

2022 NYCU EE VLSI Lab Report

Lab04 Logic synthesis

Student ID: 109611070 Name: 郭家均 Date: 2022/11/25

I. Adder

i.31bits carry-ripple adder

- 1. the smallest clock period: 7.06**

- ## 2. area report (picture)

```
*****
Report : area
Design : DW_add
Version: R-2020.09
Date   : Fri Nov 25 11:59:55 2022
*****

Library(s) Used:

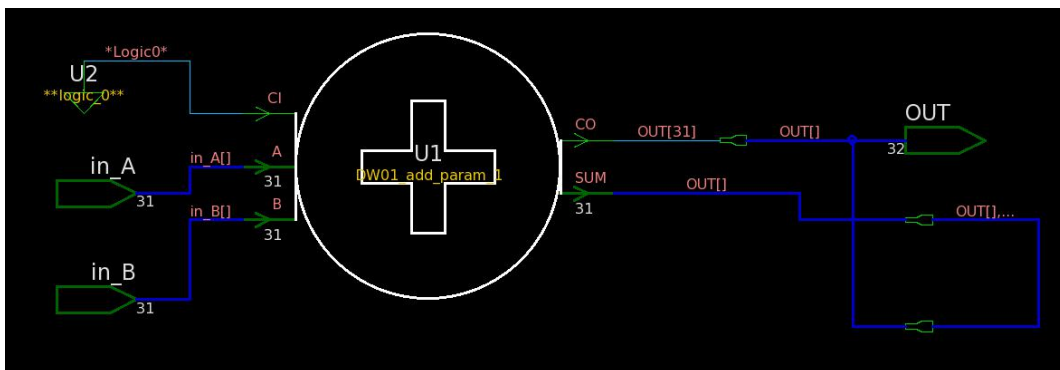
slow (File: /RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db)

Number of ports:                189
Number of nets:                 231
Number of cells:                48
Number of combinational cells:  46
Number of sequential cells:     0
Number of macros/black boxes:   0
Number of buf/inv:              3
Number of references:           1

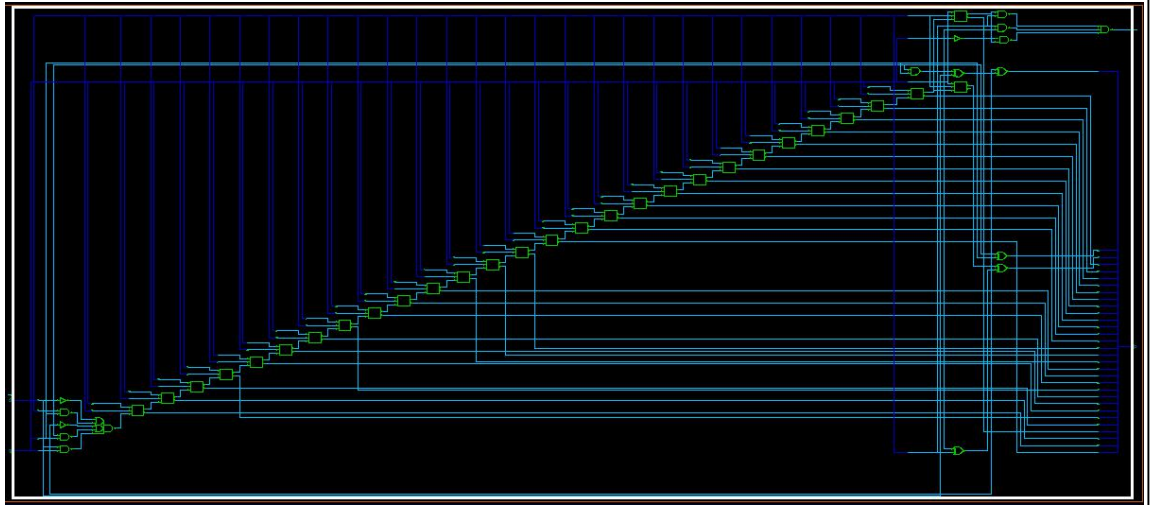
Combinational area:              3975.047931
Buf/Inv area:                   29.937601
Noncombinational area:          0.000000
Macro/Black Box area:           0.000000
Net Interconnect area:          undefined (No wire load specified)

Total cell area:                3975.047931
Total area:                     undefined
1
```

- ### 3. pre-synthesis schematic



4. post-synthesis schematic

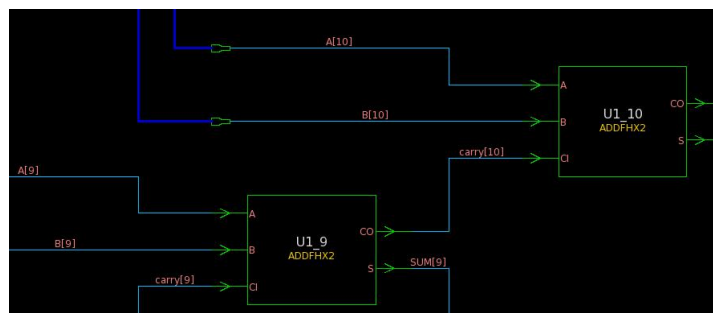


5. Please explain the difference between pre-synthesis and post-synthesis

Pre-synthesis 在電路設計過程能 verify 自己的電路在 functionality 的部分是正確的。可想成在 instance 中有一個叫 DW01_add 的 block，而我可以直接在 HDL code 中呼叫。Pre-synthesis 的結果跟 pattern 對比看結果是否正確。

Post-synthesis 中我們可以修改 adder block 的種類(RCA, CLA...)並且 02 模擬過程中，可以將每個 instance 中的架構(gate-level)實現。看到 schematic 就可以看到 adder 的完整架構已經由 stdcell 拼接好，且最後 synthesis 是考慮所有 stdcell 的 delay、area。

rpl adder 的架構很簡單。以一個 FA 為單位，最初的 $CI = 0$ 。每級 carry-out 推到下一級的 carry-in。以下放大看 schematic 的部分。可以看到 U1_9 的 carry[10]接到 U1_10 的 CI port。



6. timing report

Operating Conditions: slow Library: slow					
Wire Load Model Mode: top					
Startpoint: in_A[0] (input port)					
Endpoint: OUT[31] (output port)					
Path Group: default					
Path Type: max					
Point	Incr	Path			

input external delay	0.00	0.00 f	U1/U1_16/CO (ADDFHX2)	0.23	3.70 f
in_A[0] (in)	0.00	0.00 f	U1/U1_17/CO (ADDFHX2)	0.23	3.93 f
U1/A[0] (DW_add_DW01_add_0)	0.00	0.00 f	U1/U1_18/CO (ADDFHX2)	0.23	4.16 f
U1/U8/Y (NAND2X4)	0.06	0.06 r	U1/U1_19/CO (ADDFHX2)	0.23	4.39 f
U1/U5/Y (OAI221X2)	0.17	0.23 f	U1/U1_20/CO (ADDFHX2)	0.23	4.62 f
U1/U1_2/CO (ADDFHX2)	0.25	0.48 f	U1/U1_21/CO (ADDFHX2)	0.23	4.85 f
U1/U1_3/CO (ADDFHX2)	0.23	0.71 f	U1/U1_22/CO (ADDFHX2)	0.23	5.08 f
U1/U1_4/CO (ADDFHX2)	0.23	0.94 f	U1/U1_23/CO (ADDFHX2)	0.23	5.31 f
U1/U1_5/CO (ADDFHX2)	0.23	1.17 f	U1/U1_24/CO (ADDFHX2)	0.23	5.54 f
U1/U1_6/CO (ADDFHX2)	0.23	1.40 f	U1/U1_25/CO (ADDFHX2)	0.23	5.77 f
U1/U1_7/CO (ADDFHX2)	0.23	1.63 f	U1/U1_26/CO (ADDFHX2)	0.23	6.00 f
U1/U1_8/CO (ADDFHX2)	0.23	1.86 f	U1/U1_27/CO (ADDFHX2)	0.23	6.23 f
U1/U1_9/CO (ADDFHX2)	0.23	2.09 f	U1/U1_28/CO (ADDFHX2)	0.26	6.48 f
U1/U1_10/CO (ADDFHX2)	0.23	2.32 f	U1/U1_29/CO (ADDFHX2)	0.26	6.75 f
U1/U1_11/CO (ADDFHX2)	0.23	2.55 f	U1/U4/Y (NAND2BX2)	0.13	6.88 r
U1/U1_12/CO (ADDFHX2)	0.23	2.78 f	U1/U17/Y (NAND3X4)	0.17	7.05 f
U1/U1_13/CO (ADDFHX2)	0.23	3.01 f	U1/CO (DW_add_DW01_add_0)	0.00	7.05 f
U1/U1_14/CO (ADDFHX2)	0.23	3.24 f	OUT[31] (out)	0.00	7.05 f
U1/U1_15/CO (ADDFHX2)	0.23	3.47 f	data arrival time		7.05
max_delay				7.06	7.06
output external delay				0.00	7.06
data required time					7.06

