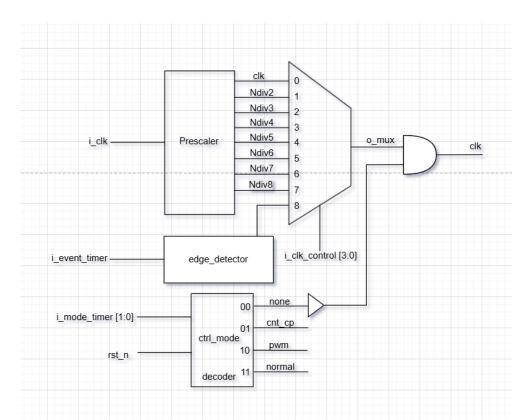
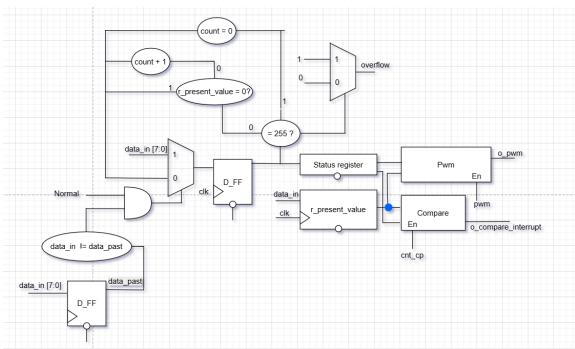
# Tổng quan:



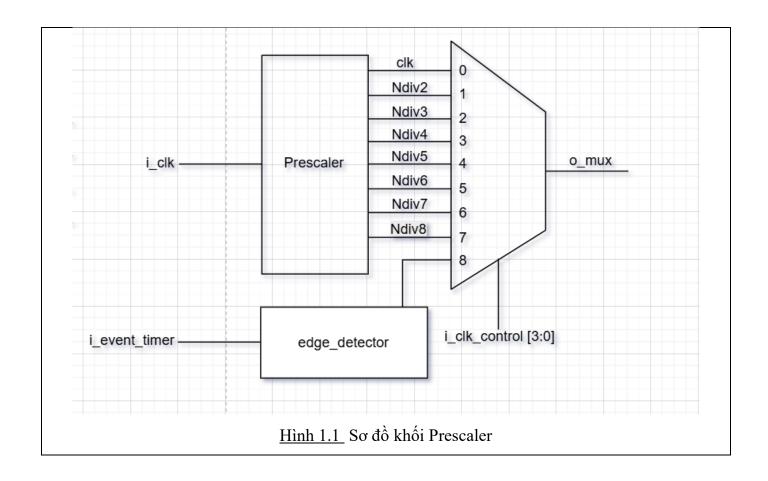


### 1. Prescaler

## 1.1 Các tín hiệu

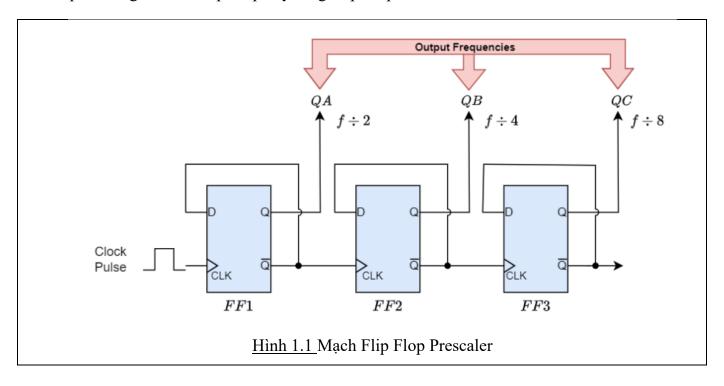
Khối prescaler có nhiệu vụ dẫn tín hiệu clock từ ngõ vào từ i\_clk để cung cấp xung cho các khối ở phía sau, nơi mà các khối cần sự đồng bộ để chuyển trạng thái. Ở tại project này, nó là bộ đếm.

