

Semiconductor IC Design with MyChipStationX

A Practical Guide to Modern ASIC Design: From Theory to Tape-out

Selconn Co., Ltd.

June 11, 2026

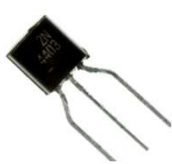
Agenda

1. IC, Integrated Circuit
2. Basic Cell and it's operation
3. Understanding IC design and production
4. About MyChipStationX
5. Basic Cell (Inverter) design example
6. Stopwatch design using MyChipStationX(Bottom-up)
7. About ASIC/SoC chipset design
8. Stopwatch design using MyChipStationX(Top-down)

1. IC, Integrated Circuit

Semiconductor Device

Transistor, MOSFET, Resistor, Capacitor...

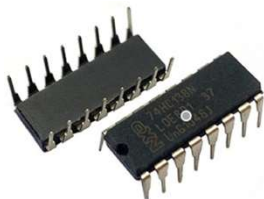


IC : IC is integrated circuit, many devices(transistors, resistors, capacitors...) is integrated on one chip (IC, ASIC, SoC...)

IC : Integrated Circuit

ASIC : Application Specific IC

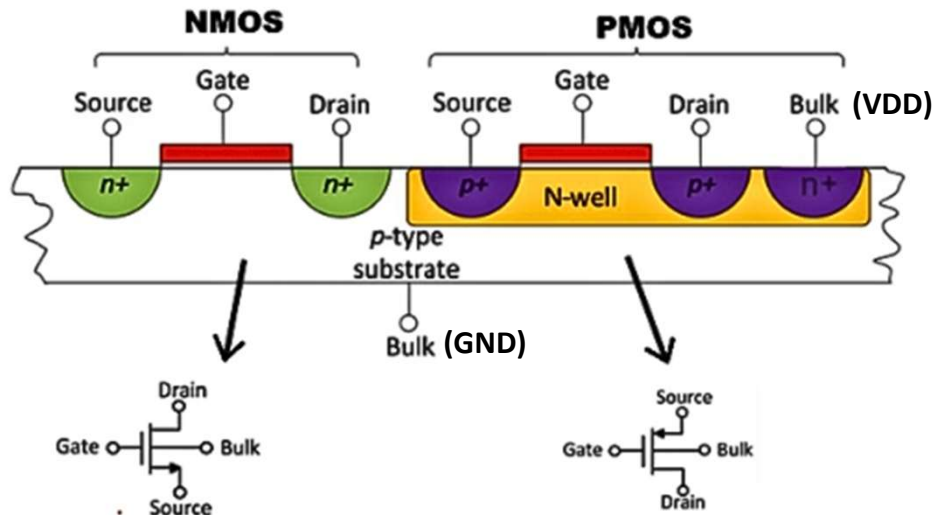
SoC : System on Chip



2. Inverter Cell and operation of CMOS

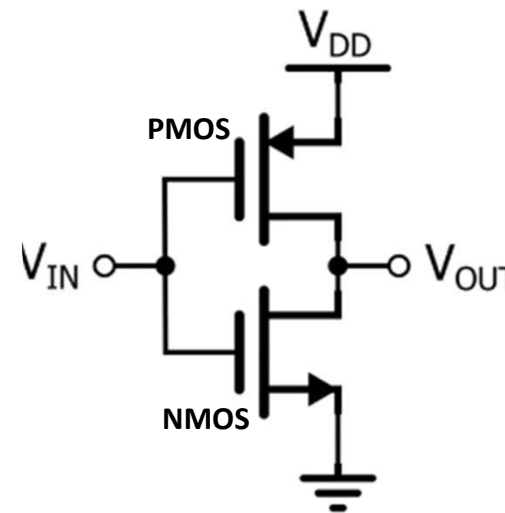
CMOS Device PMOS/NMOS (Complementary Metal-Oxide-Semiconductor)

Wafer cross section



Gate width (channel length) is important.
Matured technology : 65/90/130/180/350/500nm

Inverter



V _{IN}	V _{OUT}
0	1
1	0

PMOS Operating Principle

When the gate voltage is low (ON)

- The gate is at a lower voltage than the n-well.
- A hole channel forms at the surface of the n-well.
- Current flows (Source → Drain).

When the gate voltage is high (OFF)

- No channel is formed.
- Current flow is blocked.

