

컴퓨터 공학 기초 실험2 보고서

실험제목: 32-bit carry look-ahead adder (CLA)

실험일자: 2023년 09월 23일 (토)

제출일자: 2023년 09월 25일 (월)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 월요일 0, 1, 2

학 번: 2022202065

성 명: 박나림

1. 제목 및 목적

A. 제목

32-bit carry look-ahead adder (CLA)

B. 목적

4-bit CLA를 설계하고, 이를 이용하여 32-bit CLA를 설계하도록 한다. 32-bit RCA도 설계하여 값을 확인해 본다. Timing analysis 결과를 확인하여 CLA와 RCA의 크기 및 속도를 비교하여 차이점을 알 수 있도록 한다.

2. 원리(배경지식)

A. Carry Look-ahead Adder (CLA)

기존의 Ripple Carry Adder (RCA)는 Full adder의 시간들을 다 고려해야 되기 때문에 시간이 많이 걸리는 단점이 있었다. 이를 보완하기 위해서 나온 것이 Carry Look-ahead Adder (CLA)이며, 모든 올림수가 동시에 처리되어 계산하는 데 걸리는 시간을 단축할 수 있는 가산기이다. carry만 따로 계산하는 carry look-ahead block이 있어서 미리 값을 구할 수 있기 때문에 빨라지는 것이다.

carry look-ahead block은 generation (G_i), propagation (P_i) signal들을 이용하여 만들 수 있다. $G_i = A_i B_i$, $P_i = A_i + B_i$ 로 정의된 식들을 carry out을 구하는 식으로 만들면 $C_{i+1} = A_i B_i + (A_i + B_i)C_i$ 로 되기 때문에 $C_{i+1} = G_i + P_i C_i$ 가 된다.

따라서 4-bit CLA의 carry를 미리 계산하면 다음과 같이 나온다.

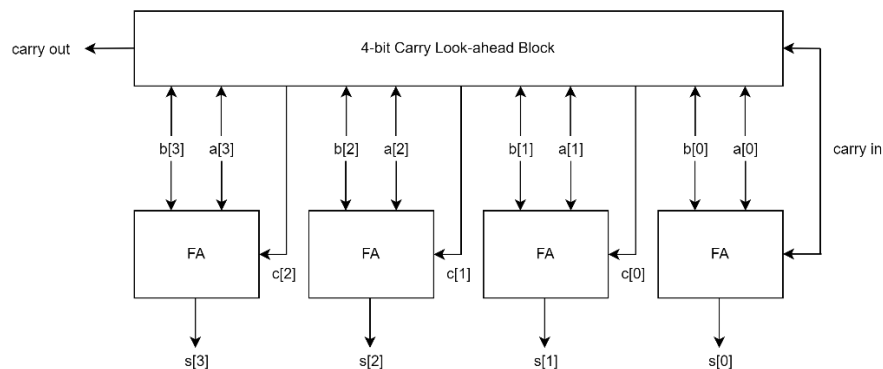
$$C_1 = G_0 + P_0 C_{in}$$

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_{in}$$

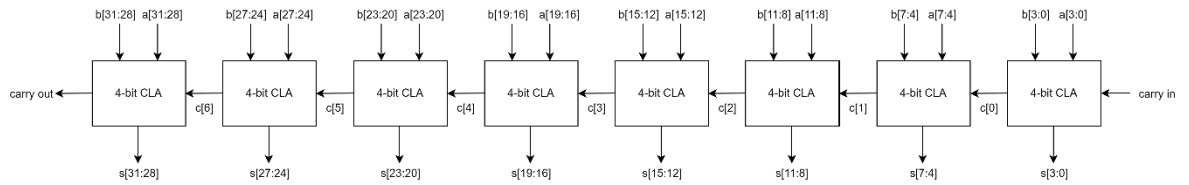
$$C_3 = G_2 + P_2(G_1 + P_1 G_0) + P_2 P_1 P_0 C_{in}$$

$$C_{out} = G_3 + P_3(G_2 + P_2(G_1 + P_1 G_0)) + P_3 P_2 P_1 P_0 C_{in}$$

이러한 과정으로 설계하면 다음과 같이 4-bit CLA를 만들 수 있다.



