Computer Architecture and System Laboratory

Laboratory 10
PWM Breathing Light

Department of Electrical Engineering
National Cheng Kung University

國立成功大學電機系

Purpose

- 熟悉 PYNQ-Z2 Board
- 了解 PWM (Pulse Width Modulation)

Outline

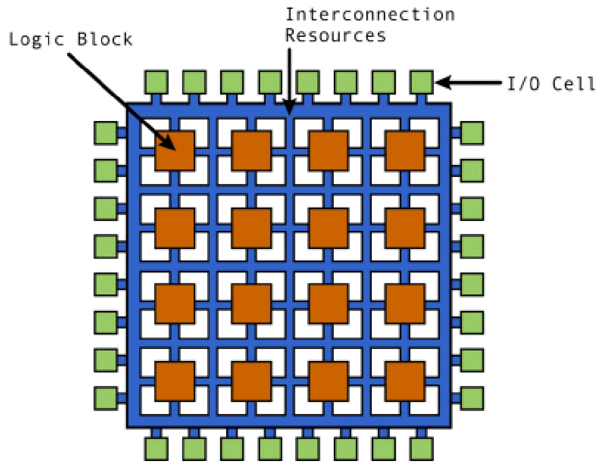
- Introduction to PYNQ-Z2
- Introduction to PWM
- Lab Problem
  - Part 1 - Behavioral Simulation
  - Part 2 - PYNQ Board Implementation

Outline

- Introduction to PYNQ-Z2
- Introduction to PWM
- Lab Problem
  - Part 1 - Behavioral Simulation
  - Part 2 - PYNQ Board Implementation

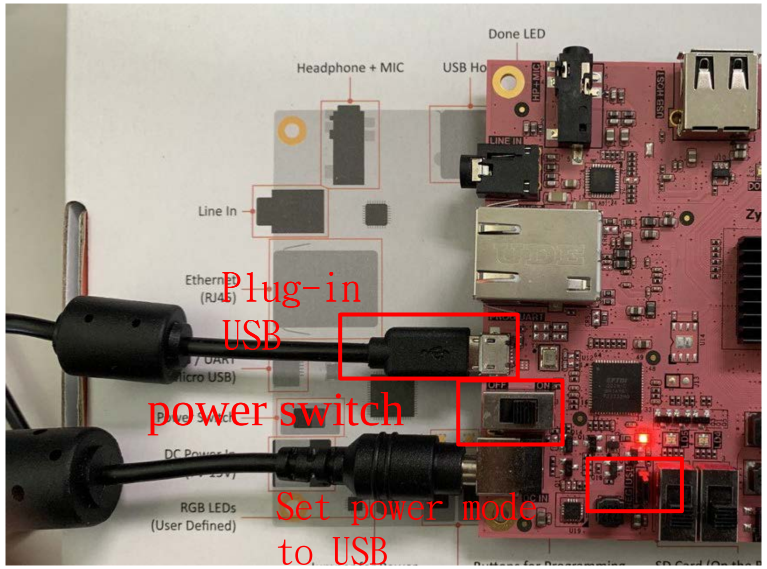
FPGA (Field Programmable Gate Array)

- 前幾次Lab中我們所完成的Verilog電路，若要被製作成真的電路，還必須經過幾個步驟。
  1. 邏輯合成(Logic Synthesis)，是將電路轉換為製造商可支援的標準元件(standard cell)型式。
  2. 布局和布線(Place & Route)，將轉換完的元件擺放和排線。
- 現場可程式化邏輯閘陣列 (Field Programmable Gate Array, FPGA)，是由許多的Logic block和接點組成的電路，透過燒錄的方式改變接點的開關，將其轉變為我們設計的電路。
- 過去幾周Lab所使用的Vivado，除了可以模擬電路，最主要的功能其實是將電路合成和PR後，燒錄至FPGA。

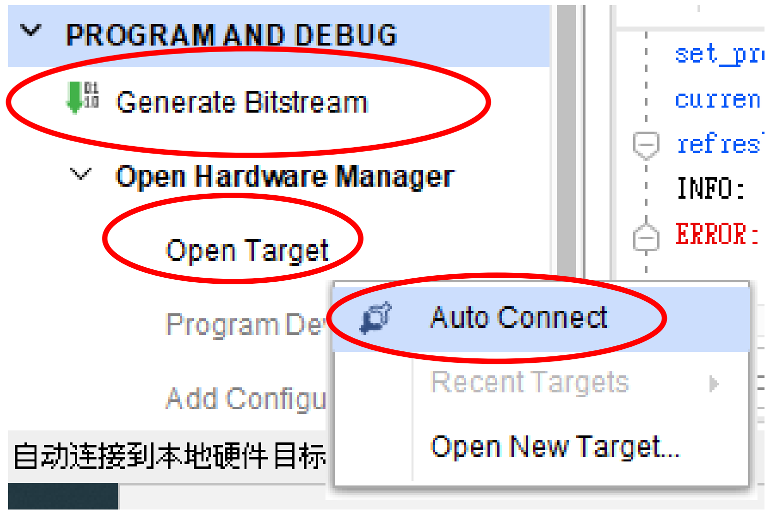


PYNQ-Z2 Board Connection Check

1. plug micro USB (**Use USB port as power supply**)
2. Turn on power switch
3. Connect to device
4. Check hardware panel

