

REPORT

[1bit ALU]



숭실대학교

• 과 목 명 : 아날로그 및 레이아웃 설계

• 담 당 교 수 : 이주석

• 제 출 일 : 2025 년 6 월 19 일

• 학 과 : 전자공학과

• 학 번 : 20202840

• 성 명 : 한정호

목 차

1. 서론	2
2. 본론	3
2.1. Inverter	3
2.2. AND GATE	4
2.3 OR GATE	6
2.4 XOR GATE	7
2.5 2x1 MUX	9
2.6 4x1 MUX	11
2.7 Full Adder	13
2.8 Logic Module	15
2.8 1bit ALU	17
2.9 4bit ALU [추가 응용]	20
3. 결론	20

1. 서론

본 강의를 통하여 Inverter, NOR Gate, D Flip-Flop 와 같은 기본 디지털 회로부터 Current Mirror, CS Amp, 5TR OP amp, 2 Stage OP 에 이르는 회로를 설계 검증하며 아날로그 회로 지식 및 Virtuoso 활용 역량을 쌓아왔다. 이제는 이러한 역량을 기반으로 컴퓨터 중앙 처리 장치에서 연산을 수행하는 필수 블록인 "산술 논리 장치[ALU, Arithmetic Logic Unit]"를 설계하고자 한다.

본 프로젝트는 CMOS Transistor 기반 1bit ALU 설계를 목표로 하며 Virtuoso Tool 를 활용하여 회로도 설계 및 시뮬레이션을 통해 기능적 동작을 검증하고자 한다.

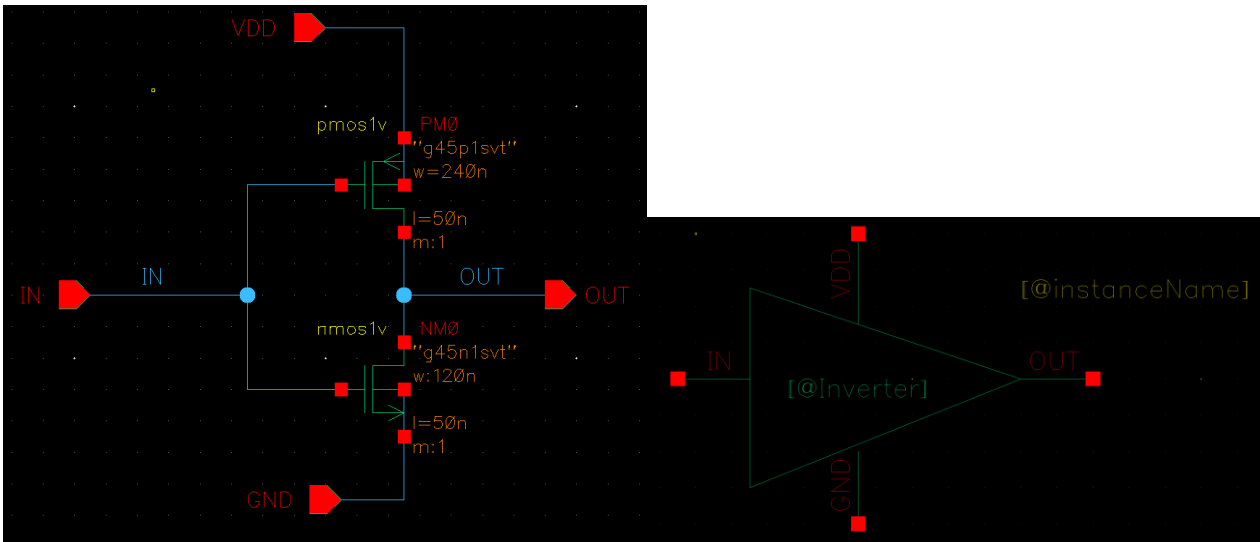
설계하고자 하는 ALU 의 기능은 다음과 같다.

S2	S1	S0	Operations	Function
0	0	0	$G = A - 1$	DECREMENT
0	0	1	$G = A + B$	ADDITION
0	1	0	$G = A + \overline{B} + 1$	SUBTRACTION
0	1	1	$G = A + 1$	INCREMENT
1	0	0	$G = A \wedge B$	AND
1	0	1	$G = A \oplus B$	XOR
1	1	0	$G = \overline{A}$	XNOR
1	1	1	$G = A \vee B$	OR

2. 본론

2.1. Inverter

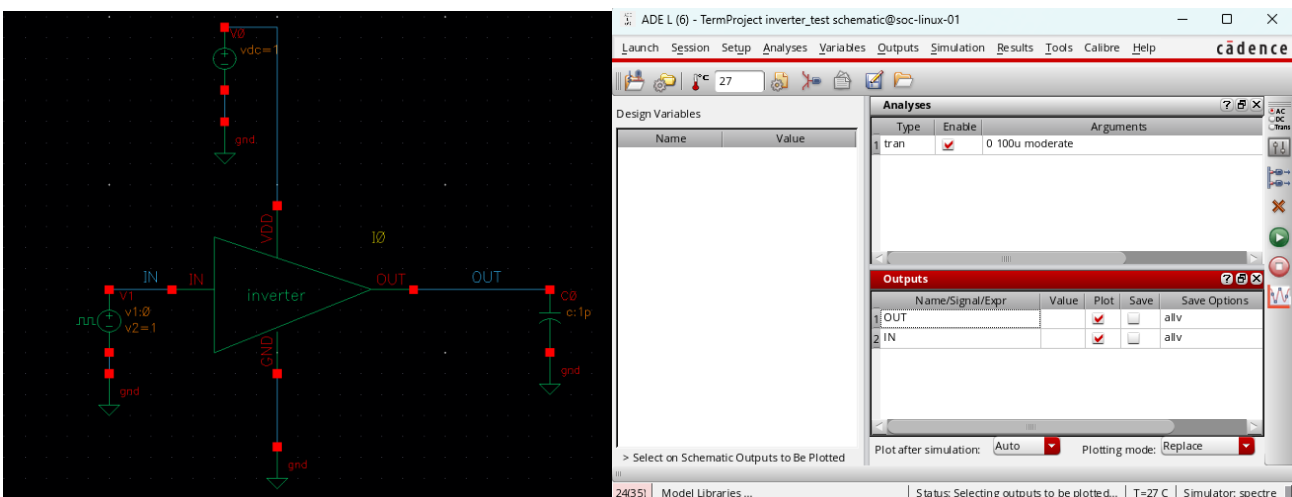
A. 회로



이 회로는 PMOS와 NMOS 트랜지스터를 결합한 기본적인 CMOS 인버터 Inverter입니다. 입력신호가 High 일시, NMOS가 켜져 출력으로 Low, 입력 신호가 Low 일시, PMOS가 켜져 High 을 출력합니다. 결과적으로 이 회로는 입력된 디지털 논리 신호를 반전시키는 역할을 합니다.

정공과 전자의 이동성 차이로 인해 PMOS의 W 값을 NMOS의 2 배로 설정하였으며, 주어진 라이브러리에서 구현 가능한 공정의 최소길이로써 50nm 을 설정하였습니다.

B. 시뮬레이션 설정

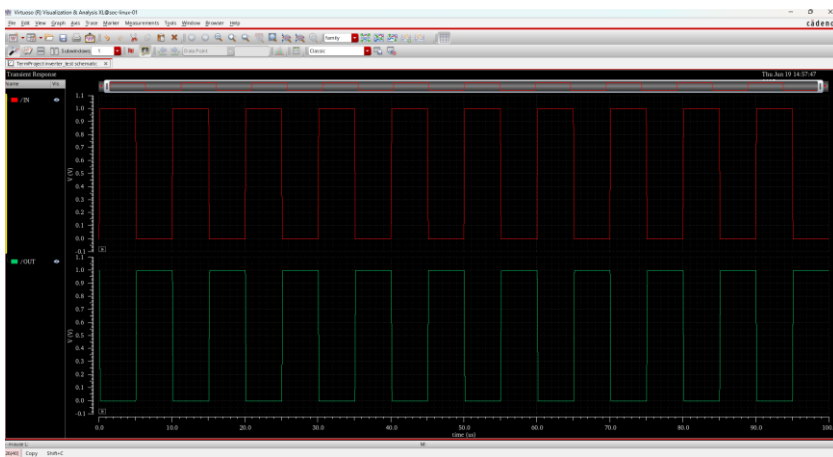


펄스 신호에 대한 인버터의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 100 μ s, Moderate

- Outputs: IN, OUT 의 Waveform

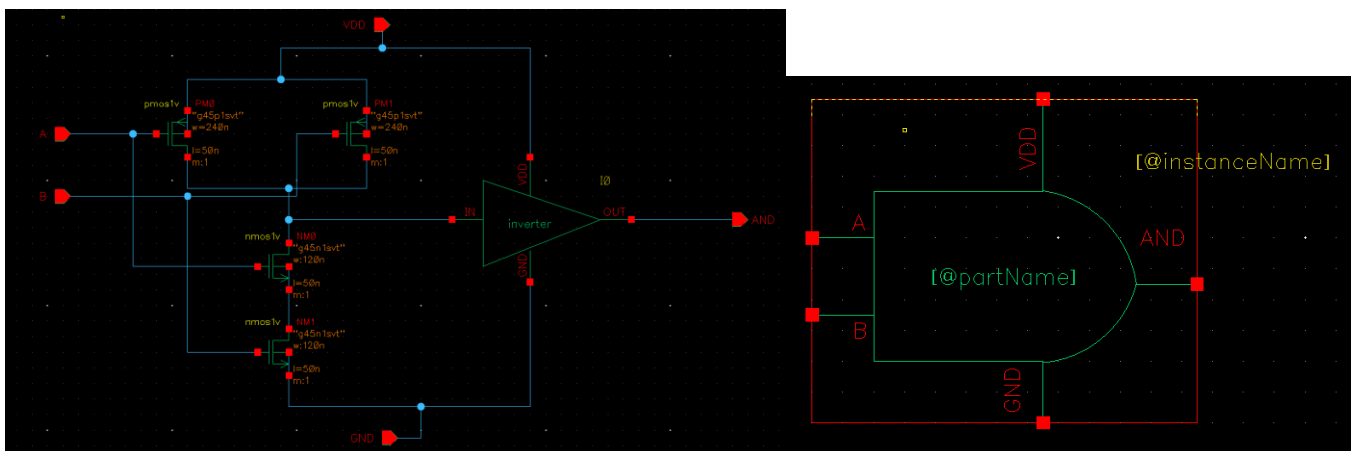
C. 시뮬레이션 결과



입력 신호와 정확히 반전된 출력 신호가 나타나 인버터가 의도한대로 동작함을 알 수 있습니다.