NMOS Operating Principle

When the gate voltage is low (OFF)

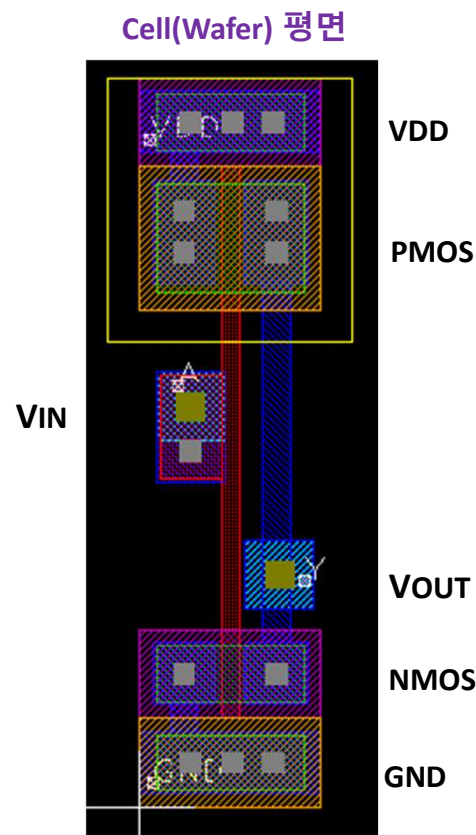
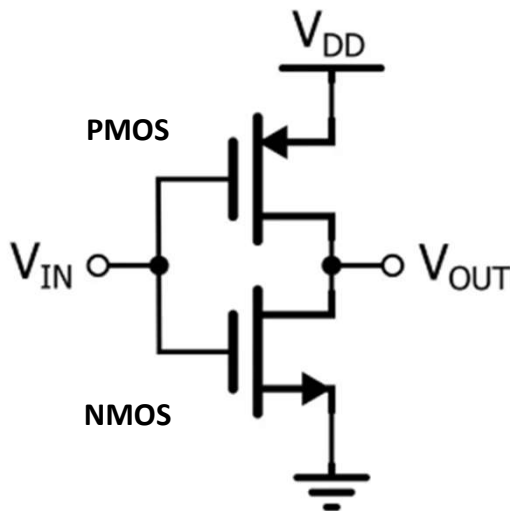
- No channel is formed.
- Current flow is blocked.

When the gate voltage is high (ON)

- The gate is at a higher voltage than the p-substrate.
- An electron channel forms at the surface of the p-substrate.
- Current flows (Drain → Source).

2. Inverter Cell and operation of CMOS

Inverter



PMOS Operating Principle

When the gate voltage is low (ON)

- The gate is at a lower voltage than the n-well.
- A hole channel forms at the surface of the n-well.
- Current flows (Source $\rightarrow$ Drain).

When the gate voltage is high (OFF)

- No channel is formed.
- Current flow is blocked.

NMOS Operating Principle

When the gate voltage is low (OFF)

