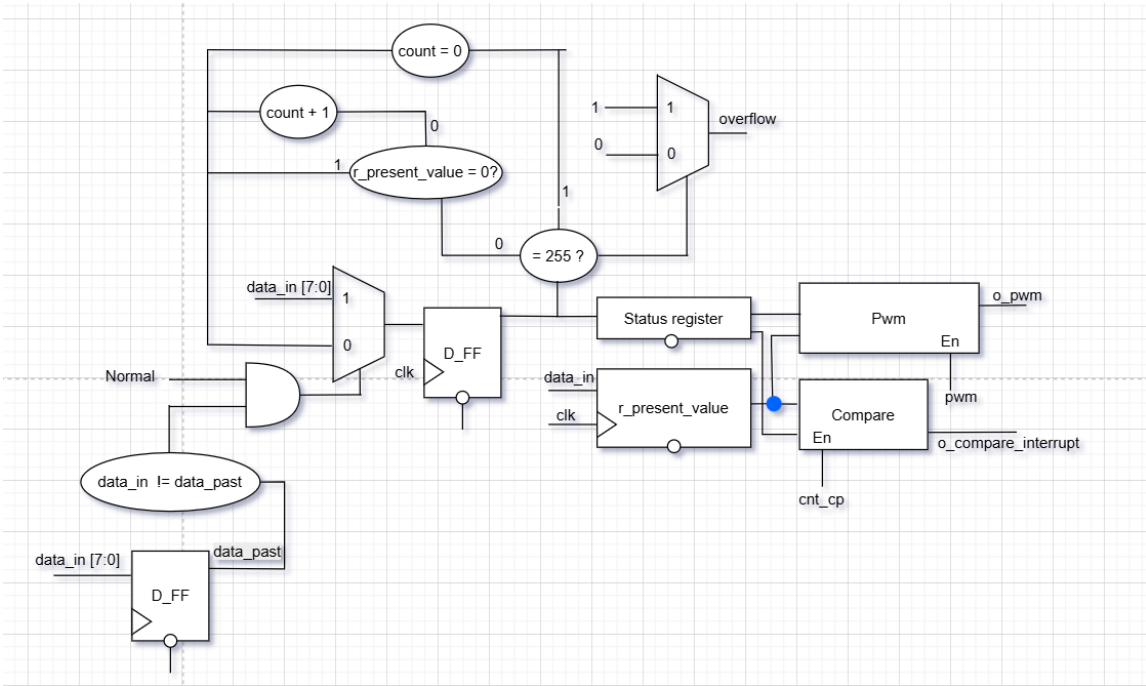
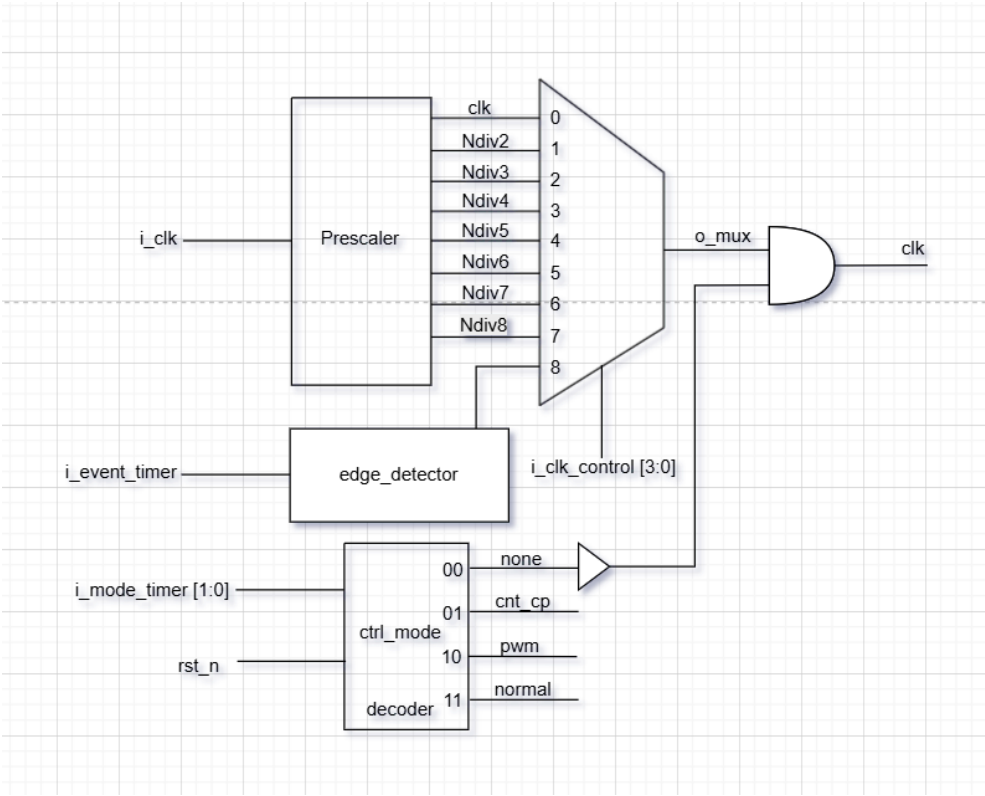


