

9주차 : XOR Gate and Full Adder

조 명 : 똑딱똑딱

1. Fig. 1 shows a delay measurement setup for an XOR gate. Design the XOR gate so that the worst-case delay from (X or Y) to Z is less than 80 ps. The inverters are of the same type as shown in the figure. Verify your design through transient simulations. For the inputs X and Y, use the 'vpwlf' devices from the 'analogLib' library with the pwl files '~ /pwl xor x' and '~ /pwl xor y', respectively. Set the 'Period of the PWL' as 1.6 ns. Use the supply voltage of 1 V.

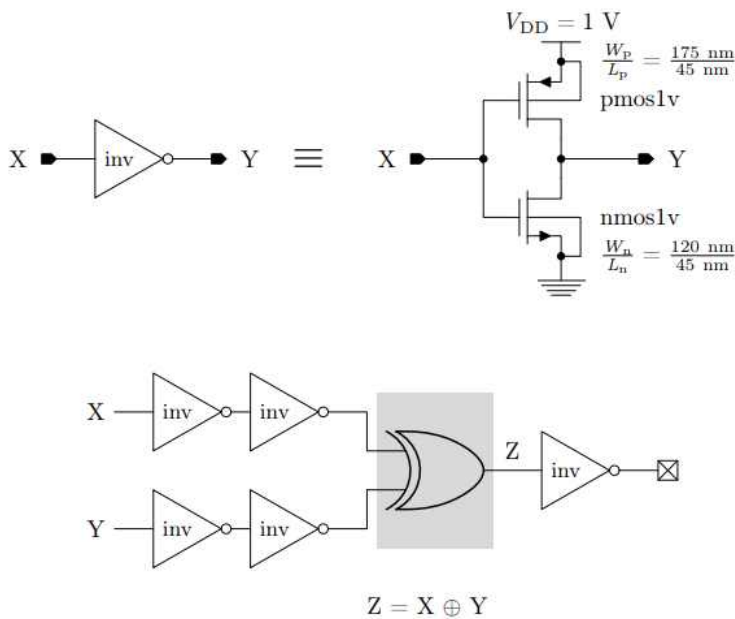


Fig. 1: A delay measurement setup for an XOR gate.

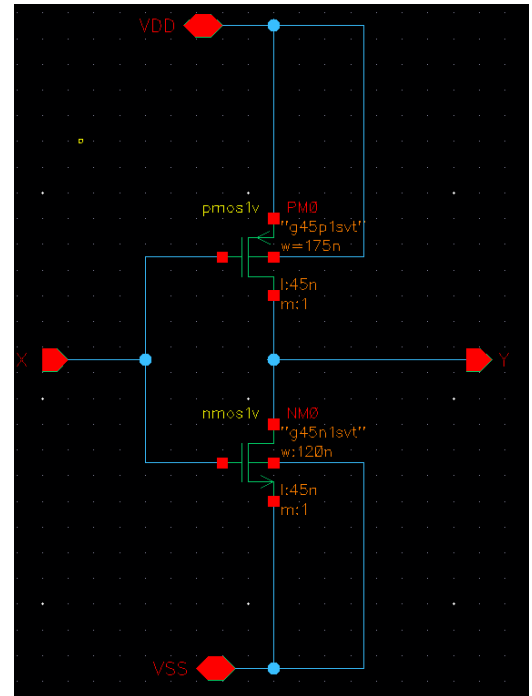


그림 1-1) Inverter 내부 회로

XOR 이나 Full Adder를 설계하기 위해서는 Inverter 의 사용이 필요하다.

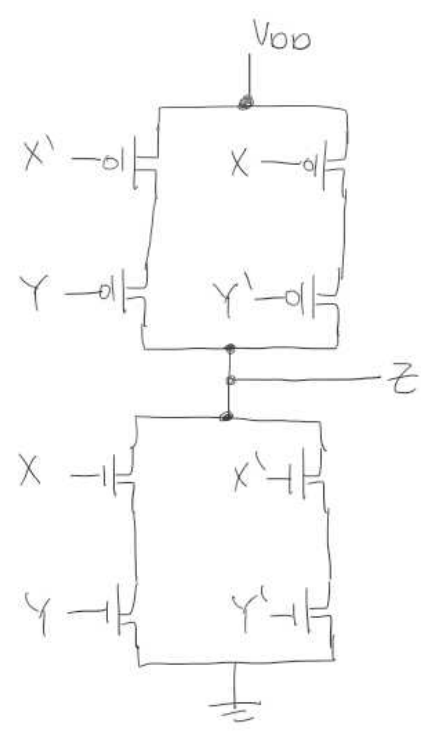
이전 과제에서 설계한 PMOS Width 와 NMOS Width 의 1.2 배의 차이를 이용하지 않고 주어진 Width를 이용해 설계를 진행하였다.

설계에 앞서 Exclusive OR 의 진리표를 통해 PMOS 와 NMOS 관점에서 회로를 구성한 후 CMOS 구조로 연결하였다.

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

표1-1) XOR 진리표

PMOS 에서는 $Z = A'B + AB'$ 으로 회로를 구성하면 된다.
 NMOS 에서는 Z' 으로 회로를 구성해야 한다. $Z' = (A'B + AB')' = (A+B')(A'+B)$ 이므로 $AA'+AB+A'B'+B'B$ 에 의
 해 $Z'=AB+A'B'$ 이 된다.



따라서 왼쪽 그림처럼 회로를 구성할 수 있다.
 이 때, 입력에 NOT 이 들어간 부분을 확인할 수 있다.
 사전에 주어진 PMOS Width 175nm, NMOS Width 120nm 의 Inverter를 이
 용해 NOT 으로 입력을 해줄 수 있다.

그리고 Worst-case에서도 XOR Gate 로 정상동작을 할 수 있도록 Sizing을
 해줄 필요성이 있다.
 앞선 과제에서 PMOS 와 NMOS 의 Width length 가 1.2 배 차이남을 확인하
 였다.

그림1-2) XOR CMOS 회로

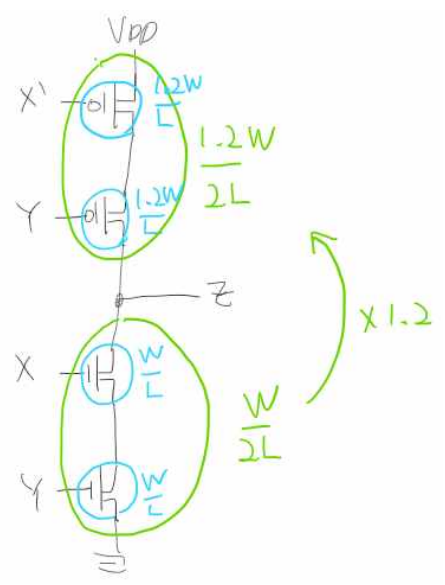


그림1-3) XOR Worst-Case

Worst-case 의 경우 다음과 같다.
 NMOS 의 사이즈를 $\frac{W}{L}$ 이라고 가정할 때, NMOS 2개를 큰 소자로 본다면 2
 배 나눠진 $\frac{W}{2L}$ 가 된다.
 여기에 pr=1.2 가 곱한 $\frac{1.2W}{2L}$ 가 PMOS 2개를 큰 소자로 본 사이즈이며,
 2배를 곱한 $\frac{1.2W}{L}$ 가 PMOS 각각의 사이즈이다.

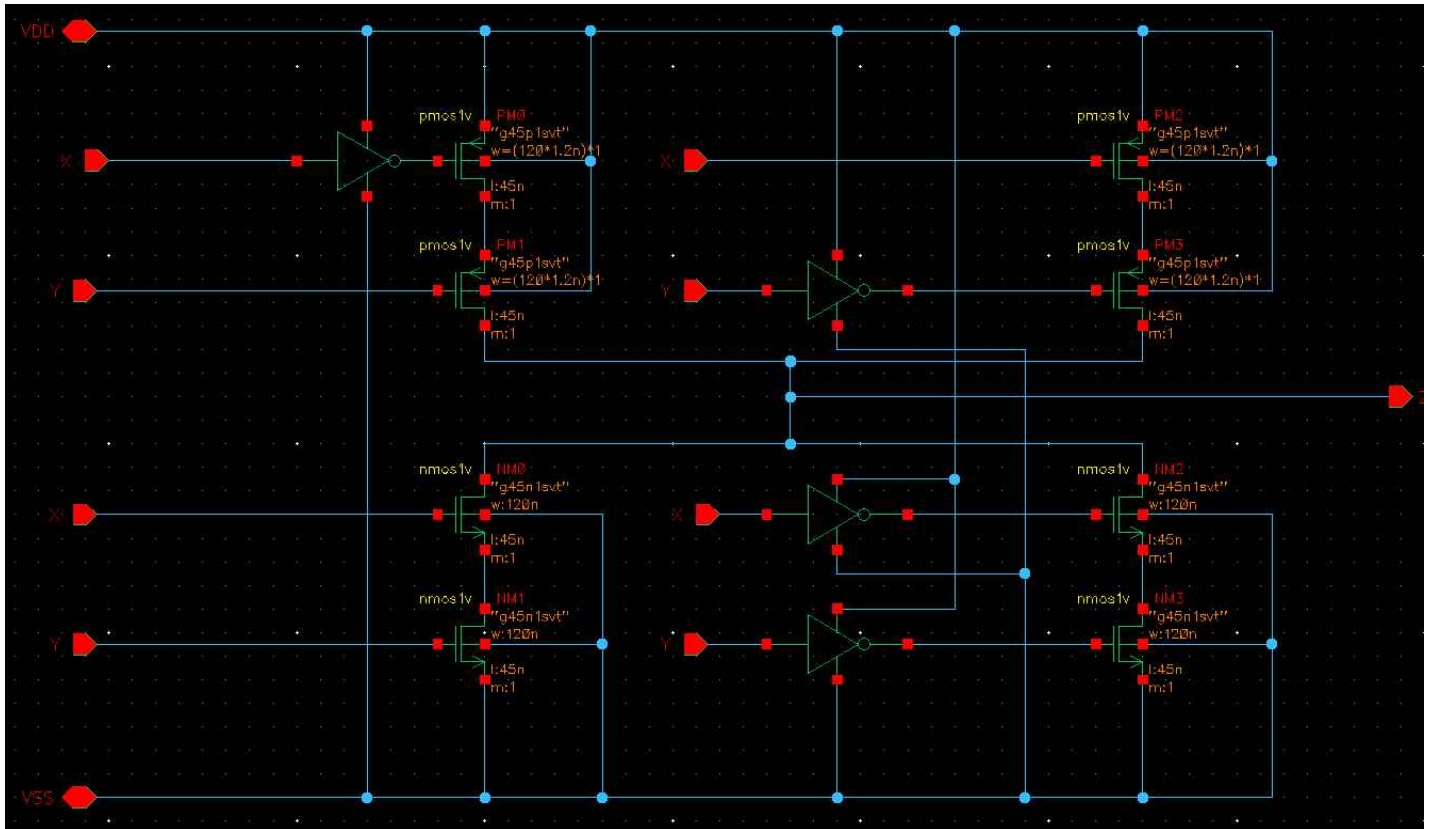


그림1-4) XOR Gate 회로도

nmos1v, pmos1v를 처음 추가시켰을 때 default 로 지정되는 width 와 length를 그대로 사용하여 schematic을 구성하였다.

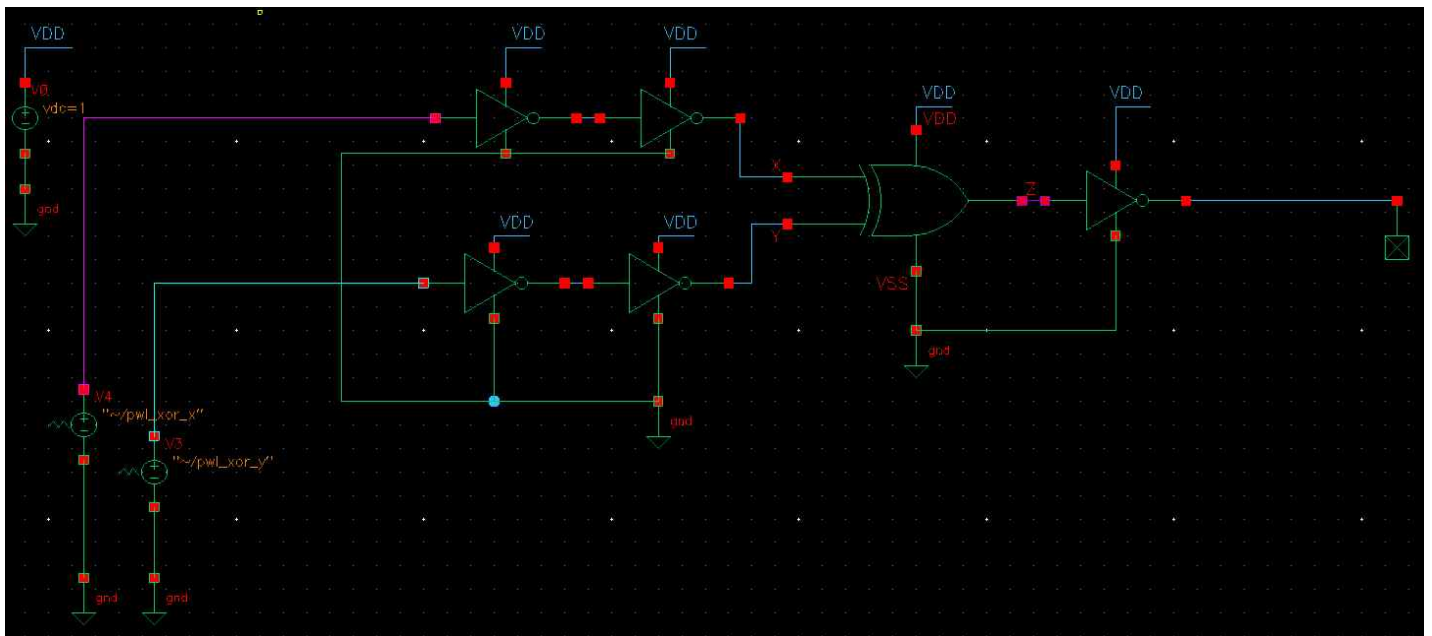


그림1-5) XOR Gate Test bench 회로도

앞서 만든 XOR Gate를 symbol 로 만들어서 delay를 측정하기 위한 회로를 구성하였다.

Type	Enable	Arguments
1 tran	<input checked="" type="checkbox"/>	0 1.6n

위와 같은 환경에서 transient Analysis를 진행하였다.

