ETRI 0.5um CMOS Std-Cell DK:

CMOS 인버터 회로도 작성과 SPICE 시뮬레이션

연구과제명	반도체 기술 개발 지원 고경력 전문인력 활용 사업(25JB1710)
연구기간	2025 년 6 월~2026 년 12 월
연구책임자	고상춘
기록자	국일호
확인자	
작성일자	2025 년 7 월 15 일

ETRI 한국전자통신연구원
www.etri.re.kr

CMOS 인버터 회로도 작성과 SPICE시뮬레이션

목차:

[부록] XSchem 실습: 계층적 회로도 그리기와 시뮬레이션 테스트벤치

목차

II. 테스트 벤치

(1) 테스트 벤치 작성

(2) 계층적 회로도

III. SPICE 시뮬레이션

(1) 시뮬레이션용 네트리스트 생성

(2) 시뮬레이션 수행

(3) 시뮬레이션 결과 보기

[주] '디자인 킷'의 깃-허브 저장소 내용이 매주 갱신되고 있습니다. 기존에 받은 내용이 있다면 갱신 하십시요.

```
$ cd ~/ETRI050_DesignKit
$ git pull
```

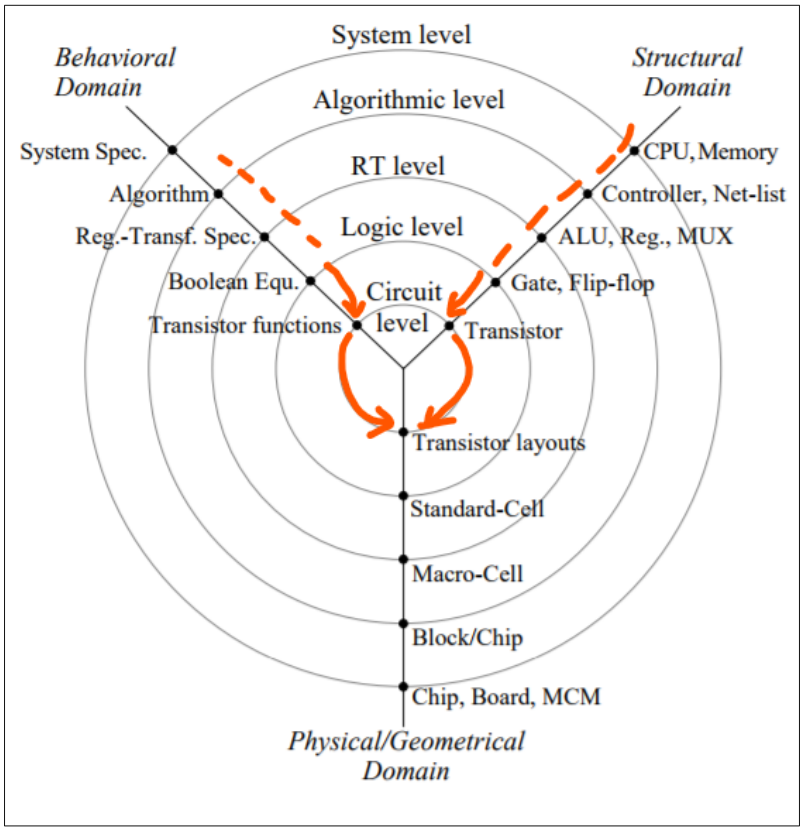
[주] '디자인 킷'을 새로 받는 경우 앞서 발행한 설치 문서를 참고하십시요.

기술노트1, 2: 가상머신 리눅스 설치[pdf], 오픈-소스 반도체 설계도구 설치[pdf] [바로가기]

---

1. 학습 개요

레이아웃(Layout)은 트랜지스터 회로를 반도체 부품으로 제작하기 위한 도면이다. 무작정 도면을 그려 제조 공장에 제출하기 전에 얻고자 하는 대로 작동할 것이라는 보장이 있어야 한다. 도면을 그리기 전에 먼저 회로도를 작성한 후 시뮬레이션을 통해 동작을 검증하기로 한다. 검증된 회로를 바탕으로 제조용 도면을 그리는 절차는 반도체 설계 뿐만 아니라 어떤 영역의 개발에서도 당연한 순서다.



트랜지스터 회로도와 레이아웃은 반도체 설계 방법론의 가장 낮은 추상화 수준에 위치한다.

오픈-소스 회로도 그리기 도구로 Xschem, 회로 시뮬레이터는 ngSpice 다. 제조 공정은 ETRI 0.5um CMOS(내칩 제작 서비스)다. 오픈-소스 도구의 설치법은 [기술문서2]를 참조한다. 이 문서는 반도체 공정이나 트랜지스터 회로를 섬세히 다루려는 것이 아니다. 반도체 설계 도구의 활용과 설계 과정을 익히는 목적이다. 연습할 회로는 CMOS 인버터(inverter)다.

2. 인버터 회로도 입력

두 종류(p-, n-채널) MOS 트랜지스터를 복합적으로 사용하는 디지털 인버터 회로를 그리고 시뮬레이션 해본다. 본 문서에서 설명하는 '디자인 킷'의 예제는 그래픽 기반 회로 그리기 도구로 Xschem, 회로 시뮬레이터는 ngSPpice를 사용한다.

2-1. 설계 디렉토리 구조

설계 자동화 도구를 사용하는 과정에서 수많은 중간 파일들이 생성된다. 이 파일들은 자동화 도구들에 의해 생성되고 사용하기 때문에 인간이 읽기 매우 곤란하다. 설계가 끝나면 정리가 필요한데 자칫 중요 설계 파일이 삭제되는 낭패를 격지 않으려면 디렉토리를 만들어 두고 적절하게 관리되어야 할 것이다. 앞서 도구 설치 절차를 통해 깃-허브에서 디자인 킷을 내려 받았을 것이다. 앞으로 진행할 예제의 편의를 위해 사용자의 홈에 소프트 링크 시켜 놓은 ETRI050_DesignKit의 해당 디렉토리로 이동하여 파일들을 확인해 보자.

```
$ cd ~/ETRI050_DesignKit/Tutorials/1-1_Inverter_Xschem

$ tree
.
├── inverter1
│   ├── inverter1.sch
│   └── simulation
└── inverter_TB
    ├── inverter.sch
    ├── inverter.sym
    ├── inverter_TB.sch
    ├── inverter_top.sch
    └── simulation
```

확장자 .sch는 XSchem의 회로도 파일이다. 확장자 .sym은 회로도를 대표하는 상징이다. 계층적 회로도를 작성할 때 하위 회로를 대표한다. 디렉토리 simulation 은 회로 시뮬레이션에 필요한 중간 파일들과 결과 파일들이 저장된다. 확장자 .spice는 회로도에서 추출한 SPICE 네트리스트(netlist)다. 그림은 인간이 회로를 그려 넣기(design entry) 편리한 방편일 뿐이다. 시뮬레이션을 하려면 문법 체계를 갖춘 언어를 빌어 묘사되어야 한다. SPICE 네트리스트도 그중 하나다. SPICE 네트리스트는 문법 체계가 지극히 단순해서 인간이 읽고 쓰기 매우 불편하기 때문에 대부분 회로 작성은 그래픽 도구를 사용한다.

[주] SPICE 네트리스트 문법과 회로 시뮬레이션 알고리즘은 표준으로 제정되어 있어서 상용 및 오픈-소스 도구들이 개발되어 다양한 이름으로 발매되고 있다. 마치 C++ 언어 표준이 제정되고 다양한 빌드(컴파일러와 링커 그리고 표준 라이브러리들을 포함한)도구들이 발매되어 있는 것과 같다. GNU C/C++ 빌드 도구들의 자리가 확고부동하듯이 전자회로 설계 도구들 가운데 언어 표준이 제정된 시뮬레이터(SPICE 뿐만 아니라 HDL 포함)의 경우 오픈-소스 도구들이 널리 사용되고 있다. 이에 영향을 받아 상용 도구들 또한 무료로 배포되거나 저렴하게 보급되고 있다.

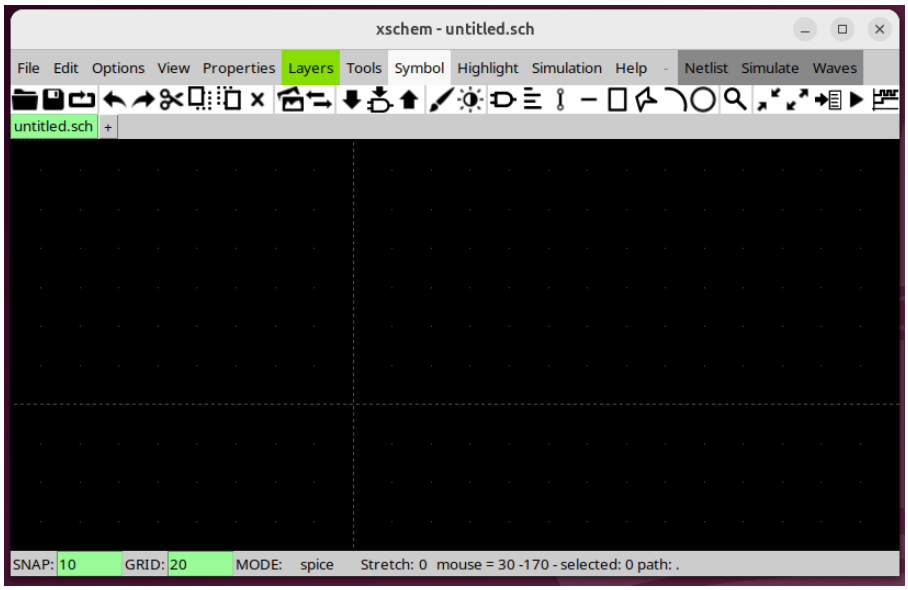
2-2. NSPL 0.5um CMOS 공정 모델 파일

반도체 소자의 물리적 특성은 제조 공장의 공정을 따른다. 제조를 의뢰하려는 반도체 공장으로부터 물성 인자들을 제공받게 되는데 이를 PDK(Process Design Kit)라 한다. 회로 시뮬레이션 도구 SPICE 에서 사용될 mos 트랜지스터 모델도 PDK의 일부다. NSPL 0.5um CMOS 공정의 PDK는 KION 을 통해 입수할 수 있다[3]. SPICE 모델 파라메터 버젼은 2024년 2월자로 파일명은 05cmos_model_240201.lib 다. 모델 파일은 디자인 킷의 tech 폴더에 저장되어있다.

2-3. XSchem 으로 CMOS 인버터 회로도 작성

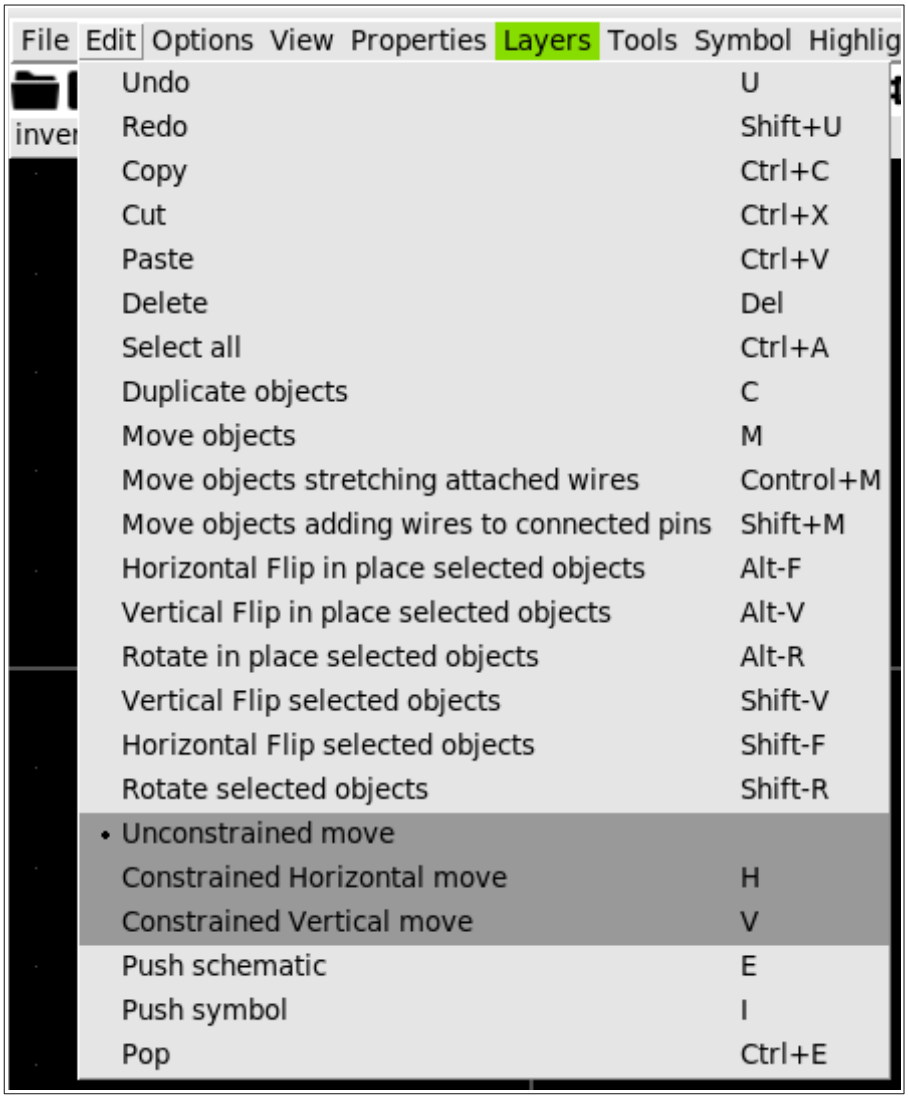
회로도는 작성 도구는 XSchem 이다. XSchem으로 아래와 같은 인버터 회로를 작성하자. 작업 디렉토리를 이동한 후 XSchem 실행한다.

```
$ cd ~/ETRI050_DesignKit/Tutorials/1-1_Inverter_Xschem
$ xschem
```



a. XSchem의 단축 키

XSchem의 GUI 화면에 보이는 메뉴는 많은 기능 중 극히 일부분이다. 많은 기능들을 단축키로 지정해 놓고 있다. 단축키를 사용하면 회로 그리기의 생산성을 높일 수 있다. 툴 사용법은 그저 생산성 향상의 방안 일 뿐이다. 필요할 때마다 찾아 익히도록 하자[Xschem Editor Commands]. 메뉴의 Help > Keys 를 따라가면 모든 단축키들을 볼 수 있다. 편집 메뉴의 단축키는 아래와 같다.



