

## 時序電路設計及應用 HW2

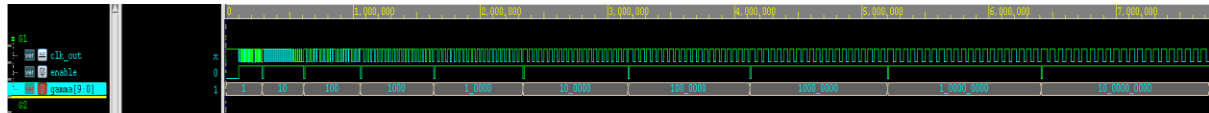
### Group 11

106061216 電機系 魏胤皓 Ying-Hao, Wei

106010006 電機系 黃詩瑜 Shih-Yu, Huang

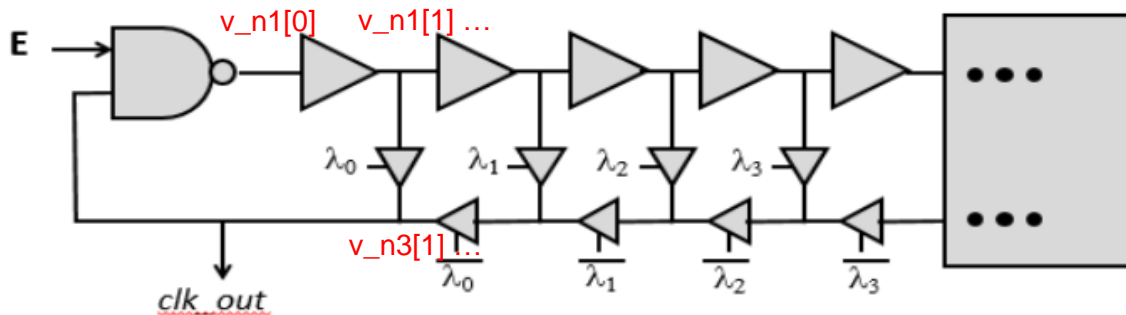
106061128 電機系 陳禾育 He-Yu, Chen

(a)



```
gamma code = 0000000001, period = 887.000000 , freq = 1.127396 GHz
gamma code = 0000000010, period = 1490.000000 , freq = 0.671141 GHz
gamma code = 0000000100, period = 2092.000000 , freq = 0.478011 GHz
gamma code = 0000001000, period = 2694.000000 , freq = 0.371195 GHz
gamma code = 0000010000, period = 3296.000000 , freq = 0.303398 GHz
gamma code = 0000100000, period = 3898.000000 , freq = 0.256542 GHz
gamma code = 0001000000, period = 4500.000000 , freq = 0.222222 GHz
gamma code = 0010000000, period = 5102.000000 , freq = 0.196002 GHz
gamma code = 0100000000, period = 5704.000000 , freq = 0.175316 GHz
gamma code = 1000000000, period = 6247.000000 , freq = 0.160077 GHz
```

我們使用 TSMC 90nm 製程，依據題目要求的 500MHz clock signal，選擇 10 個 buffer 的 longer buffer chain，control code 為 10 bits，gamma code 越大，經過的 buffer 越多造成 frequency 越小，上方圖為我們取 10 個 gamma code 對應的 period 與 frequency 資訊，可看出趨勢正確。



Path-Selection-Based RO

我們利用 generate fun.將 10 個 buffer 串在一起，使用 TBUF<sub>X1</sub> 當作最上排的 buffer，EN 為 1，使用 TBUF<sub>XL</sub> 當作中間與最下排的 buffer，gamma 是中間那排 buffer 的 EN，而 gamma code 經過 inverter 產生 v\_g2 訊號給下排的 buffer 當 EN。

