

GIGZO TFT PDK User Manual

GIST 반도체소자 시뮬레이션 연구실(SDSL)

GIGZO TFT PDK ver.2 (25.12.18)

목차

1 소개

1.1 구성

1.1.1 Helvellyn_2.1.0.beta

1.1.2 BSIMSOI

1.2 주요 기능 소개

2 사용 방법

2.1 GIGZO PDK 설치 및 환경 설정

2.1.1 Github repository를 통한 설치

2.1.2 cds.lib 환경 설정

2.1.3 .bashrc 설정

2.2 Virtuoso 실행 및 Library manager 접속

2.2.1 Virtuoso 실행

2.2.2 Library Manager 접속

2.3 IGZO TFT 단일 소자에 관한 시뮬레이션 예제 ($VDD=0.1V$, $V_{Gate}=-3V \sim 3V$ sweep)

2.3.1 ~ 2.3.3 단계별 절차

2.4 IGZO TFT 소자를 활용한 6T1C 시뮬레이션 ($VDD=0.1V$)

2.4.1 ~ 단계별 절차

2.5 3x3 어레이 시뮬레이션

2.5.1 ~ 단계별 절차

3 참고 자료

1 소개

이 문서는 GIST 반도체소자 시뮬레이션 연구실(SDSL)에서 개발한 GIGZO TFT PDK (Process Design Kit)의 목적과 사용 방법을 설명하기 위한 사용자 매뉴얼입니다.

본 PDK는 **Helvellyn 2.1.0.beta** 프레임워크와 **BSIMSOI Compact Model**을 기반으로 구축되었으며, **Linux 기반 Cadence Virtuoso** 환경에서 IGZO 박막 트랜지스터(Thin-Film Transistor)의 전기적 특성을 회로 수준에서 시뮬레이션할 수 있도록 지원합니다.

IGZO TFT PDK는 다음과 같은 목적을 가지고 개발되었습니다.

1) IGZO 기반 소자의 DC 및 Transient 특성을 회로 시뮬레이션 분석

2) BSIMSOI Compact Model 기반 어레이 시스템 모델링 및 시뮬레이션

- 기존 CMOS 공정 기반 PDK 와의 호환성 유지(SPICE)를 위한 BSIMSOI compact model 활용
- **6T1C synapse** 구조 등 다양한 회로 설계 검증을 위한 예제 제공
- 어레이 모델링 및 시뮬레이션 테스트 환경 구축

3) Layout DRC/LVS 검증

본 매뉴얼은 GIGZO TFT PDK의 설치 방법, 환경 설정(.bashrc, cds.lib), Virtuoso 실행 및 시뮬레이션 절차, 그리고 대표적인 단일 TFT 및 6T1C 회로 시뮬레이션 예제를 단계별로 설명합니다. 이를 통해 사용자는 IGZO TFT의 소자 특성 평가뿐만 아니라, 회로 레벨 응용(예: 메모리, 뉴로모픽 소자의 동작 검증까지 수행할 수 있습니다.

현재 버전은 초기 버전으로 DRC/LVS에 관한 부분은 추후 업데이트 예정.

1.1 구성

1.1.1 Helvellyn_2.1.0.beta

1.1.2 BSIMSOI

1.2 구현된 주요 기능 소개

1) Width/length 10um, channel thickness 15nm~30nm 범위의 IGZO TFT 소자 시뮬레이션

A. BSIMSOI compact model + TCAD calibration

2) 6T1C synapse 구조를 통한 retention time 시뮬레이션

2 사용 방법

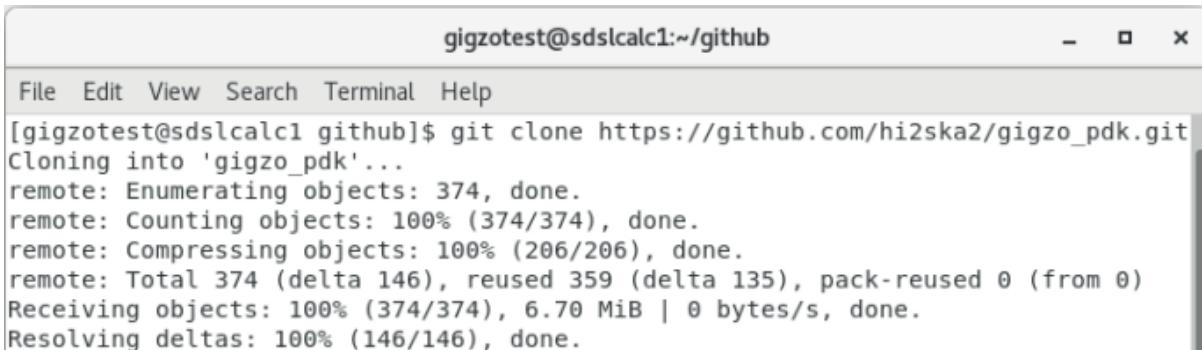
2.1 GIGZO PDK 설치 및 환경 설정

2.1.1 Github repository를 통한 설치

https://github.com/hi2ska2/gigzo_pdk.git 다운로드 또는

리눅스 터미널 창에서

git clone https://github.com/hi2ska2/gigzo_pdk.git 입력 후 엔터



```
gigzotest@sdsldcalc1:~/github
File Edit View Search Terminal Help
[gigzotest@sdsldcalc1 github]$ git clone https://github.com/hi2ska2/gigzo_pdk.git
Cloning into 'gigzo_pdk'...
remote: Enumerating objects: 374, done.
remote: Counting objects: 100% (374/374), done.
remote: Compressing objects: 100% (206/206), done.
remote: Total 374 (delta 146), reused 359 (delta 135), pack-reused 0 (from 0)
Receiving objects: 100% (374/374), 6.70 MiB | 0 bytes/s, done.
Resolving deltas: 100% (146/146), done.
```

2.1.2 cds.lib 환경 설정 (*사용자 Cadence 버전 및 폴더 위치에 따라 설정*)

```
DEFINE analogLib $(compute:THIS_TOOL_INST_ROOT)/tools/dfII//etc/cdslib/artist/analogLib
DEFINE basic $(compute:THIS_TOOL_INST_ROOT)/tools/dfII/etc/cdslib/basic
DEFINE sheets $(compute:THIS_TOOL_INST_ROOT)/tools/dfII/etc/cdslib/sheets/US_8ths
DEFINE pragLib /home/"사용자 경로"/IGZO_PDK/V2.1.0.beta.2.hotfix.1/pdk/cdslib/pragLib
DEFINE lib_common /home/"사용자 경로"/IGZO_PDK/V2.1.0.beta.2.hotfix.1/pdk/cdslib/lib_common
DEFINE avTech /tools/CADENCE/ASSURA415/tools.lnx86/assura/etc/avtech/avTech
```

(예시)



```
cds.lib
~/github/gigzo_pdk/IGZO_PDK
Save ×
Open ↗
DEFINE analogLib $(compute:THIS_TOOL_INST_ROOT)/tools/dfII//etc/cdslib/artist/analogLib
DEFINE basic $(compute:THIS_TOOL_INST_ROOT)/tools/dfII/etc/cdslib/basic
DEFINE sheets $(compute:THIS_TOOL_INST_ROOT)/tools/dfII/etc/cdslib/sheets/US_8ths
DEFINE pragLib /home/gigzotest/github/gigzo_pdk/IGZO_PDK/V2.1.0.beta.2.hotfix.1/pdk/cdslib/pragLib
DEFINE lib_common /home/gigzotest/github/gigzo_pdk/IGZO_PDK/V2.1.0.beta.2.hotfix.1/pdk/cdslib/lib_common
DEFINE avTech /tools/CADENCE/ASSURA415/tools.lnx86/assura/etc/avtech/avTech
```

2.1.3 .bashrc 설정

Text editor를 통해 .bashrc 파일 열기 (예시) gedit ~/.bashrc 또는 emacs ~/.bashrc 등등

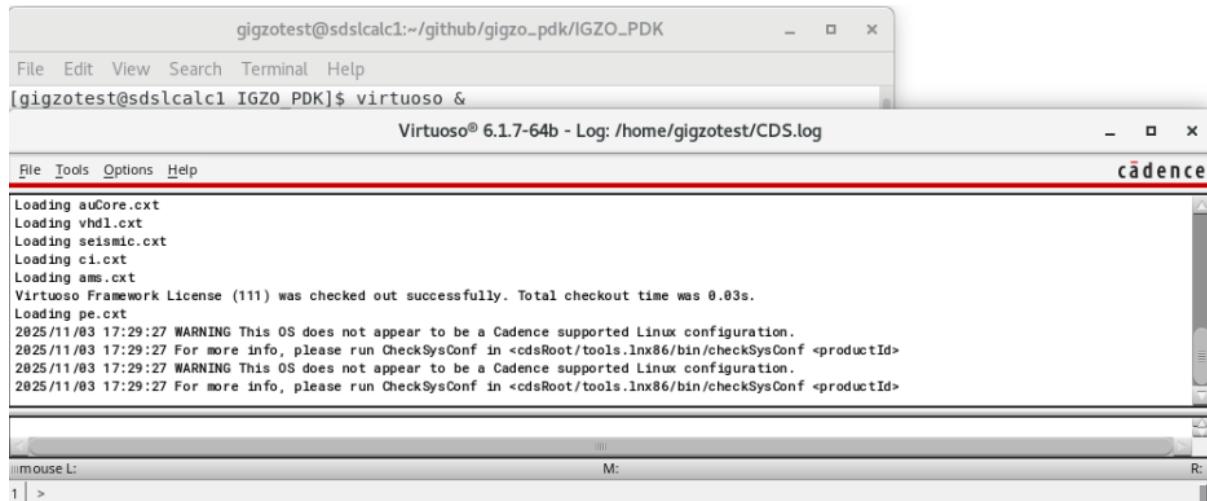
export PDK_DIR=/home/ "사용자경로"/V2.1.0.beta.2.hotfix.1/pdk" 로 설정.

```
#####
# Helvellyn 2.1.0 beta.2 PDK (GIGZO)
export PDK_DIR="/home/gigzotest/github/gigzo_pdk/GIGZO_PDK/V2.1.0.beta.2.hotfix.1/pdk"
```