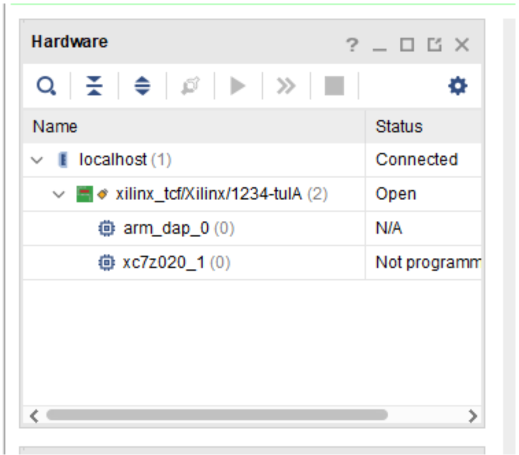


Connect to device(Left down manual)



```
set_pr
curren
refres
INFO:
ERROR:
```

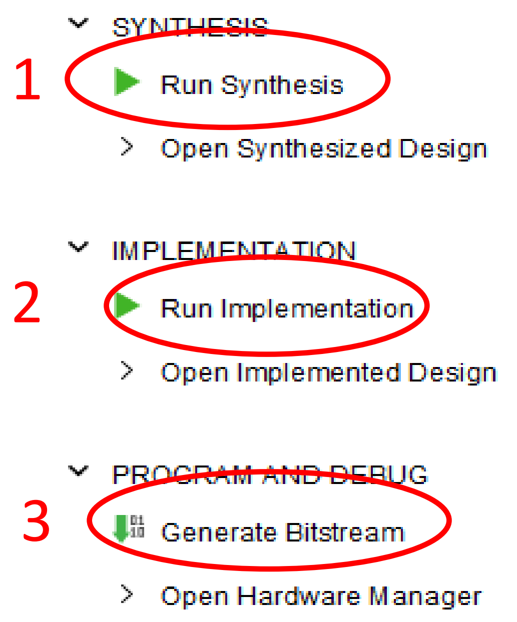
Hardware panel

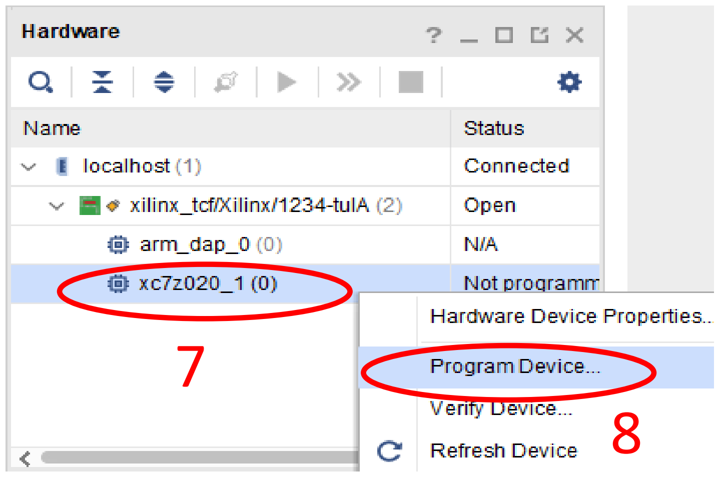


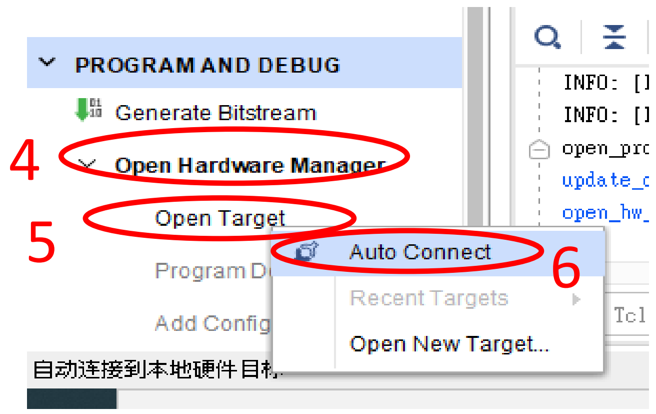
Hardware

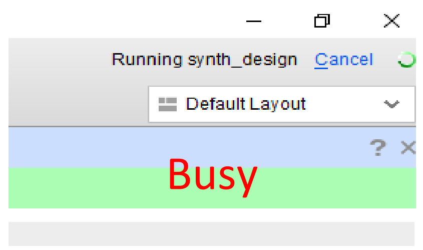
Name	Status
localhost (1)	Connected
xilinx_tcf/Xilinx/1234-tulA (2)	Open
arm_dap_0 (0)	N/A
xc7z020_1 (0)	Not programm

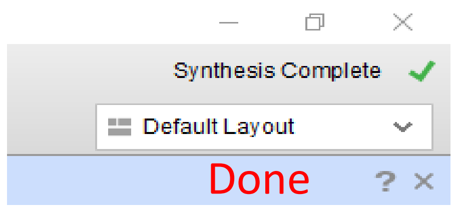
PYNQ Synthesis Procedure











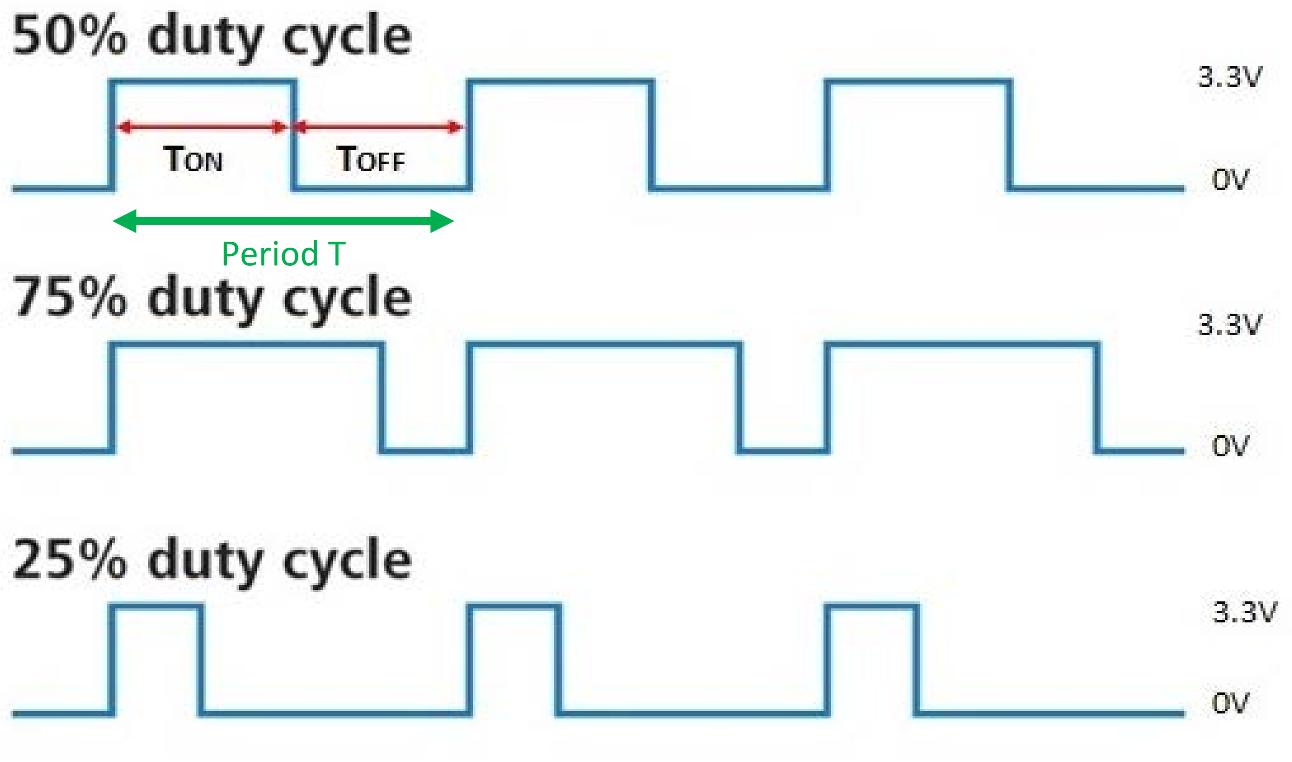
Outline

- Introduction to PYNQ-Z2
- Introduction to PWM
- Lab Problem
  - Part 1 - Behavioral Simulation
  - Part 2 - PYNQ Board Implementation

Duty Cycle

- Duty cycle : Ratio of cycle period is ON compared to the time is OFF

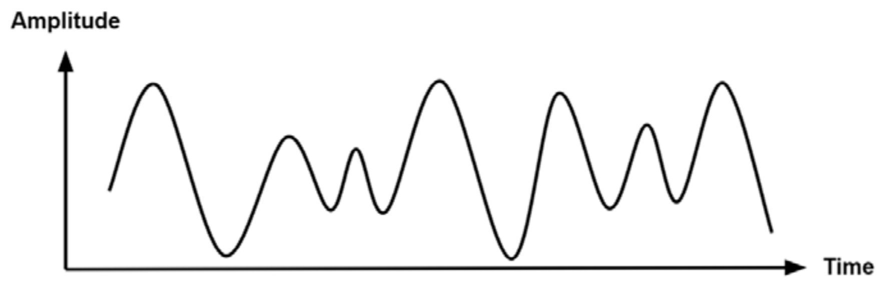
$$\text{Duty} = \frac{T_{ON}}{Period\ T}$$