Tổng quan:

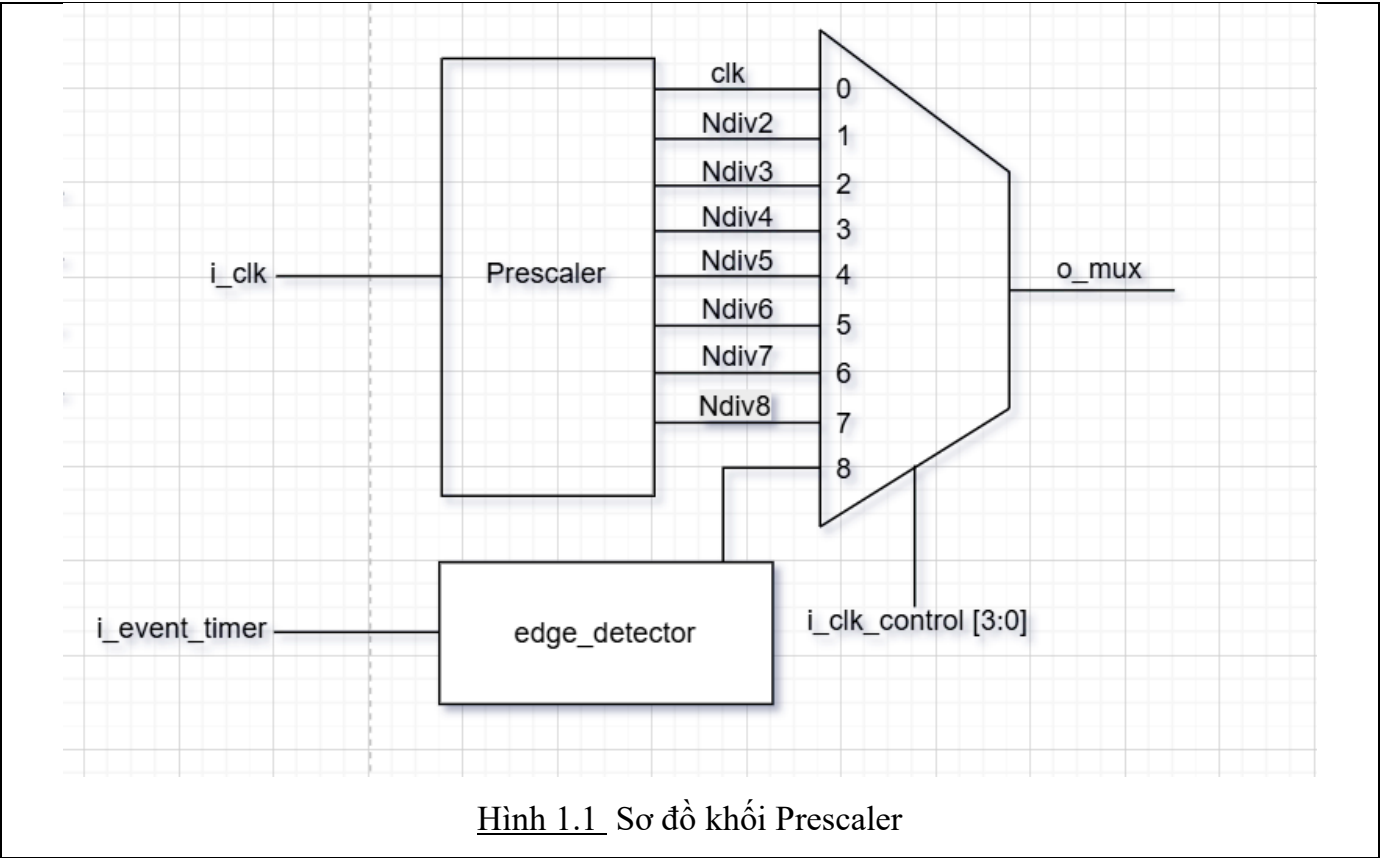


1. Prescaler

1.1 Các tín hiệu

Khối prescaler có nhiệm vụ dẫn tín hiệu clock từ ngõ vào từ i_clk để cung cấp xung cho các khối ở phía sau, nơi mà các khối cần sự đồng bộ để chuyển trạng thái. Ở tại project này, nó là bộ đếm.