2.2 Virtuoso 실행 및 Library manager 접속

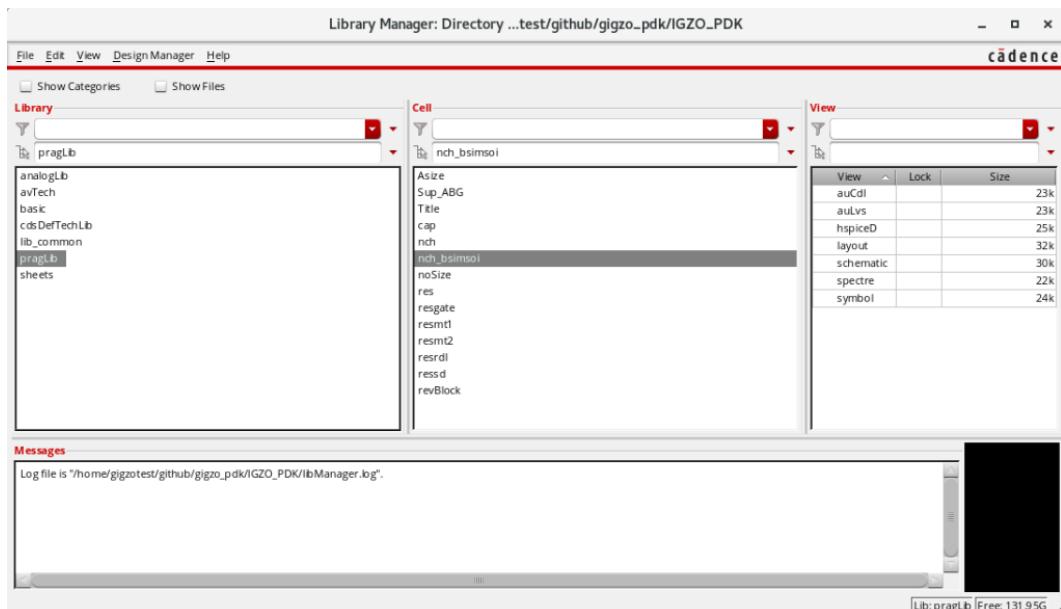
2.2.1 Virtuoso 실행

IGZO_PDK 폴더로 이동(cd IGZO_PDK) 후 리눅스 터미널창에서 **virtuoso &** 입력 후 엔터



< CIW 창 >

2.2.2 Library Manager 접속 (CIW 창에서 Tools-Library Manager 클릭)

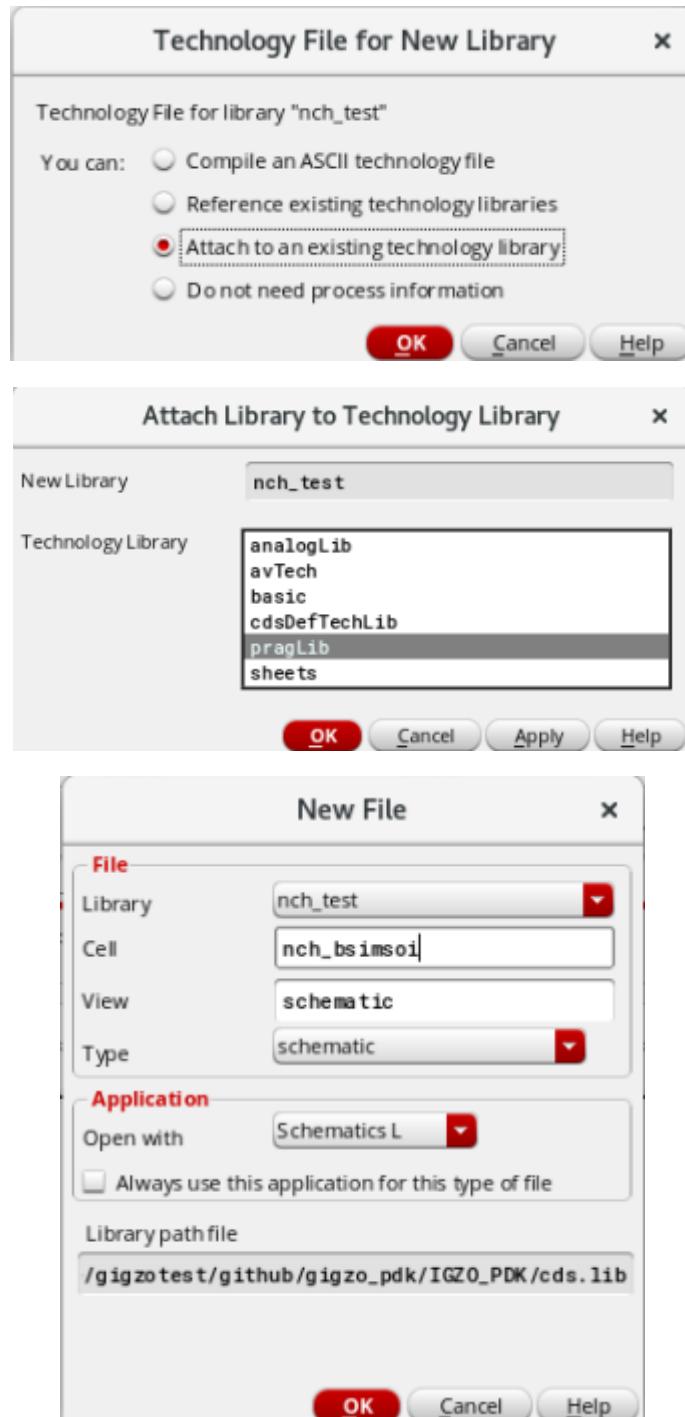


2.3 IGZO TFT 단일 소자에 관한 시뮬레이션 예제 (VDD=0.1V, V_{Gate}=-3V~3V sweep)

2.3.1 File-New-Library(Ctrl-N) 클릭 후 작업할 Library Name 설정

2.3.2 File-New-CellView 클릭 후 Technology File for New Library에서

Attach to an existing technology library 선택.

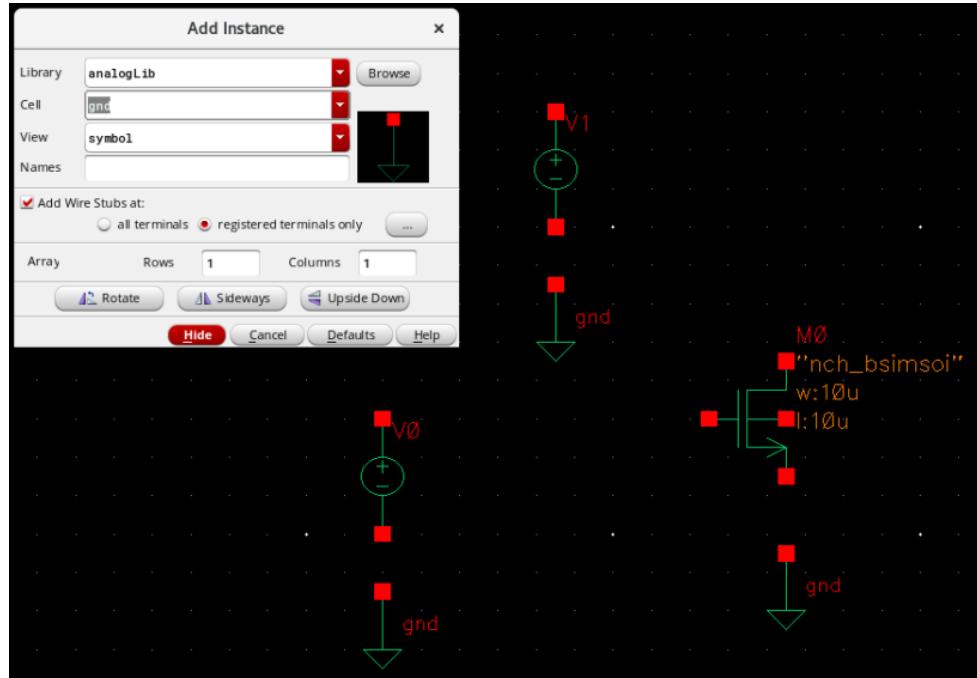


(예시) # Library : nch_test, # Cell : nch_bsimsoi

2.3.3 Virtuoso Schematic Editor L (VSE L) Schematic 및 Spectre Simulation.

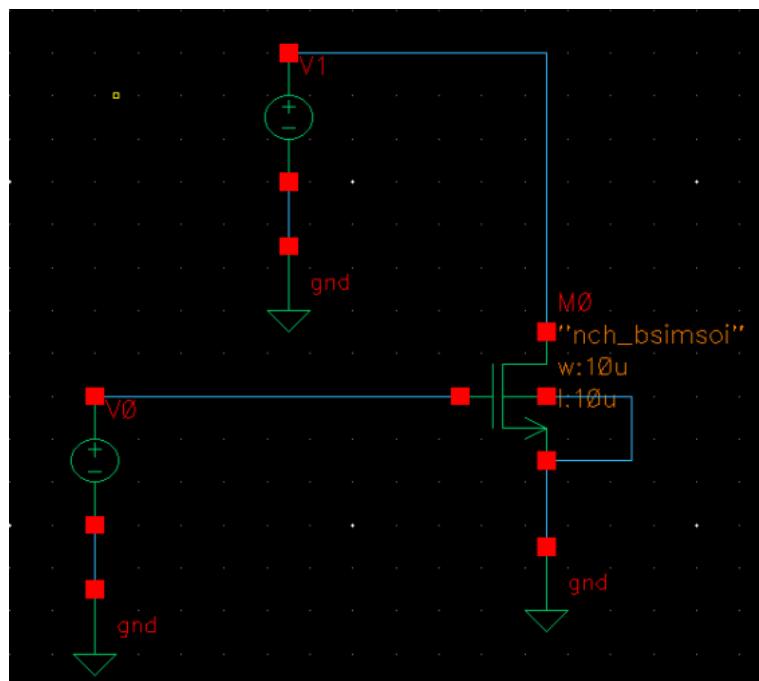
1) Add instance (키보드 i 입력)

키보드 i 입력 (Add instance) – # Library : pragLib, # Cell : nch_bsimsOI 선택 후 VSE 창에 마우스 커서 올린 후 클릭. 같은 방법으로 analogLib의 vdc와 gnd를 추가.



2) Create wire (키보드 w 입력)

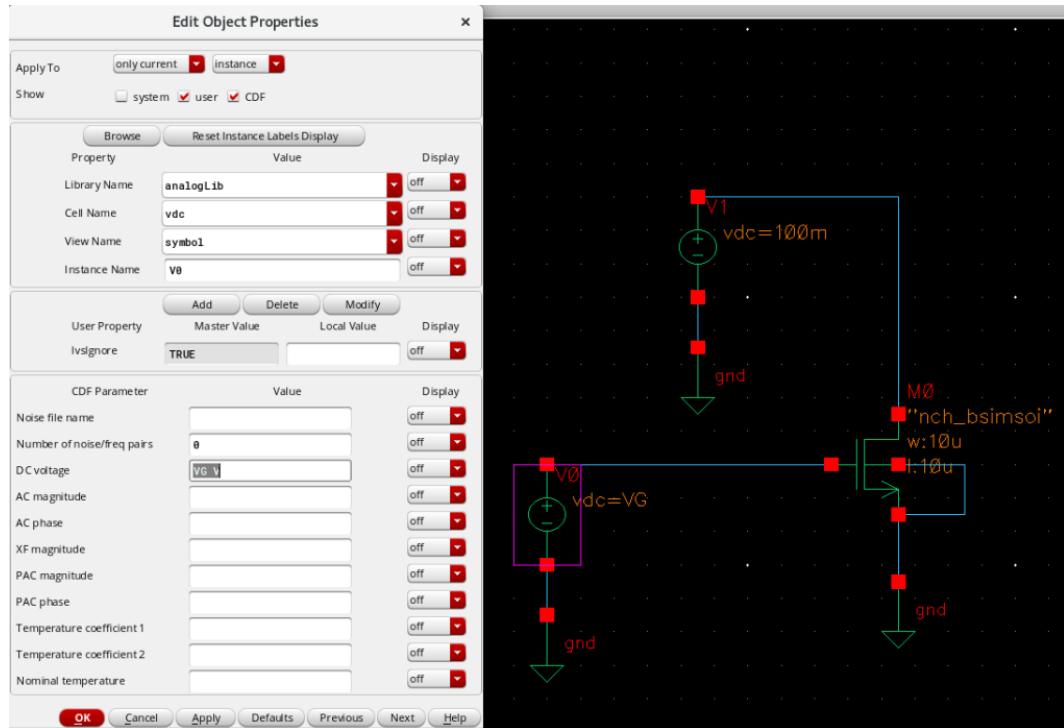
키보드 W 입력하여, instance을 연결한다.



3) vdc bias 설정.

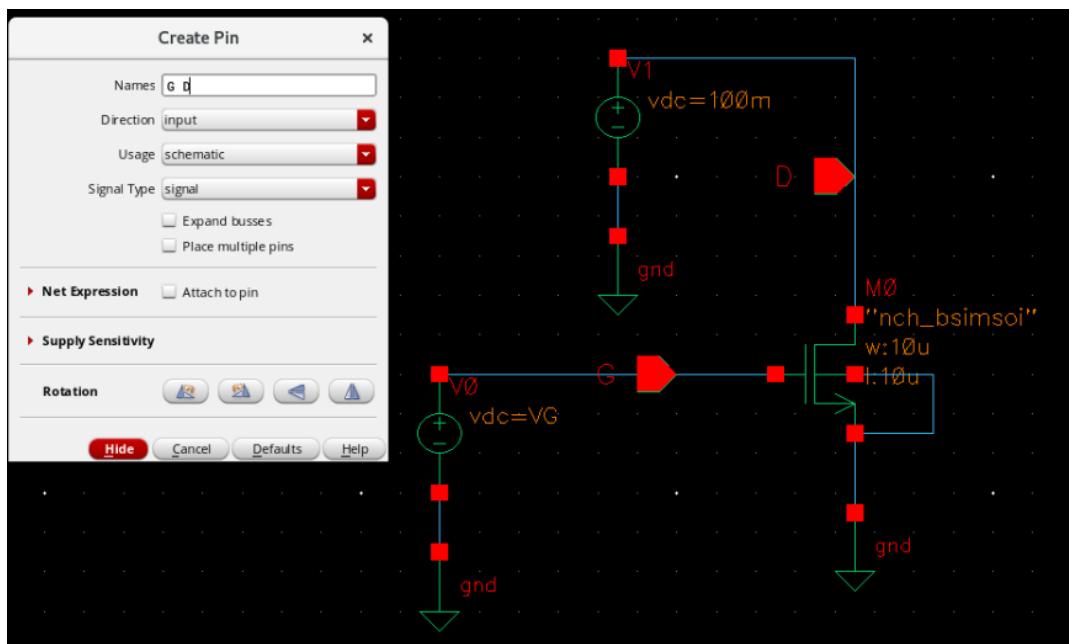
Instance 클릭 후 키보드 q 입력 하여, Object Properties를 아래 예시와 같이 수정할 수 있다.

