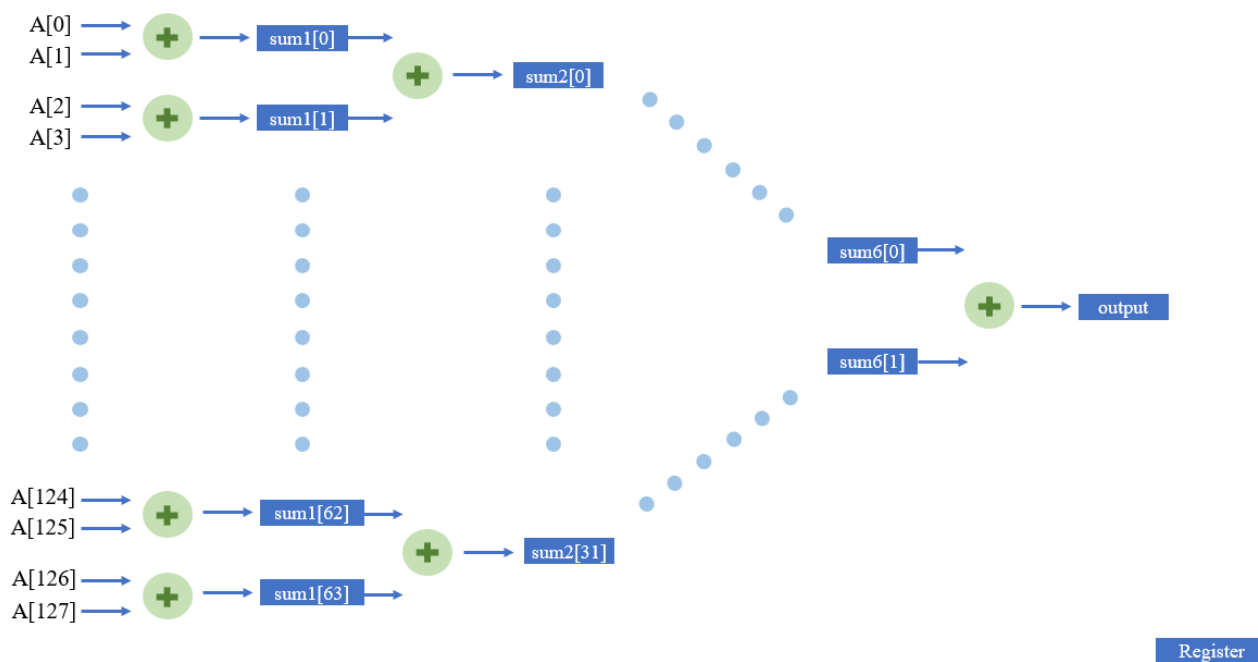


題目:製作一個 Adder tree，功能為輸入 128 個 0 或 1，並輸出多少 1 被輸入並試圖優化。比較改變數量與面積及延遲的關係，最後比較 V_{DD} 對 $\text{delay} \times \text{power}$ 的關係，選決定出最好的架構與 V_{DD} 並使用 HSPICE 模擬，並驗證操作頻率在 2GHz 以上。

Adder_tree

Adder_tree.v

首先，設定 adder_tree 的架構，並預期我們會得到一個兩兩相加且總共有七層的 adder_tree，block diagram 如下圖一。



圖一、adder_tree 的 block diagram

從 block diagram 中我們可以看到總共有 128 個 input，而每兩兩相加一次後會傳送到 sum1，sum1 則會是 64 個 2bit 的 register，而再兩兩相加後會得到 sum2，其為 32 個 4bit 的 register 一直類推到最後會出現 1 個 128bit 的 output