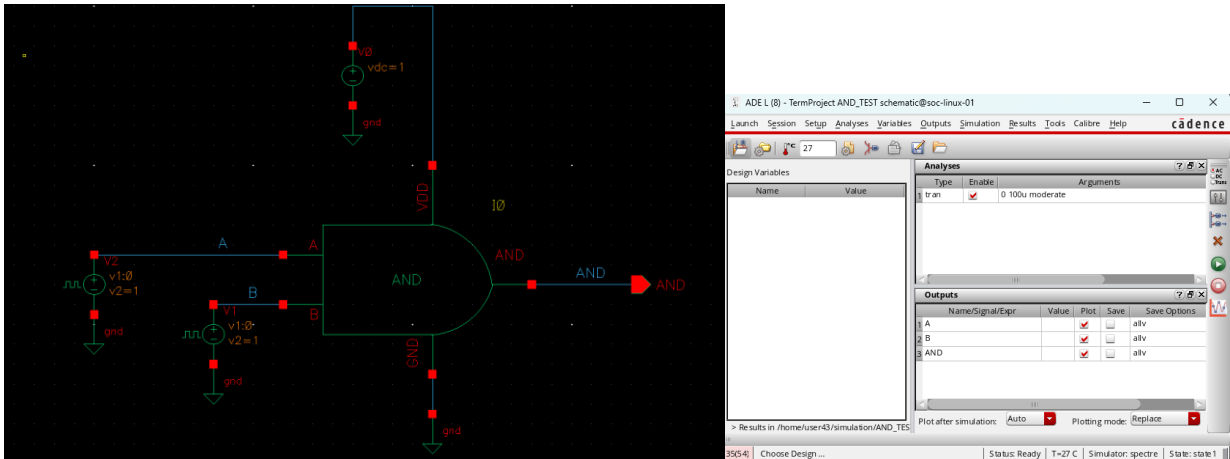
2.2. AND GATE

A. 회로



2Input NAND Gate 와 Inverter 를 직렬로 결합하여 AND Gate 를 구현하였습니다. 앞부분은 A, B 입력을 받는 NAND Gate 이며, 뒷부분의 Inverter 는 NAND Gate 의 출력을 반전시킵니다. 이를 통해 최종적으로 AND Gate 의 논리 연산 결과를 얻을 수 있습니다.

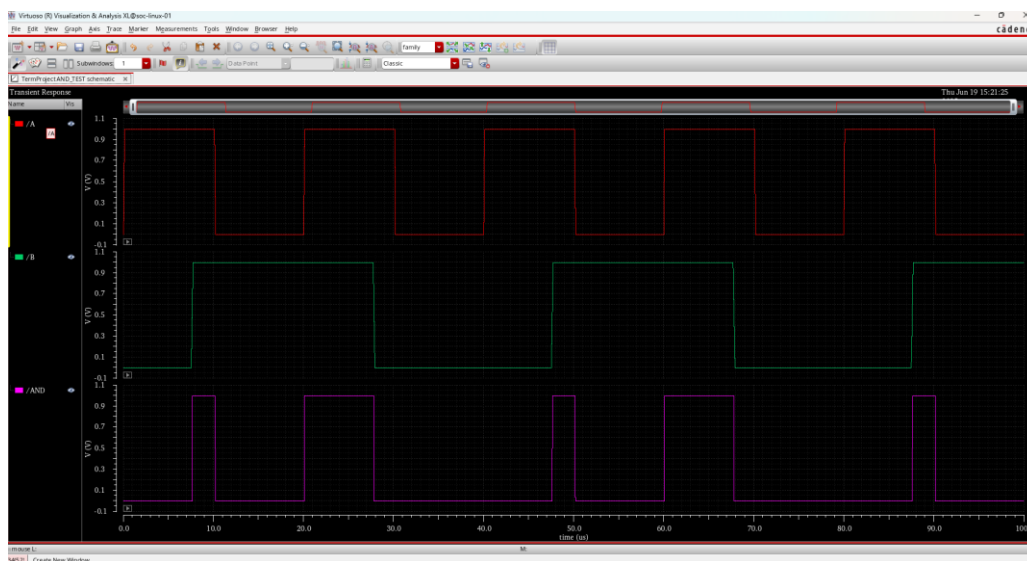
B. 시뮬레이션 설정



펄스 신호에 대한 And Gate 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 100 μ , Moderate
- Outputs: A, B, OUT 의 Waveform

C. 시뮬레이션 결과

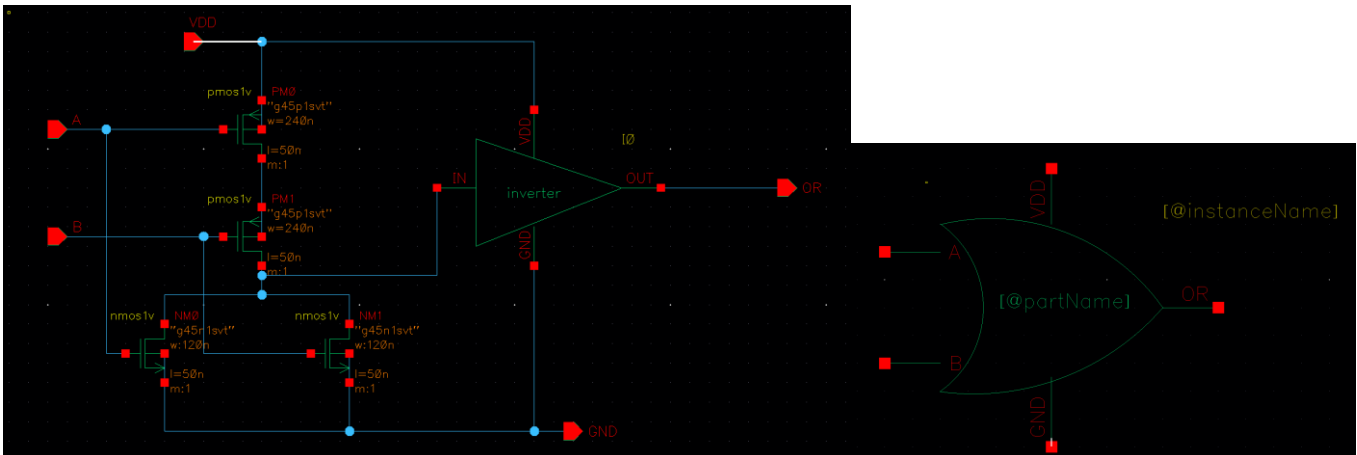


입력		출력
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

시뮬레이션 결과가 AND gate 진리표와 동일합니다. 이를 통해 회로가 의도한대로 설계되었음을 확인할 수 있습니다.

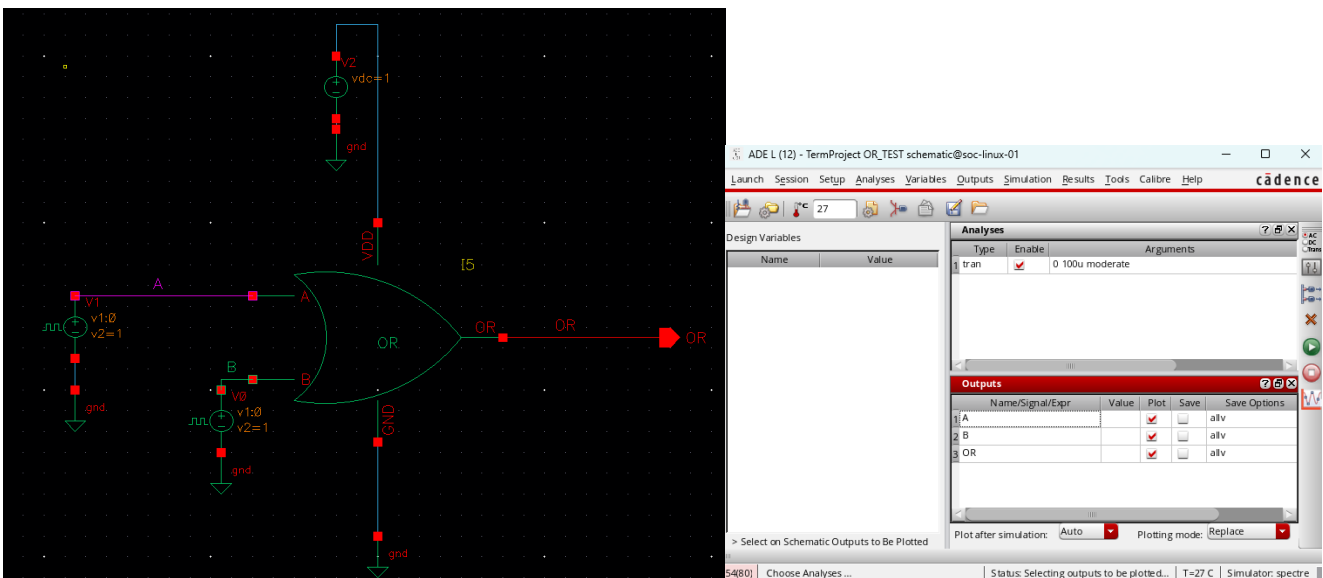
2.3 OR GATE

A. 회로



2Input NOR Gate 와 Inverter 를 직렬로 결합하여 OR Gate 를 구현하였습니다. 앞부분은 A, B 입력을 받는 NOR Gate 이며, 뒷부분의 Inverter 는 OR Gate 의 출력을 반전시킵니다. 이를 통해 최종적으로 OR Gate 의 논리 연산 결과를 얻을 수 있습니다.