V1 (VDD)의 DC voltage = 100m V, V2 (V_{Gate})의 DC voltage = VG V로 설정.



4) Pin 설정

키보드 p 입력 – # Name : G D로 설정한 뒤 아래 예시와 같이 input G, D를 각각 클릭하여 설정 해줄 수 있다.



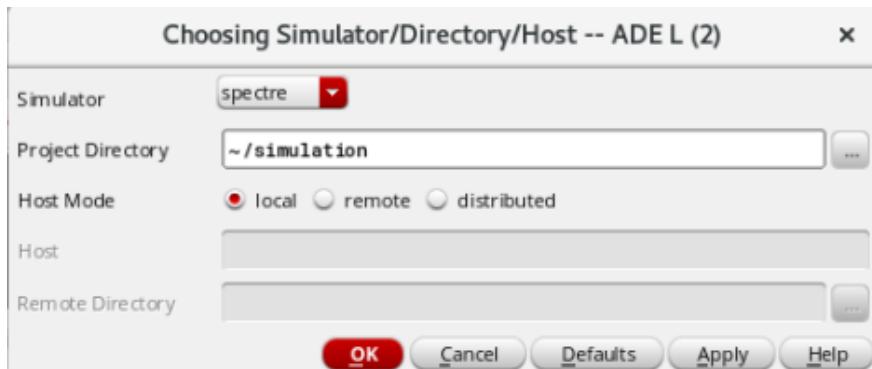
5) 회로 Simulation 환경 설정

Launch – ADE L

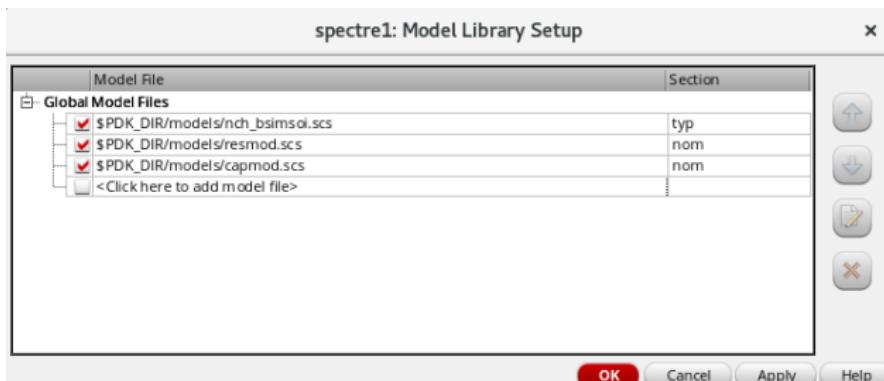
5.1) Simulator 설정 : Setup – Simulator/Directory/Host (default 설정이 되어있으니 생략 가능)

Simulator : Spectre 선택 (default)

Project Directory : 결과 저장 경로 지정 (~/simulation/0) default로 설정되어 있음.

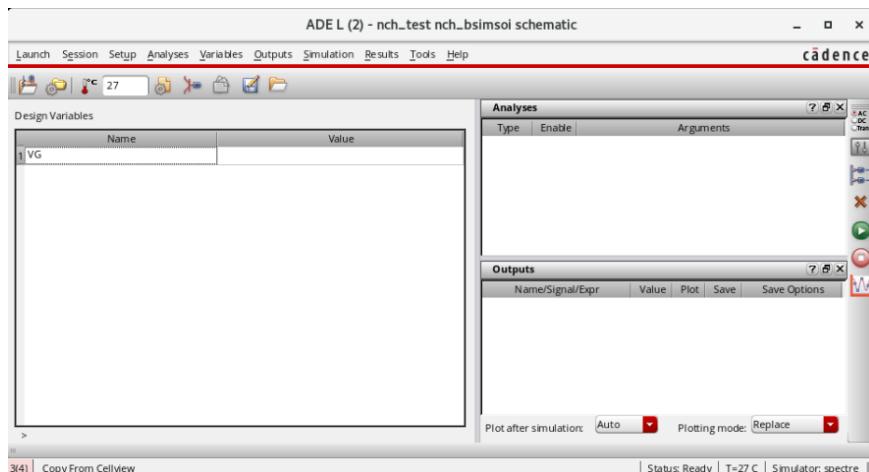


5.2) Model 파일 불러오기 : Setup – Model libraries (default 설정이 되어있으니 생략 가능)



5.3) Design variable 설정

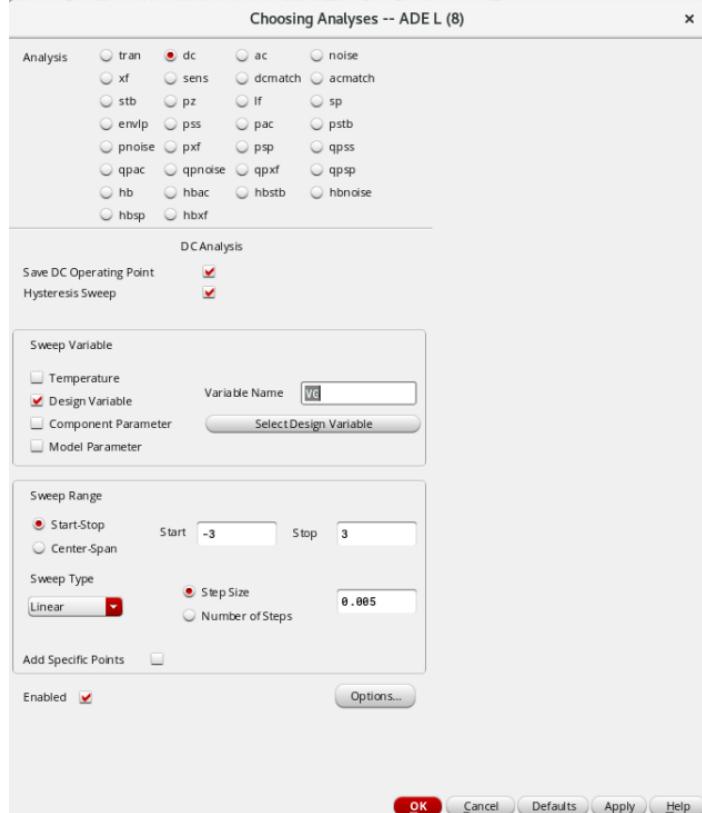
Variables – Copy from Cell View 클릭 - #VG Value : 0



5.4) Analyses 설정 (Analyses – Choose)

Analysis dc 선택 – Save DC Operating Point 체크, Hysteresis Sweep 체크 (선택) – Design Variable #Variable Name : VG - #Start : -3, Stop : 3 – Sweep Type : Linear, #Step Size : 0.005

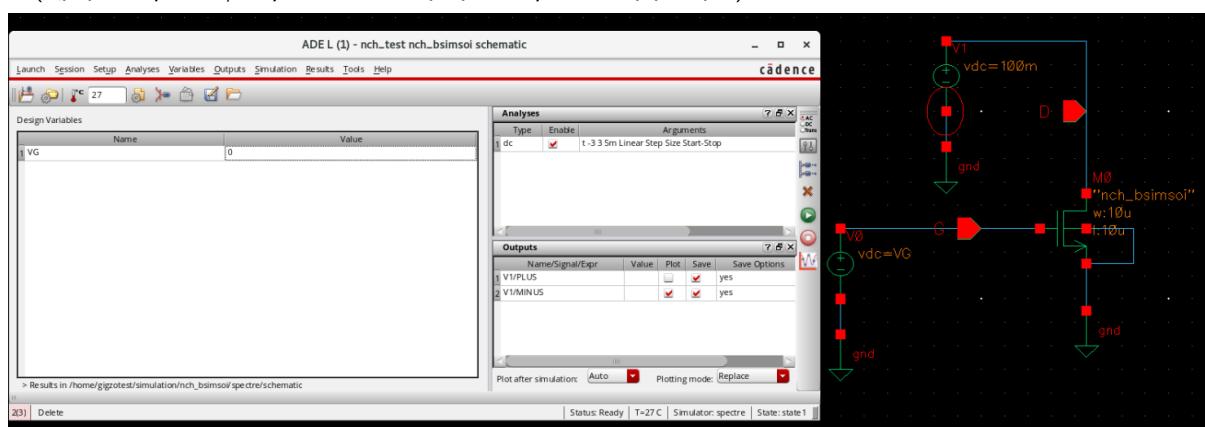
(Sweep Type : Automatic 으로 할 경우, 수렴 오차(Convergence error), reltol(relative tolerance), abstol(absolute tolerance) 등의 문제로 정상적인 IV Curve가 나오지 않음.)



5.5) Outputs 설정

Outputs – To Be Saved – Select On Design – Plot/Save 할 node 클릭

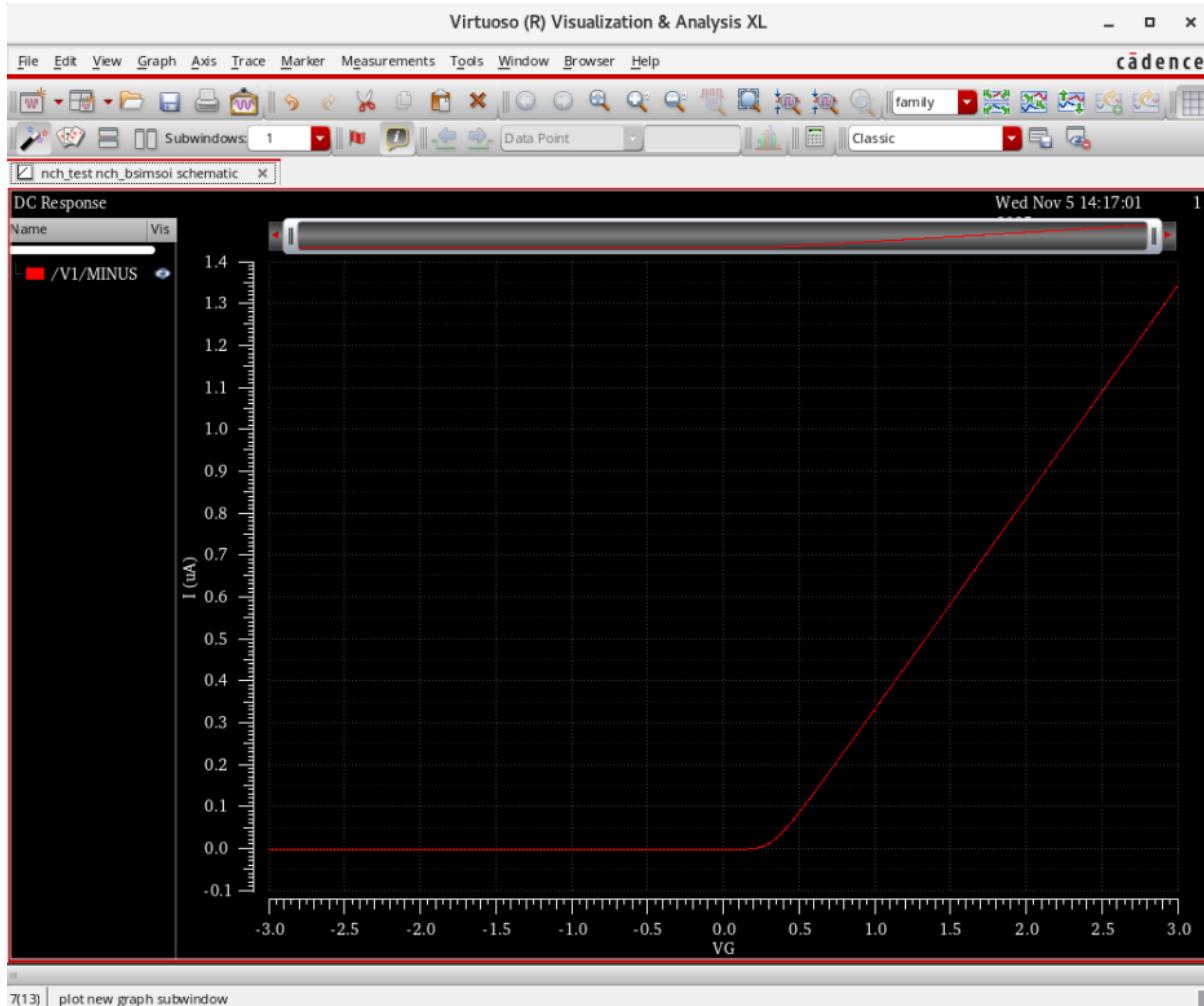
(예시 : V1/PLUS, V1/MINUS 클릭 후 Plot/Save 여부 체크)



