



國立臺北科技大學

數位多媒體晶片設計課程

LAB 1



課堂教授：范育成 博士

一. 實驗名稱

RTL 實驗

二. 實驗目的

以漢明碼為範例讓學生練習撰寫 RTL 程式碼，並以 ncverilog 軟體進行編譯及執行。

三. 實驗過程

將助教提供的漢明碼演算法以 Verilog 軟體撰寫，並以 Cadence 公司之 ncverilog 工具將 RTL 程式碼編譯及執行測試，將測試結果以 Synopsys 公司之 nWave 工具開啟以確認電路執行結果是否正確。

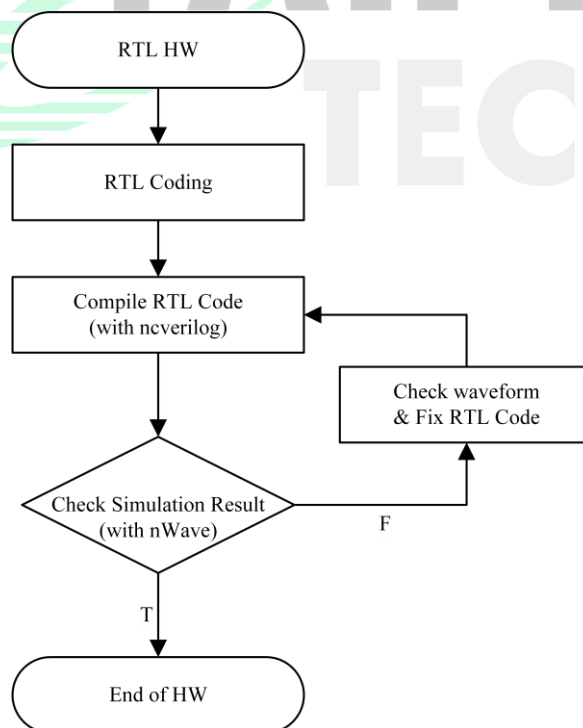


圖 3-1、實驗流程圖

```

module HD(
    // input signals pattern
    code_word1,
    code_word2,
    // output signals
    out_n
);

// input signals pattern
input  [6:0]code_word1;
input  [6:0]code_word2;

// output signals
output reg signed  [5:0]out_n;

// reg
reg [1:0]opt;
reg signed [3:0]w1;    // input word1
reg signed [3:0]w2;    // input word2
reg [1:0]c1;    // circle1
reg [1:0]c2;    // circle2
reg [1:0]c3;    // circle3

always@(*) begin

    // write input data to reg
    w1 = code_word1[3:0];
    w2 = code_word2[3:0];

    /* Check circle is odd or not
    Circle1 contain p1, x1, x2, x3
    Circle2 contain p2, x1, x2, x4
    Circle3 contain p3, x1, x3, x4 */
    c1[1] = code_word1[6] ^ code_word1[3] ^ code_word1[2] ^ code_word1[1];
    c2[1] = code_word1[5] ^ code_word1[3] ^ code_word1[2] ^ code_word1[0];
    c3[1] = code_word1[4] ^ code_word1[3] ^ code_word1[1] ^ code_word1[0];
    c1[0] = code_word2[6] ^ code_word2[3] ^ code_word2[2] ^ code_word2[1];
    c2[0] = code_word2[5] ^ code_word2[3] ^ code_word2[2] ^ code_word2[0];

```

```

c3[0] = code_word2[4] ^ code_word2[3] ^ code_word2[1] ^ code_word2[0];

// Check which input are incorrect and output to opt
// input code_word1
if(c1[1] & c2[1] & !(c3[1])) begin
    // if c1 and c2 are odd -> x2 is incorrect
    opt[1] = w1[2];
    w1[2] = !(w1[2]);
end
if(!(c1[1]) & c2[1] & c3[1]) begin
    // if c2 and c3 are odd -> x4 is incorrect
    opt[1] = w1[0];
    w1[0] = !(w1[0]);
end
if(c1[1] & !(c2[1]) & c3[1]) begin
    // if c1 and c3 are odd -> x3 is incorrect
    opt[1] = w1[1];
    w1[1] = !(w1[1]);
end
if(c1[1] & c2[1] & c3[1]) begin
    // if all circle are odd -> x1 is incorrect
    opt[1] = w1[3];
    w1[3] = !(w1[3]);
end
if(c1[1] & !(c2[1]) & !(c3[1]))
    // if only c1 are odd -> p1 is incorrect
    opt[1] = code_word1[6];
if(!(c1[1]) & c2[1] & !(c3[1]))
    // if only c2 are odd -> p2 is incorrect
    opt[1] = code_word1[5];
if(!(c1[1]) & !(c2[1]) & c3[1])
    // if only c1 are odd -> p3 is incorrect
    opt[1] = code_word1[4];

// input code_word2
if(c1[0] & c2[0] & !(c3[0])) begin
    // if c1 and c2 are odd -> x2 is incorrect
    opt[0] = w2[2];

