

## 數位邏輯實驗 — 作業一

系級：資工 2A      姓名：林博仁    學號：09957010

ㄟ 實驗名稱：2 對 1 多工器、4 對 1 多工器以及 8 對 1 多工器

ㄟ 實驗目的：

- 1) 熟悉 2 對 1 多工器、4 對 1 多工以及 8 對 1 多工器的概念
- 2) 學習 Gate Level 的使用
- 3) 熟悉 Xilinx ISE 軟體操作

ㄟ Verilog HDL code：

(個別貼三個.v 檔的 code[用框線區隔]，勿混在一起)

```
//2 對 1 多工器
`ifndef MULTIPLEXER2TO1_V_INCLUDED
  `define MULTIPLEXER2TO1_V_INCLUDED
  `timescale 1ns / 100ps

  module multiplexer2to1(out,i1,i0,s);
    input i0,i1,s;
    output out;
    wire net1, net2;

    not a1(sbar,s);
    and a2(net1,i0,sbar);
    and a3(net2,i1,s);
    or a4(out,net1,net2);
  endmodule
`endif
```

---

//4 對 1 多工器

```

`ifndef MULTIPLEXER4TO1_V_INCLUDED
    `define MULTIPLEXER4TO1_V_INCLUDED
    `timescale 1ns / 100ps
    `include
"Source_code/multiplexer2to1/multiplexer2to1.v"

    module multiplexer4to1(out, i, s);
        output out;
        input [1:0] s;
        input [3:0] i;
        wire net1,net0;

        multiplexer2to1 a1(.out(net0),
.i1(i[1]), .i0(i[0]), .s(s[0]));
        multiplexer2to1 a2(.out(net1),
.i1(i[3]), .i0(i[2]), .s(s[0]));
        multiplexer2to1 b(.out(out), .i1(net1),
.i0(net0), .s(s[1]));
    endmodule
`endif

```

---

```

/* 模組名稱: multiplexer8to1*/
`ifndef MULTIPLEXER8TO1_V_INCLUDE
    `define MULTIPLEXER8TO1_V_INCLUDE
    `timescale 1ns / 100ps
    `include
"Source_code/multiplexer4to1/multiplexer4to1.v"
    `include
"Source_code/multiplexer2to1/multiplexer2to1.v"

    module multiplexer8to1(out, i, s);
        input [7:0]i;
        input [2:0]s;
        output out;
        wire mux41_1o, mux41_0o;

        //我們需要 2 個共用 s[1:0]的 4 對 1 多工器

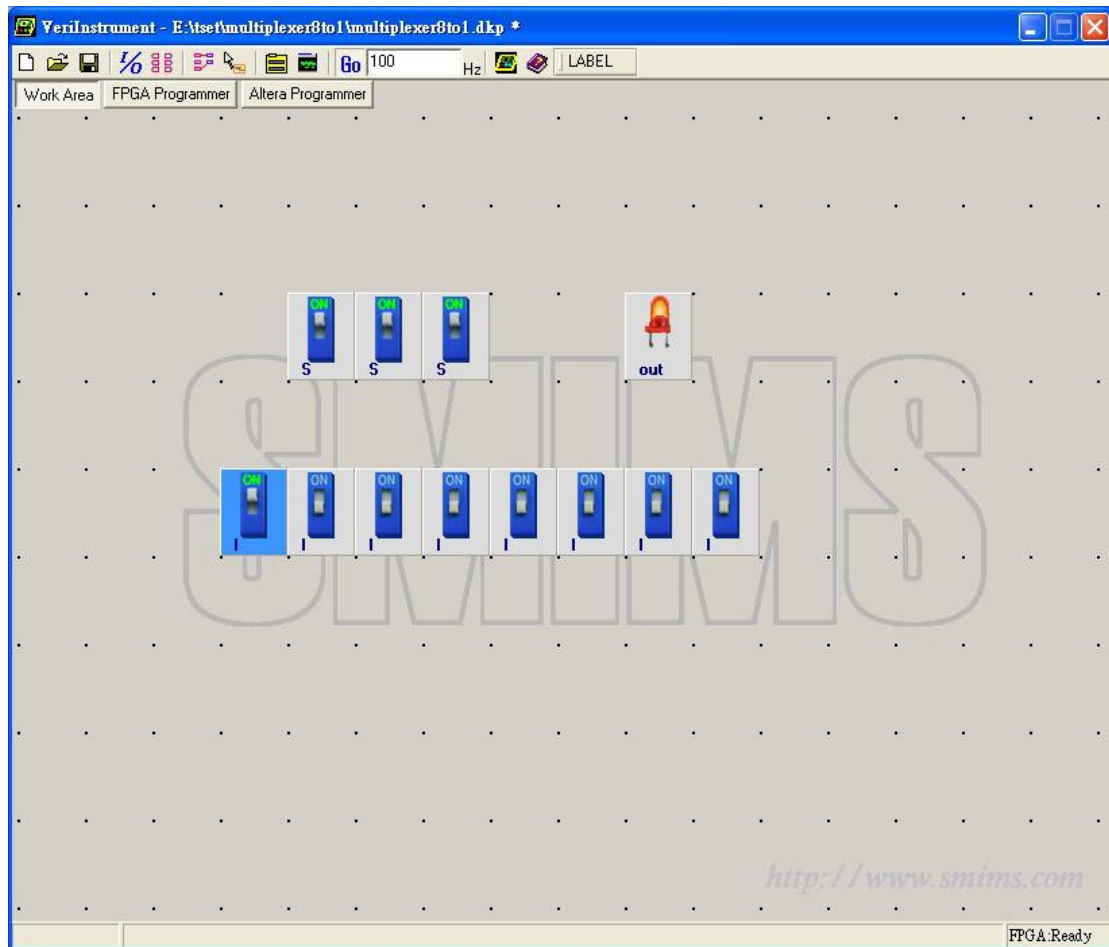
```