b. 마우스 버튼

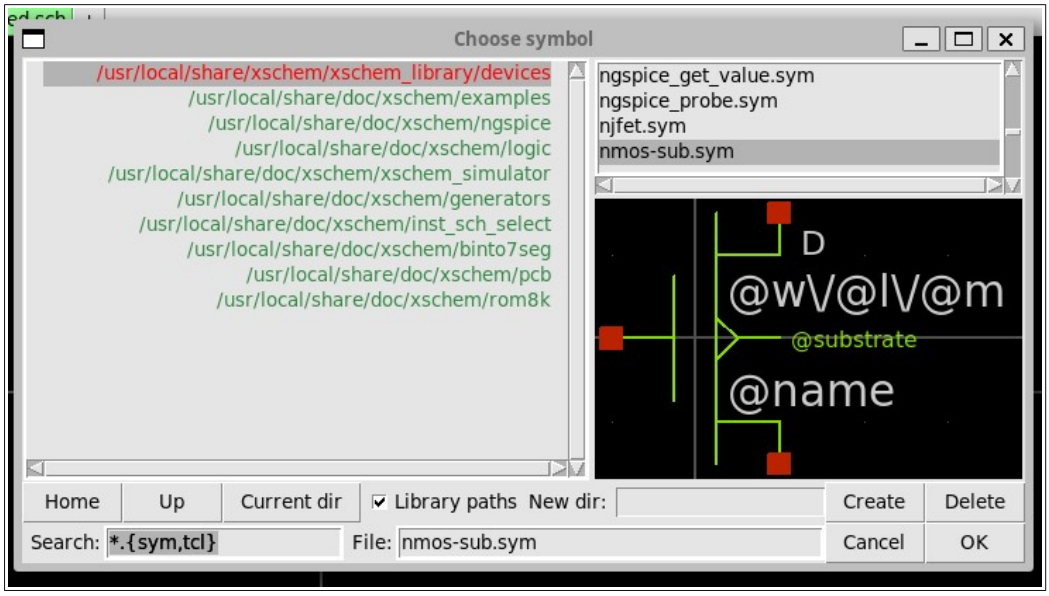
마우스 버튼의 기본 기능은 아래와 같다.

- '왼쪽 버튼'으로 회로 개체 선택한다. '왼쪽버튼 더블클릭'으로 선택한 회로 개체 속성 편집 창을 연다.
- '오른쪽 버튼'은 회로 편집용 팝-업 메뉴를 띄운다.
- '중앙 버튼'을 누른채 마우스를 움직이면 회로 도면을 상하좌우로 이동 시킨다.
- '휠'은 도면을 확대 또는 축소한다.

2-4. 회로 개체(nmos, pmos) 불러오기

도면상에 마우스 오른쪽 버튼을 눌러 팝-업 메뉴를 띄운 후 'Insert Symbol'을 선택한다. 단축키는 'Ins' 다.

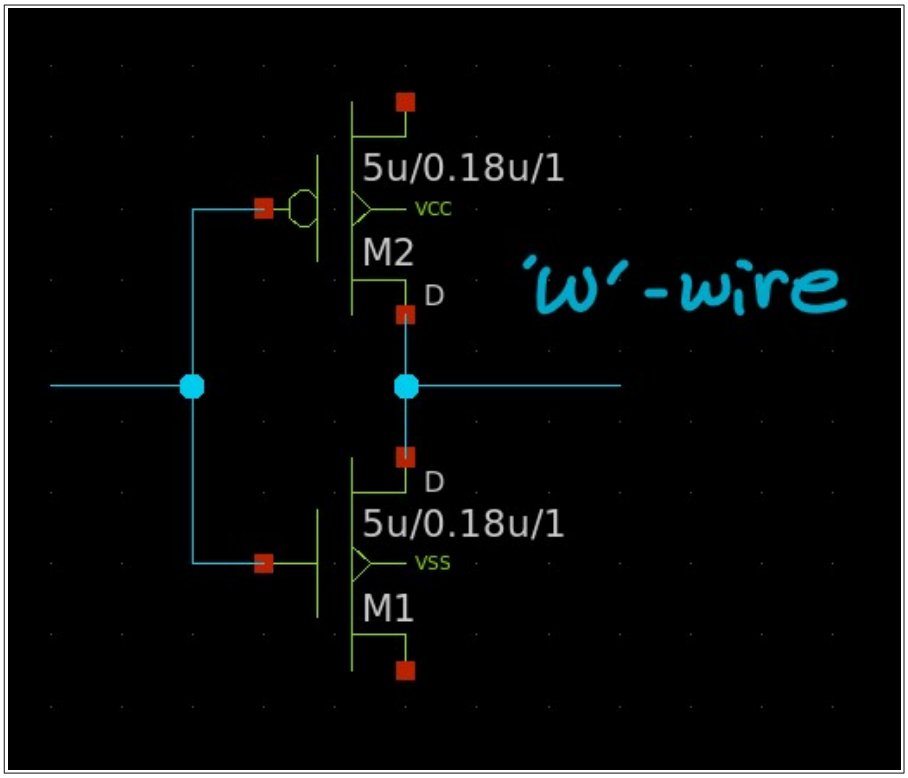
XSchem 에서 기본적으로 제공되는 심볼 중 하나를 고를 수 있다. nmos-sub.sym 과 pmos-sub.sym를 각각 선택하여 배치한다.



범용 mos의 심볼로서 레이아웃으로 그리게 될 트랜지스터의 폭(width)과 길이(length)가 인수화 되어있다. 기본값은 폭 5um, 길이는 0.18um 다. NSPL 0.5um CMOS 공정의 트랜지스터 디자인 룰(POLY1/3.PL1.W1)에 맞도록 길이를 0.5um 로 변경해야 한다. 도구를 익히는 중이므로 일단 기본값으로 두도록 한다.

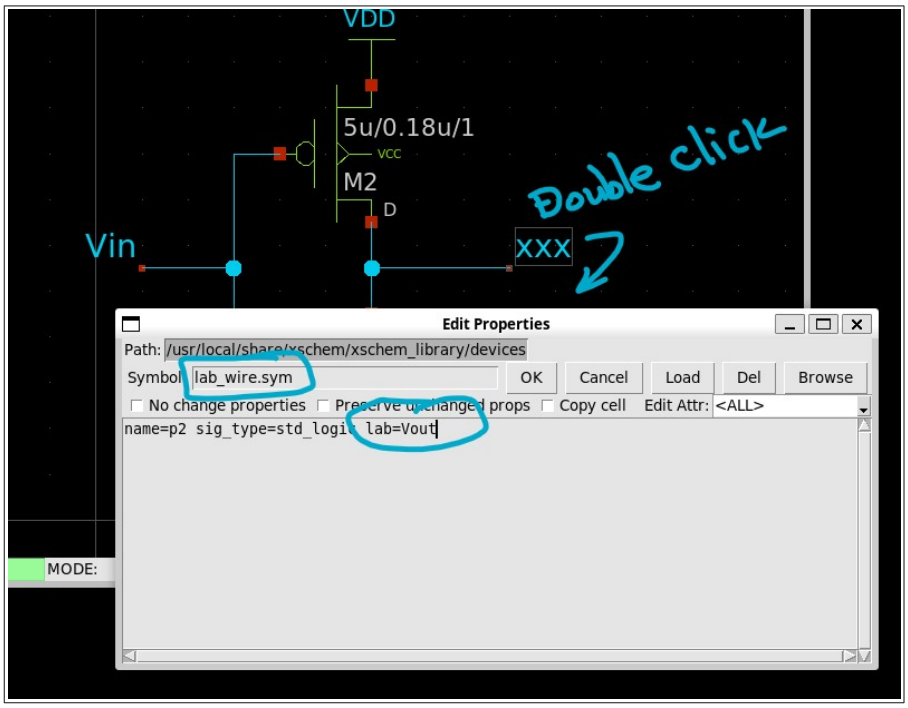
2-5. 배선

결선 도구를 써서 두 mos 트랜지스터로 구성된 인버터 회로를 배선한다. 결선 도구의 단축키는 'w'다.