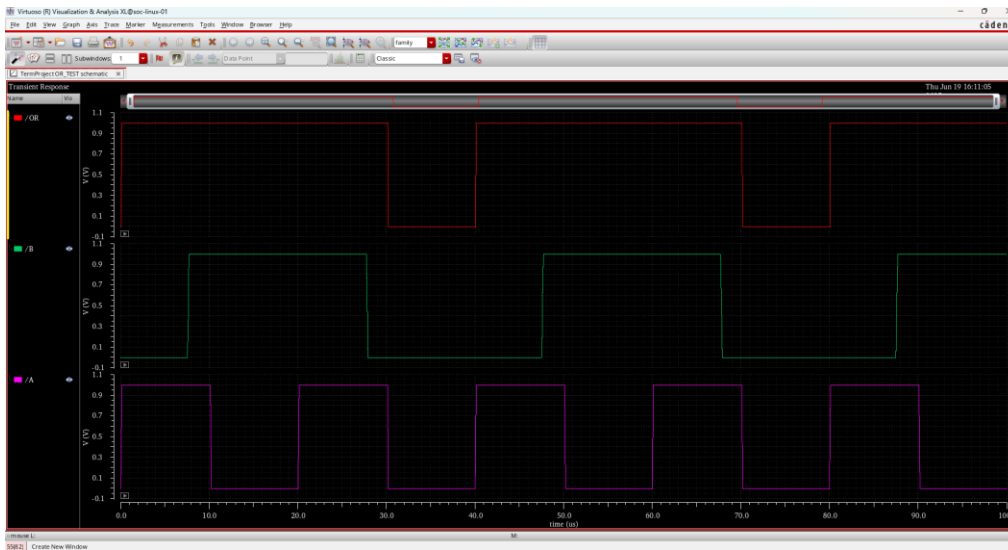
B. 시뮬레이션 설정



펄스 신호에 대한 OR Gate 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 100 μ , Moderate
- Outputs: A, B, OUT 의 Waveform

C. 시뮬레이션 결과

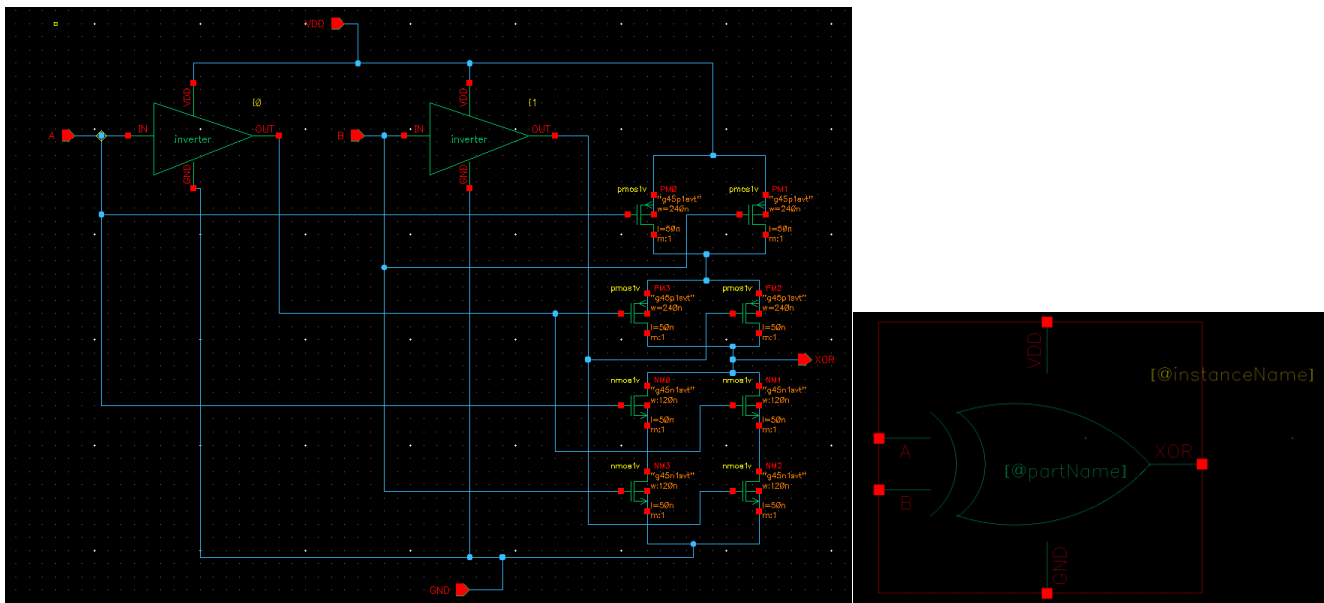


입력		출력
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

시뮬레이션 결과가 OR gate 진리표와 동일합니다. 이를 통해 회로가 의도한대로 설계되었음을 확인할 수 있습니다.

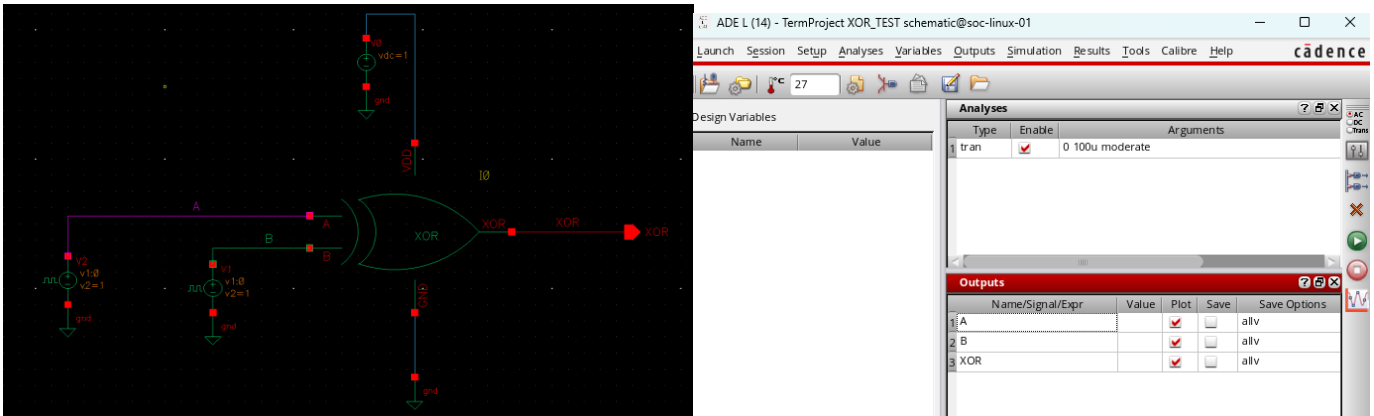
2.4 XOR GATE

A. 회로



Inverter 를 활용한 반전 입력, PMOS 를 활용한 Pull-up Network, NMOS 를 활용한 Pull-down Network 로 구성된 2Input XOR Gate 입니다. 입력이 서로 다를 때는 4 개의 PMOS 로 구성된 Pull-up Network 가 활성화되어 출력을 'High'로 만들고, 입력이 같을 때는 4 개의 NMOS 로 구성된 Pull-down Network 가 활성화되어 출력을 'Low'로 만듭니다.

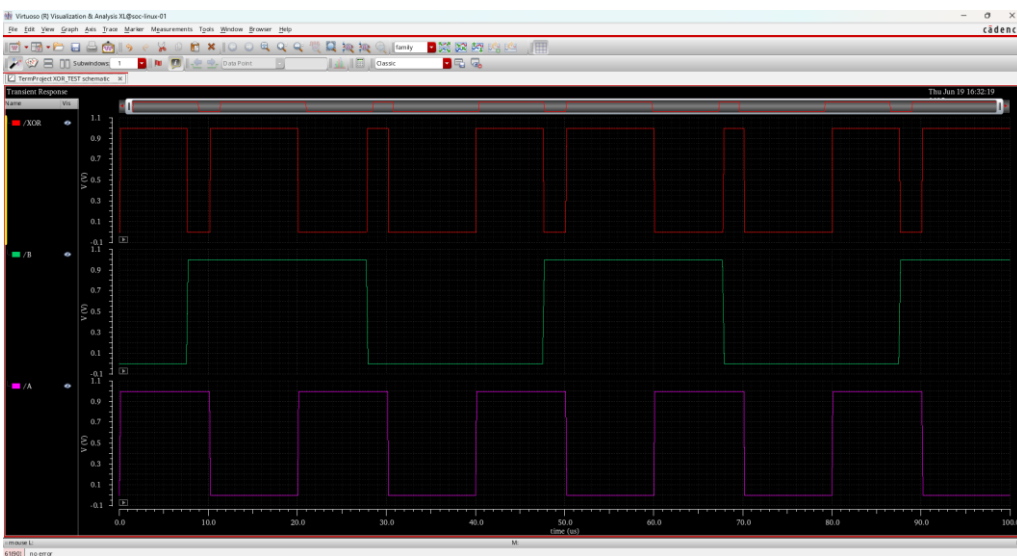
B. 시뮬레이션 설정



펄스 신호에 대한 XOR Gate 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 100 μ , Moderate
- Outputs: A, B, OUT 의 Waveform