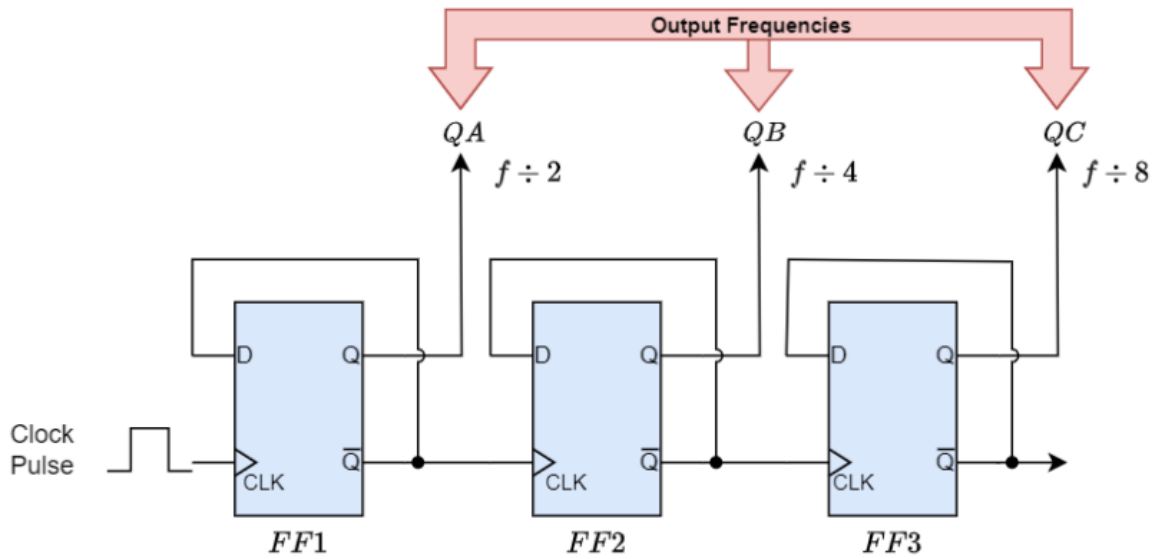
Hướng	Tên tín hiệu	Số lượng bit	Mô tả
input	i_clk	1	Xung clk tổng
output	Ndiv2	1	Xung clk tổng chia 2
output	Ndiv3	1	Xung clk tổng chia 3
output	Ndiv4	1	Xung clk tổng chia 4
output	Ndiv5	1	Xung clk tổng chia 5
output	Ndiv6	1	Xung clk tổng chia 6
output	Ndiv7	1	Xung clk tổng chia 7
output	Ndiv8	1	Xung clk tổng chia 8



Hình 1.1 Sơ đồ khối Prescaler

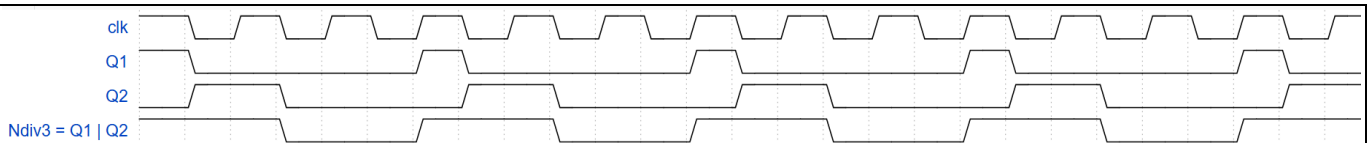
1.2 Những vấn đề ở ngõ ra:

Đối với các ngõ ra như Ndiv2, Ndiv4... hay là 2^n , ta có thể thực hiện được dễ dàng qua nối ngõ ra của Flip Flop này sang Flip Flop khác để tạo độ trễ ở sơ đồ bên dưới.



