

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HỒ CHÍ MINH



BÁO CÁO ĐỒ ÁN MÔN HỌC
MÔN : ĐỒ ÁN 1

ĐỀ TÀI: THIẾT KẾ 1 TIMER 8 BIT TRÊN FPGA KIT DE2

GVHD: TS. Nguyễn Lý Thiên Trường

Học kì: 242

Thành phố Hồ Chí Minh – 2025

THÀNH VIÊN

STT	Họ và tên	MSSV	Nhiệm vụ	Điểm số
1.	Thái Trung Kiên	2211735	100%	

GV Hướng dẫn ký tên

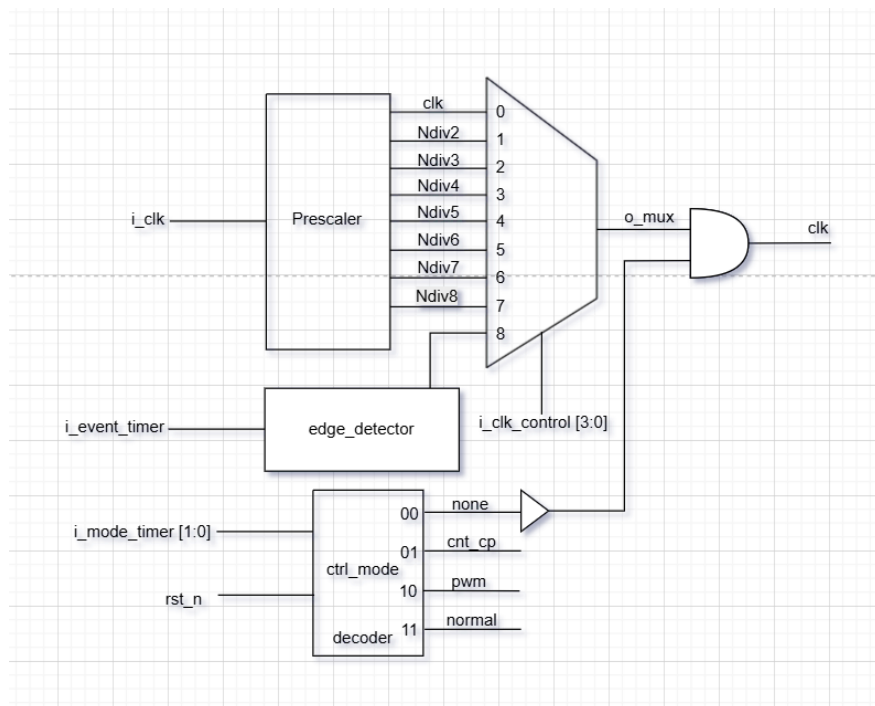
Tổng quan:

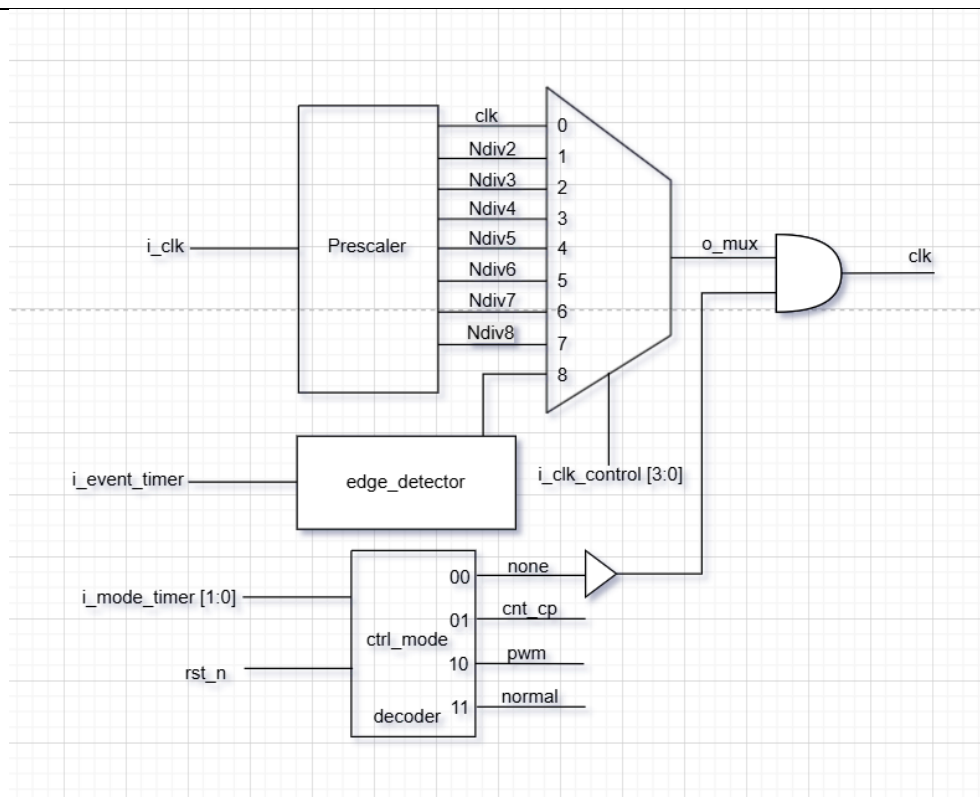
Thiết kế 1 Timer 8 bit trên FPGA KIT DE2.

Chức năng phải đảm bảo:

- Phải hỗ trợ chuyển đổi các chứng năng.
- Có khả năng tạo trễ delay.
- Có khả năng tạo độ rộng xung.
- Có khả năng đếm sự kiện từ bên ngoài.
- Có ngắt để báo hiệu thông báo.
- Có khả năng điều chỉnh tần số xung clock.

Sơ đồ khối tổng quát:

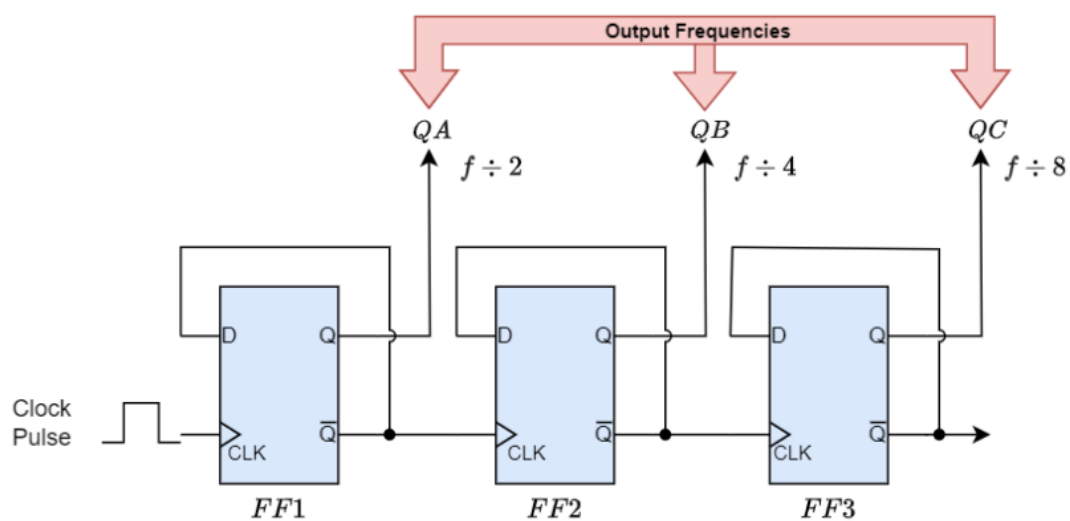




Hình 1.1 Sơ đồ khối phần 1

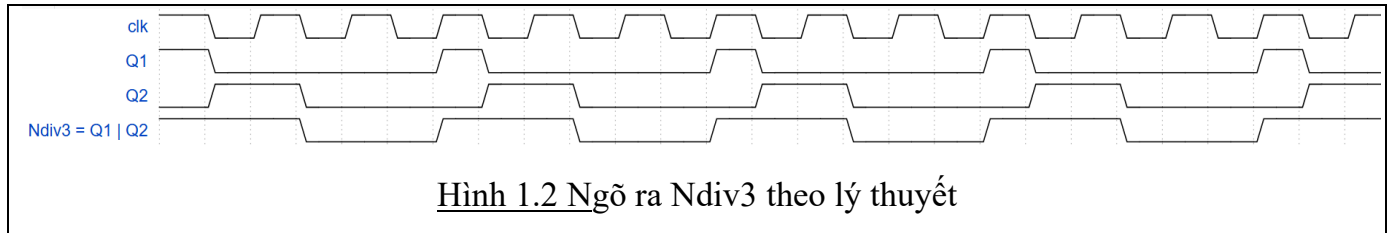
1.2 Những vấn đề ở ngõ ra:

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 clk Flip Flop khác để tạo độ trễ ở sơ đồ bên dưới.



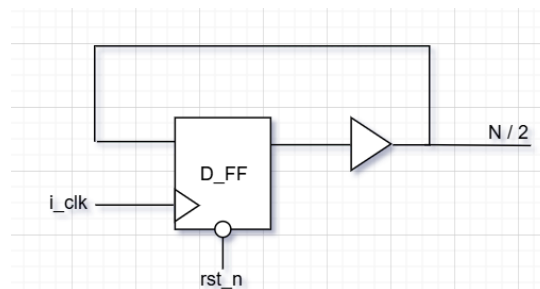
Hình 1.1 Mạch Flip Flop Prescaler

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 or với Q2 sẽ tạo ra xung clock đối xứng với chu kì bằng chu kì xung clock chia 3.



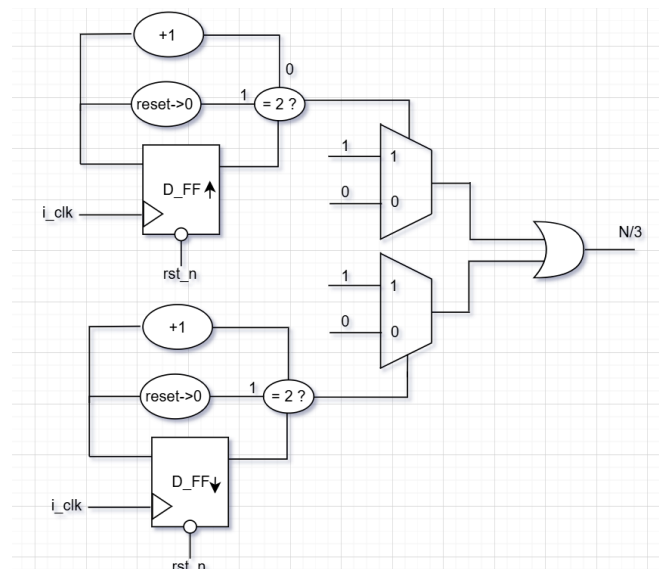
1.3 Sơ đồ khối của bộ chia tần

a. Tín hiệu i_clk chia 2

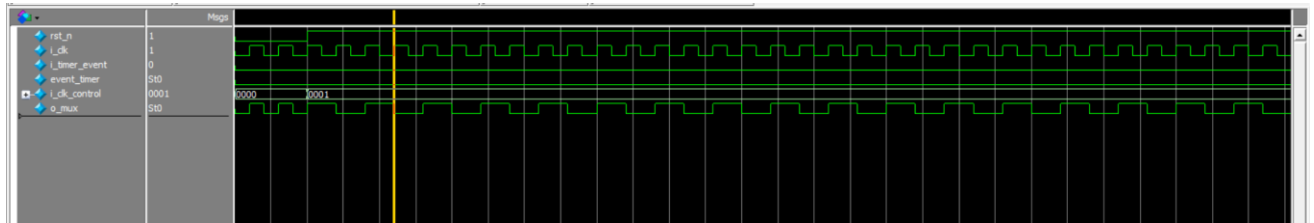


Tín hiệu lúc này đảo chiều liên tục với tần số chia 2.

b. Tín hiệu i_clk chia 3

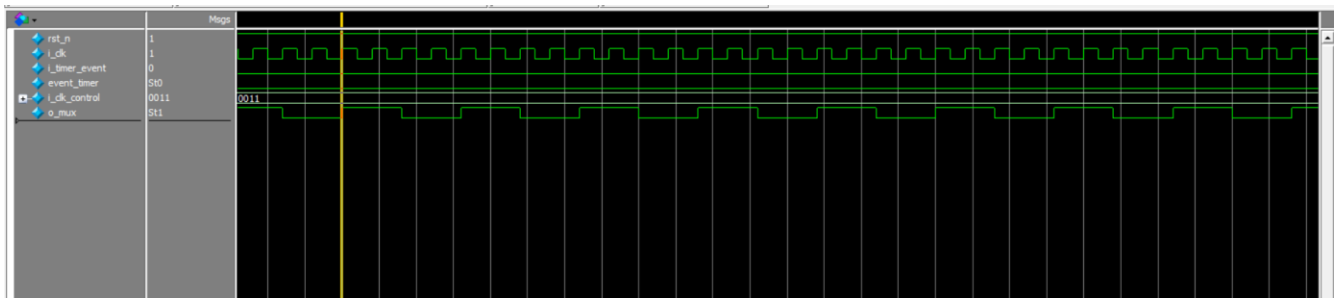
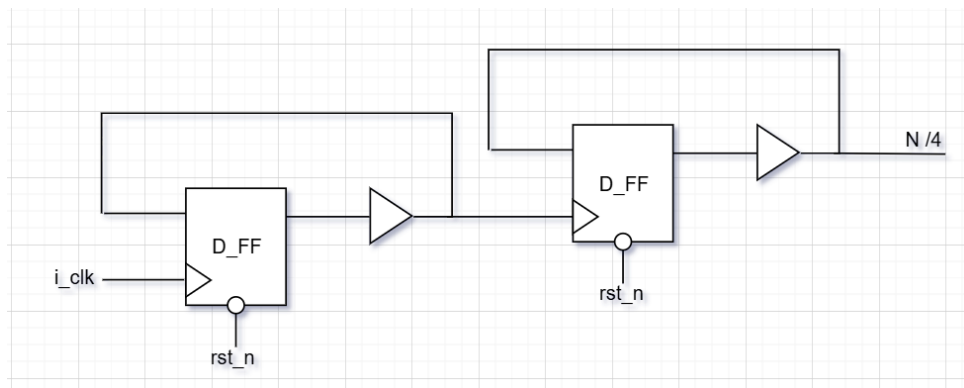