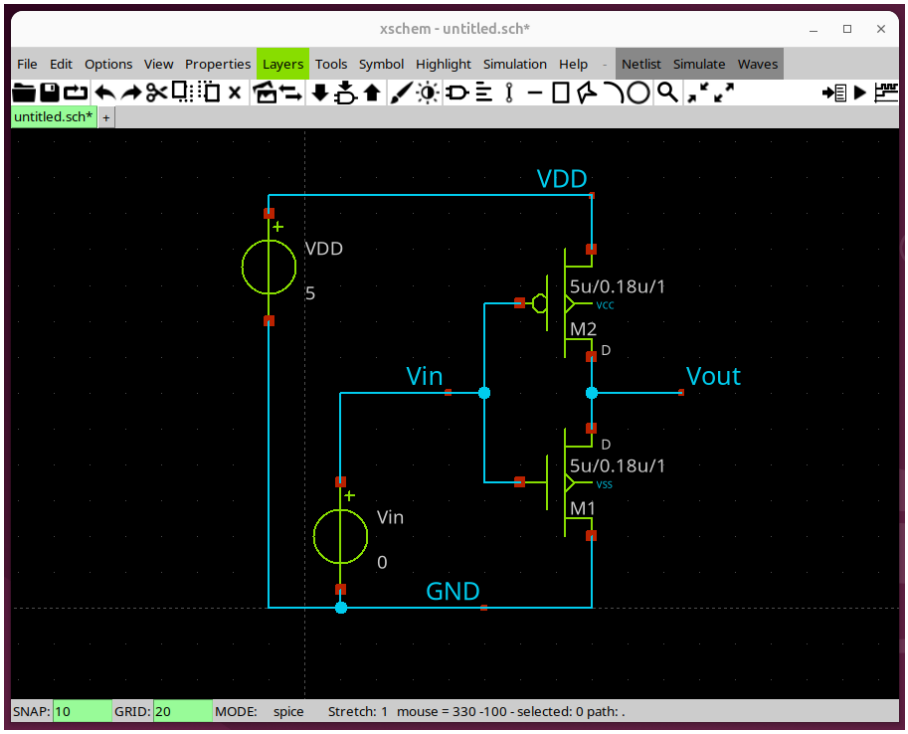


2-6. 라벨 붙이기

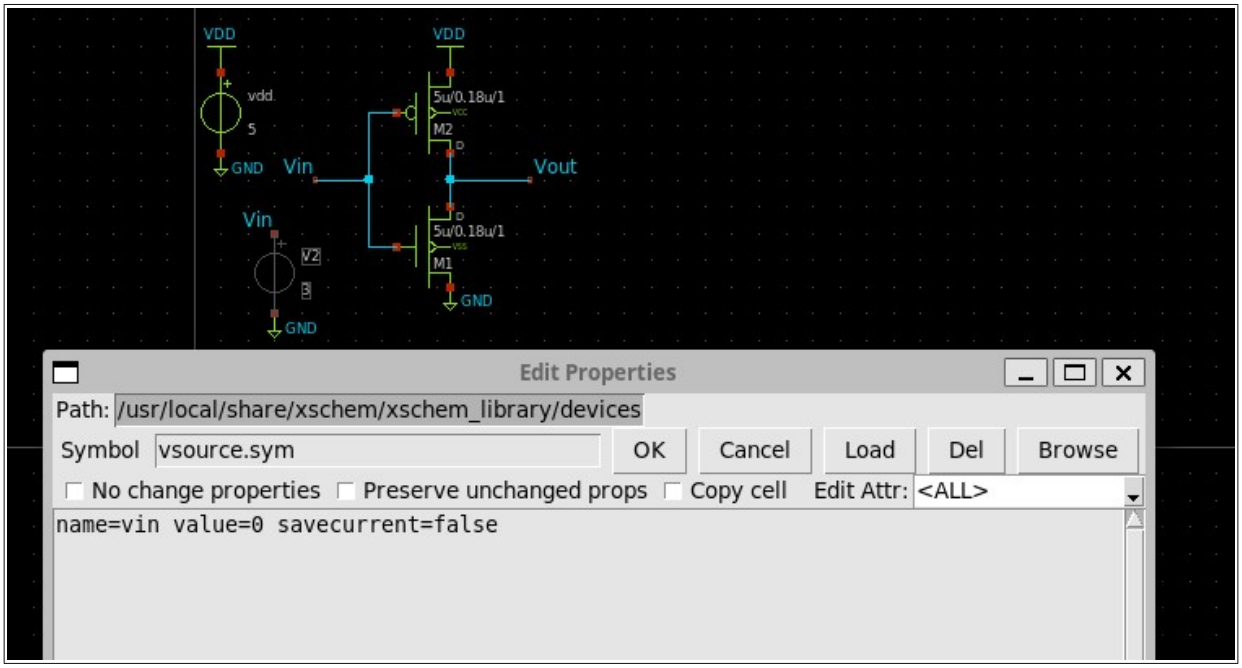
전원 VDD와 접지 GND 심볼을 붙인다. 결선에 붙일 라벨 객체의 심볼은 lab_wire.sym 이다. 이 역시 인수화 심볼이므로 더블-클릭하여 속성 편집 창을 불러내 적절한 라벨을 붙여준다.



입력 Vin 과 출력 Vout, 전원 VDD 와 접지 GND 라벨을 붙여 아래와 같이 회로를 완성한다.



전압 공급기(Voltage Source)로 회로 전원 VDD와 입력에 넣어 줄 신호 Vin를 정의한다. 입력 신호는 변화하는 전압으로 표현하게 되므로 개체 심볼 역시 전원 곱급기 vsource.sym 를 사용한다. 회로 전원의 이름은 VDD, 전압은 5V 고정되었다. 입력 신호의 전압 Vin은 회로의 시험을 위해 변화를 줄 것이다. 일단 이름을 Vin으로 지정해두고 초기 전압 값은 0으로 놓자.

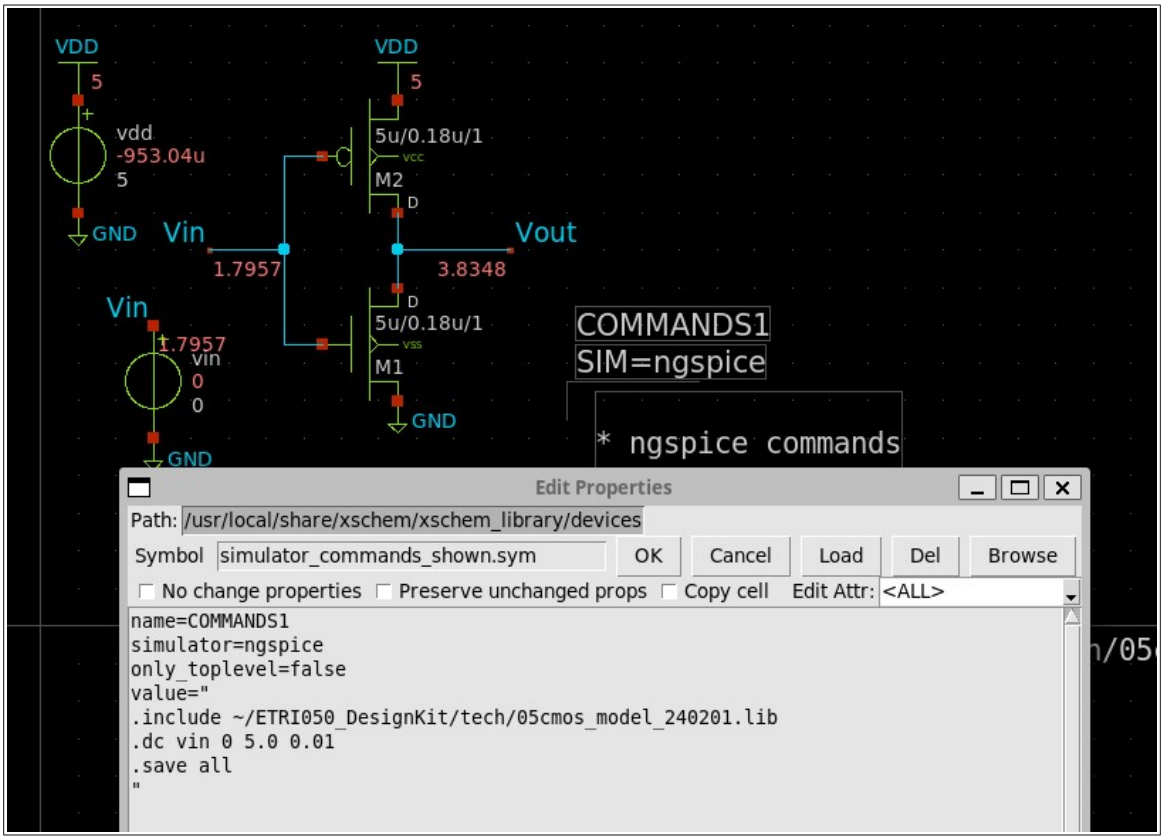


3. 시뮬레이션

인버터 회로 작성이 완료되었다. 다음은 회로 시뮬레이션이다. 시뮬레이션을 통해 동작이 검증되어야 설계라고 할 수 있다.

3-1. 시뮬레이션 명령

회로 시뮬레이션을 수행하기 위한 명령을 작성한다. 시뮬레이터는 ngSpice다. 회로 시뮬레이터 SPICE를 기반으로 만들어 진 오픈-소스 도구다. 시험 입력을 주는 방법은 관찰하려는 목적에 따라 매우 다양하다. 시간상 신호의 변화를 관찰하거나 주파수 응답 특성을 측정 할 수 있다. 회로에서 관찰할 물리량은 기본적으로 전압(V)과 전류(I)이며 그로부터 전력을 살펴볼 수 있다. SPICE 시뮬레이터에 관한 동영상 교재도 매우 다양하게 많다. 시뮬레이션 도구 사용법을 다룰 목적이므로 간단하게 입력의 전압 변화에 반응하는 인버터 회로의 출력 전압과 전류변화를 관찰하기로 한다. 시뮬레이션 명령 심볼 simulator_commands_shown.sym 을 회로도에 삽입하고 속성을 아래와 같이 편집해주자.



심볼 속성 중 value="" 에 SPICE 시뮬레이션 명령을 추가했다.

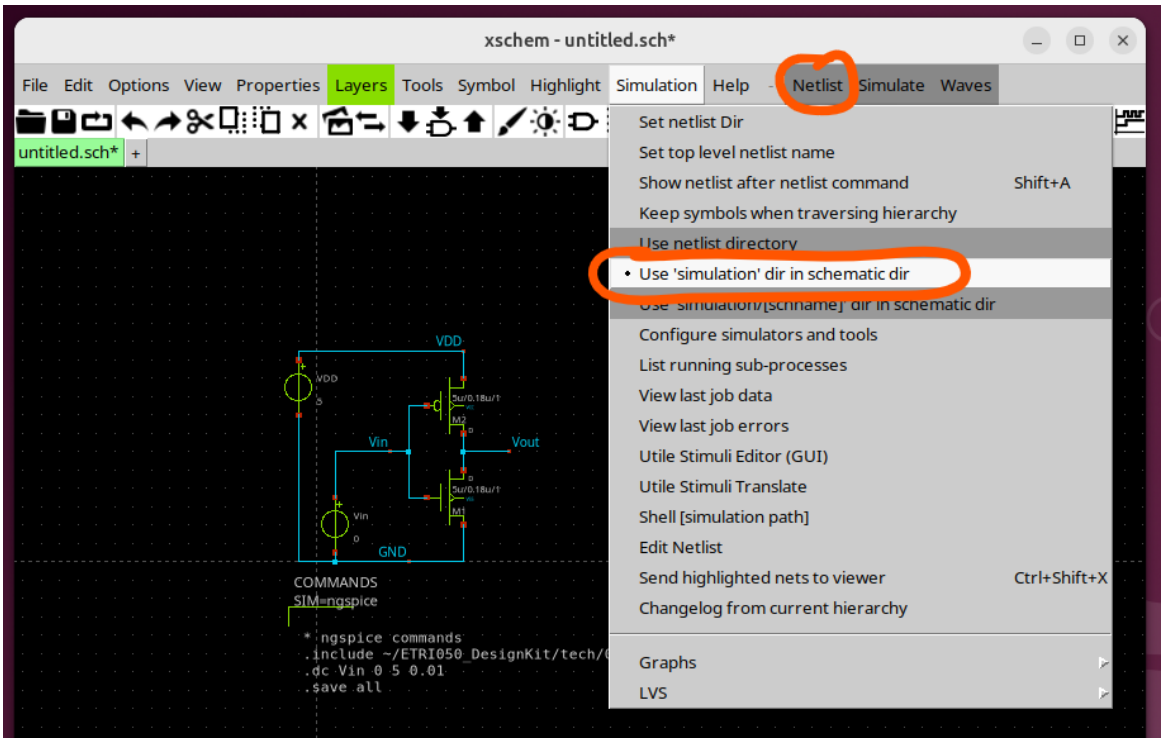
```
.include ~/ETRI050_DesignKit/tech/05cmos_model_240201.lib
.dc Vin 0 5 0.01
.save all
```

NSPL 0.5um CMOS 공정의 mos 트랜지스터 특성 인자들을 불러들이기 위해 .include 명령이 사용되었다. 이어 입력 전압원 vin의 변화를 .dc 로 기술 하였다. 전압원 vin에 대하여 0 볼트에서 5볼트까지 변화 시키는데 증분량은 0.01 볼트라는 뜻이다. 단순한 증감 뿐만 아니라 삼각함수를 포함하여 각종 수학식을 동원 할 수 있다. 입력 신호를 주는 방법을 간략하게 요약한 문서 [SPICE Quick Ref. 5]가 유용하다.

3-2. 네트리스트 추출

시뮬레이션을 하려면 회로도로부터 네트리스트를 추출 해내야 한다. 그림은 인간이 시각적으로 파악하기 좋은 형식일 뿐이다. 컴퓨터 프로그램 시뮬레이터가 회로를 읽으려면 구문으로 표현되어야 한다. SPICE는 이를 위해 간단하지만 정교한 문법 체계를 갖추고 있다. 회로를 구성하는 객체 간의 연결 관계를 문법에 따라 구문으로 표현한 것을 네트리스트(netlist)라 한다. 메뉴 앞서 그려놓은 회로도에서 SPICE 네트리스트를 뽑아낸다. 네트리스트가 저장되는 디렉토리의 위치는 기본적으로 ~/.xschem/simulation 이다. 이를 현재 회로도 저장 위치에서 하위 디렉토리 simulation 으로 바꿔주도록 하자. 메뉴 simulation > Set netlist Dir 을 따라가면 변경 할 수 있다. XSchem의 설정 파일 ~/.xschem/xschemrc 에서 아래 설정변수를 수정해주면 매번 변경하는 수고를 덜 수 있다.