이러한 4-bit CLA를 이용하면 다음과 같이 32-bit CLA를 만들 수 있다.



B. Timing Analysis

설계한 회로가 정상적으로 작동할 수 있는 조건을 찾는 과정으로, 이러한 조건에 맞을 수 있도록 회로의 delay를 분석하는 것을 Timing analysis라고 한다. 이때 조건은 주로 최대 동작 주파수, Maximum clock frequency로 찾는다. 베릴로그에서 clk이 포함된 코드를 짜게 되면 slack이 발생하는데, destination flip-flop에 데이터가 도착해야 되는 시간과 실제 도착 시간의 차를 의미한다. 이러한 slack은 양수일 시 회로가 setup, hold time 조건을 만족한다는 뜻이고 음수일 시 시간의 여유가 부족하여 조건에 만족하지 못한다는 뜻이다. 실제로 데이터가 필요한 시점보다 늦게 도착할 시, 그 만큼의 시간이 음수 slack으로 표현되어 나타난다. 이는 clock을 다시 조절해줌으로써 해결할 수 있다. 또한 이와 함께 최대 동작 주파수를 확인할 수 있어 이와 반비례 되는 주기도 알 수 있게 된다.

3. 설계 세부사항

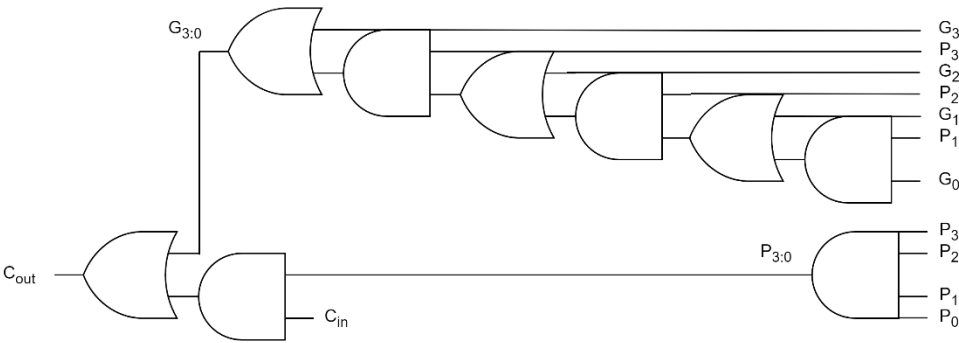
4-bit CLA를 이용하여 32-bit CLA를 설계하고, filp-flop을 추가하여 clock을 연결한 뒤 최대 동작 주파수를 찾도록 한다. 32-bit RCA 또한 같은 방법으로 설계하여 둘의 최대 동작 주파수를 비교하도록 한다.

A. 32-bit CLA (module cla32)

1) 입출력 (in/out)

Input	a (32 bit) //data
	b (32 bit) //data
	ci (1 bit) //carry in
Output	co (1 bit) //carry out
	s (32 bit) //sum
wire	c (8 bit) //carry wire

2) carry look-ahead block 회로

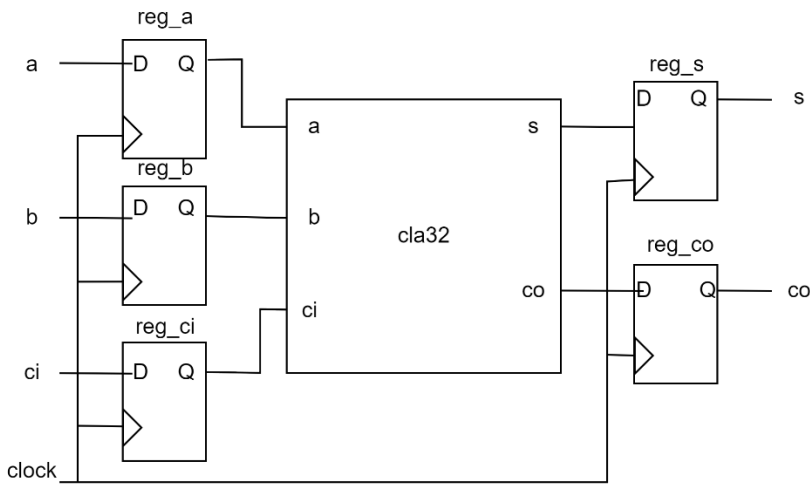


B. CLA with register (module cla_clk)

1) 입출력 (in/out)