5.6) Simulation 실행

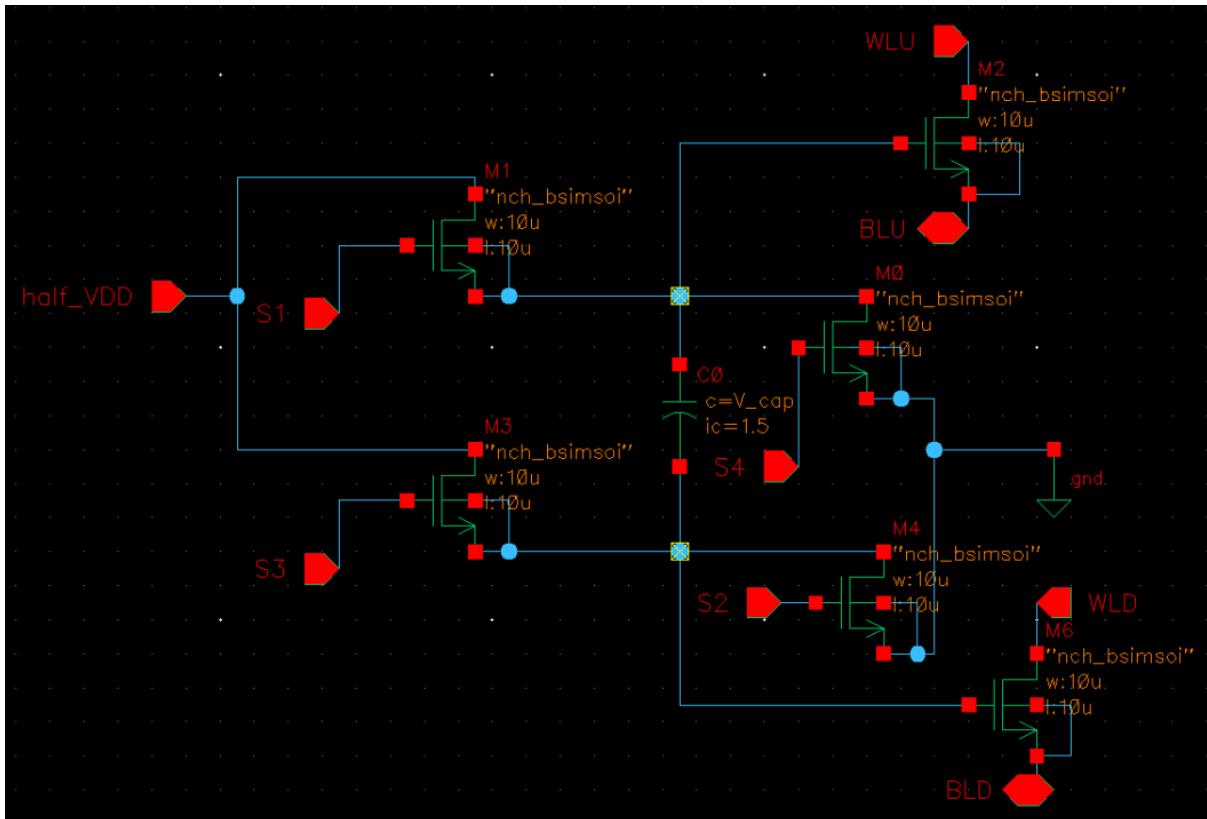
Simulation – Netlist and Run (초록색 재생 모양 버튼)

(nch_bsimsol.scs 가 수정되거나 Schematic 수정 뒤에는 반드시 저장 후 Simulation – Netlist – Recreate 후에 실행하여야 함.)



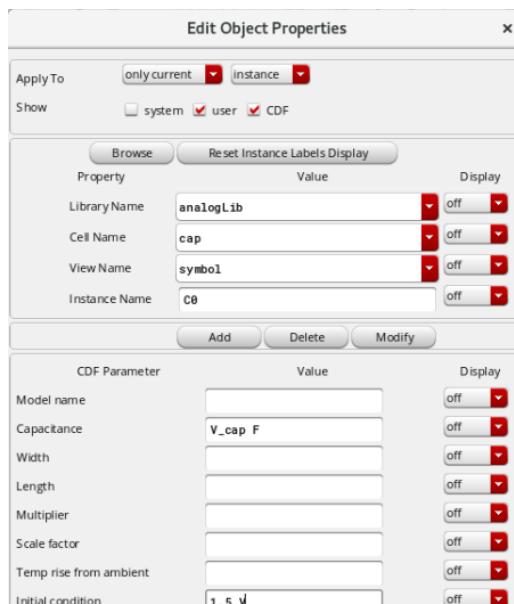
2.4 IGZO TFT 소자를 활용한 6T1C 시뮬레이션 (Retention time)

1) Add instance and connection (#c=V_cap, ic=1.5)



Check and Save 시 또는 경고 문구는 wire 생성시 자동으로 net이 겹쳤다(short)는 경고문이고, 의도한 연결이므로 무시하면 됨.

Cap(C0)의 설정은 아래 그림과 같음.



2) 회로 시뮬레이션 환경 설정

2.1) Variables – Copy from Cellview - #V_cap = 0

2.2) Pulse bias 설정하기

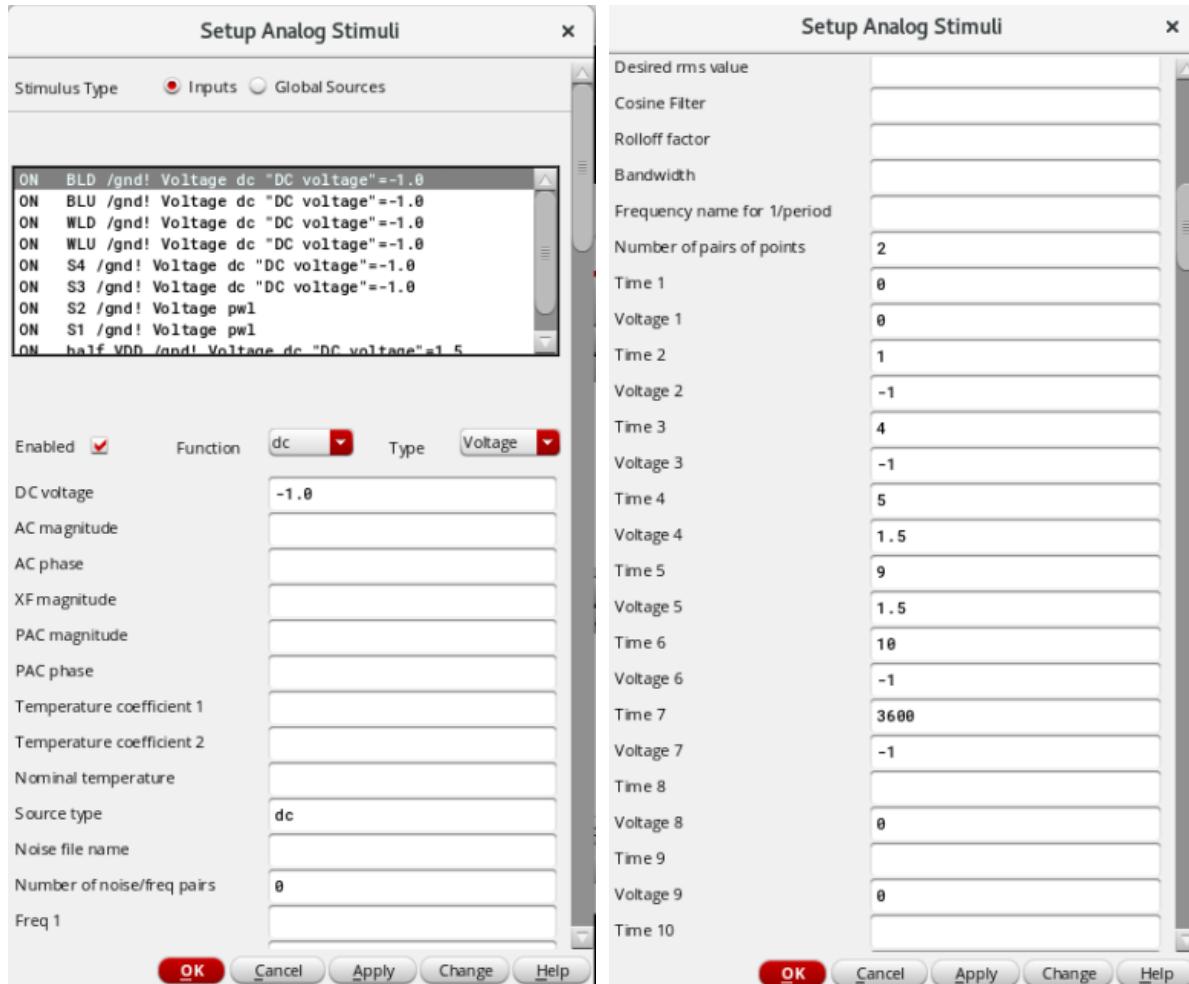
2.2.1) Setup – Stimuli

2.2.2) DC voltage 설정

WLU, WLD, BLU, BLD, S3, S4 : Enabled 체크 – Function dc 선택 - # DC voltage : -1.0 V
입력 후 Apply.

half_VDD : Enabled 체크 – Function dc 선택 - # DC voltage : 1.5 V 입력 후 Apply.

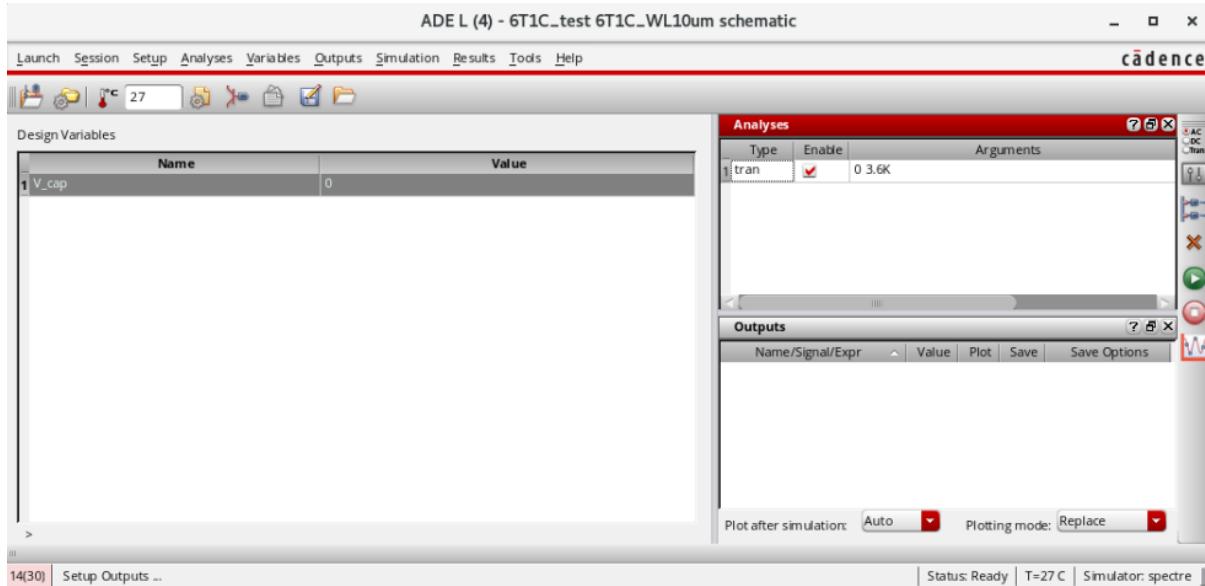
2.2.3) S1, S2 : Enabled 체크 – Function pwl 선택 – time/voltage 설정. (DC voltage 설정X)



(좌) DC voltage 설정 (WLU, WLD, BLU, BLD, S3, S4) (우) pwl voltage 설정 (S1,S2)

2.3) Analyses 설정(Analyses – Choose)