```

```

        w2[2] = !(w2[2]);
    end
    if(!(c1[0]) & c2[0] & c3[0]) begin
        // if c2 and c3 are odd -> x4 is incorrect
        opt[0] = w2[0];
        w2[0] = !(w2[0]);
    end
    if(c1[0] & !(c2[0]) & c3[0]) begin
        // if c1 and c3 are odd -> x3 is incorrect
        opt[0] = w2[1];
        w2[1] = !(w2[1]);
    end
    if(c1[0] & c2[0] & c3[0]) begin
        // if all circle are odd -> x1 is incorrect
        opt[0] = w2[3];
        w2[3] = !(w2[3]);
    end
    if(c1[0] & !(c2[0]) & !(c3[0]))
        // if only c1 are odd -> p1 is incorrect
        opt[0] = code_word2[6];
    if(!(c1[0]) & c2[0] & !(c3[0]))
        // if only c2 are odd -> p2 is incorrect
        opt[0] = code_word2[5];
    if(!(c1[0]) & !(c2[0]) & c3[0])
        // if only c1 are odd -> p3 is incorrect
        opt[0] = code_word2[4];

    // Follow pattern to define output
    case(opt)
        2'b00: out_n = 2 * w1 + w2;
        2'b01: out_n = 2 * w1 - w2;
        2'b10: out_n = w1 - 2 * w2;
        2'b11: out_n = w1 + 2 * w2;
    endcase
end

endmodule

```

四. 模擬結果相關 Report(附圖)

本次實驗以 ssh 遠端連線至 140.124.72.29 伺服器完成，主要模擬結果如下圖所示，圖 4-1 為使用 ncoverilog 軟體模擬並通過所有測試內容之截圖，圖 4-2 為模擬完成後以 nWave 軟體開啟波形檔並擷取一部份輸出內容。

```
Pass Pattern NO.      96
Pass Pattern NO.      97
Pass Pattern NO.      98
Pass Pattern NO.      99

-----
-- Congratulations !! --
-- Simulation PASS!!   --
-----

Simulation complete via $finish(1) at time 8080 NS + 0
ncsim> exit
[mm53@islabx6 RTL_HW]$
```

圖 4-1、Pass Pattern 截圖

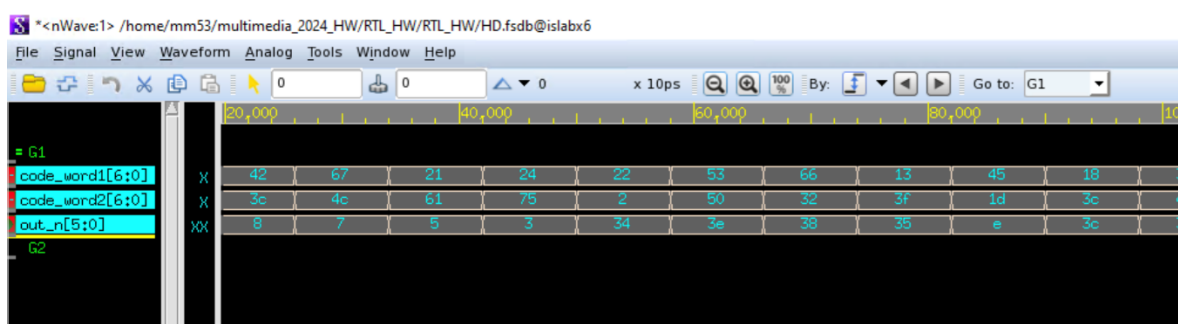


圖 4-2、nWave 軟體開啟模擬波形截圖

五. 心得

本次實驗主要目的為學習如何撰寫 VerilogHDL 程式碼，並透過簡單的漢明碼錯誤校正功能引導出一套 RTL 邏輯，進而設計出可以通過測試的數位組合邏輯電路。由於已經是碩二生，對基本的數位晶片設計流程已經算是相當熟悉，從最前端的演算法設計，接著將設計完成的演算法透過 HDL 語言轉換為 RTL 程式碼，而後將 RTL 程式碼透過合成軟體轉換為邏輯閘層級的線路圖，此後將邏輯閘電路透過自動繞線軟體將晶片完成佈局，此一完整流程為數位晶片需要將晶片生產前必經的一段路，且其中尚有如 DFT、FPGA 驗證等等應該要做的過程在上述流程中被簡略。實驗過程十分順利，雖一開始不熟悉其他實驗室的伺服器環境導致原地打轉了一段時間，詢問助教後才得知設定的方式錯誤，且軟體開啟需要給他一段時間，而後又透過軟體發現自己撰寫 RTL 時發生一點邏輯錯誤，導致模擬結果沒辦法跑完，幸好這些問題都在詢問助教後一一解決，十分感謝助教的協助。由於先前都是透過 TSRI

課程及實驗室學長口述完成流程，這是自己第一次在其他環境下完成整個流程，後面有嘗試將產生的 netlist 放到自己實驗室進行合成及後續的驗證，也順利完成後面的模擬。

