

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ



BÁO CÁO MÔN HỌC
THIẾT KẾ VLSI

Đề tài:

Thiết kế bộ counter bộ đếm 3 bit nhị phân (Dolphin)

Nhóm: 46

Nhóm sinh viên thực hiện:

Lưu Khánh Duy – 20213847

Nguyễn Tuấn Anh – 20213804

Giảng viên hướng dẫn:

Thầy Vũ Quang Thắng

Hà Nội, 2025

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG 1. YÊU CẦU KỸ THUẬT	2
1.1. Yêu cầu chung.....	2
1.2. Đề bài	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	4
2.1. Quy trình thiết kế vi mạch.....	4
2.1.1. System specification (thiết kế đặc điểm kỹ thuật)	4
2.1.2. Architectural design (thiết kế kiến trúc).....	5
2.1.3. RTL design (thiết kế RTL).....	5
2.1.4. Design verification.....	6
2.1.5. Synthesis	6
2.1.6. Static Timing Analysis.....	7
2.1.7. Physical design (thiết kế vật lý)	7
2.1.8. Sign-off	8
CHƯƠNG 3. PHÂN TÍCH, XÂY DỰNG VÀ THIẾT KẾ	8
3.1. Phân tích yêu cầu hệ thống	8
3.2. RTL design (thiết kế RTL).....	10
3.3. Design verification (kiểm thử thiết kế)	10
3.4. Synthesis (tổng hợp RTL).....	12
3.4.1. Stick Diagram	12
3.4.2. Circuit Diagram.....	13
3.5. Physical design (thiết kế vật lý)	14
3.5.1. Thiết kế layout cho từng khối	14
3.5.2. Thiết kế Layout mạch Top.....	16
TÀI LIỆU THAM KHẢO.....	17

LỜI NÓI ĐẦU

Trong thời đại công nghệ hiện nay, sự phát triển vượt bậc của ngành điện tử đã mở ra những tiềm năng vô tận cho các ứng dụng trong đời sống và công nghiệp. Một trong những bước ngoặt lớn làm thay đổi diện mạo ngành công nghiệp điện tử chính là sự ra đời của công nghệ vi mạch tích hợp quy mô lớn (VLSI - Very Large Scale Integration).

Vi mạch VLSI là công nghệ tích hợp hàng triệu đến hàng tỷ transistor trên một chip silicon đơn lẻ. Nhờ khả năng tích hợp cao, VLSI không chỉ giúp thu nhỏ kích thước của các thiết bị điện tử mà còn cải thiện hiệu suất, giảm tiêu thụ năng lượng và giảm chi phí sản xuất. Công nghệ này đã trở thành nền tảng cho sự phát triển của các hệ thống số, vi xử lý, trí tuệ nhân tạo, thiết bị IoT và các hệ thống nhúng.

Trong những năm gần đây Công nghệ VLSI không chỉ đóng vai trò quan trọng trong việc thu nhỏ kích thước thiết bị điện tử, mà còn cải thiện hiệu suất, giảm tiêu thụ năng lượng và chi phí sản xuất. Đây là nền tảng cốt lõi cho sự phát triển của các hệ thống nhúng, trí tuệ nhân tạo, thiết bị IoT, và nhiều ứng dụng hiện đại khác.

Việc nghiên cứu và ứng dụng công nghệ VLSI đòi hỏi kiến thức sâu rộng về thiết kế mạch điện tử, tối ưu hóa hiệu năng, kiểm soát năng lượng và khả năng vận hành ổn định. Đồng thời, các công cụ phần mềm chuyên dụng như Cadence, Synopsys hay Mentor Graphics đóng vai trò quan trọng trong việc hiện thực hóa ý tưởng từ thiết kế đến sản xuất.

Do đó, chúng em quyết định thực hiện đề tài: “Thiết kế bộ counter bộ đếm 3 bit nhị phân” do nhà trường liên kết với doanh nghiệp Dolphin Technology. Mục tiêu của chúng em là tìm hiểu và phân tích các khía cạnh quan trọng của công nghệ VLSI, từ cơ sở lý thuyết, phương pháp thiết kế, đến ứng dụng trong thực tế.

CHƯƠNG 1. YÊU CẦU KỸ THUẬT

1.1. Yêu cầu chung

- Viết đặc tả kỹ thuật cho thiết kế (Specification):
 - o Sơ đồ tổng quan tín hiệu vào – ra (Input – Output Block Diagram)
 - o Sơ đồ kiến trúc chi tiết của thiết kế mức RTL (Architecture)
 - o Sơ đồ timing (Timing Diagram) mô tả đầy đủ các chức năng và trường hợp xảy ra
- Triển khai thiết kế (thiết kế RTL), sử dụng ngôn ngữ Verilog hoặc SystemVerilog
- Kiểm thử thiết kế (Kiểm thử RTL) bằng phương pháp kiểm thử trực tiếp. Yêu cầu kiểm thử đầy đủ các chức năng và trường hợp có thể xảy ra.
- Tổng hợp thiết kế, kiểm thử kết quả tổng hợp được (Kiểm thử Netlist):
 - o Thiết kế sau khi tổng hợp gồm bao nhiêu cell? Nhận biết chức năng của từng cell.
 - o Kiểm thử Netlist cho kết quả mô phỏng trùng khớp với kiểm thử RTL
- Vẽ đầy đủ Stick Diagram cho các cổng logic cơ bản được sử dụng trong thiết kế
 - o ít break od/poly, tối thiểu số đường kết nối
 - o Output ngắn, thẳng
 - o Tín hiệu ngắn gọn, không cắt nhau
 - o Nhiều đường Power nhất có thể
 - o Tính thẩm mỹ (nhìn sạch đẹp, màu rõ ràng nếu có), stick kèm theo mạch nguyên lý + các path sharing,
- Vẽ Layout
 - o Layout đầy đủ cho từng cell cơ bản
 - Đúng chuẩn template của standard cell * Chỉ được dùng cao nhất đến Metal 2
 - Đảm bảo đầu đủ các tiêu chí đã nêu ở phần Stick Diagram
 - LVS, mạch phải đúng logic cho từng cell cơ bản
 - DRC, đã sửa hết lỗi DRC cho từng cổng logic cơ bản
 - o Layout Mạch Top (Toàn bộ thiết kế)
 - Mạch top yêu cầu nhỏ nhất có thể (xếp khít, không có khoảng trống giữa 2 cell cạnh nhau), cố gắng bố trí theo hình vuông
 - Các cell không được đè lên nhau, chỉ được tiếp xúc bởi layer bbox (id/datatype: 108;0)
 - Dùng cao nhất đến Metal 5, càng dùng ít layer metal càng tốt.
 - Không được đi metal vượt khỏi vùng bbox mà được hợp bởi bbox của các cell khi đi dây trên mạch top
 - Pass LVS
 - Pass DRC