Analysis tran 선택 - # Stop Time : 3600, Enabled 체크 – Apply – OK



2.4) Netlist 생성 (Netlist – Create)

```
// Generated for: spectre
// Generated on: Nov 5 16:00:55 2025
// Design library name: 6T1C_test
// Design cell name: 6T1C_WL10um
// Design view name: schematic
simulator lang=spectre
global 0
parameters V_cap=0
include "$PDK_DIR/models/nch_bsimsoi.scs" section=10um
include "$PDK_DIR/models/resmod.scs" section=nom
include "$PDK_DIR/models/capmod.scs" section=nom

// Library name: 6T1C_test
// Cell name: 6T1C_WL10um
// View name: schematic
M6 (WLD net12 BLD BLD) nch_bsimsoi w=10u l=10u
M4 (net12 S2 0 0) nch_bsimsoi w=10u l=10u
M3 (half_VDD S3 net12 net12) nch_bsimsoi w=10u l=10u
M2 (WLU net13 BLU BLU) nch_bsimsoi w=10u l=10u
M1 (half_VDD S1 net13 net13) nch_bsimsoi w=10u l=10u
M0 (net13 S4 0 0) nch_bsimsoi w=10u l=10u
C0 (net13 net12) capacitor c=V_cap ic=1.5
simulatorOptions options psfversion="1.1.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalen=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
tran tran stop=3600 write="spectre.ic" writefinal="spectre.fc" \
    annotate=status maxiters=5
finalTimeOP info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```

2.4) Cap 설정하기

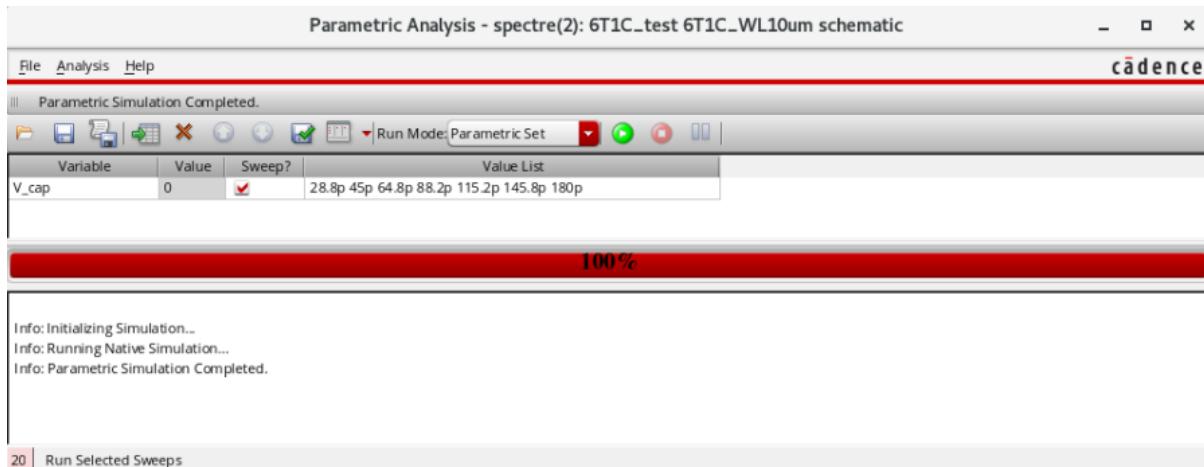