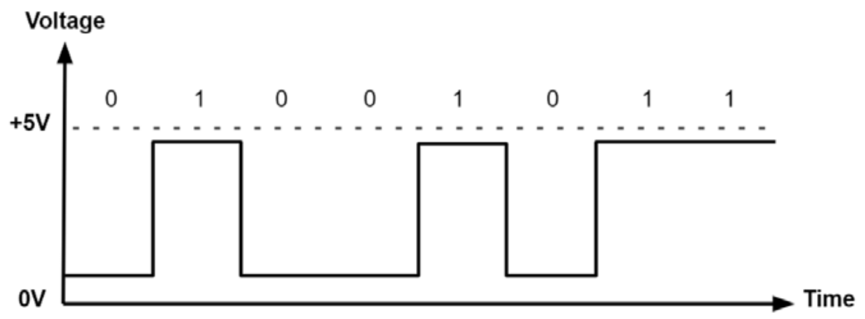
Pulse Width Modulation

- PWM is the mechanism of representing analog signals with digital signals by controlling the duty cycle of the periods.
- Analog Signal : Continuous
- Digital Signal : Discrete

Figure source : https://www.monolithicpower.com/en/analog-vs-digital-signal



Analog signal

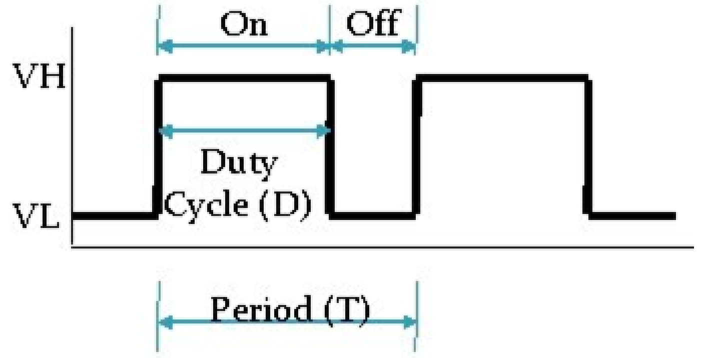


Digital signal

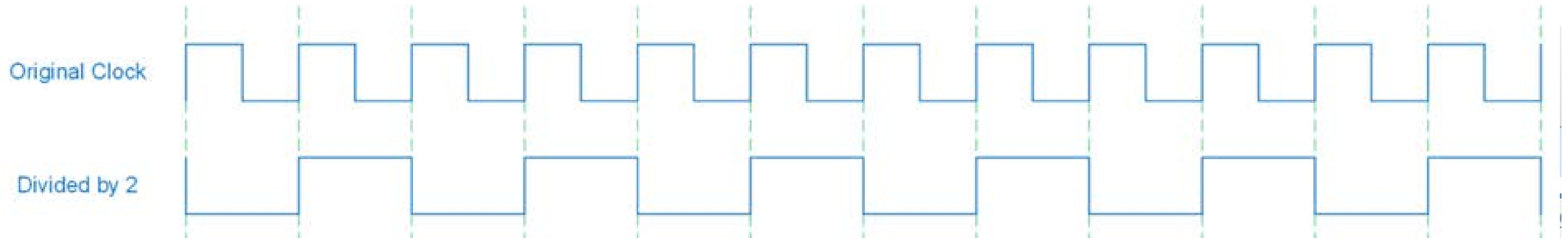
Pulse Width Modulation - Cont.

- Digital signal ACTIVATE/OFF represents switch ON/OFF of the circuit, controlling the average current of each period
- In general VL is zero; D is the duty; $V_{avg} = D \times V_H$
- The average value of the signal is directly dependent on the duty cycle D

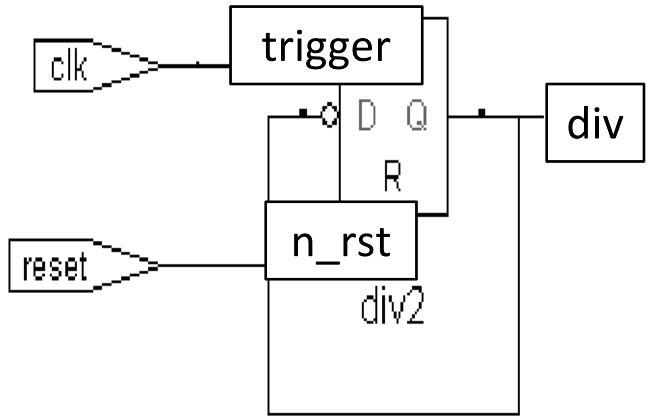
$$V_{avg} = D \cdot V_H + (1 - D) \cdot V_L$$



Clock Divider - 1



Divide clock **frequency** by 2



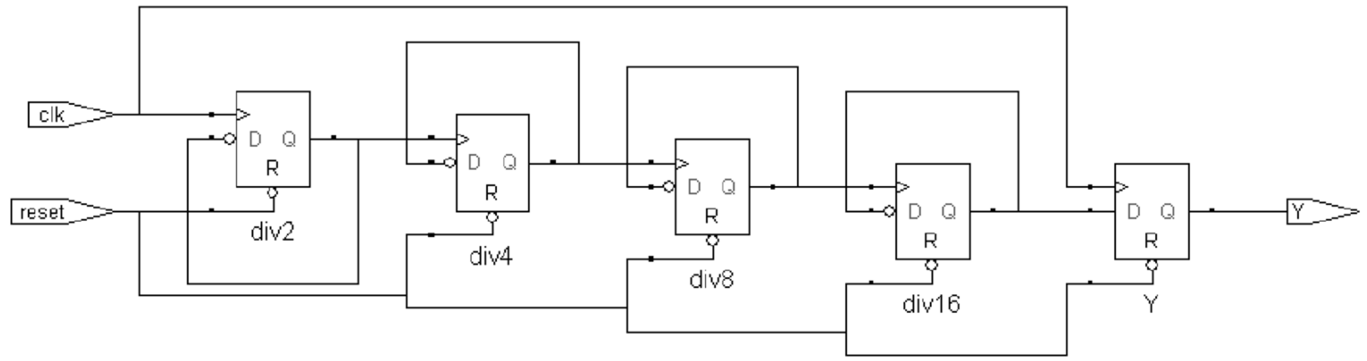
```
module dff_div(
    input n_rst,
    input trigger,
    output reg div
    );

always @(negedge trigger or negedge n_rst)begin
    if(~n_rst)begin
        div <= 1'b0;
    end else begin
        div <= ~div;
    end
end

endmodule
```

Clock Divider - 2

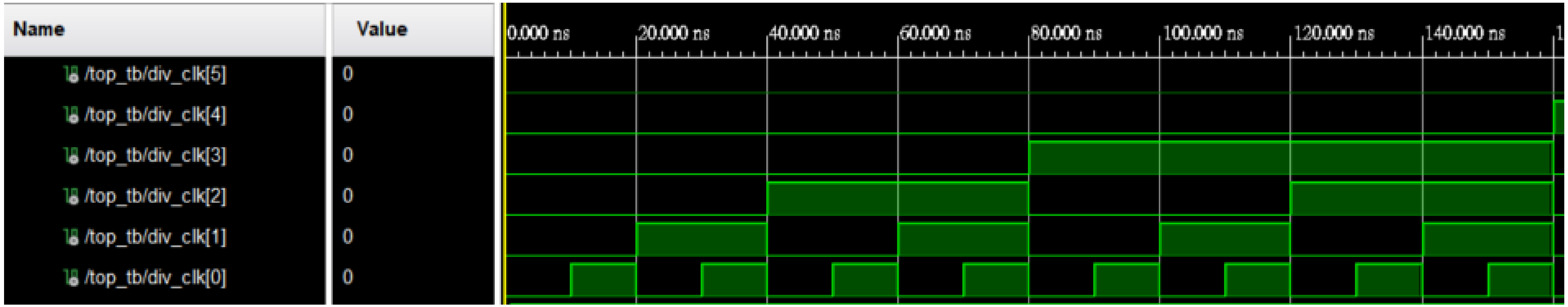
Clock divide by 2^n (n DFF個數)