1.2. Đề bài

Thiết kế bộ đếm counter 3 bit nhị phân với đầu vào – ra, và chức năng như sau:

Bảng 1. Mô tả tín hiệu Input – Output của bộ đếm dti_bincnt_ckprn

Tên Tín Hiệu	Độ Rộng	Chiều	Mô Tả
clk	1	Input	Clock, mạch tuần tự hoạt động đồng bộ theo sườn dương của clock này
reset_n	1	Input	Reset không đồng bộ, tích cực mức thấp
count_to	3	Input	Giá trị cần đếm, là một số nguyên không dấu
load	1	Input	Nạp giá trị cần đếm
count_en	1	Input	Cho phép đếm
done	1	Output	Cờ báo hiệu hoàn thành việc đếm

Yêu cầu chức năng:

- Mạch đồng bộ hoạt động theo sườn dương của clock đầu vào.
- Đầu ra done được chốt bằng Flip Flop (Nói cách khác, done được nối trực tiếp với đầu ra Q hoặc QN của 1 Flip Flop).
- Giá trị của bộ đếm, tạm gọi là count, được nạp mới hoặc thay đổi giá trị tại sườn dương clock theo quy tắc trong bảng sau:

Bảng 2. Quy tắc đếm dti_bincnt_ckprn

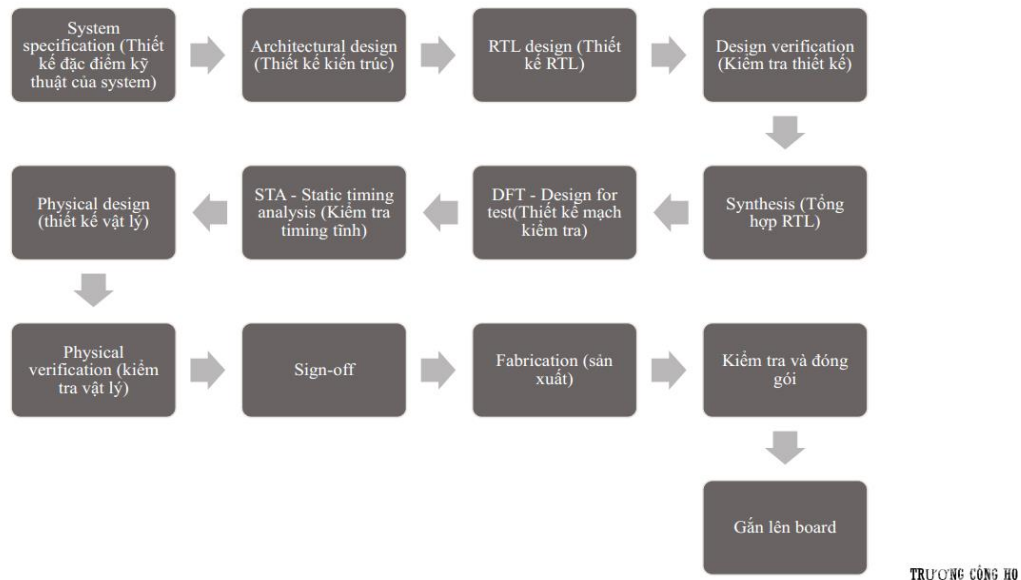
load	count_en	done	count
1'b1	x	x	Nạp mới, giá trị nạp được xác định bởi count_to
1'b0	1'b1	1'b0	Giảm 1 đơn vị
1'b0	1'b1	1'b1	Giữ nguyên giá trị
1'b0	1'b0	x	Giữ nguyên giá trị
Chú ý: x = don't care			

- Trước khi có **load** = 1, cờ **done** giữ giá trị 1'b0. Cờ **done** bật lên 1'b1 cùng lúc với **count** = 0 và giữ nguyên cho tới lần **load** tiếp theo.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Quy trình thiết kế vi mạch

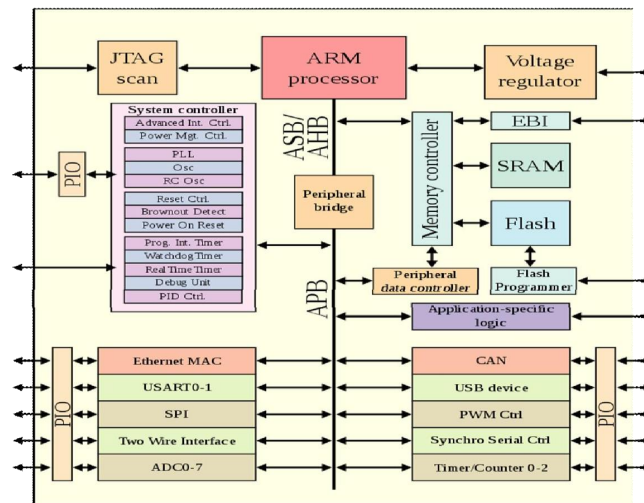
SƠ ĐỒ KHỐI QUY TRÌNH THIẾT KẾ VI MẠCH



2.1.1. System specification (thiết kế đặc điểm kỹ thuật)

- Ước tính kích thước vật lý (physical demension)
 - Công suất tiêu thụ (Power consumption, frequency, voltage)
 - Công nghệ (Process) – 12nm, 3nm, 22nm ...
 - Chức năng (functionality)
- Những yếu tố trên phải đáp ứng được yêu cầu của thị trường, về công nghệ và tính thương mại. Kết quả cuối cùng ước tính được chức năng, kích thước, hiệu suất, công suất.