程式方面，對應的 Verilog 部分如下圖二。

```
genvar i;
generate
for(i = 0;i < 64;i = i+ 1) begin
    always @(posedge clk) begin
        sum1[i] <= A[i*2] + A[i*2 + 1];
    end
end

for(i = 0;i < 32;i = i+ 1) begin
    always @(posedge clk) begin
        sum2[i] <= sum1[i*2] + sum1[i*2 + 1];
    end
end

for(i = 0;i < 16;i = i+ 1) begin
    always @(posedge clk) begin
        sum3[i] <= sum2[i*2] + sum2[i*2 + 1];
    end
end

for(i = 0;i < 8;i = i+ 1) begin
    always @(posedge clk) begin
        sum4[i] <= sum3[i*2] + sum3[i*2 + 1];
    end
end

for(i = 0;i < 4;i = i+ 1) begin
    always @(posedge clk) begin
        sum5[i] <= sum4[i*2] + sum4[i*2 + 1];
    end
end

for(i = 0;i < 2;i = i+ 1) begin
    always @(posedge clk) begin
        sum6[i] <= sum5[i*2] + sum5[i*2 + 1];
    end
end

endgenerate
```

圖二、Verilog 中對應 block diagram 程式圖

Pattern.v

而後我們製作 pattern，以 input 亂數的方式來驗證邏輯是否正確。程式部分如下圖三。

```
for(i=0;i < 10;i=j+1)begin
  A = $random;
  Golden_Output = 0;
  $monitor($time, "clk:%d", clk);
  for(i=0;i <128;i=i+1)begin
    Golden_Output = A[i] + Golden_Output;
  end
end
```

圖三、Pattern 輸入亂數程式圖

Testbed.v

之後，使用 Testbed 驗證 RTL，將 adder_tree.v 與 Pattern include 進來，並將輸入接線設為 128bit 的 A 以及輸出 8bit Output，程式部分如下圖四。

```
module TESTBED;
  //input
  wire [127:0]A;
  //output
  wire [7:0] Output;
endmodule
```

圖四、testbed 輸入輸出接口程式圖

SYN

之後我們進行合成，得到 area 與 time 的結果如下圖五、六。

```
*****
Report : area
Design : Adder_tree
Version: R-2020.09
Date   : Thu Jan 12 23:52:13 2023
*****

Library(s) Used:

  asap7sc7p5t_SIMPLE_RVT_TT_08302018 (File: /RAID2/COURSE/DIC/DIC153/final/02_SYN/asap7sc7p5t_SIMPLE_RVT_TT_08302018.db)
  asap7sc7p5t_SEQ_RVT_TT_08302018 (File: /RAID2/COURSE/DIC/DIC153/final/02_SYN/asap7sc7p5t_SEQ_RVT_TT_08302018.db)

Number of ports:          137
Number of nets:           997
Number of cells:          868
Number of combinational cells: 494
Number of sequential cells: 374
Number of macros/black boxes: 0
Number of buf/inv:        0
Number of references:      7

Combinational area:       1028.998065
Buf/Inv area:             0.000000
Noncombinational area:    1744.934334
Macro/Black Box area:     0.000000
Net Interconnect area:    undefined (No wire load specified)

Total cell area:          2773.932398
Total area:               undefined
1
```

圖五、合成後 area 結果圖

Operating Conditions: PVT_0P7V_25C Library: asap7sc7p5t_INVBUF_RVT_TT_08302018
Wire Load Model Mode: top

Startpoint: sum6_reg[1][0]
(rising edge-triggered flip-flop clocked by clk)
Endpoint: Output_reg[6]
(rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

| Point | Incr | Path |
|---|--------|----------|
| ----- | | |
| clock clk (rise edge) | 0.00 | 0.00 |
| clock network delay (ideal) | 0.00 | 0.00 |
| sum6_reg[1][0]/CLK (DFFHQNx1_ASAP7_75t_R) | 0.00 | 0.00 r |
| sum6_reg[1][0]/QN (DFFHQNx1_ASAP7_75t_R) | 48.40 | 48.40 f |
| U946/Y (AND2x2_ASAP7_75t_R) | 27.60 | 76.00 f |
| U947/Y (MAJx2_ASAP7_75t_R) | 30.25 | 106.25 f |
| U948/Y (MAJx2_ASAP7_75t_R) | 32.11 | 138.36 f |
| U949/Y (MAJx2_ASAP7_75t_R) | 32.12 | 170.49 f |
| U950/Y (MAJx2_ASAP7_75t_R) | 32.12 | 202.61 f |
| U951/Y (MAJx2_ASAP7_75t_R) | 32.12 | 234.73 f |
| U953/SN (F4x1_ASAP7_75t_R) | 34.15 | 268.87 r |
| Output_reg[6]/D (DFFHQNx1_ASAP7_75t_R) | 0.00 | 268.87 r |
| data arrival time | | 268.87 |
| | | |
| clock clk (rise edge) | 400.00 | 400.00 |
| clock network delay (ideal) | 0.00 | 400.00 |
| Output_reg[6]/CLK (DFFHQNx1_ASAP7_75t_R) | 0.00 | 400.00 r |
| library setup time | -18.98 | 381.02 |
| data required time | | 381.02 |
| ----- | | |
| data required time | | 381.02 |
| data arrival time | | -268.87 |
| ----- | | |
| slack (MET) | | 112.14 |