data required time					7.06
data arrival time					-7.05

slack (MET)					0.01

7. Describe and explain the critical path

根據 timing report 可以看到 critical path 經過每一個 block 的 CO port。
且最後一級為 OUT[31]。從 pre-synthesis schematic 可以看出 OUT[31]
為最終的 COUT port。

可想而知，critical path 從第一級的 input 推到下一級 FA 的 CI port，
一路串接到最後的 carry-out。也可以得知最後一級 FA 的 output 中，
SUM[30]的 delay 比 COUT 還小 (NAND2 + NAND3 > XOR)。

ii.31 bits carry-lookahead adder

1. the smallest clock period: 1.44
2. area report

```
*****
Report : area
Design : DW_add
Version: R-2020.09
Date   : Fri Nov 25 12:12:33 2022
*****

Library(s) Used:

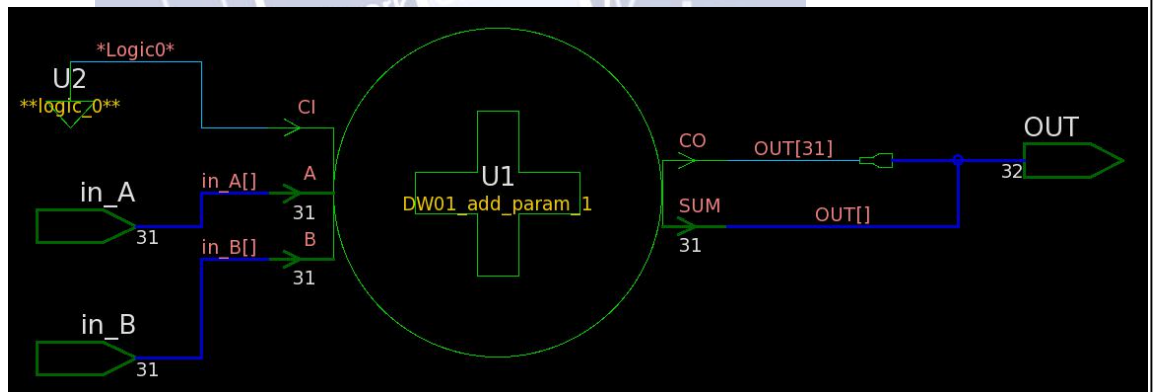
    slow (File: /RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db)

Number of ports:          189
Number of nets:           616
Number of cells:          461
Number of combinational cells: 459
Number of sequential cells:    0
Number of macros/black boxes:  0
Number of buf/inv:         134
Number of references:       1

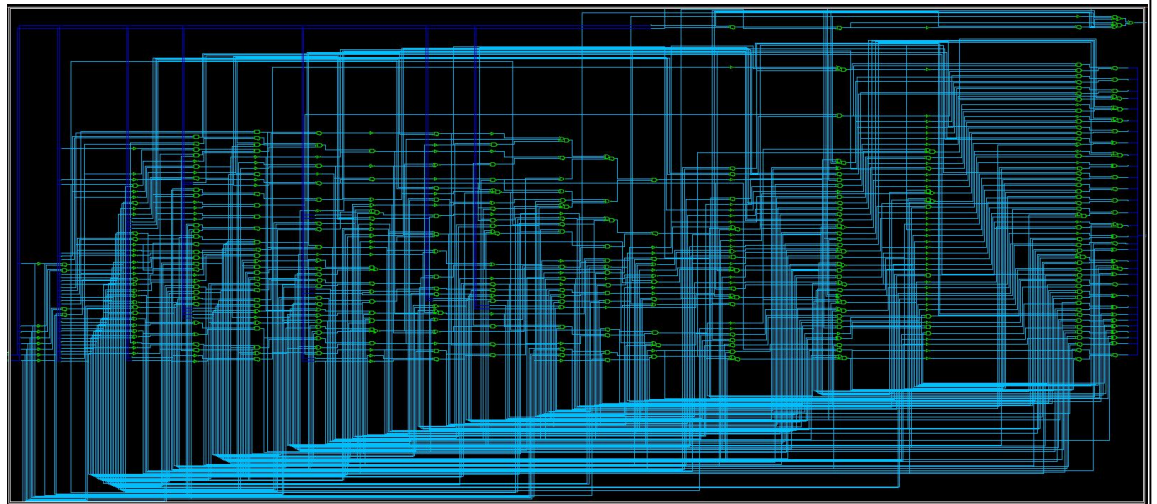
Combinational area:      8023.276881
Buf/Inv area:            1673.179229
Noncombinational area:   0.000000
Macro/Black Box area:    0.000000
Net Interconnect area:   undefined (No wire load specified)

Total cell area:         8023.276881
Total area:              undefined
1
```

3. pre-synthesis schematic (picture)



4. post-synthesis schematic (picture)



5. Please explain the difference between pre-synthesis and post-synthesis

Pre-synthesis、Post-synthesis 的差別和 i. 5.雷同。

cla 架構：

從 post-synthesis 電路可以看到整個 area 大了兩倍以上。為了加速先前 rpl 的 critical path，每一級的 SUM 不再單純由前一級的 carry-out 提供。每一級的 CI 可以由以下的判斷式提前得知

$$\begin{aligned} CI(i) &= (A_i B_i = 1) + \\ &\quad (A_i + B_i = 1) \& (carry(i-1) = 1) + \\ &\quad (A_i + B_i = 1) \& (A_{i-1} + B_{i-1} = 1) \& (carry(i-2) = 1) + \dots \end{aligned}$$

越後級的等式會越長，代表其 propagation 也會過大，不切實際，因此推估應該是使用 n bit 為一組的 CLA 架構 + tree 架構。如此 time complexity 為 $O(\log n)$ 比較符合和 rpl adder 之間時間的比較。更詳細的原因寫在 critical path 的 analysis 中。

只要其中一行的 logic = 1，CI 就為 1。如此加快 carry-in 的速度，因此最後可以發現 fastest clock 整個快了 4、5 倍左右。

6. timing report (picture)

Operating Conditions: slow Library: slow
Wire Load Model Mode: top

Startpoint: in_A[4] (input port)
Endpoint: OUT[21] (output port)
Path Group: default
Path Type: max

