

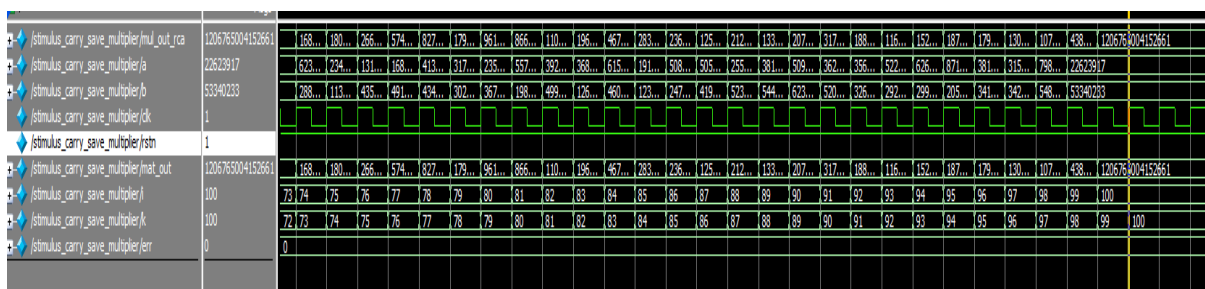
(1) Experimental Goal

Experiment의 main content는 26bit *26bit multiplier를 carry saving adder로 개량하여, 성능을 얻고, 이후 마지막 adder를 carry select adder를 apply하여 얼마나 회로가 개선되는지 확인하는 실험이다. 해당 실험의 Trade off를 고려하자.

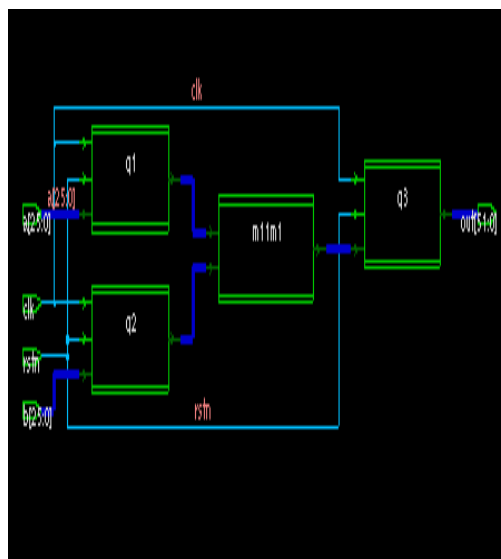
(2) Result

26bit-Carry save adder (gate level)

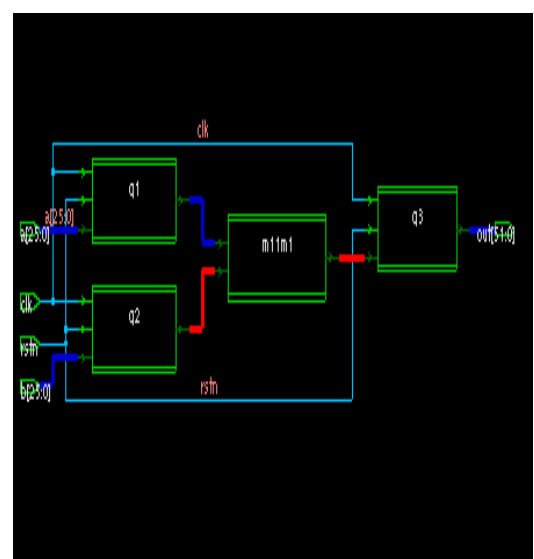
1. Timing diagram (for 1cycle)