圖六、合成後 time 結果圖

至此，再通過 gate-level simulation 後，我們已大致上完成了我們的 adder_tree，接下來我們將透過幾個變因，以優化 adder_tree 的速度、面積、功耗。

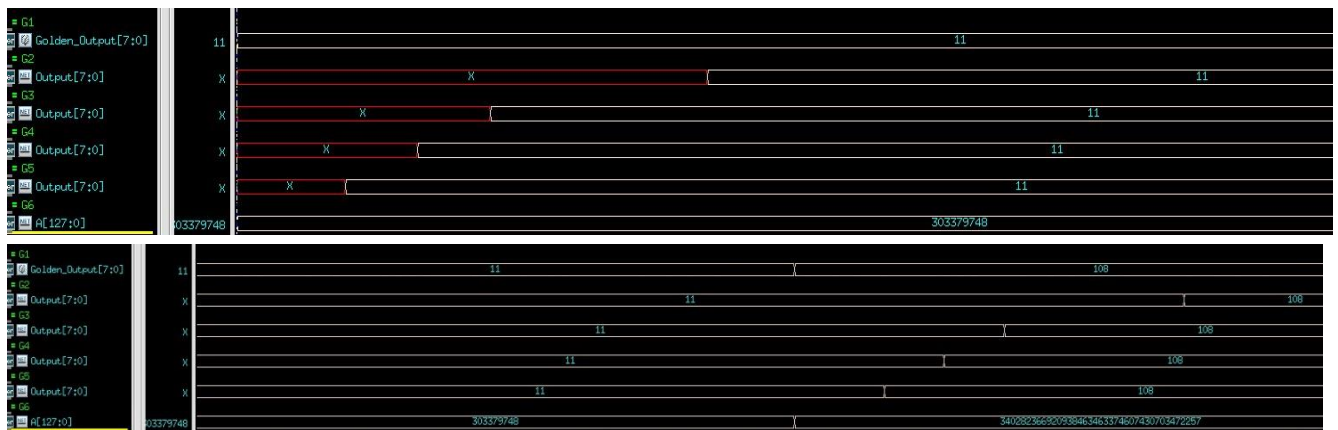
Adder_tree_optimize

Add number- $\text{Power} \times \text{Delay}$

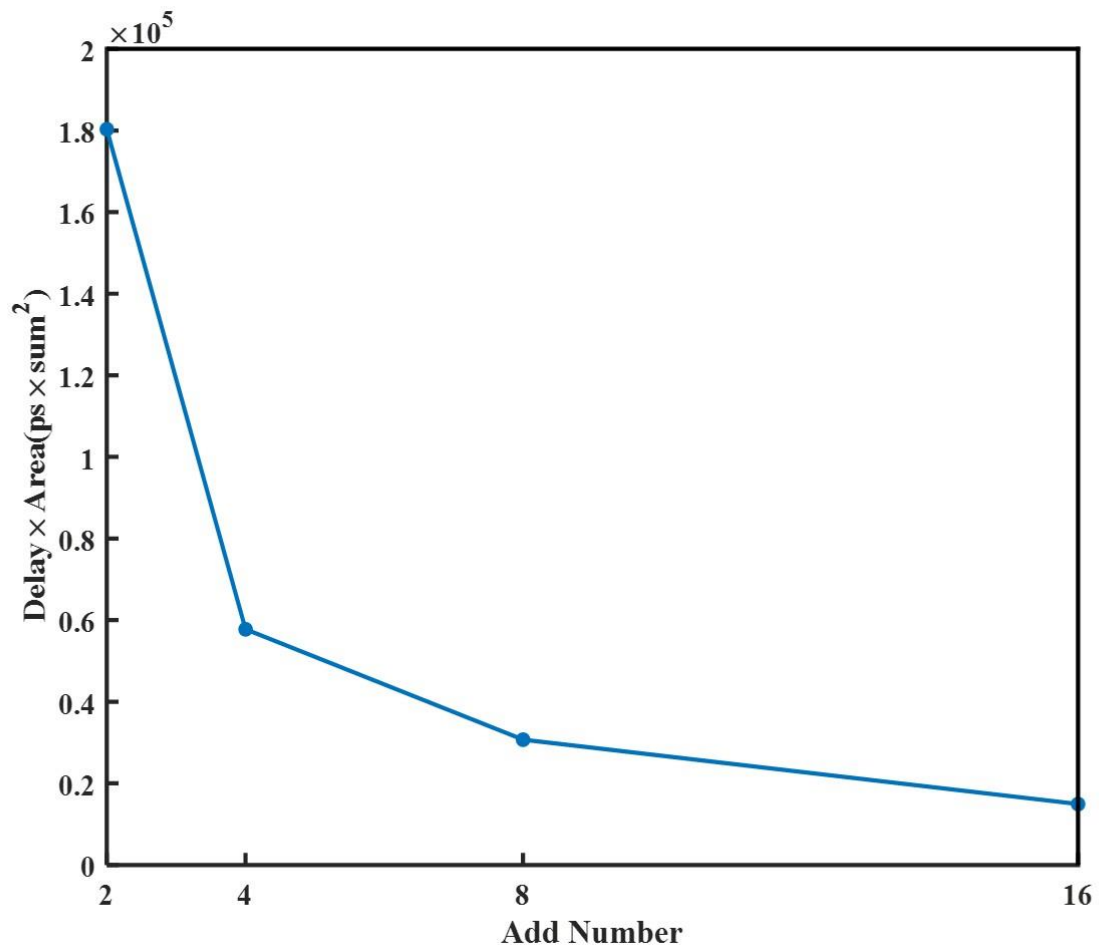
優化的部分，我們將改變每次相加的數量，並在相同的 time period 下合成，藉由改變每次相加的數量(定義為 Add number)與 $\text{power} \times \text{delay}$ 的關係圖，我們總共做出了四種電路分別為

Add_number 為 2、4、8、16，並選出 $\text{power} \times \text{delay}$ 最小的電路，而我們將合成後的電路進行 gate level 驗證，並將結果以 nWave 輸出如下圖七。而 add number 與 $\text{power} \times \text{delay}$ 的關係如下

圖八



圖七、通過 gate-level simulation 之 nWave 圖



圖八、add number 與 power×delay 關係圖

SYN

我們選用 add number 為 16 的電路作為我們優化的電路架構，透過將要求的時間不斷調小，直到合成可接受的最小值，此時電路的 area 與 time report 如下圖九、圖十。

```

*****
Report : area
Design : Adder_tree
Version: R-2020.09
Date   : Sun Jan 15 07:30:29 2023
*****

Library(s) Used:

  asap7sc7p5t_SIMPLE_RVT_TT_08302018 (File: /RAID2/COURSE/DIC/DIC143/final/optimizel6/SYN/asap7sc7p5t_SIMPLE_RVT_TT_08302018.db)
  asap7sc7p5t_INVBUFF_RVT_TT_08302018 (File: /RAID2/COURSE/DIC/DIC143/final/optimizel6/SYN/asap7sc7p5t_INVBUFF_RVT_TT_08302018.db)
  asap7sc7p5t_SEQ_RVT_TT_08302018 (File: /RAID2/COURSE/DIC/DIC143/final/optimizel6/SYN/asap7sc7p5t_SEQ_RVT_TT_08302018.db)

Number of ports:          137
Number of nets:           656
Number of cells:          476
Number of combinational cells: 428
Number of sequential cells:  48
Number of macros/black boxes: 0
Number of buf/inv:        95
Number of references:      21

Combinational area:       769.823990
Buf/Inv area:             70.217281
Noncombinational area:    226.281592
Macro/Black Box area:     0.000000
Net Interconnect area:    undefined (No wire load specified)

Total cell area:          996.105583
Total area:               undefined
1