Input	clk (1 bit) //clock
	a (32 bit) //data
	b (32 bit) //data
	ci (1 bit) //carry in
Output	co (1 bit) //carry out
	s (32 bit) //sum
reg	reg_a (32 bit) //register
	reg_b (32 bit) //register
	reg_ci (1 bit) //register
	reg_s (32 bit) //register
	reg_co (1 bit) //register

2) 회로도

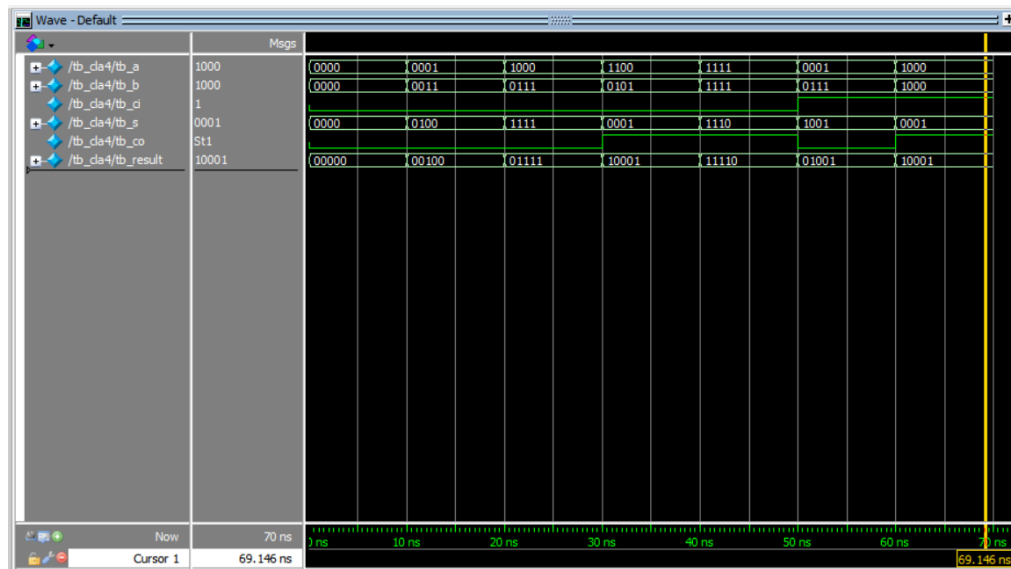


4. 설계 검증 및 실험 결과

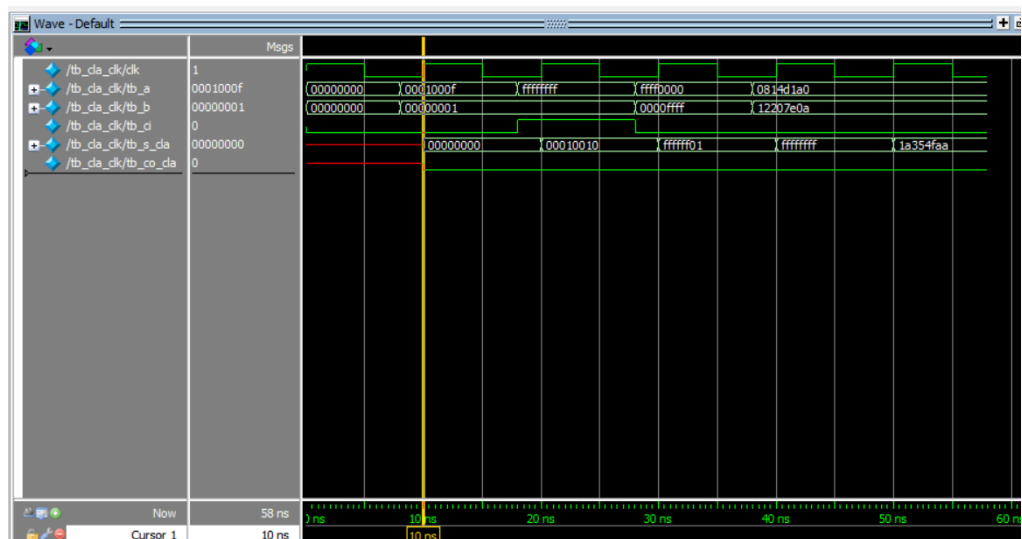
A. 시뮬레이션 결과

4-bit CLA는 지난번 실험 때의 4-bit RCA와 동일한 테스트 벤치 값으로 진행하였다. 그 다음 32-bit CLA와 32-bit RCA도 서로 같은 값으로 진행하여 각각 결과 값들이 서로 똑같이 나온다는 걸 확인할 수 있었다. 32 bit으로 테스트 벤치 값을 넣을 때는 좀 더 보기 편하게 하기 위해 16진수를 이용하여 값을 넣었다. 또한 추가로 clock을 넣어서 10ns 뒤부터 값이 변하는 것도 알 수 있다.