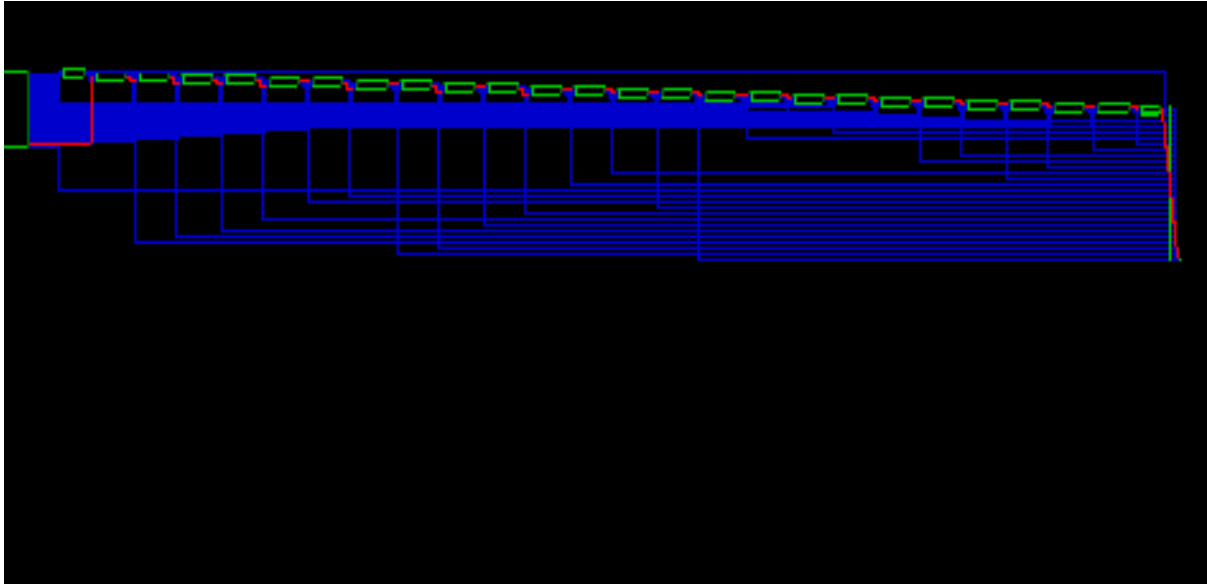
2. Schematic diagram



<before synthesis>



<after synthesis>



3. Synthesis result (Area , Power)

```
design_vision-xg-t> report_area

*****
Report : area
Design : CSM11
Version: Z-2007.03-SP4
Date   : Mon Apr 10 01:30:55 2023
*****

Library(s) Used:

    lec25dsc25_SS (File: /home/admin/lib/lec25/lec25dsc25_SS.db)

Number of ports:      106
Number of nets:       210
Number of cells:       4
Number of references:  4

Combinational area:   257433.409580
Noncombinational area: 17501.187119
Net Interconnect area: undefined (No wire load specified)

Total cell area:      274934.593750
Total area:           undefined
1
```

Operating Conditions: nom_pvt Library: lec25dsc25_SS
Wire Load Model Mode: top

Global Operating Voltage = 2.25
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000pf
Time Units = 1ns
Dynamic Power Units = 1mW (derived from V,C,T units)
Leakage Power Units = 1pW

Cell Internal Power = 6.7895 mW (70%)
Net Switching Power = 2.9776 mW (30%)

Total Dynamic Power = 9.7671 mW (100%)

Cell Leakage Power = 1.1309 mW

4. Synthesis result (Timing)

Operating Conditions: nom_pvt Library: lec25dscc25_SS
Wire Load Model Mode: top

Startpoint: q1/q_reg[25]
(rising edge-triggered flip-flop clocked by clk)
Endpoint: q3/q_reg[51]
(rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point	Incr	Path

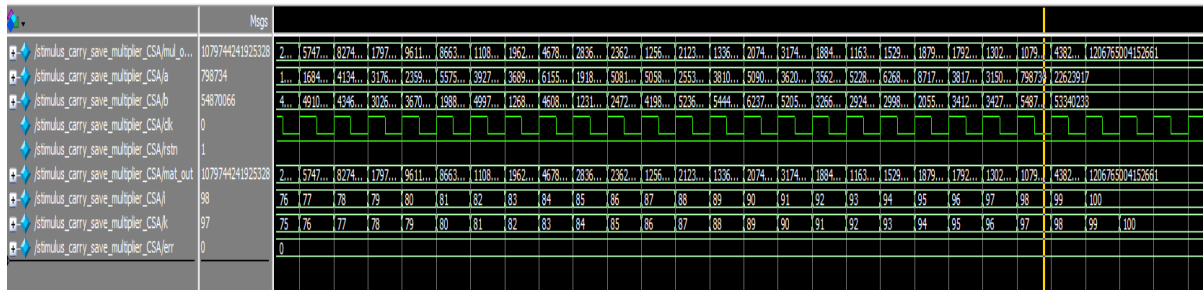
clock clk (rise edge)	1.00	1.00
clock network delay (ideal)	0.00	1.00
q1/q_reg[25]/CLK (dffcs2)	0.00	1.00 r
q1/q_reg[25]/QN (dffcs2)	0.23	1.23 r
q1/q_reg[25]/Q (dffcs2)	0.19	1.42 f
q1/q[25] (d_ff_26b_0)	0.00	1.42 f
m11m1/a[25] (CSMwithoutclk)	0.00	1.42 f
m11m1/mnna/a[25] (make_aandb1)	0.00	1.42 f
m11m1/mnna/aab3/a[25] (aandb_24)	0.00	1.42 f
m11m1/mnna/aab3/U4/Q (and2s1)	0.41	1.83 f
m11m1/mnna/aab3/out[25] (aandb_24)	0.00	1.83 f
m11m1/mnna/aandb2[25] (make_aandb1)	0.00	1.83 f
m11m1/ga21/a15[25] (Fulladd13withHA_gate_0)	0.00	1.83 f
m11m1/ga21/ha122/a (Halfadd_gate_26)	0.00	1.83 f
m11m1/gqa21451/gt91/c_in (fulladd_gate_7)	0.00	18.77 r
m11m1/gqa21451/gt91/U3/Q (nnd2s3)	0.10	18.87 f
m11m1/gqa21451/gt91/U2/Q (nnd2s3)	0.11	18.98 r
m11m1/gqa21451/gt91/c_out (fulladd_gate_7)	0.00	18.98 r
m11m1/gqa21451/gt102/c_in (fulladd_gate_6)	0.00	18.98 r
m11m1/gqa21451/gt102/U3/Q (nnd2s3)	0.10	19.08 f
m11m1/gqa21451/gt102/U2/Q (nnd2s3)	0.11	19.19 r
m11m1/gqa21451/gt102/c_out (fulladd_gate_6)	0.00	19.19 r
m11m1/gqa21451/gt26/c_in (fulladd_gate_5)	0.00	19.19 r
m11m1/gqa21451/gt26/U4/Q (nnd2s3)	0.10	19.29 f
m11m1/gqa21451/gt26/U3/Q (nnd2s3)	0.11	19.40 r
m11m1/gqa21451/gt26/c_out (fulladd_gate_5)	0.00	19.40 r
m11m1/gqa21451/gt216/c_in (fulladd_gate_4)	0.00	19.40 r
m11m1/gqa21451/gt216/U4/Q (nnd2s3)	0.10	19.50 f
m11m1/gqa21451/gt216/U3/Q (nnd2s3)	0.11	19.61 r
m11m1/gqa21451/gt216/c_out (fulladd_gate_4)	0.00	19.61 r
m11m1/gqa21451/gt36/c_in (fulladd_gate_3)	0.00	19.61 r
m11m1/gqa21451/gt36/U4/Q (nnd2s3)	0.10	19.71 f
m11m1/gqa21451/gt36/U3/Q (nnd2s3)	0.12	19.82 r
m11m1/gqa21451/gt36/c_out (fulladd_gate_3)	0.00	19.82 r
m11m1/gqa21451/gt46/c_in (fulladd_gate_2)	0.00	19.82 r
m11m1/gqa21451/gt46/U3/Q (nnd2s3)	0.10	19.92 f
m11m1/gqa21451/gt46/U2/Q (nnd2s3)	0.12	20.04 r
m11m1/gqa21451/gt46/c_out (fulladd_gate_2)	0.00	20.04 r
m11m1/gqa21451/gt56/c_in (fulladd_gate_1)	0.00	20.04 r
m11m1/gqa21451/gt56/U1/Q (nnd2s3)	0.10	20.14 f
m11m1/gqa21451/gt56/U2/Q (nnd2s3)	0.12	20.25 r
m11m1/gqa21451/gt56/c_out (fulladd_gate_1)	0.00	20.25 r
m11m1/gqa21451/gt11111/b (Halfadd_gate_1)	0.00	20.25 r
m11m1/gqa21451/gt11111/U1/Q (i1s3)	0.07	20.32 f
m11m1/gqa21451/gt11111/U5/Q (nnd2s2)	0.08	20.40 r
m11m1/gqa21451/gt11111/U2/Q (nnd2s1)	0.10	20.49 f
m11m1/gqa21451/gt11111/sum (Halfadd_gate_1)	0.00	20.49 f
m11m1/gqa21451/sum[25] (Last_term)	0.00	20.49 f
m11m1/out[51] (CSMwithoutclk)	0.00	20.49 f
q3/d[51] (d_ff_52b)	0.00	20.49 f
q3/q_reg[51]/CLRB (dffcs2)	0.00	20.49 f
data arrival time		20.49
clock clk (rise edge)	21.00	21.00
clock network delay (ideal)	0.00	21.00
q3/q_reg[51]/CLK (dffcs2)	0.00	21.00 r
library setup time	-0.50	20.50
data required time		20.50