Point	Incr	Path

input external delay	0.00	0.00 r
in_A[4] (in)	0.00	0.00 r
U1/A[4] (DW_add_DW01_add_1)	0.00	0.00 r
U1/U380/Y (INVX4)	0.03	0.03 f
U1/U453/Y (NAND2X4)	0.08	0.11 r
U1/U154/Y (NAND2X4)	0.06	0.18 f
U1/U262/Y (INVX4)	0.06	0.24 r
U1/U261/Y (NAND2X4)	0.07	0.31 f
U1/U448/Y (NOR2X4)	0.10	0.41 r
U1/U447/Y (NAND2X4)	0.08	0.49 f
U1/U403/Y (CLKINVX4)	0.06	0.56 r
U1/U424/Y (NAND2X4)	0.06	0.62 f
U1/U237/Y (NAND3X2)	0.15	0.77 r
U1/U222/Y (NAND2X4)	0.08	0.85 f
U1/U423/Y (NAND2BX4)	0.10	0.95 r
U1/U218/Y (INVX4)	0.08	1.03 f
U1/U156/Y (OAI21X4)	0.14	1.18 r
U1/U124/Y (OAI2BB1X4)	0.26	1.44 r
U1/SUM[21] (DW_add_DW01_add_1)	0.00	1.44 r
OUT[21] (out)	0.00	1.44 r
data arrival time		1.44

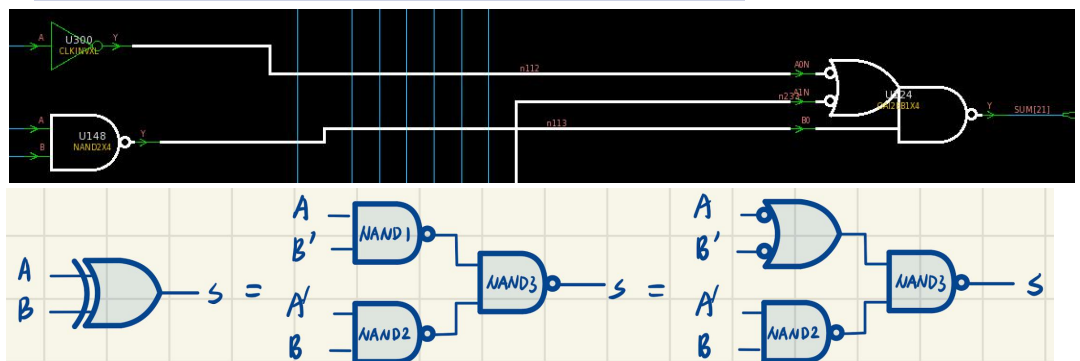
max_delay	1.44	1.44
output external delay	0.00	1.44
data required time		1.44

data required time		1.44
data arrival time		-1.44

slack (MET)		0.00

7. Describe and explain the critical path

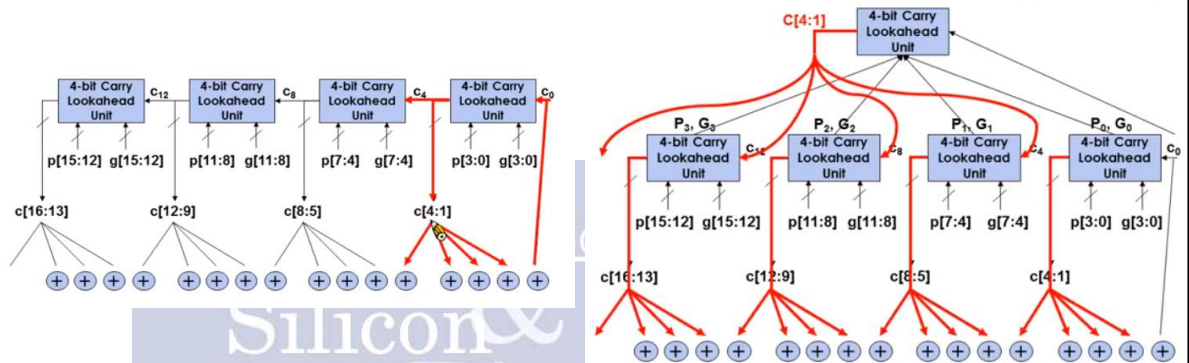
根據 timing report，發現整條 critical path 是從 input 端 (A[4])經過了 15 級的 gate 最後 output 為 SUM[21]。觀察 SUM[21]output 最後兩級簡化後可以發現就是一個簡單的 XOR 的邏輯如下圖所示。



又因為 critical path 的起點在 A[4]，可以推得主要造成 delay 的是 A[21] + B[21] + CI(21) 中 carry-in 的部分。由此可以討論幾個可能的 CLA 架構。一個是最基礎的每級 CI 都由一塊 combinational logic 運算。但後級的 CI 會因為式子過長所以 propagation 太長，不切實際。第二種架構為採 n-bit 唯一單元的 CLA，一次計算完 n-bit 的 CI，n+1

級的 CI 由 n COUT 提供。像 rpl adder 和 CLA 組合如下圖左。但此架構也不是很合理，照理來說若為此架構，critical path 會位於最後一級的 SUM，但結果不是如此，因此猜想第三種架構。

第三種架構是第二種的延伸，將 CI 的邏輯做成 tree 的架構，time complexity 為 $O(\log n)$ ，類似右下的圖。如此 critical path 的終點就可以位在 SUM[21]而不會有問題。且根據公式可以算 31 約為 2^5 。因此猜測 CLA 以兩個 bit 為單位，做了五層的 tree。



iii. Compare the pros and cons of architectures of carry ripple adder and carry-lookahead adder.

Carry ripple adder Pros :面積小，結構方便切 pipeline 去做加速。

Cons : Critical path 很長，不切 pipeline 會 latch 住 CLK

Carry look ahead Pros : 速度大大優於 RCA。有多種架構、活性高。

Cons : 面積相較 RCA 大許多，成本高。

II. Multiplier

i.31bits and-multiplier

1. the smallest clock period: 3.40

2. area report (picture)

```
*****
Report : area
Design : DW_mul
Version: R-2020.09
Date   : Fri Nov 25 12:33:45 2022
*****

Library(s) Used:

    slow (File: /RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db)

Number of ports:          129
Number of nets:           1164
Number of cells:          866
Number of combinational cells: 864
Number of sequential cells:    0
Number of macros/black boxes:  0
Number of buf/inv:         134
Number of references:       2

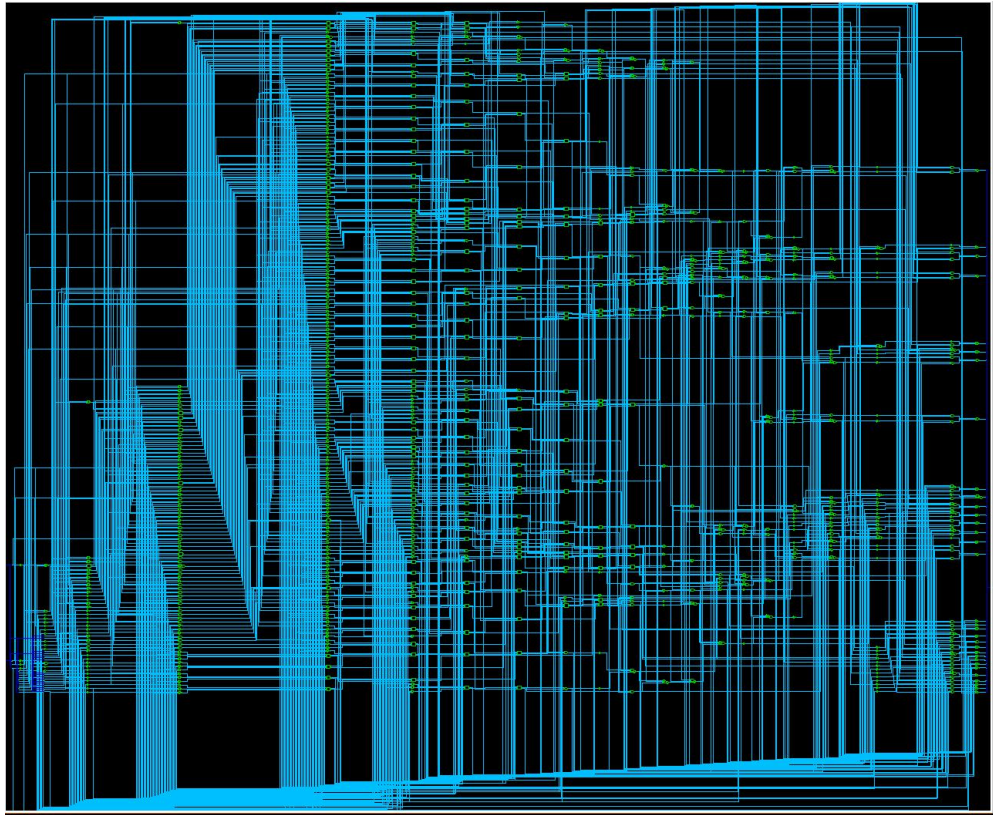
Combinational area:      32355.892459
Buf/Inv area:            1836.172819
Noncombinational area:   0.000000
Macro/Black Box area:    0.000000
Net Interconnect area:   undefined (No wire load specified)

Total cell area:         32355.892459
Total area:              undefined
```