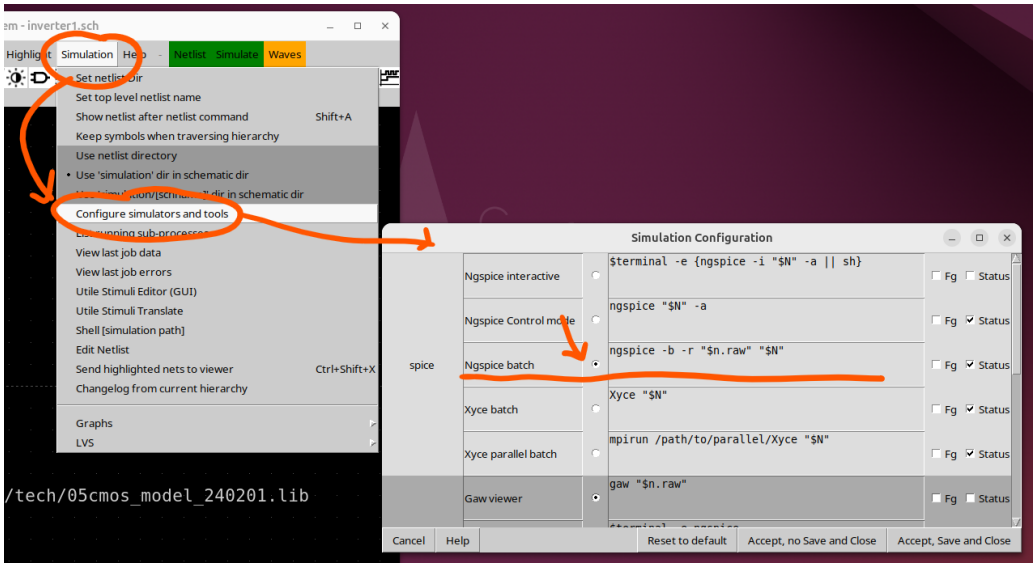
```
set local_netlist_dir 1
```



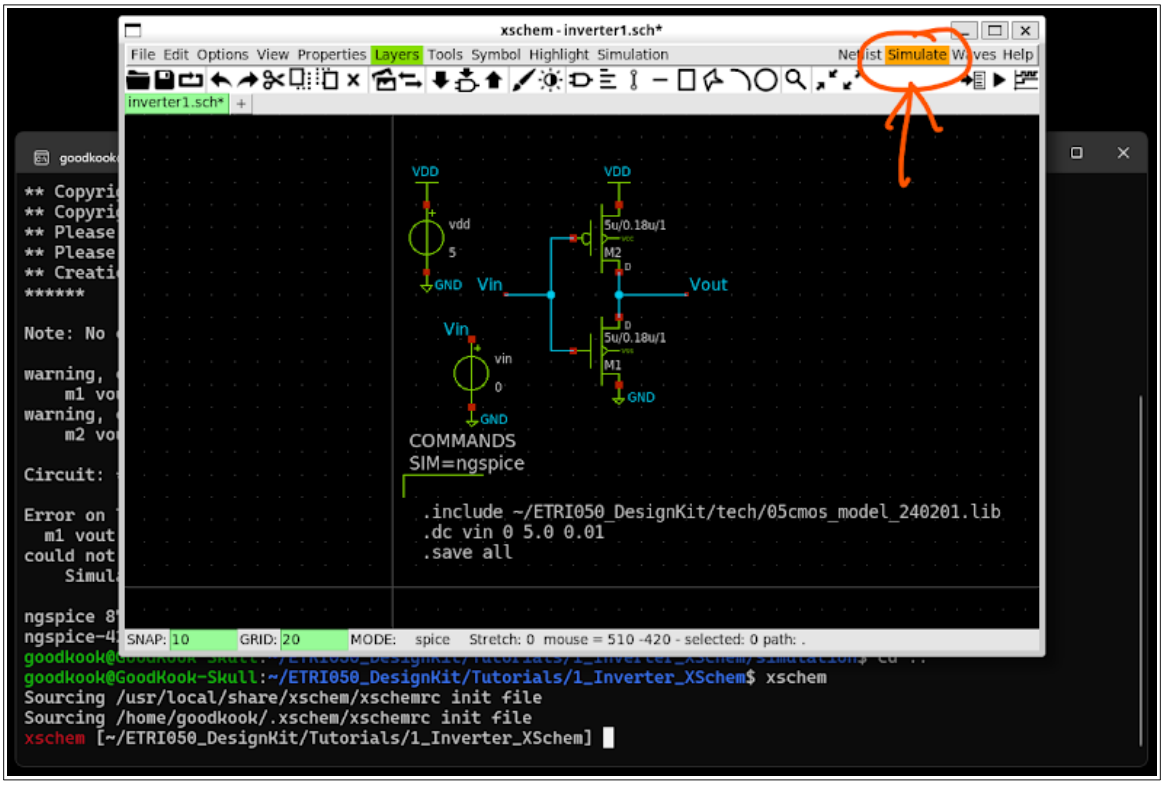
회로가 완성되었으면 메뉴 File > Save As 를 따라가서 'inverter1'으로 저장한다. 이어 메뉴바의 Netlist를 누르면 네트리스트가 현재 디렉토리의 simulation 에 inverter1.spice 로 저장되어 있다.

3-4. 시뮬레이션 실행

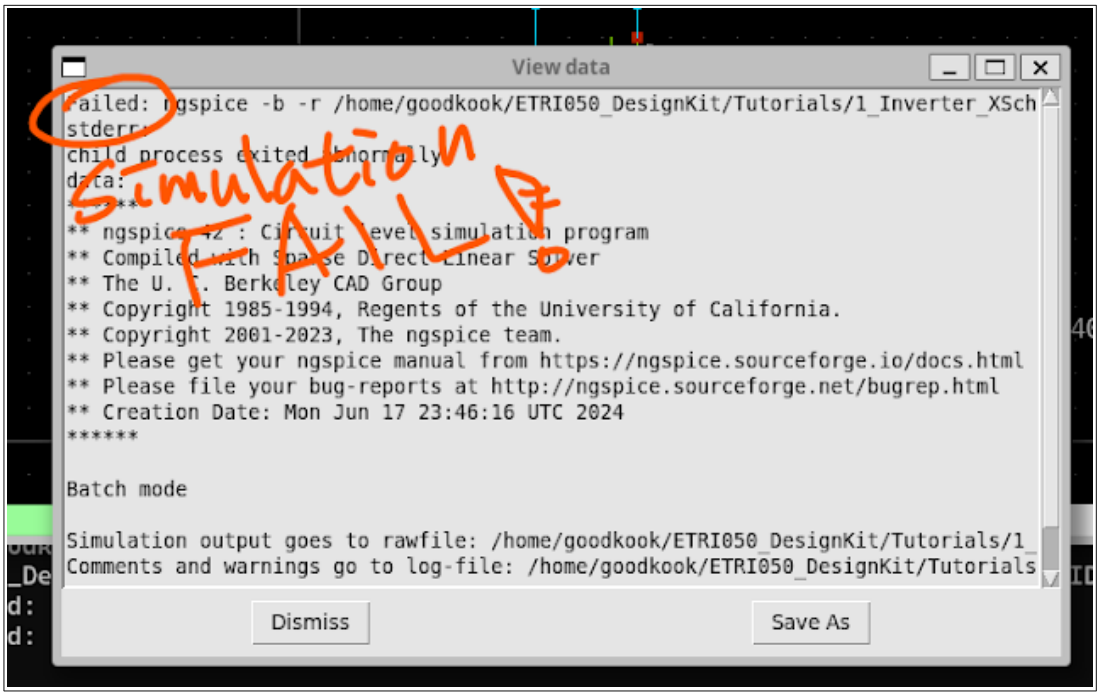
회로도에서 추출된 네트리스트를 읽어 SPICE 시뮬레이션을 수행한다. 시뮬레이션 도구가 ngSpice로 지정되어 있는지 확인 한다. 메뉴 Simulation > Configure simulators and tools 를 선택하면 시뮬레이터 설정창이 뜬다. XSchem은 ngspice를 띄울 때 대화형(interactive) 모드가 기본이다. 이를 배치 모드(batch)로 바꿔준다.



지정한 도구로 시뮬레이션 수행을 위해 메뉴에서 Simulate를 누른다.



아래와 같이 시뮬레이션 실패라는 보고를 받게 될지도 모른다. 좌절하지 말고 이유를 찾아 고쳐보자.



시뮬레이션 실행의 실패는 도구가 만들어 준 네트리스트에 문법 오류가 원인이므로 회로도를 수정해야 한다. 오류의 원인을 찾기에는 XSchem 에서 보여주는 정보가 너무 적다. ngSpice를 명령줄에서 실행시켜서 문제를 찾아보자. 새 터미널을 열어 네트리스트가 있는 폴더로 이동한다.

```
$ cd ~/ETRI050_DesignKit/Tutorials/1-1_Inverter_XSchem/simulation
```

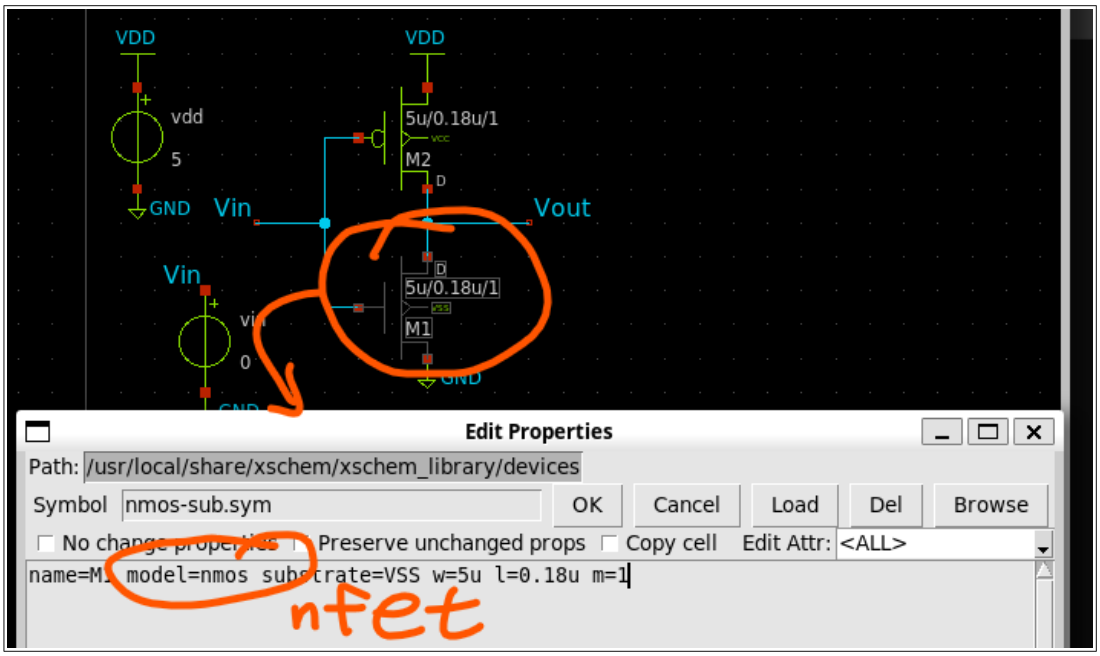
XSchem이 생성한 네트리스트는 회로도 이름에 .spice 가 붙는다.

```
$ ll
-rw-r--r-- 1 goodkook goodkook   416 Jul 13 21:36 inverter1.spice
```

ngSpice를 명령줄에서 실행시켜 보자.

```
$ ngspice inverter1.spice
....
warning, can't find model 'pmos' from line
    m2 vout vin vdd vcc pmos w=5u l=0.18u m=1
Circuit: ** sch_path:
   ....
Error on line 4 or its substitute:
  m2 vout vin vdd vcc pmos w=5u l=0.18u m=1
could not find a valid modelname
    Simulation interrupted due to error!
ngspice 87 ->
```

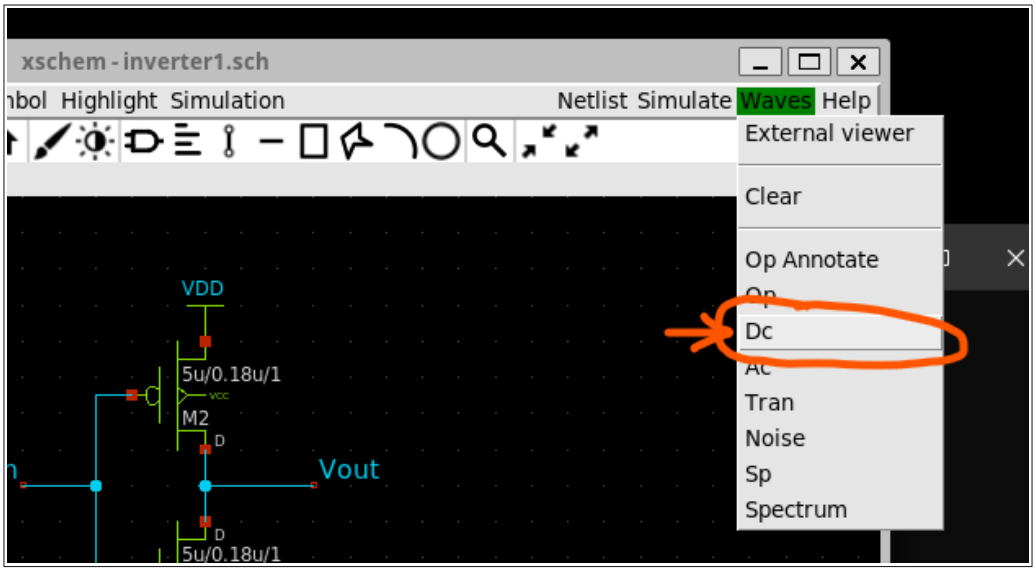
원인은 쉽게 찾아졌다. mos 트랜지스터의 모델 이름이 일치하지 않았기 때문이다. '디자인 킷'의 라이브러리 트랜지스터 모델명이 'nfet' 과 'pfet' 인데 회로도에는 'nmos'와 'pmos' 였다.