```

圖九、合成後 area 結果圖

```

Startpoint: A[0] (input port clocked by clk)
Endpoint: sum1_reg[0][2]
           (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

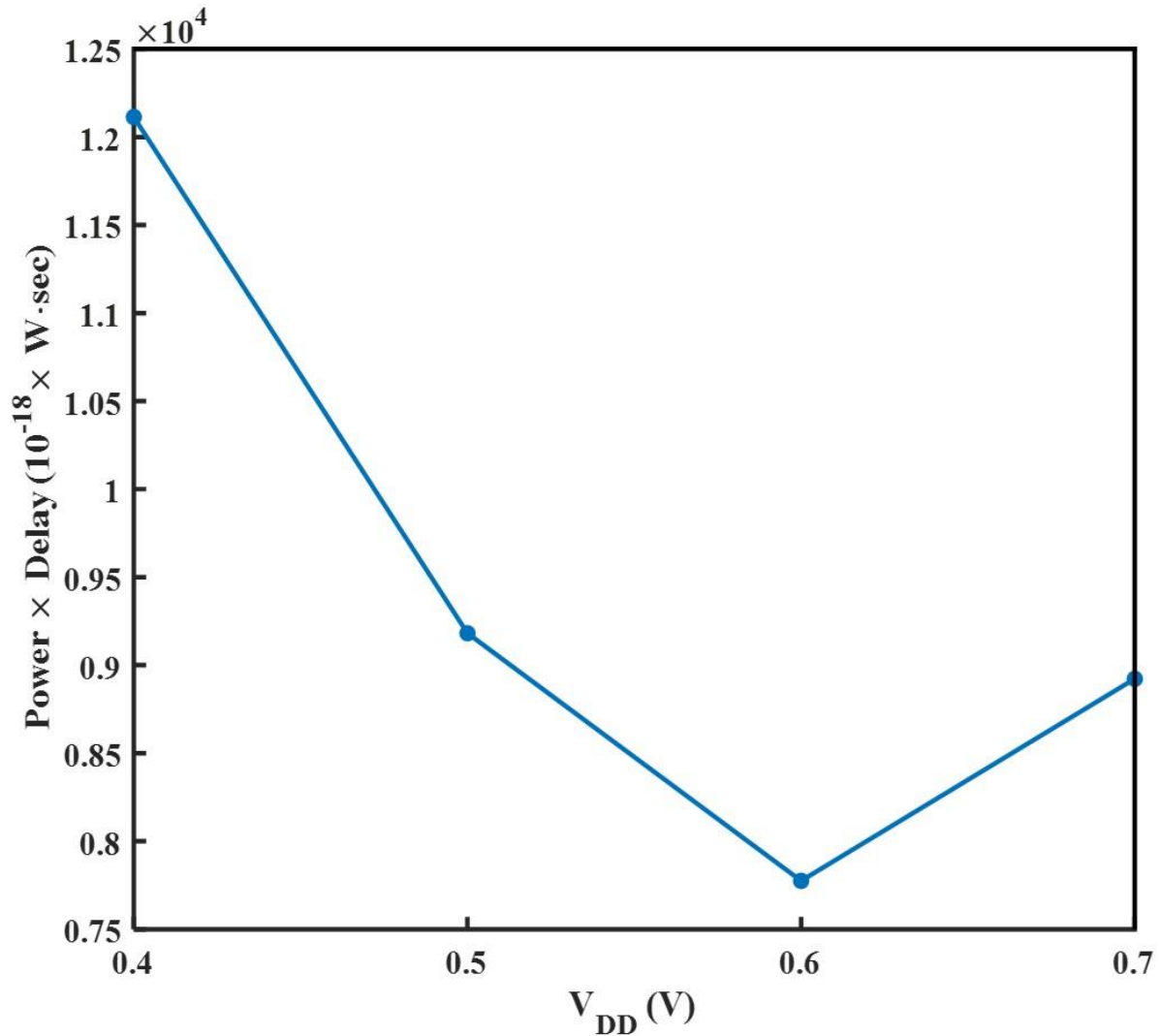
```