1) 4-bit CLA



2) 32-bit CLA



3) 32-bit RCA

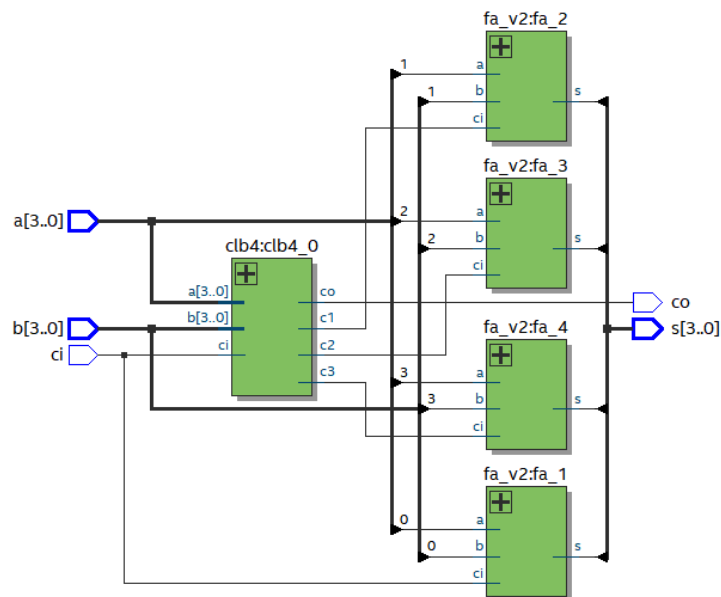


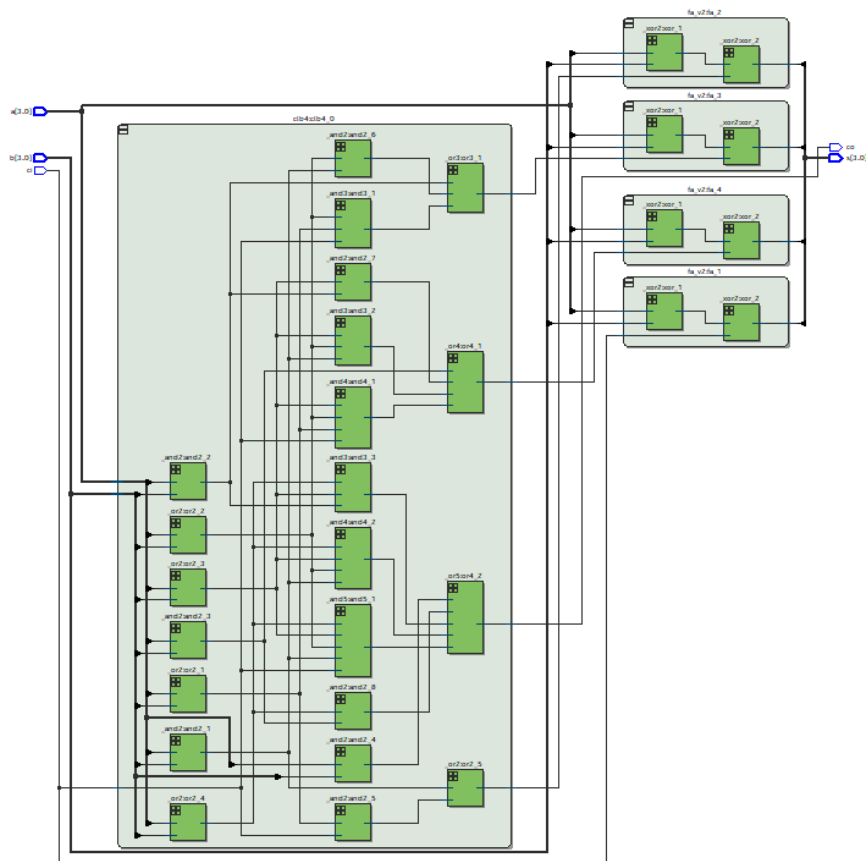
B. 합성(synthesis) 결과

각각 RTL view와 Flow summary를 확인해 본 결과이다.

1) 4-bit CLA

-RTL view



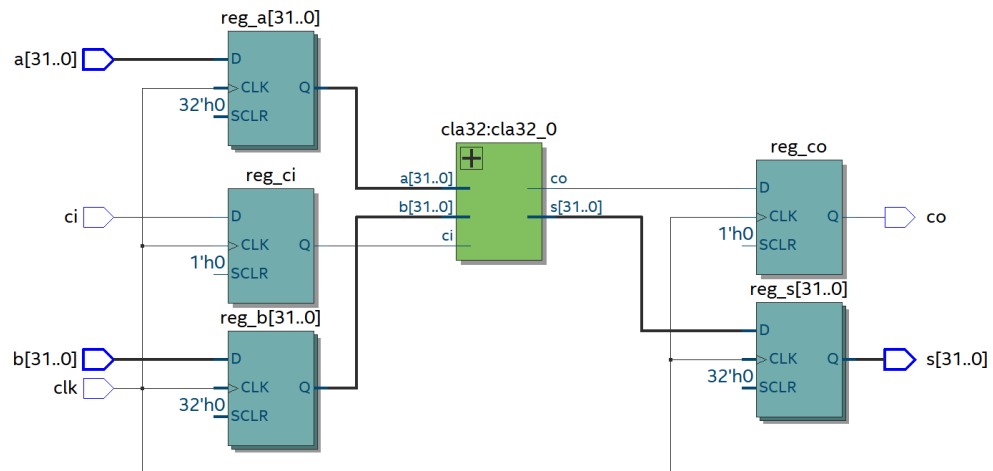


-Flow summary

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sat Sep 23 15:33:47 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	cla4
Top-level Entity Name	cla4
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	4 / 41,910 (< 1 %)