3. pre-synthesis schematic (picture)



4. post-synthesis schematic (picture)

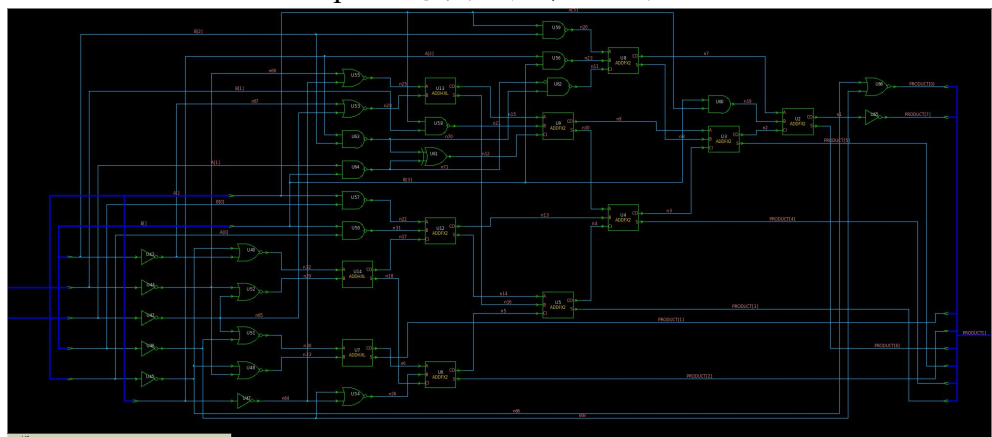


5. Please explain the difference between pre-synthesis and post-synthesis

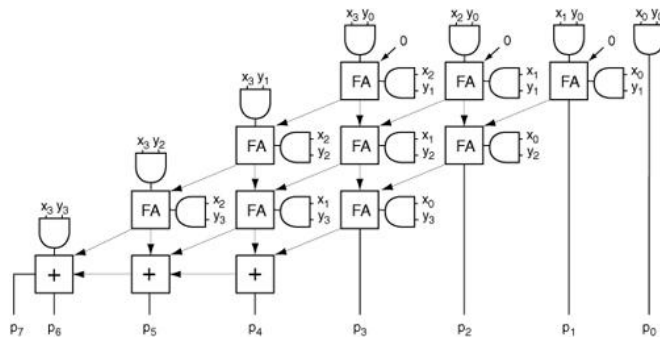
Pre-synthesis 看不出電路的架構，只是判斷 function 正確。

Post-synthesis 架構：

And multiplier 的架構上很直觀，和直式乘法的概念很像。我有叫出 4-bit*4-bit 的 And multiplier 觀察其架構如下圖：



可以發現每個對應的 A B input 都接上了 AND gate (INV + NOR)。算出兩個 bit 相乘的數值後，由一些 HA、FA 去計算 product sum 和每一級的進位，整個架構初步推估類似下圖(完整架構分析看 7.):



6. timing report (picture)

Operating Conditions: slow Library: slow
Wire Load Model Mode: top

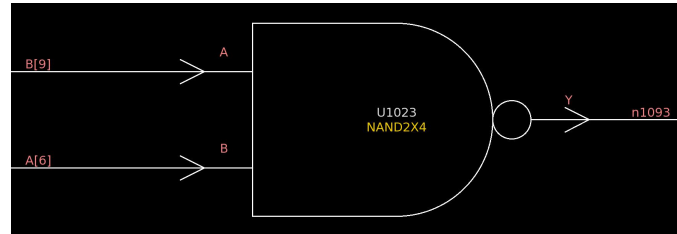
Startpoint: in_A[6] (input port)
Endpoint: OUT[23] (output port)
Path Group: default
Path Type: max

Point	Incr	Path
input external delay	0.00	0.00 f
in_A[6] (in)	0.00	0.00 f
U1/A[6] (DW_mul_DW02_mult_1)	0.00	0.00 f
U1/U1023/Y (NAND2X4)	0.06	0.06 f
U1/U871/Y (CLKINX4)	0.06	0.12 f
U1/U870/CO (ADDFHX2)	0.43	0.56 f
U1/U999/CO (ADDFHX4)	0.46	1.01 f
U1/U850/CO (ADDFHX2)	0.44	1.46 f
U1/U1194/CO (ADDFHX2)	0.49	1.94 f
U1/U363/CO (ADDFHX4)	0.38	2.32 f
U1/U1055/Y (NOR2X4)	0.16	2.48 f
U1/U152/Y (OAI21X4)	0.10	2.58 f
U1/U1094/Y (AOI21X4)	0.19	2.77 f
U1/U878/Y (INVXL)	0.12	2.89 f
U1/U877/Y (AOI21X2)	0.14	3.03 f
U1/U1268/Y (INX2)	0.08	3.11 f
U1/U1266/Y (XOR2X4)	0.29	3.40 f
U1/PRODUCT[23] (DW_mul_DW02_mult_1)	0.00	3.40 f
OUT[23] (out)	0.00	3.40 f
data arrival time		3.40
max_delay	3.40	3.40
output external delay	0.00	3.40
data required time		3.40
data required time		3.40
data arrival time		-3.40
slack (MET)		0.00

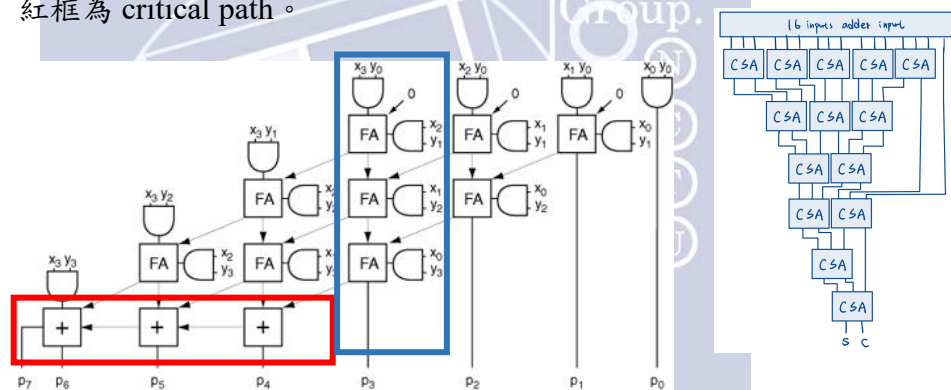
7. Describe and explain the critical path