| Point | Incr | Path |
|--|--------|----------|
| ----- | | |
| clock clk (rise edge) | 0.00 | 0.00 |
| clock network delay (ideal) | 0.00 | 0.00 |
| input external delay | 200.00 | 200.00 r |
| A[0] (in) | 0.00 | 200.00 r |
| DP_OP_382J1_122_6874/U34/SN (FAX1_ASAP7_75t_R) | 28.57 | 228.57 r |
| U3I2/Y (INVxp67_ASAP7_75t_R) | 12.08 | 240.64 f |
| U94/Y (INVx1_ASAP7_75t_R) | 11.62 | 252.26 r |
| U313/Y (XNOR2xp5_ASAP7_75t_R) | 19.80 | 272.06 r |
| U314/Y (XOR2xp5_ASAP7_75t_R) | 26.91 | 298.97 r |
| U207/Y (MAJ1xp5_ASAP7_75t_R) | 16.66 | 315.63 f |
| U206/Y (MAJ1xp5_ASAP7_75t_R) | 21.41 | 337.04 r |
| U126/Y (HB1xp67_ASAP7_75t_R) | 26.12 | 363.17 r |
| U125/Y (XOR2xp5_ASAP7_75t_R) | 22.23 | 385.39 r |
| sum1_reg[0][2]/D (DFFHQNx1_ASAP7_75t_R) | 0.00 | 385.39 r |
| data arrival time | | 385.39 |
| | | |
| clock clk (rise edge) | 400.00 | 400.00 |
| clock network delay (ideal) | 0.00 | 400.00 |
| sum1_reg[0][2]/CLK (DFFHQNx1_ASAP7_75t_R) | 0.00 | 400.00 r |
| library setup time | -14.55 | 385.45 |
| data required time | | 385.45 |
| ----- | | |
| data required time | | 385.45 |
| data arrival time | | -385.39 |
| ----- | | |
| slack (MET) | | 0.06 |

