|  |  |
| --- | --- |
| **Lab 2** | |
| 學號: 111062118 | 姓名: 江佩霖 |

1. **Lab Implementation**
2. Lab2\_Adv1

(1) Block diagram

一張含有 文字, 圖表, 螢幕擷取畫面, 方案 的圖片

AI 產生的內容可能不正確。

(2)-1 透過function的方式撰寫重複的檢查/賦值

function eval\_lit;

        input [2:0] lit;      // literal encoding

        input [2:0] val;      // {A,B,C}

        begin

            case(lit)

                3'b000: eval\_lit = val[2];     // A

                3'b100: eval\_lit = ~val[2];    // ~A

                3'b001: eval\_lit = val[1];     // B

                3'b101: eval\_lit = ~val[1];    // ~B

                3'b010: eval\_lit = val[0];     // C

                3'b110: eval\_lit = ~val[0];    // ~C

                default: eval\_lit = 1'b0;      // invalid literal = false

            endcase

        end

    endfunction

// function: evaluate one clause

    function eval\_clause;

        input [8:0] clause;

        input [2:0] val;

        reg l1, l2, l3;

        begin

            l1 = eval\_lit(clause[8:6], val);

            l2 = eval\_lit(clause[5:3], val);

            l3 = eval\_lit(clause[2:0], val);

            eval\_clause = l1 | l2 | l3;

        end

    endfunction

(2)-2 在test file中，使用for回圈內嵌case的方式依序輸入指定的0~7 clause cases。並且在initial block中加上適當的延遲以符合題目的要求。(詳細說明如註解)

initial begin

        @(negedge clk); // let rst signal start at 15ns

        @(negedge clk);

        rst = 1;

        clause\_1 = 0; clause\_2 = 0; clause\_3 = 0; clause\_4 = 0;

        @(negedge clk);

        @(negedge clk); // let rst signal work for 20ns

        rst = 0;

        for (i = 0; i < 8; i = i + 1) begin

            clause\_1 = 0; clause\_2 = 0; clause\_3 = 0; clause\_4 = 0;

            rst = 0;

            case (i)

                0: begin

                    clause\_1 = 9'b000\_001\_010;

                    clause\_2 = 9'b100\_101\_110;

                    clause\_3 = 9'b000\_101\_110;

                    clause\_4 = 9'b100\_001\_010;

                end

                1: begin

                    clause\_1 = 9'b000\_000\_000;

                    clause\_2 = 9'b100\_100\_100;

                    clause\_3 = 9'b001\_001\_001;

                    clause\_4 = 9'b101\_101\_101;

                end

                //這邊省略2~6 case code

                7: begin

                    clause\_1 = 9'b000\_000\_001;

                    clause\_2 = 9'b001\_001\_010;

                    clause\_3 = 9'b010\_010\_000;

                    clause\_4 = 9'b100\_101\_110;

                end

            endcase

            @(negedge clk); // wait for a clock cycle

        end

        @(posedge clk); // let simulation run til 120000ns.

**$display**("All test cases done.");

**$finish**;

    end

(4)Wave form

一張含有 螢幕擷取畫面, 軟體, 多媒體軟體, 電子產品 的圖片

AI 產生的內容可能不正確。

- Testing process :

(i) check rst work

(ii) check load clause time correct, with function call has right result ( by $display value)

(iii) check output update at right time.

(ix) check all signal in wave form works correctly

1. Lab2\_Adv\_2

(1)Block diagram

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

AI 產生的內容可能不正確。

(2)-1 同Lab2\_1，使用function撰寫重複的檢查/賦值(code略)

(2)-2 如題述，監聽的clock edge需同時檢查clock以及reset的posedge。

always @(posedge clk or posedge rst) begin

// skip code here

end

(2)-3 由於output的update具有時間關係(有順序性，需使用blocking方式賦值)

case (switch\_inputtype)

          2'b00: clause1    = switch\_input;

          2'b01: clause2    = switch\_input;

          2'b10: clause3    = switch\_input;

          2'b11: assign\_reg = switch\_input[2:0]; // only SW2..SW0 used

          default: ;

        endcase

        // note: outputs are computed in combinational block below

        c1\_sat = eval\_clause(clause1, assign\_reg);

        c2\_sat = eval\_clause(clause2, assign\_reg);

        c3\_sat = eval\_clause(clause3, assign\_reg);

        // final satisfaction output

        satisfaction = (c1\_sat && c2\_sat && c3\_sat) ? 1'b1 : 1'b0;

        out[13:10] = (c1\_sat)? 4'b1111 : 4'b0000;

        out[8:5] = (c2\_sat)? 4'b1111 : 4'b0000;

        out[3:0] = (c3\_sat)? 4'b1111 : 4'b0000;

1. **Questions and Discussions**
2. What are the differences between a combinational circuit and a sequential circuit? Please explain how each of them works in detail.

Ans.

Combinational: Outputs depend only on present inputs. No memory. Example: adder, multiplexer.

Sequential: Outputs depend on present inputs and past state stored in flip-flops/latches. Example: counter, FSM.

1. For an unsatisfiable 3-SAT problem, how would you extend your Verilog design to solve the MAX-SAT problem (find a Boolean assignment that satisfies the maximum number of clauses)?

Ans.

Suppose to use the design stucture below :

* Clause evaluator: For each clause, compute whether it is satisfied.
* Popcount: Count how many clauses are satisfied → score.
* Best register: Store highest score and corresponding assignment.
* Control FSM: Generate candidate assignments (brute force with a counter for small n, or heuristic search like WalkSAT for large n).
* Update: If score > best, save as new best. After search ends, output assignment\_best.
* Optimization: Use Gray code to change one variable at a time and update only affected clauses. Parallel evaluation is possible on FPGA for speed.

1. In lab2\_adv\_2, what unexpected behavior might occur if the valid signal is removed? Explain why?

Ans.

Without valid, inputs from switches are sampled asynchronously.

This can cause glitches, partial updates, or metastability in flip-flops.

Outputs may flicker or show inconsistent results while switches change.

valid ensures inputs are only latched at the right clock edge.

1. **Problem Encountered** 
   * 1. Lab2\_1

第一題沒遇到什麼問題，只有最後檢查wave form才發現輸入輸出的clk edge設定還有整個simulation要結束在120000ns等等，因此花了一些時間調整

(以下是我截圖的wave form進展史)。

一張含有 螢幕擷取畫面, 電子產品, 軟體, 多媒體軟體 的圖片

AI 產生的內容可能不正確。一張含有 螢幕擷取畫面, 文字, 軟體, 多媒體軟體 的圖片

AI 產生的內容可能不正確。

一張含有 螢幕擷取畫面, 軟體, 多媒體軟體, 電子產品 的圖片

AI 產生的內容可能不正確。

* + 1. Lab2\_2

設計這個電路時，當下沒有特別想要進行 debug（因為寫錯板子就是不會亮 😊，而且也不太想寫test file去測試…）。

因此有遇到一個問題，最初的設計用了兩個 always，一個主要負責驅動並更新 out 的值，另一個則依據 clk 與 rst 來決定更新與載入新 clause 的時機。不過在實作後發現這樣很難確認 signal 是否被正確驅動，且板子完全沒有亮。因此做了一些調整，將判斷邏輯集中在同一個 always block 裡，直接使用clk, rst驅動所有的信號更新。

1. **Suggestions**

我覺得很棒，沒什麼建議所以附上笑話 :

皮卡丘很愛搗蛋被揍了一頓，會變成什麼?

卡丘，因為他不敢再皮了。