data required time		20.50
data arrival time		-20.49

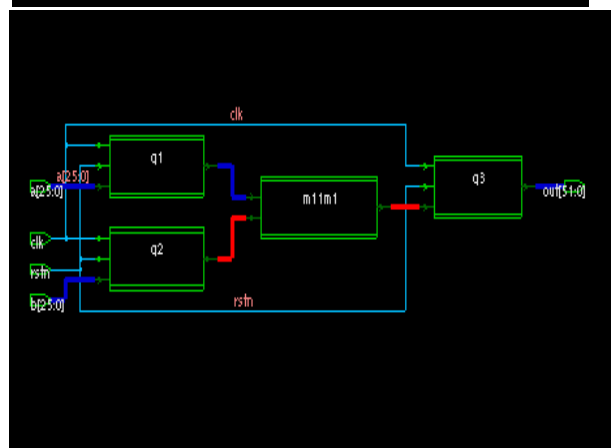
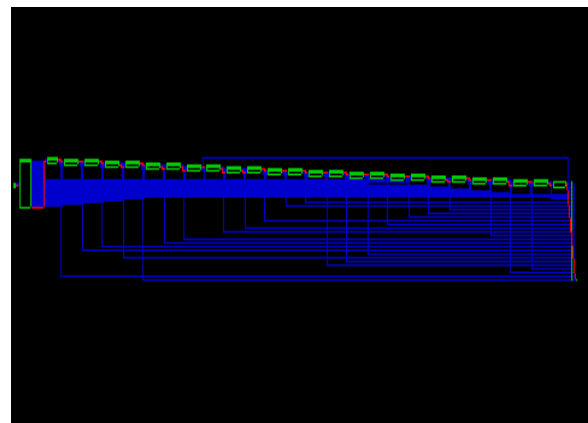
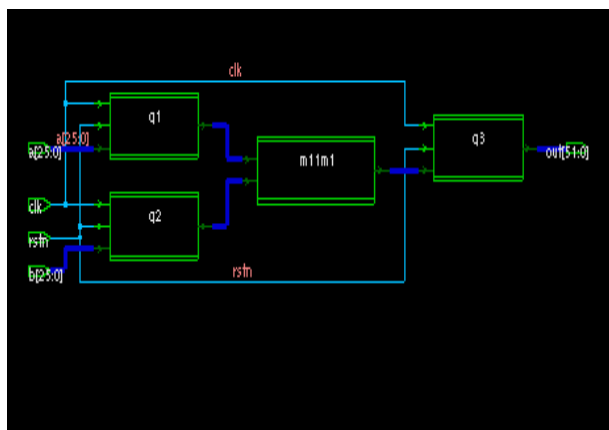
slack (MET)		0.00

<period 20.0 met>

Timing diagram (for 1cycle)