28.8pF, 45pF, 64.8pF, 88.2pF, 115.2pF, 145.8pF, 180pF 으로 총 7개의 Cap의 그래프를 한번에 보기위한 설정.

2.4.1) Tools – Parametric Analysis

2.4.2) Run Mode : Parametric Set 선택, Variable : V_cap 선택

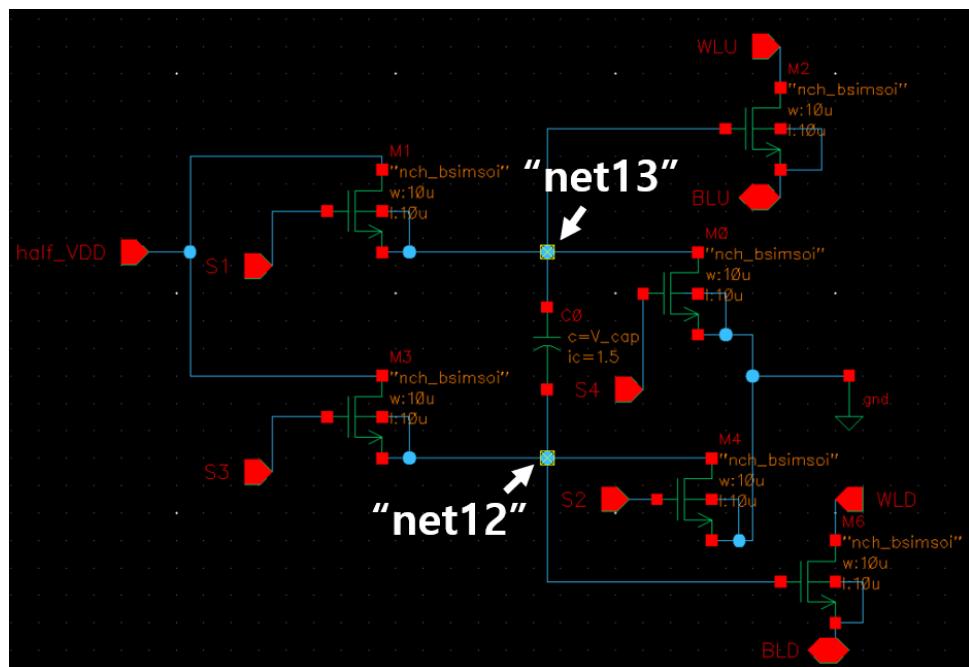
2.4.3) # Value List : 28.8p 45p 64.8p 88.2p 115.2p 145.8p 180p 입력

2.4.4) Run Selected Sweep 클릭 (초록색 재생 모양 버튼)



2.5) Output 설정 (Cap 양단의 voltage 차이를 계산하여 plot 하기위한 설정)

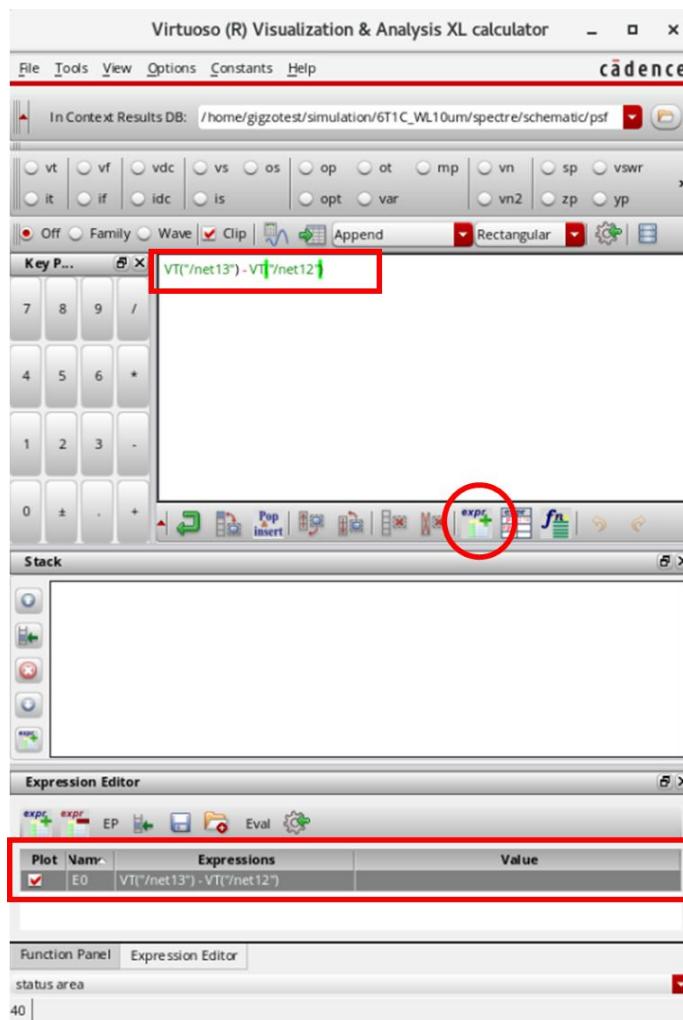
2.5.1) Cap 양단의 net 번호 확인 (마우스 커서 올린 후 Schematic 창 왼쪽 하단에서 확인)



2.5.2) Outputs – Setup – Calculator : Open 클릭

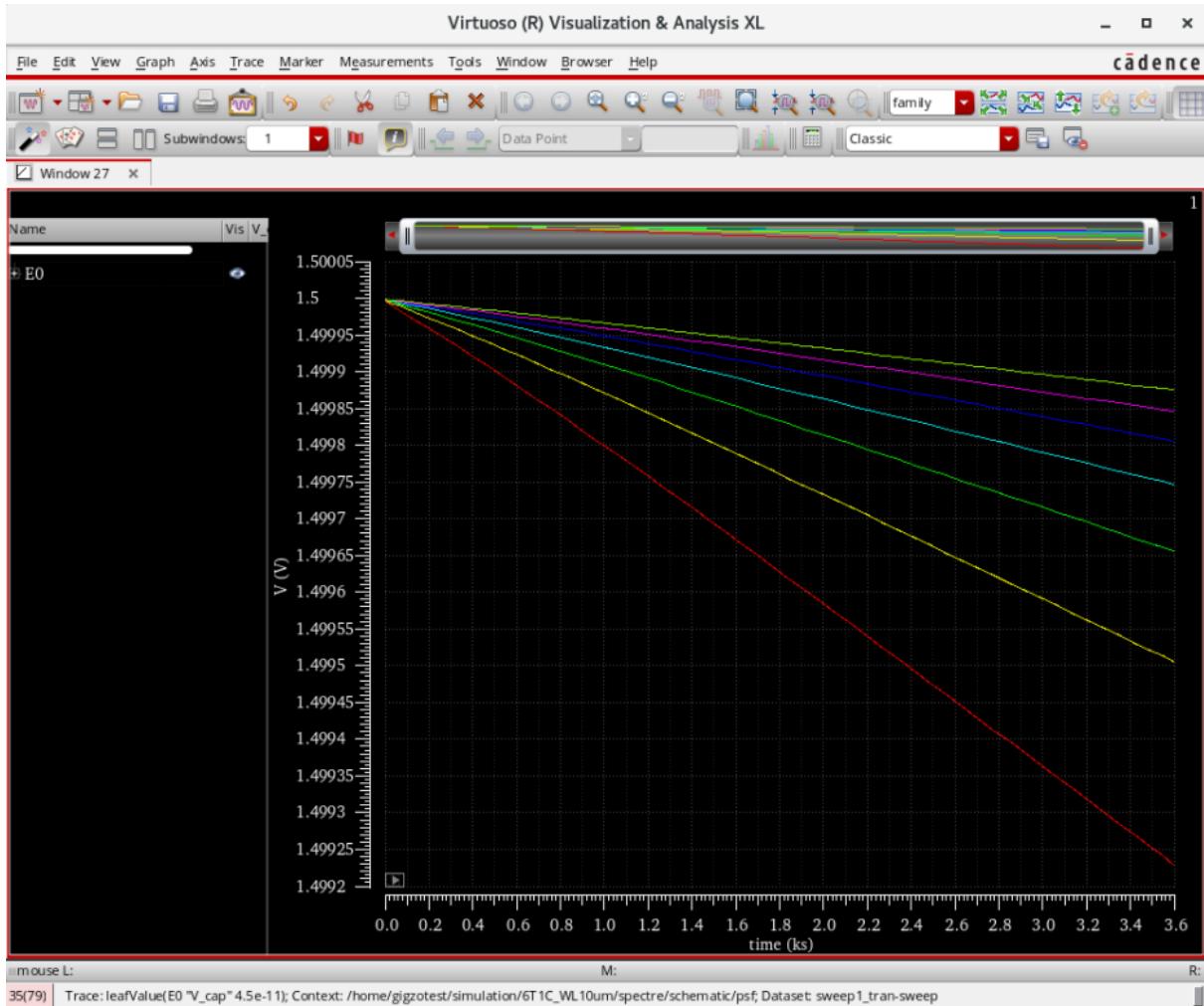


2.5.3) Virtuoso (R) Visualization & Analysis XL calculator 창에서 VT("/net13")-VT("/net12") 입력한 뒤 Expr+ 모양 클릭하여 Expression Editor에 calculation 식을 적용.



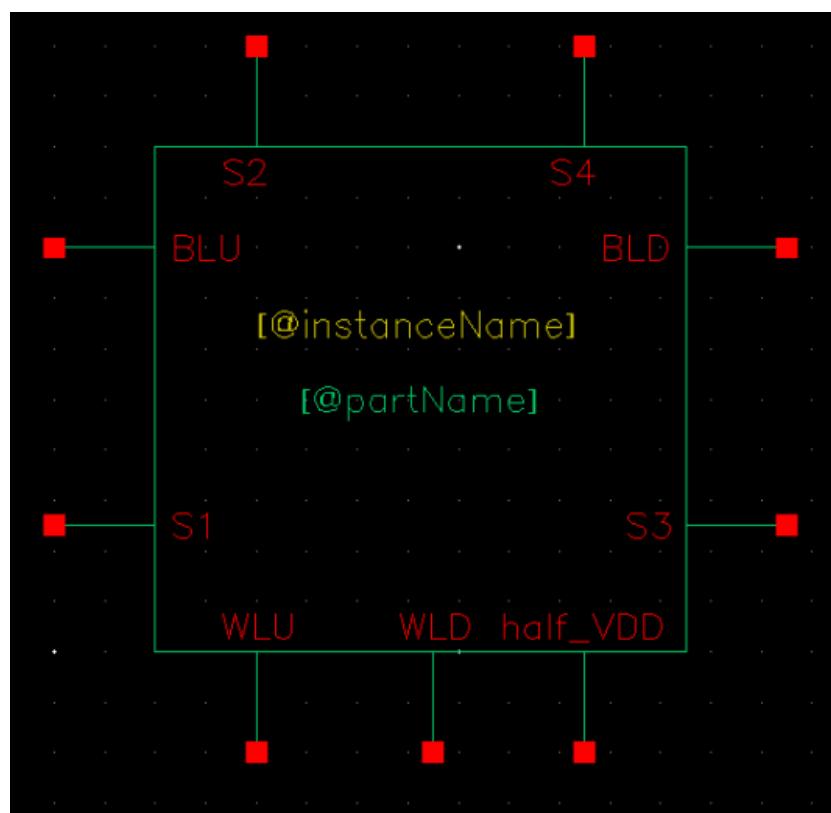
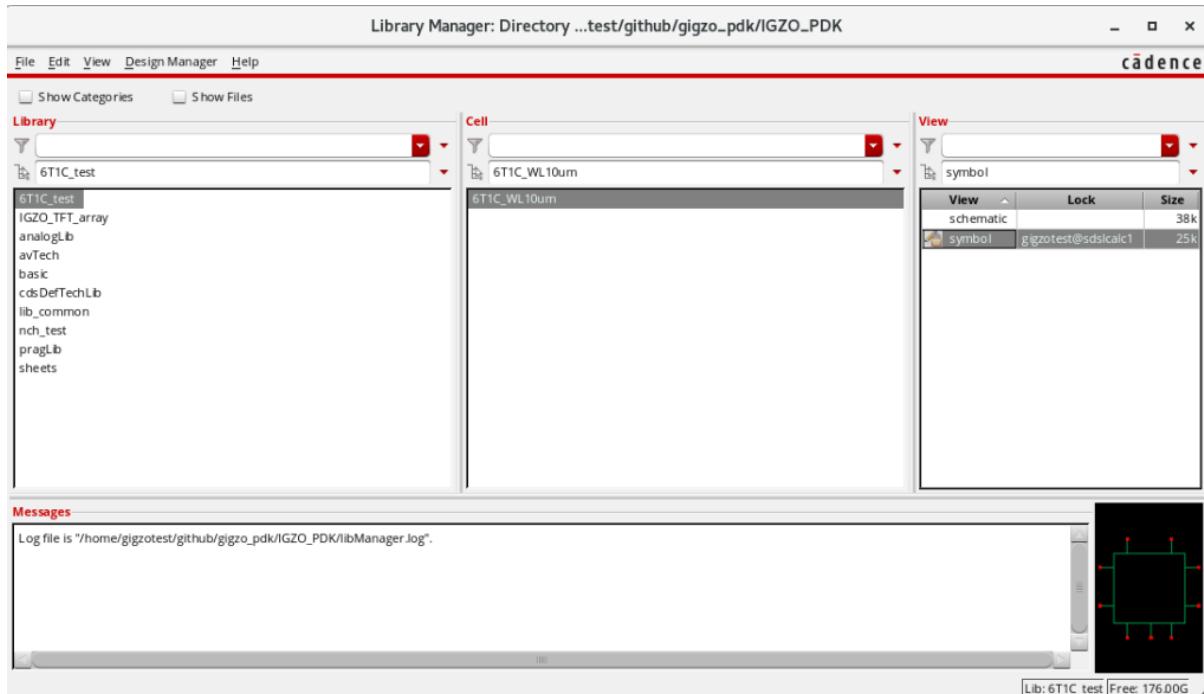
3) 시뮬레이션 실행

Virtuoso (R) Visualization & Analysis XL calculator 창에서 Expression Editor 메뉴 중 Eval 클릭하여 simulation 실행.



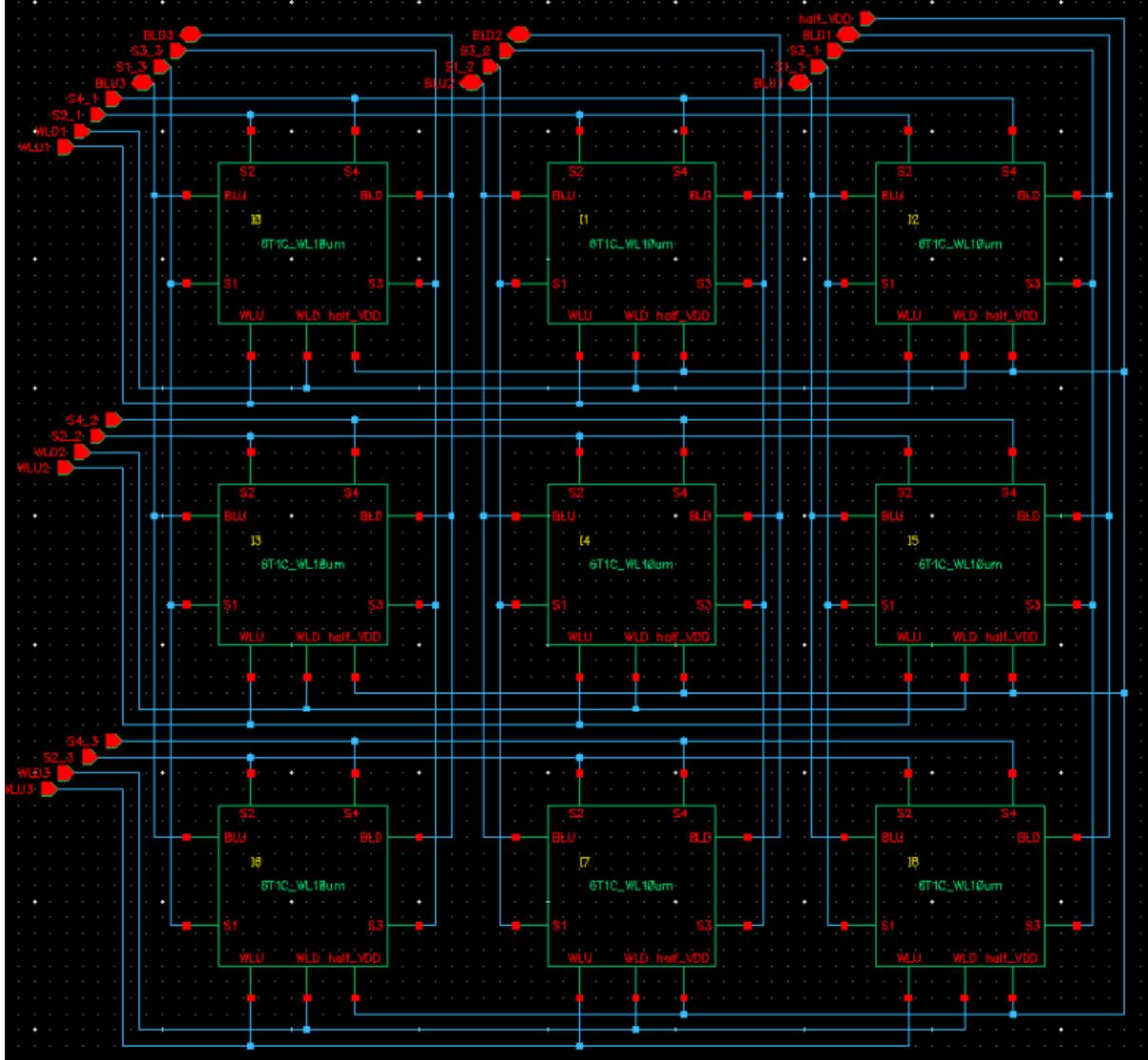
2.5 3x3 어레이 시뮬레이션

- 1) 앞선 IGZO TFT 6T1C를 symbol화 한다.



이후 Retention time 시뮬레이션과 달리 초기 V_{cap0} 0에서 시작하도록 Schematic의 $c=V_{cap}$, $ic=0$ 로 설정.

2) Add instance and connection



3) 회로 시뮬레이션 환경 설정

3.1) Variables – Copy from Cellview - #V_cap = 180p

3.2) Pulse bias 설정하기

3.2.1) Setup – Stimuli