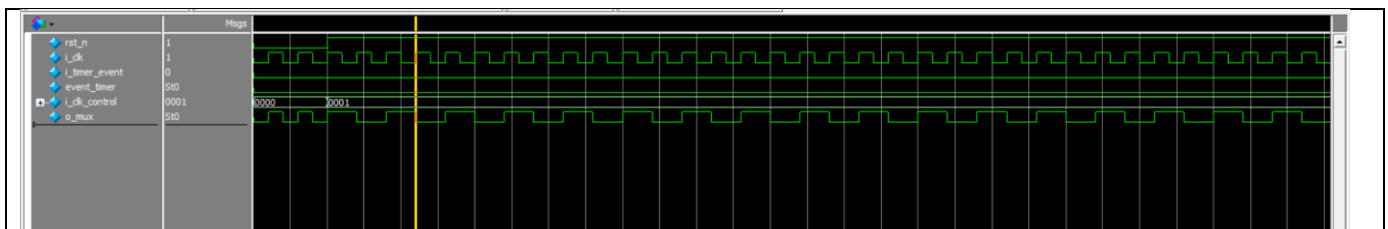
Hình 1.1 Mạch Flip Flop Prescaler

Vậy, các ngõ ra lẻ khác như Ndiv3, Ndiv5, Ndiv7.. ta có thể làm như sau. Cụ thể ở Ndiv3, ta sử dụng 1 tín hiệu trung gian để tạo ra một xung có $T_H = \frac{1}{6}$ và xung thứ 2 là có $T_H = \frac{2}{6}$. Ngõ ra Ndiv3 = Q1 | Q2 sẽ tạo ra xung clock đối xứng với chu kì bằng chu kì xung clock chia 3.