C. 시뮬레이션 결과

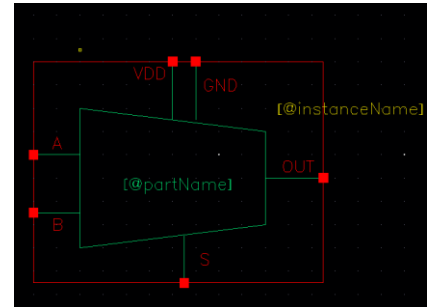
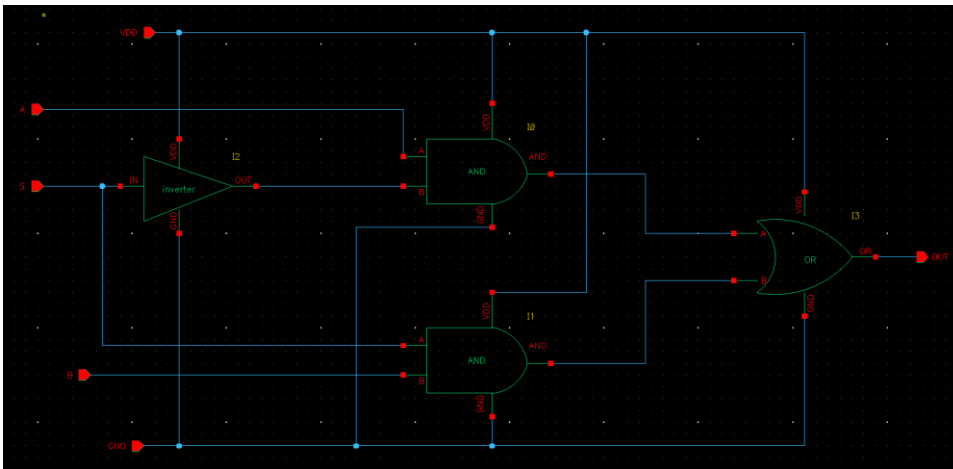


입력		출력
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

시뮬레이션 결과가 XOR gate 진리표와 동일합니다. 이를 통해 회로가 의도한대로 설계되었음을 확인할 수 있습니다.

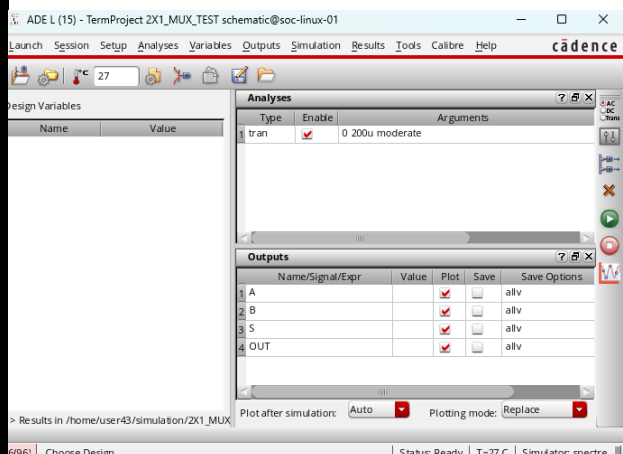
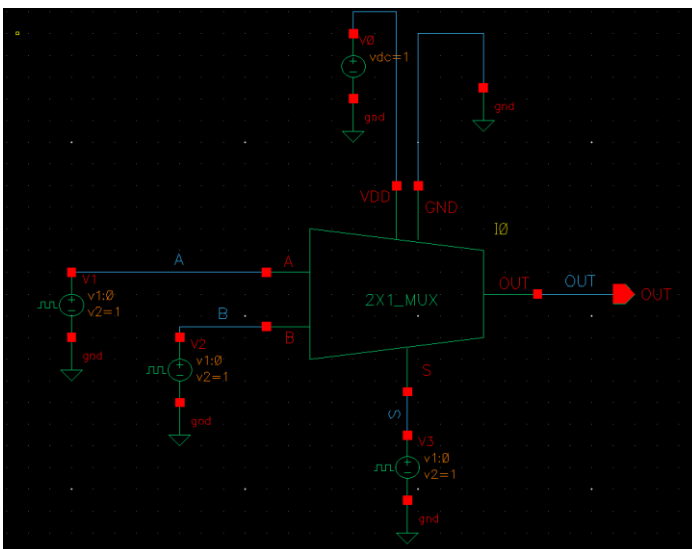
2.5 2x1 MUX

A. 회로



선택 신호 'S'에 따라 입력 A, B 중 하나를 골라 출력하는 2x1 Multiplexer의 논리를 구현한 회로입니다. 선택 신호 Inverter를 활용하여 S=0 일 때, 위쪽의 AND Gate가 활성화되어 입력 A가 선택되고, S=1 일 때, 아래의 AND Gate가 활성화되어 입력 B가 선택됩니다. 마지막으로 OR Gate는 선택된 두 경로 중 하나의 신호를 OUT으로 출력합니다.

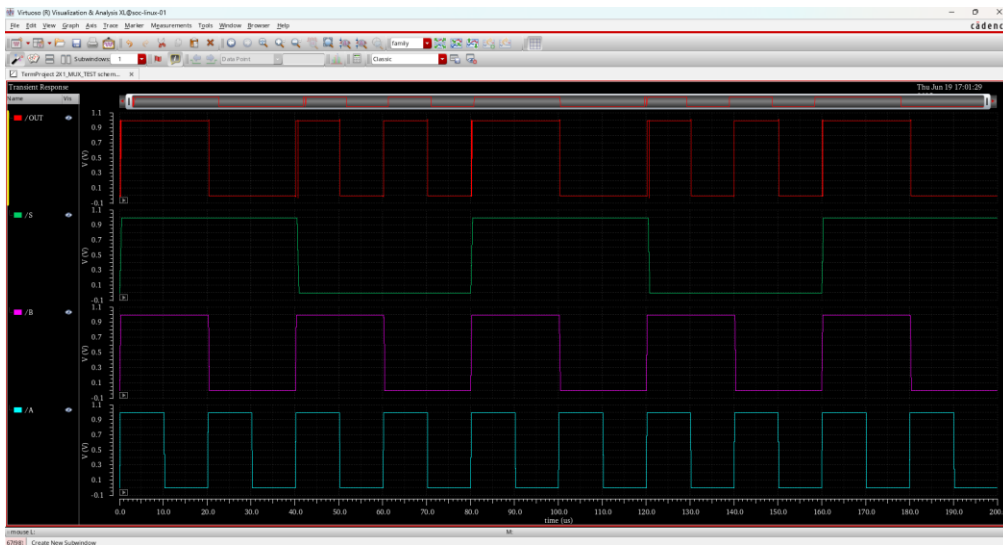
B. 시뮬레이션 설정



펄스 신호에 대한 2x1 MUX의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 200 μ , Moderate
- Outputs: A, B, S, OUT의 Waveform

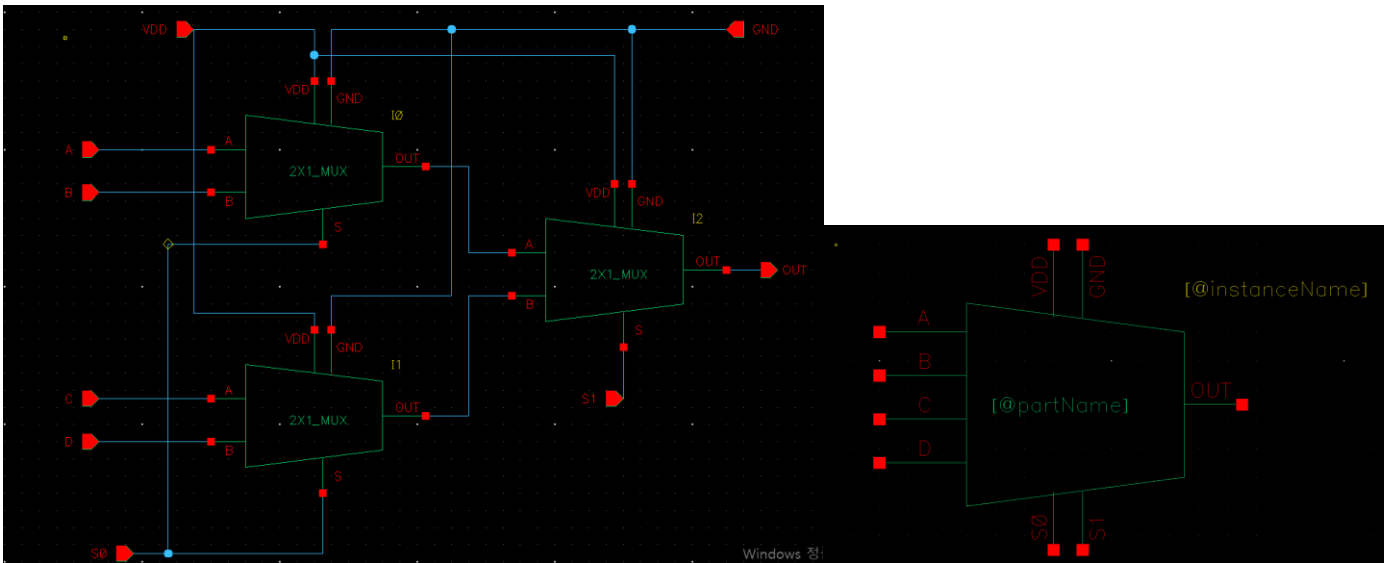
C. 시뮬레이션 결과



S=0 일때 OUT=A, S=1 일 때 OUT=B 가 되어 2x1 MUX 로써 동작하는 시뮬레이션 결과를 확인하였습니다. 이를 통해 회로가 의도한 대로 동작함을 확인할 수 있습니다.