Tần số chia 3 gồm có 2 bộ đếm chạy dịch pha với nhau qua bộ đếm trên là chỉ đếm khi có cạnh xuống. Vậy ngõ ra $N/3 = Q1$ or với $Q2$ sẽ tạo ra xung clock đối xứng với chu kỳ bằng chu kỳ xung clock chia 3 đúng với lý thuyết qua kết quả mô phỏng ở dưới.



Hình 1.3 Ngõ ra $N/3$

c. Tín hiệu i_clk chia 4

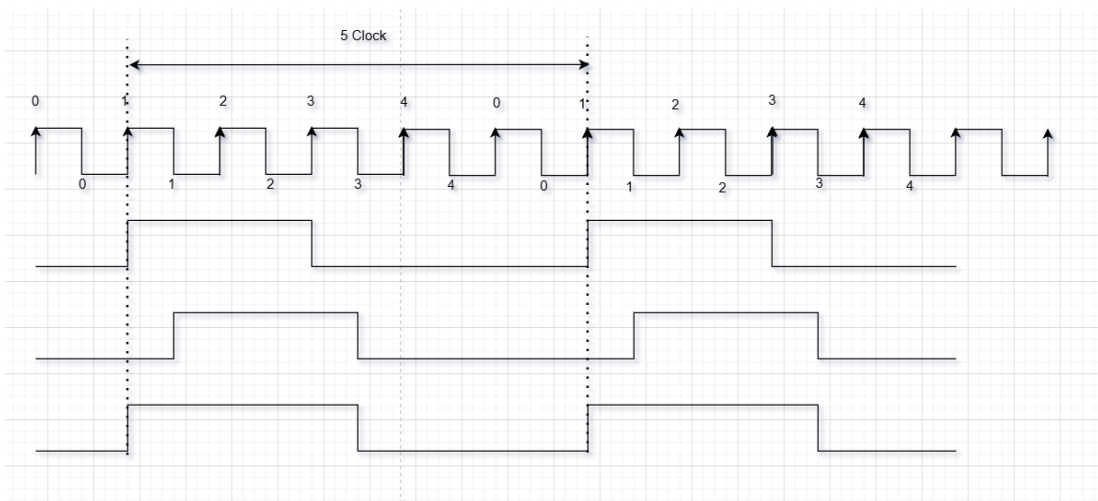
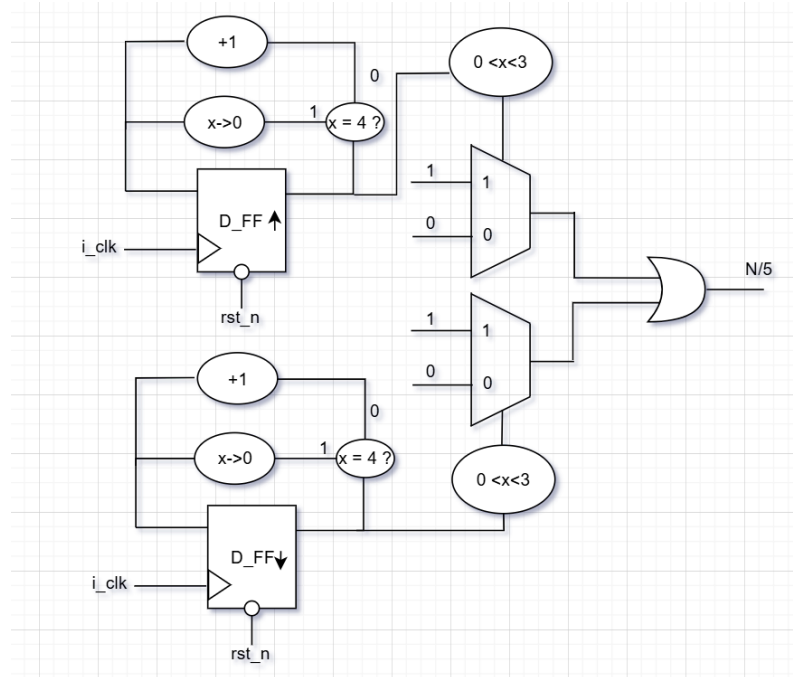


Hình 1.4 Ngõ ra $N/4$

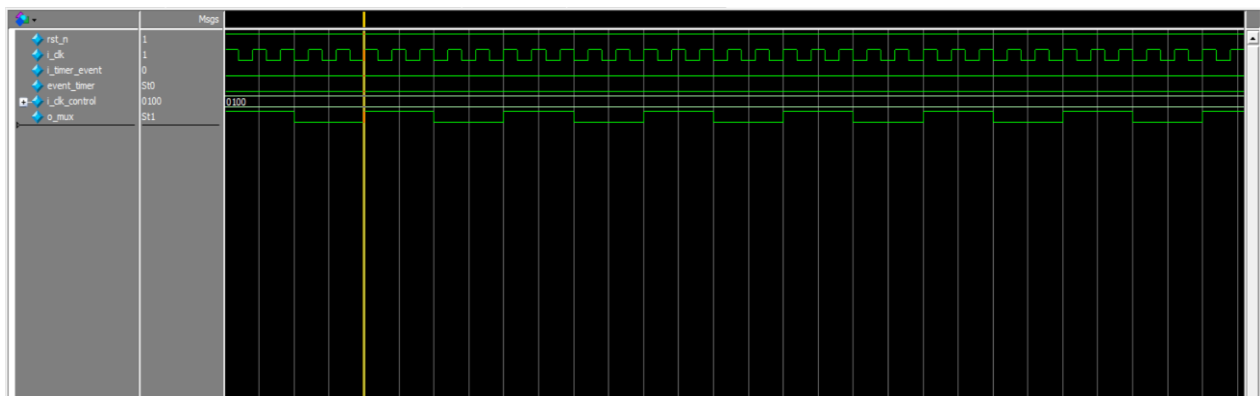
Ngõ ra xung clock $N/4$ được thể hiện qua tính chất nối nối tiếp các flop flop với nhau được thể hiện qua kết quả mô phỏng ở trên.

d. Tín hiệu i_clk chia 5

Giống với tần số chia 3, bộ chia 5 gồm có 2 bộ đếm chạy dịch pha với nhau qua bộ đếm trên là chỉ đếm khi có cạnh xuống. Nhưng điểm khác ở đây là giá trị của 2 bộ đếm sẽ tăng thêm cho 2. Và điều kiện bằng 1 của 2 bộ mux cũng khác nhau để minh họa dễ hơn ta có waveform ở phía dưới.