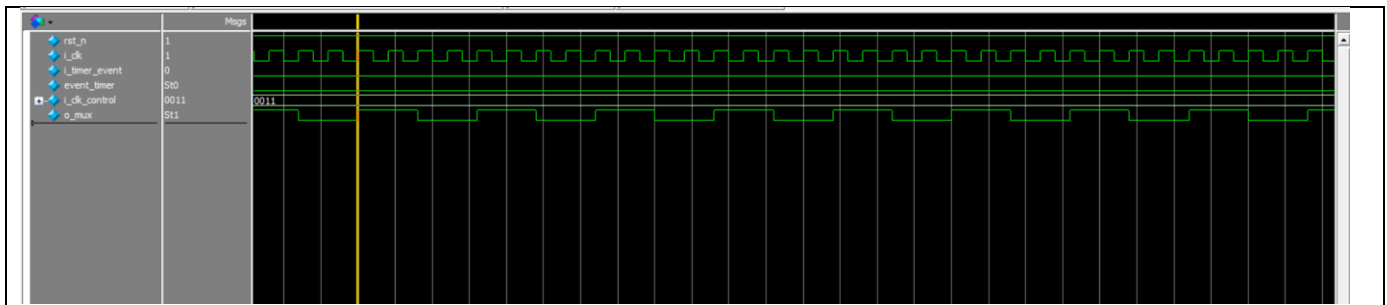


Hình 1.2 Ngõ ra Ndiv3 theo lý thuyết

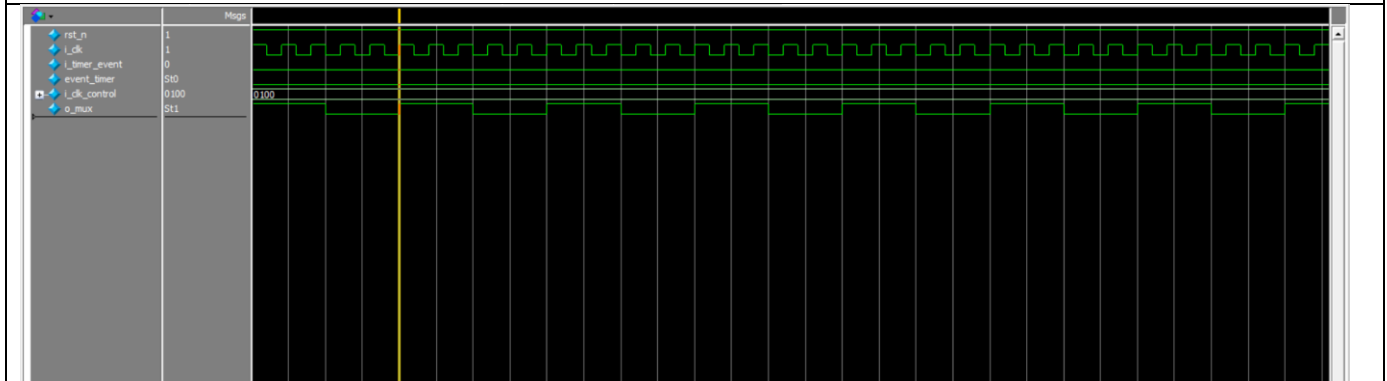
1.3 Kết quả mô phỏng



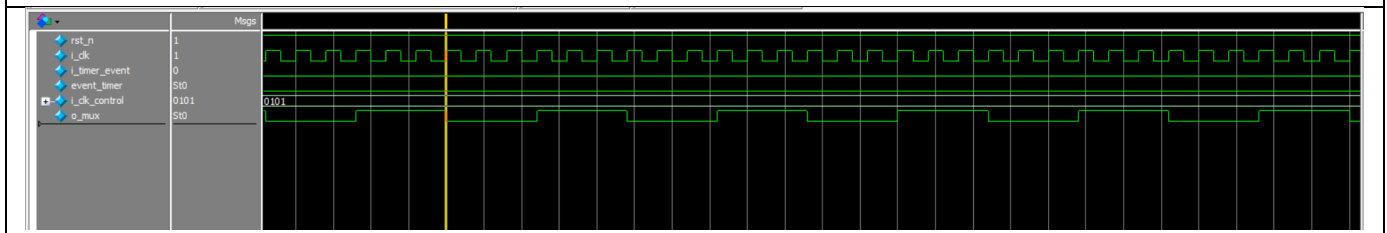
Hình 1.3 Ngõ ra Ndiv3



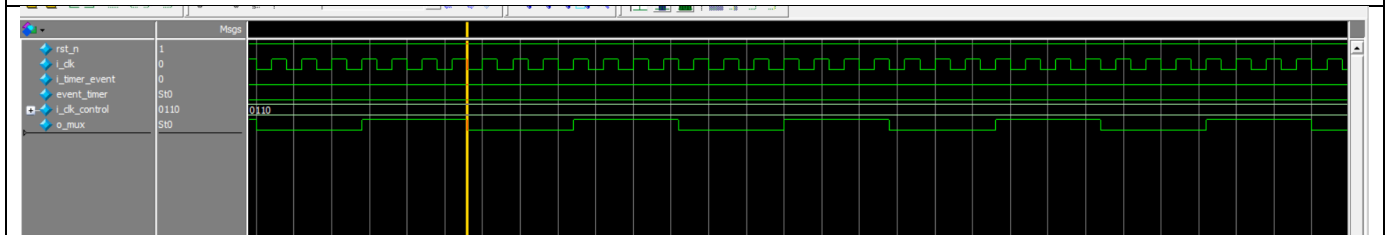
Hình 1.4 Ngõ ra Ndiv4



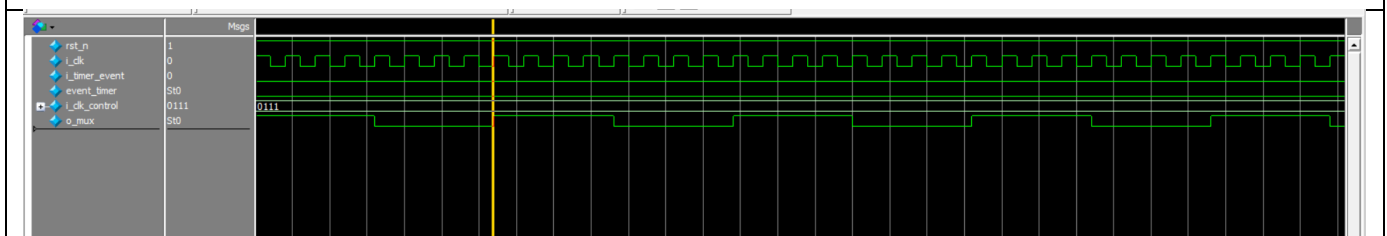
Hình 1.5 Ngõ ra Ndiv5



Hình 1.5 Ngõ ra Ndiv6



Hình 1.6 Ngõ ra Ndiv7



Hình 1.6 Ngõ ra Ndiv8

Nhận xét: Ta có thể thấy, chu kì ngõ ra gấp N lần giá trị ngõ vào với N từ 2 đến 8. Điều này chứng tỏ Prescaler đã hoạt động đúng với yêu cầu.