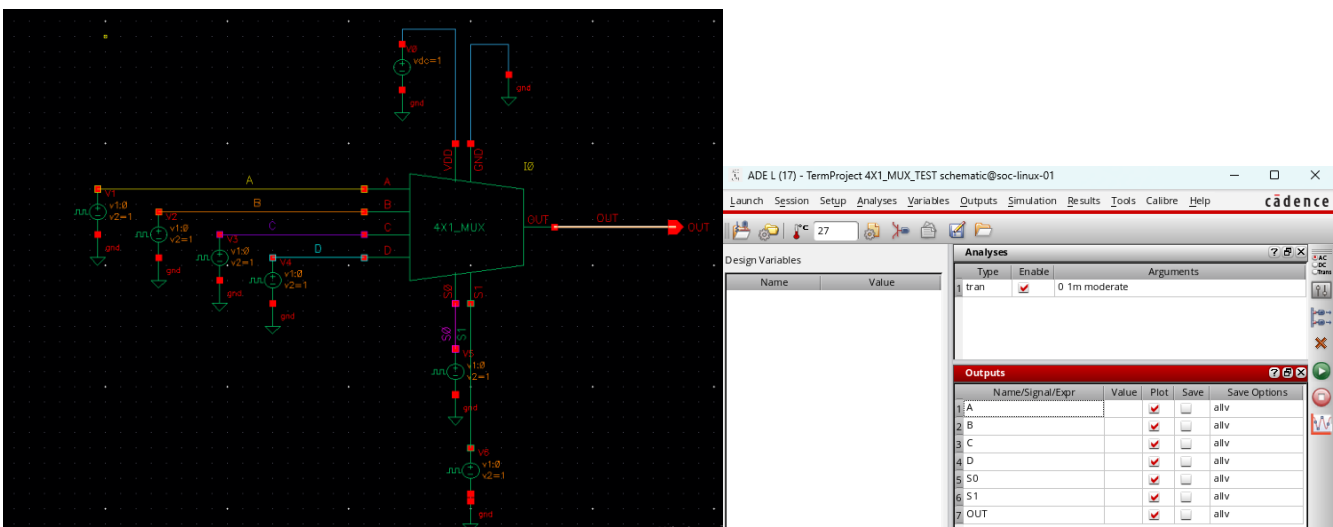
2.6 4x1 MUX

A. 회로



3 개의 2x1 MUX 를 계층적으로 연결하여, 2 개의 선택 신호 S0, S1 로 4 개의 입력 A, B, C, D 중 하나를 선택하는 4x1 MUX 의 논리를 구현한 회로입니다. 첫 번째 단계에서 선택 신호 S0 가 A, B 중 하나, C, D 중 하나를 각각 고르면, 두 번째 단계에서 선택 신호 S1 이 두 개의 중간 결과 중 최종 하나를 선택하여 출력으로 내보냅니다. 결과적으로 S1 과 S0 의 조합에 따라 4 개의 입력 중 원하는 하나를 골라내는 논리 회로가 완성됩니다.

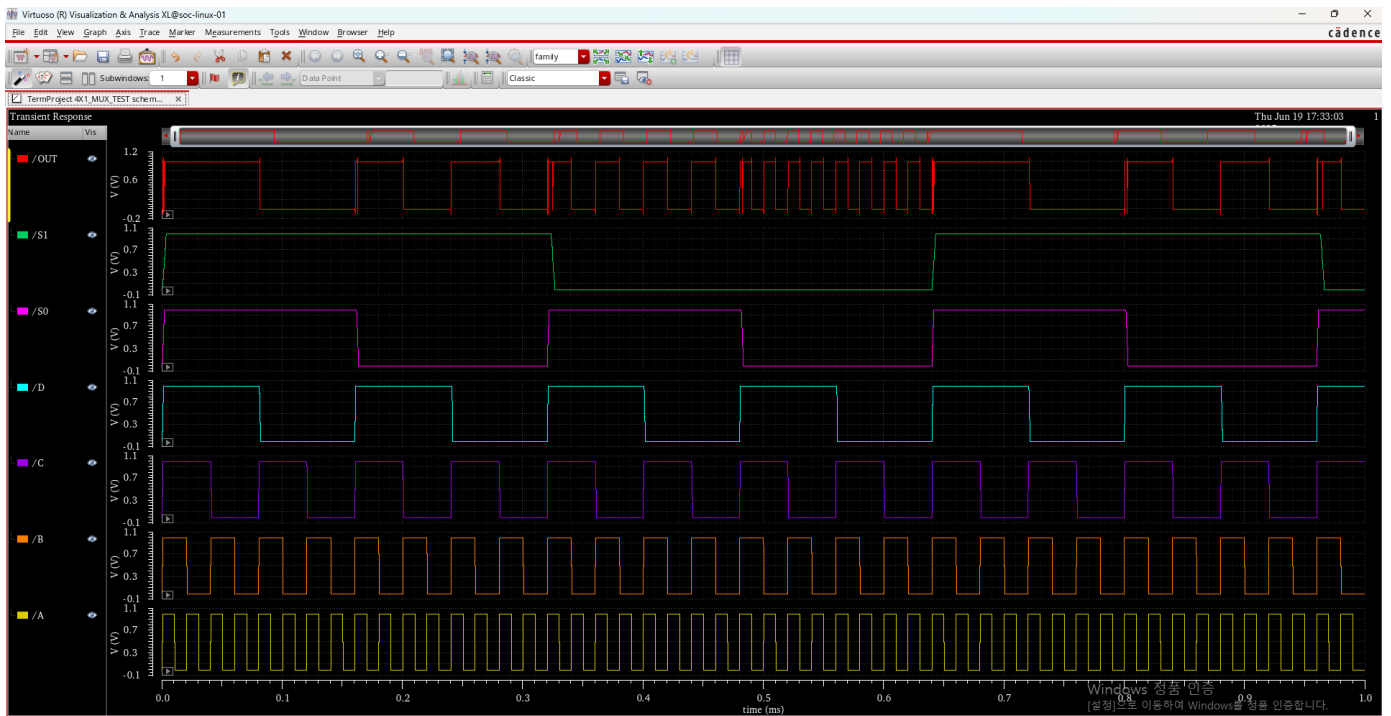
B. 시뮬레이션 설정



펄스 신호에 대한 2x1 MUX 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 1000 μ , Moderate
- Outputs: A, B, C, D, S0, S1, OUT 의 Waveform

C. 시뮬레이션 결과

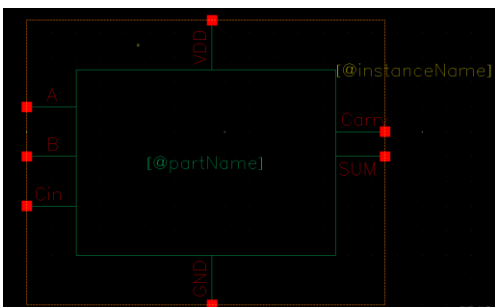
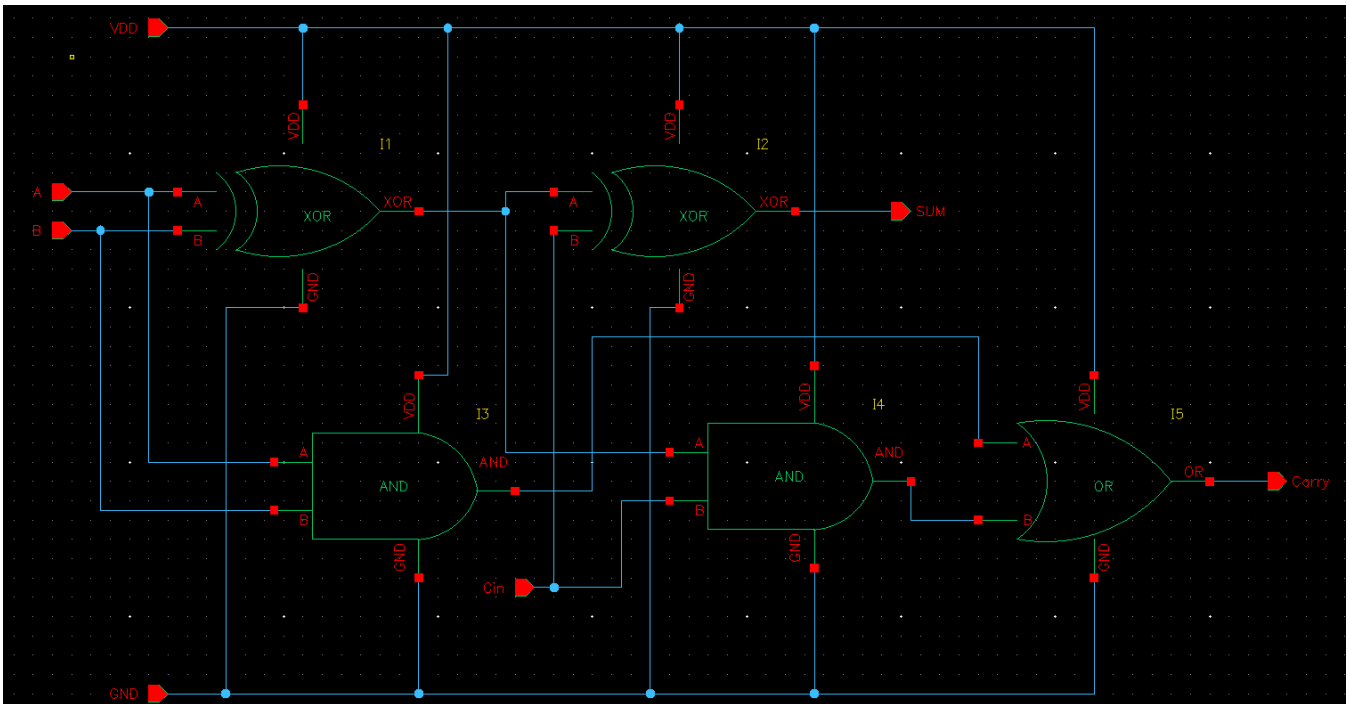


$*S=\langle S1, S0 \rangle$

$S=\langle 00 \rangle$ 일 때 $OUT=A$, $S=\langle 01 \rangle$ 일 때 $OUT=B$, $S=\langle 10 \rangle$ 일 때 $OUT=C$, $S=\langle 11 \rangle$ 일 때 $OUT=D$ 가 되는 시뮬레이션 결과를 보여줍니다. 이는 4x1 MUX 의 논리적 결과와 동일하며 이를 통해 회로가 의도한 대로 동작함을 확인할 수 있습니다.