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

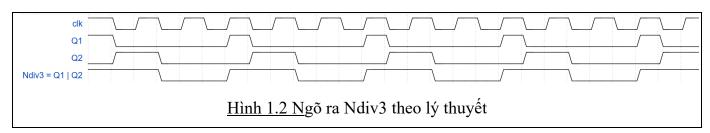


# 1.2 Những vấn đề ở ngõ ra:

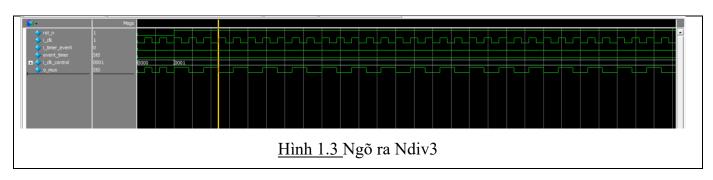
Đối với các ngõ ra như Ndiv2, Ndiv4... hay là  $2^n$ , ta có thể thực hiện được để dàng qua nối ngõ ra của Flip Flop này sang Flip Flop khác để tạo độ trễ ở sơ đồ bên dưới.

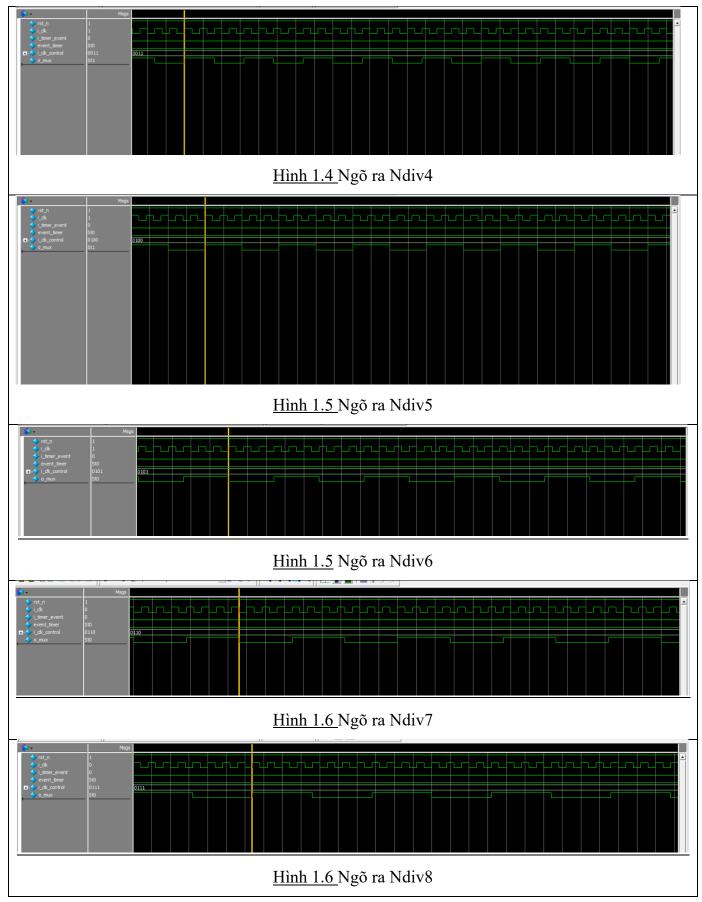


Vậy, các ngõ ra lẻ khác như Ndiv3, Ndiv5, Ndiv7.. ta có thể làm như sau. Cụ thể ở Ndiv3, ta sử dụng 1 tín hiệu trung gian để tạo ra một xung có  $T_H = \frac{1}{6}$  và xung thứ 2 là có  $T_H = \frac{2}{6}$ . Ngõ ra Ndiv3 = Q1 | Q2 sẽ tạo ra xung clock đối xứng với chu kì bằng chu kì xung clock chia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