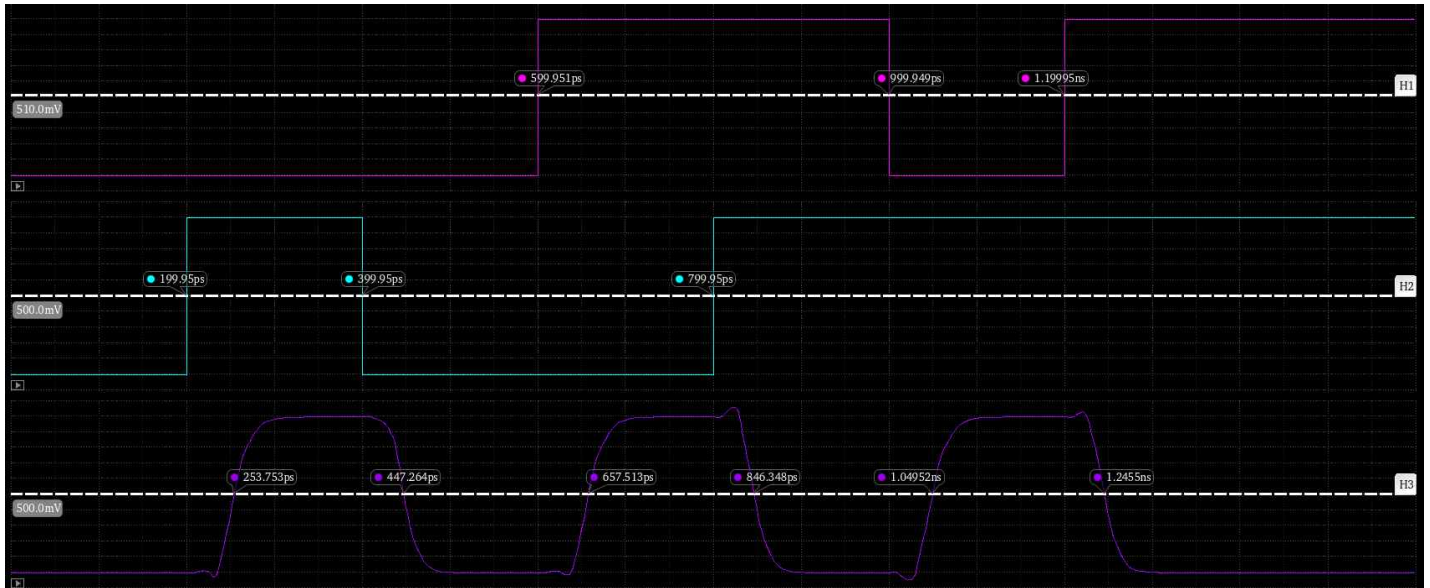


그림1-5) XOR Gate Test bench 신호흐름선도 (Width 120nm)

테스트 해본 결과, 정상적으로 Exclusive OR Gate로 작동을 하였다.

X	Y	Z	Delay
	199.95	235.753	35.803
	399.95	447.264	47.314
599.951		657.513	57.562
	799.95	846.348	46.398
999.949		1049.52	49.571
1199.95		1245.5	45.55

표1-2) Width 120nm 기준 Delay (단위 : ps)

설계조건인 80ps 이내의 Delay를 가지므로 적합한 설계라고 판단할 수 있다.

Width 가 어떤 수치를 가질 때 delay 가 80ps 가 되는지 확인해보았다.



그림1-5) Width 변화에 따른 XOR Gate 신호흐름선도

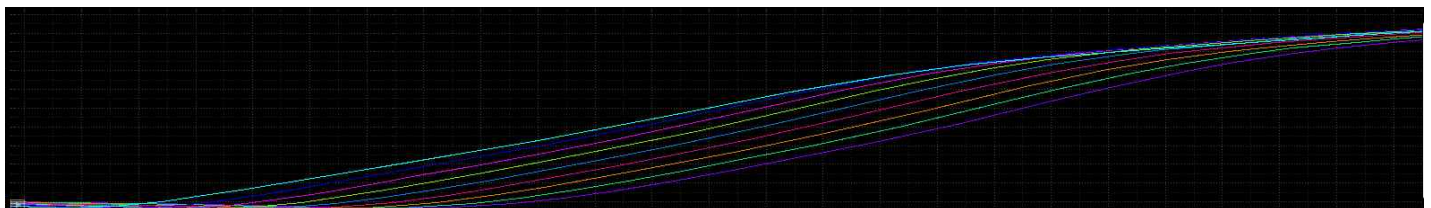


그림1-6) Width 변화에 따른 XOR Gate 신호흐름선도 확대

Width 가 커질수록 delay 가 길어지고 있음을 확인할 수 있다.

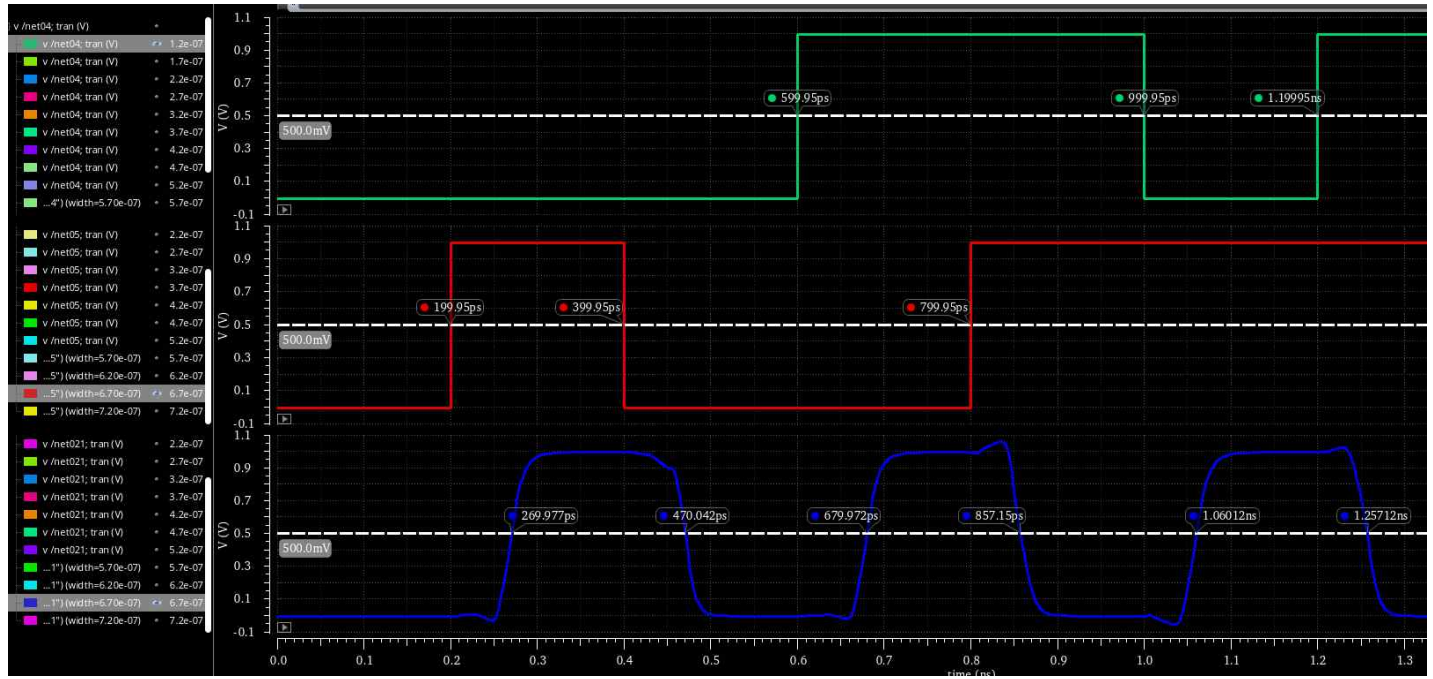


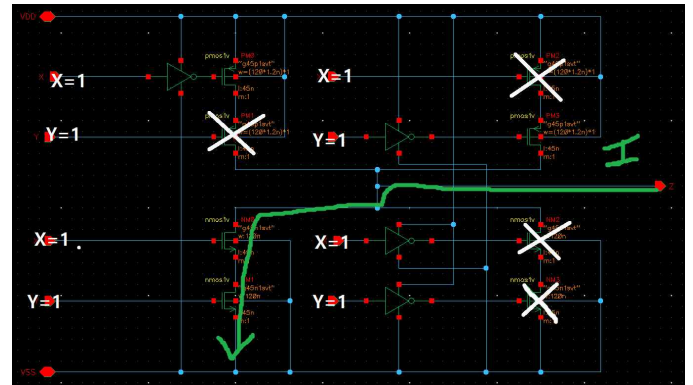
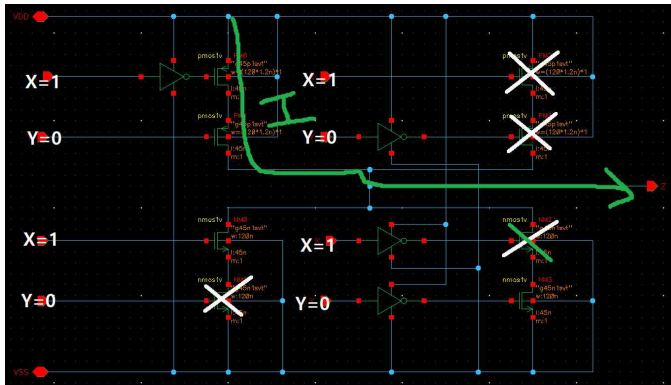
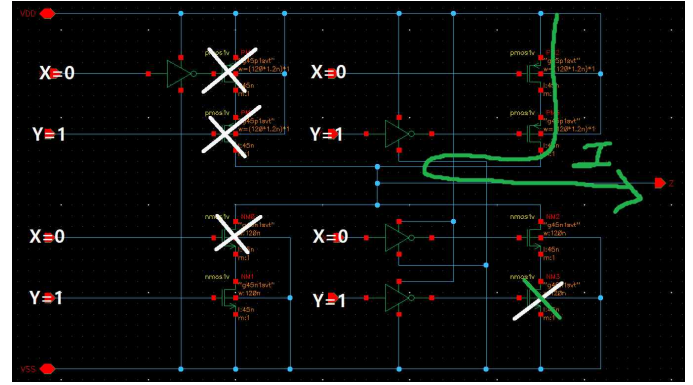
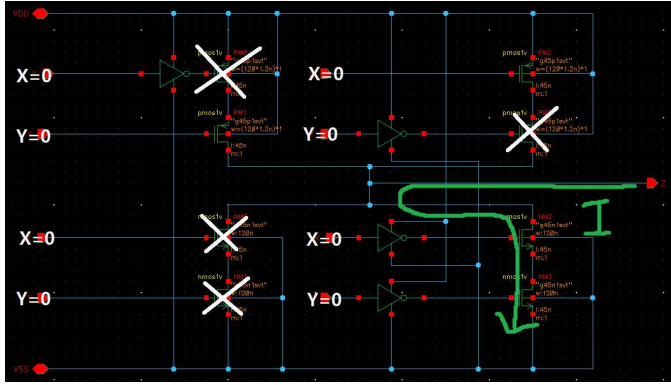
그림1-7) XOR Gate Test bench 신호흐름선도 (W 670nm)

X	Y	Z	Delay
	199.95	269.977	70.027
	399.95	470.042	70.092
599.951		679.972	80.021
	799.95	857.145	57.195
999.949		1060.12	60.171
1199.95		1257.12	57.17

표1-3) Width 670nm 기준 Delay (단위 : ps)

Width 가 670nm 일 때 delay 가 80ps 로 걸치게 된다.

XOR Gate 동작을 위해 전류흐름



Channel Width 가 커질수록 동작 속도가 나빠지는 것에 대한 고찰

일반적으로 $I = Aqnv$ 이므로 Channel Width 가 증가하면 electron 과 hole 이 많이 움직일 수 있기 때문에 동작 속도는 좋아져야 한다.

하지만 실제로는 Inverter 와 XOR Gate 에서의 delay가 증가했다.

Subthreshold Leakage $I_{D(OFF)} = \mu_n C_{ox} \frac{W}{L} \frac{\gamma V_T^2}{2 \sqrt{2\Phi_F + V_{SB}}} e^{\frac{-V_{TH}}{nV_T}}$ 이므로, W 가 증가할수록 Off current가 증가한다고 볼 수 있다.

해당 현재 gpdk045을 사용한 공정은 45nm 공정으로 집적도가 상당히 높고, V_{DD} 가 낮은 동작을 하고 있으므로

Power Equation $P = C_L V_{DD}^2 f + \frac{2}{3} \beta_n \left(\frac{V_{DD}}{2 - V_{THn}} \right)^3 t_r f + V_{DD} I_{OFF}$ 에서 V_{DD} 가 큰 공정에 비해 Leakage Power $V_{DD} I_{OFF}$ 가 크게 차지할 것으로 예상된다. 그 이유는 Dynamic Power, Short-circuit Power 부분에서 제곱, 세제곱이 있지만 1을 제곱하므로 1V 로 계산되기 때문이다.

이로 인해 Leakage Current 증가로 소자의 동작 특성이 나빠진 것 같다.

2. Fig. 2 shows a delay measurement setup for a full adder. Design the full adder such that the worst-case delay from (A, B, or C_{IN}) to (S or C_{OUT}) is less than 150 ps. The inverters are of the same type as the ones in Fig. 1. Verify your design through simulations. Use the pwl files '~\pwl fa a', '~\pwl fa b', and '~\pwl fa cin' for the inputs A, B, and C_{IN}, respectively. Set the 'Period of the PWL' as 8.8 ns. Use the supply voltage of 1 V.

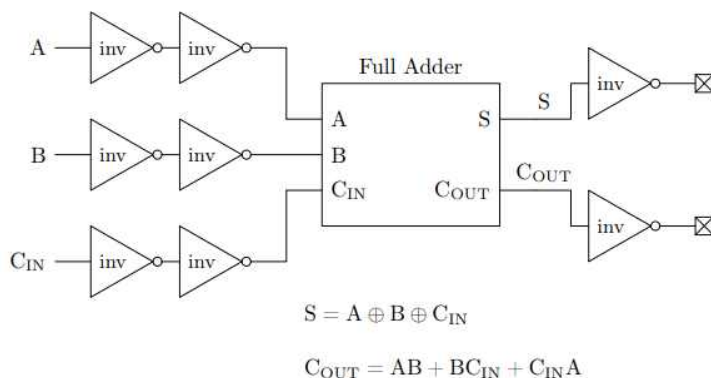


Fig. 2: A delay measurement setup for a full adder.

