Week7 Half Adder and Full Adder

Basic Setting

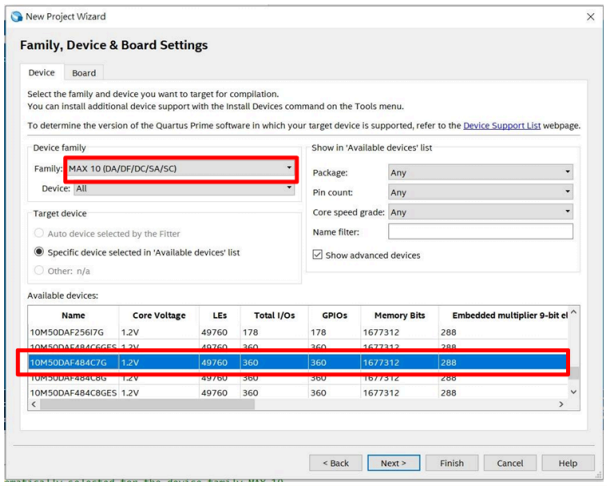
1. **Device Family:** MAX 10(DA/DF/DC/SA/SC)
2. **Available Devices:** 10M50DAF484C7G

Quartus II Tutorial (5/19)

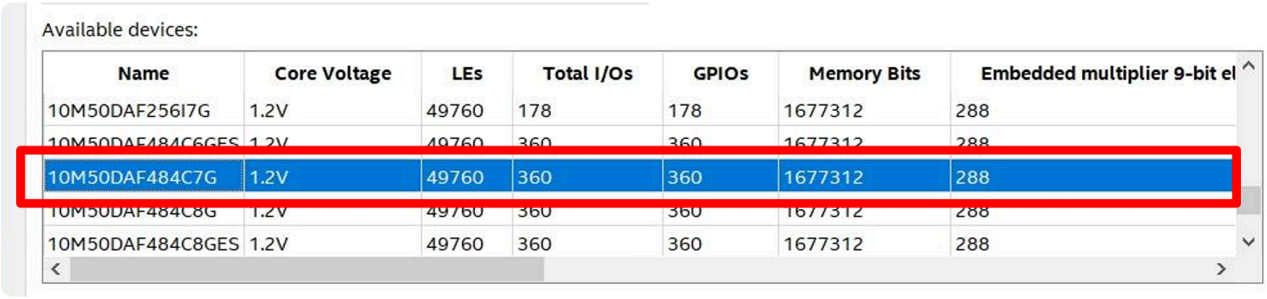
- **Specify device settings - (DE10-Lite Device family are used). Click “Next.”**

MAX 10(DA/DF/DC/SA/SC)