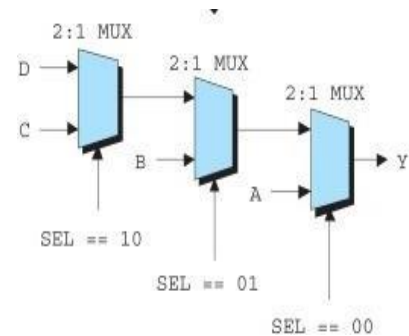
2.1.2. Architectural design (thiết kế kiến trúc)



- Đưa ra những giải pháp, quyết định cho việc sử dụng tài nguyên phần cứng (hardware resources) cần thiết để tính toán, xử lý dữ liệu cùng với việc tổ chức sự tương tác giữa chúng để ứng được các thông số kỹ thuật ở bước “System specification”
- Định nghĩa các khối chức năng
- Tính, lựa chọn và quyết định các giải pháp cho từng khối, chia thành từng miền (domain)
- Sắp xếp tổ chức các khối chức năng

2.1.3. RTL design (thiết kế RTL)

- Thiết kế “Specification”, tài liệu mô tả về chức năng của module
- Thiết kế mạch logic
- Code RTL
- Kiểm tra
- Có thể Synthesis
- Viết SDC, “Synopsys design constraint” các định nghĩa, mối liên hệ về timing
- Viết UPF/CPF, định nghĩa, mối liên hệ về power



```

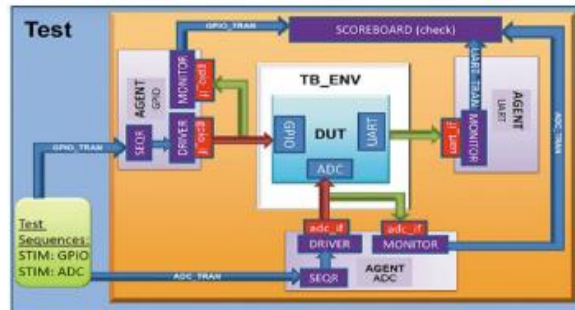
if      SEL == 00" then Y = A;
elseif SEL == 01" then Y = B;
elseif SEL == 10" then Y = C;
else
                                Y = D;
end if;

```

2.1.4. Design verification

DESIGN VERIFICATION (KIỂM TRA THIẾT KẾ)

- Nghiên cứu “Specification”
- Tạo môi trường kiểm tra
- Tạo mẫu test (Test pattern)
- Mô phỏng
- Kiểm tra coverage
- Kiểm tra connection “CT - Connection test”, kiểm tra chức năng “UT – Unit test”, kiểm tra ở mức hệ thống “ST – System test”
- Ngôn ngữ sử dụng: C, SystemC, Assembly, System Verilog

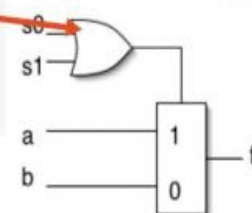


2.1.5. Synthesis

SYNTHESIS (TỔNG HỢP RTL)

- Tổng hợp từ dạng RTL sang dạng gate đáp ứng được
 - Hiệu suất cao nhất - Performace
 - Công suất thấp nhất - Power
 - Kích thước nhỏ nhất – Area
 - Kết hợp của những yếu tố trên
- Các yêu cầu cụ thể cho từng thiết kế về PP/ thực tế hoặc các yếu tố vật lý khác

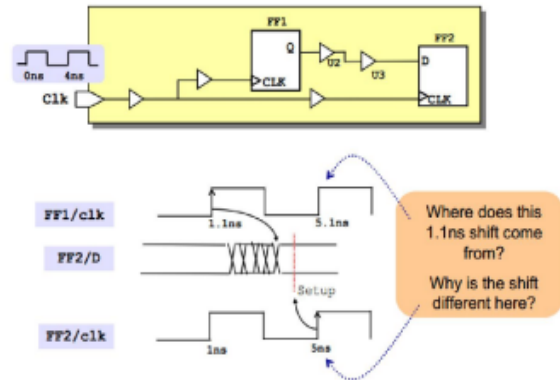
```
module foo (a,b,s0,s1,f);  
  input [3:0] a;  
  input [3:0] b;  
  input s0,s1;  
  output [3:0] f;  
  reg f;  
  
  always @ (a or b or s0 or s1)  
  if (s0 && s1 || s0)  
    f=a;  
  else  
    f=b;  
endmodule
```



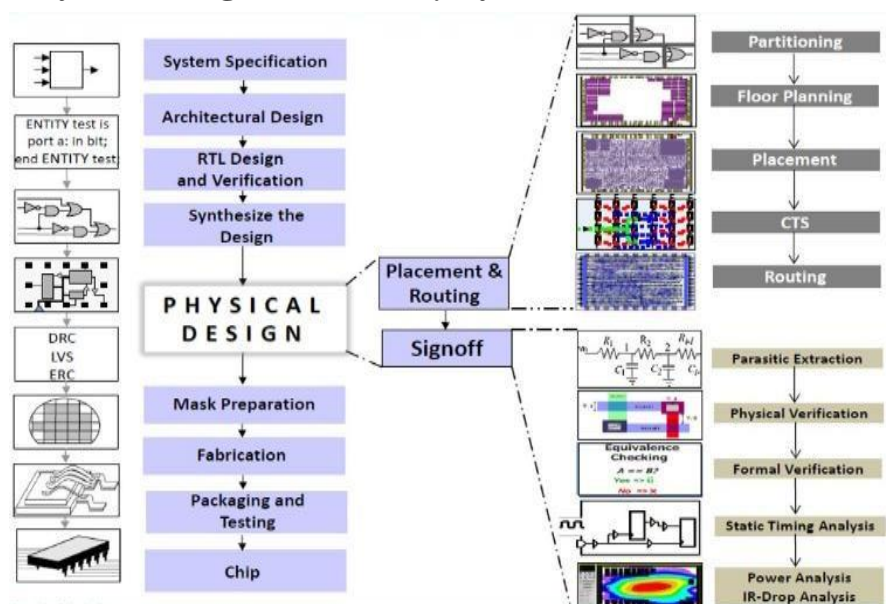
2.1.6. Static Timing Analysis

STA - STATIC TIMING ANALYSIS (KIỂM TRA TIMING TĨNH)