1. Schematic diagram



<before synthesis>

<after synthesis>

2. Synthesis result (Timing)

Operating Conditions: nom_pvt Library: lec25dscc25_SS
Wire Load Model Mode: top

Startpoint: q2/q_reg[1]
(rising edge-triggered flip-flop clocked by 123)
Endpoint: q3/q_reg[38]
(rising edge-triggered flip-flop clocked by 123)
Path Group: 123
Path Type: max

Point	Incr	Path

clock 123 (rise edge)	1.00	1.00
clock network delay (ideal)	0.00	1.00
q2/q_reg[1]/CLK (dffcs2)	0.00	1.00 r
q2/q_reg[1]/QN (dffcs2)	0.25	1.25 f
q2/q_reg[1]/Q (dffcs2)	0.12	1.37 r
q2/q[1] (d_ff_26b_1)	0.00	1.37 r
m11m1/b[1] (CSMwithoutclk11)	0.00	1.37 r
m11m1/mux/b[1] (mux_add5to1)	0.00	1.37 r

m11m1/crs/ad2/fa2/a[0] (fulladd5_gate_1)	0.00	14.69 r
m11m1/crs/ad2/fa2/gt1/a (fulladd_gate_31)	0.00	14.69 r
m11m1/crs/ad2/fa2/gt1/U3/Q (xnrs2s2)	0.30	14.99 r
m11m1/crs/ad2/fa2/gt1/U2/Q (oai21s3)	0.23	15.22 f
m11m1/crs/ad2/fa2/gt1/c_out (fulladd_gate_31)	0.00	15.22 f
m11m1/crs/ad2/fa2/gt2/c_in (fulladd_gate_30)	0.00	15.22 f
m11m1/crs/ad2/fa2/gt2/U5/Q (nnd2s2)	0.16	15.38 r
m11m1/crs/ad2/fa2/gt2/U6/Q (nnd2s2)	0.12	15.50 f
m11m1/crs/ad2/fa2/gt2/c_out (fulladd_gate_30)	0.00	15.50 f
m11m1/crs/ad2/fa2/gt3/c_in (fulladd_gate_29)	0.00	15.50 f
m11m1/crs/ad2/fa2/gt3/U5/Q (nnd2s2)	0.11	15.61 r
m11m1/crs/ad2/fa2/gt3/U6/Q (nnd2s2)	0.12	15.73 f
m11m1/crs/ad2/fa2/gt3/c_out (fulladd_gate_29)	0.00	15.73 f
m11m1/crs/ad2/fa2/gt4/c_in (fulladd_gate_28)	0.00	15.73 f
m11m1/crs/ad2/fa2/gt4/U6/Q (nnd2s2)	0.11	15.84 r
m11m1/crs/ad2/fa2/gt4/U7/Q (nnd2s2)	0.12	15.96 f
m11m1/crs/ad2/fa2/gt4/c_out (fulladd_gate_28)	0.00	15.96 f
m11m1/crs/ad2/fa2/gt5/c_in (fulladd_gate_27)	0.00	15.96 f
m11m1/crs/ad2/fa2/gt5/U1/Q (hnb1s1)	0.60	16.56 f
m11m1/crs/ad2/fa2/gt5/U2/Q (xor2s1)	0.44	17.00 f
m11m1/crs/ad2/fa2/gt5/sum (fulladd_gate_27)	0.00	17.00 f
m11m1/crs/ad2/fa2/sum[4] (fulladd5_gate_1)	0.00	17.00 f
m11m1/crs/ad2/sum2[4] (d5_adder)	0.00	17.00 f
m11m1/crs/ma2/s2[4] (mux_add5to1)	0.00	17.00 f
m11m1/crs/ma2/m5/D1 (m21_17)	0.00	17.00 f
m11m1/crs/ma2/m5/U1/Q (dsmxc31s2)	0.41	17.41 f
m11m1/crs/ma2/m5/Y (m21_17)	0.00	17.41 f
m11m1/crs/ma2/sum[4] (mux_add5to1)	0.00	17.41 f
m11m1/crs/sum[12] (carry_select_adder_26b_44567)	0.00	17.41 f
m11m1/out[38] (CSMwithoutclk11)	0.00	17.41 f
q3/d[38] (d_ff_52b)	0.00	17.41 f
q3/q_reg[38]/CLRB (dffcs1)	0.00	17.41 f
data arrival time		17.41

clock 123 (rise edge)	18.00	18.00
clock network delay (ideal)	0.00	18.00
q3/q_reg[38]/CLK (dffcs1)	0.00	18.00 r
library setup time	-0.51	17.49
data required time		17.49

data required time		17.49
data arrival time		-17.41

slack (MET)		0.08

<period 17.0-met>

Operating Conditions: nom_pvt Library: lec25dscc25_SS
Wire Load Model Mode: top

Global Operating Voltage = 2.25
Power-specific unit information :
 Voltage Units = 1V
 Capacitance Units = 1,000000pF
 Time Units = 1ns
 Dynamic Power Units = 1mW (derived from V,C,T units)
 Leakage Power Units = 1pW

Cell Internal Power	=	8.4854 mW	(69%)
Net Switching Power	=	3.7369 mW	(31%)

Total Dynamic Power	=	12.2223 mW	(100%)
Cell Leakage Power	=	383.3634 uW	

<power>

3. Synthesis result (Area)

```
design_vision-xg-t> report_area
```

```
*****  
Report : area  
Design : CSM111  
Version: Z-2007.03-SP4  
Date   : Mon Apr 10 02:11:19 2023  
*****
```

Library(s) Used:

lec25dscc25_SS (File: /home/admin/lib/lec25/lec25dscc25_SS.db)

Number of ports:	106
Number of nets:	210
Number of cells:	4
Number of references:	4
Combinational area:	270787.433968
Noncombinational area:	17517.776123
Net Interconnect area:	undefined (No wire load specified)
Total cell area:	288305.218750
Total area:	undefined
1	

(3) Discussion

먼저, multiplier의 자체적인 time complexity를 조사해보자. 해당 정성적으로, 기존의 Adder가 Linear하게 증가했으므로, 해당 adder를 base로 한 Multiplier 역시 Adder gate의 수가 Critical path기준으로 linear하게 증가하고, 이에 따라 Time complexity 역시 linear하게 증가할 것임을 알 수 있다.