10M50DAF484C7G



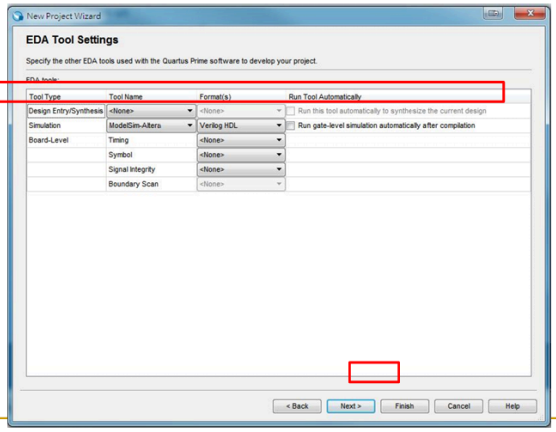
6

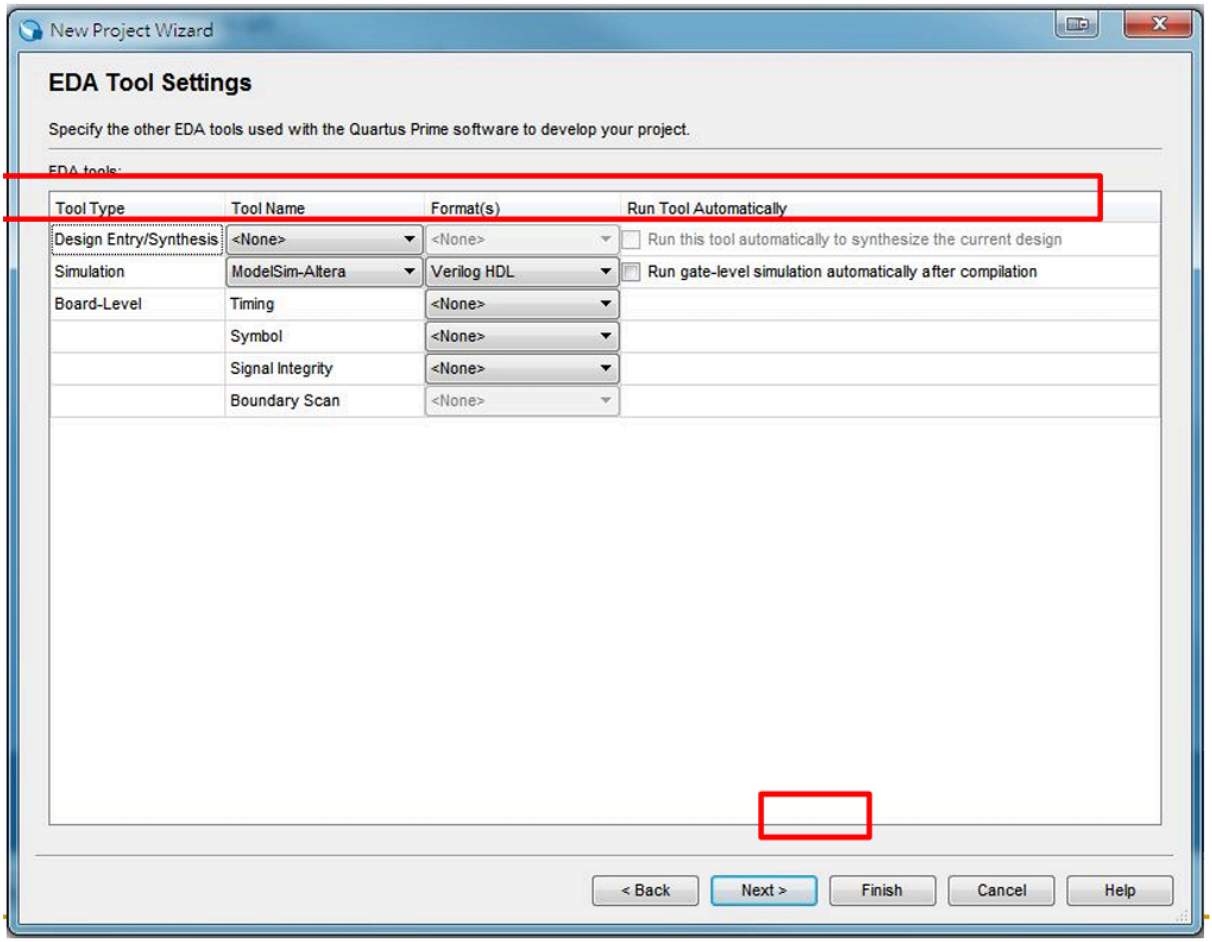


3. **EDA Tool Settings, Simultation:** Modelsim-Altera, Verilog HDL

Quartus II Tutorial (6/19)

- **Specify EDA Tool – (Modelsim-Altera is selected for simulation). Click “Finish.”**





Half Adder

Truth Table

- Input: a, b
- Output: sum, c_out

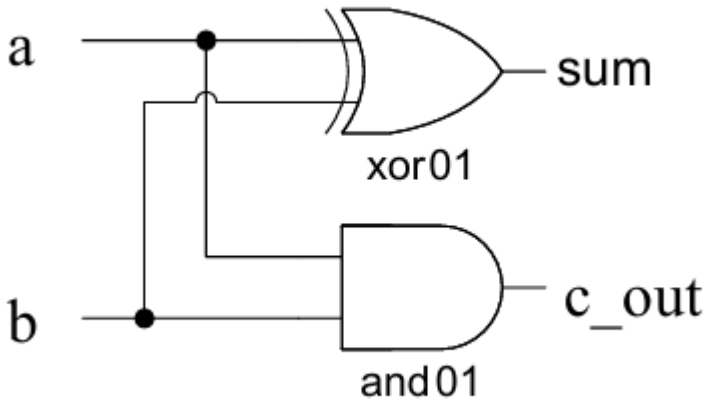
a	b	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Boolean Algebra