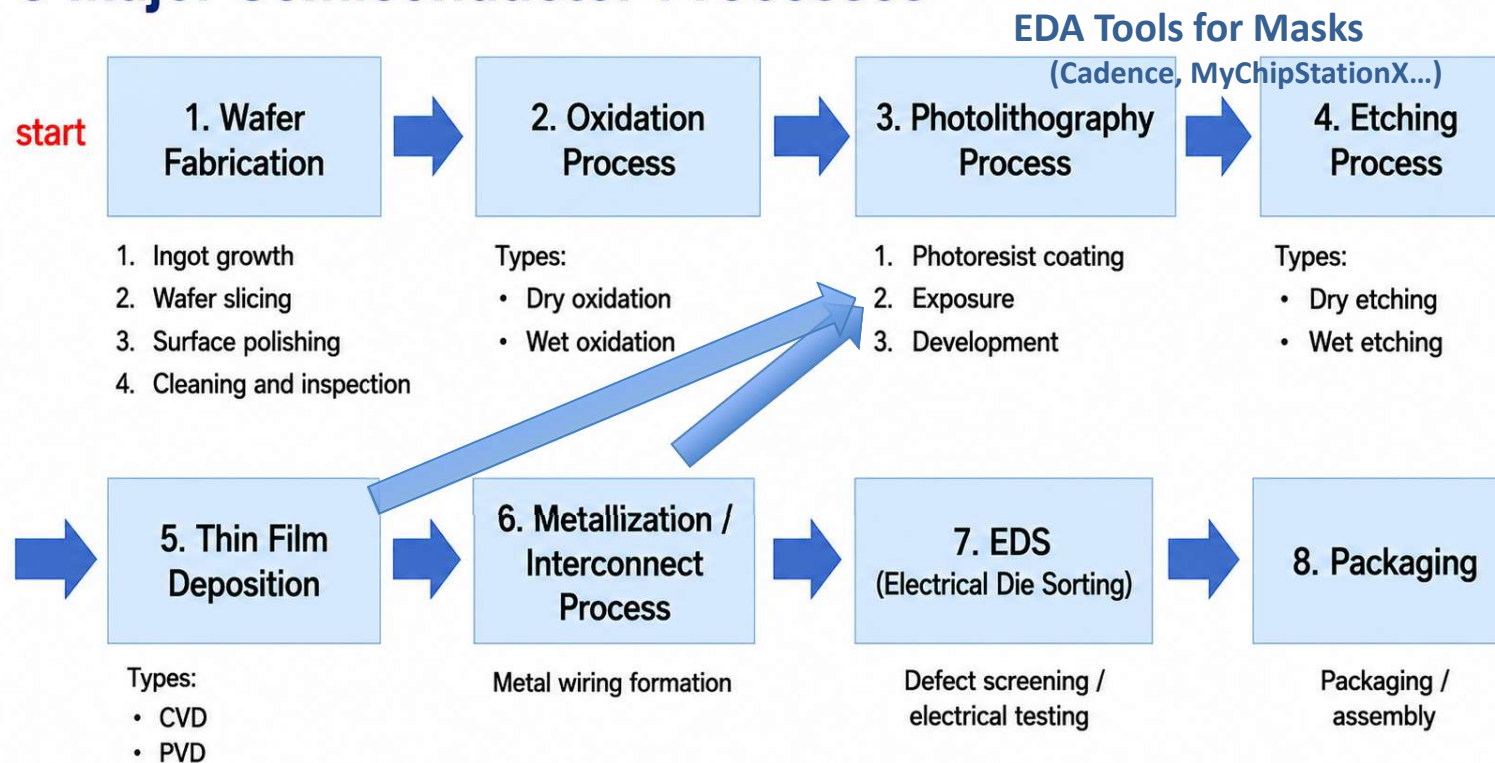
- No channel is formed.
- Current flow is blocked.

When the gate voltage is high (ON)

- The gate is at a higher voltage than the p-substrate.
- An electron channel forms at the surface of the p-substrate.
- Current flows (Drain $\rightarrow$ Source)

3. Understanding IC design and production

8 Major Semiconductor Processes

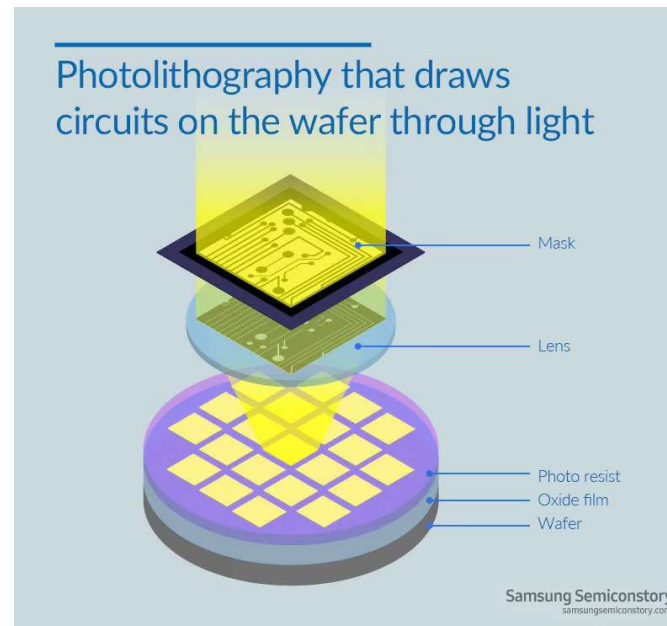


3. Understanding IC design and production

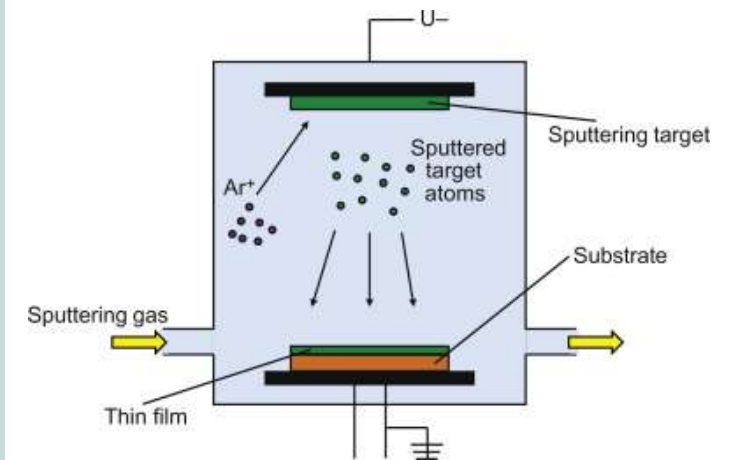
Some examples of semiconductor fabrication Process