# 1.3 Kết quả mô phỏng





**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

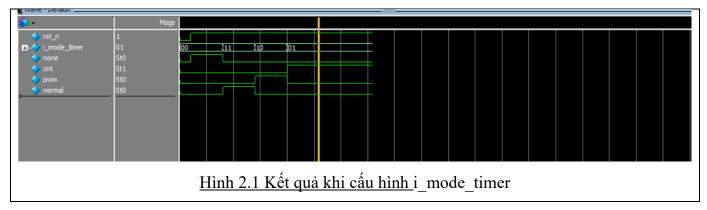
## 2.1 Các tín hiệu

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 báo 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

Case	Mode	c_clk	Data_in	even	over	o_pwn	cpi	rst_n	counter	Detail	Check
Reset bộ	none	1	0	0	0	0	0	0	0	Reset timer, counter ko	true
đếm										đếm	
Khi vào	none	1	255	0	0	0	0	1	0	Mode none timer không	true
none thì										đếm	
timer											
không đếm											
So sánh	cnt_cp	1	120	0	1 khi	0	1	1	0->120-	Bằng 1 khi đạt giá trị so	true
khi có giá					bộ				255->0	sánh	
trị vào					đếm						
					255						
So sánh	cnt_cp	1	255	0	1 khi	0	1	1	255 -> 0	Cờ ngắt tích cực khi bộ	true
khi có giá					bộ					đếm 255	
trị vào tối					đếm						
đa					255						
So sánh	cnt_cp	1	0	0	0	0	0	1	Dừng tại	Khi data_in = 0 timer	true
khi có giá									23	không chạy	
trị vào nhỏ											
nhất											

Case	Mode	c_clk	Data_in	even	over	o_pwn	cpi	rst_n	counter	Detail	Check
So sánh	cnt_cp	1	1	0	1 khi	0	1 khi	1	0->1-	Cờ ngắt bằng 1 khi bộ	true
khi có giá					count		bộ		>255 -> 0	đếm bằng 1, bằng 0 khi	
trị vào là 1					er =		đếm			bộ đếm không phải 1	
					255		là 1				
So sánh	cnt_cp	1	8'b1010	0	1 khi	0	1 khi	1	0->170	Cò ngắt mức cao khi tại	true
khi có giá			1010		count		bộ		>255-> 0	bộ đếm 170	
trị vào số					er =		đếm				
đặt biệt					255		là				
							170				
Chuyển	pwm	1	240	0	1 khi	Tích	0	1	0->104-	Giá trị 104 về trước còn ở	true
mode từ so					count	cực			>255-> 0	cnt_cp nên o_pwm không	
sánh ->					er =	từ				hoạt động, từ 104 về sau	
pwm					255	104-				o_pwm mức cao	
						240					
Độ rộng	pwm	1	255	0	1 khi	Hoàn	0	1	0->255-	Ngõ ra o_pwn hoàn toàn	true
xung 100%					count	toàn			>0	mức cao	
					er =	mức					
					255	cao					
				•••		•••					

Case	Mode	c_clk	Data_in	even	over	o_pwn	cpi	rst_n	counter	Detail	Check
Độ rộng	pwm	1	128	0	1 khi	Độ	0	1	0->128-	Ngõ tích cực 1 nữa trong	true
xung 50%					count	rộng			>0	suốt thời gian đếm	
					er =	đạt 1					
					255	nữa					
Đếm sự	pwm	8	128	0-	0	Độ	0	1	128->134	O_pwm mức thấp khi	true
kiện +				>1		rộng				trong suốt thời gian đếm	
pwm				6		lớn					
không				lần		hơn					
trong vùng						50%					
tích cực											
của											
o_pwm											
Đếm sự	pwm	8	230	0-	1 khi	Hoạt	0	1	134->190	O_pwm mức cao trong	true
kiện +				>1	count	động				134->190	
pwm trong				56	er=	khi					
vùng				lần	255	data_in					
tích cực						>					
của						counte					
o_pwm						r					
				••••	••••				••••		

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