Lecture 9

Always Block

- Always has to be powered

```
module mux(input logic X1, X2, S, output logic f);
    always_comb
    begin
        if (s==0)
            f = X1;
        else
            f = X2;
    end
endodule
```

Case Statements

```
module seg7(input logic [3:0] SW, output logic [6:0] HEX0);
    always_comb
    begin
        case(SW)
            0: HEX0 = 7'b1000000;
            1: HEX0 = 7'b1111001;
            //so on
            9: HEX0 = 7'b0010000;
            default: HEX0 = 7'b0000000;
        endcase
    end
endmodule
```

Karnaugh Maps

- Optimisations using algebra is awkward and error prone
- Minterms that can be combined are adjacent

$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

If we have functions

x_1	x_2	Minterm	f_1	f_2
0	0	m_0	1	0
0	1	m_1	1	1
1	0	m_2	0	0

x_1	x_2	Minterm	f_1	f_2
1	1	m_3	0	1

Then

$$f_1 = m_0 + m_1 = \overline{x_1}\overline{x_2} + \overline{x_1}x_2 = \overline{x_1}$$

$$f_2 = m_1 + m_3 = \overline{x_1}x_2 + x_1x_2 = x_2$$

Consider

x_1	x_2	f
0	0	1
0	1	1
1	0	0
1	1	1

$$f = \overline{x_1}\overline{x_2} + \overline{x_1}x_2 + x_1x_2$$

The cost, i.e. total number of gates and inputs, is

$$3 \text{ AND } + 1 \text{ OR } + 2 \text{ NOT } + 3 \times 2 + 1 \times 3 + 2 \times 1 = 17$$

where a 3-input OR gate is used.
If the minimal SOP were used instead,

$$f = \overline{x_1} + x_2$$

$$\text{cost } = 1 \text{ OR } + 1 \text{ NOT } + 2 + 1 = 5$$