2.7 Full Adder

A. 회로



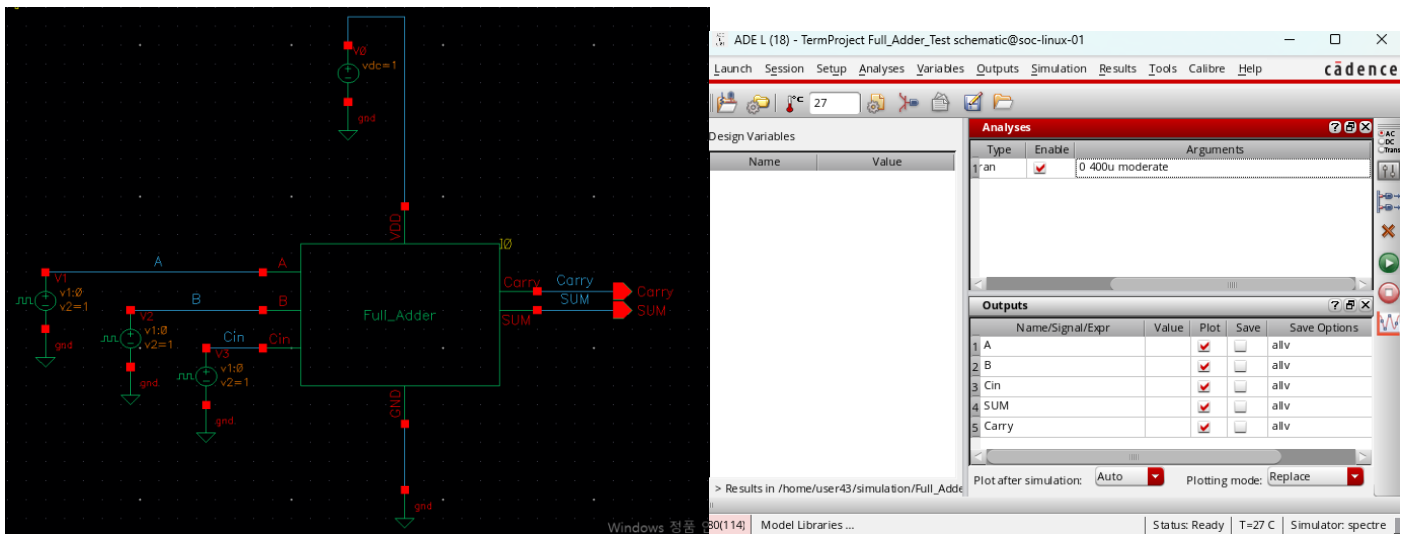
기본적인 논리 Gate [XOR, AND, OR]를 활용하여 1-bit Full Adder 의 논리를 구현한 회로입니다.

- $SUM = A \oplus B \oplus Cin$: Sum 은 두 개의 XOR Gate 를 활용하여 계산됩니다. 첫 번째 XOR Gate 를 통하여 $A \oplus B$ 을 계산합니다. 이 계산 값이 두 번째 XOR Gate 의 입력으로 들어간 후 Cin 과 XOR 연산을 거쳐 최종적으로 SUM 을 출력합니다.
- $Carry = (A \wedge B) \vee (Cin \wedge (A \oplus B))$: Carry 는 두 개의 AND Gate, 한 개의 OR Gate 를 활용하여 계산됩니다. 첫 번째 AND Gate 를 통하여 $A \wedge B$ 를 계산합니다. 해당 연산은 A, B 가 모두 1 일 때 자리 올림이 발생하는 경우를 처리합니다.

이후 두 번째 AND Gate 를 통하여 $Cin \wedge (A \oplus B)$ 연산을 진행합니다. 해당 연산은 A, B 중 하나만 1 인 상태에서 Cin 이 1 일 때 자리 올림이 발생하는 경우를 처리합니다.

두 조건 중, 하나라도 만족하면 자리올림이 발생합니다. 해당 논리를 구현하기 위해 최종적으로 OR gate 를 활용하여 각 AND Gate 의 논리 출력 값을 계산합니다.

B. 시뮬레이션 설정



펄스 신호를 활용하여 Full Adder 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 400 μ s, Moderate
- Outputs: A, B, Cin, SUM, Carry 의 Waveform

C. 시뮬레이션 결과



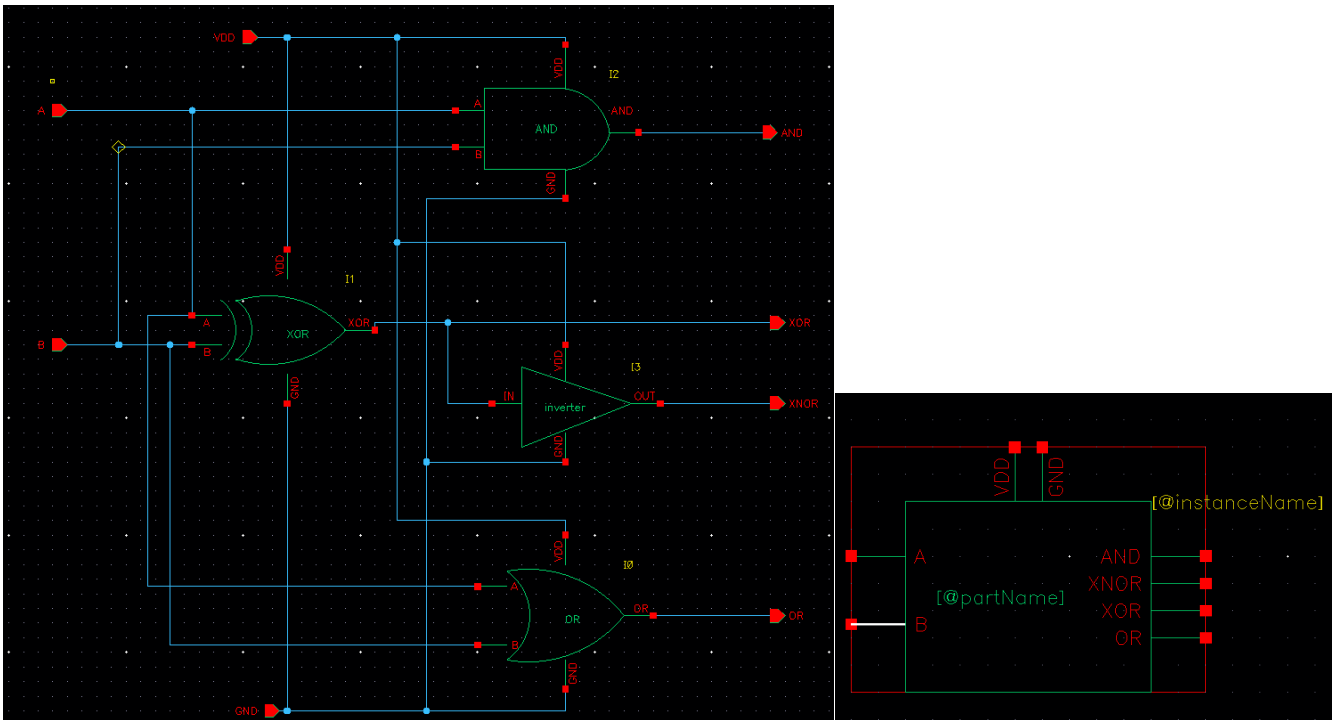
Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Input = A, B, Cin / Output = <Sum, Carry>

입력이 모두 0 일 때 Output=<00>, 입력 중 1 개만 1 일 때 Output=<10>, 입력 중 2 개가 1 일 때 Output=<01>, 입력이 모두 1 일 때 S=<11>이 되는 시뮬레이션 결과를 보여줍니다. 이는 Full Adder 의 논리적 결과와 동일하며 이를 통해 회로가 의도한 대로 동작함을 확인할 수 있습니다.