Divided clock frequency

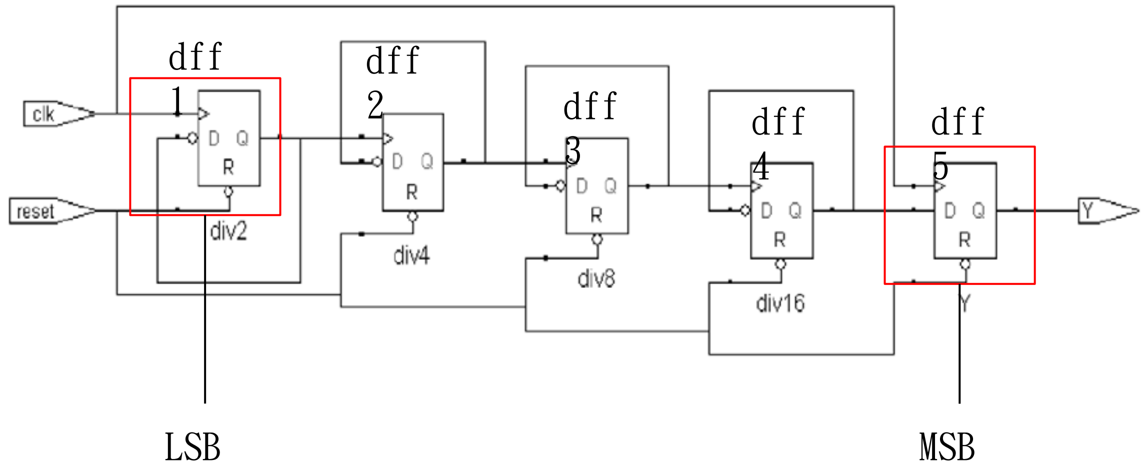
$$\text{freqY} = \frac{125\text{MHZ}}{2^n}$$

(n=#DFF)



Ripple Counter

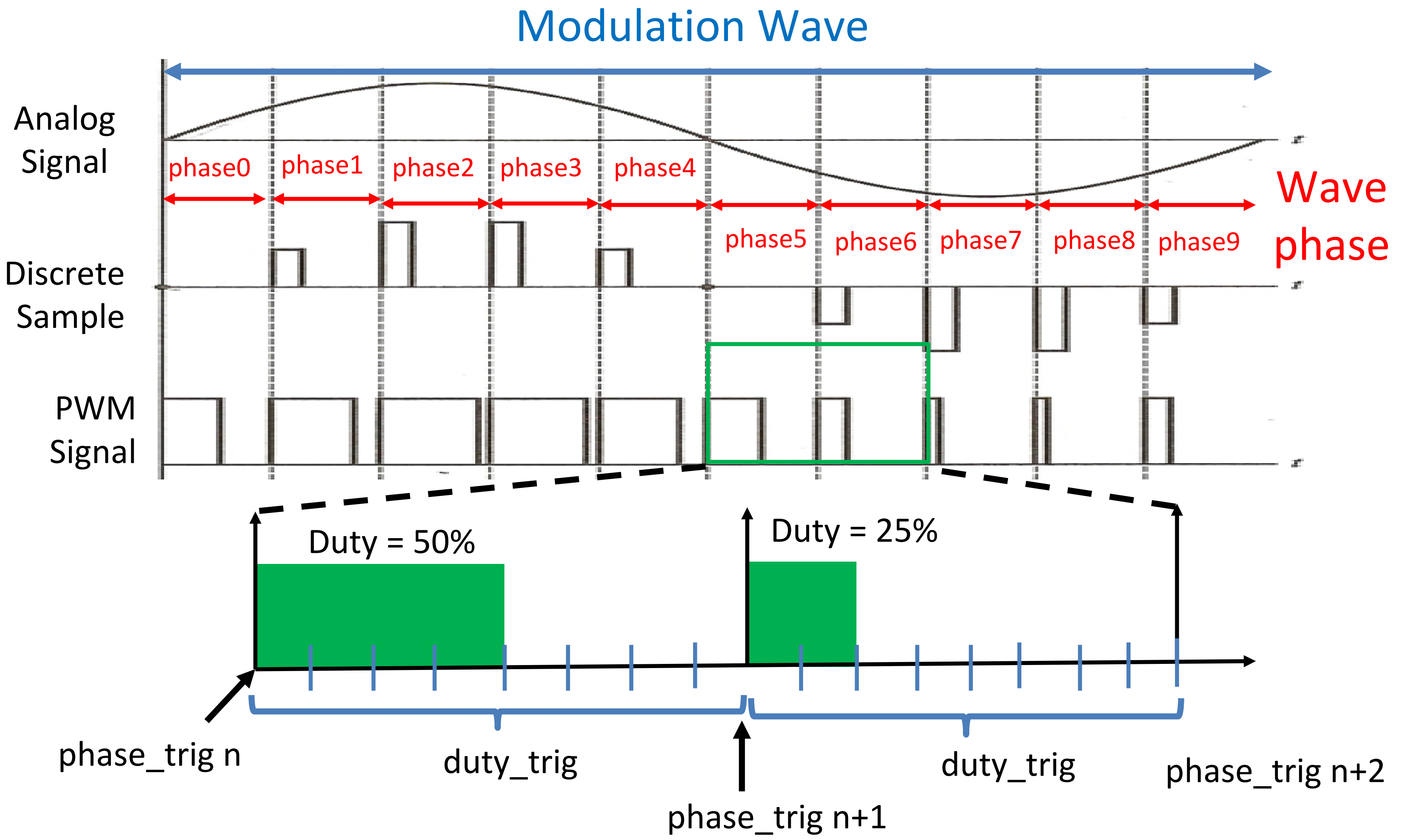
- 多個Clock Divider 串接，可以利用dff之間有著propagate的效果，將多個串接的dff視為一個counter；第一個的divider的output為LSB，而最後一個dff的output為MSB。



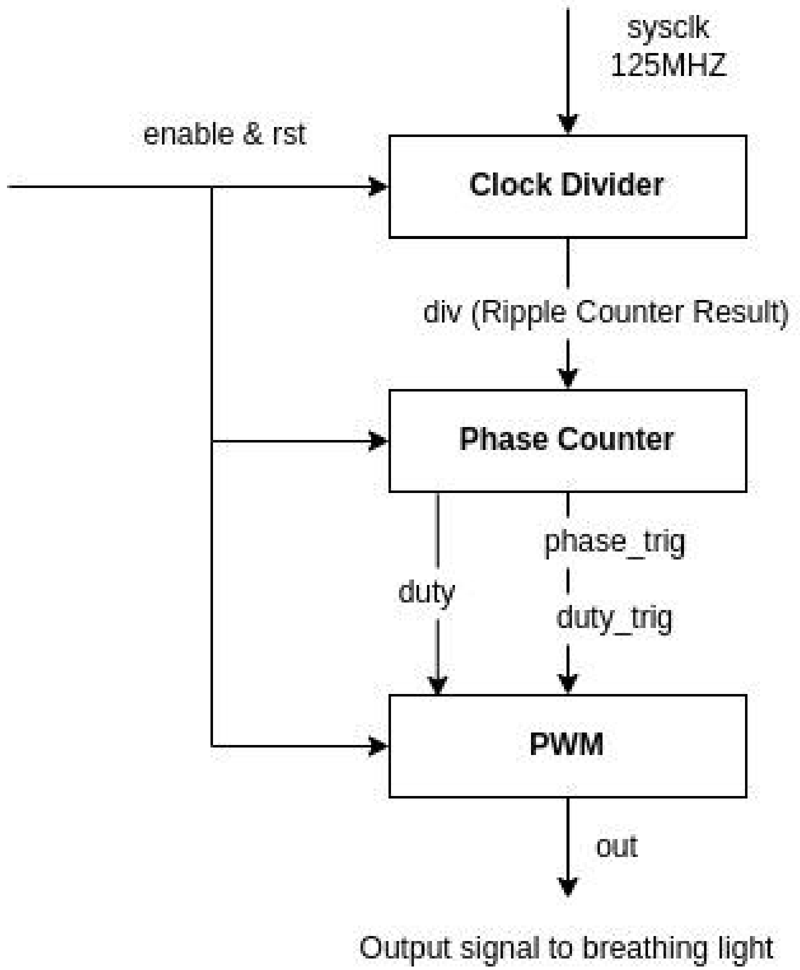
Cycle	dff5	dff4	dff3	dff2	dff1	Value
1	0	0	0	0	0	0
2	0	0	0	0	1	1
3	0	0	0	1	0	2
4	0	0	0	1	1	3
5	0	0	1	0	0	4
6	0	0	1	0	1	5

