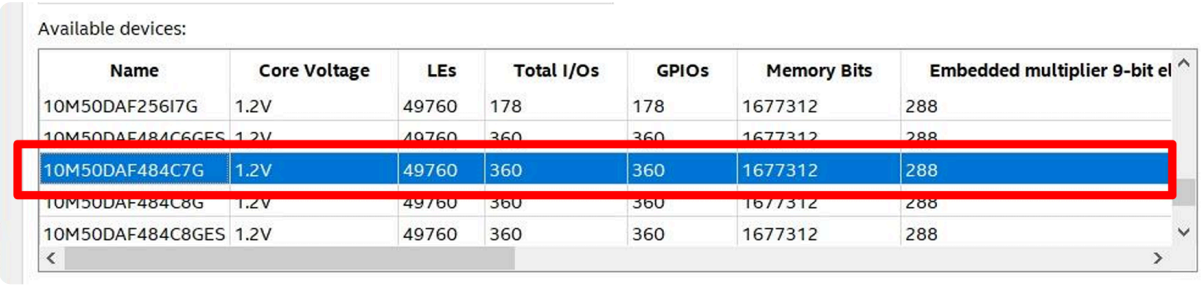
Week8 Adder Subtractor and Encoder

Basic Setting

1. **Device Family:** MAX 10(DA/DF/DC/SA/SC)
2. **Available Devices:** 10M50DAF484C7G


Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit el
10M50DAF256I7G	1.2V	49760	178	178	1677312	288
10M50DAF484C6GES	1.2V	49760	360	360	1677312	288
10M50DAF484C7G	1.2V	49760	360	360	1677312	288
10M50DAF484C8G	1.2V	49760	360	360	1677312	288
10M50DAF484C8GES	1.2V	49760	360	360	1677312	288

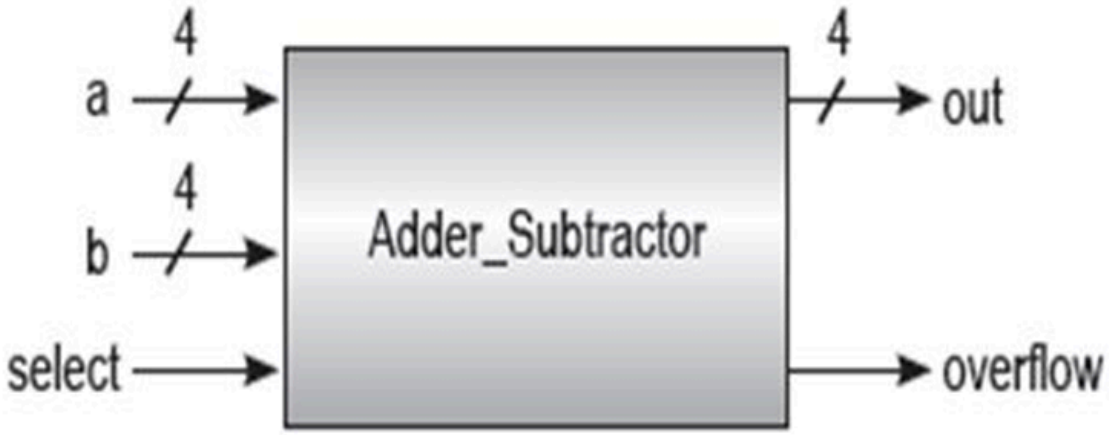
3. **EDA Tool Settings, Simultation:** Modelsim-Altera, Verilog HDL
4. Click 「Assignments > Pin Planner > Location」分配 Pin
5. 到裝置管理員，「其他裝置 > USB-Blaster」，右鍵「更新驅動程式」
6. 瀏覽電腦上的驅動程式
7. 路徑 `C:\intelFPGA_lite\16.1\quartus\drivers\usb-blaster` （16.1 或 18.1）
8. 到裝置管理員，「通用序列匯流排控制器 > Altera USB-Blaster」，有就代表成功
9. Click 「Tools > Programmer」
10. Click 「Hardware Setup > USB Blaster」，然後 Currently selected hardware 選「USB-Blaster [USB0]」
11. Click 「Add file > ~.sof > Open」（注意是在 `outputfiles` 裡面）
12. Click 「Start」顯示 Success 成功

Adder Subtracter

Logic Gate

- Input: a(4bits), b(4bits), select
  - select == 0 means a-b
  - select == 1 means a+b

- Output: out(4bits), overflow



Verilog Method 1

```
module main(a, b, select, out, overflow);
input[3:0] a, b;
input select;

output reg [3:0] out;
output reg overflow;

always@(*)
begin
        if(select)
        begin
                {overflow, out} = a+b;
        end
        else
        begin
                {overflow, out} = a-b;
        end
end
endmodule
```

Verilog Method 2

```
module main(a, b, select, out, overflow);
input [3:0] a, b;
input select;
output reg [3:0] out;
output reg overflow;

always@(*)
begin
        {overflow, out} = (select) ? a+b : a-b;
end
endmodule
```

Decoder

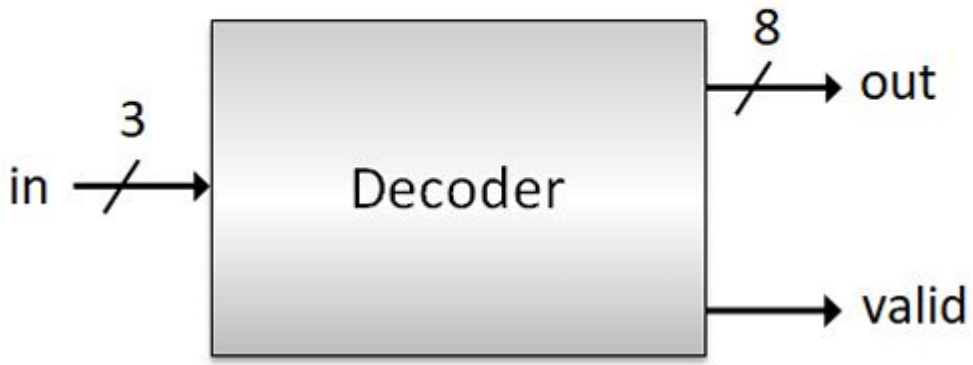
Truth Table

- Input: in
- Output: out, valid
  - valid == 0 means 輸入的值並非 000 ~ 111
  - valid == 1 means 輸入的值是 000 ~ 111

in[2]	in[1]	in[0]	out[7]	out[6]	out[5]	out[4]	out[3]	out
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	1	1	0	0	0	0	1	0
1	0	0	0	0	0	1	0	0
1	0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0

輸入(input)			輸出(output)								
in[2]	in[1]	in[0]	valid	out[7]	out[6]	out[5]	out[4]	out[3]	out[2]	out[1]	out[0]
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
其餘的輸入情況			0	0	0	0	0	0	0	0	0

Logic Gate



Verilog

```
module main(in, out, valid);
input [2:0] in;
output reg [7:0] out;
output reg valid;

always @(*)
begin
        case(in)
                3'b000: out = 8'b00000001;
                3'b001: out = 8'b00000010;
                3'b010: out = 8'b00000100;
                3'b011: out = 8'b00001000;
                3'b100: out = 8'b00010000;
                3'b101: out = 8'b00100000;
                3'b110: out = 8'b01000000;
                3'b111: out = 8'b10000000;
                default: out = 8'b00000000;
        endcase
        valid = (out != 0) ? 1 : 0;
end
endmodule
```