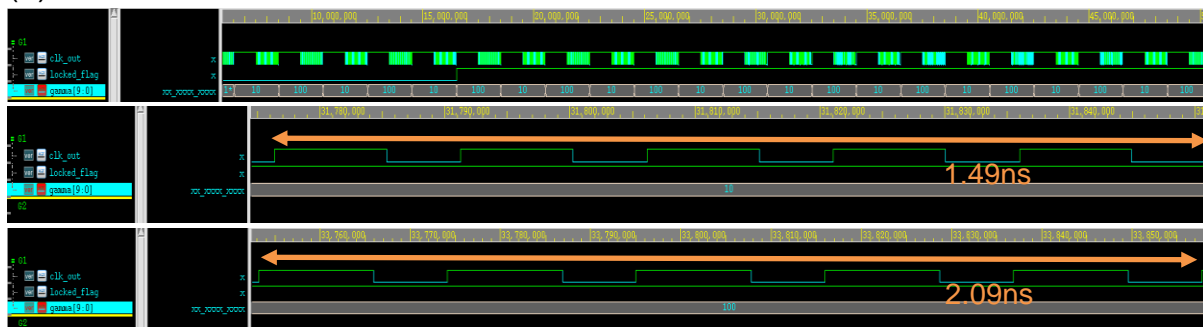
```

genvar i;
generate
  for(i=1;i<vertebra_num-1;i=i+1)begin:spine
    vertebra(
      .n1(v_n1[i]),
      .n2(v_n1[i+1]),
      .n4(v_n3[i+1]),
      .n3(v_n3[i]),
      .g1(gamma[i]),
      .g2(v_g2[i-1])
    );
  end
  for(i=0;i<vertebra_num-1;i=i+1)begin:inv_chain
    INVXL INV0(.A(gamma[i]), .Y(v_g2[i]));
  end
endgenerate

module vertebra(
  input  n1,
  output n2,
  input  n4,
  output n3,
  input  g1,
  input  g2
);
  TBUF01 BUF0(.A(n1), .OE(1'b1), .Y(n2));
  TBUF01 TB0(.A(n2), .OE(g1), .Y(n4));
  TBUF01 TB1(.A(n4), .OE(g2), .Y(n3));
endmodule

```

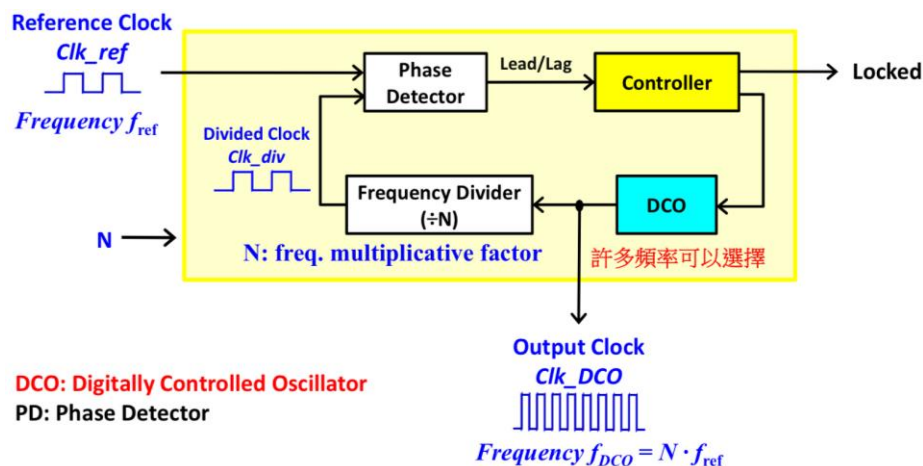
(b)



Gamma code	average clock cycle times (5 cycles)	error
2	1.49 ns / 671MHZ	25.5%
4	2.09 ns / 478MHZ	4.5%

Total average clock cycle times (10 cycles)	error
1.79 ns / 558MHZ	10.5%

要將 output clock 控制在 ideal period 2000ps，我們的 gamma code 會在 2 與 4 之間變動，我們取當 locked 訊號起來後，gamma 為 2 和 4 的 5 個 cycles 平均，與綜合的 10 個 cycles 平均，與 ideal clock 相比計算出 error。



frequency divider 參考講義的做法，利用 counter，當它數到 25 時會歸零，配合 EN 訊號產生 clk\_out。在做 post-layout 時，發現 CK 與 RN 訊號會有 setup time violation，便將控制 RN 訊號的 enable 先經過 delay chain 產生 enable\_delay 再給進 CK。

```

DFFRX2 counter_0 ( .D(counter_in[0]), .CK(clk), .RN(enable), .Q(counter[0]), .QN() );
DFFRX2 counter_1 ( .D(counter_in[1]), .CK(clk), .RN(enable), .Q(counter[1]), .QN() );
DFFRX2 counter_2 ( .D(counter_in[2]), .CK(clk), .RN(enable), .Q(counter[2]), .QN() );
DFFRX2 counter_3 ( .D(counter_in[3]), .CK(clk), .RN(enable), .Q(counter[3]), .QN() );
DFFRX2 counter_4 ( .D(counter_in[4]), .CK(clk), .RN(enable), .Q(counter[4]), .QN() );

DFFRX2 clk_out_0 ( .D(!clk_out), .CK( (counter == 0) && enable_delay ), .RN(enable), .Q(clk_out), .QN() );

delay_chain #(
    .r_width(20)
) enable_delay_0(
    .delay_in(enable),
    .delay_out(enable_delay)
);

always@(*)begin
    if(counter == 25)begin
        counter_in = 0;
    end
    else begin
        counter_in = counter +1;
    end
end
end

```

我們原本設計的 Delay chain 是使用 TBUF，但發現 pull up 與 pull down 速度相差太多，給予高頻的 input 訊號，經過多個 delay chain 後，會導致 pulse shrinking 過於嚴重，讓 output 訊號便為 1。