⋮

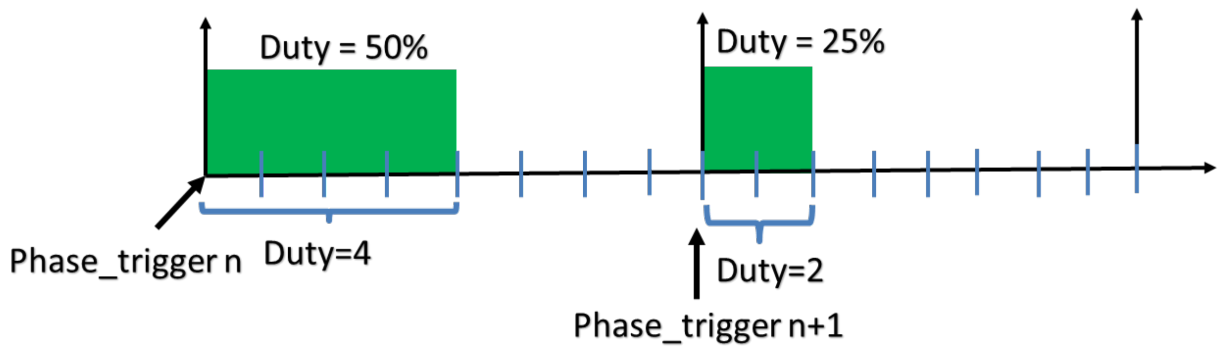
PWM Implementation



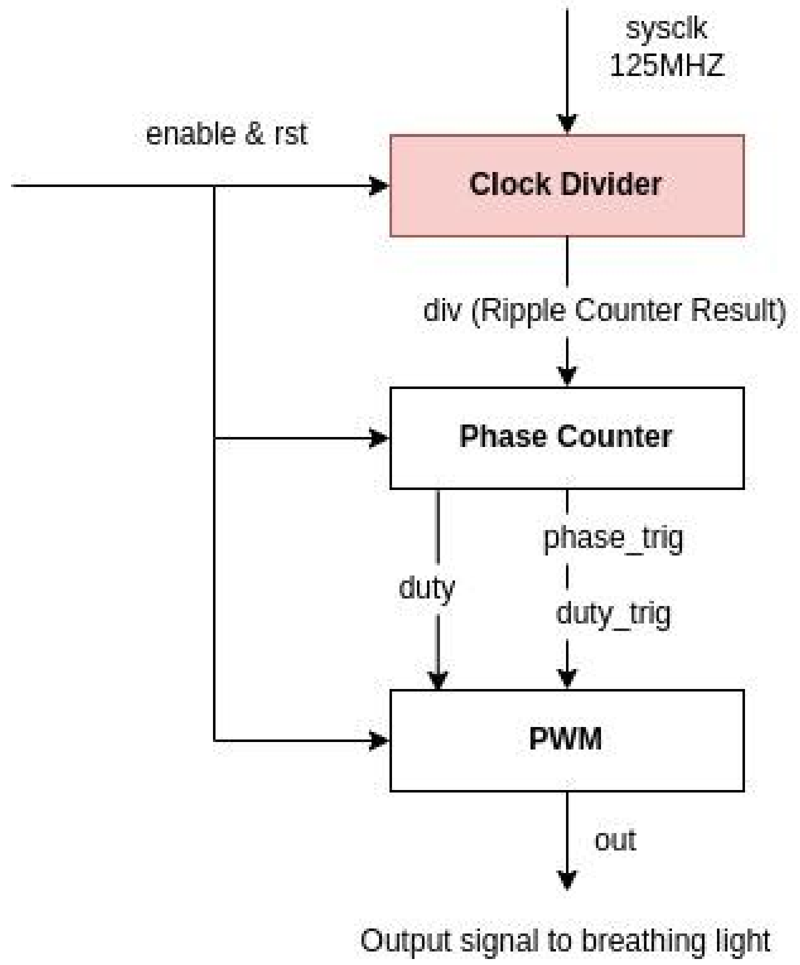
Simple PWM



- 由於sysclk的頻率為125MHZ，請先嘗試以clock divider將頻率降至適合操作的頻率
- 當phase的週期小於人眼圖像滯留的時間(約為40ms)，可以減少燈光閃爍的現象
- Duty 為每次phase trigger之後set high的cycles，下圖的一個phase有八個cycle，duty分別為4和2，因此兩個phase的duty ratio分別為50%和25%

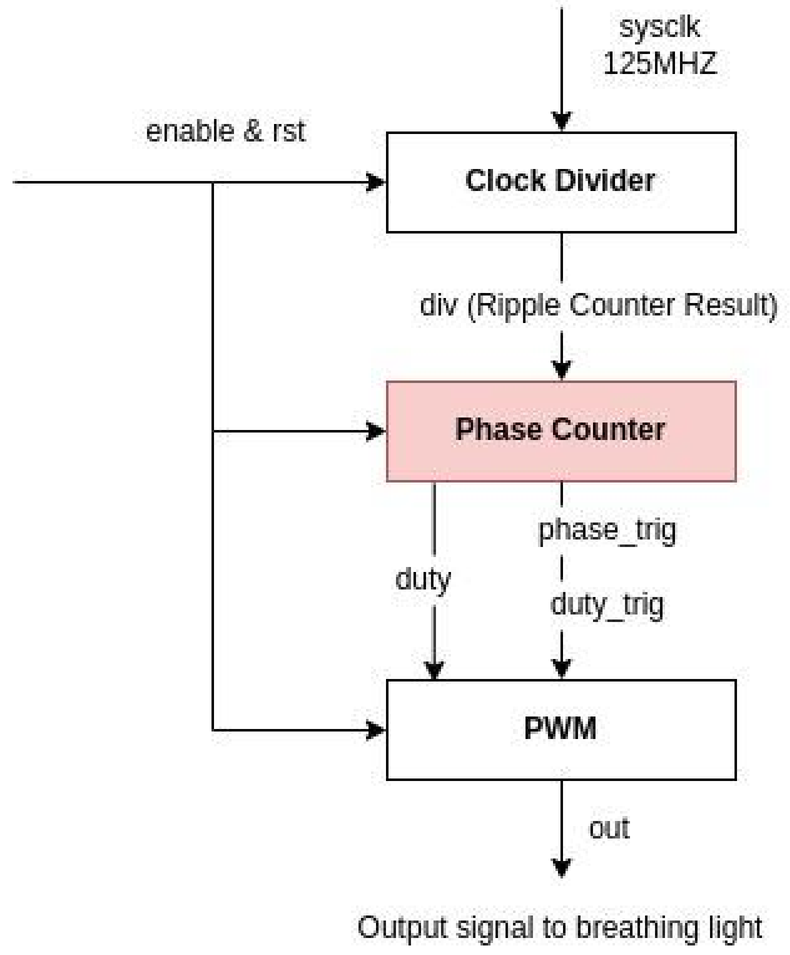


Simple PWM - Clock Divider



```
module ripple_div(
   input rst,
   input sysclk,
   output wire [31:0] div
   );
//Implement your code
endmodule
```