根據 5. 分析的架構，可得知 critical path 必經的路段是中間 partial product 重疊最多的部分，可以看成一個 multiple input adder。

我們觀察 timing report critical path 的 input 可以看到是發生在 NAND U1023 且兩個輸入為 A[6]、B[9] 如下圖

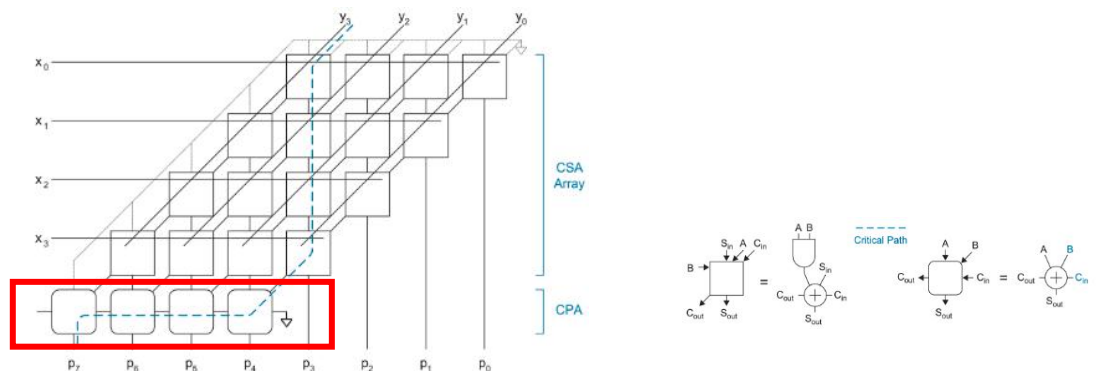


可以推得是最中間 16 input multiple adder 的其中一個 input，驗證先前的架構。不過終點可以看到在 product[23]，不是 product[31]，而且 critical path 經過的 HA、FA 數遠遠不及 16 個，因此可以推測最下方的 FA 應該不是單純的 RCA 架構，如下圖(用少 bit 架構代替表示)。在 16*16 的 multiplier 中，紅框部分為類似 16-bit RCA 的架構，藍+紅框為 critical path。



上圖藍框的部分可以改為 CSA 的作法，如上右圖。可以發現用 CSA 的作法，藍框的 propagation delay 最多過六級的 adder。

至於紅框部分，可以不採用 RCA 的方式，改成 CLA tree 的架構，如此將 16 bit adder 的 delay 再繼續下壓，且 critical path 的 output 出現不在 MSB 就合理了。整體架構如下圖(紅框部分改成 CLA)：



ii.31bits radix-4 booth multiplier

1. the smallest clock period: 3.26
2. area report (picture)

```
*****
Report : area
Design : DW_mul
Version: R-2020.09
Date   : Fri Nov 25 12:24:30 2022
*****

Library(s) Used:

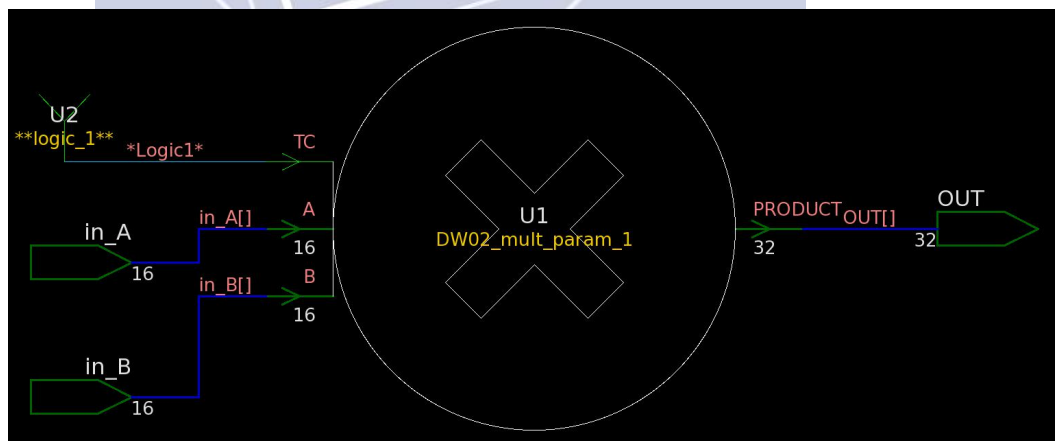
    slow (File: /RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db)

Number of ports:                129
Number of nets:                 1072
Number of cells:                883
Number of combinational cells:  881
Number of sequential cells:     0
Number of macros/black boxes:   0
Number of buf/inv:             175
Number of references:           3

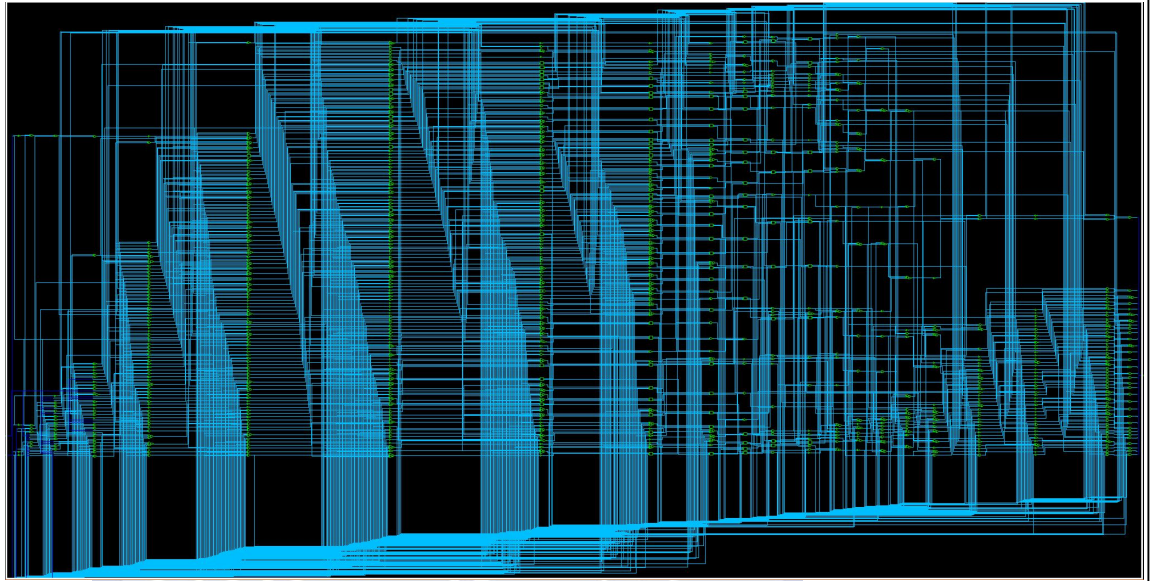
Combinational area:            26365.046267
Buf/Inv area:                  1939.291252
Noncombinational area:         0.000000
Macro/Black Box area:          0.000000
Net Interconnect area:         undefined (No wire load specified)

Total cell area:                26365.046267
Total area:                     undefined
```

3. pre-synthesis schematic (picture)



4. post-synthesis schematic (picture)