TBUF:

Delays at 25°C, 1.8V, Typical Process

Description	Intrinsic Delay (ns)								
	XL	X1	X2	X3	X4	X8	X12	X16	X20
A → Y↑	0.050	0.056	0.064	0.071	0.073	0.067	0.065	0.069	0.064
A → Y↓	0.106	0.059	0.079	0.086	0.084	0.079	0.075	0.078	0.073

查閱 TSMC Standard Cell Library Databook，找出適合的 INV 元件，選用較大的 size 製作 delay chain，其 pull up 與 pull down 速度差距較小，可以避免高頻 input pulse shrinking 效應。

INV:

Delays at 25°C, 1.8V, Typical Process

Description	Intrinsic Delay (ns)								
	XL	X1	X2	X3	X4	X8	X12	X16	X20
A → Y↑	0.023	0.023	0.020	0.022	0.018	0.018	0.122	0.119	0.117
A → Y↓	0.015	0.014	0.012	0.013	0.011	0.012	0.119	0.113	0.110

Controller 的 input 為 lead 訊號，使用 finite state machine，產生 output locked\_flag 與 gamma code，最初 gamma 值為 10'b000000001，當 lead 為 1 且 gamma 的 MSB 為 0 時，gamma code 會往左 shift 一位，值變大，當 lead 為 0 且 gamma 的 LSB 為 0 時，gamma code 會往右 shift 一位，值變小。使用 cnt counter 作為判斷 lead/lag 交替的訊號，當 counter 數到 3 則表示 lead 為 101，代表達到 locked 條件，將 locked\_flag 設為 1。

```

always@(posedge clk or negedge rstn)begin
    if(rstn == 0)begin
        state <= 0;
        gamma <= 1;
        enable <= 0;
        pre_lead <= 0;
    end
    else begin
        if(state == 0)begin
            gamma <= gamma;
            enable <= 0;
            state <= 1;
        end
        else if(state == 1)begin
            gamma <= gamma;
            enable <= 1;
            state <= 2;
        end
        else if(state == 2)begin
            state <= 1;
            enable <= 0;
            pre_lead <= lead;
            if(lead == 1 && !gamma[9])begin
                gamma <= (gamma << 1);
            end
            else if(lead == 0 && !gamma[0]) begin
                gamma <= (gamma >> 1);
            end
        end
    end
end

always@(posedge pre_lead or negedge rstn)begin
    if(rstn==0)begin
        cnt <= 0;
        locked_flag <= 0;
    end
    else begin
        if(~locked_flag)begin
            cnt <= cnt + 1;
            if(cnt==2'd3) begin
                locked_flag <= 1;
            end
        end
    end
end

