);
endmodule
```

- Adders
  - Half Adder
  - s = a + b where + means addition
  - 2 bit result

a	b	s_1	s_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Full Adder
  - What if we want to add multi-bit numbers, e.g. a and b are 8 bit?

c_i	a_i	b_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

where c_i stands for the carry term.