(operating complexity는 quadratic하게 증가할 것으로 예상된다.)

이를 확인하기 위하여 n bit multiplier를 apply하여 확인해보자. 이는 Behavior level modeling으로 진행하였으며, 이에 따라 Time complexity가 바뀔 수 있음을 시사한다.

많은 bit를 apply하지 않고 2,4,8,16 bit까지 try하고 fitting을 진행하자. 순서대로 non clocked 2,4,8,16bit multiplier의 timing report이다.

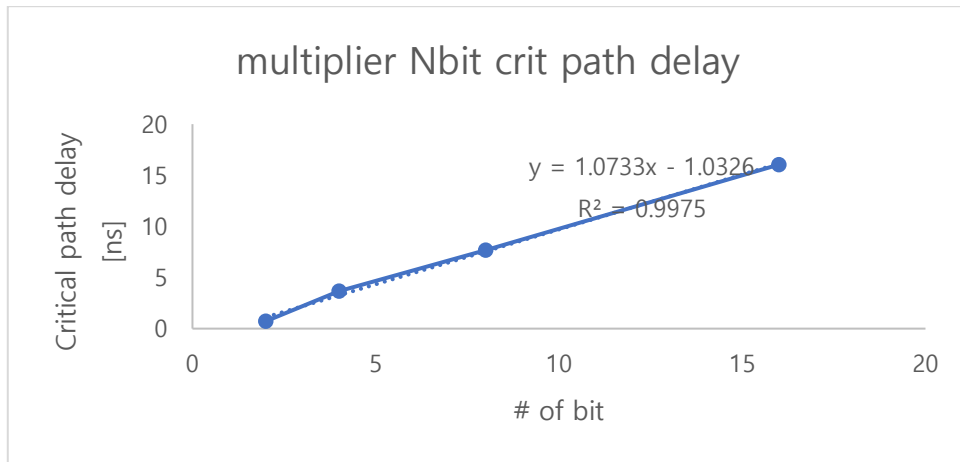
Startpoint: a[0] (input port) Endpoint: out[2] (output port) Path Group: (none) Path Type: max			Startpoint: a[1] (input port) Endpoint: out[6] (output port) Path Group: (none) Path Type: max		
Point	Incr	Path	Point	Incr	Path
input external delay	0,00	0,00 r	input external delay	0,00	0,00 r
a[0] (in)	0,00	0,00 r	a[1] (in)	0,00	0,00 r
U14/Q (<and2s1)	0,22	0,22 r	mult_15/a[1] (mul4_DW_mult_uns_1)	0,00	0,00 r
U10/Q (<nnd3s1)	0,17	0,39 f	mult_15/U45/Q (<ibis1)	0,13	0,13 f
U9/Q (<and3s1)	0,33	0,72 f	mult_15/U13/Q (<nor2s1)	0,20	0,32 r
out[2] (out)	0,00	0,72 f	mult_15/U13/OUTS (<haddis2)	0,59	0,92 f
data arrival time		0,72	mult_15/U6/OUTC (<faddis2)	0,50	1,42 f
(Path is unconstrained)			mult_15/U5/OUTC (<faddis2)	0,53	1,95 f
			mult_15/U4/OUTC (<faddis2)	0,47	2,42 f
			mult_15/U3/OUTC (<faddis2)	0,47	2,90 f
			mult_15/U2/OUTS (<faddis2)	0,76	3,65 r
			mult_15/product[6] (mul4_DW_mult_uns_1)	0,00	3,65 r
			out[6] (out)	0,00	3,65 r
			data arrival time		3,65
			(Path is unconstrained)		