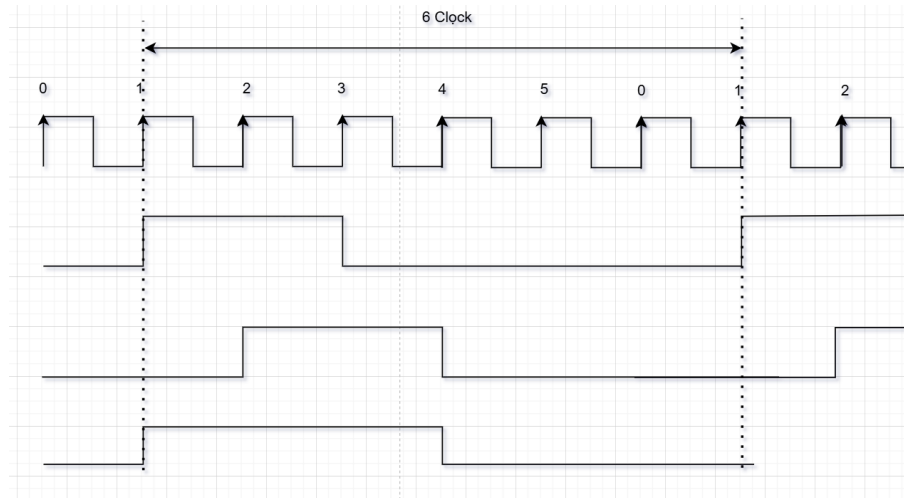
Bộ đếm thứ nhất chạy ở cạnh lên xung i_clk và bộ đếm thứ 2 chạy cạnh dưới của i_clock . Lúc này bộ mux của bộ đếm thứ nhất sẽ tích cực tại giá trị bộ đếm là $0 < x < 3$ và bộ đếm ở dưới chỉ lệnh pha nửa chu kỳ i_clk . Ngõ ra $N/5$ chỉ cần OR 2 tín hiệu vào là ta thu được ngõ ra $N/5$. Kết quả mô phỏng ở dưới



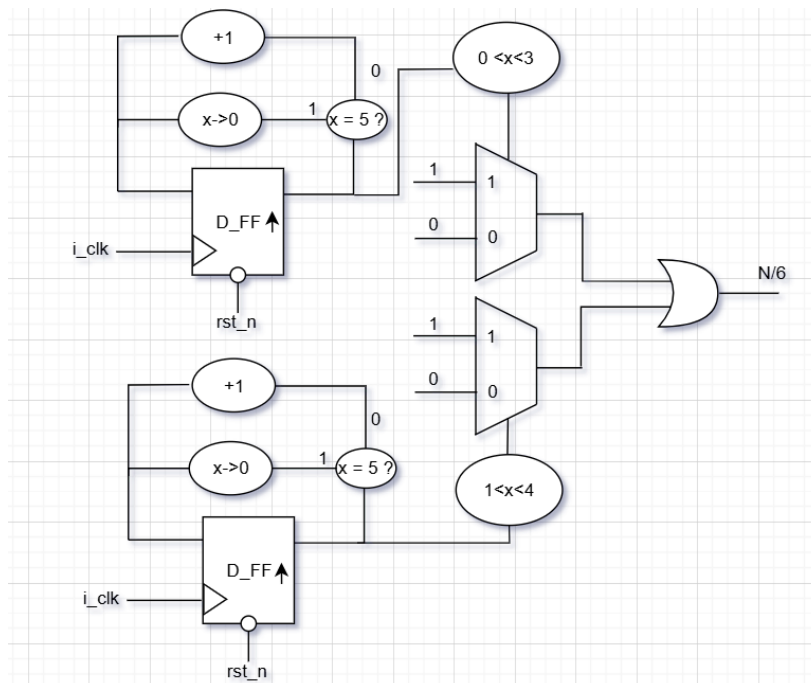
Hình 1.5 Ngõ ra Ndiv5

e. Tín hiệu i_clk chia 6, 7 và 8

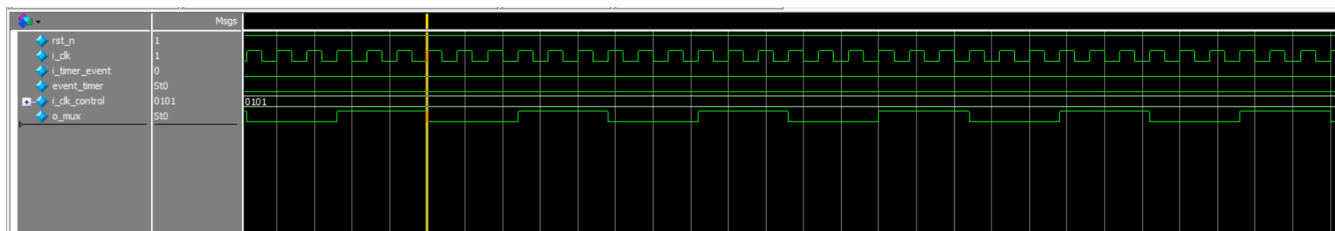
Đối với ngõ ra với tần số chia 6 ta cũng cần 2 bộ đếm như trên nhưng 2 bộ này sẽ là đếm theo cạnh lên. Điều kiện lần lượt là $0 < x < 3$ và $1 < x < 4$ có waveform theo lý thuyết là:



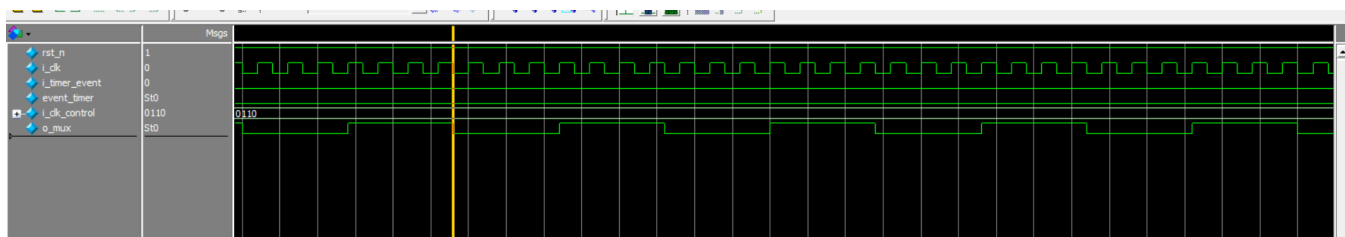
Điều này có nghĩa là ta dịch pha 1 clock cho bộ đếm ở dưới.