Ingot/Wafer

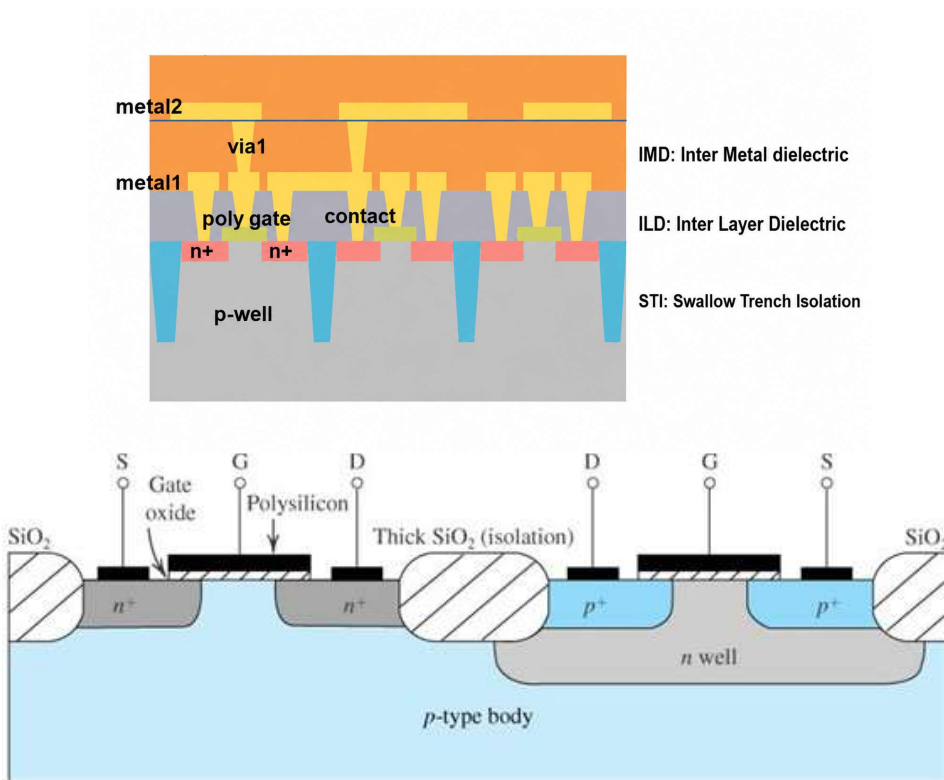


Photolithography



**Physical Vapor
Deposition**

3. Understanding IC design and production



The purpose of EDA tools is to create accurate masks required for the semiconductor manufacturing process.

CMOS Fabrication Flow

1. Wafer Preparation

- Start with a silicon wafer
- Form the initial oxide layer

2. Isolation Formation (Active Mask / STI)

- Define active regions
- Form **STI (Shallow Trench Isolation)** by filling non-active regions with oxide

- PR coating → Exposure → Development → Etching / Oxide fill

3. Well Formation

- Form **n-well** and/or **p-well** regions using well masks
- PR coating → Exposure → Development → Ion implantation + Annealing