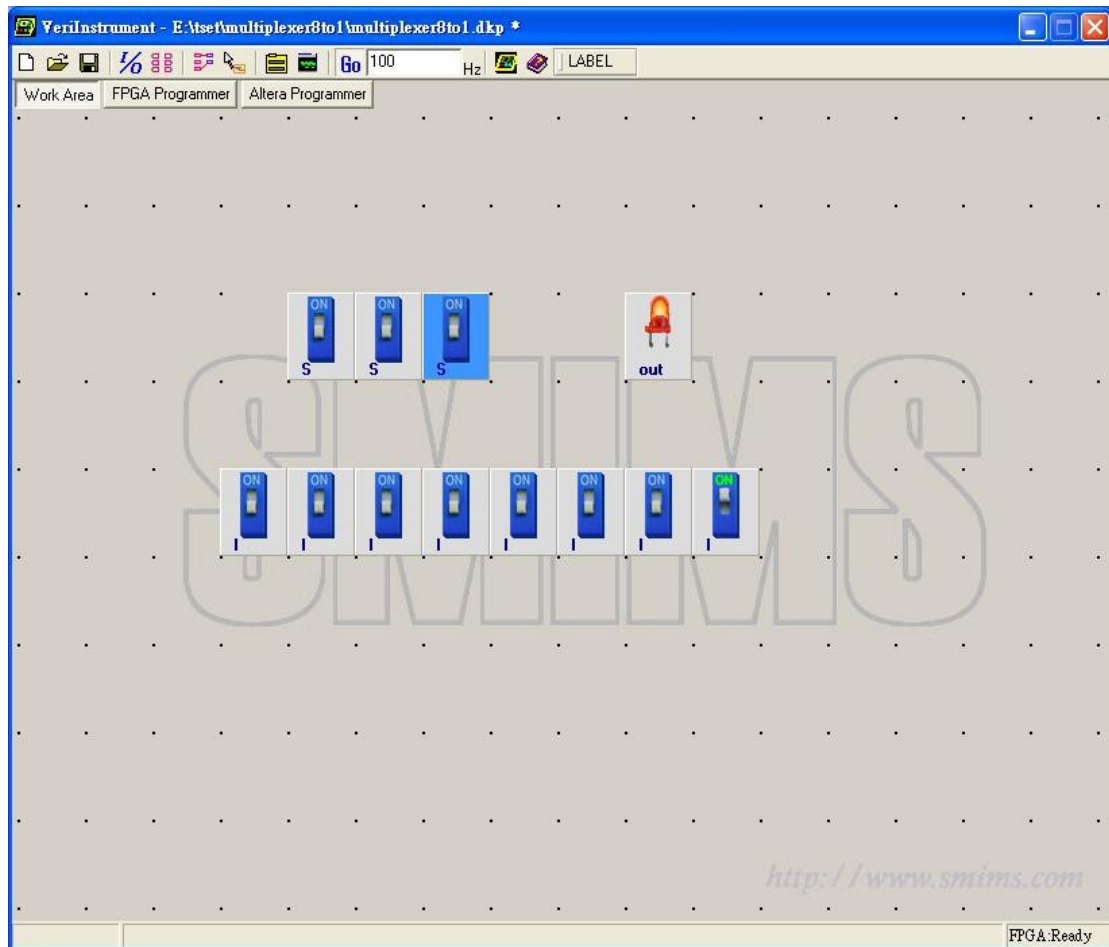
```

multiplexer4to1
  mux41_1(
    .out(mux41_0o),
    .i(i[3:0]),
    .s(s[1:0])
  ),
  mux41_2(
    .out(mux41_1o),
    .i(i[7:4]),
    .s(s[1:0])
  );
//用 1 個 2 對 1 多工器選擇第 2 位
multiplexer2to1
  mux21(
    .out(out),
    .i1(mux41_1o),
    .i0(mux41_0o),
    .s(s[2])
  );
endmodule
`endif

```

📁 模擬結果：[\(跑 VerilInstrument 結果的截圖\)](#)





心得討論：

- 實作了在 GNU/Linux 作業系統底下透過 Icarus verilog compiler、Gtkwave 進行 Verilog 程式碼的編譯以及 testbench 模擬。
- 利用`include 編譯器指令 include 所有 module 所依賴的子 module。
- Testbench 的人工撰寫。