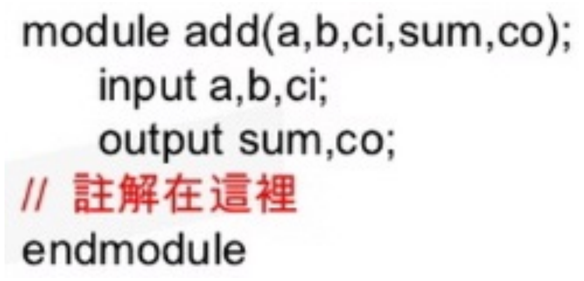
Verilog 初體驗

Verilog – 基本介紹

- Hardware Description Language(HDL)
- 用來描述數位設計的語言
  Ex. memory, flip-flop…etc
- 親近、易學
  Like C

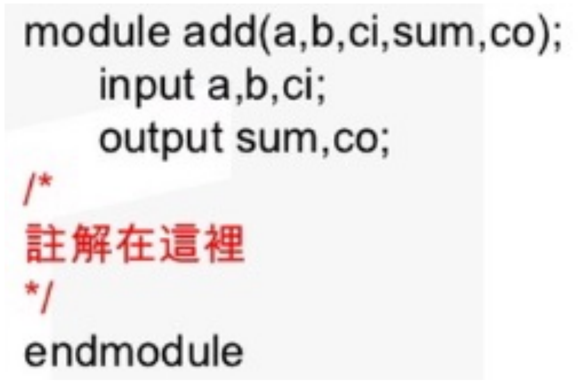
Verilog語法 – 註解

- 單行註解



```
module add(a,b,ci,sum,co);
      input a,b,ci;
      output sum,co;
// 註解在這裡
endmodule
```

- 多行註解



```
module add(a,b,ci,sum,co);
      input a,b,ci;
      output sum,co;
/*
註解在這裡
*/
endmodule
```

Verilog語法 – 變數宣告

- wire – input, output…etc
  * 附值需使用assign

- reg – unsigned
  * 使用在begin…end內

- Integer – signed(32 - bit)
  * for loop

Verilog語法 – 附值

```
wire a ;        //純量變數
wire [7 : 0]  bus ;  // 8-bit  匯流排
wire [31 : 0]  busA, busB, busC ;
          //3個32-bit寬度的匯流排
reg clock ;
reg [0 : 40]  virtual_addr ;
//41 bits寬度虛擬位址。
```

Verilog語法 – 附值(取值)

- busA [7]   // 匯流排 A 之第七個位元
- bus [2:0]

// 從bus第2個bit取到第0個bit

Verilog語法 –

- Sized numbers
  - 4'b1111 // 4-bit 二進位數
  - 12'habc // 12-bit 十六進位數
  - 16'd255 // 16-bit 十進位數
- Unsized numbers
  - 23456 //   32 bits 十進位數
  - 'hc3 //   32 bits 十六進位數
  - 'o21 //   32 bits 八進位數

Verilog語法 –

- x -不確定的值
  - Ex. 判斷式 A = 11’ hx

- singed
  - Ex. reg signed [3:0] j;
  - Ex. -6’sd3

    // 有號整數(Signed Integer)

Verilog語法 – Array

- integer count[0:7]；//8個整數組成的陣列
- reg bool[31:0]；
  //32個一位元布林暫存器變數組成的陣列

- reg [4:0] port_id [0:7]; //8個5位元組成的陣列
- integer matrix [4:0] [0:255]；
  //二維陣列

Verilog語法 – Operators

- ~ not

  Ex. ~x
- & and

  Ex. 0 & x = 0, 1&x = x & x = x
- | or

  Ex. 1 | x = 1, 0 | x = x | x = x
- ^ xor

  Ex. 0^x = 1^x = x^x = x

Verilog語法 –

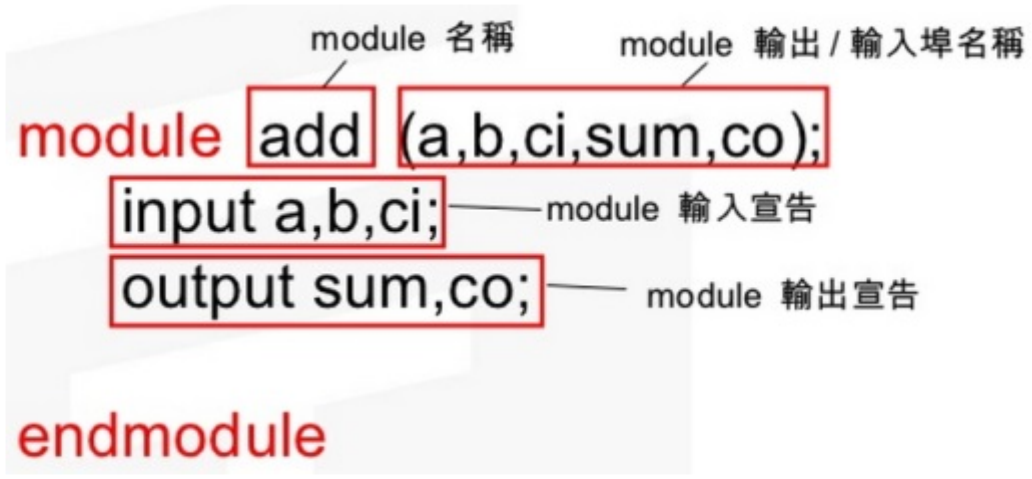
- {} – 用來將不同信號組合在一起

```
reg [3:0]a;
reg b;
wire [10:0]c;
C[10:0] = {a[2:0], 4'b1101, a[3], 1'b1, 1'b0}
```

- {n{m}} – Replication : replicate value m, n times

```
{b, {3{c,d}}} = {b, c, d, c, d, c, d}
```

開始第一行Verilog



```
module add (a,b,ci,sum,co);
    input a,b,ci;
    output sum,co;

endmodule
```

各種判斷式

- 跟C一模一樣
  {}替換為begin…end

initial, always

```
initial
  begin
    a = 1; // Assign a value to reg a at time 0
    #1; // Wait 1 time unit
    b = a; // Assign the value of reg a to reg b
  end

always @(a or b) // Any time a or b CHANGE, run the process
begin
  if (a)
    c = b;
  else
    d = ~b;
end // Done with this block, now return to the top (i.e. the @ event-control)

always @(posedge a)// Run whenever reg a has a low to high change
  a <= b;
```

Verilog – System Tasks

- $display(p1, p2, p3….);
- $time – 顯示目前timer時間

- Ex. $display(“At time %d virtual adderess is %h”,$time, virtual_adder);

- 特殊字元 : %%、\n

Verilog – System Tasks

- $monitor(p1, p2, p3…);
  - 每次僅有一個monitor可執行

ex. $monitor ($time,"Value of signals clock = %b reset=%b,click,reset);

Result :

0 Value of signals clock = 0 reset = 1

Verilog – System Tasks

- $stop;
  = system(“pause”);

- %finish

Verilog – Compiler Directives

- define
  // 在程式碼中使用 ‘ WORD_SIZE
  define WORD_SIZE 32
  // 定義一個別名，當s出現時用$stop取代
  define s $stop;

- include
  include header.v