4. Gate Oxide Growth

- Grow a thin gate oxide layer

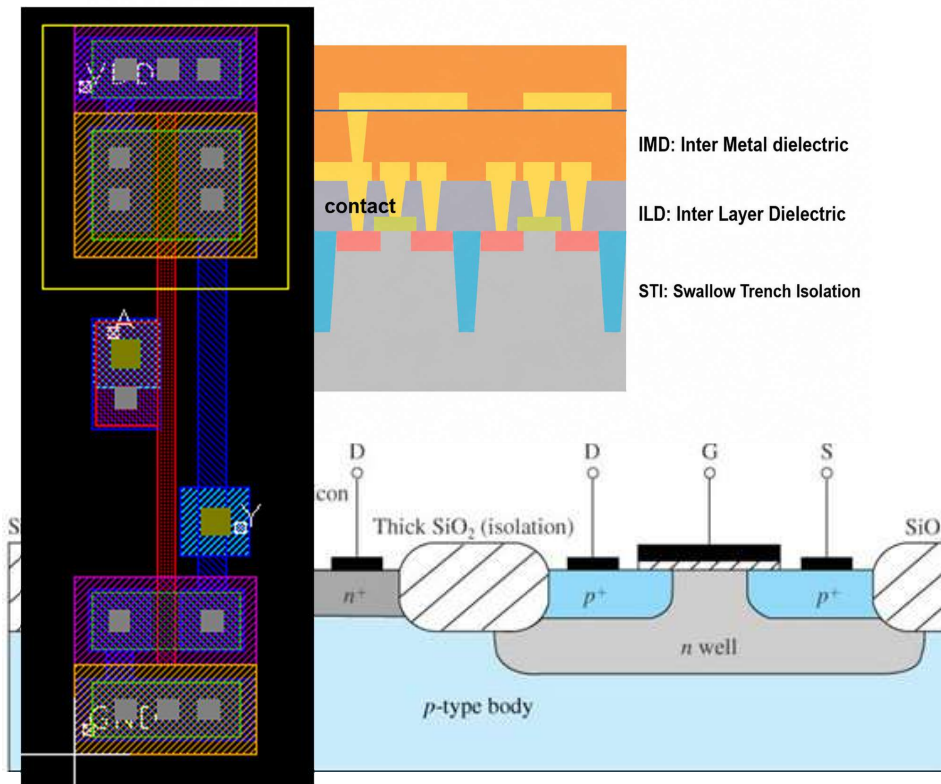
5. Polysilicon Deposition

- Deposit a polysilicon layer for gate formation

6. Gate Formation (Poly Mask)

- Pattern the polysilicon gate
- PR coating → Exposure → Development → Etching

3. Understanding IC design and production



The purpose of EDA tools is to create accurate masks required for the semiconductor manufacturing process.

CMOS Fabrication Flow (Continue)

7. Source/Drain Formation

- Define source/drain regions using implant masks
- PR coating → Exposure → Development
- LDD implantation → Spacer formation → Heavy source/drain implantation

8. Interlayer Dielectric (ILD) Deposition

- Deposit the ILD layer

9. Contact Formation (Contact Mask)

- Open contact holes to source/drain and gate regions
- PR coating → Exposure → Development → Etching

10. Contact Fill

- Fill contact holes with a conductive material such as tungsten

11. Metal Interconnect Formation (Metal Mask)

- Deposit the metal layer
- Pattern interconnects using photolithography and etching
- Metal deposition → PR coating → Exposure → Development → Etching

12. Passivation / Final Processing

- Deposit the passivation layer for protection
- Complete final testing and packaging

3. Understanding IC design and production

Process Design Kit(PDK)



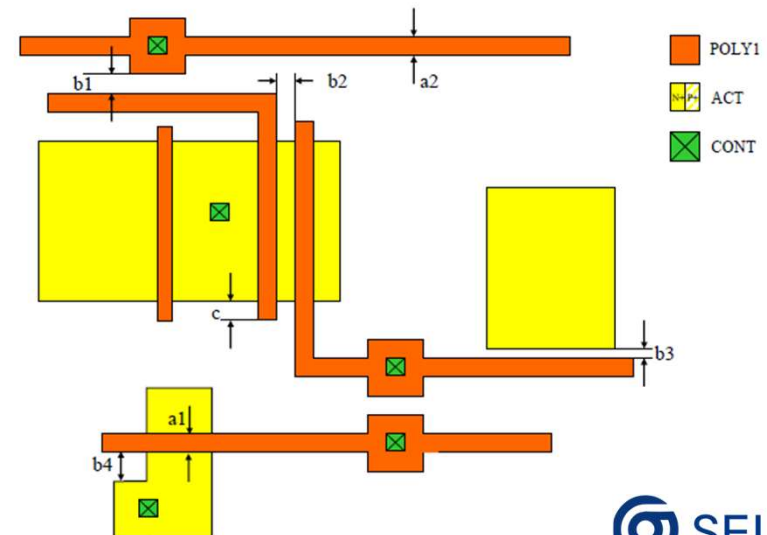
2. Layer Information

2.1 Layers for Standard Process

Layer Name	Layer No.	Function	Digitized Area
CHIPBOUNDARY	0	Define chip boundary	C
KEY	1	Define Align key	C
NWELL	2	Define N-well for PMOS body	C
PWELL	3	Define P-well for NMOS body	C
ACT	11	Define Active area	D
POLY1	12	Define Poly-Si 1 for gate of transistor	D
POLY2	13	Define Poly-Si 2 for resistor and capacitor	D
NSD	14	Define N-type implant region for N-LDD and N+	C
PSD	15	Define N-type implant region for P-LDD and P+	C
ESDI	26	Define ESD implant	C
CONT	31	Define contact from MET1 to ACT, POLY1 and POLY2	C
MET1	32	Define Metal-1 for interconnection	D
VIA1	33	Define Via1 connecting MET1 and MET2	C
MET2	34	Define Metal-2 for interconnection	D
VIA2	35	Define Via2 connecting MET2 and MET3	C
MET3	36	Define Metal-3 for interconnection	D
PAD	37	Define open region of passivation for bonding pad	C