3.2.2) 비활성화 cell의 bias 설정 (Potentiation 시)

WLU, WLD, BLU, BLD, S1, S2, S3, S4 : Enabled 체크 – Function dc 선택 - # DC voltage : 0
입력 후 Apply.

3.2.3) 활성화 cell의 bias 설정

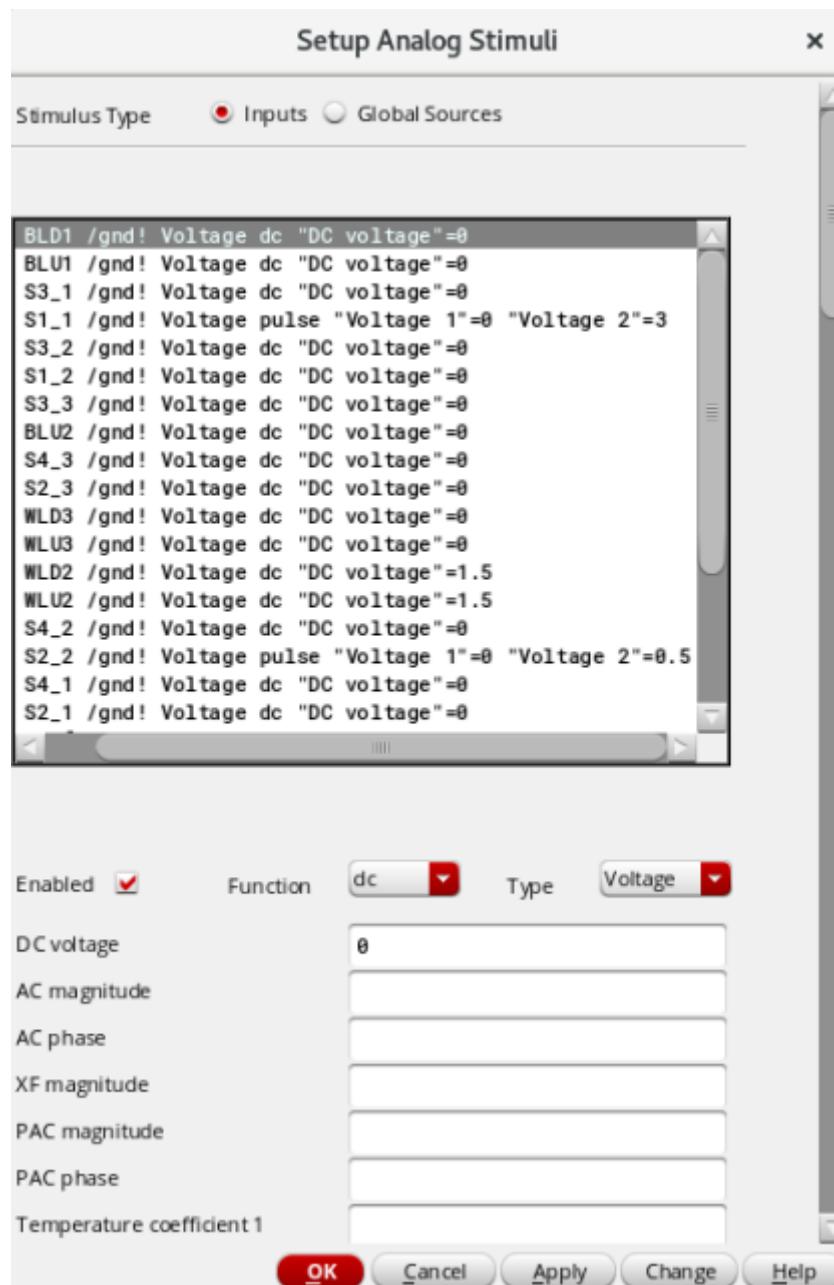
WLU,WLD : Enabled 체크 – Function dc 선택 - # DC voltage : 1.5 입력 후 Apply

BLU,BLD = Enabled 체크 – Function dc 선택 - # DC voltage : 0 입력 후 Apply

S1 : Enabled 체크 – Function pulse 선택 – # Period : 200n, Delay time : 100n, Rise time = Fall time = 5n, Pulse width = 100n, Voltage1 = 0, Voltage2 = 3 입력 후 Apply.

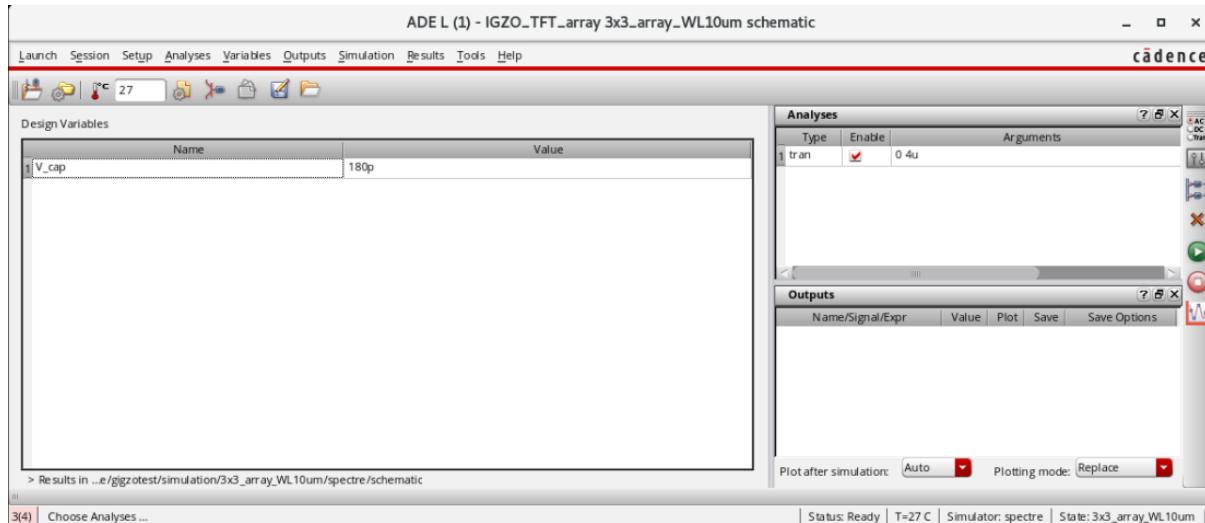
S3 : Enabled 체크 – Function pulse 선택 – # Period : 200n, Delay time : 100n, Rise time = Fall time = 5n, Pulse width = 100n, Voltage1 = 0, Voltage2 = 0.5 입력 후 Apply.

S2 = S4 = Enabled 체크 – Function dc 선택 - # DC voltage : 0 입력 후 Apply



3.3) Analyses 설정(Analyses – Choose)

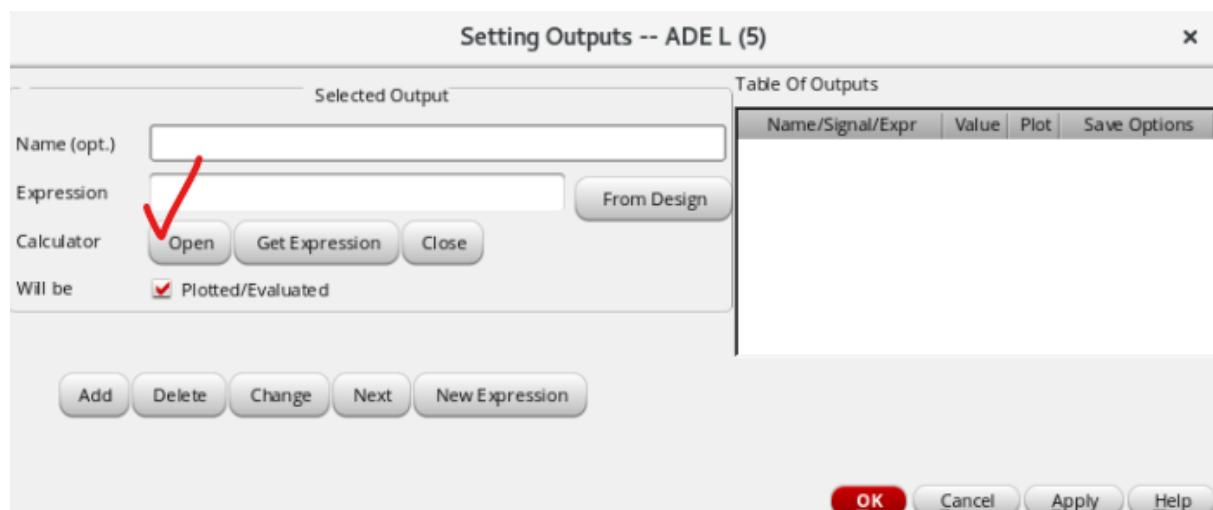
Analysis tran 선택 - # Stop Time : 4u, Enabled 체크 – Apply – OK



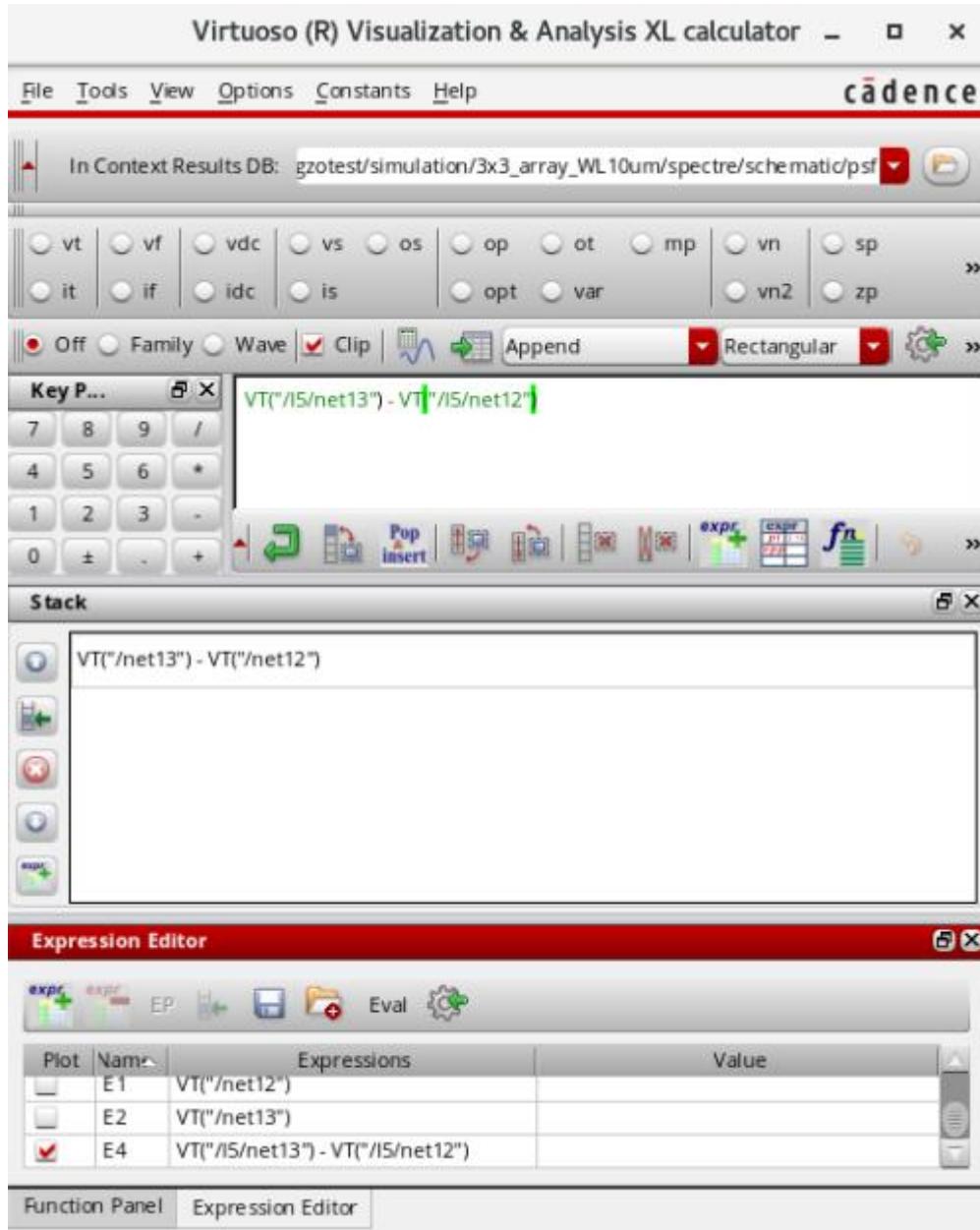
3.4) Netlist 생성 (Netlist – Create)

3.5) Output 설정

3.5.1) Outputs – Setup – Calculator : Open 클릭



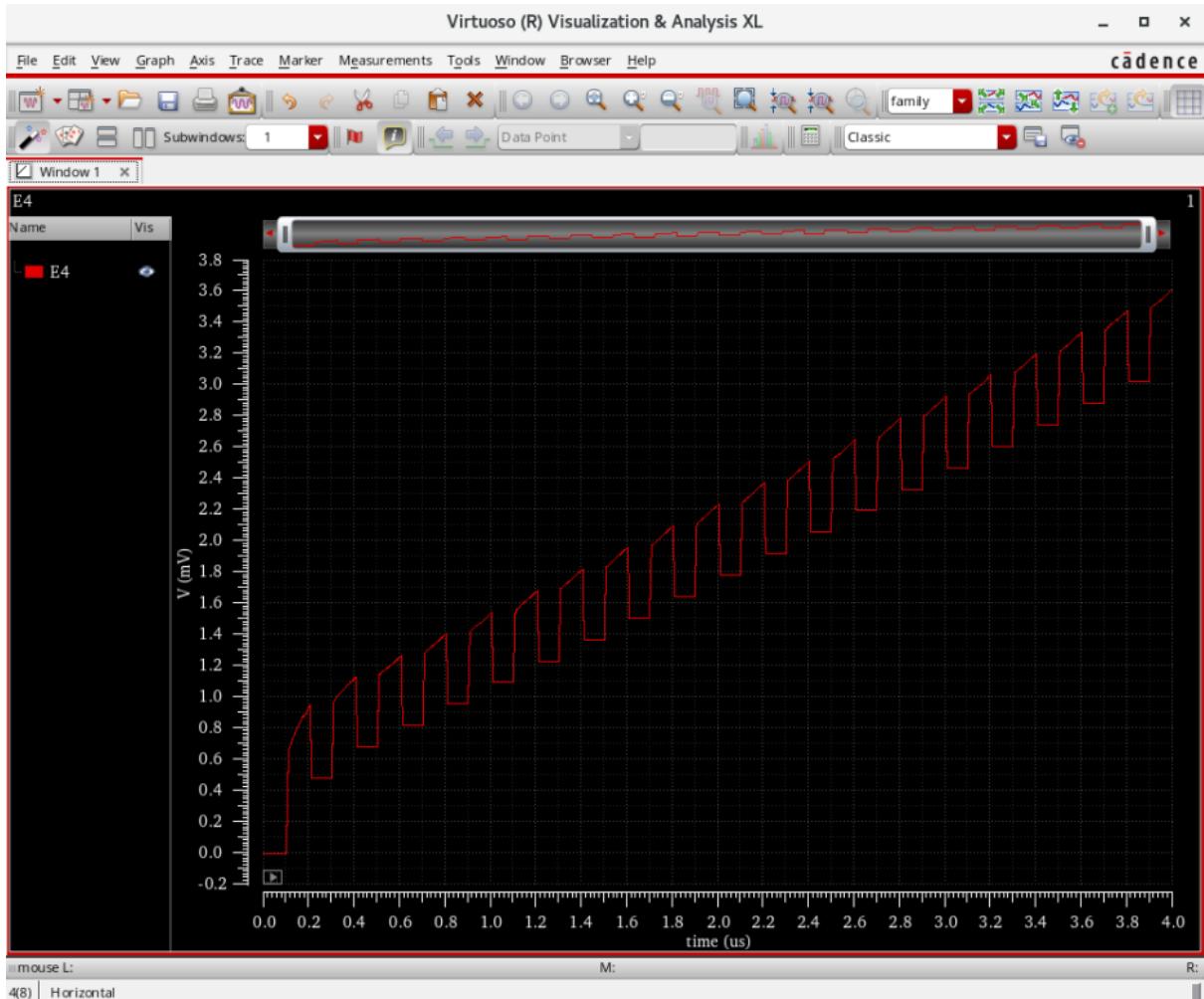
3.5.2) Virtuoso (R) Visualization & Analysis XL calculator 창에서 VT("/활성화인스턴스/net13")-VT("/활성화인스턴스/net12") 입력한 뒤 Expr+ 모양 클릭하여 Expression Editor에 calculation 식을 적용.



< 활성화인스턴스 : I5인 경우 설정 예시 >