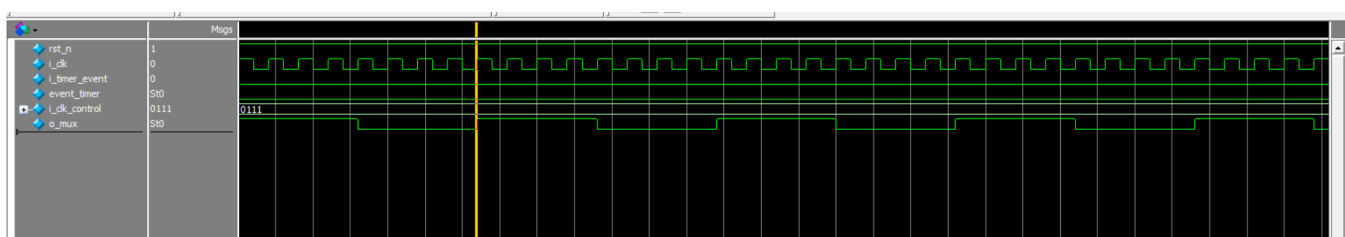
Tương tự như cách làm của các bộ đếm trên ta thu được dạng sóng mô phỏng ở dưới .



Hình 1.6 Ngõ ra Ndiv6



Hình 1.6 Ngõ ra Ndiv7



Hình 1.7 Ngõ ra Ndiv8

Nhận xét: Ta có thể thấy, chu kì ngõ ra có tần số gấp N lần giá trị ngõ vào i_clk 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 rơi đúng vào mode mà người dùng muốn sử dụng tùy vào những mục đích cụ thể, hoặc nếu phát triển thiết kế này cao hơn thì người dùng thông qua trung gian là CPU mà cấu hình một chức năng nào đó của timer.

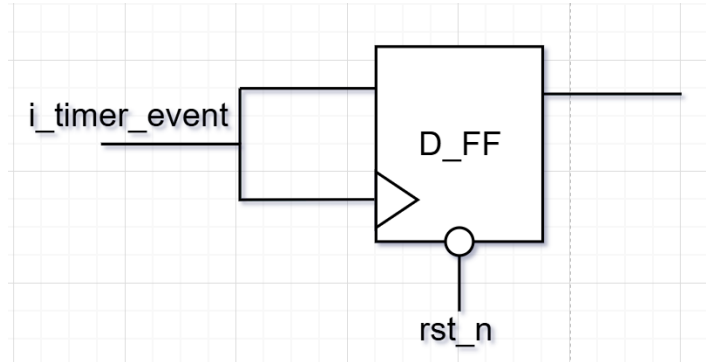
2.2 Kết quả thực hiện



Hình 2.1 Kết quả khi cấu hình i_mode_timer

Nhận xét: Ngõ ra của khối đúng với yêu cầu khi cấu hình 2 bit của tín hiệu i_mode_timer mà điện áp mức 1 xuất hiện tương ứng.

2.3 Khối dò cạnh lên edge_detector



Khối dò cạnh lên thực chất là khối flip flop với clock vào cũng chính là nó. Khi mà có cạnh lên từ i_timer_event thì lập tức ngõ ra này sẽ bằng tín hiệu timer event và giữ tín hiệu mức cao.

3. Timer 8 bit

3.1 Thanh ghi đặt trước và thanh ghi trạng thái

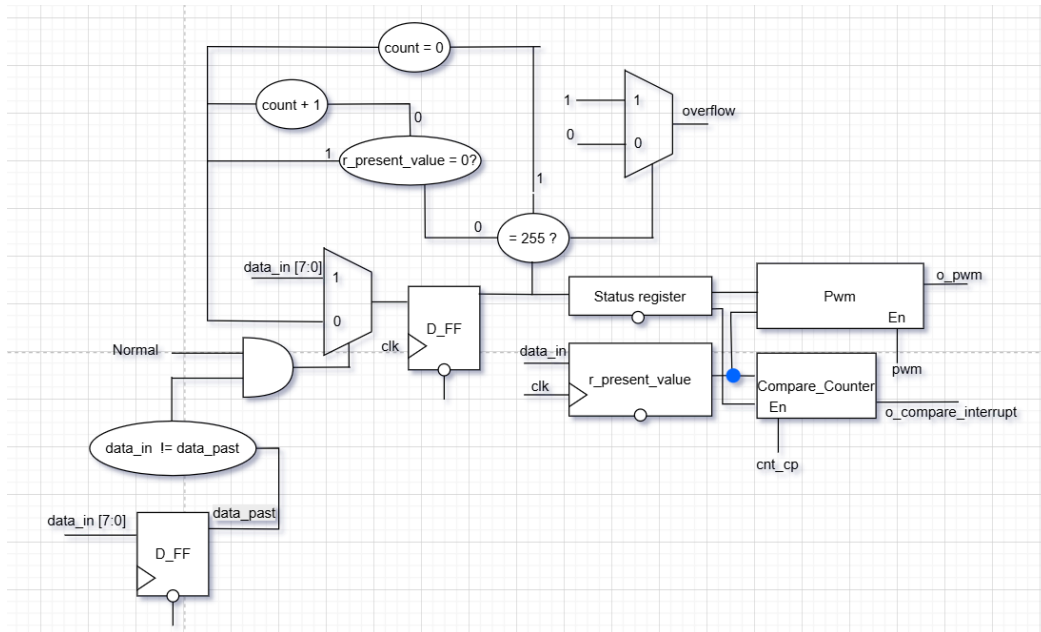
a. Thanh ghi trạng thái Status_register

Thanh ghi này có tác dụng là nơi để ghi nhớ trạng thái bộ đếm, lưu trữ các giá trị của bộ đếm, nó cũng như là cầu nối để truyền dữ liệu từ bộ đếm sang khối chức năng PWM và khối Compare_Counter.

b. Thanh ghi đặt trước r_present_value

Đây là nơi để lưu các giá trị đặt trước với ngõ vào là data_in. Khi nội dung trong thanh ghi trống hay là 8'b00000000 lúc này bộ đếm sẽ không hoạt động, muốn bộ đếm hoạt động phải nạp dữ liệu khác 0 vào.

3.2 Khối bộ đếm timer



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	Hiện lưu giá trị của bộ đếm
output	o_compare_interrupt	1	r_timer_count = bộ so sánh sẽ bằng 1
input	i_data_in	8	Ghi vào khi ở mode normal
output	o_pwm	1	Ngõ ra pwm

Vì khối này, nó được sử dụng gồm 3 khối nhỏ tạo thành 1 khối lớn. Thứ nhất là khối bộ đếm 8 bit cũng là khối lớn nhất có chức năng đếm, ghi giá trị đặt trước và lưu giá trị trong thanh ghi r_timer_count và nếu data_in không có dữ liệu thì timer cũng không đếm. Muốn

đếm tại giá trị ghi vào thì cần 2 điều kiện, thứ nhất là timer phải ở chế độ normal lúc này chế độ pwm và counter compare sẽ không hoạt động. Thứ 2 là nội dung của data_in phải khác với giá trị trước đó. Nguyên nhân của thiết kế này bởi vì, khi mà data_in khi vào timer phải cần điều kiện không cập nhật giá trị của bộ đếm nhằm chỉ cho phép duy nhất 1 giá trị vào timer. Nếu không có điều kiện data_in khác data_in trước đó thì lúc nào timer chỉ luôn luôn ở giá trị 20 mà không thể đếm lên 21.