설계에 앞서 Full Adder 의 진리표를 통해 PMOS 와 NMOS 관점에서 회로를 구성한 후 CMOS 구조로 연결하였다.

X	Y	Z	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

표2-1) 2bits Full Adder 진리표

우선 Sum 에 대한 회로를 만들기 위해 Sum 과 Sum' 의 논리식을 알아야 한다.
Sum 과 Sum' 의 카르노맵은 다음과 같다.

AB \ Cin	0	1
00	0	1
01	1	0
11	0	1
10	1	0

표2-2) Sum 의 카르노맵

AB \ Cin	0	1
00	1	0
01	0	1
11	1	0
10	0	1

표2-3) Sum' 의 카르노맵

이 카르노맵을 논리식으로 정리하면 다음과 같다.

$$\text{Sum} = A'B'C_{in} + ABC_{in} + A'BC_{in}' + AB'C_{in}'$$

$$\text{Sum}' = A'B'C_{in}' + A'BC_{in} + ABC_{in}' + AB'C_{in}$$

PMOS 에서는 Sum 으로 NMOS 에서는 Sum' 으로 회로를 구성해야 한다.

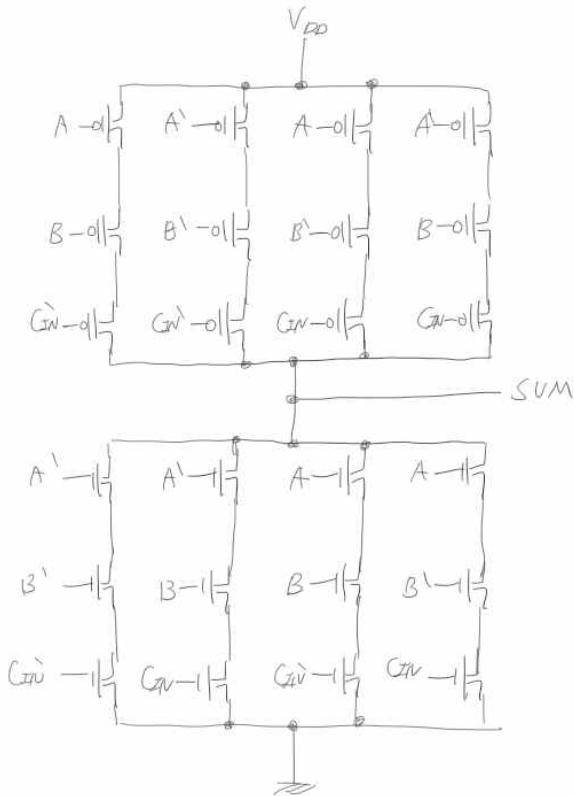


그림2-1) Full Adder 의 Sum 회로

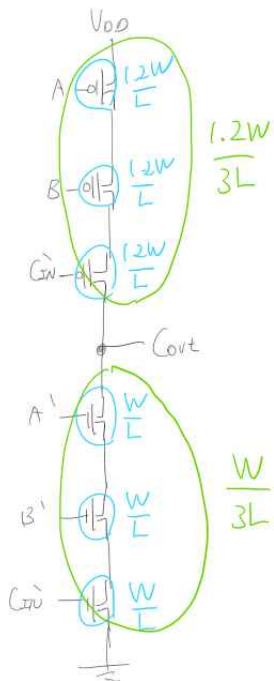
따라서 왼쪽 그림처럼 회로를 구성할 수 있다.

이 때, 입력에 NOT 이 들어간 부분을 확인할 수 있다.

사전에 주어진 PMOS Width 175nm, NMOS Width 120nm 의 Inverter를 이용해 NOT 으로 입력을 해줄 수 있다.

그리고 Worst-case에서도 XOR Gate 로 정상동작을 할 수 있도록 Sizing을 해줄 필요성이 있다.

앞선 과제에서 PMOS 와 NMOS 의 Width length 가 1.2 배 차이남을 확인하였다.



Worst-case 의 경우 다음과 같다.

NMOS 의 사이즈를 $\frac{W}{L}$ 이라고 가정할 때, NMOS 3개를 큰 소자로 본다면 3배 나뉘진 $\frac{W}{3L}$ 가 된다.

여기에 pr=1.2 가 곱한 $\frac{1.2W}{3L}$ 가 PMOS 3개를 큰 소자로 본 사이즈이며,

3배를 곱한 $\frac{1.2W}{L}$ 가 PMOS 각각의 사이즈이다.

그림2-2) Full Adder
Sum Worst-Case

다음으로 Cout 에 대한 회로를 만들기 위해 Cout 과 Cout' 의 논리식을 알아야 한다.
Sum 과 Sum' 의 카르노맵은 다음과 같다.

AB \ Cin	0	1
00	0	0
01	0	1
11	1	1
10	0	1

표2-4) Cout 의 카르노맵

AB \ Cin	0	1
00	1	1
01	1	0
11	0	0
10	1	0

표2-5) Cout' 의 카르노맵

이 카르노맵을 논리식으로 정리하면 다음과 같다.
 $Cout = AB + BCin + CinA$
 $Cout' = A'B' + A'Cin' + B'Cin'$
 PMOS 에서는 Cout 으로 NMOS 에서는 Cout' 으로 회로를 구성해야 한다.

따라서 다음 그림처럼 회로를 구성할 수 있다.
 그리고 Worst-case에서도 XOR Gate 로 정상동작을 할 수 있도록 Sizing을 해줄 필요성이 있다.

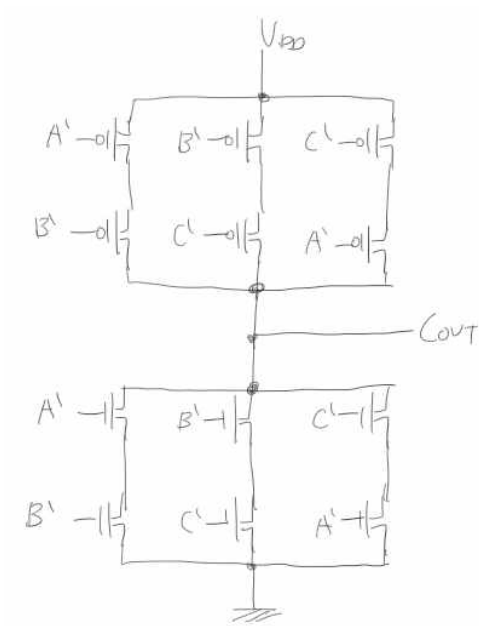


그림2-3) Full Adder 의 Cout 회로

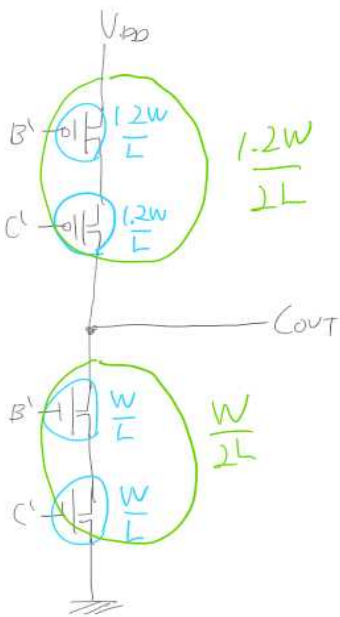


그림2-4) Full Adder
Cout Worst-Case

Worst-case 의 경우 NMOS 의 사이즈를 $\frac{W}{L}$ 이라고 가정할 때, NMOS 3개를 큰 소자로 본다면 2배 나뉘진 $\frac{W}{2L}$ 가 된다. 여기에 pr=1.2 가 곱한 $\frac{1.2W}{2L}$ 가 PMOS 2개를 큰 소자로 본 사이즈이며, 2배를 곱한 $\frac{1.2W}{L}$ 가 PMOS 각각의 사이즈이다.