5. Please explain the difference between pre-synthesis and post-synthesis

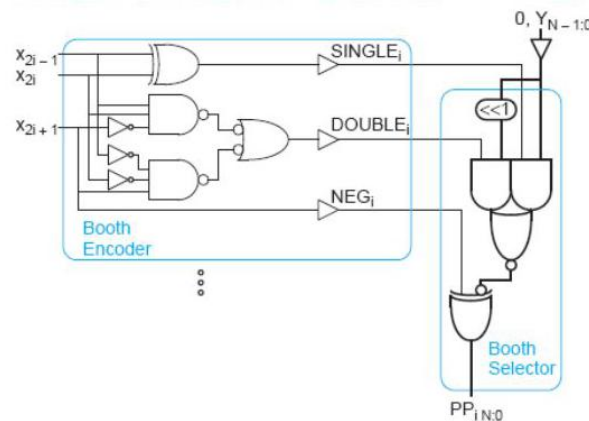
Pre-synthesis 看不出電路的架構，只是判斷 function 正確。

Post-synthesis 架構：

Radix-4 booth multiplier 為了解決 AND multiplier **partial products 過多**的問題而引入的。Radix-4 代表以 4 為基底，去替換掉原本以二進制表示的 multiplier。因此 multiplicand 的倍數可能有 0, 1, 2, 3。其中 **3 倍的運算會很複雜**，因此使用 **Booth encoding** 如下表取代 3：

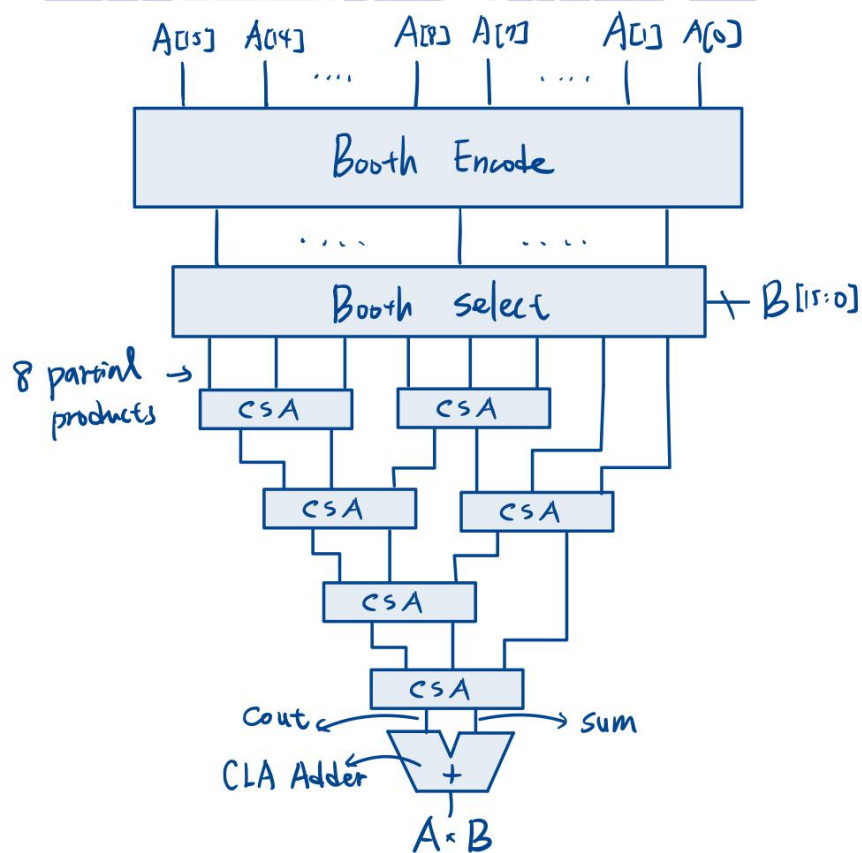
Inputs			Partial product	Booth Select		
X_{2i+1}	X_{2i}	X_{2i-1}	PP_i	S_i	D_i	N_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	2Y	0	1	0
1	0	0	-2Y	0	1	1
1	0	1	-Y	1	0	1
1	1	0	-Y	1	0	1
1	1	1	0	0	0	1

其中， X 為 multiplier。可以藉由 multiplier 得到 Booth select 的三個數值。其中 $S_i = 1$ 代表 PP_i 為 1 倍的 multiplicand； $D_i = 1$ 代表 PP_i 為 2 倍的 multiplicand； $N_i = 1$ 代表 PP_i 是負數。結果對應可以由上表看出。以下是 Booth encoding 和 Booth select 的電路：



因此這個 multiplier 的架構在經過 booth 後只需要加 8 組的 partial product (每個 partial product 是 32 bits)，相當於砍半。partial product 也可以用 AND-Multiplier 最後的 CSA + CLA 架構去加速。

以下為我推估該乘法器的大概架構：



6. timing report

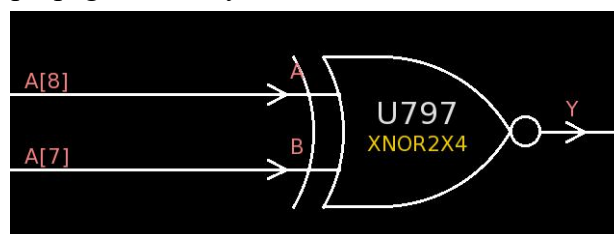
Operating Conditions: slow Library: slow
Wire Load Model Mode: top

Startpoint: in_A[8] (input port)
Endpoint: OUT[30] (output port)
Path Group: default
Path Type: max

Point	Incr	Path
input external delay	0.00	0.00 f
in_A[8] (in)	0.00	0.00 f
U1/A[8] (DW_mul_DW02_mult_1)	0.00	0.00 f
U1/U797/Y (XNOR2X4)	0.27	0.27 r
U1/U1082/Y (NAND2X4)	0.12	0.39 f
U1/U987/Y (OAI22XL)	0.26	0.65 r
U1/U1314/CO (ADDFX1)	0.64	1.29 r
U1/U1365/CO (ADDFX2)	0.56	1.85 r
U1/U919/Y (NOR2X2)	0.12	1.97 f
U1/U1000/Y (NOR2X4)	0.13	2.10 r
U1/U105/Y (NAND2X4)	0.09	2.19 f
U1/U1084/Y (INVX2)	0.06	2.26 r
U1/U1086/Y (NAND2X1)	0.06	2.31 f
U1/U1085/Y (OR2X1)	0.26	2.58 f
U1/U1073/Y (OAI21X1)	0.19	2.76 r
U1/U1108/Y (AOI21X1)	0.10	2.86 f
U1/U1555/Y (INVX2)	0.13	2.99 r
U1/U860/Y (CLKINVX4)	0.08	3.07 f
U1/U858/Y (NAND2X4)	0.08	3.15 r
U1/U859/Y (NAND2X4)	0.11	3.26 f
U1/PRODUCT[30] (DW_mul_DW02_mult_1)	0.00	3.26 f
OUT[30] (out)	0.00	3.26 f
data arrival time		3.26
max_delay	3.26	3.26
output external delay	0.00	3.26
data required time		3.26
data required time		3.26
data arrival time		-3.26
slack (MET)		0.00