모델명을 수정하고 다시 네트리스트를 생성하여 시뮬레이션을 수행할 수 있을 것이다.

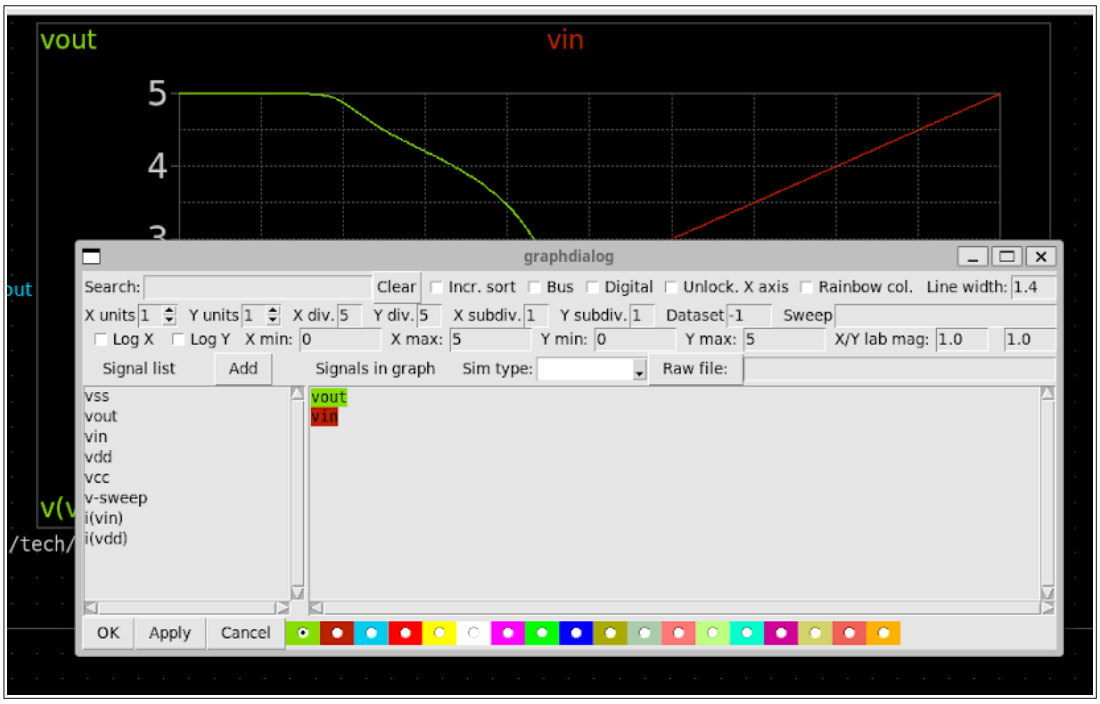
3-4. 결과 파형 관찰

시뮬레이션 결과를 파형으로 관찰한다. 시뮬레이션 결과는 .raw 파일에 저장된다. 시뮬레이션 명령이 DC 변화 관찰이었으므로 해당 양식으로 읽는다.

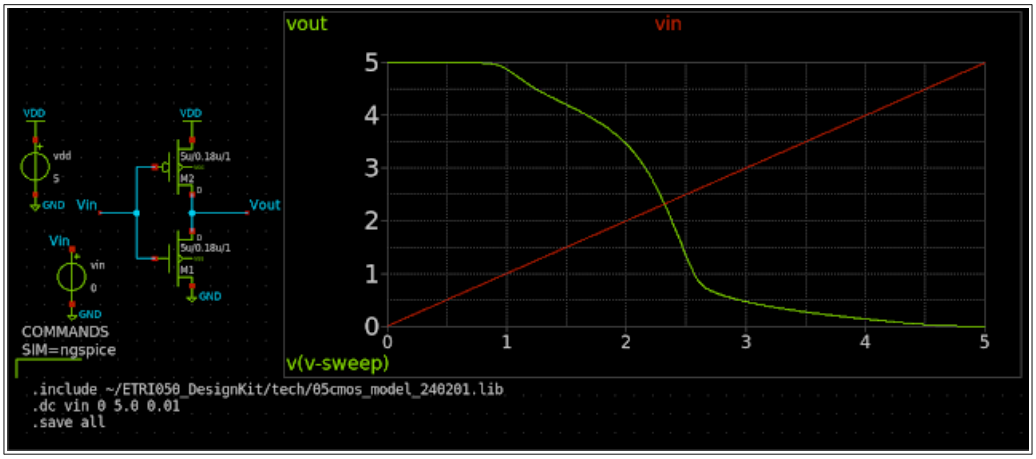


XSchem의 도면위에 파형을 표시할 수 있다. 메뉴 Simulation > Graphs > Add waveform graph 를 선택하자. 이어서 그래프 객체를 더블 클릭하여 그래프 대화창(graph dialog)에서 원하는 신호를 선택한다. 파형보기는 이외에도 다양한 기능을 가지고 있다[8].

마우스 포인터를 그래프 축에 올려 놓으면 커서가 십자 모양으로 바뀐다. 이때 왼쪽 버튼을 더블-클릭 하면 그래프에 표시할 신호들을 설택 할 수 있는 대화창이 나타난다. 그래프 상에서 마우스 포인터를 가로축(또는 세로축)에 놀려 커서가 십자 모양일때 'f' 키를 누르면 전체 표시된다.



그나저나 인버터의 출력 모습이 좀 괴이하다.



4. 결론

반도체 설계는 제조도면(Layout)을 그리기 전에 회로 설계와 검증이 선행되어야 한다. 회로도 작성과 회로 시뮬레이션 도구의 사용법에 대하여 간략하게 살펴봤다. 계층적 회로 구성도 가능한 XSchem은 상용도구 못지않은 기능을 갖추고 있다. 참고 [9]의 동영상 교재를 통해 더 많은 내용을 배울 수 있을 것이다. 다음 편은 Magic 도구로 레이아웃 그리기와 LVS 다.

[부록] XSchem 실습: 계층적 회로도 그리기와 시뮬레이션 테스트벤치

목차

I. XSchem: 인버터 회로도

인버터의 회로도를 작성한다. 재사용을 위해 하위회로(sub-circuit) 형식으로 작성한다. XSchem 도구는 계층화된 회로도 그리기를 할 수 있다.

디자인 킷의 예제 디렉토리로 이동,

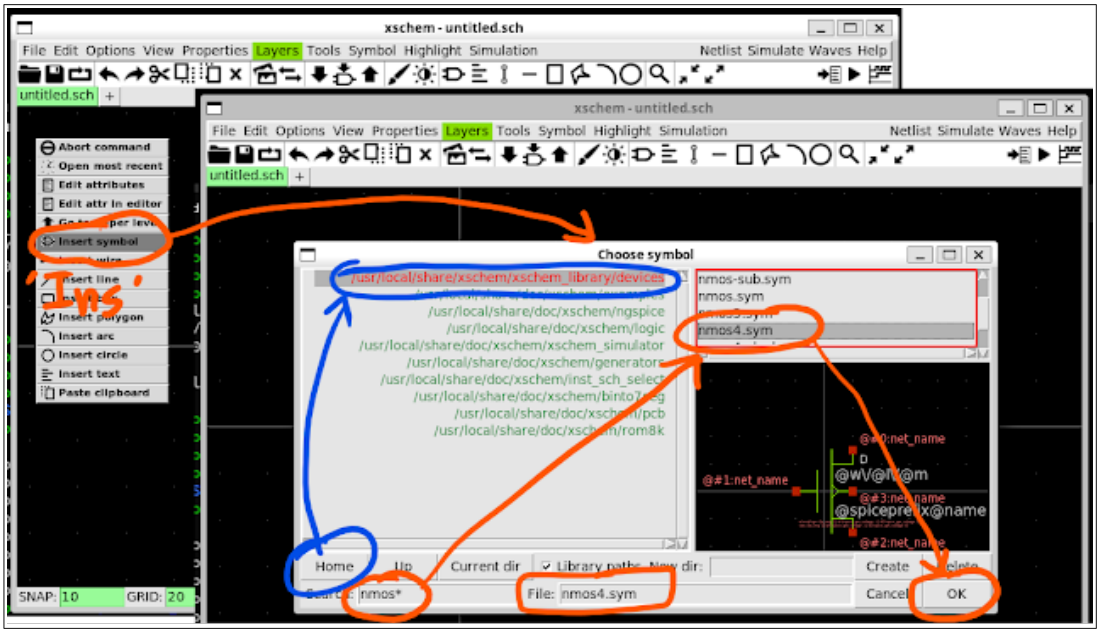
```
$ cd ~/ETRI050_DesignKit/Tutorials/1-1_Inverter_XSchem
```

I-2. 회로도 그리기

(1) 심볼 불러오기

인버터 회로에 사용될 소자 nmos4, pmos4의 심볼을 불러온다.

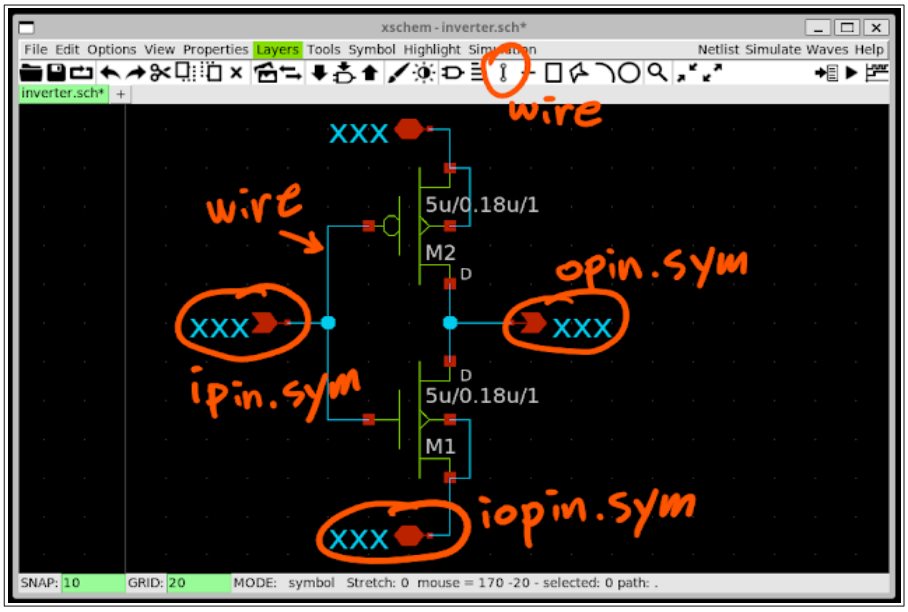
- 단축 키 'Ins' 또는 그리기 바탕에서 마우스 오른쪽 버튼 클릭 후 플로팅 메뉴에서 'Insert symbol' 선택
- 내장 심볼의 위치는 'Home'으로 지정된 디렉토리



(2) 인버터 회로도 작성

회로도를 그리기위해 불러온 소자의 심볼을 재치하고 배선하여 인버터 회로도를 작성한다.

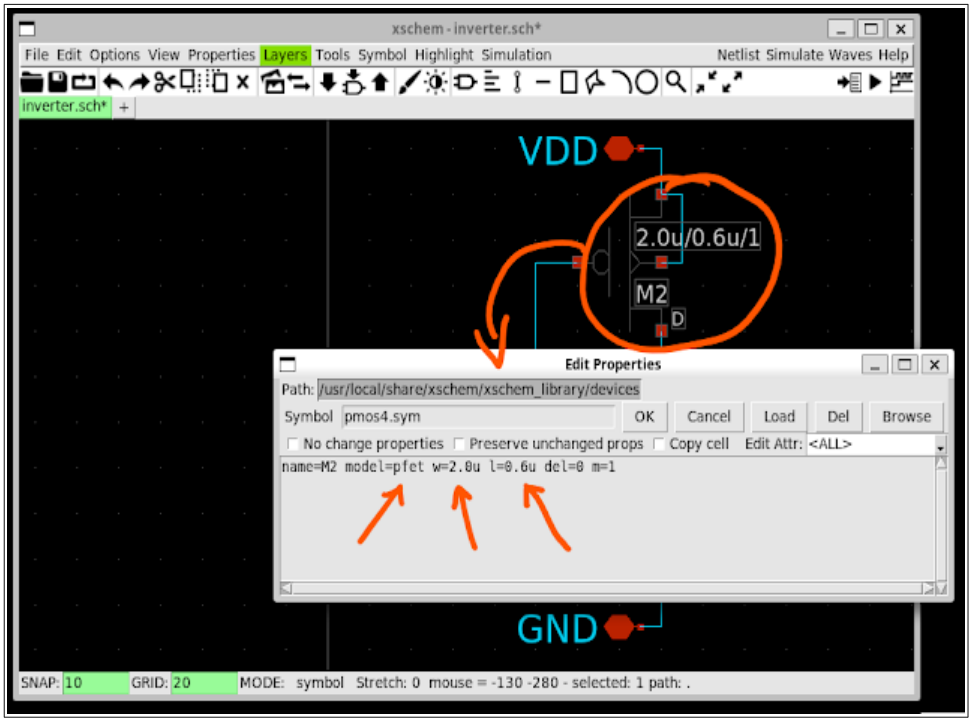
- 두 소자 nmos4 와 pmos4 배치 후 배선
- 배선은 단축키 'w'
- 재사용 가능한 하위 회로(sub-circuit)가 되려면 입출력 핀 심볼 ipin.sym, opin.sym 을 붙여야 함