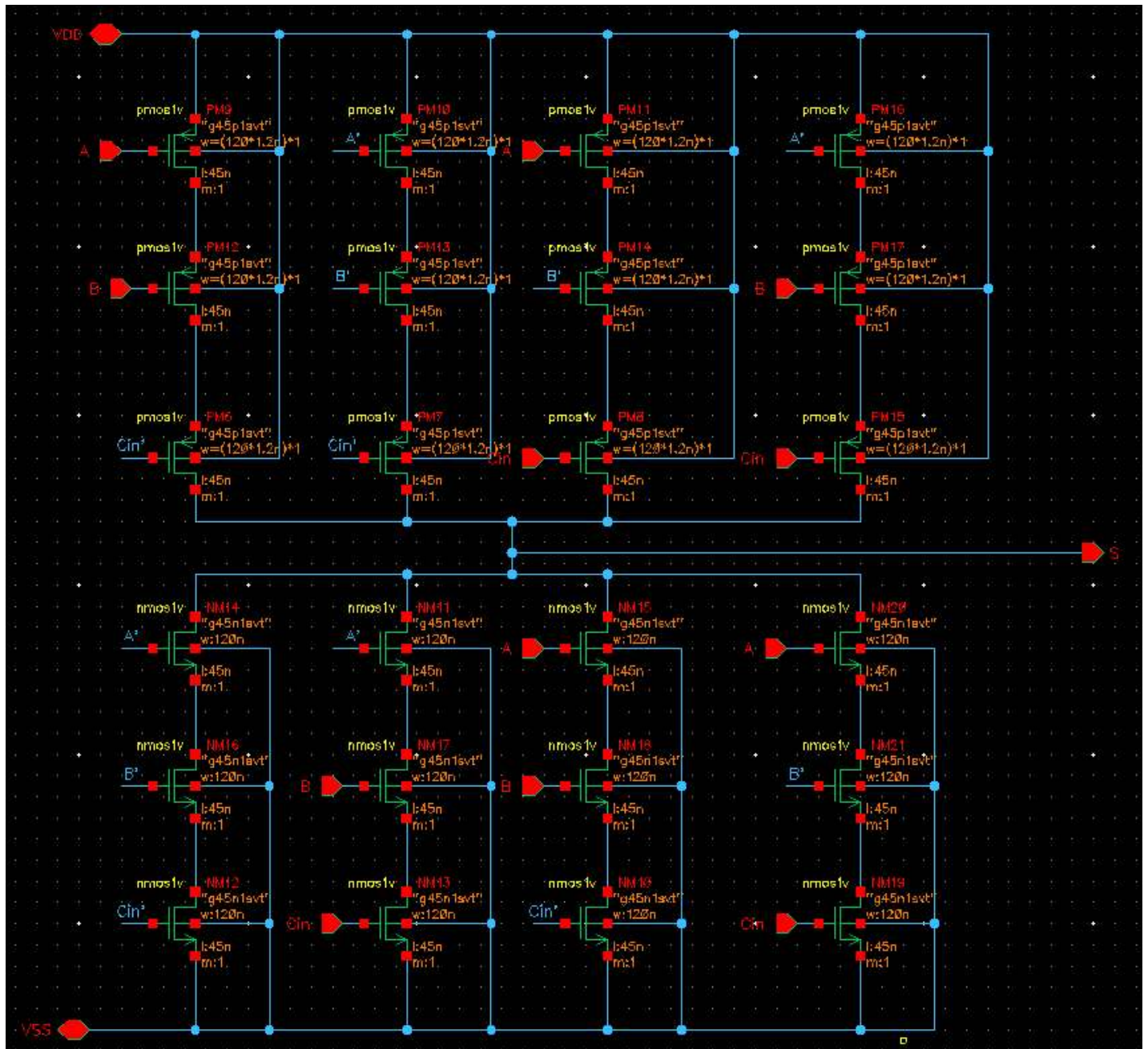


그림2-5) Full Adder 의 Sum 회로도

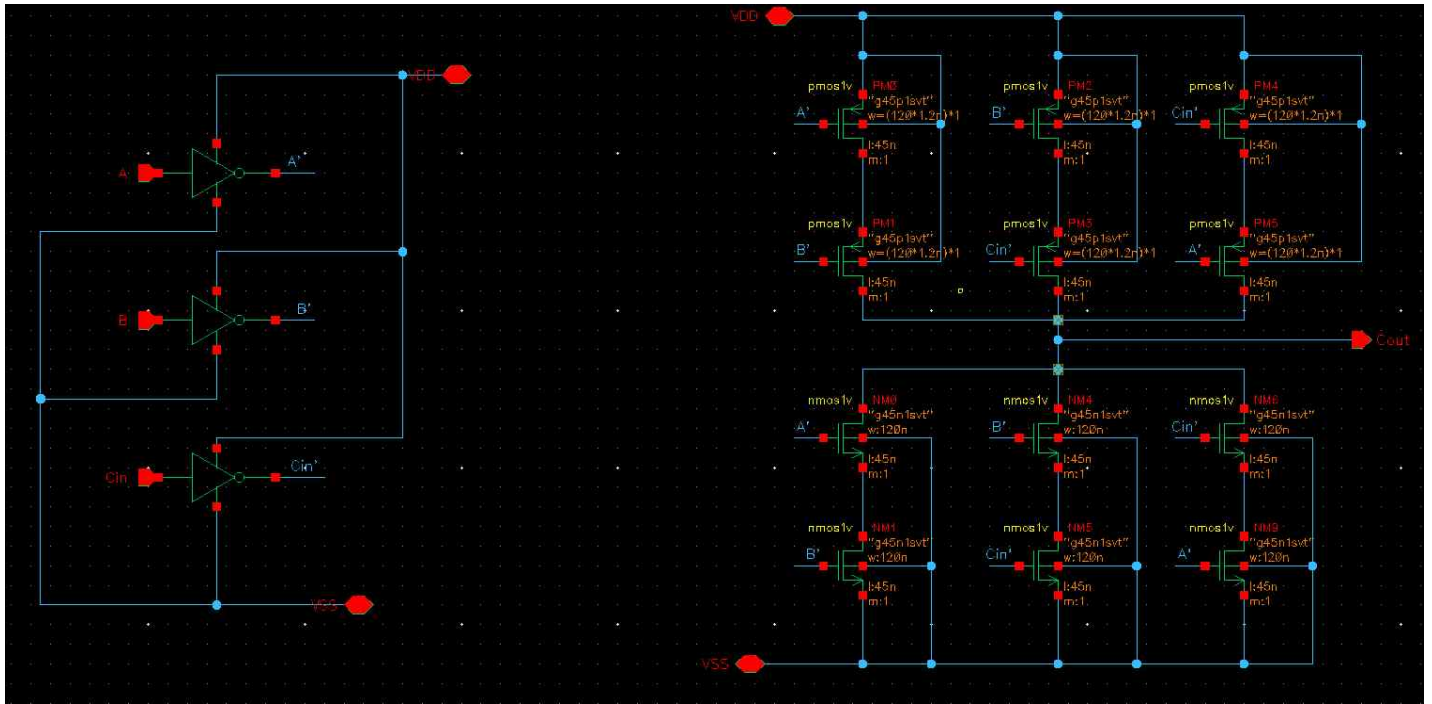


그림2-6) Full Adder 의 Cout 회로도

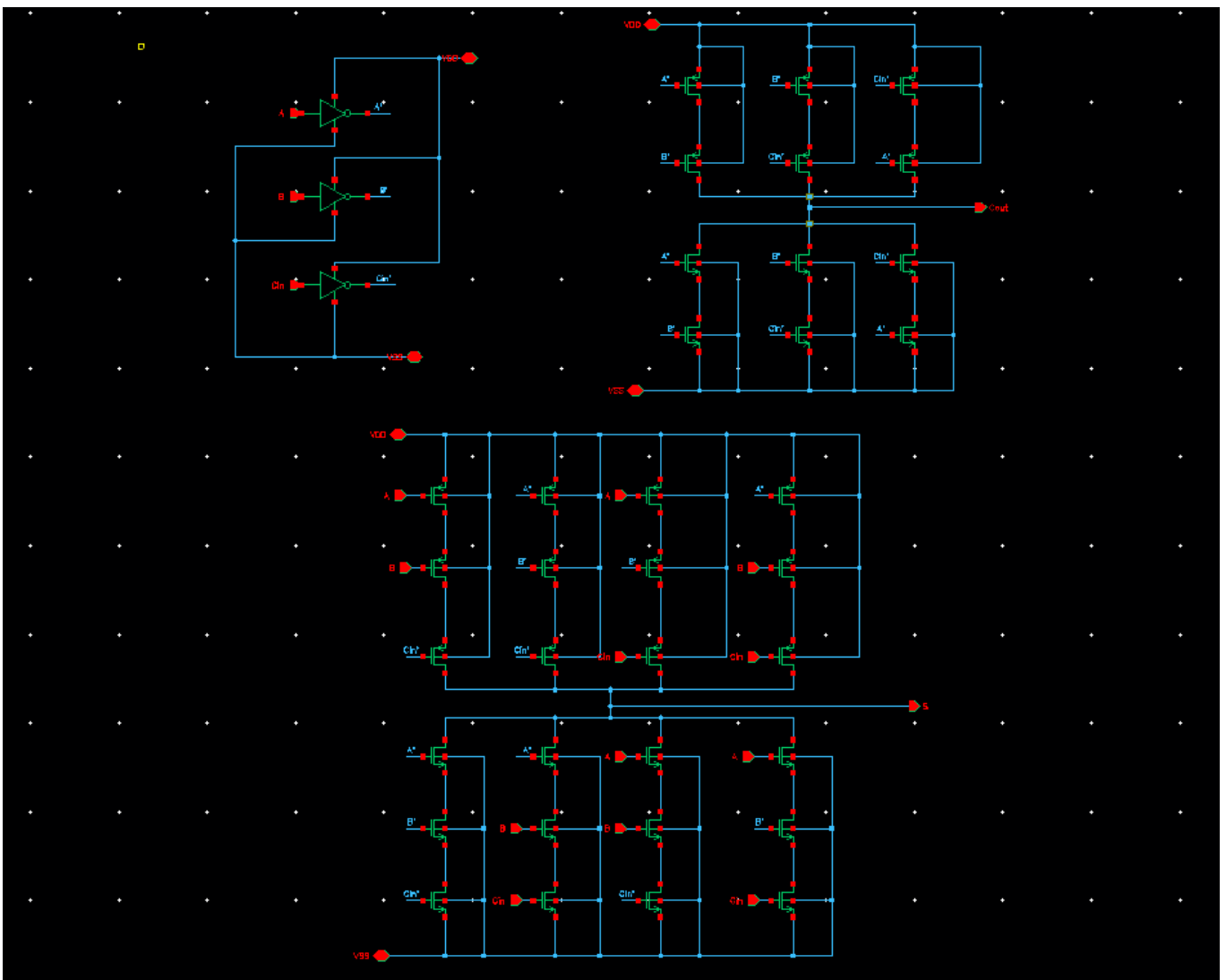


그림2-7) Full Adder 전체 회로도

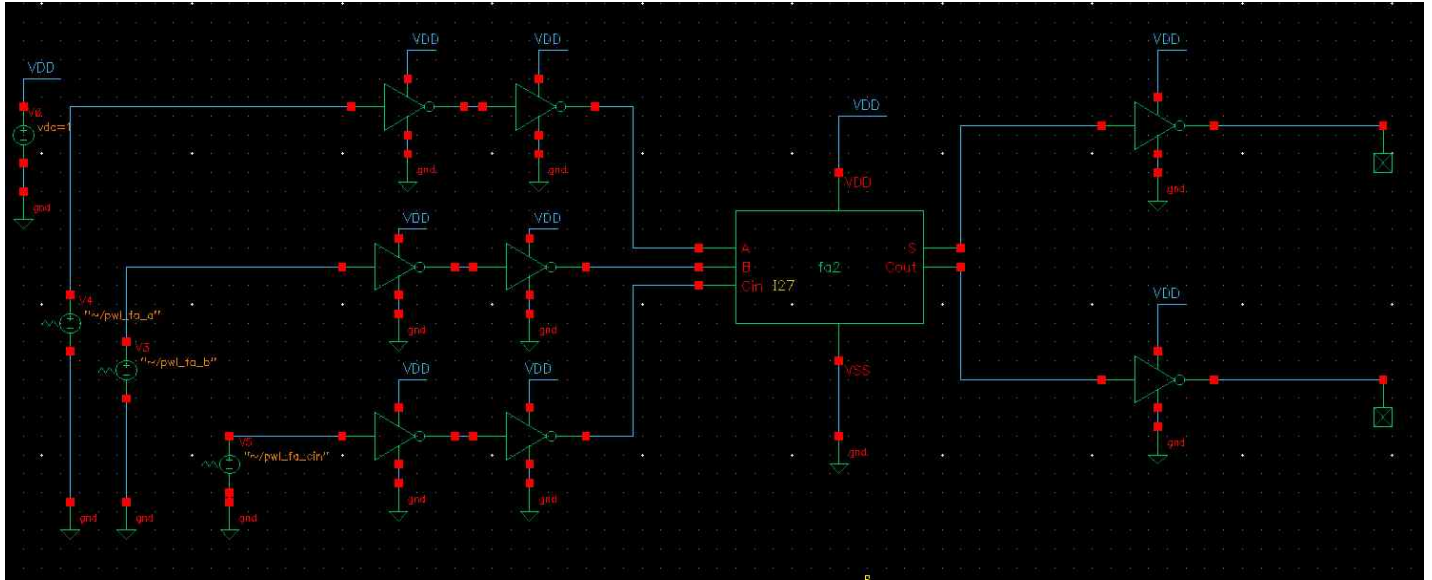


그림2-8) Full Adder Test bench 회로도

앞서 만든 Full Adder를 symbol 로 만들어서 delay를 측정하기 위한 회로를 구성하였다.

Type	Enable	Arguments
1 tran	<input checked="" type="checkbox"/>	0 8.8n

위와 같은 환경에서 transient Analysis를 진행하였다.

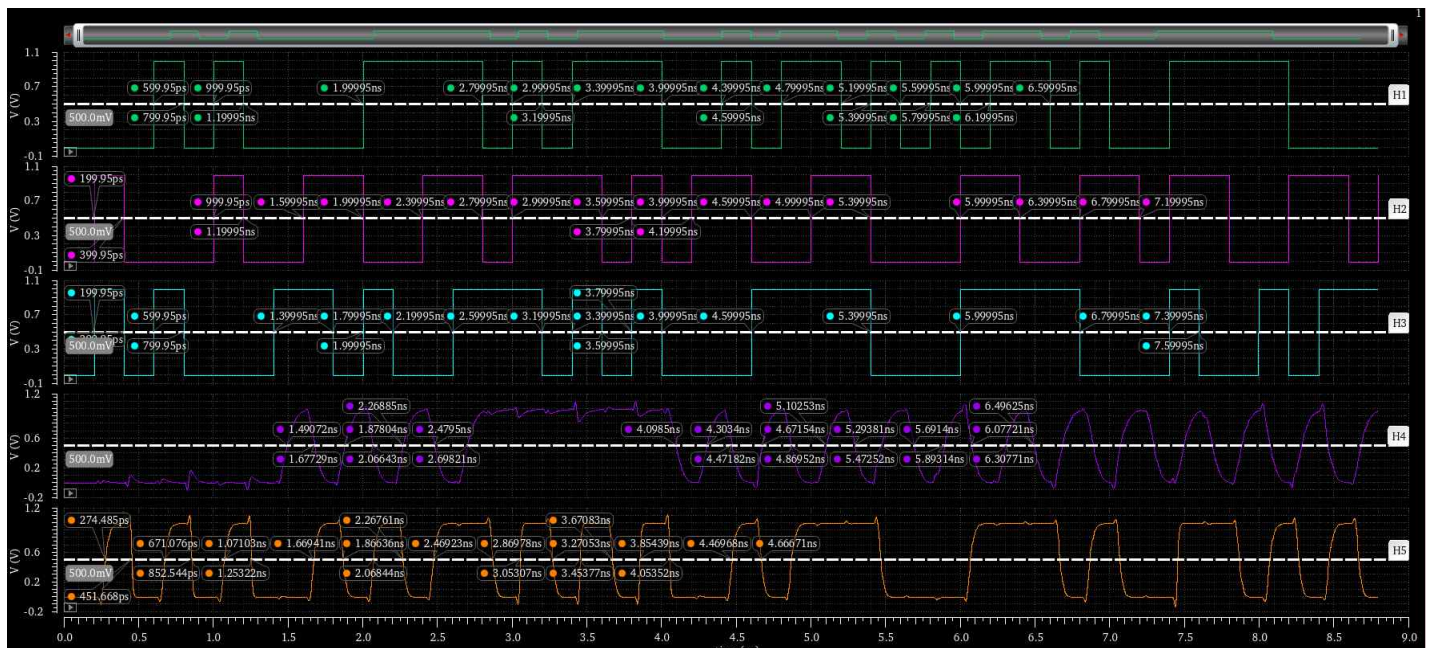


그림2-9) Full Adder Test bench (Width 120nm)

테스트 결과 Full Adder 에 맞는 연산 결과를 나타내고 있다.

하지만, Sum 의 일부분에서 노이즈가 타는 모습을 볼 수 있었다.

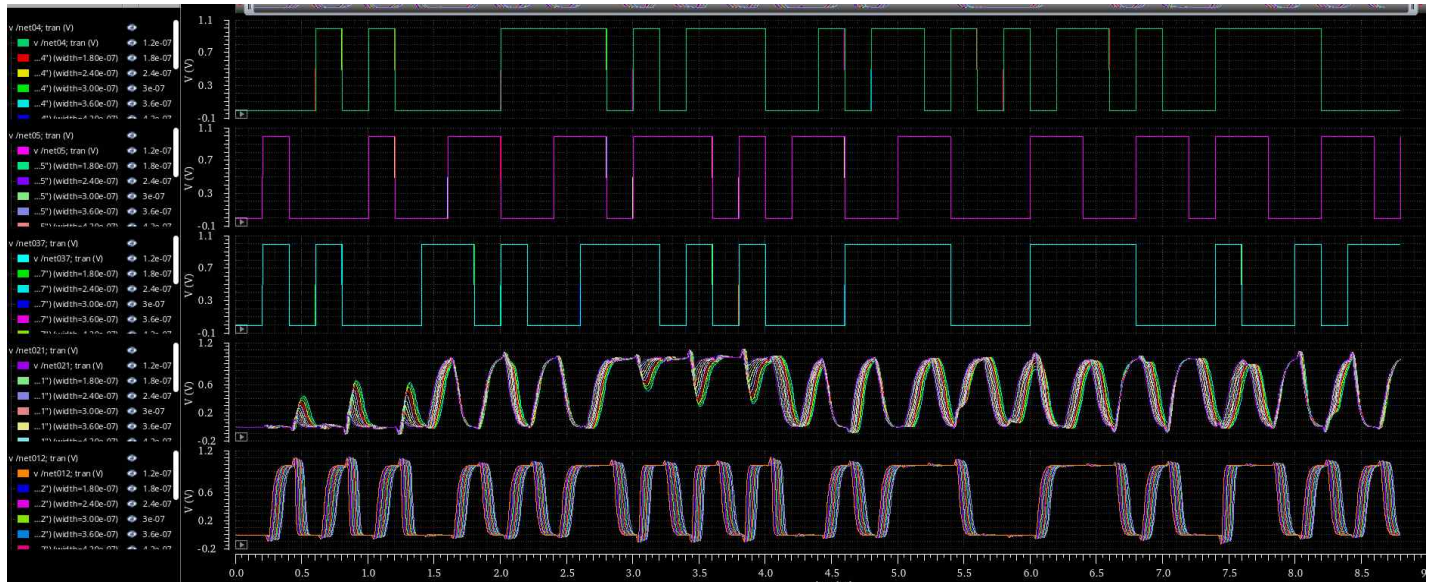
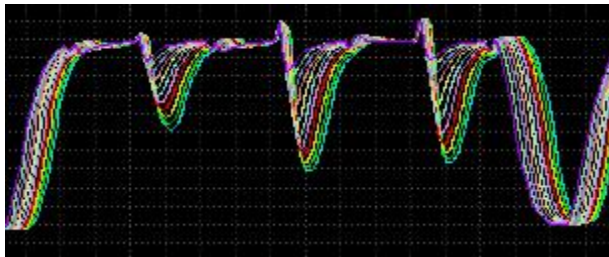
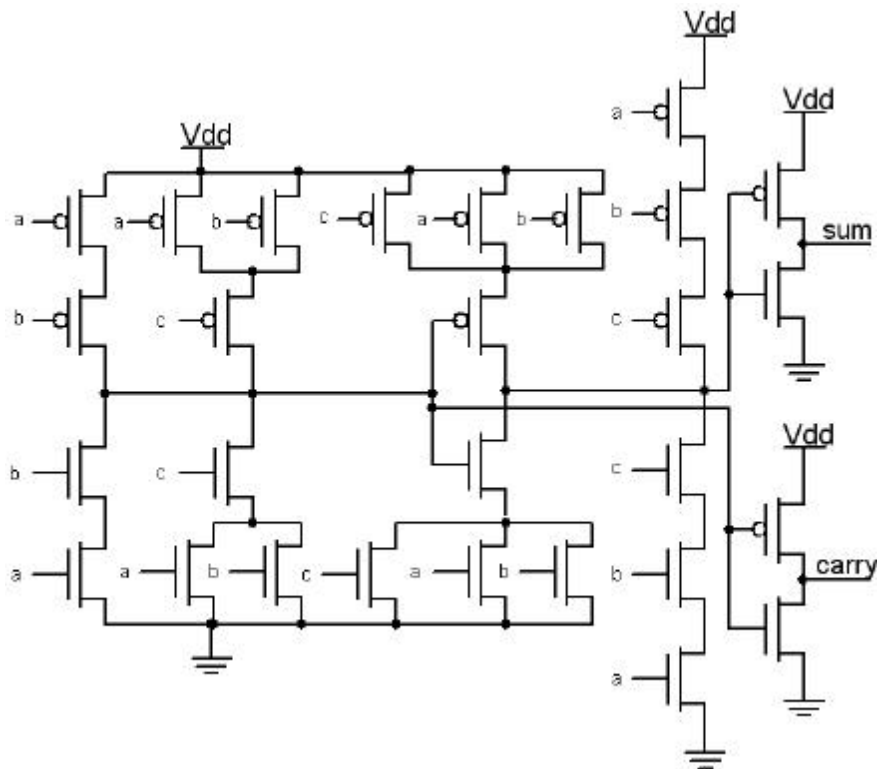


그림2-10) Width 변화에 따른 Full Adder 신호흐름선도



Width 가 증가할수록 delay 시간이 증가하였고, 일부 모양이 변형되는 부분이 있었다.