4.3 POLY1 Rules ETRI 500nm FAB (Poly1 width > 500nm)

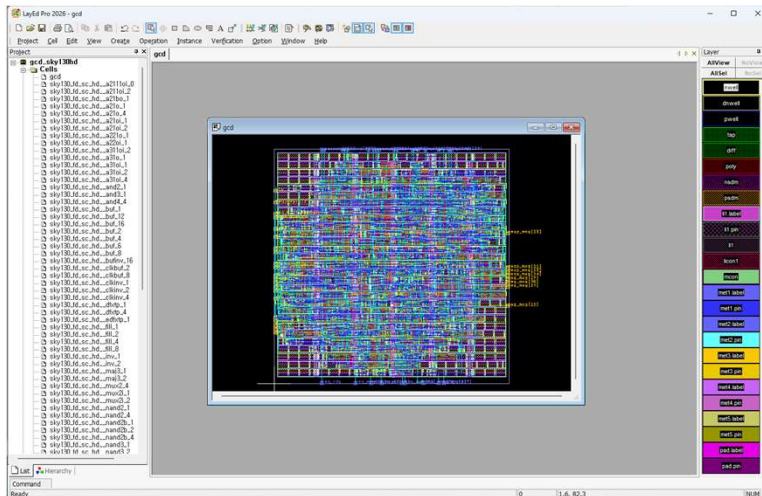
Rule No.	Description	Label		Value (μm)
3.PL1.W1	POLY1 width over ACT (NMOS or PMOS)	a1	≧	0.5
3.PL1.W2	Maximum POLY1 width (Transistor Channel Length)	a1	≦	100.0
3.PL1.W3	POLY1 width for interconnect	a2	≧	0.5
3.PL1.S1	POLY1 to POLY1 space in field	b1	≧	0.6
3.PL1.S2	POLY1 to POLY1 space on ACT	b2	≧	0.6
3.PL1.S3	Space from POLY1 in field to ACT	b3	≧	0.5
3.PL1.S4	ACT space to POLY1 in field	b4	≧	0.5
3.PL1.O	Overlap of POLY1 extended into field oxide	c	≧	0.5
3.PL1.ANT	Maximum ratio of field poly area to gate poly for a given transistor(antenna rule)		≦	200



4. MyChipStationX - introduction

MyChipStationX is a modernized semiconductor design platform developed by Selconn, based on proven MyCAD technology heritage dating back to 1990 and worldwide deployment across industrial, academic, and research environments.

- A complete EDA environment for analog/digital IC design
- Layout editing and verification (DRC/ERC/LVS/PEX)



SchedPro : Schematic Capture

Logic2Spice : Schematic to Spice netlist conversion

LayEdPro : MyChipStationX main program including Layout Editor

CifGdsPro : File format converter, including **GDSII** import/export

MyDRCPro : Design rule checking(FAB specific)