(3) 소자의 속성

- 트랜지스터 소자의 속성 값 수정

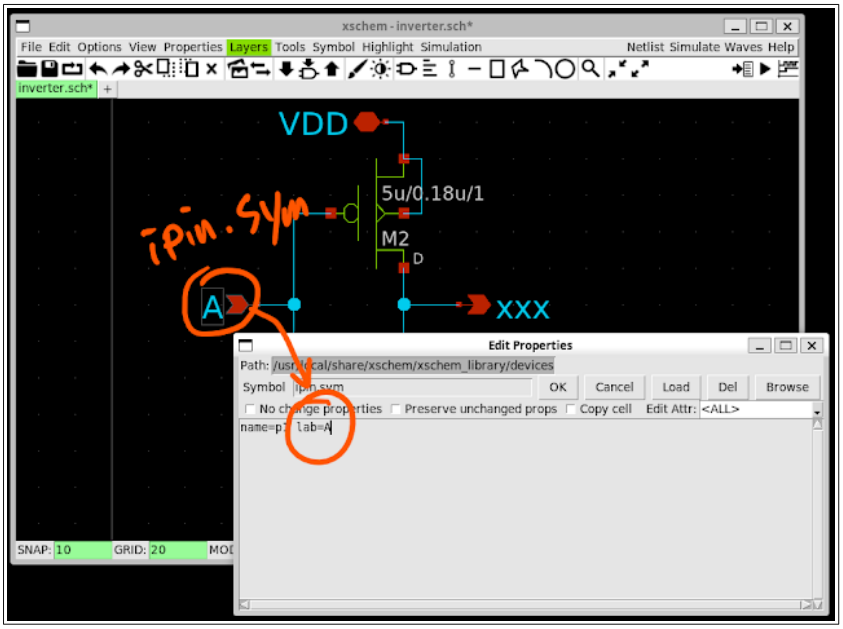
NSPL 0.5um 공정의 레이아웃 규정에 맞게 트랜지스터 채널의 폭(width)과 길이(length)를 수정한다. 트랜지스터의 모델명을 PDK에서 제공하는 물리 모델명과 일치시킨다. XSchem의 트랜지스터 이름은 pmos, nmos 다. 이를 NSPL의 모델명 nfet, pfet로 수정해 주어야 한다.



(4) 입출력 핀 속성

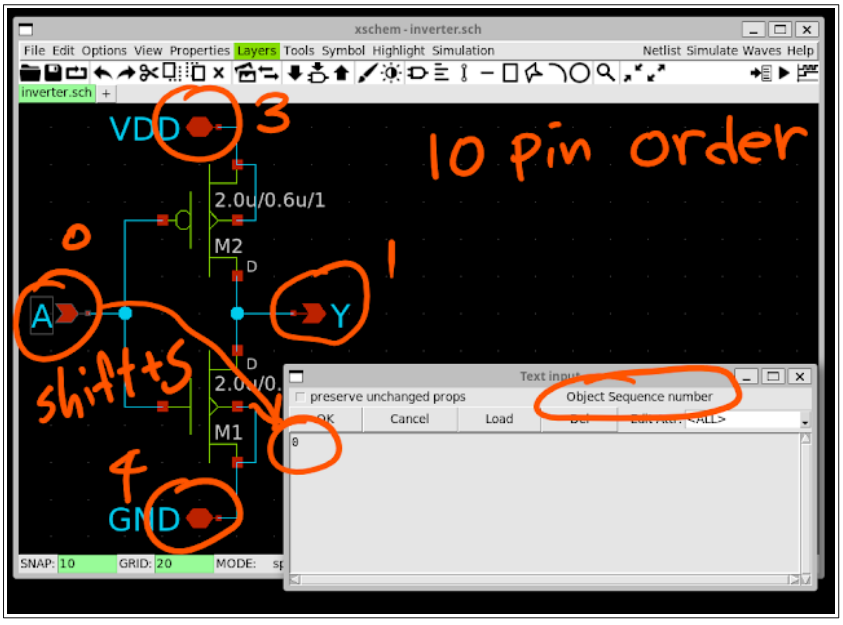
하위회로(sub-circuit)에서 외부로 노출될 핀(포트) 속성(이름 및 방향)을 지정한다.

- 입출력 핀 ipin.sym, opin.sym 을 더블 클릭하여 속성(이름) 변경
- 전원은 양방향 핀 심볼 iopin.sym 사용



(5) 입출력 핀 순서(핀 번호)

회로도에 입출력 핀의 순서를 지정한다. 외부 설계 자동화 도구들이 하위 회로를 다룰 때 입출력 포트 매핑(바인딩)시 순서를 따르는 경우를 대비해야 한다. 상위 수준의 설계 도구들, 예를 들어 베릴로그는 이름 매핑(named mapping)을 권장하지만 SPICE의 네트리스트는 하위 회로 연결(배선)시 위치 매핑(positional mapping)만 지원 한다.

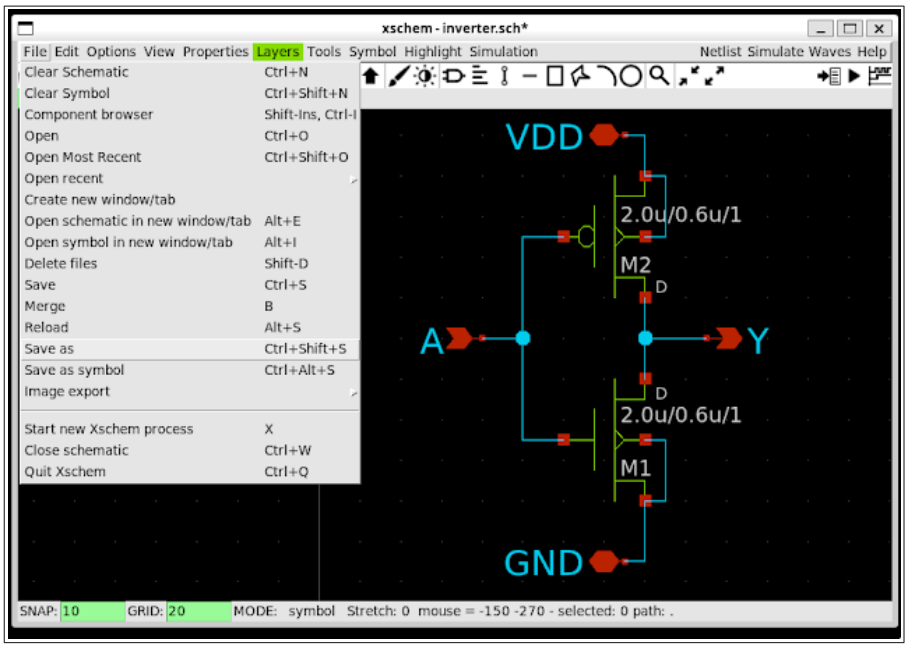


편의 상 입력, 출력 그리고 전원 핀의 순서를 매겨 두도록 한다. 입출력 핀을 선택 한 후 Shift+S를 눌러 순서를 0 번부터 매기도록 한다.

(6) 회로도 저장

- 메뉴: File > Save As

회로도를 저장할 때 파일 명이 회로 명이 되므로 유의한다.



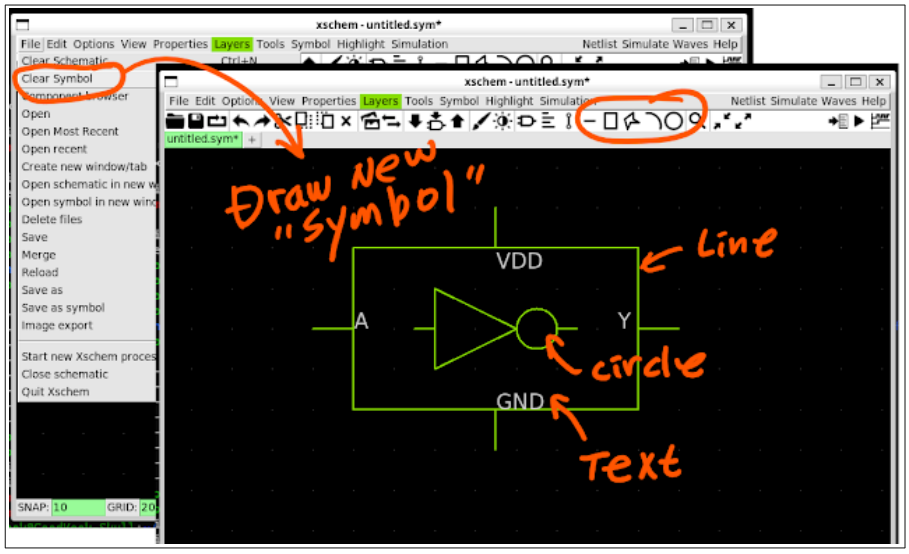
I-2. 심볼 그리기

(1) 심볼 그림

상위 회로에서 사용될 인버터의 심볼을 그린다. 심볼은 상위 회로에서 표시될 대표 표시(상징)일 뿐 회로 자체와는 관련 없다. 그림을 그리는 도구는 선(line), 원(circle), 문자(text)등을 사용하여 보기 좋게 그린다.

- 메뉴: File > Clear Symbol

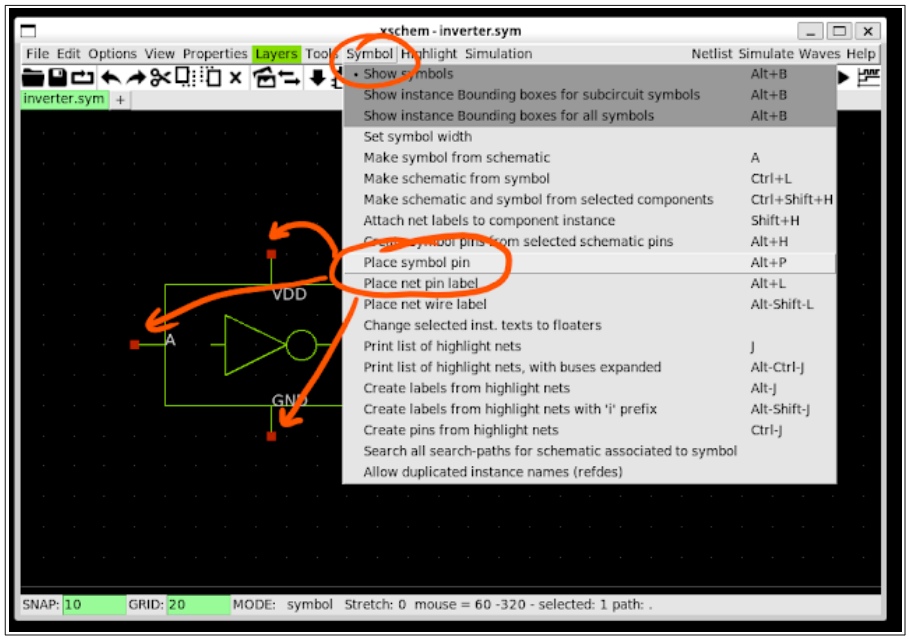
[주] 메뉴 명에 New Symbol이 아닌 Clear Symbol로 되어 있다.



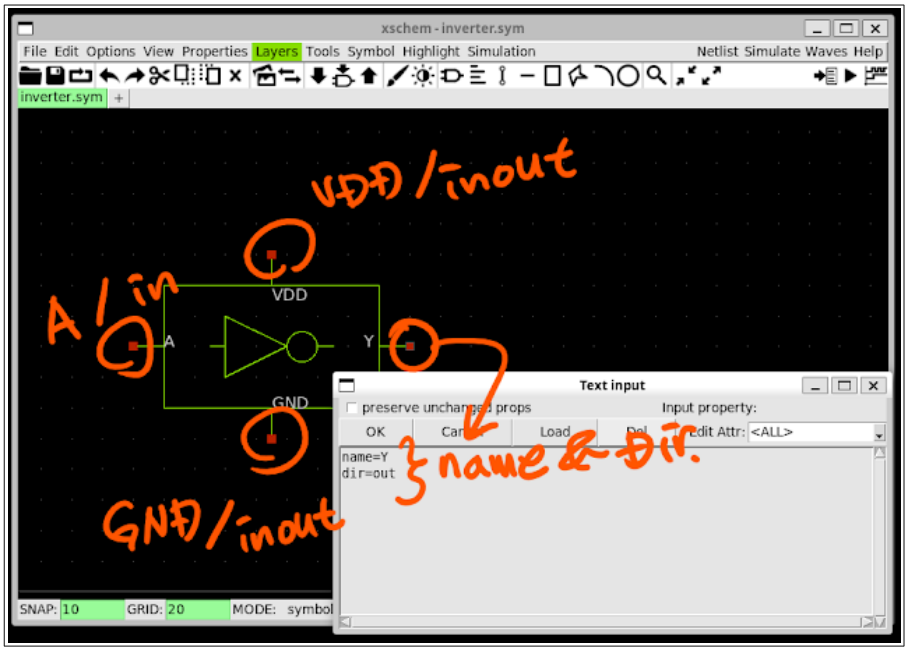
(2) 심볼 핀 배치

심볼 핀(symbol pin)은 하위 회로에서 노출되는 포트다.