다음으로 관습적으로 사용하는 Full Adder CMOS 구조를 사용하였다.



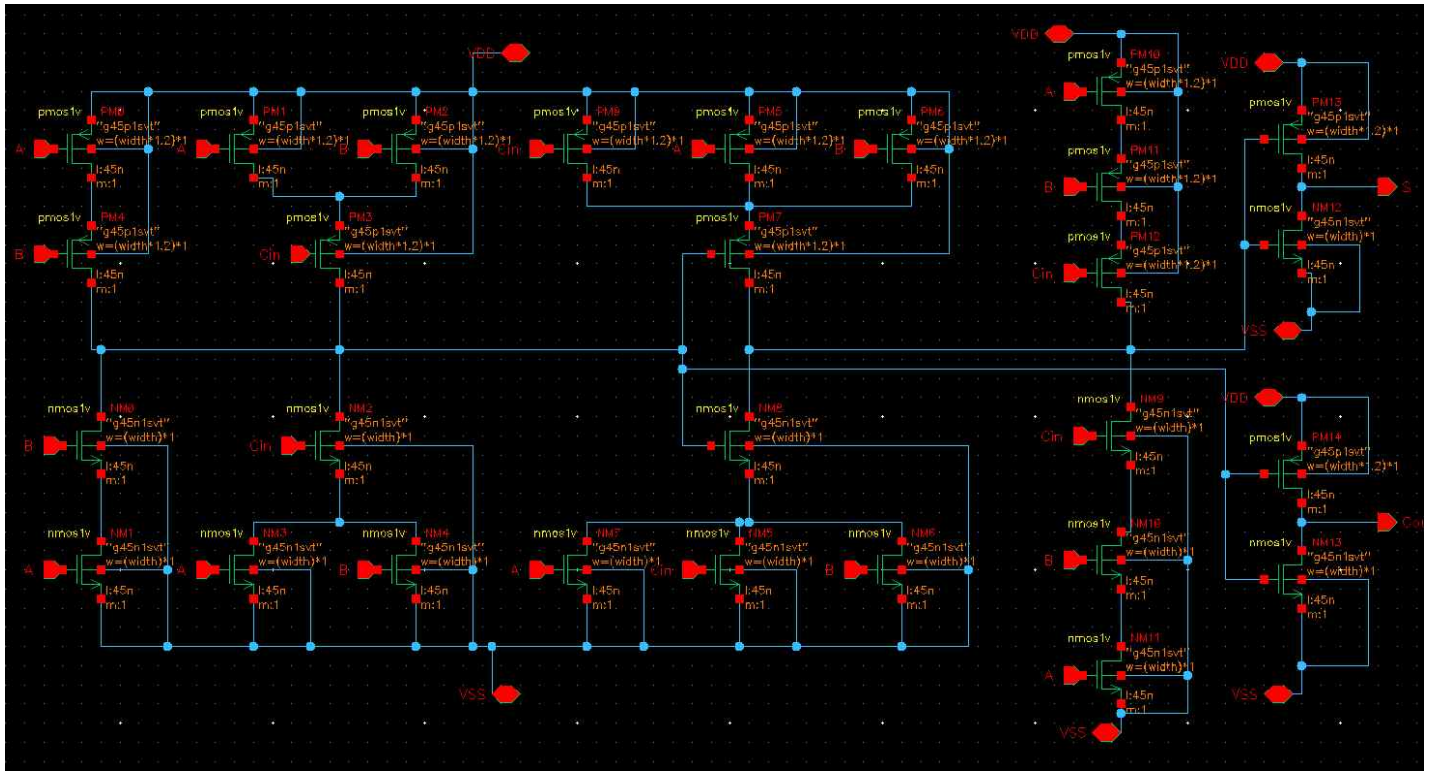


그림2-11) Conventional Full Adder Test bench 회로도

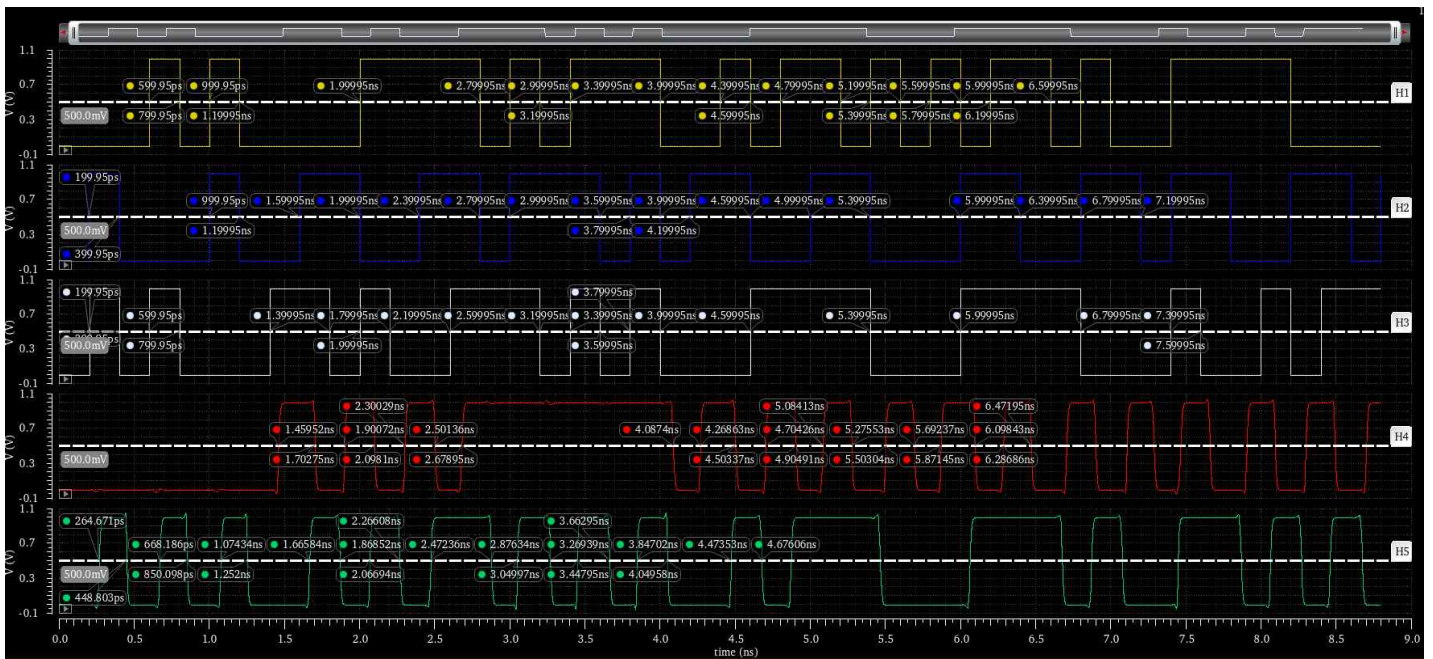


그림2-12) Conventional Full Adder Test bench (Width 120nm)

기존에 설계한 Full Adder 에 비해 노이즈 요소가 많이 줄어들었다.

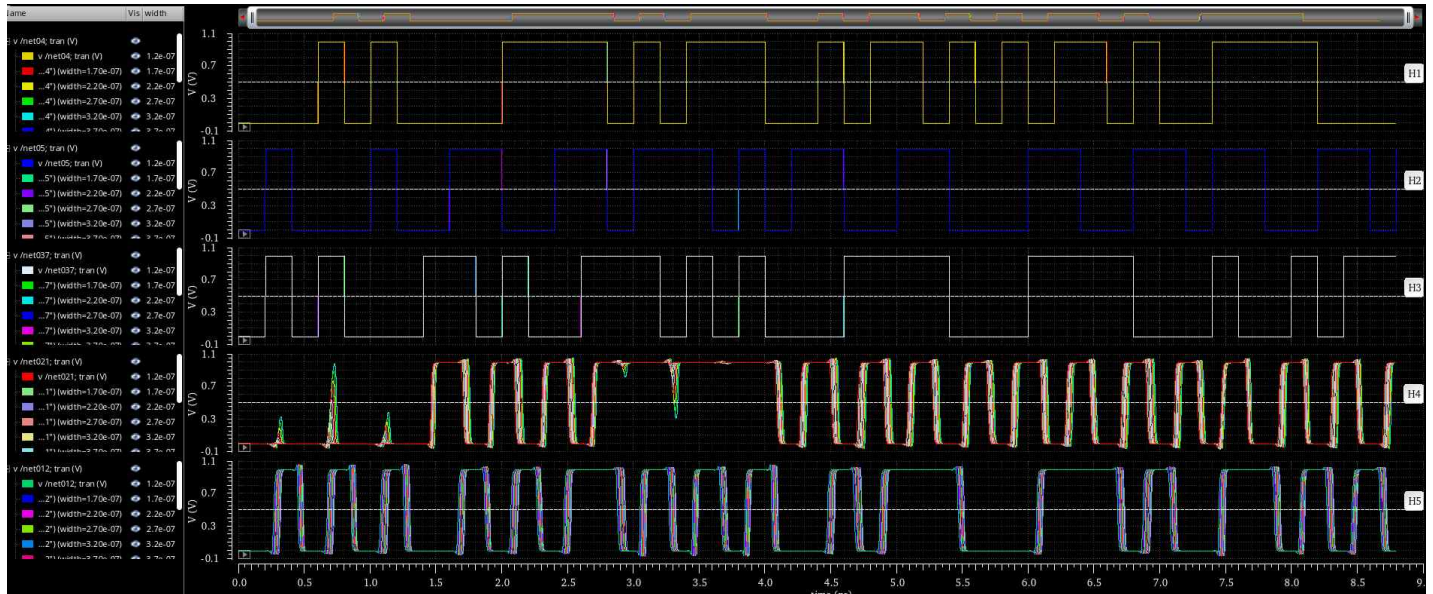


그림2-10) Width 변화에 따른 Conventional Full Adder 신호흐름선도



마찬가지로 Width 가 증가할수록 delay 시간이 증가하였고, 일부 모양이 변형되는 부분이 있었다.

A	B	Cin	Sum	Cout	Sum Delay	Cout Delay
	0.1995ns			0.2646ns		65.17ps
0.59995ns				0.6681ns		68.236ps
0.79995ns				0.85ns		50.148ps
0.99995ns				1.07434ns		74.39ps
1.19995ns				1.252ns		52.05ps
		1.39995ns	1.45952ns		59.57ps	
	1.59995ns		1.70275ns	1.66584ns	102.8ps	65.89ps
		1.79995ns	1.9ns	1.86852ns	100.05ps	68.57ps
1.99995ns			2.0981ns	2.06694ns	98.15ps	66.99ps
		2.19995ns	2.3ns		100.05ps	
	2.39995ns		2.5013ns	2.47236ns	101.35ps	72.41ps
		2.59995ns	2.6789ns		78.95ps	
	2.79995ns			2.87634ns		76.39ps

표2-6) Width 120nm 기준 Delay (단위 : ps)

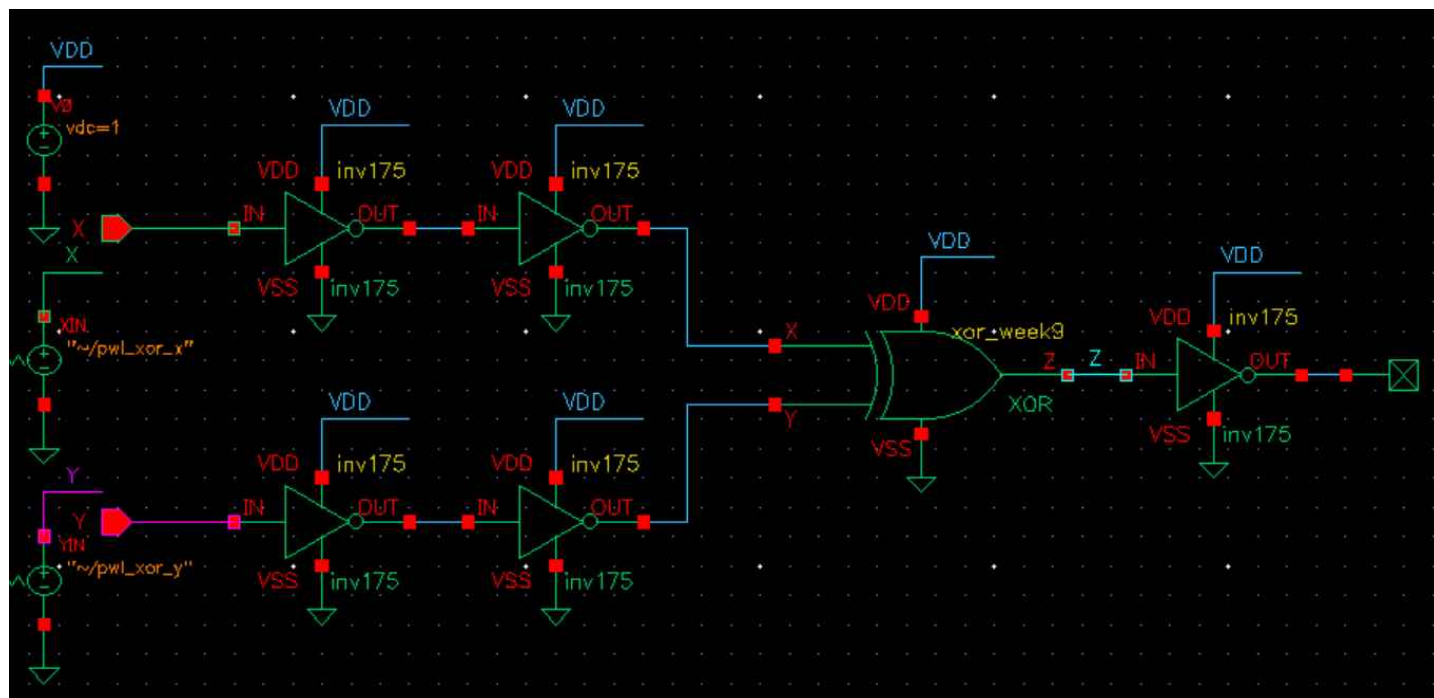
Full Adder 결과의 일부의 Delay를 측정하여 기록한 표 이다.