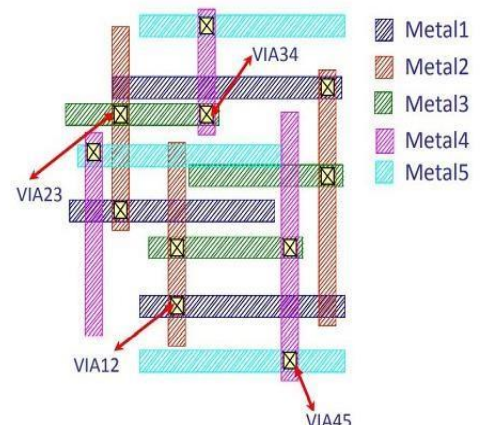
- Nghiên cứu mối quan hệ về timing của thiết kế
- Kiểm tra thiết kế có đáp ứng được tần số yêu cầu hay không trong những điều kiện và thay đổi của môi trường khác nhau. (-40c, 125c, min/max voltage, process condition)
- Kiểm tra STA “design rule” DRC
- Theo dõi, kiểm tra, đề xuất phương án để cải thiện timing của thiết kế cho đến khi sign-off



2.1.7. Physical design (thiết kế vật lý)



- Sắp xếp các khối chức năng (Floorplan)
- Thiết kế lưới nguồn
- Đặt cell (placement)
- Clock Tree Synthesis (CTS)
- Đi dây (routing)
- Sửa những sai phạm về định thời (fix timing)
- Phân tích công suất (Power) và sụt áp (IR drop)
- Sắp xếp các khối chức năng (Floorplan)



2.1.8. Sign-off

SIGN-OFF

▪ Logical Checks

- LEC (Logical Equivalence Checks)
- Post Layout STA

▪ Physical Checks

- LVS (Layout vs Schematics)
- DRC (Design Rule Checks)
- ERC (Electrical Rule Check)
- Antenna Check

▪ Power Checks

- Dynamic IR
- EM (Electromigration)

Tham khảo đường dẫn:

<http://vlsibegin.blogspot.com/p/basic-sign-off.html>

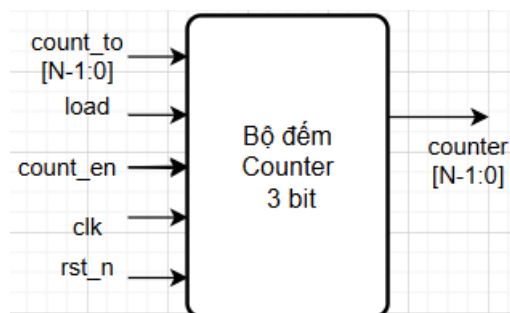
CHƯƠNG 3. PHÂN TÍCH, XÂY DỰNG VÀ THIẾT KẾ

3.1. Phân tích yêu cầu hệ thống

Bảng 1. Mô tả tín hiệu Input – Output của bộ đếm dti_bincnt_ckprn

Tên Tín Hiệu	Độ Rộng	Chiều	Mô Tả
clk	1	Input	Clock, mạch tuần tự hoạt động đồng bộ theo sườn dương của clock này
reset_n	1	Input	Reset không đồng bộ, tích cực mức thấp
count_to	3	Input	Giá trị cần đếm, là một số nguyên không dấu
load	1	Input	Nạp giá trị cần đếm
count_en	1	Input	Cho phép đếm
done	1	Output	Cờ báo hiệu hoàn thành việc đếm

- Sơ đồ tổng quan tín hiệu vào – ra (Input – Output Block Diagram) :