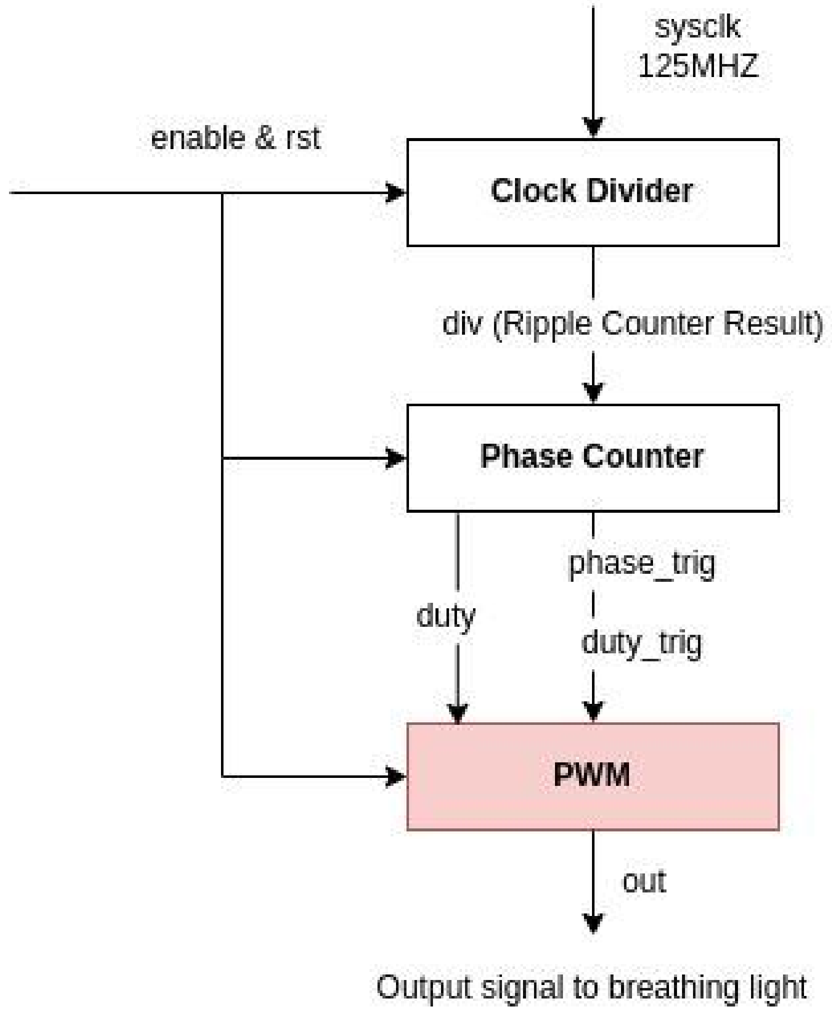
Simple PWM - Phase Counter



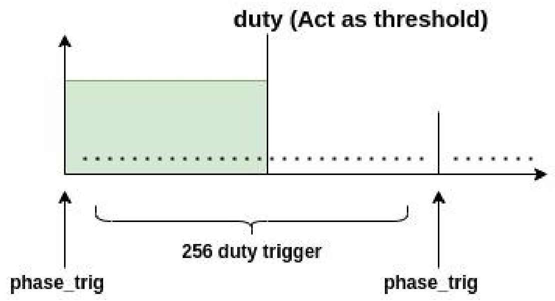
```
module phase_counter(
   input enable,
   input [7:0] phase_shift,
   input[31:0] div,
   output wire duty_trig,
   output wire phase_trig,
   output wire [7:0] duty
);
//Implement your code

endmodule
```

Simple PWM - 8 bit phase PWM



```
module pwm(
    input rst,
    input enable,
    input duty_trig,
    input phase_trig,
    input [7:0] duty,
    output wire out
);
reg[7:0] count;
assign out = ((count < duty) | phase_trig) & enable;
always@(posedge duty_trig or posedge rst)begin
    if(rst)begin
        count <= 8'b0;
    end else if(phase_trig)begin
        count <= 8'b0;
    end else if(duty_trig)begin
        count <= count + 1;
    end else begin
        count <= count;
    end
end
endmodule
```



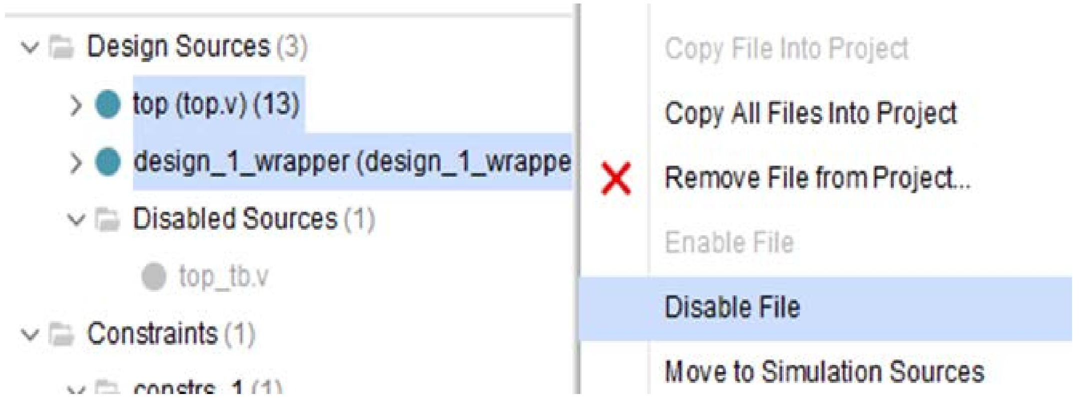
Outline

- Introduction to PYNQ-Z2
- Introduction to PWM
- Lab Problem
  - Part 1 - Behavioral Simulation
  - Part 2 - PYNQ Board Implementation

Configurations

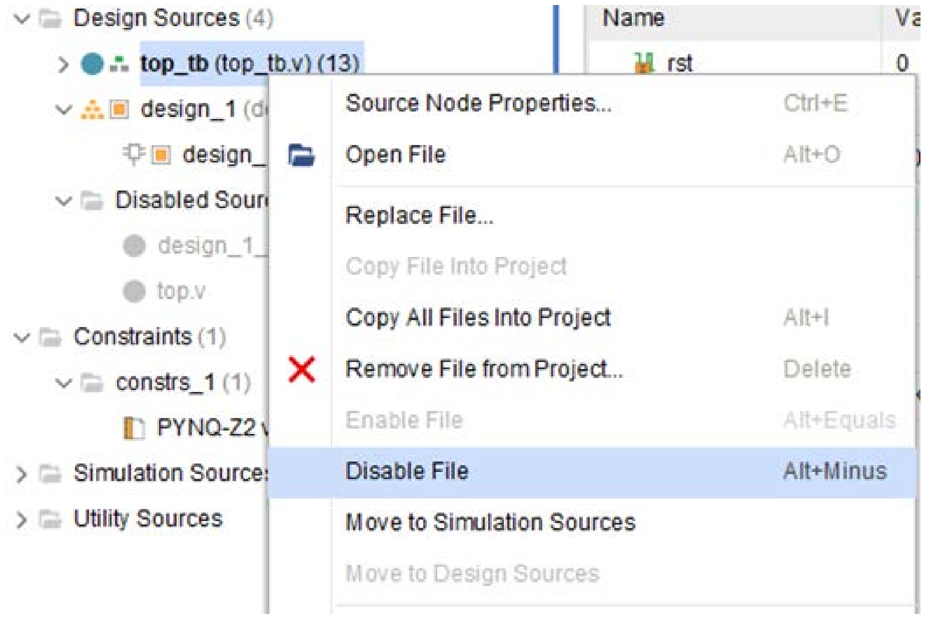
- Behavioral Simulation

1. Disable **top.v**
2. Disable **design_1_wrapper.v**
3. Enable **top_tb.v**
4. Set **top_tb.v** as top



- PYNQ Board Implementation

1. Disable **top_tb.v**
2. Enable **top.v**
3. Enable **design_1_wrapper.v**
4. Set **design_1_wrapper.v** as top



I/O port of module

- I/O port of module dff_div

Name	I/O	Width	Description
n_rst	Input	1	Active-low asynchronous reset signal
trigger	Input	1	Input clock
div	Output	1	Divided result