과제에서 제안된 150ps 이내로 작동되는 모습을 확인할 수 있다.

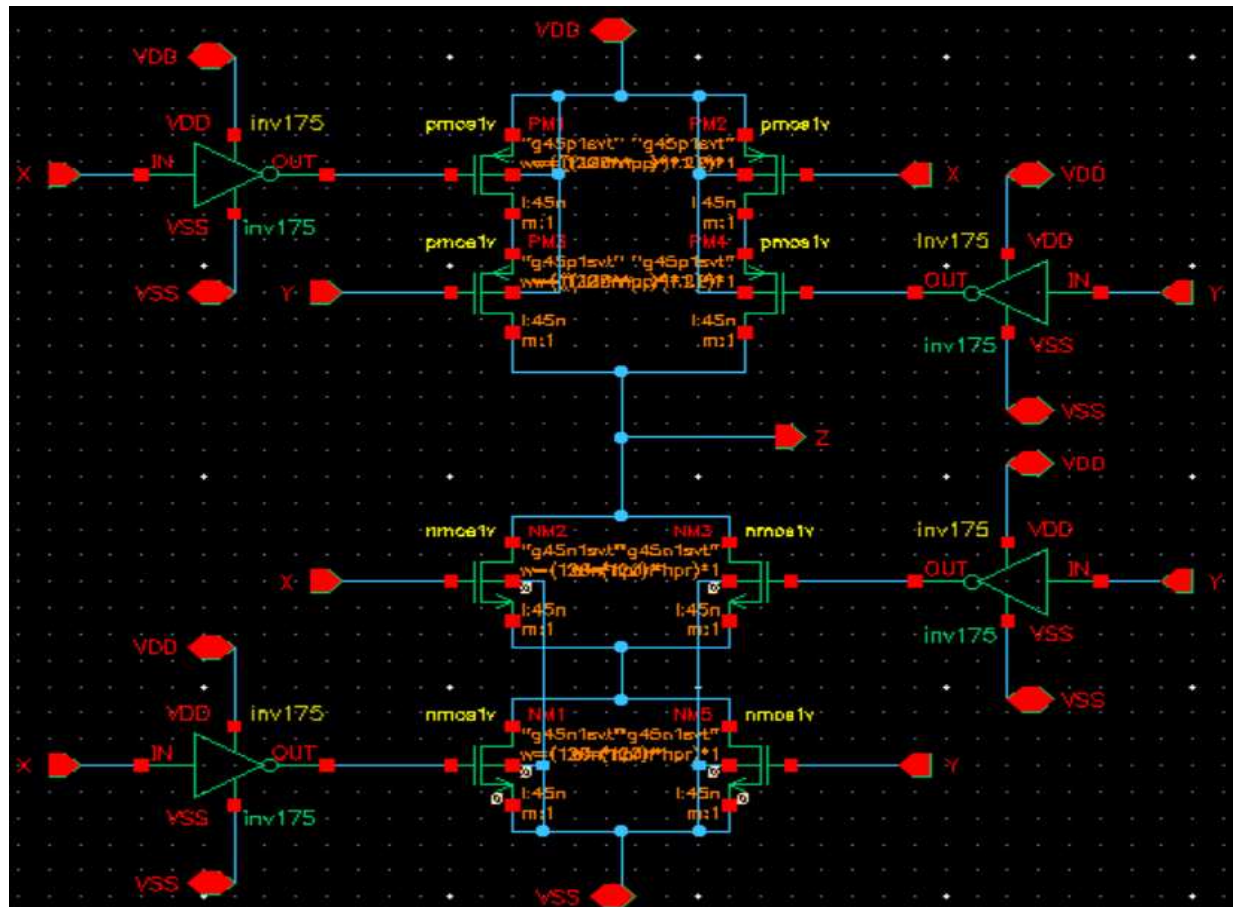
#Design1

1. Schematic

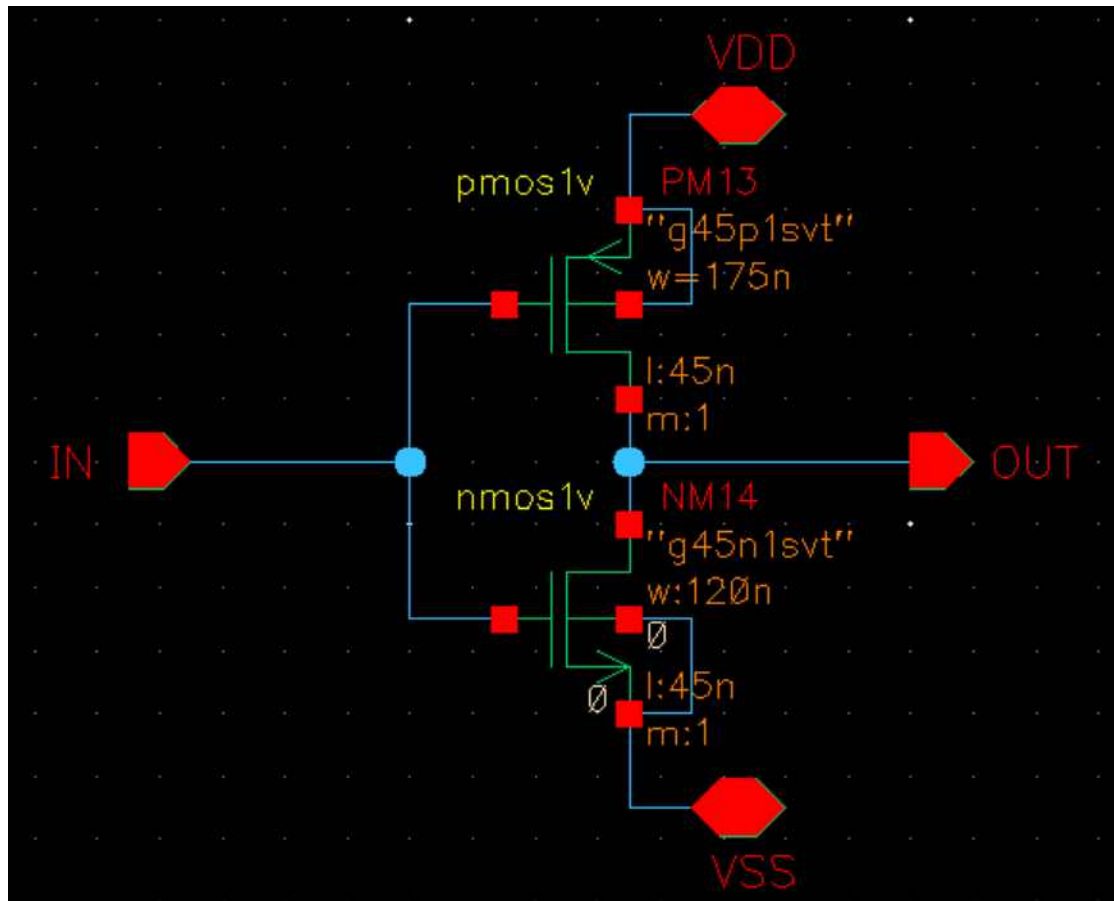
1) 전체 회로



2) XOR Gate ($W_p = 1.2 W_n$, Worst case sizing)



3) Inverter (주어진 설계 값 175nm 사용)

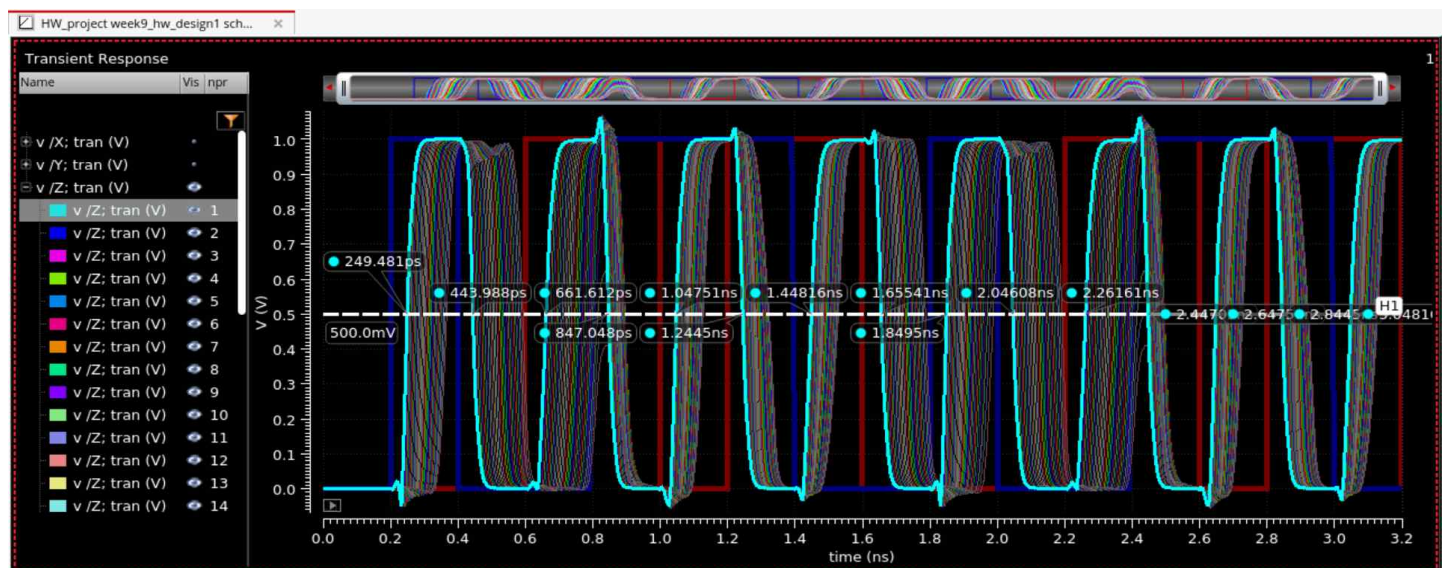


2. Simulation

1) Simulation 설정 ($W_n = 120n * npr$)

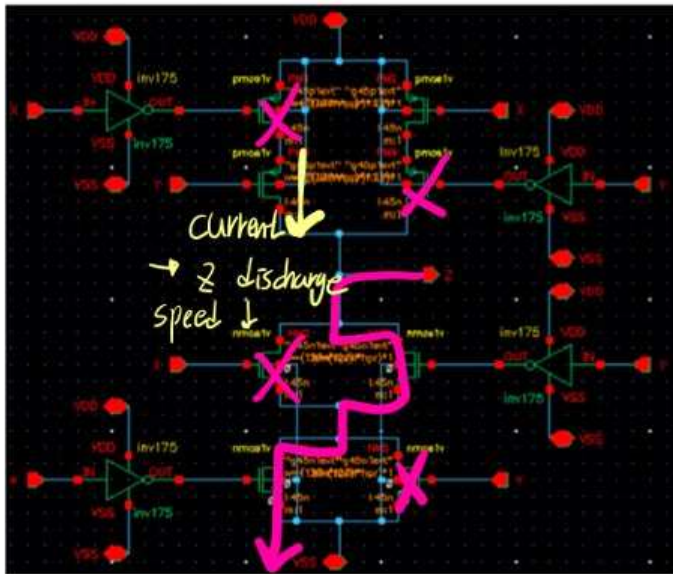
Variable	Value	Sweep?	Range Type	From	To	Step Mode	Total Steps
npr	1	<input checked="" type="checkbox"/>	From/To	1	30	Auto	30

2) $npr = 1$ 일 때, $W_n = 120[nm]$, $W_p = 144[nm]$

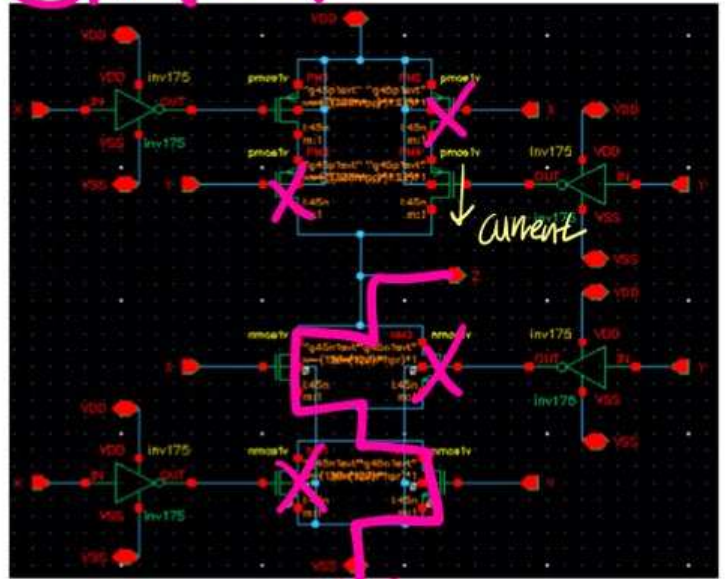


3) 2번 결과를 보면 $(X, Y) = (1, 0)$ 일 때, delay 가장 크게 나타남. ($t=600\text{ps}$ 에서 delay 61ps)
그 이유는 다음과 같다고 생각함.