Total registers	0
Total pins	14 / 499 (3 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

2) 32-bit CLA

-RTL view

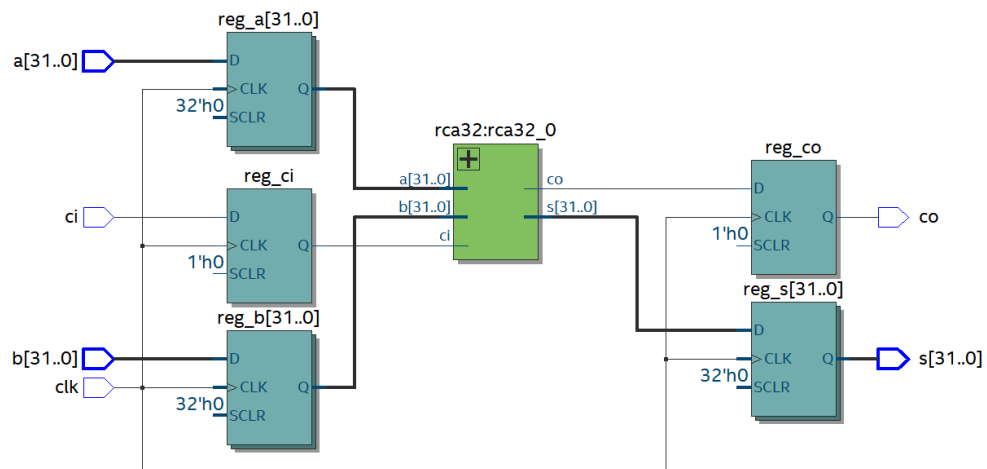


-Flow summary

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Sep 25 11:02:44 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	cla_clk
Top-level Entity Name	cla_clk
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	43 / 41,910 (< 1 %)
Total registers	98
Total pins	99 / 499 (20 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