- I/O port of module ripple_div

Name	I/O	Width	Description
n_rst	Input	1	Active-low asynchronous reset signal
sysclk	Input	1	System clock (125 MHZ)
div	Output	32	Divided result

I/O port of module

- I/O port of module phase counter

Name	I/O	Width	Description
enable	Input	1	SW[0] | SW[1] on FPGA board
phase_shift	Input	8	Offset of duty
div_clk	Input	32	Ripple Counter Result
duty_trig	Output	1	Trigger on every 256 cycles of sysclk
phase_trig	Output	1	Triggered when the startup of phase (every 256 duty trig cycles)
duty	Output	8	Threshold of phase (Duty ratio)

- I/O port of module LUT_sin

Name	I/O	Width	Description
phase_idx	Input	8	Output of signal duty in phase counter
data	Output	8	Transform the threshold from linear to sine wave

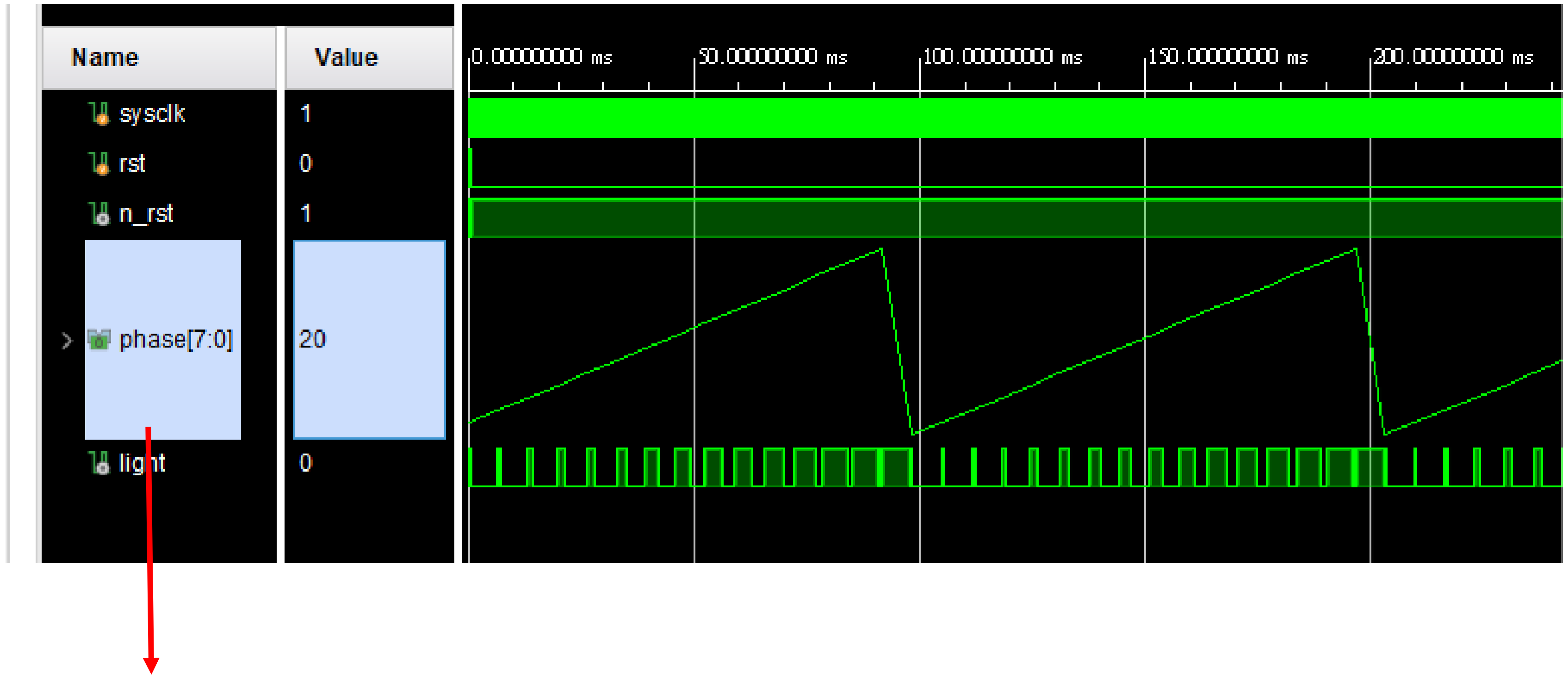
I/O port of module

- I/O port of module pwm

Name	I/O	Width	Description
rst	Input	1	Active-high asynchronous reset signal
enable	Input	1	SW[0] | SW[1] on FPGA board
duty_trig	Input	1	High every 256 cycles of sysclk
phase_trig	Input	1	High when the startup of phase (every 256 duty trig cycles)
duty	Input	8	Threshold of phase
out	Output	1	Switch ON/OFF of led on FPGA board