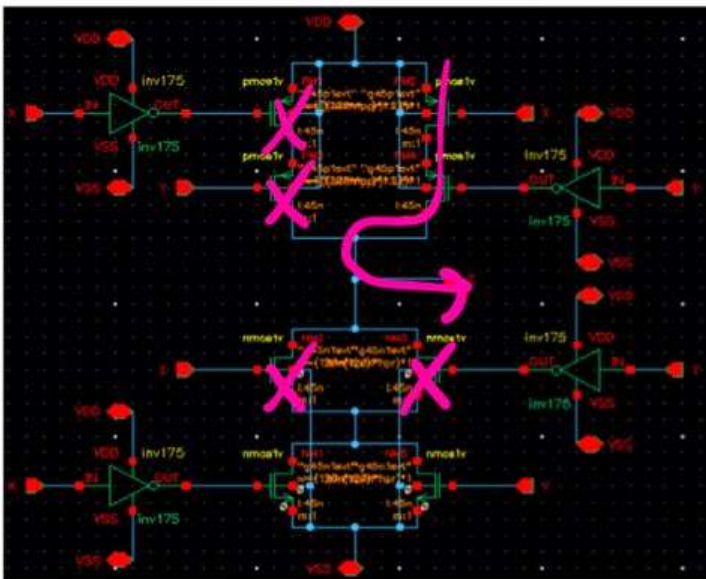
① $XY = 00$



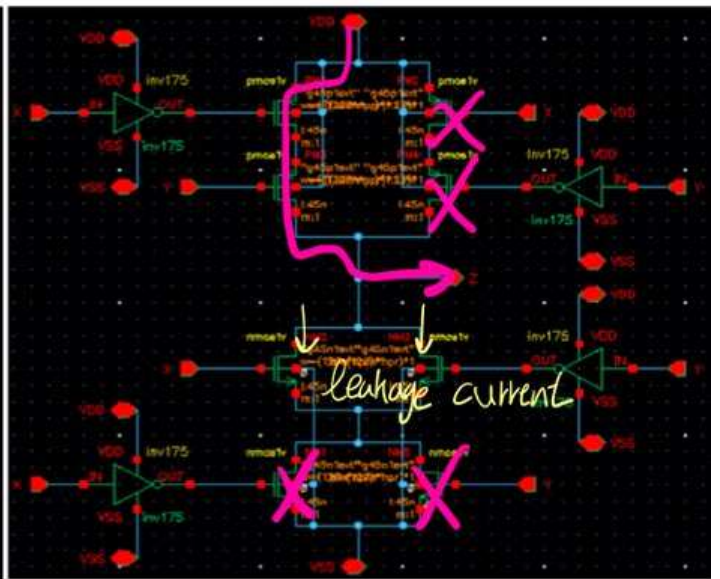
② $XY = 11$



③ $XY = 01$

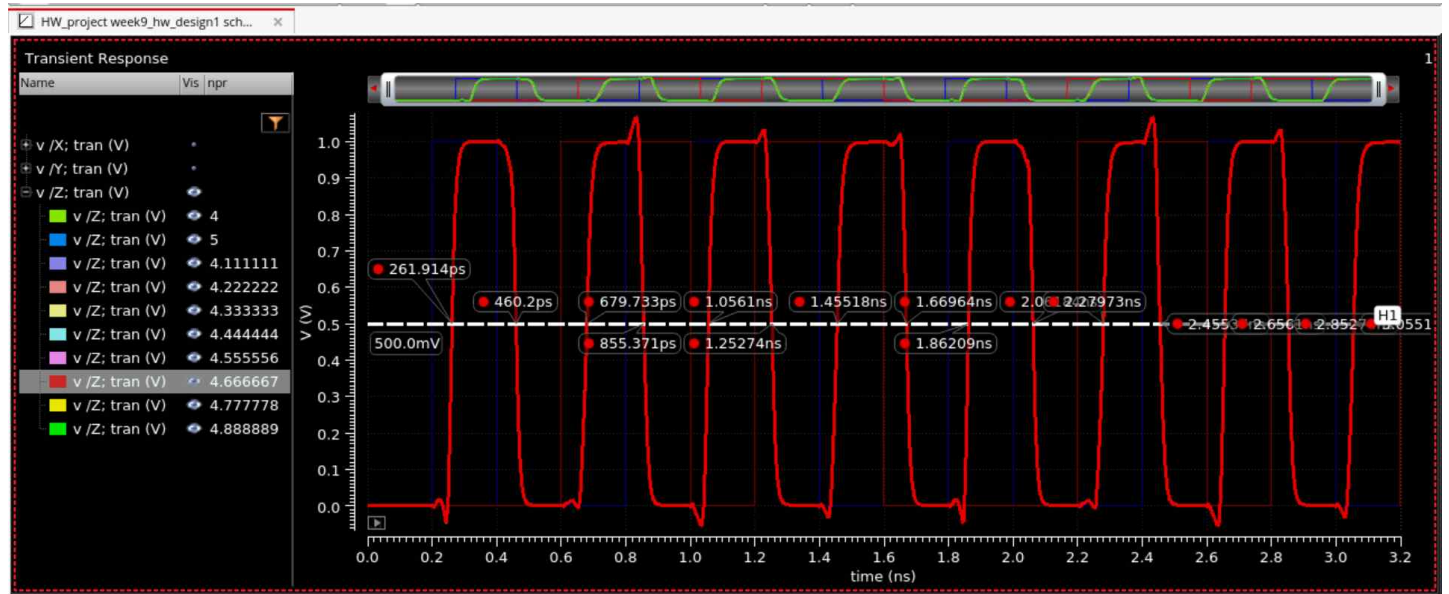


④ $XY = 10$



=> $XY = 10$ 일 때, leakage path가 2군데 존재하기 때문에 delay가 크다고 유추해 볼 수 있음.

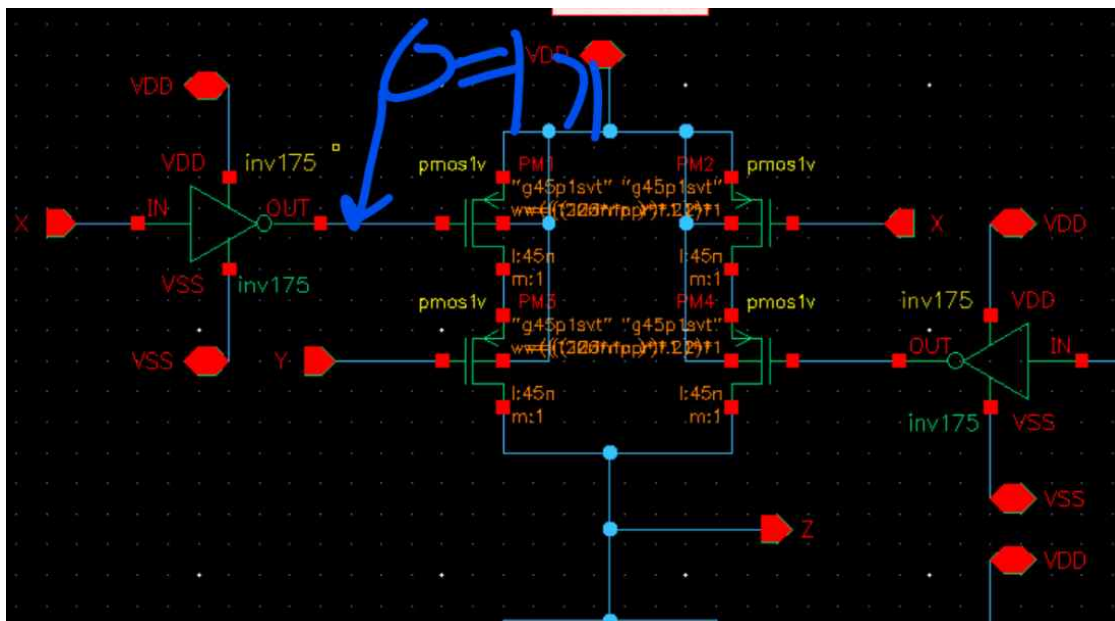
4) 3번 결과를 바탕으로 설계 사양(80ps delay)을 만족 못할 때의 npr 값 찾아보았음.(t=600ps에서 delay 79.73ps)



대충 $W_n = 559.2[nm]$, $W_p = 671.04[nm]$ 일 때, 설계 사양 만족 못함.

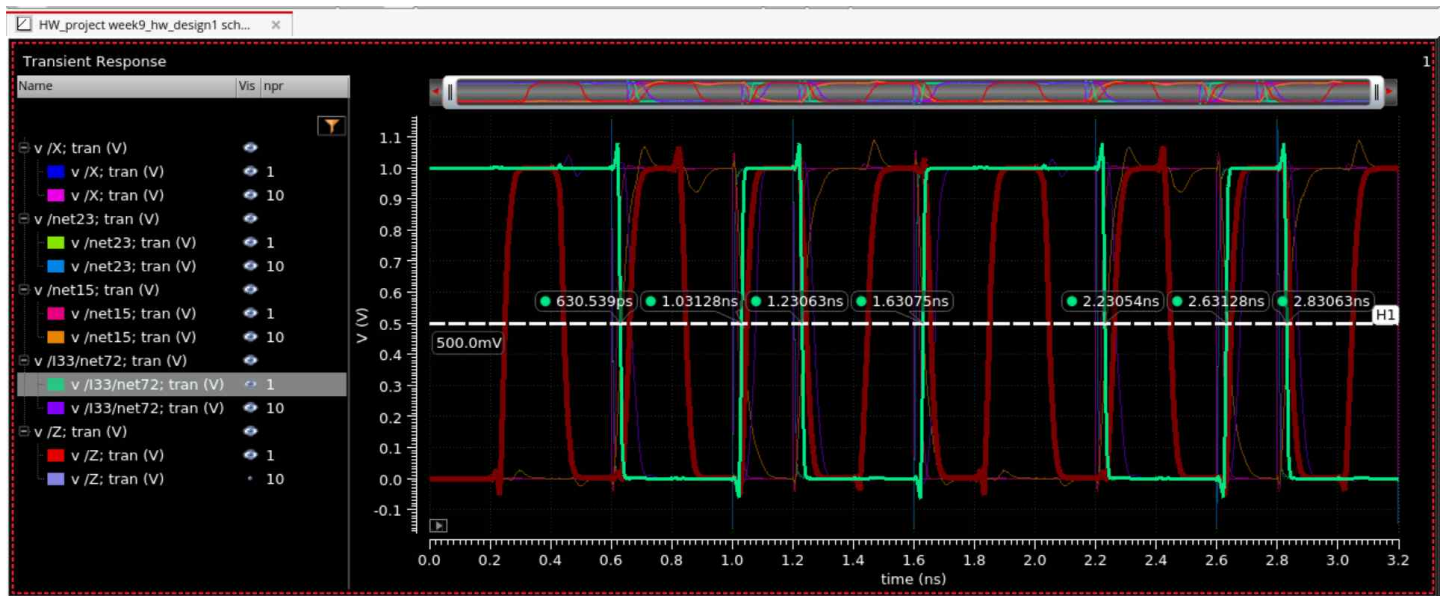
3. 결론 및 고찰

- 왜 Channel width가 커질수록 동작속도는 나빠지는가 ? 일반적으로 Channel width가 커지면 저 많은 전자 또는 정공들이 움직일 수 있기 때문에 동작 속도는 좋아져야하는데?
- => 여러군데 찍어서 확인해보니 채널 폭이 커질수록 175nm inverter에서 delay가 점점 커진다는 것을 확인하였음.
- 지금 밑에 사진에서 “여기” 부분은 X입력이 inverter 3개를 지난(delay를 3번이나 거친) 부분임.



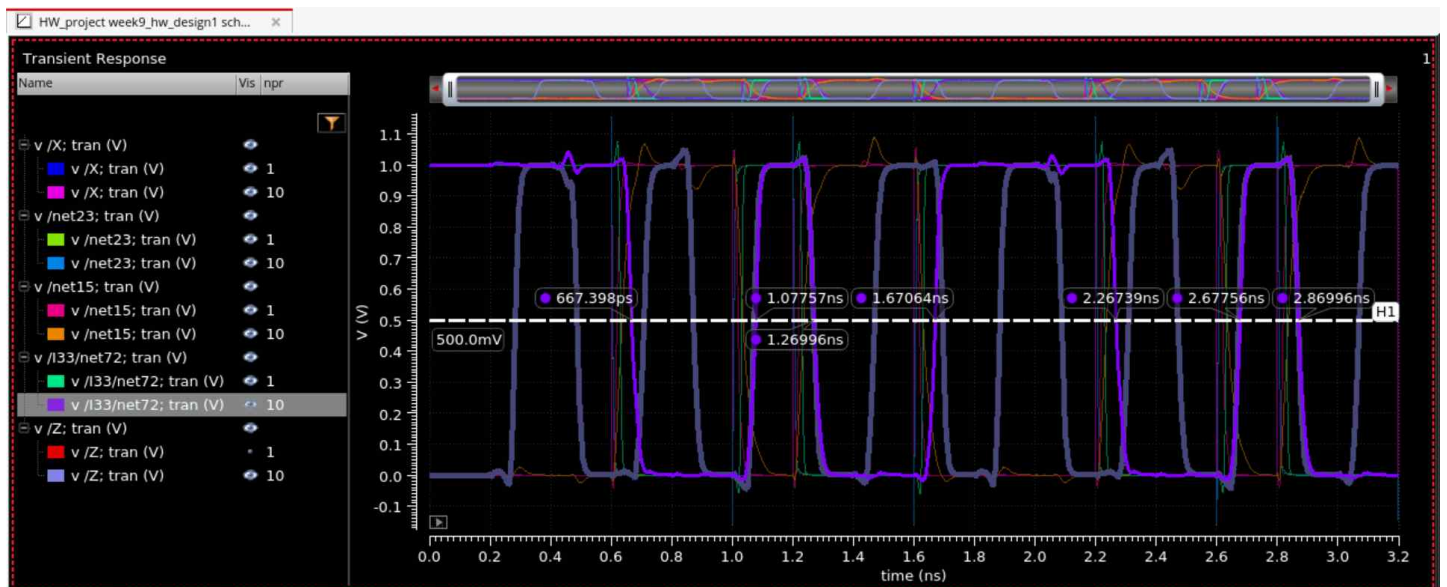
그 때, 밑에 시뮬레이션에서 초록색선이 “여기” node의 전압을 찍은 결과이고, (npr=1일 때) 빨간선은 출력 Z임(초록색 선이 VDD/2 까지 떨어지는데 걸리는시간은 600ps에서 30ps정도 걸림. 즉 원래는 600ps에서 입력 X가 switching 되지만 inverter 3개를 지나기 때문에 30ps delay 이후에 XOR gate로 들어감)

밑의 결과를 보면 “여기” node가 VDD/2 까지 떨어질 때 출력 Z도 RISING하는 것을 볼수있음. 즉 초록색 “여기” 노드가 VDD/2까지 떨어지기 시작해야 출력도 변화가 생김



밑의 결과는 npr=10일 때, 즉 채널폭을 키웠을때의 결과임.