3) 32-bit RCA

-RTL view



-flow summary

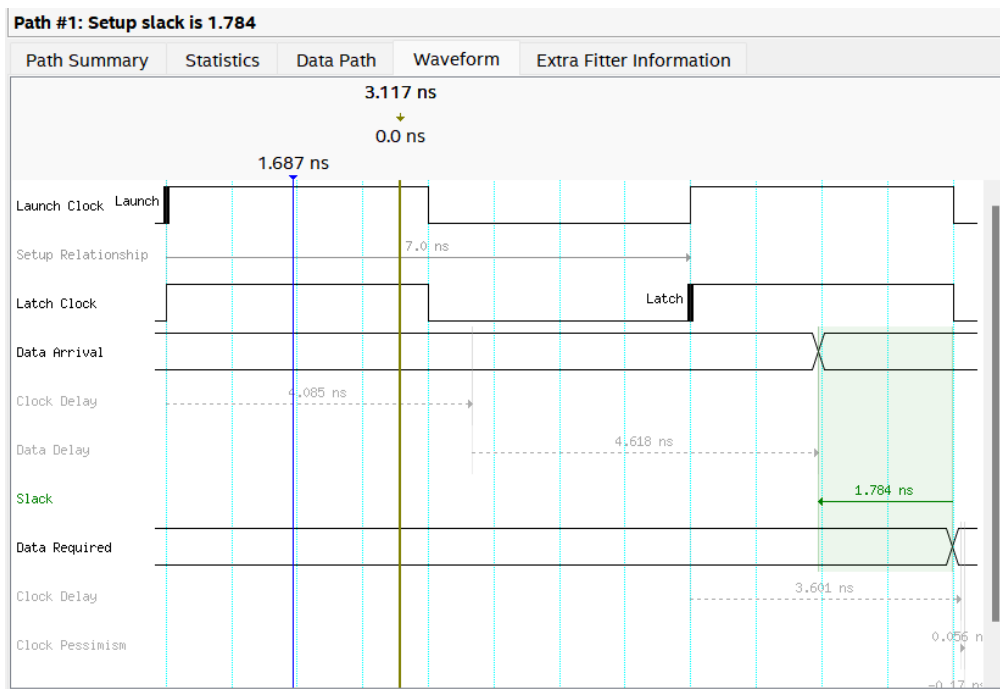
Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun Sep 24 14:09:14 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	rca_clk
Top-level Entity Name	rca_clk
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	31 / 41,910 (< 1 %)
Total registers	98
Total pins	99 / 499 (20 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

C. Timing analysis 결과

1) 32-bit CLA

-Setup clock: Waveform

7ns로 clock을 조절한 후, slack이 양수로 나왔으므로 clock 조건을 위반하지 않게 되었다는 것을 알 수 있다.



-Setup clock: Fmax

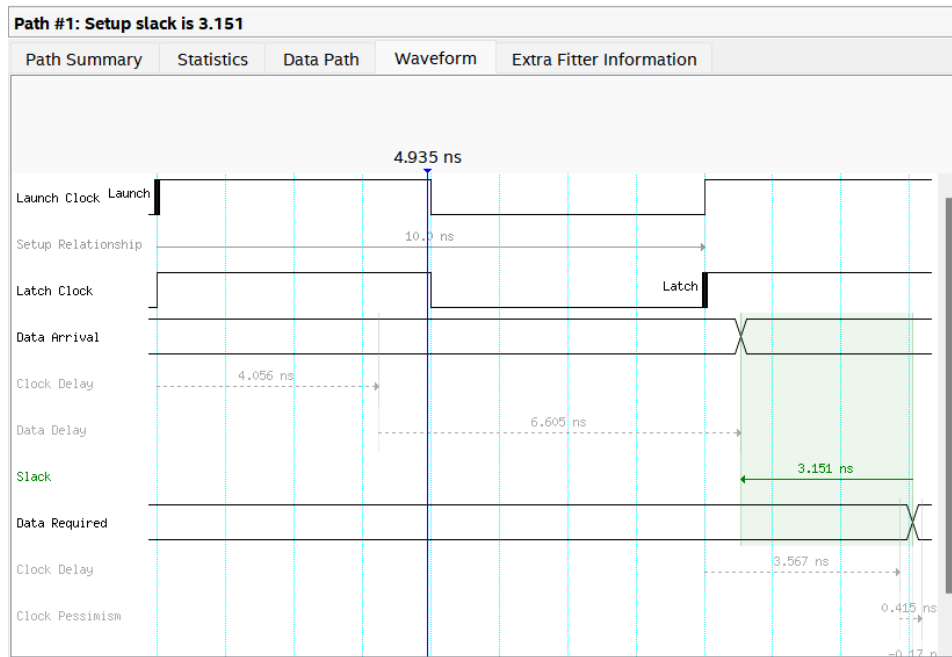
cla_clk 모듈에서 조건을 위반하지 않는 최대 frequency는 191.72 MHz라는 것을 보여준다.