Cách hoạt động của bộ đếm:

Khi có tín hiệu rst_n tích cực thấp làm cho giá trị bộ đếm khởi tạo là 0 và khi có tín hiệu xung clock lập tức tín hiệu sẽ trải qua các điều kiện nếu là 255 và tín hiệu overflow sẽ báo tràn, tại thời điểm này giá trị bộ đếm sẽ lập tức về 0 còn nếu tín hiệu chưa đạt 255 thì sẽ +1 vào giá trị bộ đếm.

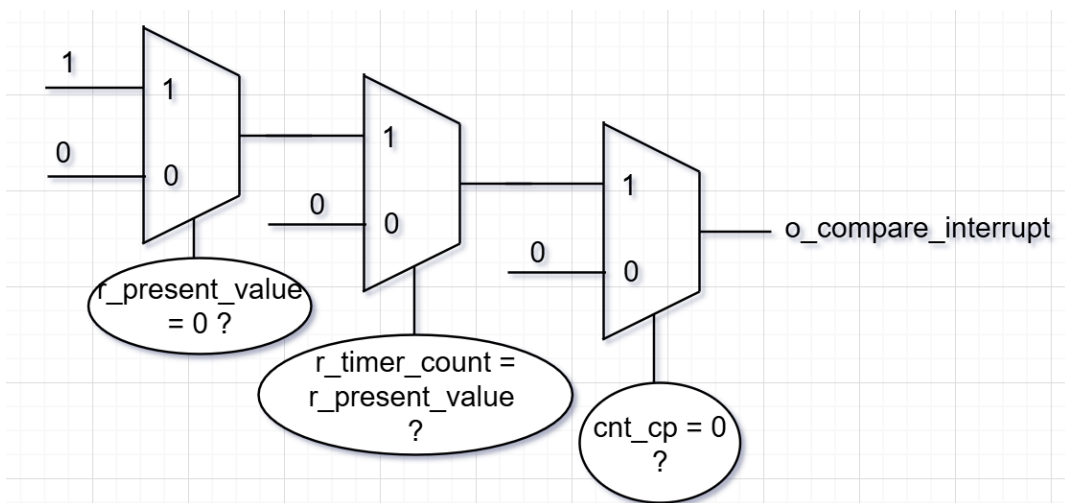
3.2 Khối Counter Compare và khối PWM

a. Khối Counter Compare:

Để tín hiệu ngắt được kích hoạt thì phải thỏa mãn các điều kiện sau:

- Timer phải ở chế độ counter compare.
- Giá trị trong 2 thanh ghi phải bằng nhau.
- Thanh ghi đặt trước có phải khác 0.

Ta có sơ đồ khối như sau:



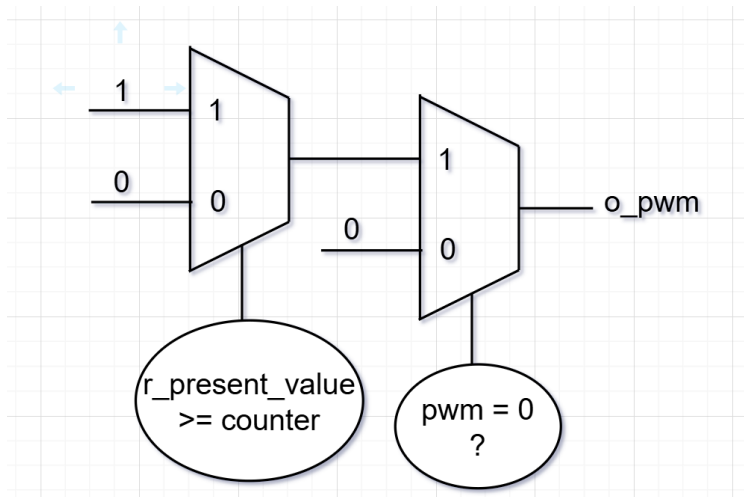
Dựa vào mục tiêu thiết kế mà ta ưu tiên mức điều kiện, cao nhất là cnt_cp bởi khi cnt_cp = 1 điều này sẽ giúp timer hiểu là người dùng muốn sử dụng mode counter compare. 2 Điều kiện còn lại có thể bằng nhau nhưng trong thiết kế này sẽ có sự ưu

tiên như sơ đồ khối trên. Khi mà thỏa mãn tất cả các điều kiện thì tín hiệu 1 sẽ truyền qua 3 bộ mux đến ngõ ra.

b. Khối PWM:

Điều kiện để điều chế độ rộng xung kích hoạt cũng gần giống với cách mà điều kiện ngắt được kích hoạt nhưng có chút sự khác biệt:

- Timer phải ở chế độ PWM qua tín hiệu pwm.
- Giá trị trong thanh ghi đặt trước lúc nào cũng lớn hơn giá trị bộ đếm.



Chỉ khi tất cả điều kiện được kích hoạt thì ngõ ra o_pwm sẽ được giữ ở mức cao cho đến khi giá trị bộ đếm lớn hơn giá trị trong thanh ghi đặt trước.

4. Verification Plan

Case	Mode	c_clk	Data_in	even	over	o_pwn	cp__i	rst_n	counter	Detail	Check
Reset bộ đếm	none	0	0	0	0	Không hoạt động	Không hoạt động	0	0	Reset timer, counter không đếm	true
Khi vào none thì timer không đếm	none	0	255	0	0	Không hoạt động	Không hoạt động	1	0	Mode none timer không đếm	true
So sánh khi có giá trị vào	cnt_cp	0	120	0	1 khi bộ đếm 255	Không hoạt động	1 khi data_in = counter	1	0->120	Bằng 1 khi đạt giá trị so sánh	true
So sánh khi có giá trị vào tối đa	cnt_cp	0	255	0	1 khi bộ đếm 255	Không hoạt động	1	1	255 -> 0	Cờ ngắt tích cực khi bộ đếm 255	true
So sánh khi có giá trị vào nhỏ nhất	cnt_cp	0	0	0	0	Không hoạt động	0	1	Dừng tại 2	Khi data_in = 0 timer không chạy và dừng tại giá trị 2	true

[illegible]

[illegible]