2. Ctrl_mode

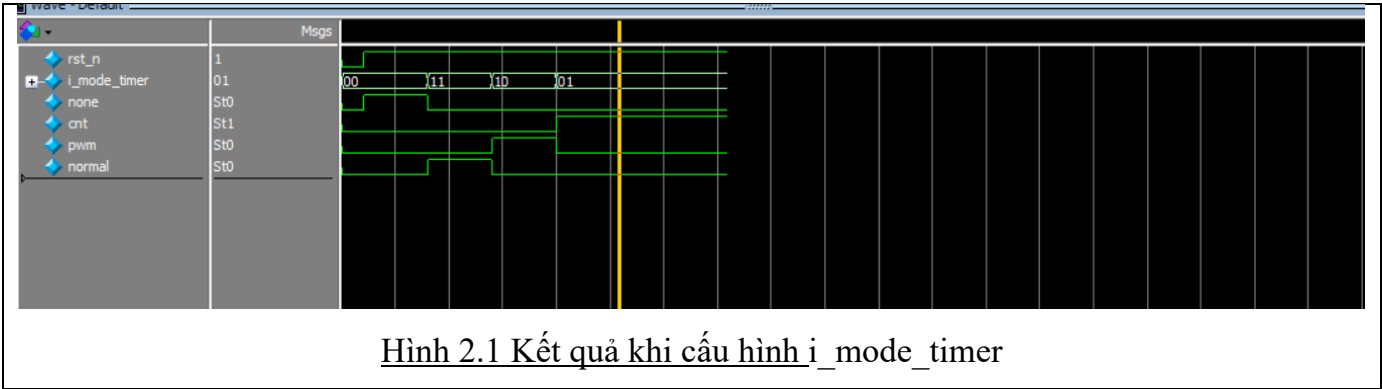
2.1 Các tín hiệu

Khối này cho phép chúng ta điều khiển các chế độ làm việc của timer qua tín hiệu `i_mode_timer`, kết quả với từng ngõ ra tương ứng sẽ lên mức cao. Để đảm bảo về đáp ứng thời gian, khi cấu hình ngõ vào là lập tức xuất hiện ngõ ra cho nên mạch này là 1 mạch tổ hợp bộ, như vậy chức năng của nó giống bộ giải mã 2 sang 4.

Hướng	Tên tín hiệu	Số lượng bit	Mô tả
input	i_mode_timer	2	Điều khiển chức năng của Timer.
input	rst_n	1	Reset các ngõ ra (bằng 0)
output	none	1	none = 1 khi i_mode_timer = 2'b00
output	cnt	1	cnt = 1 khi i_mode_timer = 2'b01
output	pwm	1	pwm = 1 khi i_mode_timer = 2'b10
output	normal	1	normal = 1 khi i_mode_timer = 2'b11

Mục đích của khối này là để người dùng cấu hình timer, hoặc CPU khi cần sử dụng một chức năng nào đó.

2.2 Kết quả thực hiện



Nhận xét: Ngõ ra của khối đúng với yêu cầu.

3. Timer 8 bit

3.1 Khối bộ đếm timer

Tại khối này, nó được sử dụng gồm 3 khối nhỏ tạo thành 1 khối lớn. Thứ nhất là khối bộ đếm 8 bit, khối compare và khối pwm.

Hướng	Tên tín hiệu	Số lượng bit	Mô tả
input	rst_n	1	rst_n = 0 giá trị trong r_timer_count = 0
output	overflow	1	r_timer_count = 255 sẽ bật trong 1 chu kì clk
output	r_timer_count	8	Hiển lưu giá trị của bộ đếm
output	o_compare_interrupt	1	r_timer_count = bộ so sánh sẽ bằng 1
input	data_in	8	Ghi vào khi ở mode normal
output	o_pwm	1	Ngõ ra pwm

4. Verification Plan

[illegible]

[illegible]

[illegible]

Case	Mode	c_clk	Data_in	even	over	o_pwn	cp__i	rst_n	counter	Detail	Check
Chức năng delay 1 khoảng	Normal	1	220	0	1 khi đạt 255	0	1 khi counter = 220	1	190 -> 220 -> 255->0	Có 2 thời điểm được bật tại 2 thời khác nhau, cờ o_compare_interrupt = 1 khi data = 220 và tắt, overflow mức cao khi giá trị bộ đếm = 255	true
Tạo trễ	Normal	1-7	124	0	1 khi đạt 255	0	1 khi counter = 124	1	0->124->255->0	Giá trị đếm phải mất gấp 1,2,3 khi cấu hình c_clk tương ứng, ngõ ra vẫn hoạt động đúng, và cờ hoạt động đúng	true