- 메뉴: Symbol > Place symbol pin
- 적절한 위치에 하위 회로의 입출력 핀을 배치

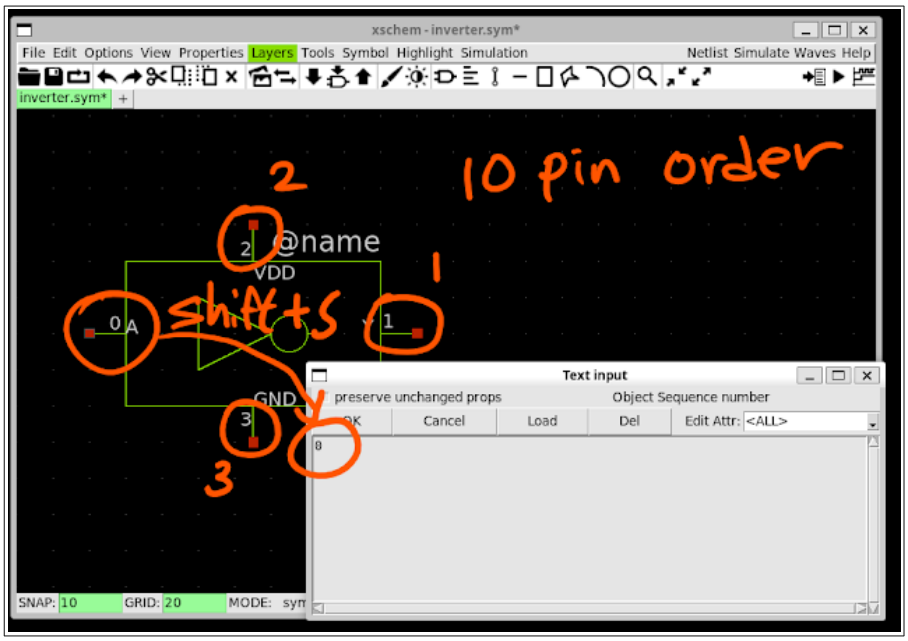


- 하위 회로의 입출력 핀에 대응 하도록 심볼에 핀을 배치
- 핀의 이름(name)과 입출력 방향(dir)을 맞출 것



(3) 핀 번호(순서)

하위 회로도의 핀 번호(순서)와 일치 시킨다.



네트리스트를 생성할 때 의 하위 회로(sub-circuit)의 입출력 포트의 위치(순서)에 영향을 주므로 주의할 것

```
** sch_path: ~/Tutorials/1-1_Inverter_XSchem/inverter.sch
.subckt inverter A Y VP VN
*.ipin A
```

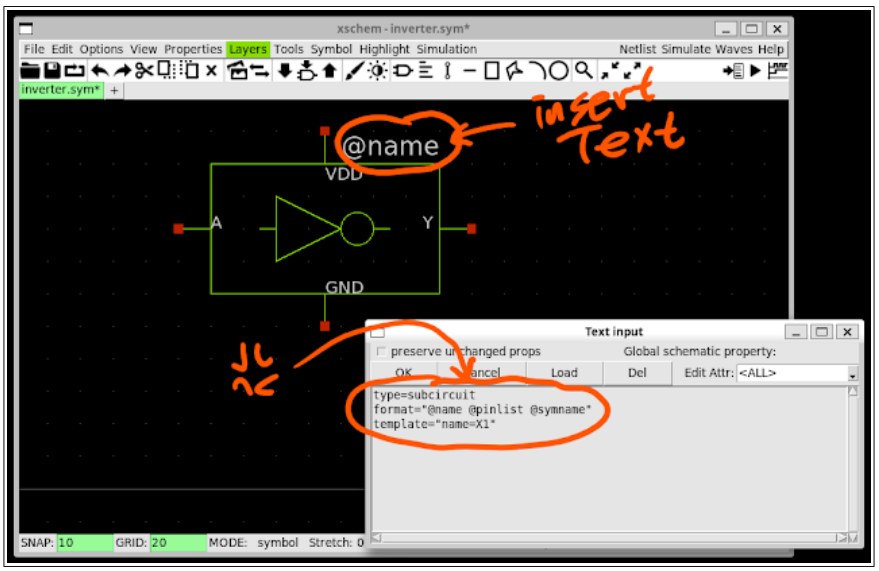
```
*.opin Y
*.iopin VP
*.iopin VN
M1 Y A VN VN nfet w=5u l=0.18u m=1
M2 Y A VP VP pfet w=5u l=0.18u m=1
.ends
```

(4) 심볼의 속성 부여

상징의 역활을 속성에 정의한다. 회로도 바탕을 더블-클릭을 하여 속성 정의 창에 아래와 같이 입력한다.

```
type=subcircuit
format="@name @pinlist @symname"
template="name=X1"
```

해당 심볼은 하위회로(subcircuit)를 대표하는 것으로 인스턴스 명(name), 입출력 핀의 목록(pinlist) 그리고 심볼 이름(symname) 등 세가지 속성을 가진다고 명시하고 있다. 심볼의 이름은 하위 회로의 회로도 파일명과 동일하다. inverter.sch의 심볼은 inverter.sym 이다. 아울러 상위회로에서 사례화 되면서 자동으로 이름이 붙도록 문자(insert text)를 추가하였다. @는 변수의 역활을 한다. 심볼 속성에서 name 붙이는 방식을 X1으로 지정되었다. 이 심볼이 사례화 되면 X 문자 뒤로 숫자가 붙는다.



II. 테스트 벤치

(1) 테스트 벤치 작성

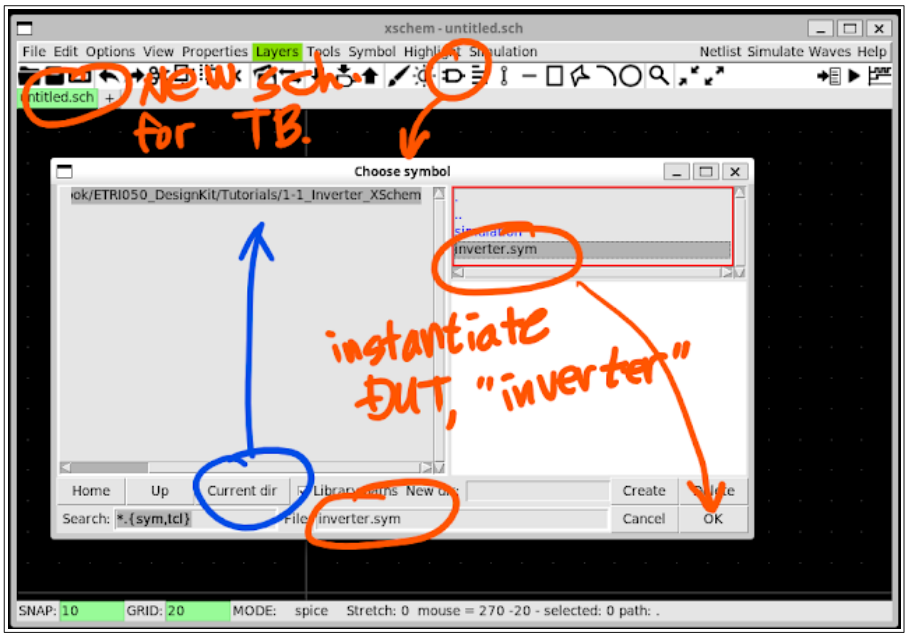
시뮬레이션을 위한 테스트 벤치를 작성한다.

- 메뉴: File > Clear Symbol

  메뉴 명에 New Symbol이 아닌 Clear Symbol로 되어 있다.

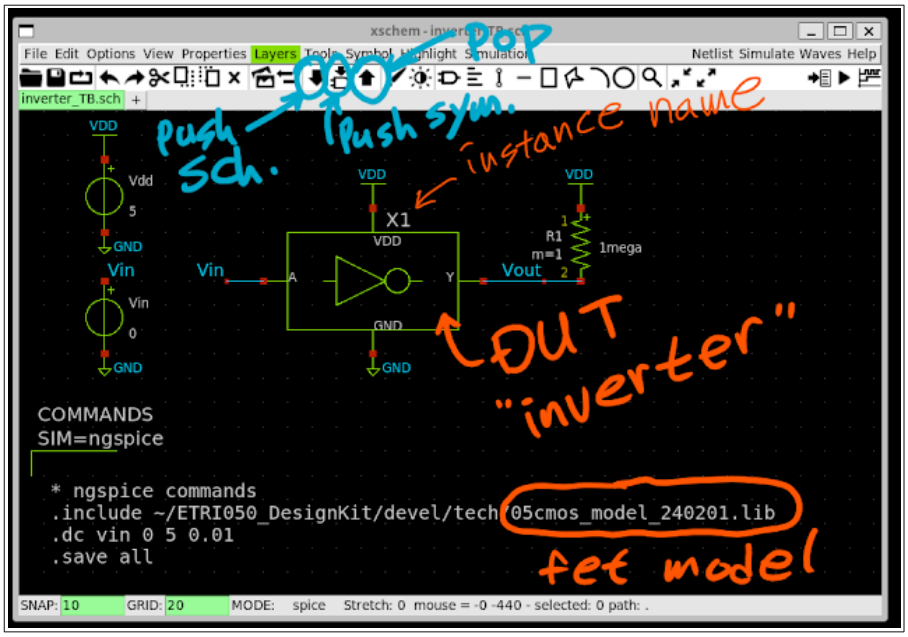
- DUT에 해당하는 'inverter'의 심볼을 불러와 사례화(instantiate) 한다.

앞서 작성해 놓은 'inverter.sym' 은 현재 작업 디렉토리에 저장되었으므로 'Current Dir'을 클릭



(2) 계층적 회로도

사례화 된 심볼을 푸쉬(push)하면 심볼(push symbol) 또는 하위 회로(push schematic)로 내려간다. 하위회로에서 상위 회로로 팝(pop)도 가능하다. 상위 회로도(테스트 벤치)에 하위 회로에 사용된 소자의 모델의 포함되어야 한다. PDK의 모델을 불러오는 경로와 파일명이 지정되었다.

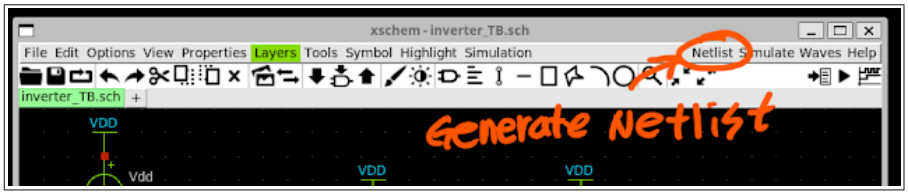


출력에 상당히 큰 용량(1M ohm)의 저항을 달아서 전자 회로가 플로팅 되지 않도록 한다. 테스트벤치 회로도를 inverter_TB.sch 로 저장한다.

III. SPICE 시뮬레이션

(1) 시뮬레이션용 네트리스트 생성

오픈-소스 SPICE 시뮬레이션 도구는 ngSpice다. 회로도는 인간이 읽고 쓰기 좋은 그림이다. 시뮬레이션 소프트웨어는 엄격한 문법 체계에서 표현된 네트리스트를 통해 회로를 이해한다. 검증을 위해 시험 대상 DUT(Design Under Test)이 사례화 되어 있는 테스트 벤치의 네트리스트 추출한다.



생성된 네트리스트에 DUT 회로 inverter.sch 가 하위 회로(sub-sircuit)로 포함된다.