圖十、合成後 time 結果圖

V_{DD} -Power \times Delay

最後，我們改變 V_{DD} 並比較其與 Power \times Delay 的關係，如下圖十一。

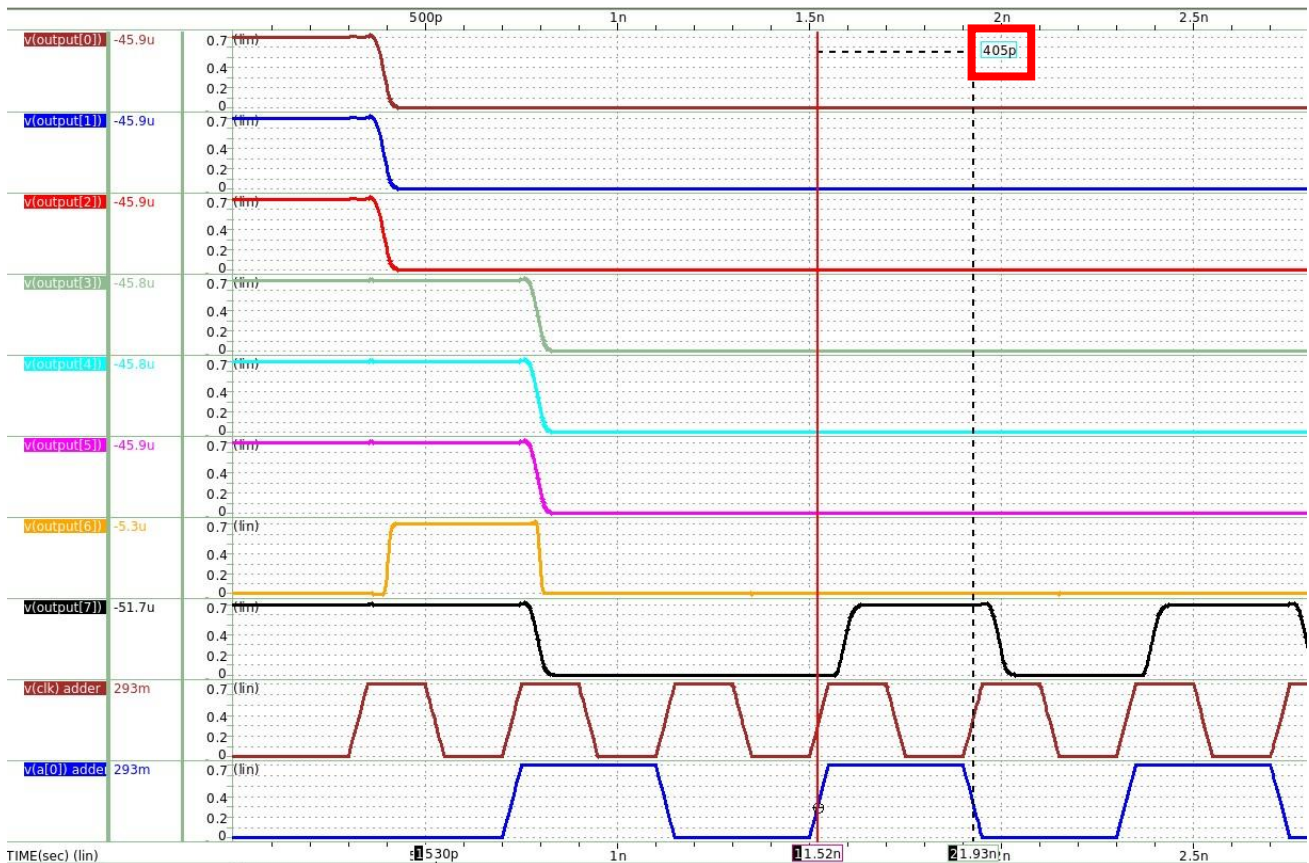


圖十一、 V_{DD} 對於 Power \times Delay 關係圖

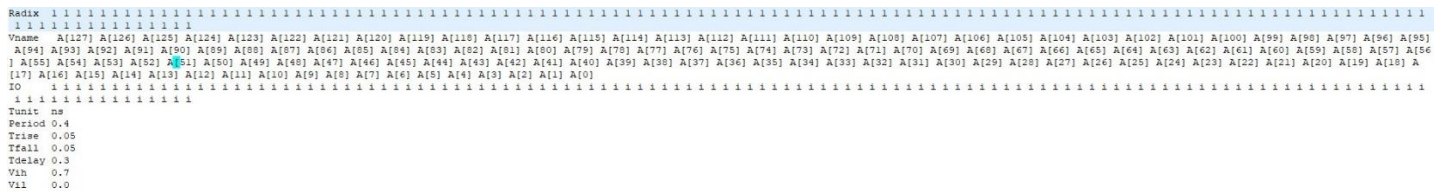
當我們將 V_{DD} 往下調時，Power 會隨著降低，但 Delay 則會隨之增加，而對於 Power \times Delay 而言，從圖中可以看到當 V_{DD} 為 0.6V 時，可以得到最小的 Power \times Delay。

HSPICE result

最後我們使用 HSPICE 檢驗波型，而前面一開始不穩定的波形是來自於未 reset 時的雜訊如下圖十二，而我們輸入的波形為 128 個 0 與 128 個 1 交互替換如下圖十三，最後可以發現一個週期的時間為 405ps 即操作頻率大約等於 2.5 GHz 大於助教要求的最低頻率 2 GHz。



圖十二、HSPICE 結果波形圖



圖十三、輸入波形圖(週期為 0.4ns)

Conclusion

比較了 add_number 為 2、4、8、16 的架構之後，得出了 add_number 為 2 的速度會最快，但是 Area 也最大，因此綜合考量後決定選用 add_number 為 16 的來進行優化。優化的過程中，我們選擇改變 V_{DD} 的大小來觀察 Power×Delay 的變化，過程中我們發現 power 會隨著 V_{DD} 變小而變小，而相反的 delay 則是隨著 V_{DD} 變小而變大，綜合評估後會發現在 $V_{DD} = 0.6V$ 時會達到最小值。