Report : timing -path Full -delay max -max_paths 1 Design : mul8 Version: Z-2007.03-SP4 Date : Mon Apr 10 02:18:43 2023			Operating Conditions: nom_pvt Library: lec25dscc25_SS Wire Load Model: top		
Startpoint: a[3] (input port) Endpoint: out[14] (output port) Path Group: (none) Path Type: max					
Point	Incr	Path	Point	Incr	Path
input external delay	0,00	0,00 f	input external delay	0,00	0,00 f
a[3] (in)	0,00	0,00 f	mult_24/a[3] (mul8_DW_mult_uns_0)	0,00	0,00 f
mult_24/U149/Q (<ibis1)	0,21	0,21 r	mult_24/U149/Q (<ibis1)	0,21	0,21 r
mult_24/U96/Q (<nor2s1)	0,27	0,48 f	mult_24/U96/Q (<nor2s1)	0,27	0,48 f
mult_24/U54/OUTS (<faddis2)	0,81	1,29 r	mult_24/U54/OUTS (<faddis2)	0,81	1,29 r
mult_24/U52/OUTS (<faddis2)	0,80	2,09 f	mult_24/U52/OUTS (<faddis2)	0,80	2,09 f
mult_24/U12/OUTC (<faddis2)	0,54	2,63 f	mult_24/U12/OUTC (<faddis2)	0,54	2,63 f
mult_24/U11/OUTC (<faddis2)	0,47	3,10 f	mult_24/U11/OUTC (<faddis2)	0,47	3,10 f
mult_24/U10/OUTC (<faddis2)	0,47	3,58 f	mult_24/U10/OUTC (<faddis2)	0,47	3,58 f
mult_24/U9/OUTC (<faddis2)	0,47	4,06 f	mult_24/U9/OUTC (<faddis2)	0,47	4,06 f
mult_24/U8/OUTC (<faddis2)	0,47	4,53 f	mult_24/U8/OUTC (<faddis2)	0,47	4,53 f
mult_24/U7/OUTC (<faddis2)	0,47	5,00 f	mult_24/U7/OUTC (<faddis2)	0,47	5,00 f
mult_24/U6/OUTC (<faddis2)	0,47	5,48 f	mult_24/U6/OUTC (<faddis2)	0,47	5,48 f
mult_24/U5/OUTC (<faddis2)	0,47	5,95 f	mult_24/U5/OUTC (<faddis2)	0,47	5,95 f
mult_24/U4/OUTC (<faddis2)	0,47	6,42 f	mult_24/U4/OUTC (<faddis2)	0,47	6,42 f
mult_24/U3/OUTC (<faddis2)	0,47	6,90 f	mult_24/U3/OUTC (<faddis2)	0,47	6,90 f
mult_24/U2/OUTS (<faddis2)	0,76	7,66 r	mult_24/U2/OUTS (<faddis2)	0,76	7,66 r
mult_24/product[14] (mul8_DW_mult_uns_0)	0,00	7,66 r	mult_24/product[14] (mul8_DW_mult_uns_0)	0,00	7,66 r
out[14] (out)	0,00	7,66 r	out[14] (out)	0,00	7,66 r
data arrival time		7,66	data arrival time		7,66
(Path is unconstrained)					

Startpoint: b[0] (input port)
 Endpoint: out[31] (output port)
 Path Group: (none)
 Path Type: max

Point	Incr	Path
input_external delay	0.00	0.00 r
b[0] (in)	0.00	0.00 r
mult_34/b[0] (mul16_DW_mult_uns_1)	0.00	0.00 r
mult_34/U691/Q (nbis3)	0.45	0.45 r
mult_34/U121/Q (hiis1)	0.71	1.16 f
mult_34/U597/Q (nnd2s2)	0.23	1.39 r
mult_34/U505/Q (oai22s2)	0.19	1.58 f
mult_34/U100/OUTC (hadd1s2)	0.42	2.00 f
mult_34/U99/OUTC (fadd1s2)	0.45	2.45 f
mult_34/U98/OUTC (fadd1s2)	0.47	2.92 f
mult_34/U97/OUTC (fadd1s2)	0.47	3.40 f
mult_34/U96/OUTC (fadd1s2)	0.47	3.87 f
mult_34/U95/OUTC (fadd1s2)	0.47	4.35 f
mult_34/U94/OUTC (fadd1s2)	0.47	4.82 f
mult_34/U93/OUTC (fadd1s2)	0.47	5.30 f
mult_34/U92/OUTC (fadd1s2)	0.47	5.77 f
mult_34/U91/OUTC (fadd1s2)	0.47	6.25 f
mult_34/U90/OUTC (fadd1s2)	0.47	6.72 f
mult_34/U89/OUTC (fadd1s2)	0.47	7.19 f
mult_34/U88/OUTC (fadd1s2)	0.47	7.67 f
mult_34/U87/OUTC (fadd1s2)	0.47	8.14 f
mult_34/U86/OUTC (fadd1s2)	0.47	8.62 f
mult_34/U85/OUTC (fadd1s2)	0.47	9.09 f
mult_34/U84/OUTC (fadd1s2)	0.47	9.57 f
mult_34/U83/OUTC (fadd1s2)	0.47	10.04 f
mult_34/U82/OUTC (fadd1s2)	0.47	10.51 f
mult_34/U81/OUTC (fadd1s2)	0.47	10.99 f
mult_34/U80/OUTC (fadd1s2)	0.47	11.46 f
mult_34/U79/OUTC (fadd1s2)	0.47	11.94 f
mult_34/U78/OUTC (fadd1s2)	0.47	12.41 f
mult_34/U77/OUTC (fadd1s2)	0.47	12.89 f
mult_34/U76/OUTC (fadd1s2)	0.47	13.36 f
mult_34/U75/OUTC (fadd1s2)	0.47	13.83 f
mult_34/U74/OUTC (fadd1s2)	0.47	14.31 f
mult_34/U73/OUTC (fadd1s2)	0.47	14.78 f
mult_34/U72/OUTC (fadd1s2)	0.47	15.26 f
mult_34/U71/OUTC (fadd1s2)	0.44	15.70 f
mult_34/U69/Q (xor2s1)	0.34	16.04 r
mult_34/product[31] (mul16_DW_mult_uns_1)	0.00	16.04 r
out[31] (out)	0.00	16.04 r
data arrival time		16.04

(Path is unconstrained)



위와 같이 Linear하게 떨어지는 multiplier의 time complexity를 알 수 있었다.