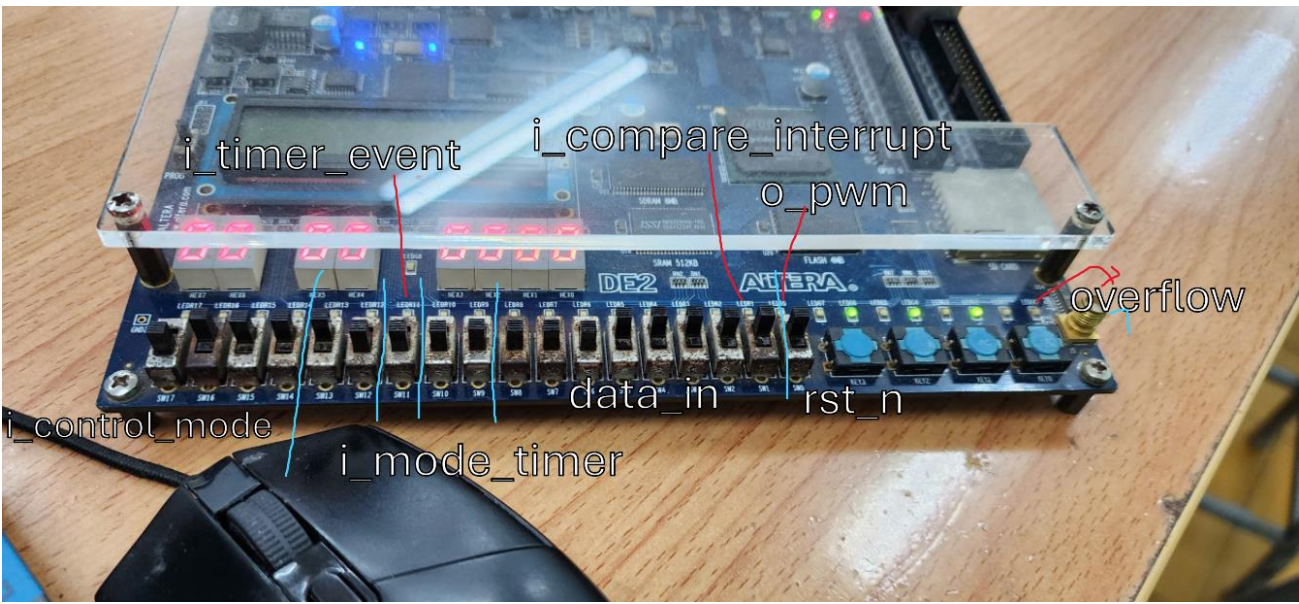
Case	Mode	c_clk	Data_in	even	over	o_pwn	cp__i	rst_n	counter	Detail	Check
Chức năng delay 1 khoảng	Normal	0	220	0	1 khi đạt 255	Không hoạt động	Không hoạt động	1	220 -> 255	Bộ đếm bắt đầu chạy từ 220	true
Chức năng delay 1 khoảng + đếm sự kiện	Normal	8	12	0->1 13 lần	1 khi đạt 255	Không hoạt động	Không hoạt động	1		Bộ đếm bắt đầu chạy từ 12 và đếm sự kiện từ 12 sang 24	true
Ghi 2 lần giá trị data_in	Normal	0	20	0	1 khi đạt 255	Không hoạt động	Không hoạt động	1	20 -> 255	Bộ đếm chỉ nhận dữ liệu mới khi khác dữ liệu ban đầu và bắt đầu chạy từ 20	true

Để Thuận tiện nhất để kiểm tra bộ chia tần có thể nạp lên kit FGPA DE 2 và kết quả ở bên dưới

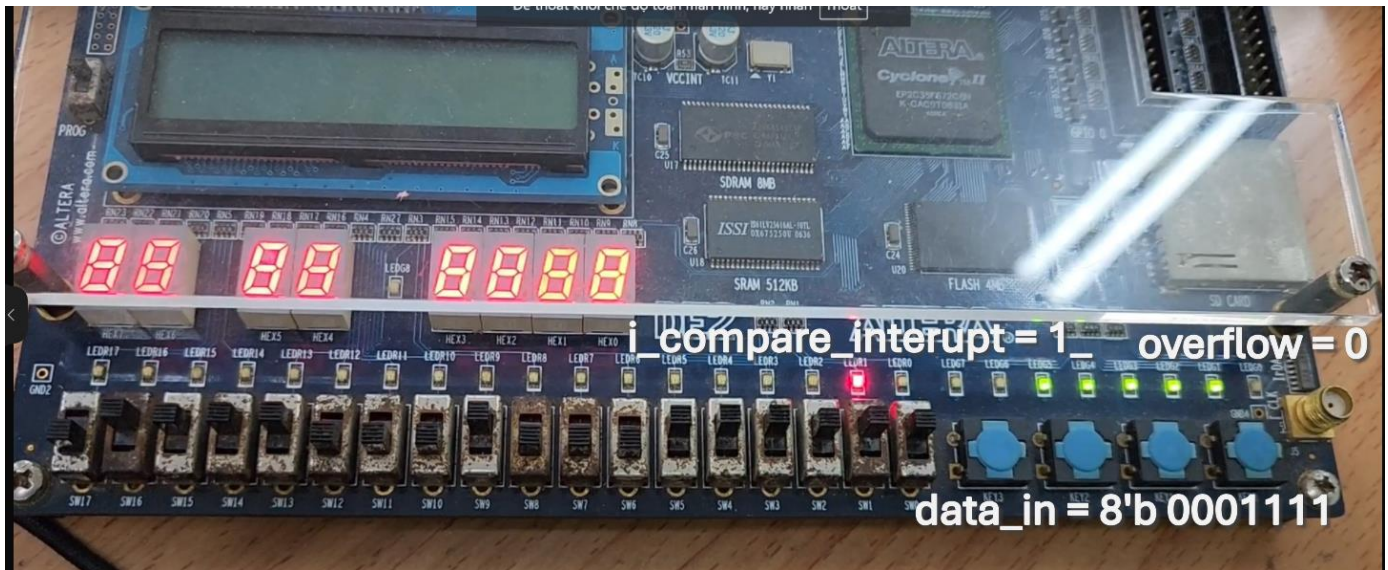
5. Nạp lên kit FPGA

Khi nạp trên kit trên FPGA có 1 vấn đề là thiết kế sẽ chạy với tần số rất là nhanh 27M, cho nên khi nạp trên FPGA em đã tinh chỉnh lại i_control_clk để cho mắt người có thể thấy thông qua các led.

Signal	I/o
rst_n	Sw 0
i_clk	Sử dụng clock 27Mhz
i_timer_event	Sw 11
[3:0] i_clk_control	SW 14 - 17
i_data_in	SW 1 - 8
i_mode_timer	Sw 9 - 10
overflow	Led green 0
o_compare_interrupt	Led red 1
o_pwm	Led red 0
r_timer_count	Led green 1 - 8



5.1 Test Compare_counter mode với data_in = 8'b0001111:



[Video trên kit](#)

Kết quả: Khi data_in = 00011111 thì ngắt xuất hiện.