LayNetPro : ERC checking & netlist extraction

MyLVSPPro : Layout vs Schematic comparison

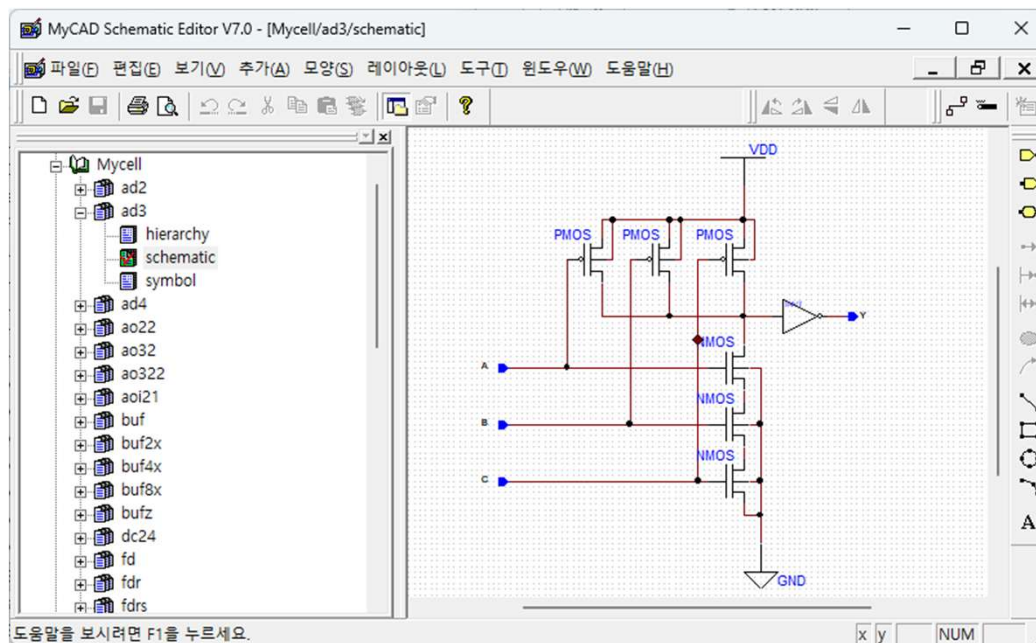
MyOpenROAD : OpenROAD (open-source Synthesis & PnR tool) Launcher

*GDSII (Graphic Data System II): Final layout file for photomask generation.)

4. MyChipStationX – Schematic Capture

SchedPro : Schematic Editor/Capture

Logic2Spice : Schematic to Spice netlist conversion



AND3 Schematic

The screenshot shows the netlist output generated by the Logic2Spice tool. The netlist is displayed in a window titled 'KANC_NNFC drc.log Inverter.cir'. The netlist content is as follows:

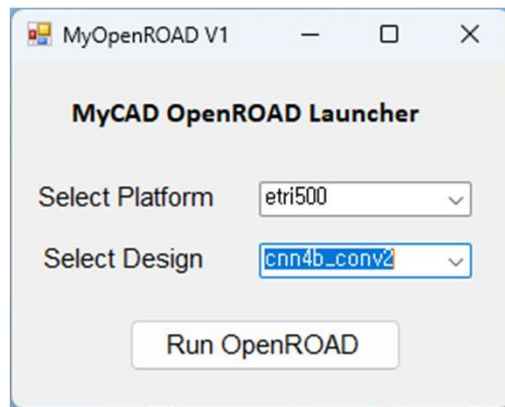
```
*****  
* MyAnalog V7.0 SPICE netlist generator  
* Cell Name :: ad3  
* Flatten Extraction for Berkely SPICE 3  
* Generated Date :: 2026/4/23 10:14:31  
*****  
*.GLOBAL A B C Y  
*.GLOBAL VDD GND  
*****  
MT_I5 I7_PNS A I4_NND GND NMOS L=0.5U W=1.6U  
MT_I1 VDD A I7_PNS VDD PMOS L=0.5U W=3.2U  
MT_I0 VDD B I7_PNS VDD PMOS L=0.5U W=3.2U  
MT_I4 I4_NND B I6_NND GND NMOS L=0.5U W=1.6U  
MT_I6 I6_NND C GND GND NMOS L=0.5U W=1.6U  
MT_I7 VDD C I7_PNS VDD PMOS L=0.5U W=3.2U  
MT_I8_I1 VDD I7_PNS Y VDD PMOS L=0.5U W=3.2U  
MT_I8_I5 Y I7_PNS GND GND NMOS L=0.5U W=1.6U  
*****  
VDD VDD 0 DC 5  
VSS GND 0 DC 0  
*****
```

The status bar at the bottom indicates '줄 1, 열 1 9,195자 일반 텍스트 100% Windows (CRLF) UTF-8'.