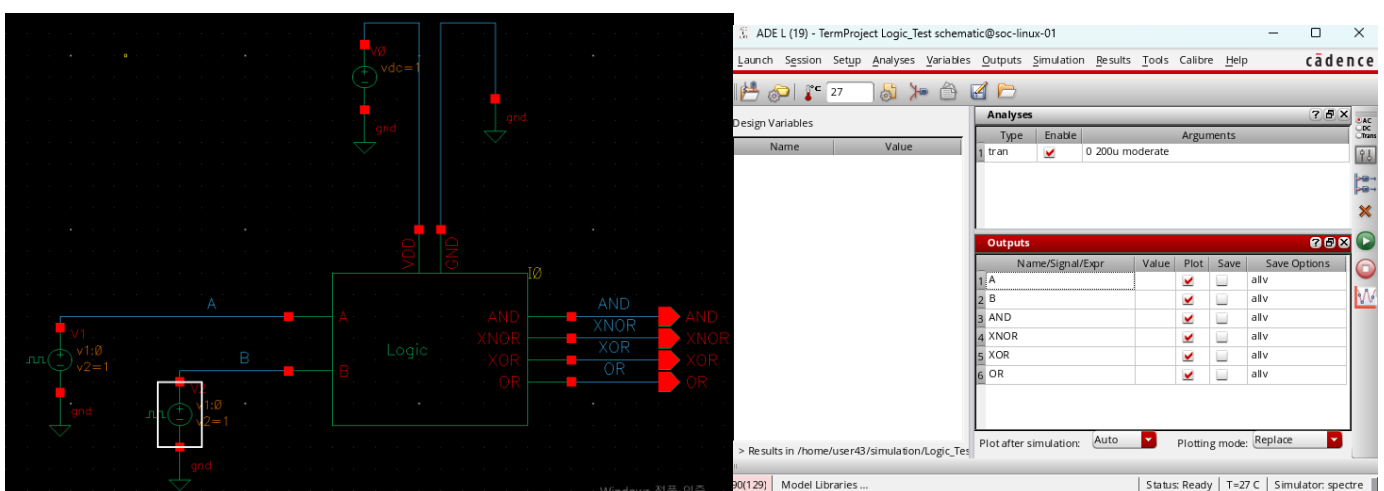
2.8 Logic Module

A. 회로



2 Input 을 활용하여 4 개의 기본적인 논리 연산(AND, XOR, XNOR, OR)를 동시에 출력하는 회로이며 ALU 에서 Logic Module 로써 기능하기 위해 설계되었습니다.

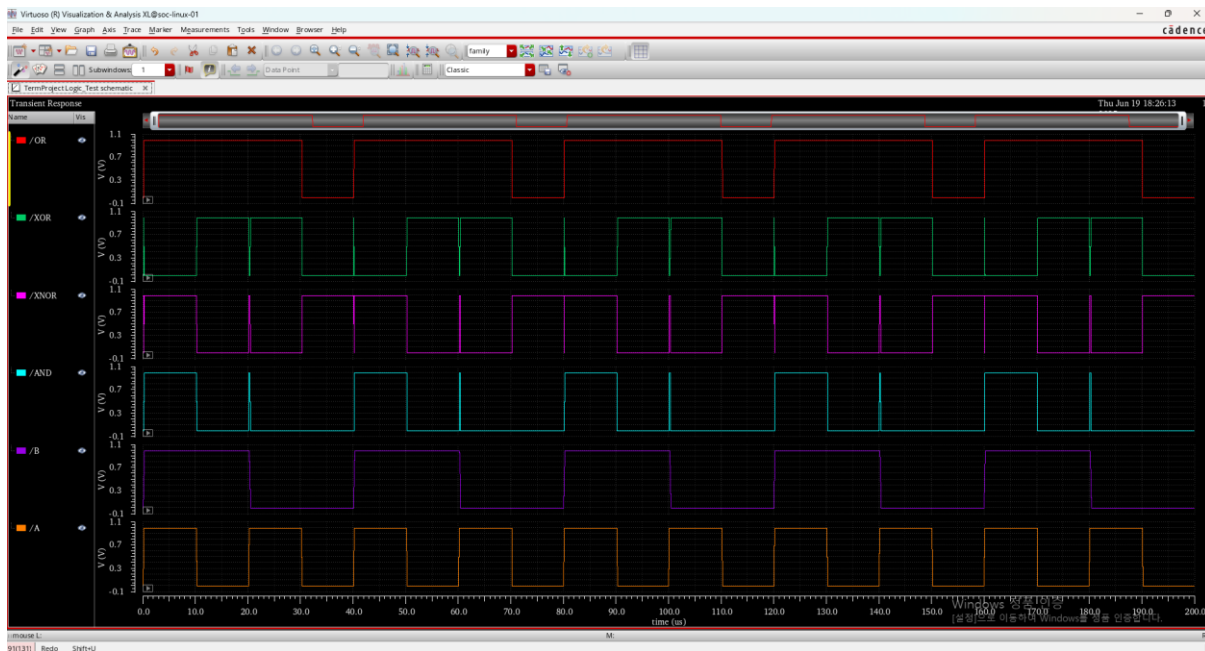
B. 시뮬레이션 설정



펄스 신호에 대한 Logic Module 의 동작 특성을 확인하기 위한 테스트 회로입니다

- tran Analysis: 200 μ , Moderate
- Outputs: A, B, AND, XNOR, XOR, OR 의 Waveform

C. 시뮬레이션 결과



시뮬레이션 결과가 AND, OR, XOR, XNOR gate 진리표와 동일합니다. 이를 통해 회로가 의도한대로 설계되었음을 확인할 수 있습니다.

AND

입력		출력
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

입력		출력
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

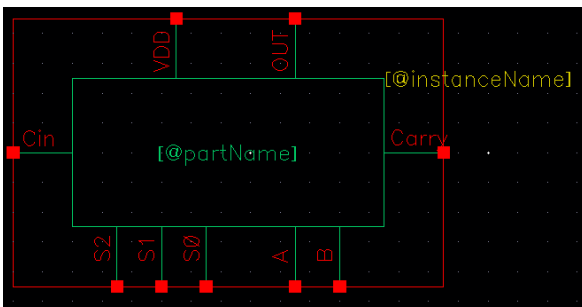
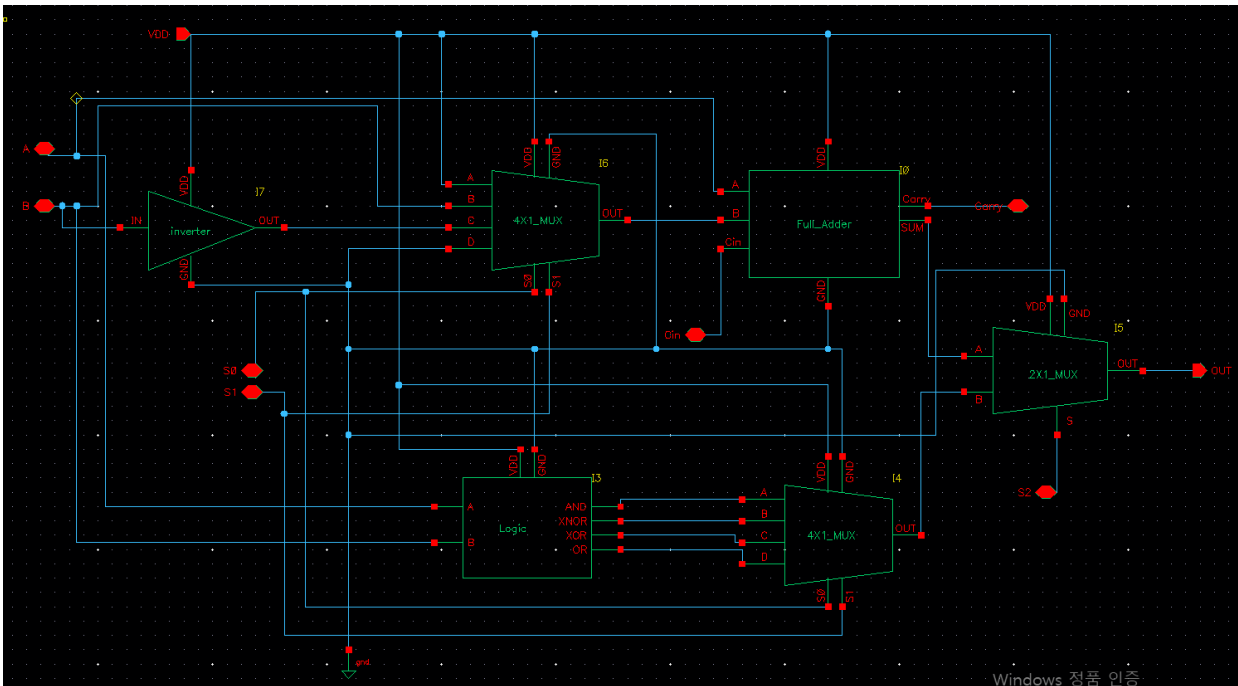
입력		출력
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

입력		출력
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

2.8 1bit ALU

A. 회로

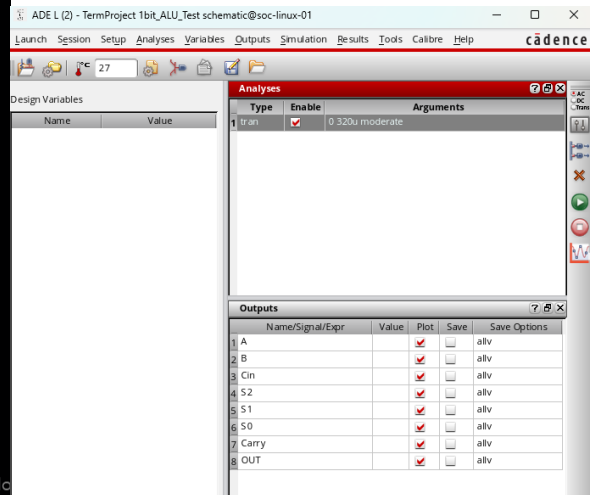
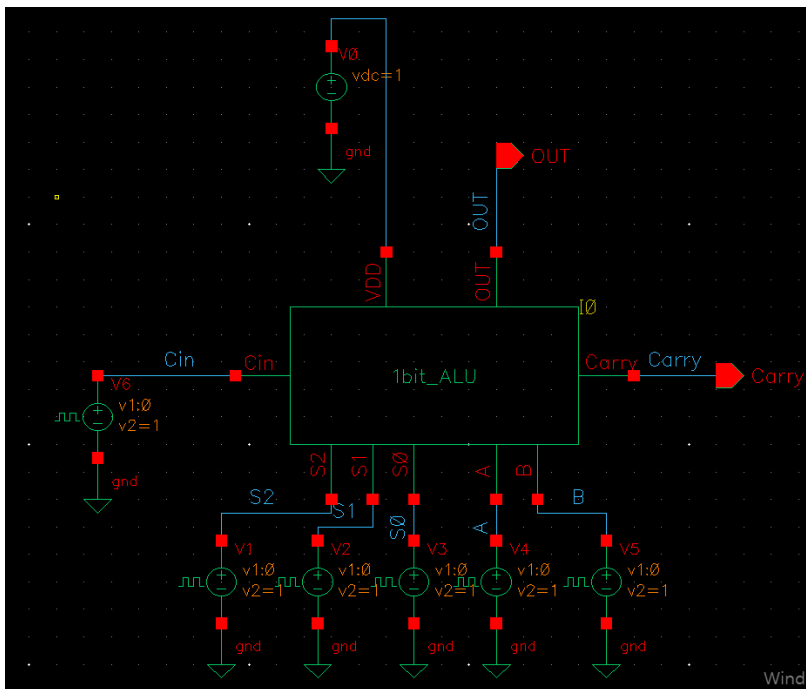


3 개의 선택 신호 S2, S1, S0 와 3 개의 입력신호 A, B, Cin 을 사용하여 산술 및 논리 연산을 수행하기 위하여 설계한 1bit ALU(Arithmetic Logic Unit)입니다. 1 개의 Inverter, 1 개의 2x1 MUX, 1 개의 Logic Module, 2 개의 4x1 MUX, 1 개의 Full Adder 를 활용하여 설계되었으며, 해당 회로는 2 개의 출력신호 OUT, Carry 로 구성되었습니다.