Slow 1100mV 85C Model				
	Fmax	Restricted Fmax	Clock Name	Note
1	191.72 MHz	191.72 MHz	clk	

2) 32-bit RCA

-Setup clock: Waveform

10ns로 clock을 조절한 후, slack이 양수로 나왔으므로 clock 조건을 위반하지 않게 되었다는 것을 알 수 있다.



-Setup clock: Fmax

rca_clk 모듈에서 조건을 위반하지 않는 최대 frequency는 146.01 MHz라는 것을 보여준다.

Slow 1100mV 85C Model				
	Fmax	Restricted Fmax	Clock Name	Note
1	146.01 MHz	146.01 MHz	clk	

5. 고찰 및 결론

A. 고찰

timing analysis를 진행할 때, CLA는 7ns로 새 clock을 설정하여 정상적으로 slack이 양수로 나오는 것을 확인하였다. 하지만 RCA에서는 똑같이 7ns로 설정하였다가 slack이 여전히 음수로 나오는 상황이 발생하였다. 이에 slack을 다시 확인해본 결과, RCA는 클럭 주기의 시간이 더 필요하다는 것을 알게 되어 10ns로 수정하였다. 그 결과 값이 양수로 나오면서 정상적인 웨이브폼을 확인할 수 있었다.

B. 결론

Flow summary에 의해 32-bits CLA와 32-bits RCA의 크기 및 속도 차이를 알 수 있었다. logic utilization을 보면 논리 회로의 크기를 볼 수 있는데, 32-bits CLA의 경우 43, 32-bits RCA는 31로 CLA의 크기가 더 크게 나온 것을 알 수 있다. 하지만 Timing analysis를 통해 속도를 확인해 보면 CLA의 최대 frequency는 191.72 MHz, RCA는 146.01 MHz였다. 이러한 주파수는 클럭 신호의 주기의 역수이므로, CLA의 주기가 더 짧아서 속도가 빠르다는 것을 알 수 있다. 실제로 slack을 양수로 바꾸기 위해 clock을 조절했을 때, CLA는 7ns, RCA는 10ns로 했을 때 정상적으로 나올 수 있었으므로, RCA가 더 오래 걸린다는 것을 깨달을 수 있었다.

또한 기존의 32-bits CLA에서 carry look-ahead block을 수정하여 modified 32-bits CLA를 설계하여 비교를 크기 및 속도를 비교해보았다. modified 32-bits CLA는 carry propagate를 최적화하여 비트 전파를 효율적으로 바꾸게 되므로 계산하는 데 걸리는 시간이 더 단축된다. 크기는 설계에 따라 다르게 나올 수 있지만, 이번에 실험한 결과에서는 회로의 크기가 47로 기존 CLA보다 조금 더 크게 나왔다.

이러한 결과들을 봤을 때, RCA보다 CLA가 크기는 더 크지만 속도는 더 빠르고, 수정된 CLA는 크기는 거의 비슷하지만 속도는 더 빠르게 나올 수 있다는 걸 깨달았다.

6. 참고문헌

David Money Harris and Sarah L. Harris / Digital Design and Computer Architecture / Elsevier / 2007