```

PD 我們選擇有 racing circuit 的那組，使用 standard cell 建置，可合成且成功 layout，PD 判斷 input clk\_in 與 FB 兩個訊號，FB 為經過 frequency divider 的 output，PD 會輸出 output lead 訊號給 controller。

```

//input stage
DFFRX2 FB_diff(
    .Q(up),
    .QN(),
    .D(1'b1),
    .CK(FB),
    .RN(reset)
);

DFFRX2 clk_in_dff(
    .Q(down),
    .QN(),
    .D(1'b1),
    .CK(clk_in),
    .RN(reset)
);

/*
DFFSRX2 FB_dff ( .D(1'b1), .E(reset), .CK(FB), .Q(up) );
DFFSRX2 clk_in_dff ( .D(1'b1), .E(reset), .CK(clk_in), .Q(down) );
*/

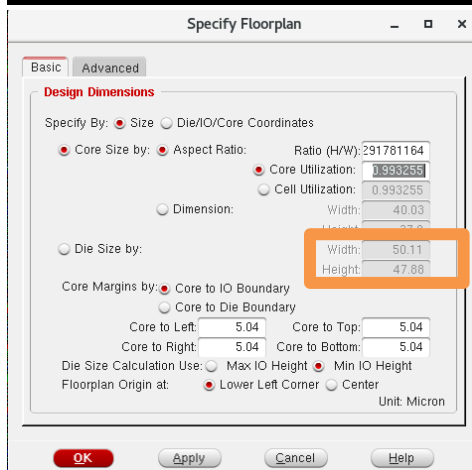
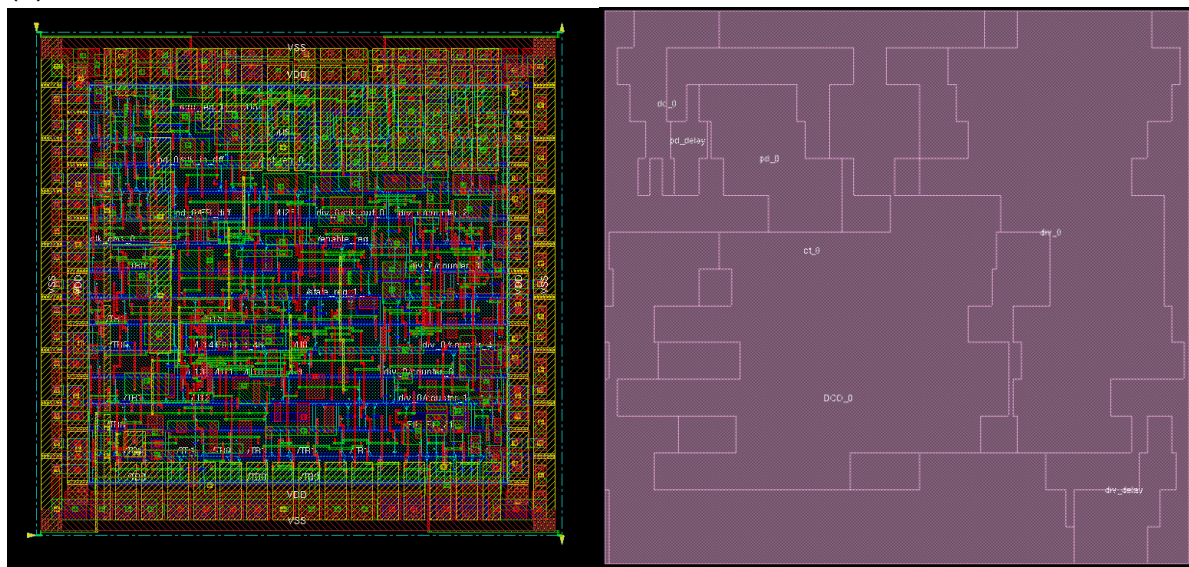
//reset circuit
NAND2X4 NAND0(.A(up), .B(down), .Y(reset_en));
NAND2X4 NAND1(.A(up), .B(down), .Y(reset_dummy));
AND2X4 U22(.A(reset_en), .B(enable), .Y(reset));

//racing circuit
NAND2X4 NANDA0(.A(up), .B(R), .Y(S));
NAND2X4 NANDB0(.A(down), .B(S), .Y(R));

//sr latch
NAND2X4 NANDA1(.A(S), .B(dummy), .Y(LEAD));
NAND2X4 NANDB1(.A(R), .B(LEAD), .Y(dummy));

```

(c)



Layout size (Width\*Height) : 50.11\*47.88 (um<sup>2</sup>)

進行 post-layout 時，發現 tool 會自動把我們用 INV 建立的 delay chain 化簡，因此我們使用了 setDoAssign off 指令避免這個情況。

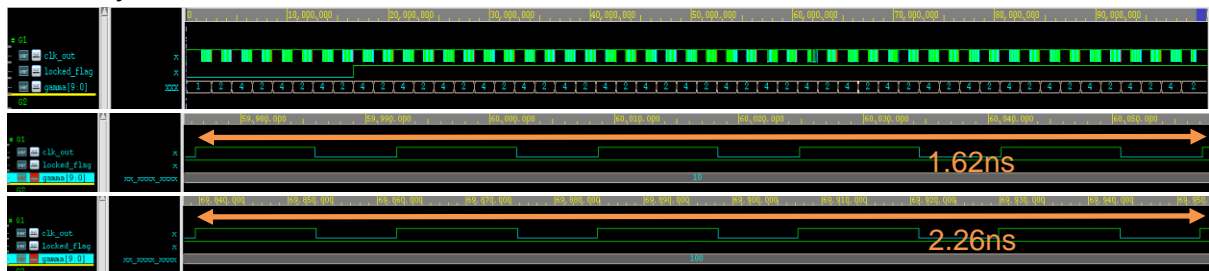
(d)

Pre-layout:

Gamma code	average clock cycle times (5 cycles)	error
2	1.49 ns / 671MHZ	25.5%
4	2.09 ns / 478MHZ	4.5%

Total average clock cycle times (10 cycles)	error
1.79 ns / 558MHZ	10.5%

Post-layout:



Gamma code	average clock cycle times (5 cycles)	error
2	1.62 ns / 617MHZ	19%
4	2.26 ns / 442MHZ	13%

Total average clock cycle times (10 cycles)	error
1.94 ns / 515MHZ	3%

	Pre-layout	Post-layout
oscillation frequency	1.79 ns / 558MHZ	1.94 ns / 515MHZ

Post-layout 的結果因為在 gamma code 2/4 的 error 剛好比較平均，10 個 cycles 平均下來的 frequency 較接近 ideal clock 500MHZ。

組員分工:

魏胤皓 : Verilog coding/simulation

黃詩瑜 : 報告撰寫

陳禾育 : APR