Behavioral Simulation - Problem 1

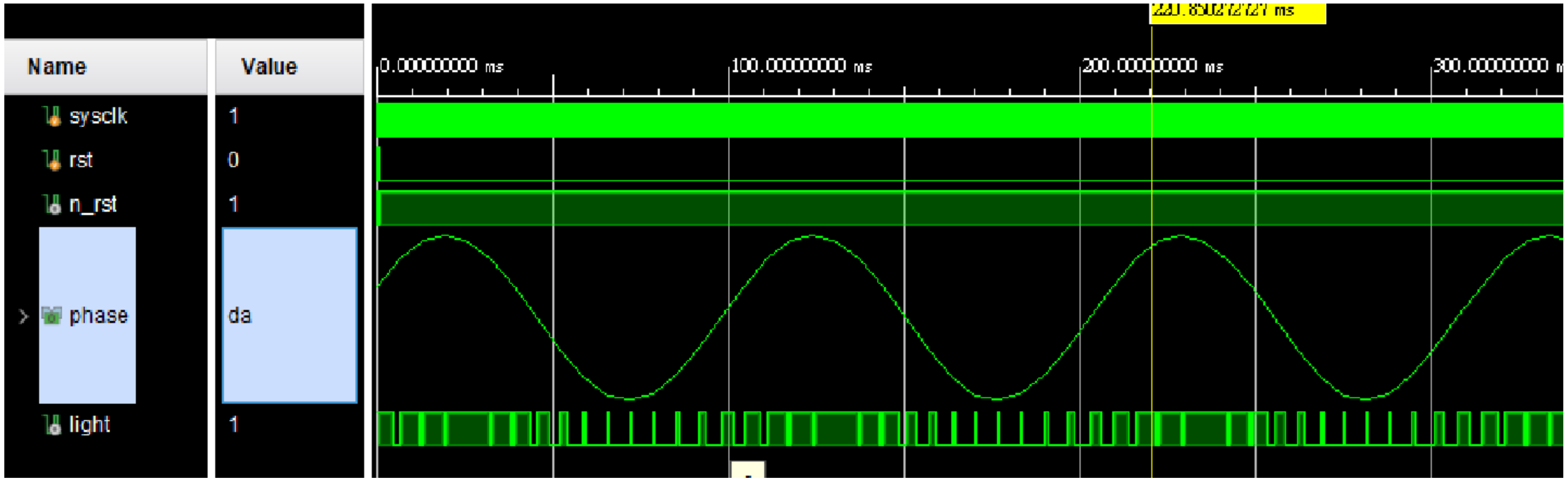
- 產生一組週期性的三角波，結報需解釋波型的產生方法與波型圖截圖(如下圖)。



右鍵 > Waveform Style > Analog

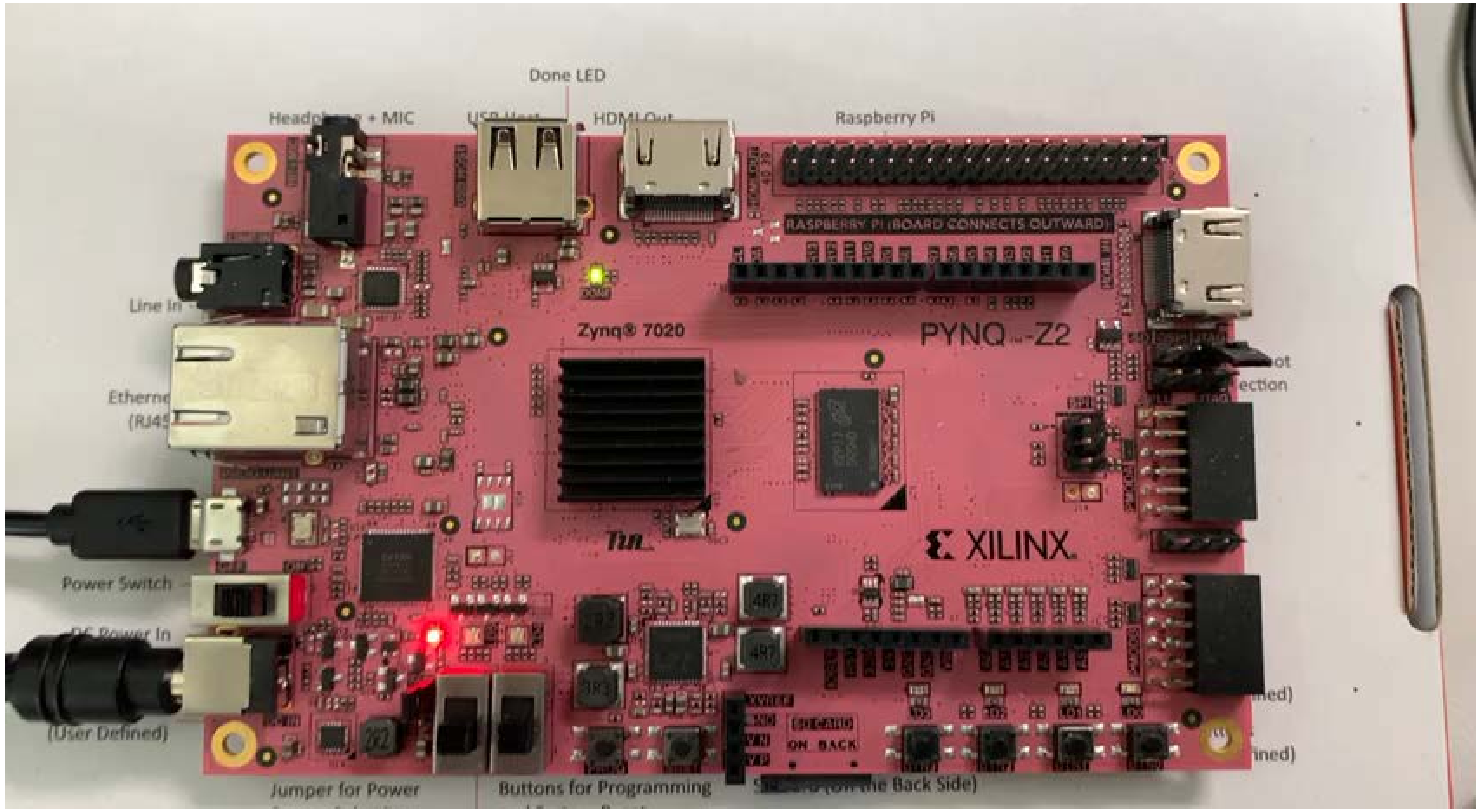
Behavioral Simulation - Problem 2

- 請利用 LUT_sin module 產生一組正弦
- 結報需解釋波型的產生方法與波型圖截圖(如下圖)。



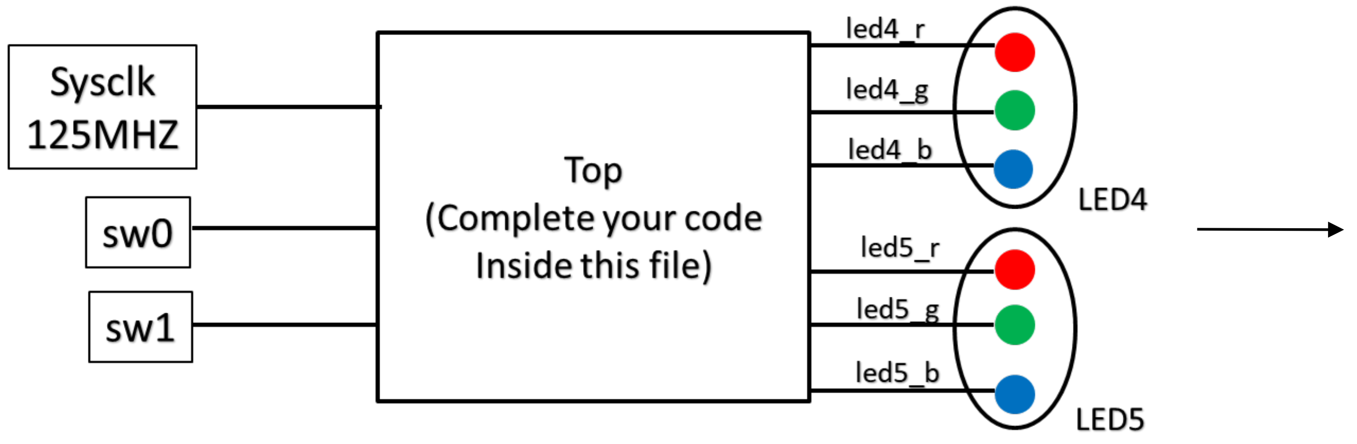
PYNQ Implementation - Breathing Light

- 白光呼吸燈
- 混色光呼吸燈



PYNQ Implementation - Breathing Light

- 實作top.v檔實作呼吸燈電路。可依需求修改 top檔案內非port name的部分(切記不可修改top.v 的port name，不然會合成失敗)。接著合成Bitstream檔並燒入到FPGA內。
- Note : FPGA內LED4、LED5內有三種獨立RGB LED燈。可自行挑選。沒有使用的output線路記得要在top module內接地。(Assign to 0)。



```
module top(
    input sysclk,
    input [1:0] sw,
    input [3:0] btn,
    output wire [3:0] led,
    output wire led4_b,
    output wire led4_g,
    output wire led4_r,
    output wire led5_b,
    output wire led5_g,
    output wire led5_r
);
```

本次結報需要附上的內容

- Behavioral Simulation
  - 波形產生解釋和截圖波形
  - 實驗心得
- PYNQ Board Implementation
  - 實驗心得(實作遇到什麼困難等等...)