3) 시뮬레이션 실행

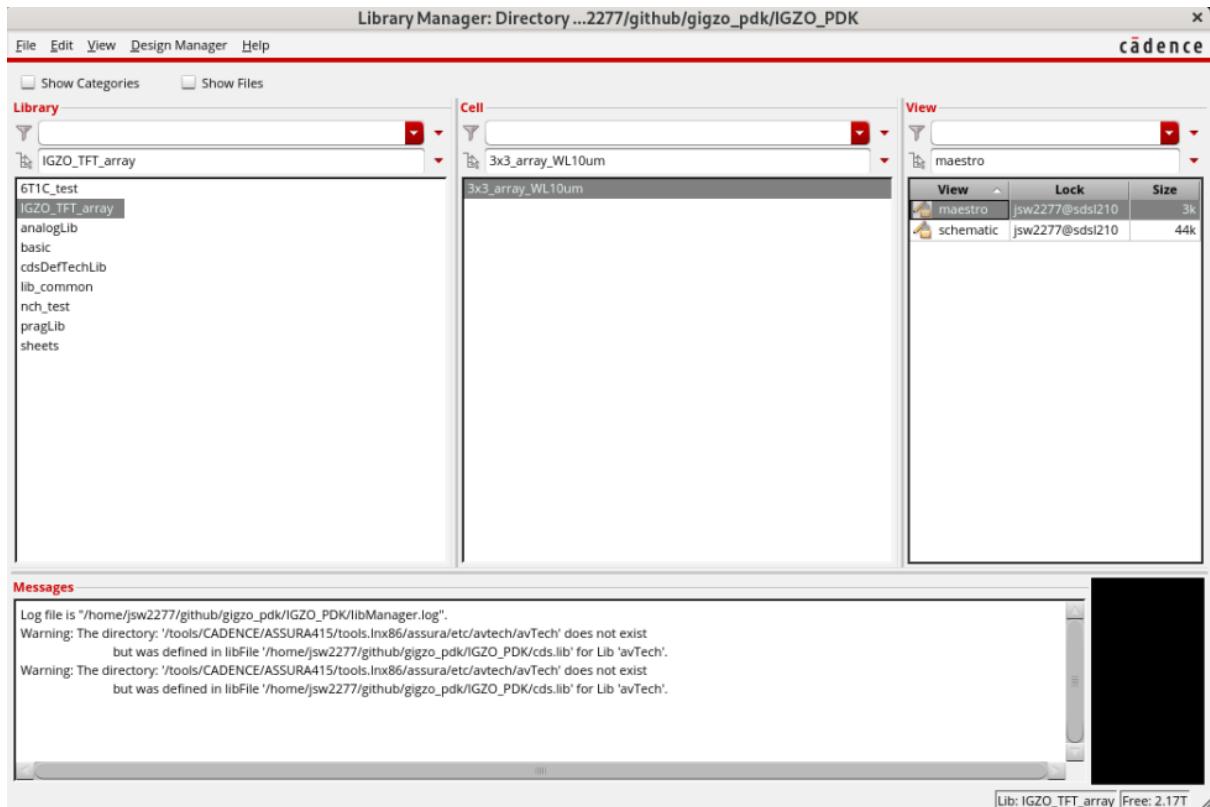
Virtuoso (R) Visualization & Analysis XL calculator 창에서 Expression Editor 메뉴 중 Eval 클릭하여 simulation 실행.



3. 참고자료

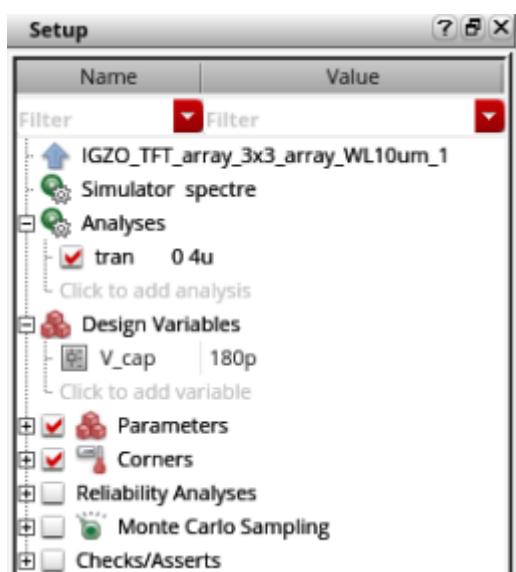
3.1. IC23.1 버전 사용법

3.1.1. maestro 라는 test bench를 통해 simulation을 진행함.



3.1.2. maestro View 클릭 후 상단 menu 탭에서 transient time 및 Variables 설정 가능.

설정값은 좌측 Setup 창에서 확인이 가능함.



3.1.3. Stimuli 설정

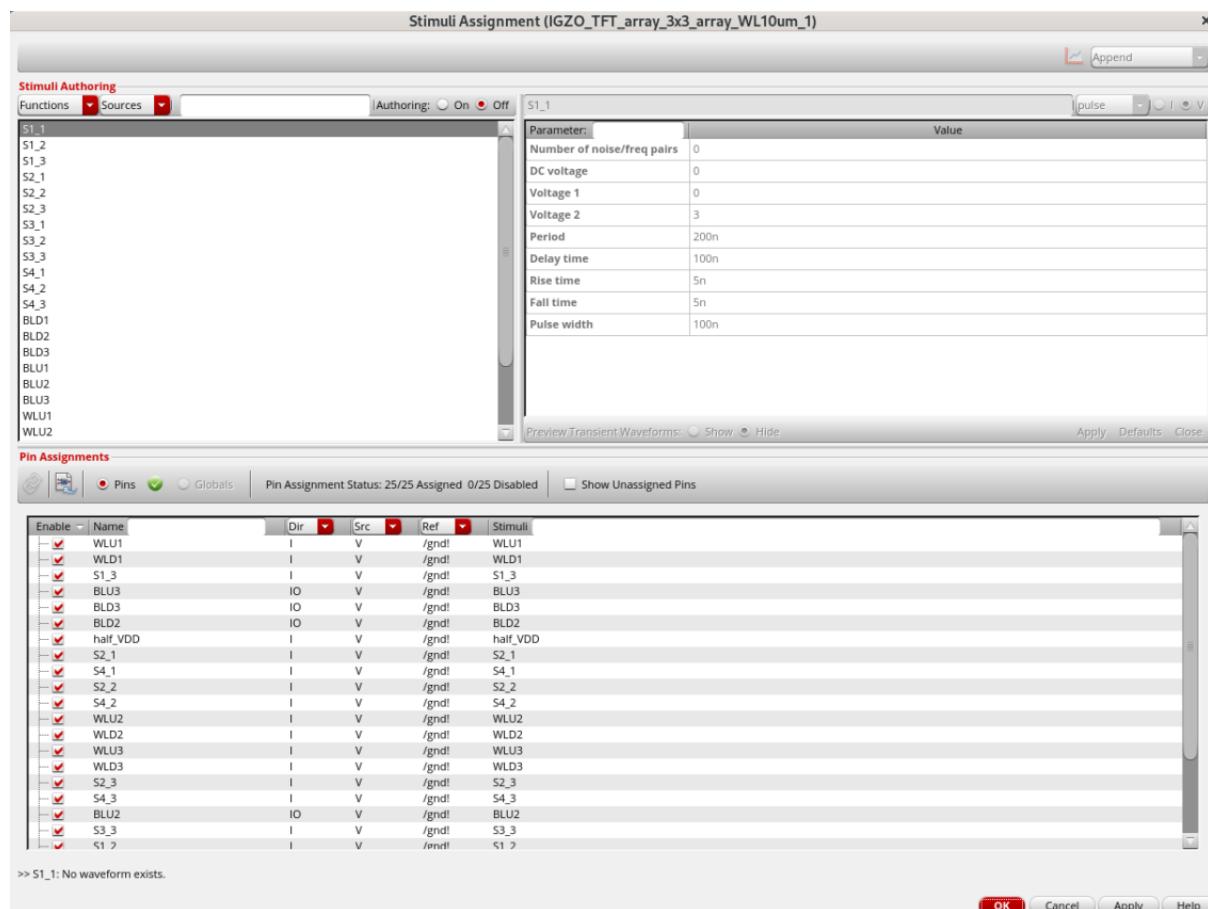
3.1.3.1. Stimuli Atuthoring의 Authoring : On으로 바꾼 뒤 우측에 보이는 dc 다운드랍탭 바로 왼쪽의 긴 탭에 인가할 Bias name을 지정함.

3.1.3.2. 해당 bias의 종류 (dc,pwl,pulse etc)를 선택하고 Parameter 값을 설정한 뒤 Apply.

3.1.3.3. 그렇게 Stimuli Authoring에 bias를 모두 설정해주고, Authoring : Off 클릭.

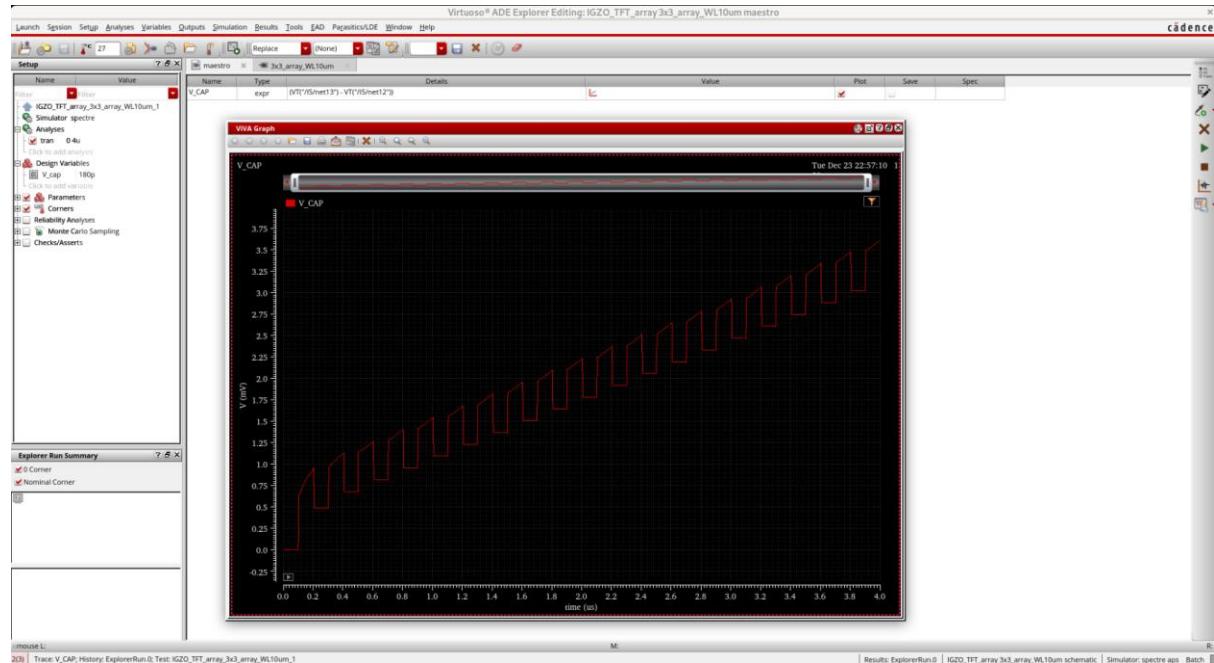
3.1.3.4. Stimuli Authoring 중 하나 (ex. S1_1)를 클릭하여 회색탭이 되도록 만들고, 인가할 하단의 Pin (ex. S1_1)을 우클릭하여 Assign Stimuli to Selected Pins 클릭. (Pin 할당)

3.1.3.5. 모두 할당하면 아래 그림과 같은 상태가 되고, 이를 Apply-OK를 통해 Stimuli를 설정 완료함.



3.1.4. Virtuoso (R) Visualization & Analysis XL calculator에서 설정했던 VT("/활성화인스턴스/net13")-VT("/활성화인스턴스/net12")는 상단 Outputs-Add-expression에서 설정 가능.

3.1.5. 이후 오른쪽 Run Simulation (초록색 재생버튼)을 클릭하여 Viva graph를 통해 시뮬레이션 결과 확인.



3.2. IC23.1 버전에서 pwl 설정 방법

Pwl의 Number of pairs의 default 값은 2로 설정되어 있으나, 이를 늘리면 Time domain을 늘릴 수 있음.