$$\text{sum} = \bar{a}b + a\bar{b} = a \oplus b$$
$$\text{carry} = a \cdot b$$

Logic Gate

圖片中的 c_cout = carry



Structural Level

```
1  module main(a, b, sum, c_out);
2  input a, b;
3  output sum, c_out;
4
5  xor xor1(sum, a, b);
6  and and1(c_out, a, b);
7  endmodule
```

Data Flow Level

```
module main(a, b, sum, c_out);
input a, b;
output sum, c_out;

assign sum = a ^ b;
assign c_out = a & b;
endmodule
```

Behavior Level

```
module main(a, b, sum, c_out);
input a, b;
output reg sum, c_out; // 記住要有 reg

always@(*)
begin
    {c_out, sum} = a+b;
end
endmodule
```

Full Adder

Truth Table

- Input: a, b, cin
- Output: sum, cout

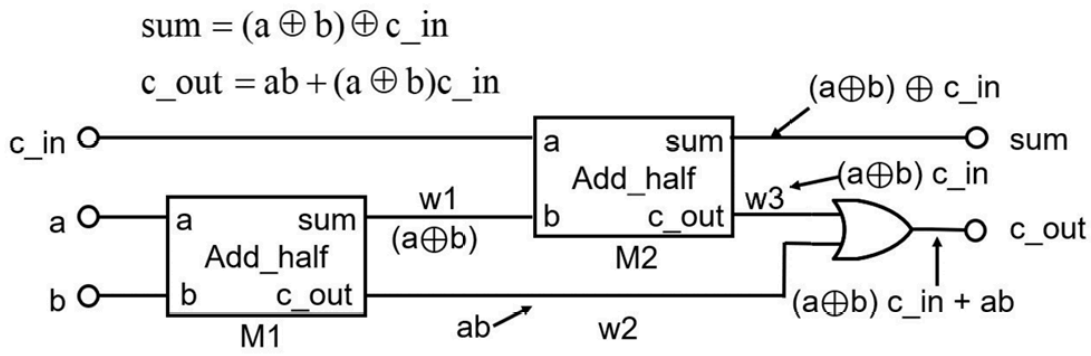
a	b	cin	sum	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Algebra

$$\text{sum} = \bar{a}\bar{b}\text{cin} + \bar{a}b\bar{\text{cin}} + a\bar{b}\bar{\text{cin}} + ab\text{cin} = a \oplus b \oplus \text{cin}$$
$$\text{cout} = ab + a\text{cin} + b\text{cin} = (a \cdot b) + (\text{cin} \cdot (a \oplus b))$$

Logic Gate

我們可以利用 2 Half Adder 合成 1 Full Adder



Half Adder

```
module half_adder(a, b, sum, c_out);
input a, b;
output reg sum, c_out;

always@(*)
begin
        {c_out, sum} = a + b;
end
endmodule
```

Structural Level

```
module main(c_in, a, b, sum, c_out);
input c_in, a, b;
output sum, c_out;

wire w1, w2, w3;

half_adder h1(a, b, w1, w2);
half_adder h2(c_in, w1, sum, w3);
or or1(c_out, w3, w2);
endmodule
```

Data Flow Level

```
module main(c_in, a, b, sum, c_out);
input c_in, a, b;
output sum, c_out;

wire w1, w2, w3;

half_adder h1(a, b, w1, w2);
half_adder h2(c_in, w1, sum, w3);
assign c_out = w2 | w3;
endmodule
```

Behavior Level

```
module main(c_in, a, b, sum, c_out);
input c_in, a, b;
output reg sum, c_out;

always@(*)
begin
        {c_out, sum}=c_in + a + b;
end
endmodule
```