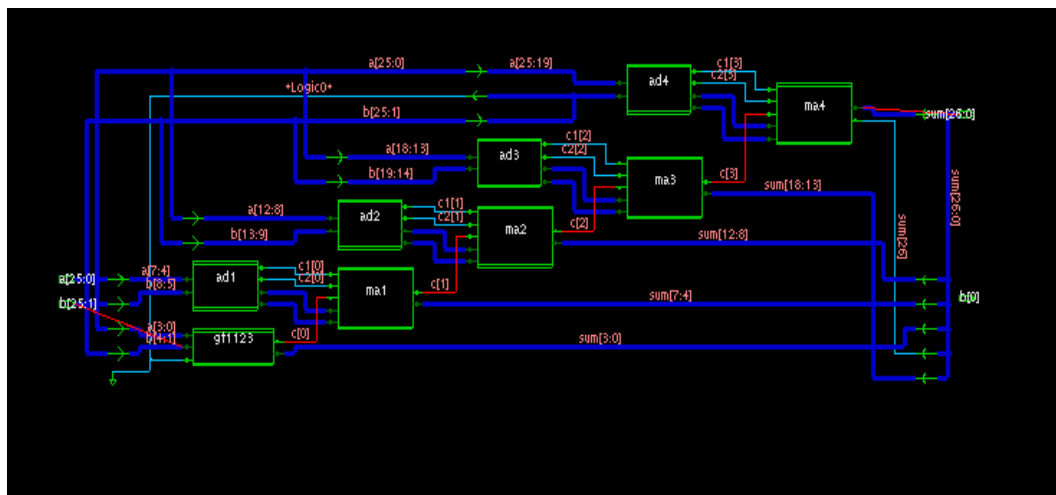
또한, Critical path역시 예상한대로 input a의 LSB part에서 output의 MSB (-1) port로
 의 path가 Critical path임을 알 수 있었다. 이제 본격적으로 실험 내용을 시작해보자.

기존에 우리가 실험했던 결과에 따르면, Clock을 Block형태로 apply하는 것과 Mux를
 하나로 묶어 처리하는 방법이 가장 효율적임을 알 수 있었다. 이를 바탕으로 코드를
 짰 후, 실험 Data를 확인하자.

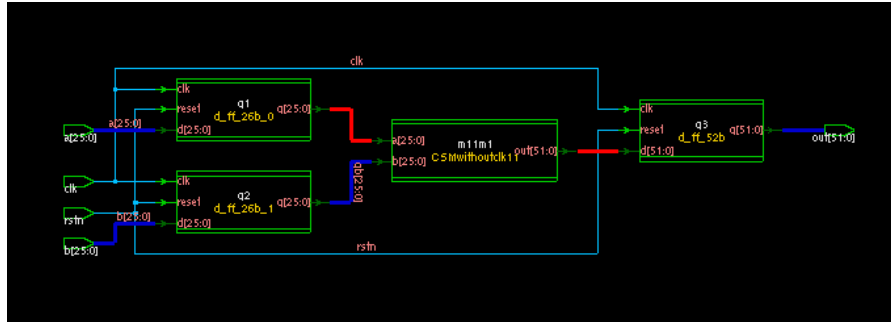
	RCA multiplier	CSA multiplier	비교 [%]
Min period [ns]	20	17	+15
Clock speed [Hz]	$5.00 \cdot 10^7$	$5.88 \cdot 10^7$	+15
Data arrival time [ns]	20.49	17.41	-15.1
Area [μm^2]	274935	288305	+4.8
Total power [mW]	9.77	12.2	+24.8

위와 같은 결과를 알 수 있다. 이를 통하여 RCA 대신 Last bit를 처리하는 과정에서 CSA가 performance는 15% 만큼 향상되지만, Area와 Total power에서는 더 떨어지는 모습을 보여주었다. 이를 통해 RCA와 CSA를 apply하는 것이 여러가지 변수들 사이에서 Tradeoff를 고민해야 된다는 사실을 알 수 있었다.

이제 Critical path를 고려하자. 먼저 CSA를 apply한 path에서 마지막 처리 부분인 26bit CSA를 Zoom해보면 아래와 같다.

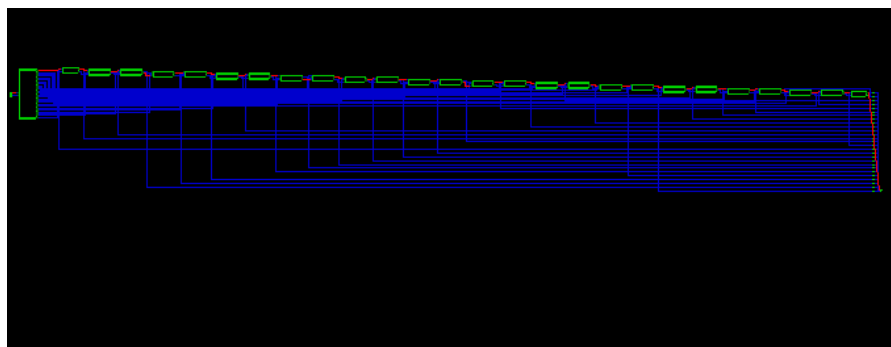
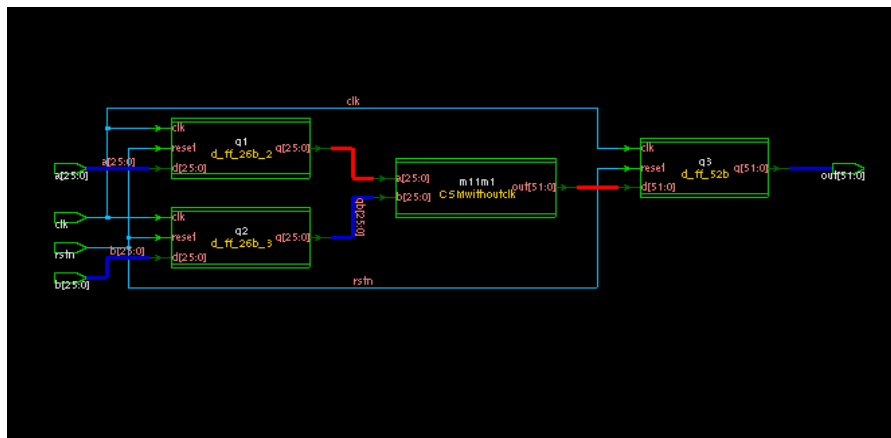


