

Coding environment: win10 64 位元 / Verilog: v11 – 20190809 – x64

Implement:

CPU.v :宣告很多 wire 把他們放進小 module 的 output 跟 input , 當作把它們接起來

ALU_control.v: 把 funct_i[9:0]跟 ALUOp_i[1:0], 如果 ALUOp_i == 00 那就是 ADD 輸出, 其他的 e 根據 funct_i 判斷

ALU.v: 根據 ALUCtrl_i 對 data1_i, data2_i 做對應動作

```
`define AND 3'b000
```

```
`define OR 3'b001
```

```
`define ADD 3'b010
```

```
`define SUB 3'b110
```

```
`define MUL 3'b011
```

MUX32.v: 根據 select_i 決定 data_o = data1 還是 data2

Signed_Extend.v: 把 12bit 做 extend 變成 32bit

```
data_o[11:0] = data_i[11:0];
```

```
data_o[31:12] = {20{data_i[11]}};
```

Control.v: 根據 Op_i, 輸出對應的 ALUOp_o, ALUSrc_o, RegWrite_o 給 ALUcontrler, MUX, register

Adder.v: 只是個 32bit adder

OUTPUT 有一條 warning 消除不掉 應該沒關係吧