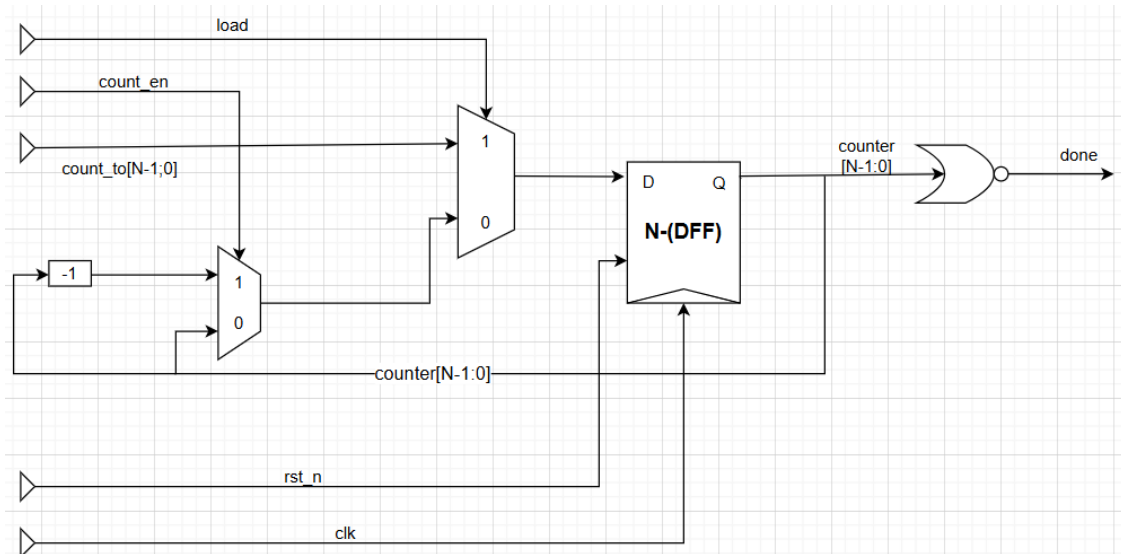


Bảng 2. Quy tắc đếm dti_bincnt_ckpm

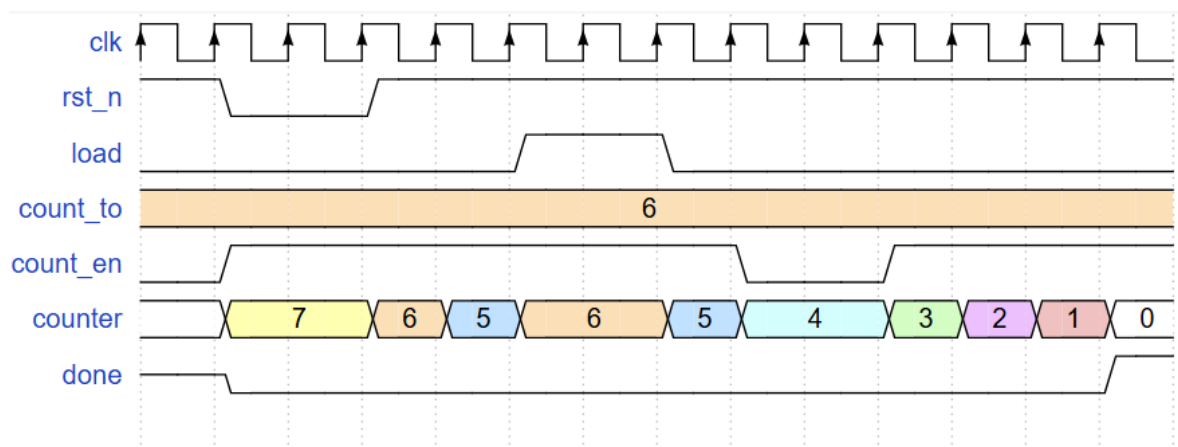
load	count_en	done	count
1'b1	x	x	Nạp mới, giá trị nạp được xác định bởi count_to
1'b0	1'b1	1'b0	Giảm 1 đơn vị
1'b0	1'b1	1'b1	Giữ nguyên giá trị
1'b0	1'b0	x	Giữ nguyên giá trị

Chú ý: x = don't care

- Sơ đồ kiến trúc chi tiết của thiết kế mức RTL (Architecture) :



- Sơ đồ timing (Timing Diagram) mô tả đầy đủ các chức năng:



3.2. RTL design (thiết kế RTL)

```
1 module counter (clk,rst_n,load,count_to,count_en,done);
2     parameter N = 3;
3     input  clk, rst_n, load, count_en;
4     input  [N-1:0] count_to;
5     output reg done;
6
7     reg [N-1:0] counter;
8
9     // Luôn chạy khi có cạnh lên của xung clock
10    always @(posedge clk or negedge rst_n) begin
11        if (!rst_n) begin
12            counter <= 3'b111;
13            done <= 1'b0;
14        end else begin
15            if (load) begin
16                counter <= count_to;
17                done <= 1'b0;
18            end
19            else if (count_en) begin
20                if (counter == 3'b001) begin
21                    counter <= 3'b000; // Giảm count xuống 0
22                    done <= 1'b1; // Thiết lập done thành 1 ngay lập tức
23                end
24                else if (counter == 3'b000) begin
25                    done <= 1'b1; // Nếu count đã là 0, giữ done ở 1
26                end
27                else begin
28                    counter <= counter - 1;
29                    done <= 1'b0;
30                end
31            end
32        end
33    end
34 endmodule
```

3.3. Design verification (kiểm thử thiết kế)

- Bằng phương pháp kiểm thử trực tiếp:

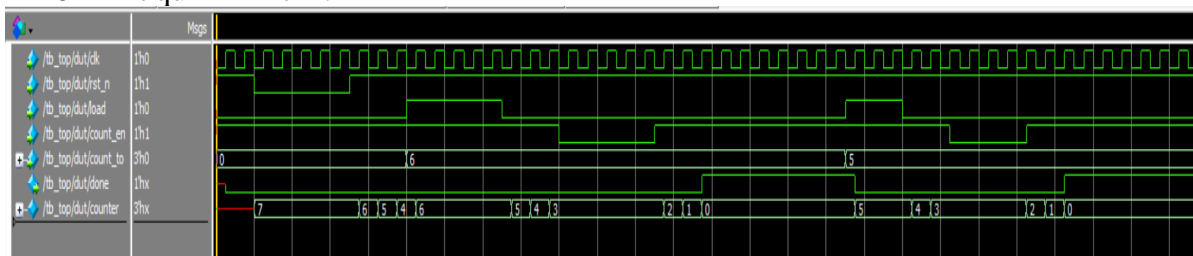
```
1 `timescale 10ns/1ps
2 module tb;
3     reg clk,reset_n, load, count_en;
4     reg [2:0] count_to;
5     wire done;
6     wire [2:0] counter;
7     counter dut (
8         .clk(clk),
9         .count_to(count_to),
10        .reset_n(reset_n),
11        .load(load),
12        .count_en(count_en),
13        .counter(counter),
14        .done(done)
15    );
16    task waitforclk (input integer n);
17        repeat (n)
18            @(posedge clk);
19    endtask
```

```

21 task preload_ (input [2:0] n);
22 begin
23     load = 1;
24     count_en = 0;
25     count_to = n;
26     repeat (3)
27         @(posedge clk);
28         load = 0;
29     count_en = 1;
30 end
31 endtask
32
33 task stopforclk (input integer n);
34 begin
35     count_en = 0;
36     repeat (n)
37         @(posedge clk);
38     count_en = 1;
39 end
40 endtask
41
42 initial begin
43     clk = 0;
44     forever
45         #1) clk = ~clk;
46 end
47
48
49 initial begin
50     $dumpfile("tb.vcd");
51     $dumpvars();
52     load = 0; count_to = 0; count_en = 1; reset_n = 0;
53     waitforclk(3);
54     reset_n = 1;
55     waitforclk(2);
56     preload_(5);
57     waitforclk(20);
58     stopforclk(10);
59     preload_(6);
60     waitforclk(20);
61     $finish();
62 end
63 endmodule