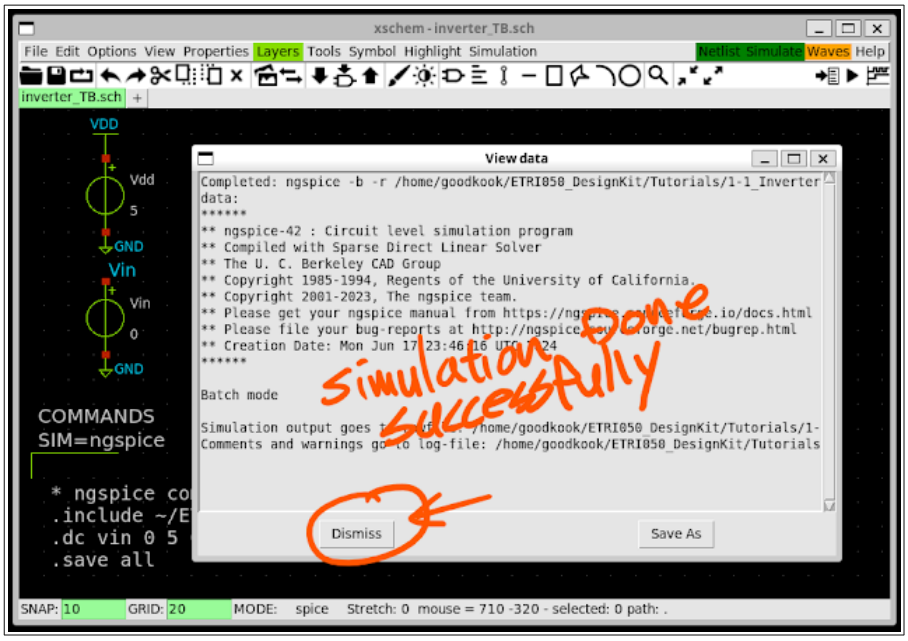
```
** sch_path: ~/Tutorials/1-1_Inverter_XSchem/inverter_TB.sch
**.subckt inverter_TB
X1 Vin Vout VDD GND inverter
Vdd VDD GND 5
Vin Vin GND 0
R1 VDD Vout 1mega m=1
**** begin user architecture code
* ngspice commands
.include ~/ETRI050_DesignKit/devel/tech/05cmos_model_240201.lib
.dc vin 0 5 0.01
.save all
**** end user architecture code
**.ends
* expanding symbol: ~/Tutorials/1-1_Inverter_XSchem/inverter.sym #
of pins=4
** sym_path: ~/Tutorials/1-1_Inverter_XSchem/inverter.sym
** sch_path: ~/Tutorials/1-1_Inverter_XSchem/inverter.sch
.subckt inverter A Y VDD GND
*.ipin A
*.opin Y
*.iopin GND
*.iopin VDD
M1 Y A GND GND nfet w=2.0u l=0.6u m=1
M2 Y A VDD VDD pfet w=2.0u l=0.6u m=1
.ends
.GLOBAL VDD
.GLOBAL GND
.end
```

테스트벤치 회로도에서 추출한 SPICE 네트리스트를 보면 DUT를 하위 회로로 사례화 한 후 입출력 포트를 위치 매핑 하고 있다. 계층적 회로 설계에서 다수의 입출력을 갖는 모듈의 포트 매핑의 일관성 유지는 매우 중요하다.

(2) 시뮬레이션 수행

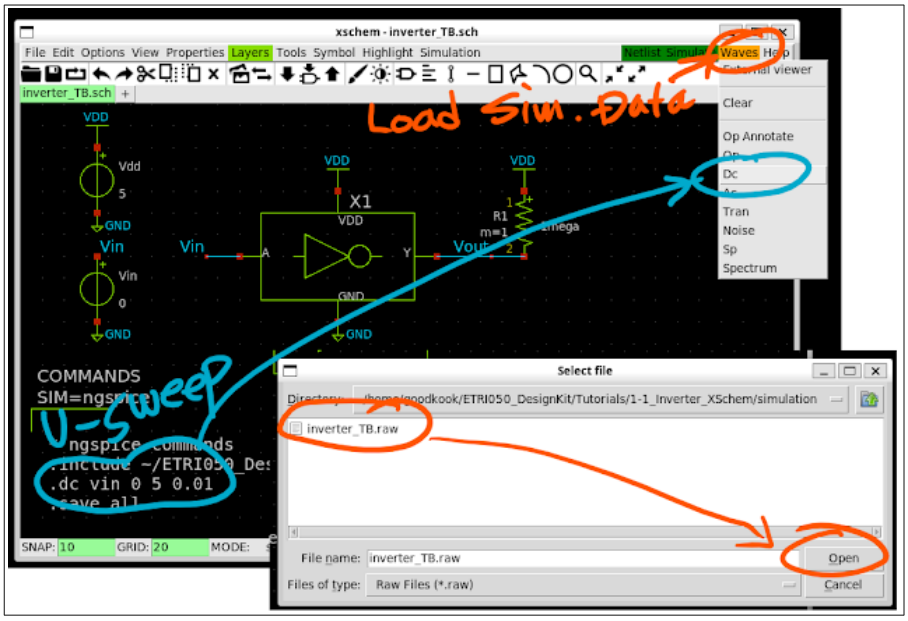
회로도에서 추출한 네트리스트를 읽어 회로 시뮬레이션을 실행한다.

단순한 회로지만 시뮬레이션에 시간이 걸린다.

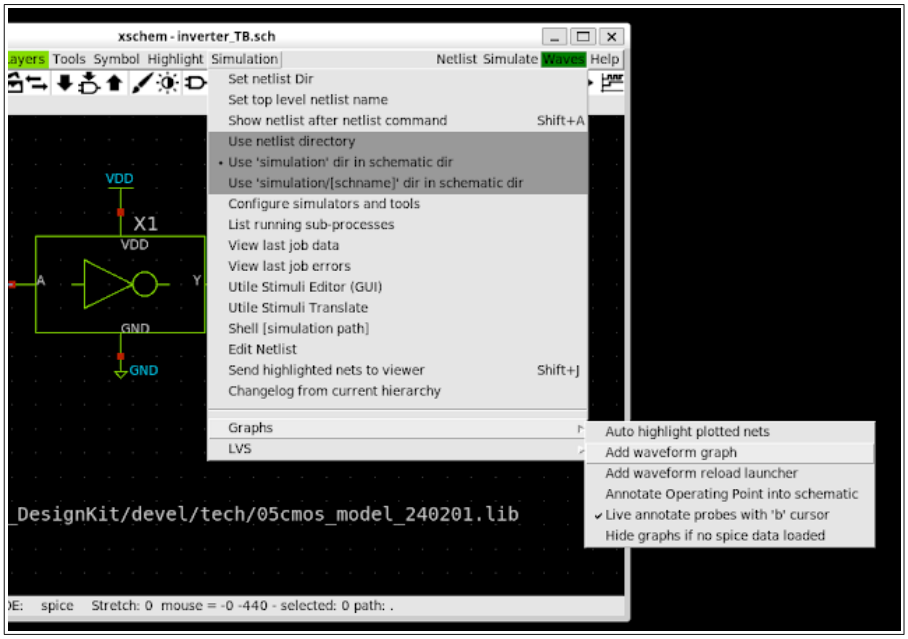


(3) 시뮬레이션 결과 보기

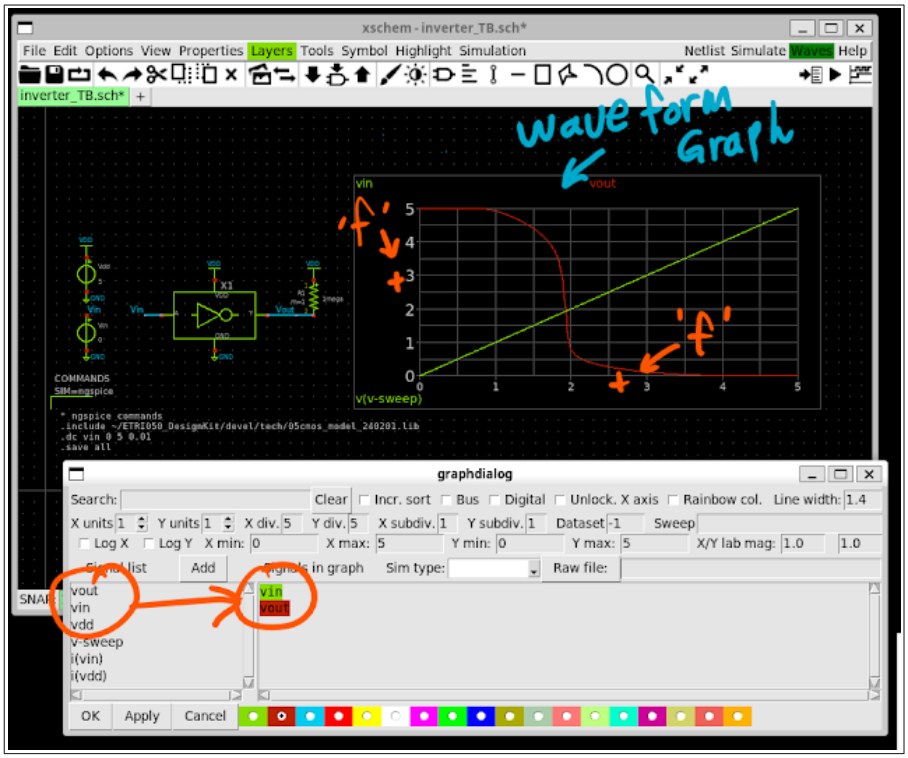
입력 vin의 전압을 점차 증가 시키면서(voltage sweep) 출력을 관찰한다. 회로의 모든 지점에서 회로의 물리량(전압과 전류)를 기록하기 위해 .save all 명령이 사용되었다.



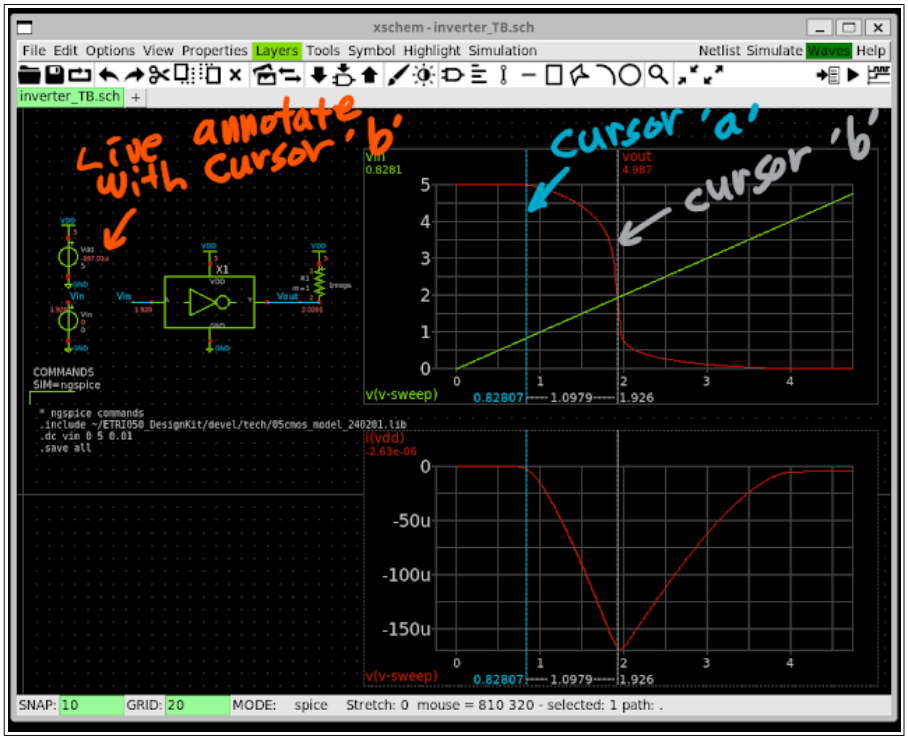
- 파형 보기는 메뉴: Simulation> Graphs > Add waveform graph



- 관찰할 신호 선택
- 파형 전체보기: 가로와 세로축에 마우스 커서를 위치한 후 'f'



- 전압과 전류 변화 관찰
- 파형에 두개의 커서를 넣을 수 있음('a' 커서와 'b' 커서).
- 'b' 커서를 파형에서 움직이면 해당 값이 실시간으로 회로도에 표시됨



[참조]

[1] 오픈-소스 반도체 설계도구 설치,

[2] NSPL 0.5um CMOS 디자인 킷, https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK

[3] 내 칩(My Chip) MPW 제작 서비스, http://mpw.kion.or.kr/info/notice_list.asp

[4] SPICE 교육 동영상으로 Fesz Electronics 채널의 LTSpice tutorial을 추천한다.
https://www.youtube.com/watch?v=JRcyHuyb1V0&list=PLT84nve2j1g_wgGcm0Bv3K4RSl2Jdjsey

[5] SPICE 'Quick' Reference Sheet, https://web.stanford.edu/class/ee133/handouts/general/spice_ref.pdf

[6] ROHM Simulation-SPICE, https://techweb.rohm.com/product/simulation/7688/

[7] HSPICE Tutorial, http://www.hkn.umn.edu/resources/files/spice/

[8] xschem displaying simulation waveforms without using external tools., https://youtu.be/bP9w3zm1qv4?si=AjpN9Z4OSgCKeaZt

[9] MADVLSI Tutorials, https://www.youtube.com/playlist?list=PLgsDG5BJZpBTEUaxjfvYUiMPpUPU_vQpr

[10] XSchem Manual, https://xschem.sourceforge.io/stefan/xschem_man/xschem_man.pdf