7. Describe and explain the critical path

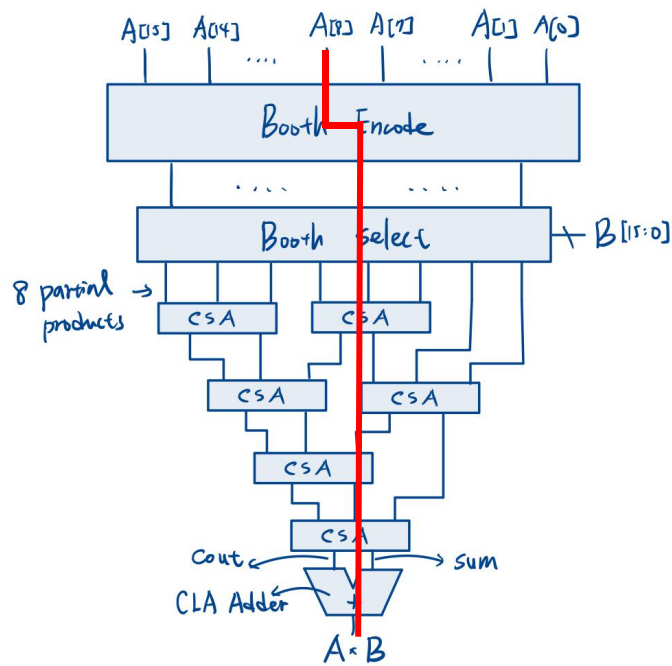
根據 timing report 可以看到 critical path 起點是下圖的 XNOR。輸入是 A[8] A[7]。此為 Booth encoder 邏輯的一部分 (用於生成 Booth select 的訊號)。因此可得知此架構將 A 訂為 multiplier，B 訂為 multiplicand。且 A[8]、A[7] 訊號剛好是 16*16 乘法的中間，因此最後面做 CSA 時會有多多 input 要處理 (更多 level 的 CSA)，propagation delay 會是最長的。



以下是 critical path 的流程：

Input (A[8] A[7])=> Booth Encode => Booth select

=> multiple CSA Level => CLA adder => output



iii. 31-bit radix-8 booth multiplier

1. the smallest clock period: 3.73
2. area report (picture)

```

*****
Report : area
Design : DW_mul
Version: R-2020.09
Date   : Fri Nov 25 12:44:24 2022
*****

Library(s) Used:

    slow (File: /RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db)

Number of ports:          129
Number of nets:           1181
Number of cells:          1018
Number of combinational cells: 1016
Number of sequential cells:    0
Number of macros/black boxes:  0
Number of buf/inv:         234
Number of references:      2

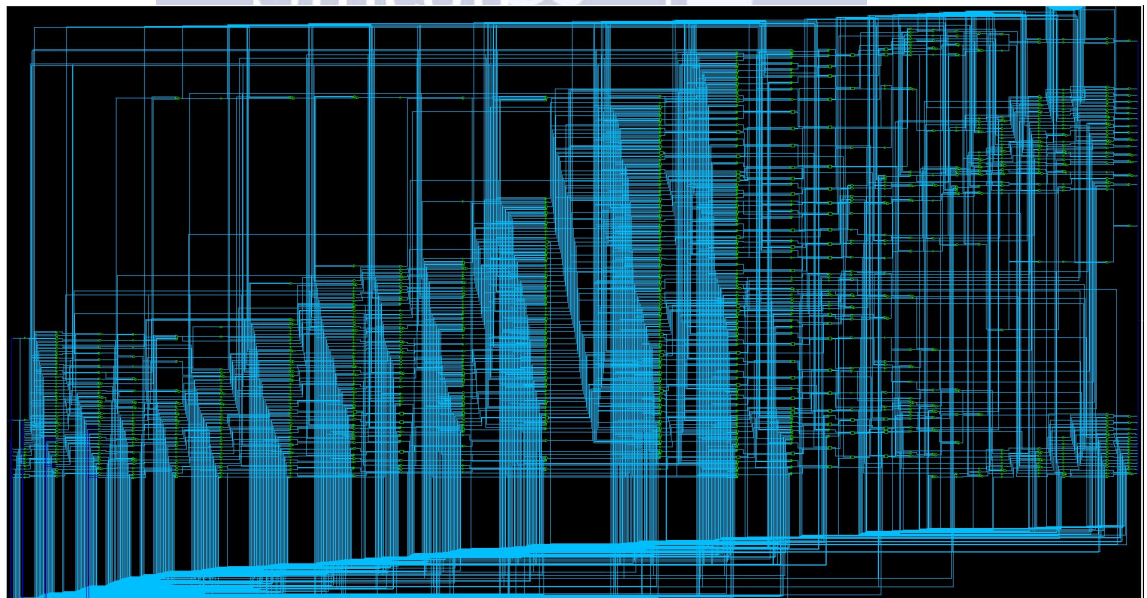
Combinational area:       30177.100871
Buf/Inv area:             2707.689674
Noncombinational area:    0.000000
Macro/Black Box area:     0.000000
Net Interconnect area:    undefined (No wire load specified)

Total cell area:          30177.100871
Total area:               undefined
  
```

3. pre-synthesis schematic (picture)



4. post-synthesis schematic (picture)



5. Please explain the difference between pre-synthesis and post-synthesis

Pre-synthesis 看不出電路的架構，只是判斷 function 正確。

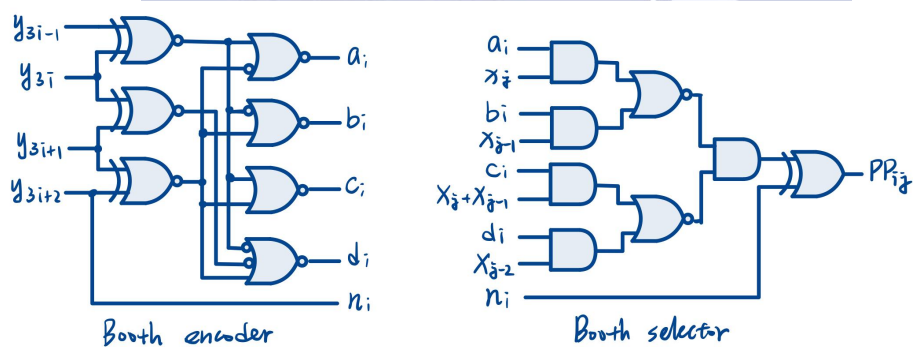
Post-synthesis 架構：

和 radix-4 booth multiplier 結構雷同，但 radix-8 的 encode 又更複雜。去替換掉原本以二進制表示的 multiplier。因此 multiplicand 的倍數可能有 0~7，其中 5, 6, 7 我們可以用 -3, -2, -1 去 encode。

根據下表的 encode 可以將 binary based 的 multiplier 轉成 octave based

i.e. $25_{10} = 0110010_2 \Rightarrow \text{radix-8 encode } (0110)(0010) \Rightarrow 1 \cdot 8^0 + 3 \cdot 8^1$

Multiplier Bits				Operation on Multiplicand
A	B	C	D	X
0	0	0	0	0X
0	0	0	1	+1X
0	0	1	0	+1X
0	0	1	1	+2X
0	1	0	0	+2X
0	1	0	1	+3X
0	1	1	0	+3X
0	1	1	1	+4X
1	0	0	0	-4X
1	0	0	1	-3X
1	0	1	0	-3X
1	0	1	1	-2X
1	1	0	0	-2X
1	1	0	1	-1X
1	1	1	0	-1X
1	1	1	1	0X



上方有 booth encoder 和 booth select 的邏輯，以下是 booth encoder 的 output 對應的意思。Selector output (PP_{ij}) 則為 partial product。

a_i	b_i	c_i	d_i	n_i
$ PP_i = 1$	$ PP_i = 2$	$ PP_i = 3$	$ PP_i = 4$	$PP_i < 0$

在此 radix-8 乘法器中，partial product 的總數會比 radix-4 來的更少。最終只有 $16/3$ 取上界 = 6 個 partial product 要做相加。因此後方所需的 CSA level 和 CLA 的面積部分會減少。整體結構上和 radix-4 booth multiplier 差不多。(更細部比較寫在後方討論)

6. Timing report

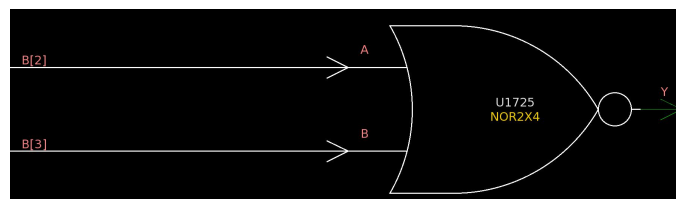
Operating Conditions: slow Library: slow
Wire Load Model Mode: top