```

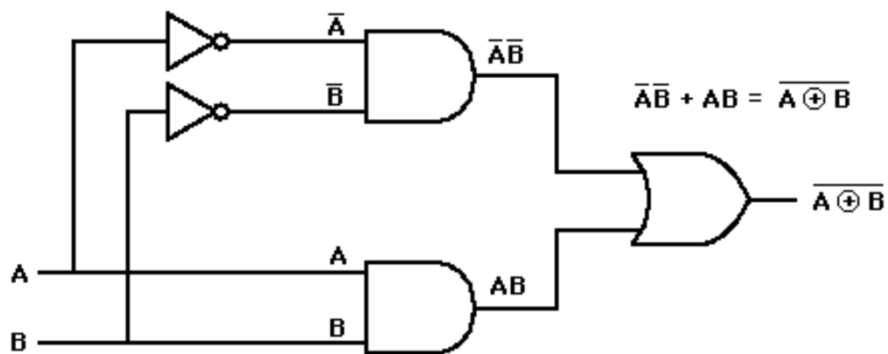
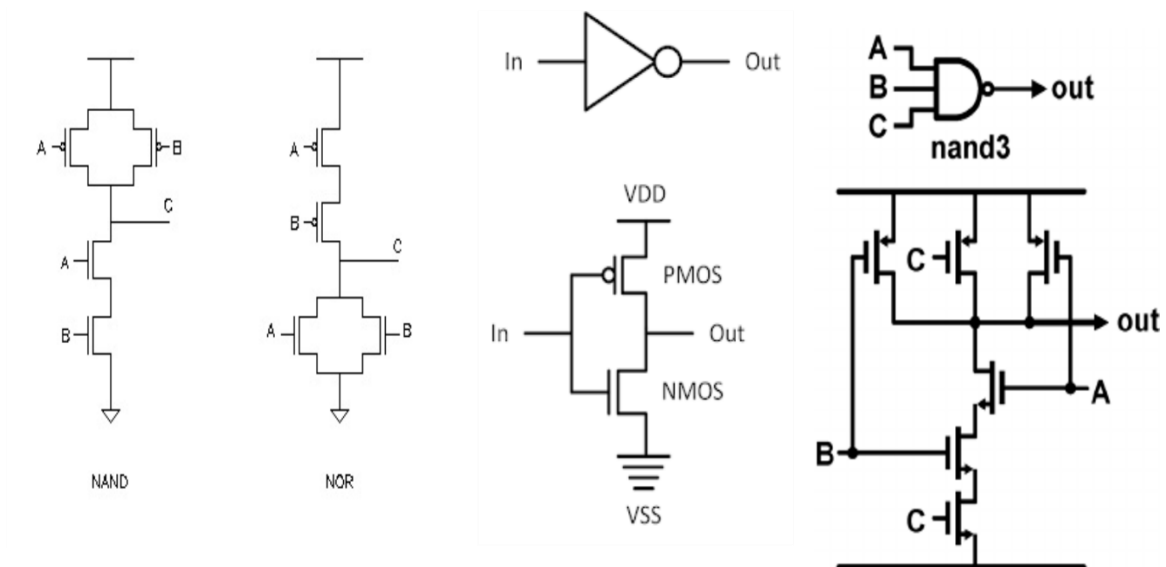
○ Kết quả kiểm thử:



3.4. Synthesis (tổng hợp RTL)

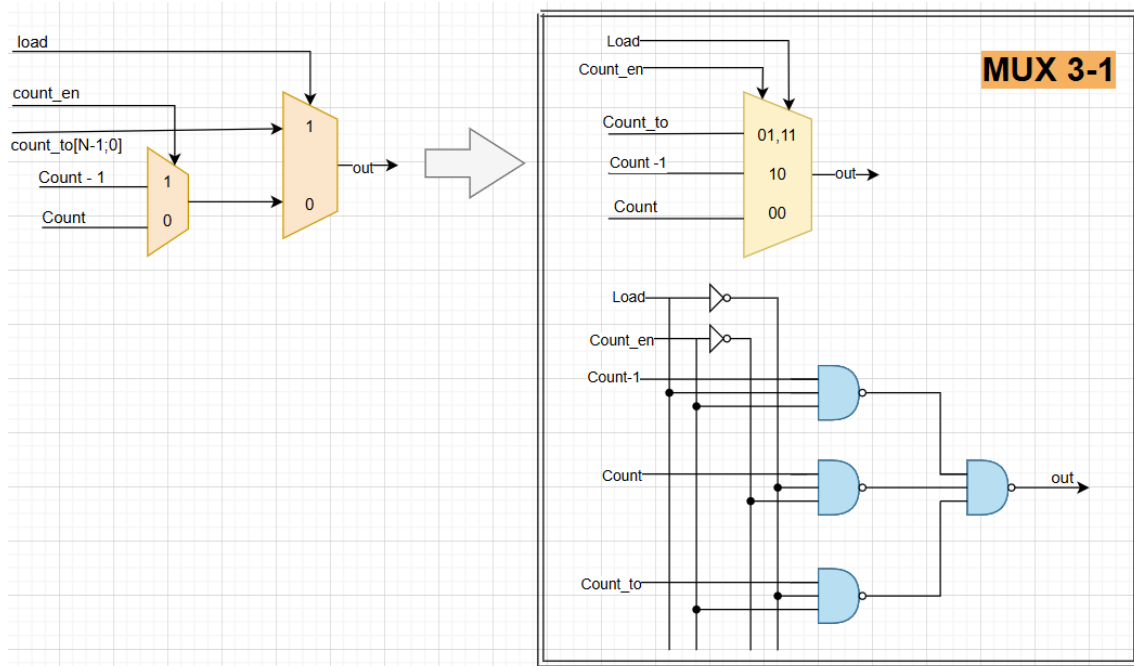
3.4.1. Stick Diagram

- Các cổng logic cơ bản được sử dụng trong thiết kế:

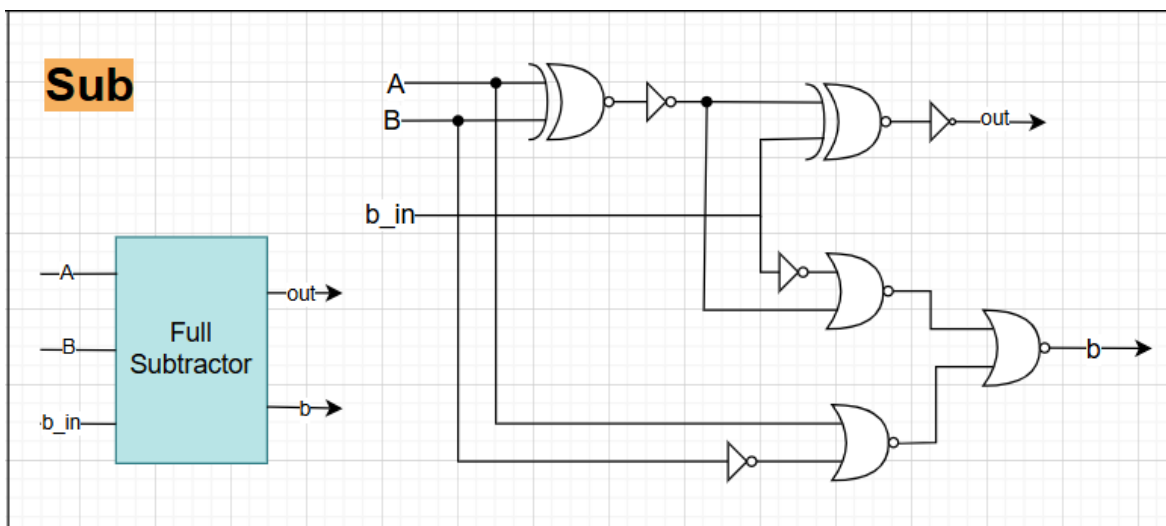


3.4.2. Circuit Diagram

- Khối Mux3-1

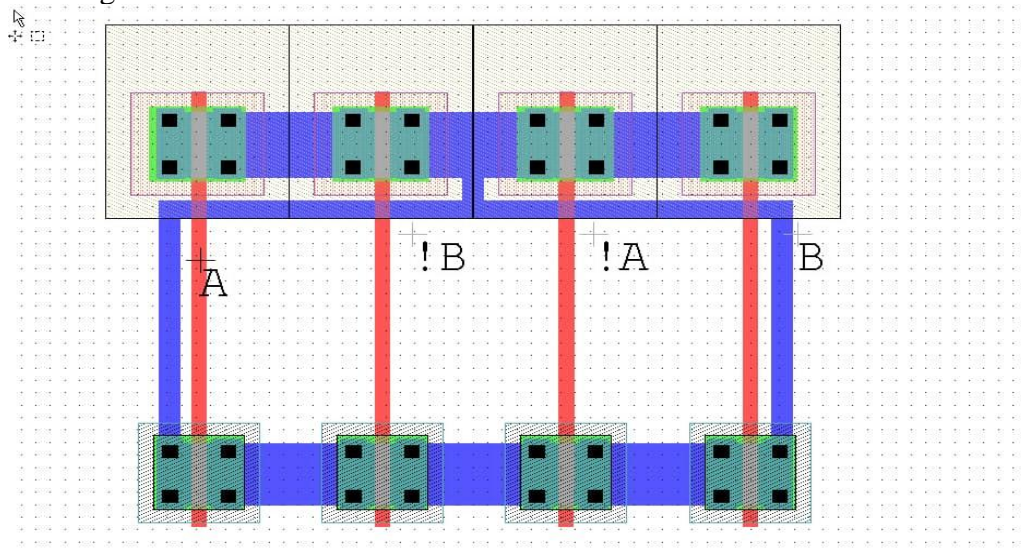


- Khối Full Subtractor

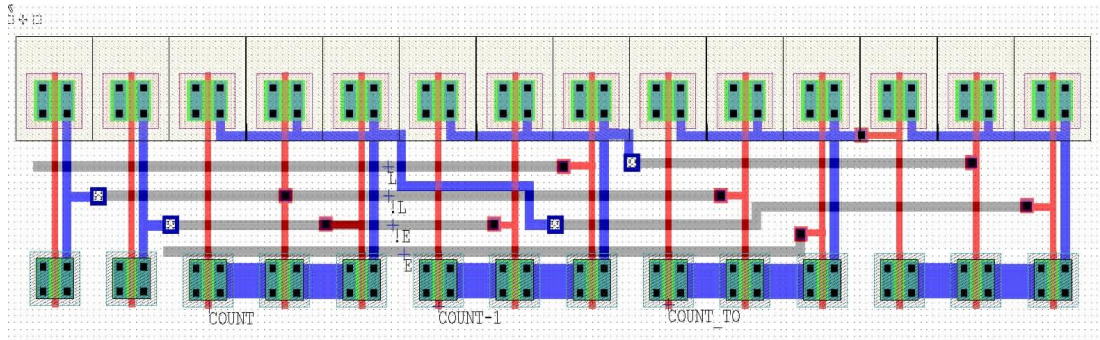


- ## D-FF

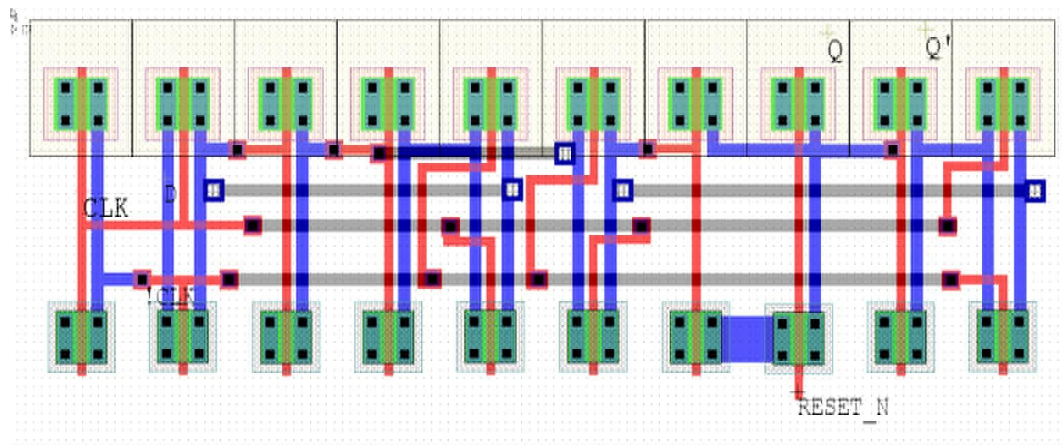
- Cổng Nxor



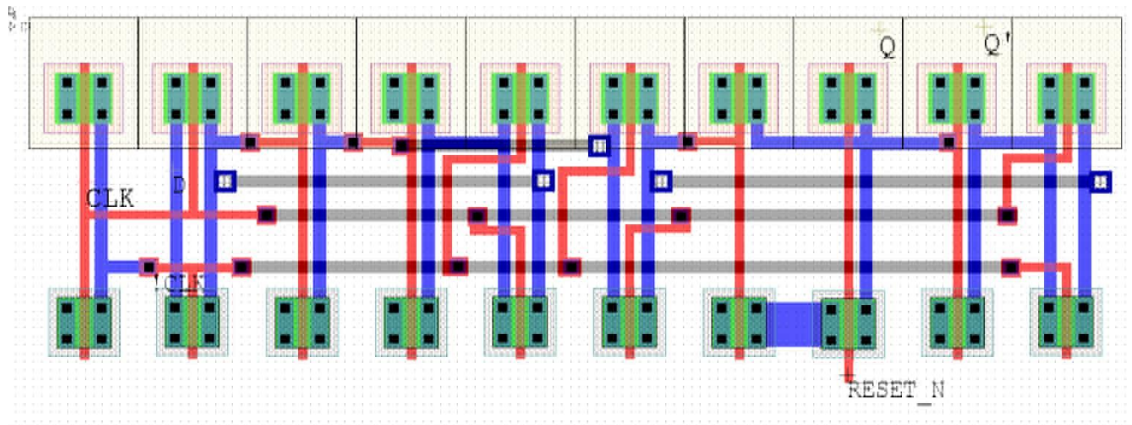
- Khối Mux 3-1



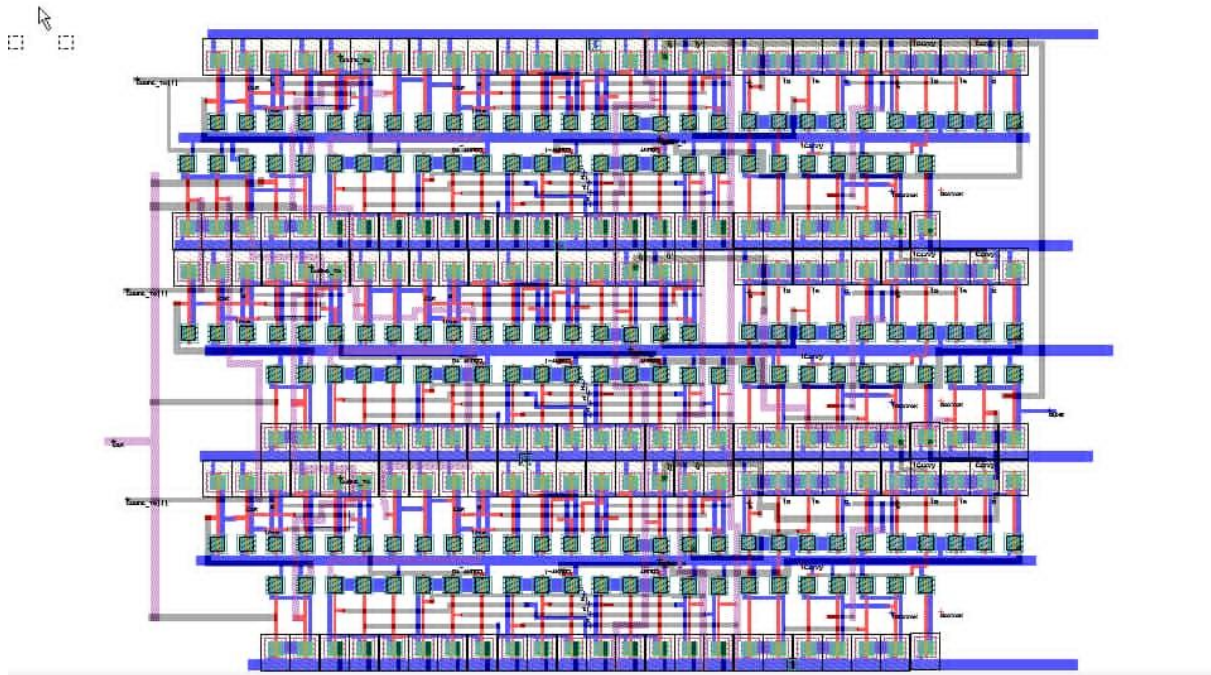
- Khối Full Subtractor



- Khối D Flip-Flop



3.5.2. Thiết kế Layout mạch Top



Hình mạch layout của hệ thống

Và cuối cùng là kiểm tra:

- DRC (design rules check): Kiểm tra mạch có bị ngắn mạch hay hở mạch.
- LVS (layout versus schematic): Kiểm tra xem phần logic có đúng với phần layout không.

TÀI LIỆU THAM KHẢO

Verilog HDL và ứng dụng, Nhà xuất bản đại học Quốc gia TP.Hồ Chí Minh, 2014.

- <https://vlsi-backend-adventure.com/cts.html>
- <https://www.eng.biu.ac.il/temanad/files/2018/11/Lecture-3-Synthesis-Part-1.pdf>
- <https://www.vlsiguru.com/pd-basics-clock-tree-synthesis/>
- <https://vlsi-backend-adventure.com/routing.html>
- <http://vlsibegin.blogspot.com/p/basic-sign-off.html>

