101 學年度上學期　金門大學　資工系三年級　計算機結構　期中考（解答）出題者 ： 陳鍾誠

學號 ：　　姓名 ：　　分數 ：

1. 解釋名詞 (共 15%)

(a) 2 補數 (3%)

二進位的負數表示法，乃是用 1 補數加 1 後得到的結果，其好處是正數加法與負數加法的運算方法完全相同。

(b) 馮紐曼架構 (3%)

包含「控制 、運算(ALU) 、記憶 、匯流排 、輸出入」的處理器或電腦架構，其特色是將程式與資料一同儲存在記憶體，透過指令擷取後再執行的架構。

(c) ALU (3%)

算術邏輯單元，可用來作「加減乘除、移位 、AND、OR、XOR」等功能的計算單元，是 CPU 當中作運算的主要元件。

(d) 控制單元 (3%)

CPU 當中用來控制指令的「擷取 、解碼 、執行」等動作的子元件，可指揮「ALU 、暫存器 、輸出入」等單元進行適當的動作。

(e) Verilog 中的 wire 之意義 (3%)

wire 代表電路中的線，就像位元是軟體程式的基本元素一樣，線則是 verilog 這類硬體語言的基本元素。

2. 請畫出 XOR 閘的 (1). 電路符號與 (2) 真值表。(10%)

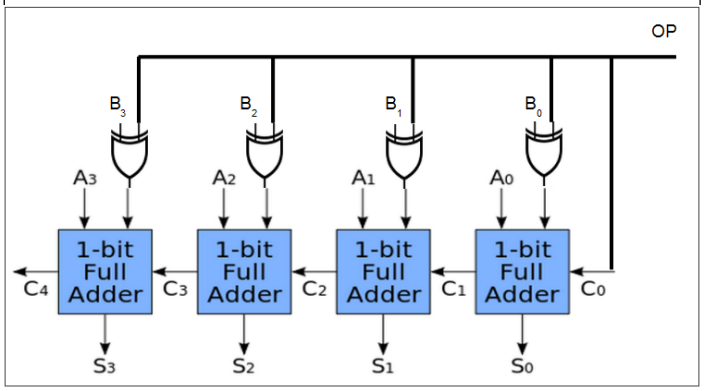
INPUT A	INPUT B	OUTPUT A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

3. 請寫出全加器 (Full Adder, FA) 的真值表並畫出其電路 (共 10%)。

a	b	c_in	c_out	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

電路在最後面 ...

4. 請用四個全加器 FA 與 4 個 XOR 閘，設計一個加減器，並說明其運作原理 (10%)



當 OP = 0 時， XOR 閘會形同沒有作用，於是 A 與 B 會進行加法。當 OP=1 時，XOR 閘會將 B 反相，於是變成 B 的 1 補數，接著 C0=OP=1 輸入進位為 1，這讓電路運作成為 A+(B XOR

	1) + 1，也就是 A 加上 B 的 2 補數，相當於 A-B。 所以 OP=0 時會做 A+B，OP=1 時會做 A-B。

學號 ：　　姓名 ：

5. 請為下列程式寫上註解，說明每一行的意義 (10%)

```
module fulladder (input a, b, c_in, output sum, c_out); // 全加器模組定義，a,b,c_in 為 3 條輸入線，sum, cout 為 2 條輸出線。
wire s1, c1, c2; // 中間線路，參考最後的圖。
xor g1(s1, a, b); // s1 = a xor b
xor g2(sum, s1, c_in); // sum = (a xor b) xor c_in
and g3(c1, a,b); // c1 = a and b
and g4(c2, s1, c_in) ; // c2=(a xor b) and c_in
or g5(c_out, c2, c1) ; // c_out = (a and b) or ((a xor b) and c_in)
endmodule

module main; // 測試程式
reg a, b, c_in; // a, b, c_in 為 1 位元暫存器
wire sum, c_out; // sum, c_out 為線路

fulladder fa1(a, b, c_in, sum, c_out); // 宣告一個全加器 fa1，輸入 a, b, c_in 後，輸出 sum, c_out

initial begin // 初始化區段
  a = 0;  b = 0;  c_in = 0; // 一開始讓 a, b, c_in=0
  $monitor("%04dns monitor: a=%d b=%d c_in=%d c_out=%d sum=%d", $stime, a, b, c_in, c_out, sum); // 監視這些變數，a,b,c_in,c_out, sum，有任何改變就印出來。
  #1000 $finish; // 在 1000ns 的時後結束。
end

always #50 c_in = c_in+1;//每 50ns 讓 c_in+1(反向)。
always #100 b = b+1;// 每 100ns 讓 b+1 (反向)。
```