Startpoint: in_B[3] (input port)
Endpoint: OUT[23] (output port)
Path Group: default
Path Type: max

Point	Incr	Path
input external delay	0.00	0.00 f
in_B[3] (in)	0.00	0.00 f
U1/U1725/Y (Dw_mul_Dw02_mult_1)	0.00	0.00 f
U1/U1725/Y (NOR2X4)	0.10	0.10 r
U1/U1545/Y (NOR2X4)	0.10	0.20 f
U1/U1937/Y (AOI21X4)	0.17	0.37 r
U1/U1305/Y (AOI2BB1X2)	0.24	0.62 r
U1/U1546/Y (XOR2X4)	0.27	0.89 r
U1/U1603/Y (MXI4X2)	0.34	1.23 f
U1/U1390/Y (MXI2X1)	0.20	1.43 f
U1/U1459/CO (ADDFHX2)	0.48	1.91 f
U1/U1198/S (ADDFHX2)	0.33	2.24 r
U1/U441/S (ADDFHX4)	0.39	2.63 r
U1/U1929/Y (NOR2X4)	0.08	2.71 f
U1/U1230/Y (AOI2BB1X2)	0.15	2.85 r
U1/U1928/Y (NAND2X2)	0.10	2.96 f
U1/U1933/Y (OAI21X4)	0.23	3.19 r
U1/U1277/Y (OAI2BB1X2)	0.22	3.42 r
U1/U1733/Y (CLKINX2)	0.08	3.50 f
U1/U1731/Y (NAND2X2)	0.11	3.61 r
U1/U1732/Y (NAND2X4)	0.12	3.73 f
U1/PRODUCT[23] (Dw_mul_Dw02_mult_1)	0.00	3.73 f
OUT[23] (out)	0.00	3.73 f
data arrival time		3.73
max_delay	3.73	3.73
output external delay	0.00	3.73
data required time		3.73
data required time		3.73
data arrival time		-3.73
slack (MET)		0.00

7. Describe and explain the critical path

根據 timing report 可以看到 critical path 起點是下圖的 NOR。輸入是 B[2] B[3]。此為 Booth encoder 邏輯的一部分 (用於生成 Booth select 的訊號)。因此可得知此架構將 B 訂為 multiplier，A 訂為 multiplicand。



以下是 critical path 的流程(大架構和 radix-4 雷同)：

Input (B[2] B[3])=> Booth Encode => Booth select

=> multiple CSA Level => CLA adder => output

iv. Compare the pros and cons of architectures of and-multiplier, radix-4 booth multiplier and radix-8 booth multiplier.

先將 and-multiplier 和 booth multiplier (radix-4, 8) 之間做比較：

And-multiplier Pros：面積小、硬體架構整齊

Cons: partial product 很多，速度會隨 bit 數上升大幅變慢。

Booth-multiplier Pros：partial product 少非常多，在後級加法的計算上較快。

Cons: 面積上因為多出 booth encoder、booth select 等電路，因此 cost 較大。

接著我們細究 radix-4 和 radix-8 在 16×16 的 case 有甚麼差別：

我們可以根據最後的結果看到就速度來說，radix-4 相對 and-multiplier 有稍微的提升，但 radix-8 的 case 不但沒有變快，反而還變慢了。在 radix-4 的部分，因為 Booth encode select 的 input 訊號可以直接由 multiplicand 本身或是 shift 過($\times 2$)來表示，因此硬體上部會有太大的 prop. delay。大概只多了五、六級的 gate delay，而因為 partial product 相較 and-mult. 而言是砍半的，因此綜合起來最終 critical path 有稍微縮減。

Radix-8 的 booth selector input 訊號有一個 $x_j + x_{j-1}$ 項，代表多了 Adder。且 booth encode selector 的電路上比 radix-4 複雜、龐大，且最終 partial product 只比 radix-4 少了兩組，在後方加法以 CSA 實現的前提下並沒有優勢。因此此次 Lab 做下來的結果 radix-4 的速度是最好的，radix-8 面積最大，效能也不出色。

在此值得注意的是，如果 bit 數不多的情況下，將 booth encoding radix 的 base number 增大並沒有好處。如果今天增加 radix-number，減少的 partial product delay 可以大於我 booth encoder selector 複雜化帶來的 prop. delay，那速度上才會再提升。不過考慮到 radix-num 上升會使面積增加不少，這兩者速度、面積之間的 tradeoff 也是選擇 multiplier 要考慮的點。

III. Questions

1. What is the difference between cell-based design and full-custom design? Please list out their advantages and disadvantages.

Full-custom design: Logic cells、circuit 到最後 layout 全部由工程師設計。

Pros: 設計上有較高的彈性，可以達到最 optimal 的面積和 performance

Cons: 設計流程繁瑣、成本高，大規模晶片整合 time-to-market 不切實際

Cell-based design: 使用已經預先設計好的 component (stdcell) 作為 building block。而 layout 的拉線部分是 customize (大規模由 EDA 做 routing)。

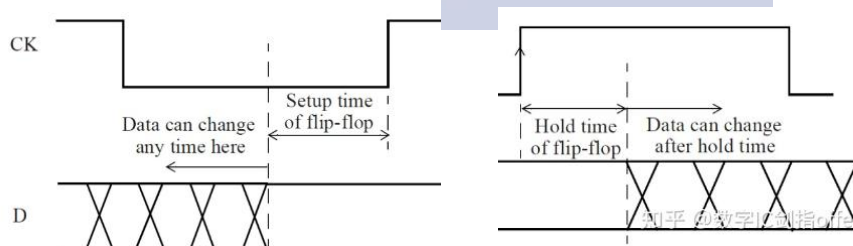
Pros: 設計速度快，較符合 time-to-market。由於 stdcell 通常高度固定，很方便由 EDA tool 去做 place & route。因此對於大規模的晶片設計會使用 cell-based design + EDA。

Cons: 由於 stdcell 已固定，因此無法達到最優化的 performance 和面積。整合度不會像 full-custom 那麼高。

2. What is set-up time and hold-up time? If there are violation, how to fix it?

在數位電路中，電路的速度和 CLK rate 有關。因此如何在極高的 CLK freq. 下不違反其 timing 是很重要的觀念。

今天 register (DFF) 在 clk 正緣處發前，D input 必須達到穩定，而這兩者之間的最短時間就是 set-up time，如左下圖所示：



hold-up time 則是正緣觸發後，D input 須維持穩定的最短時間。(右上圖)

簡而言之，DFF 的 input 要確保在 posedge 的前後，共維持穩定 set-up time + hold-up time，如此才可以確保 DFF 的 output 是我們想要的值。

產生 violation 即代表 D input 在 posedge 觸發前 set-up time 內，或是 posedge 觸發後 hold-up time 內改變了數值，如此 DFF output Q 是 unknown 狀態。

解決方法有：將 clk freq 降低、加快 DFF 前方 combination circuit 速度(即為降低數位電路的 critical path delay)、或是將前方 circuit 中間加入 pipeline design (加入更多級 DFF)

Summary:

本次 Lab 相較乍看比之前的 Lab 稍微輕鬆一些，但其實不簡單。要分析大量 bit 的加法器、乘法器不同架構，非常的耗費時間。要不斷上網爬閱資料，還要用 design vision 重複檢視。中間很多分析並不適百分之百確定是否和他生成的電路相同。不過我想大架構會是正確的方向，這次 Lab 學到了乘法器進階的架構。我想是為了之後的 project 做準備吧 QQ