기존에 우리는 Adder에 대비하여 Mux의 Anti-clk-friendly한 부분을 알고 있었다. 이에 따라 마지막 mux(ma4)로 들어가는 모든 input이 Critical path가 될 수 있음을 알 수 있었고 이를 이번 실습에서도 확인할 수 있었다. 이제 범위를 넓혀 전체적인 구도를 보자.



위와 같은 Critical path를 확인할 수 있었다. Input A reg, CSM module, output Reg로 이어지는 경로를 찾을 수 있었다. 위의 사진은 위에 첨부했으며 처음 들어간 q_reg1[1] 부터 q_reg[38] 로 이어지는 path임을 알 수 있었다.

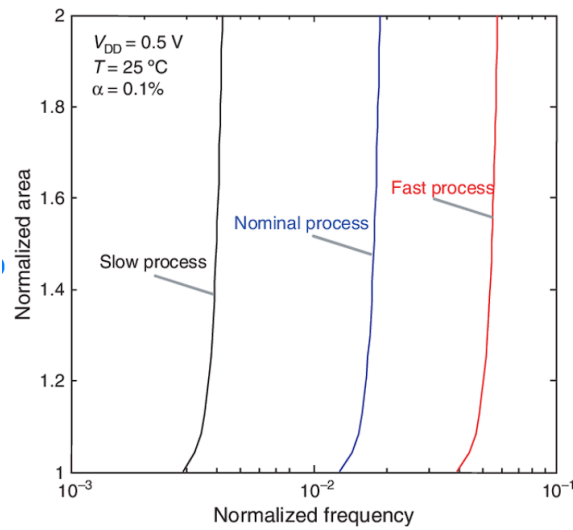
다음은 RCA가 apply된 Multiplier를 살펴보자.



위와 같이 순차적으로 Critical path가 이어짐을 알 수 있었다. 마지막으로, 26bit Adder를 살펴봐도 그대로 순차적으로 이어짐을 알 수 있었다. 이 때의 Critical path는 Q[25]인 LSB에서 시작하여 output의 MSB로 가는 path를 볼 수 있었다.

마지막으로 Area에 대하여 살펴보자. Area는 Performance를 improve하기 위한 여러가지 장치를 추가할수록 더욱 큰 면적을 가지게 되는 것을 알 수 있었다. 특히나 Clock을 push하였을 때, 면적이 커지는 것도 볼 수 있었는데, 이는 단순히 Gate의 수가 아닌

Clock의 Push도 Area에 영향을 주는 것임을 알 수 있었다. 또한 이는



<Research gate -Frequency vs Area>

위의 Graph에서도 실험적으로 검증된 사실이다. 이를 우리 실험에서도 검증할 수 있었다

마지막으로 회로의 본질적으로 중요한 점인 Verification을 보자. Test bench는 내가 verification하는 design의 input을 만들어 주어야 하며, Reference model과 동일하게 operation하는지 output을 check하는 부분으로 이루어져 있다. Operating 시나리오를 기반으로 test case의 constraint를 만들고 그 안에서 random한 input을 apply한다. 이를 마지막 score board에서 Wave form으로 check하며, RTL algorithm이 동일하게 구현되었는지 error를 check한다 IP의 모든 부분이 connected 되었는지 check하기 위하여 coverage를 check하는 것도 필요하다. 이제 Verification 결과를 간단하게 확인하자. 우리는 Matlab을 통하여 input과 해당 value 간의 덧셈을 이용한 output 결과를 만들었다. 즉, 앞서 언급한 operating 시나리오를 위한 constraint를 matlab을 통하여 txt로 만든 것이다. 이를 testbench file에 apply하고, 앞에서 본 결과를 살펴보면 먼저 a와 b의 sum이 잘 되었는지 확인하기 위하여 시나리오 상의 output constraint와 실제 회로의 output과 비교를 진행했고, 이후 해당 값과 오차를 error로 출력하였다. 앞의 Testbench의 결과로 clk를 apply한 것의 유무에 관계없이, error가 0을 유지하는 것을 볼 수 있다. 이를 통하여 회로의 logic이 26bit multiplier의 역할을 잘 수행하고 있음을 알 수 있었다.

(4) Reference

Verilog HDL -joseph Cavanagh

Verilog HDL 이론 -한양대학교 전자전기공학부

Quora- CPU clock speed vs power vs area

ResearchGate – Circuit Area vs frequency

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=laonple&logNo=220926179193>

고려대학교 전자전기공학부 – VLSI design practice05