회로는 크게 산술 연산, 논리 연산, 최종 결과 세 개의 부분으로 나눌 수 있습니다.

- 논리 연산 : Logic Module 이 논리 연산을 수행하며 S1, S0 조합과 아래쪽 4x1MUX 를 활용하여 Logic Module 의 결과를 선택합니다.
- 산술 연산 : Full Adder 와 위쪽에 있는 4x1 MUX 를 통해 수행되며, A 와 연산할 신호를 4x1 MUX 와 S1, S0 조합을 활용하여 A, B, not B, 1 중에 결정합니다. Cin 은 S0 에 연결되어 있습니다.
- 최종 선택 : S2 와 2x1 MUX 를 활용하여 산술 연산의 결과와 논리 연산의 결과 중 무엇을 OUT 출력할지 결정합니다.

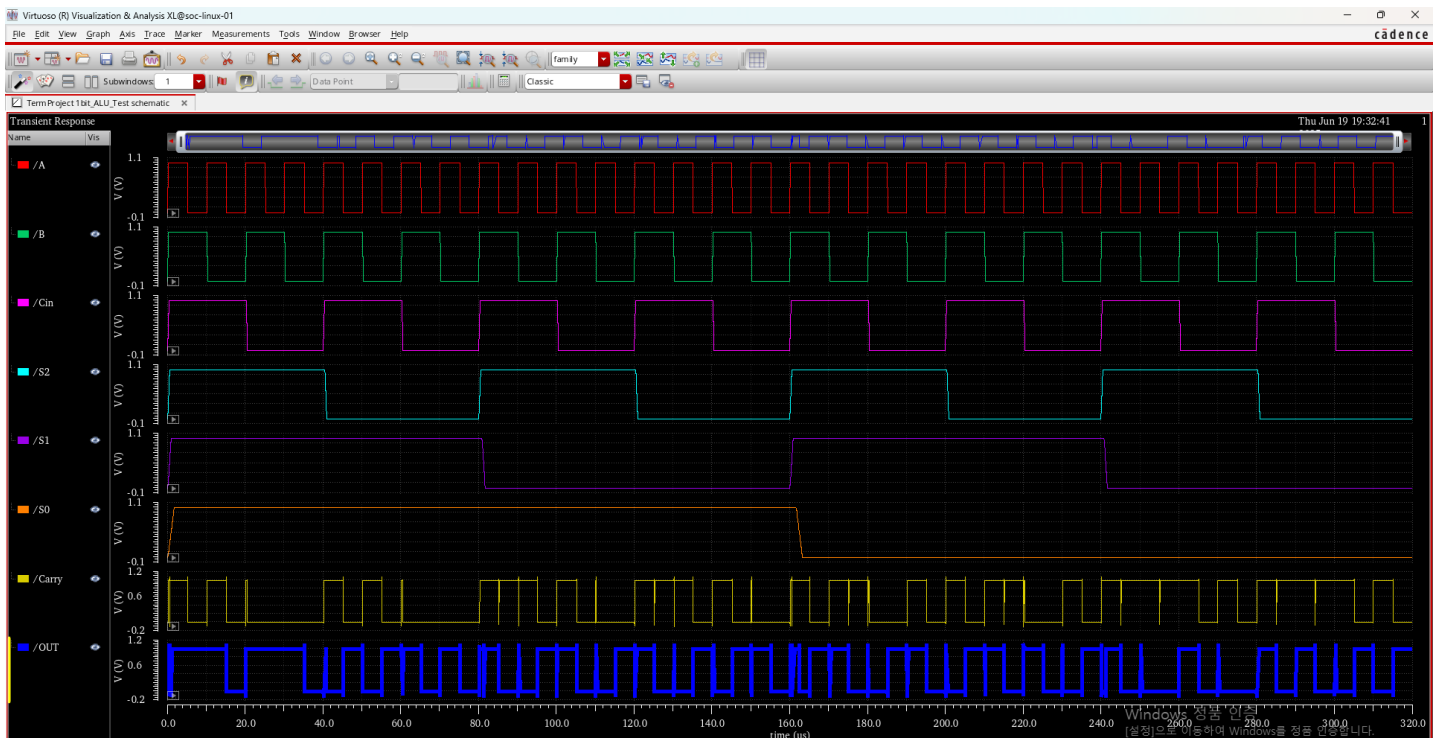
B. 시뮬레이션 설정



펄스 신호를 활용하여 1bit ALU 의 동작을 확인하기 위한 테스트 회로입니다

- tran Analysis: 320 μ , Moderate
- Outputs: A, B, Cin, S2, S1, S0, Carry, OUT 의 Waveform

C. 시뮬레이션 결과



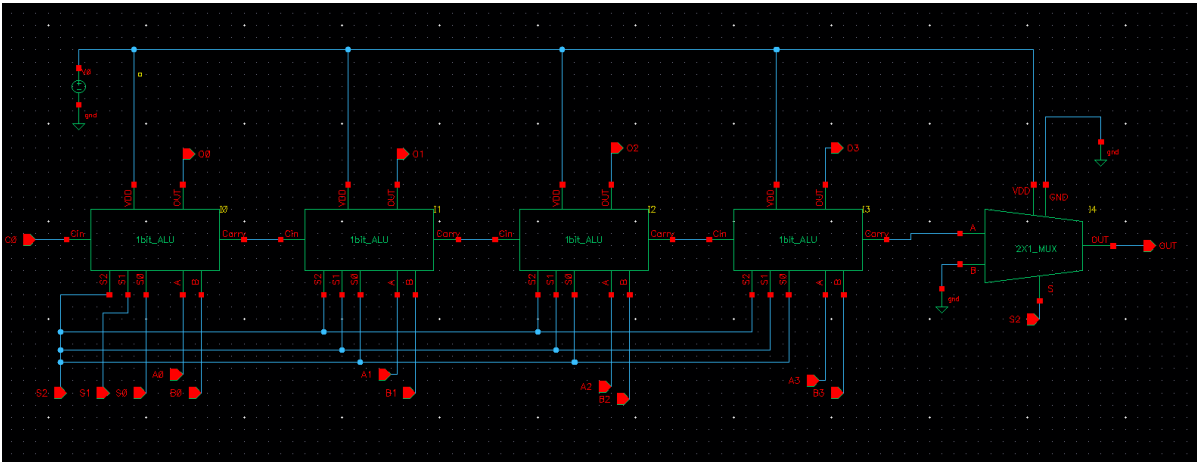
시뮬레이션 결과가 처음 의도했던 ALU의 기능이 정확히 일치합니다.

이를 통해 목표로 하던 1bit ALU가 구현되었음을 확인할 수 있습니다.

S2	S1	S0	Operations	Function
0	0	0	$G = A - 1$	DECREMENT
0	0	1	$G = A + B$	ADDITION
0	1	0	$G = A + \overline{B} + 1$	SUBTRACTION
0	1	1	$G = A + 1$	INCREMENT
1	0	0	$G = A \wedge B$	AND
1	0	1	$G = A \oplus B$	XOR
1	1	0	$G = \overline{A}$	XNOR
1	1	1	$G = A \vee B$	OR

2.9 4bit ALU [추가 응용]

A. 회로



1bit ALU 4 개와 2x1 MUX 를 활용하여 만든 4bit ALU 입니다.

이와 동일한 메커니즘으로 4bit ALU 를 기반으로 이를 확장해 나가며 8bit ALU, 16bit ALU, 32bit ALU 를 설계함을 추론할 수 있습니다.

3. 결론

Cadence Virtuoso Tool 을 활용하여 1bit ALU 회로를 설계하고, 과도 응답(Transient Response) 시뮬레이션을 통해 동작 특성을 분석하였습니다. 이를 통해 각 선택 신호(S2, S1, S0) 조합에 따른 산술 연산(덧셈 등)과 논리 연산(AND, OR, XOR 등) 결과가 정상적으로 출력됨을 확인하고, 각 연산에 대한 논리적 정합성을 설계 목표와 비교 검증할 수 있었습니다. 시뮬레이션 결과 파형은 각 연산 모드에 대한 ALU 의 이론적 진리표 및 동작 원리와 잘 부합함을 확인하였습니다.

결론적으로, 본 실습은 Cadence Virtuoso Schematic Editor 및 ADE 를 활용하여 디지털 조합 논리 회로의 설계, 시뮬레이션 설정, 결과 파형 분석 및 데이터 해석 능력 등 디지털 집적회로 설계 및 검증 과정에 필요한 실질적인 역량을 강화하는 중요한 계기가 되었습니다. 이러한 경험은 향후 더욱 복잡하고 다양한 기능을 수행하는 CPU 와 같은 디지털 시스템을 설계하고 분석하는 데 필요한 견고한 기초를 다지는 데 크게 기여할 것으로 기대합니다