6. 請設計一顆簡易的 CPU，並寫出下列描述內容：(本題的 CPU 設計愈獨特，與老師和其他同學設計愈不同者，分數愈高) (空間不夠請註明並寫在背面)

(a) 請問該 CPU 的暫存器是幾位元的？有哪些暫存器？ (5%)

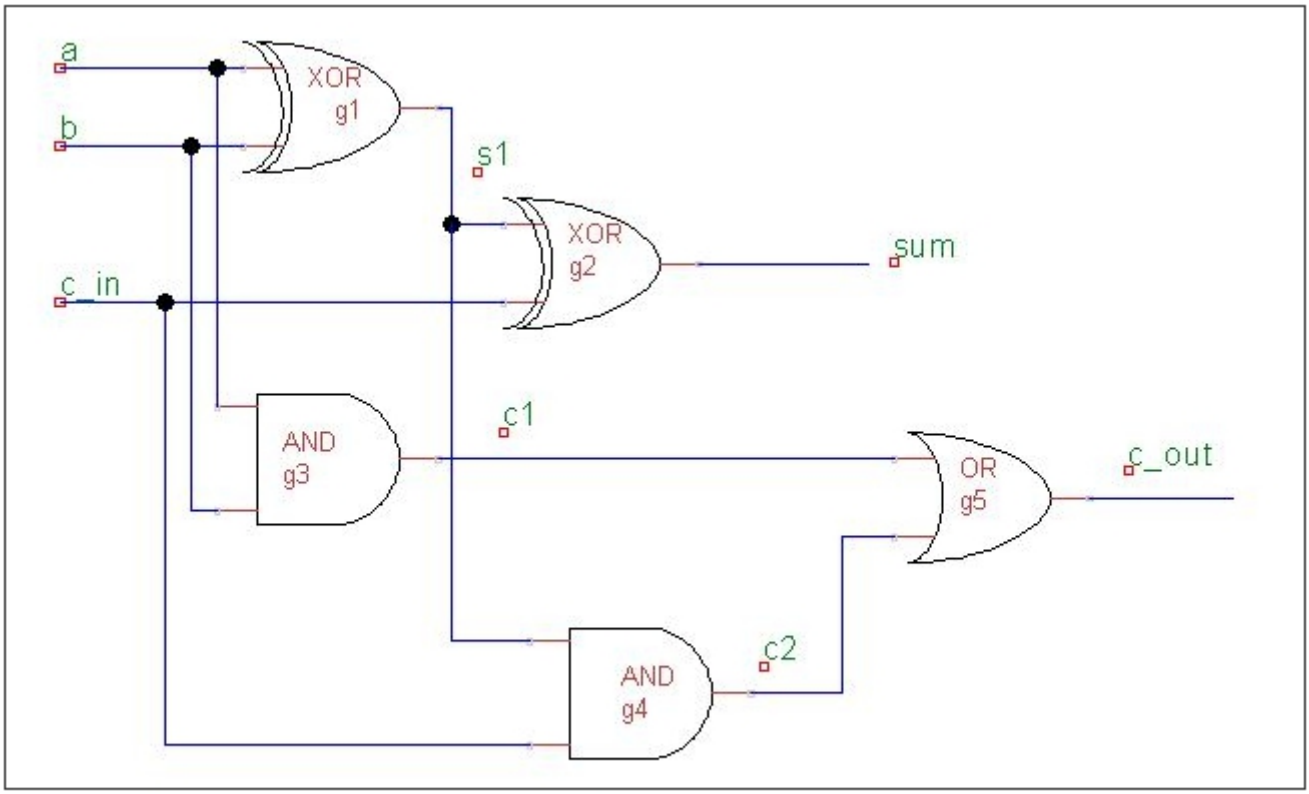
(b) 請描述該 CPU 的指令集，有那些指令，每個指令的功能為何？ (15%)
有
(c) 請用這些指令的組合語言寫出一個可以計算 sum=1+...+n 的結果的程式，其中的 sum 與 n 都是放在記憶體的變數。 (10%)

(d) 請為這些指令的組合語言編上位址 (5%)

(e) 請為這些指令的組合語言編上機器碼。(10%)

參考網頁：
https://dl.dropboxusercontent.com/u/101584453/web/oc/htm/cpu16m.html

always #200 a = a+1;// 每 200ns 讓 a+1 (反向)。 **// 這樣會造成 000 001 010 011 100 101 110 111 輪流出現的形式。** **endmodule**	



全加器電路圖