보라색 “여기”노드가 VDD/2까지 떨어지는데 667ps로 아까전보다 delay가 37ps 커진 것을 알 수 있음. 따라서 굵은 소라색으로 표시된 출력 Z가 RISING하기 시작하는 시간 자체가 뒤로 밀린 거 랑 같음.



따라서 채널 폭이 커질수록 delay measurement setup을 위한 inverter 자체에서의 delay가 커져버리고, XOR GATE의 propagation delay도 커지게 됨.

-> 채널폭이 커진다고 XOR GATE의 동작 특성(SPEED)가 나빠지는 것이 아님. 단지 Inverter에

서의 delay가 커져서 XOR GATE가 올바른 입력을 받는 시간 자체가 늦어지는 것뿐임. 근데 그게 XOR gate의 동작 스피드가 안 좋아지는 것처럼 보이는 것뿐임.

그렇다면 왜 xor gate의 npr변수를 늘렸는데(채널폭을 키웠는데), inverter에서의 delay가 커지는 것일까?? (inverter의 size는 120nm 175nm로 정해져있는데 말이지)

=> 여기서부터 뇌피셜인데 xor gate cmos는 ON이 되어 있지 않더라도 여러 가지 이유로 인해 (subthreshold current 등등) leakage current가 분명히 존재함. 그런데 그 leakage current의 에너지원은 VDD일 수 밖에 없음.

따라서 XOR 채널폭이 커질수록 leakage current에 의해 VDD의 potential을 뺏게 되고, 이것은 inverter의 전원 공급원인 VDD의 전압을 낮추는 결과를 초래하게 됨.

=> 따라서 inverter 동작 특성이 나빠지고,(어쩌면 동작 speed가 나빠지고) delay가 증가하는 것임.

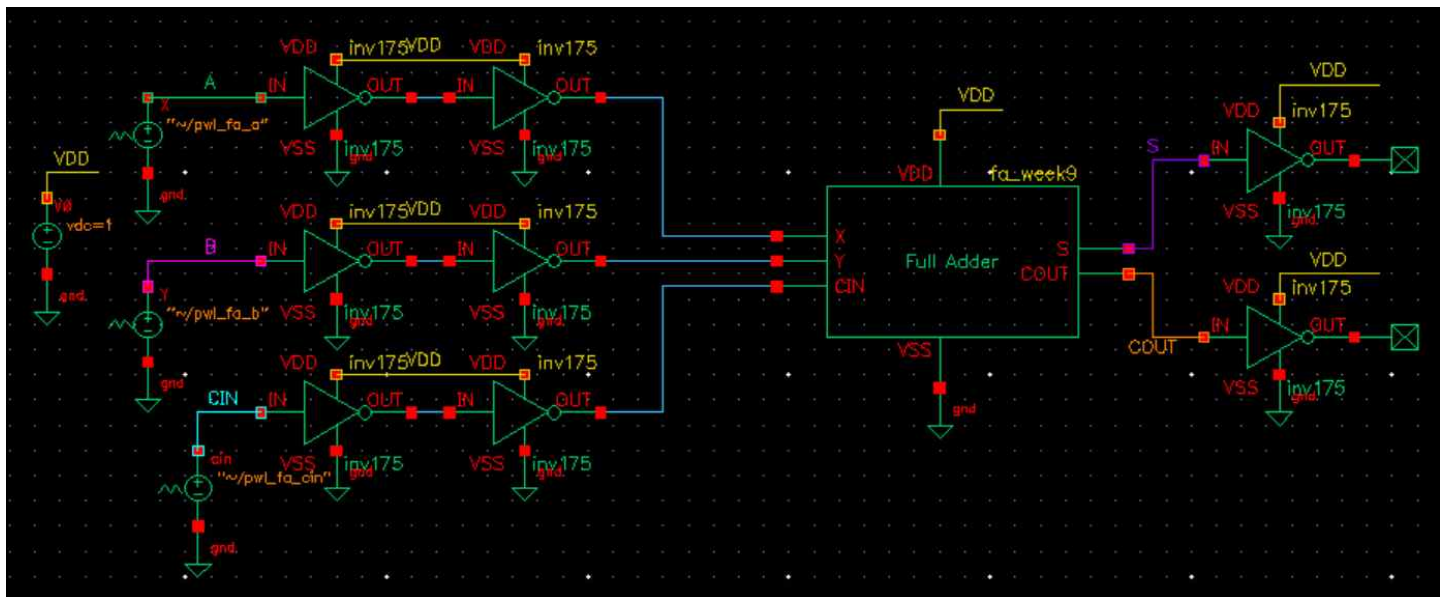
최종적으로 설계목표인 X,Y 변화에 대해 Z가 80ps안으로 들어오게 하는 것은

$W_N = 120[nm]$, $W_P = 144[nm]$ 로 굳이 채널폭을 키워서 TR 면적을 키울 필요가 없다고 생각함.

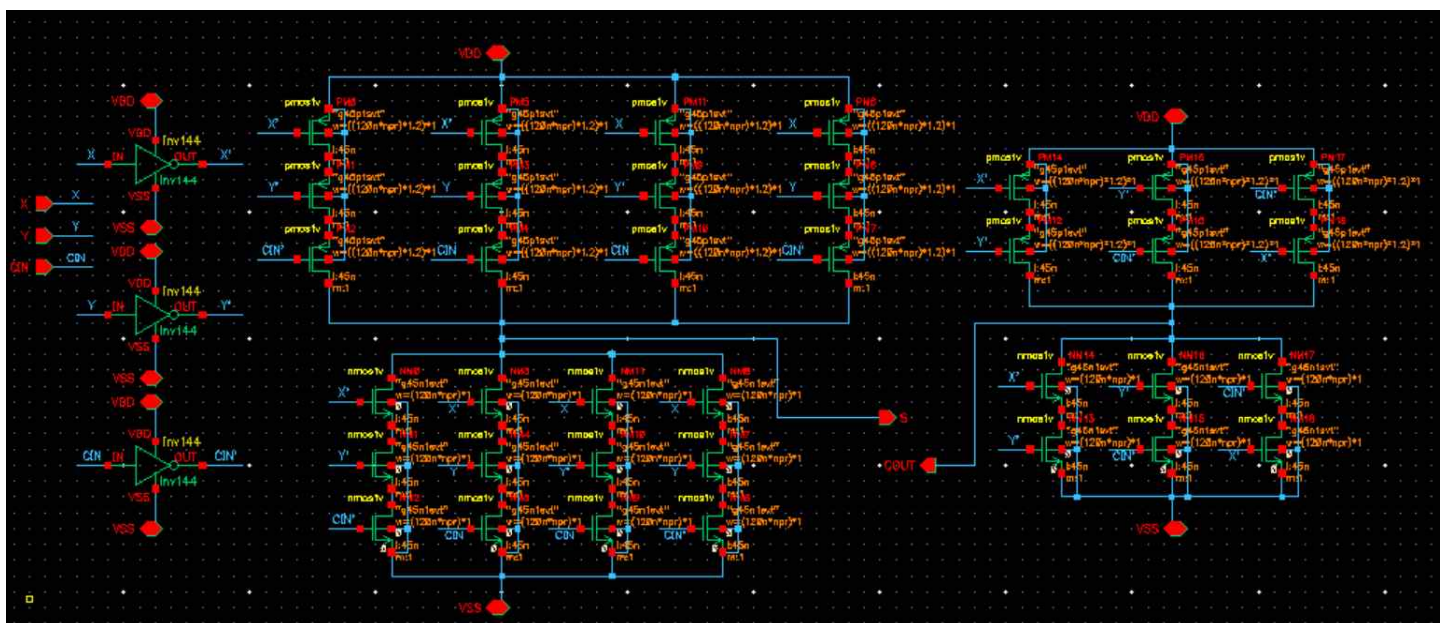
Design2 Full Adder 설계

1. F/A를 설계를 하기 위해 입력 x,y,cin / 출력 s,cout 에 대한 진리표를 구성함.
2. 진리표를 토대로 카노맵을 작성하여 출력식 s, cout을 최대한 간략화 시킴
-> 사용되는 게이트의 수 감소 -> 단가 절감
3. 간략화된 출력식 S, COUT에 대하여 cmos static logic 구성
-> PUN, PDN에 대하여 각각 따로 구성
4. worst case sizing -> 출력 S와 COUT에 대하여 $W_P = 1.2 W_N$ 으로 사이징하였음. 150ps의 설계사양을 맞추기위하여 $W_N = 120n * npr$ 으로 설정하였음. 만약 설계 사양을 만족 못 한다면 npr을 조정해볼 생각임.

스케메틱 결과는 다음과 같음.



<전체 회로>



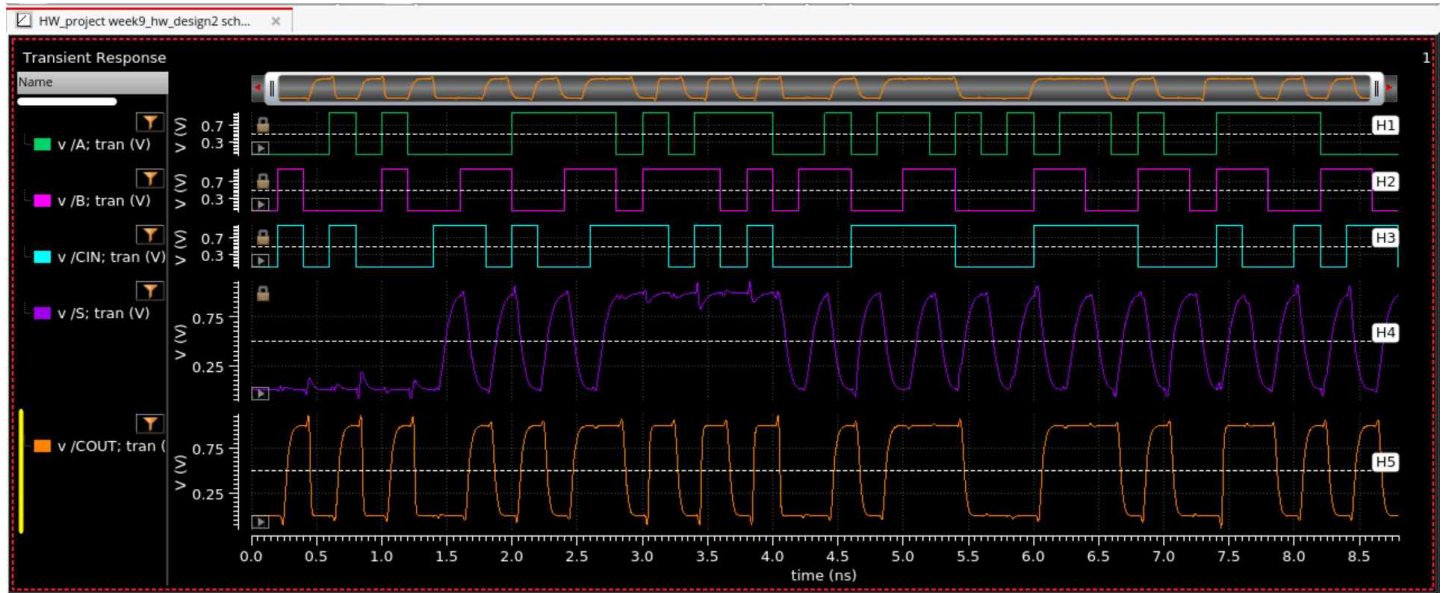
<Full Adder>

제일 왼쪽은 cell 입출력 및 전원공급원 / 중간은 S / 오른쪽 COUT임.

고찰1 게이트 사용 최소화를 위해 X',Y',CIN'에 대하여 굳이 inverter 다시 사용 할 필요 없음.

고찰2 F/A 내부 입력반전을 위한 Inverter는 $W_p = 175[nm]$ 로 사이징 할 필요 없음. 175nm는 단지 지연시간 측정장치에서 사용하는 inverter의 size를 말함. 따라서 f/a chip의 동작속도를 위해 내부 inverter는 best performance를 내도록 $W_p = 144nm$ 로 사이징하였음.

다음은 시뮬레이션 결과임.



위에서부터 입력 X, Y, CIN / 출력 S, COUT임.

0~8.8ns의 tran 해석에서 모든 시간에 대해 f/a가 정상동작한 것을 확인하였음.(진리표대로)

그리고 모든 시간에 대해 설계사양 (X,Y,CIN 스위칭에 대해 출력 S,COUT의 delay가 150ps 보다 작아야함.)을 만족하였음. 가장 큰 지연시간은 약 $106[ps]$ 정도로 측정되었음.

고찰3



t=7.4ns에서 입력 (x,y,cin)=(1,1,1)이 되었음. 이 때 당연히 sum과 carry out은 (1,1)이 되야함. 근데 사진에서 보다시피 sum의 지연시간은 약 95ps이고, carry out의 지연시간은 약 54ps임.

즉 둘다 이전 logic state가 0이었는데 logic threshold voltage vdd/2까지 올라가는데 걸리는 시간이 달랐음.

=>이것은 당연한 결과임. sum의 PUN은 TR 3개가 직렬로 연결되어 있고, carry out의 PUN은 TR 2개가 직렬연결 되어 있음. 따라서 sum에서의 채널길이가

carry out에서의 채널길이의 1.5배가 되는 것임. 그래서 당연히 동작속도가 느리고, 이를 시뮬레이션 7.4 ns에서 확인할 수 있었음.

결론

굳이 npr(NMOS 채널폭 파라미터)을 키우지 않아도 설계사양을 만족하므로 굳이 채널폭을 키워서 chip 면적을 잡아 먹을 필요가 없으므로 $W_N = 120[nm]$ / $W_P = 144[nm]$ 로 설계하였음.

고찰4 F/A 칩 내부에는 두 개의 출력 S, COUT이 존재함. 즉 한 개의 칩안에 두 개의 다른 기능을 내는 소자가 존재한다고 볼 수 있음. 하지만 F/A 칩을 만들때는 한꺼번에 공정을 진행하므로 S와 COUT의 채널폭은 같게 사이징하는 것이 공정 입장에서 유리하다고 생각됨. 그래서 S와 COUT의 모든 PMOS 및 NMOS의 사이즈는 같게 설계하였음.