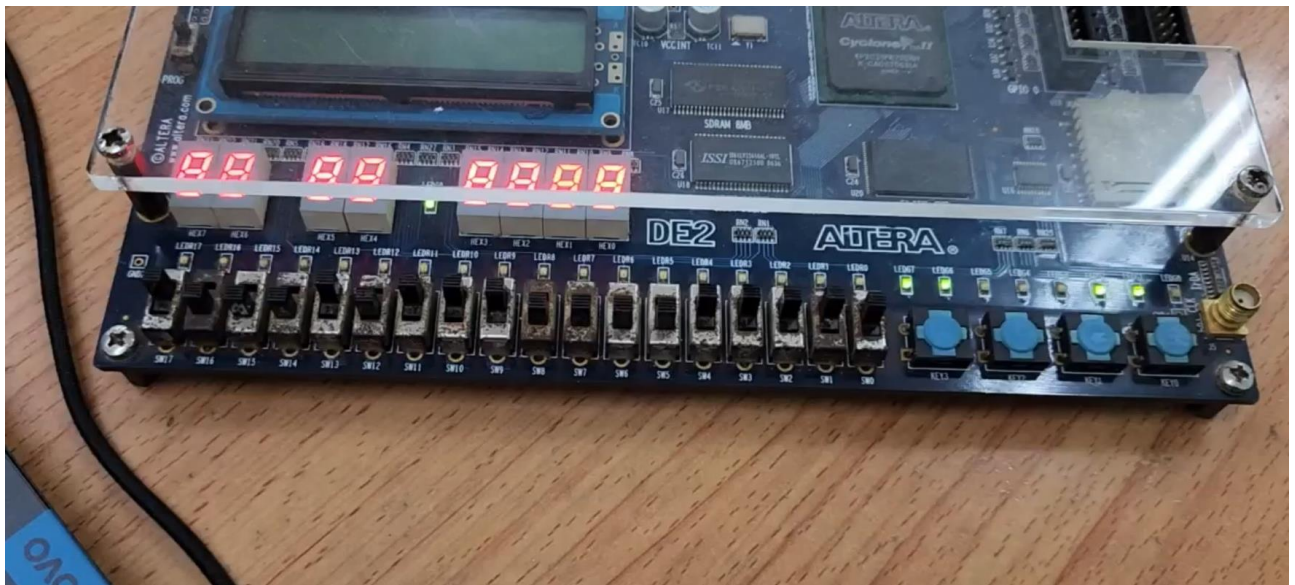
5.2 Test PWM mode + Event với data_in = 8'b0001111:



[Video trên kit](#)

Kết quả: khi data_in lớn hơn giá trị của counter thì o_pwm ở mức cao

5.3 Test chuyển mode liên tục



[Video trên kit](#)

Kết quả: khi chuyển mode từ none sang 1 mode bất kì liên tục sẽ bị tăng giá trị bộ đếm, đây là lỗi sai so với thiết kế, bởi vì tín hiệu none chung với tín hiệu xung clock.

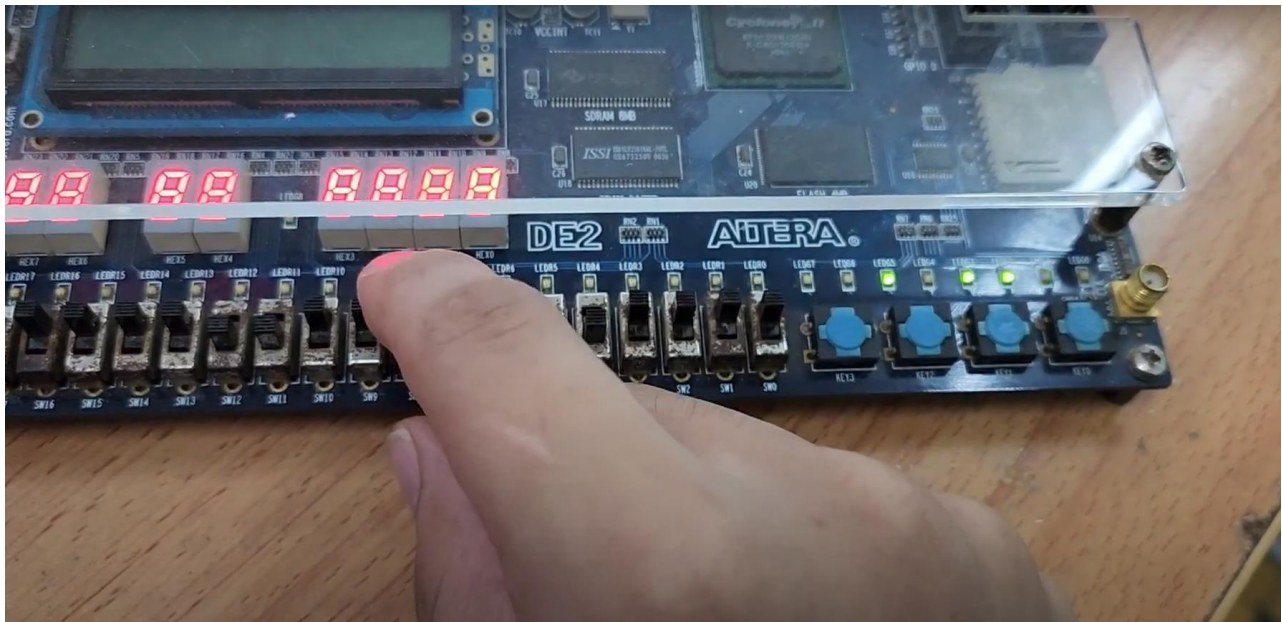
5.4 Test PWM độ rộng xung 100% và độ rộng xung khi data_in = 8'b00001111



[Video trên kit](#)

Kết quả: Khi giá trị tối đa ngõ ra pwm luôn luôn hoạt động, khi nạp 8'b00001111 vào lập tức sẽ sáng trong 1 khoảng thời gian.

5.5 Test Mode normal khi ghi giá trị vào sẽ cập nhật giá trị data_in vào bộ đếm



[Video trên kit](#)

Kết quả: Khi nạp giá trị vào từ 01000111 -> 01000111 thì giá trị bộ đếm sẽ bắt đầu đếm từ 01000111.

6. Hạn chế của thiết kế

Thiết kế được làm và phát triển trên FPGA nên có 1 số ưu và nhược điểm sau:

- Thiết kế nhỏ chưa có ứng dụng thực tế.
- Chưa có giao tiếp để điều khiển Timer.
- Phù hợp để nắm rõ cấu trúc của 1 Timer bất kì, có thể phát triển trong tương lai.
- Còn gặp 1 lỗi khi chuyển mode, chỉ phát hiện ra khi nạp trên FPGA, bởi vì trên mô phỏng chưa nghĩ đến trường hợp đó.

7. Hướng phát triển

Thiết kế này có thể được phát triển để phù hợp trên thực tế chẳng hạn sẽ có giao tiếp AXI, AMBA, ... để giao tiếp với CPU. Có thể tăng dung lượng bộ đếm từ 8 bit đến 32 hoặc 64 bit để phù hợp với 1 Timer công nghiệp.