Netlist of AND3

4. MyChipStationX – Synthesis and PnR

MyOpenROAD.exe : OpenROAD launcher for synthesis and PnR



MyOpenROAD.exe

```
Windows PowerShell
..../flow/logs/etri500/stopwatch/base
Log
Elapsed/s Peak Memory/MB sha1sum .odb [0:20)
1_1_yosys_canonicalize 0 22 fe60d3595119b88f8a4b
1_2_yosys 0 26 b13215b11904f155c861
1_synth 0 91 971505229e3f3f5820d1
2_1_floorplan 0 93 ad8726627e187d8dc74f
2_2_floorplan_macro 0 91 ad8726627e187d8dc74f
2_3_floorplan_tapcell 0 90 9fd8f23eea672ff00d33
2_4_floorplan_pdn 0 92 69317c3f5e692548e557
3_1_place_gp_skip_io 0 91 e727e99805aa387db94d
3_2_place_iop 0 90 c756d5c28587c7d9524e
3_3_place_gp 1 136 28ea289c4b2206803fc4
3_4_place_resized 0 110 688f98c46c219813c6d5
3_5_place_dp 0 94 62e8e9d1cef18aad6ada
4_1_cts 1 119 4befe9053f1d4103afaf
5_1_grt 1 166 197acf6b1dbc8e71a50e
5_2_route 8 430 a8fdcbbd869a588353096
5_3_fillcell 0 92 cf92e0a16e0d1f74d503
6_1_fill 0 90 cf92e0a16e0d1f74d503
6_1_merge 2 418 N/A
6_report 1 192 N/A
Total 14 430
2026-06-11 22:24:17 Build finished. ExitCode=0
2026-06-11 22:24:17 DONE
2026-06-11 22:24:17 ==== Build end: 06/11/2026 22:24:17 ====
PS C:\MyChipProX\openroad>
```

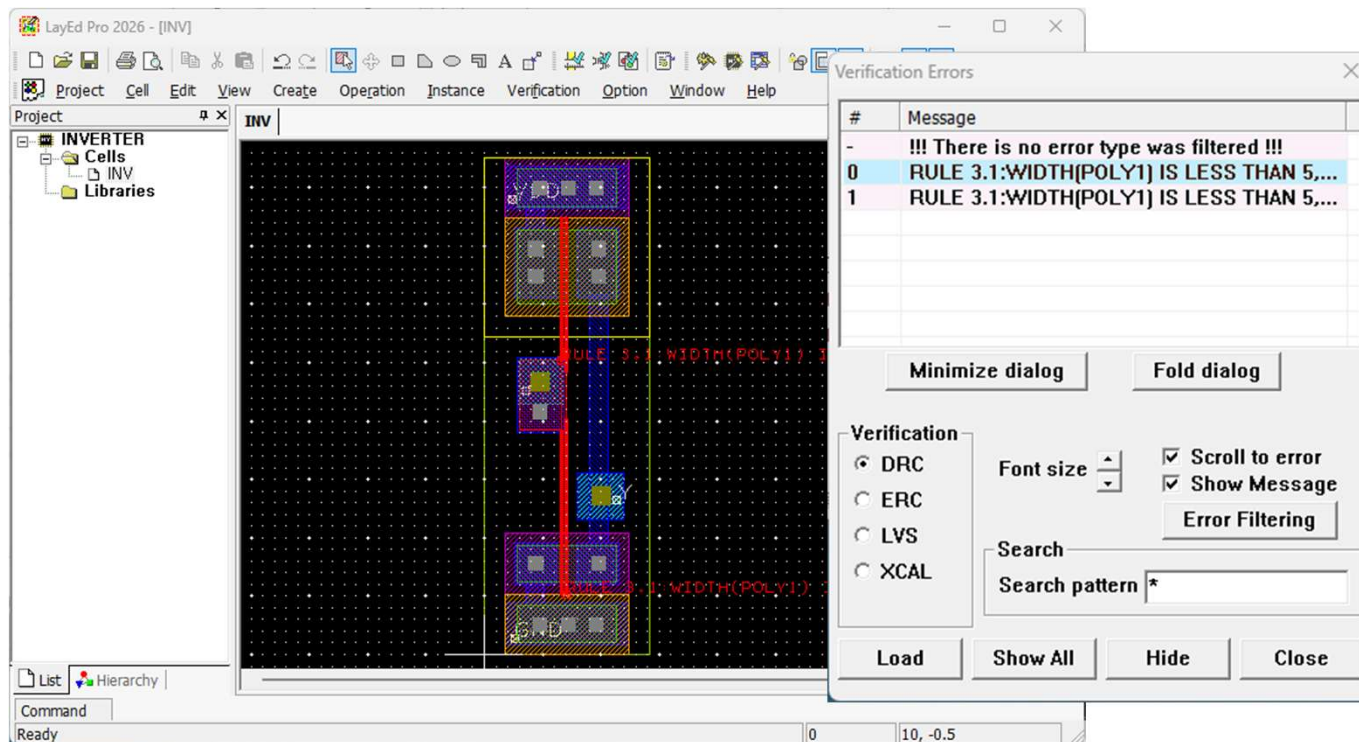
Result of OpenROAD

CifGdsPro : File format converter, including GDS import/export



4. MyChipStationX - Verification

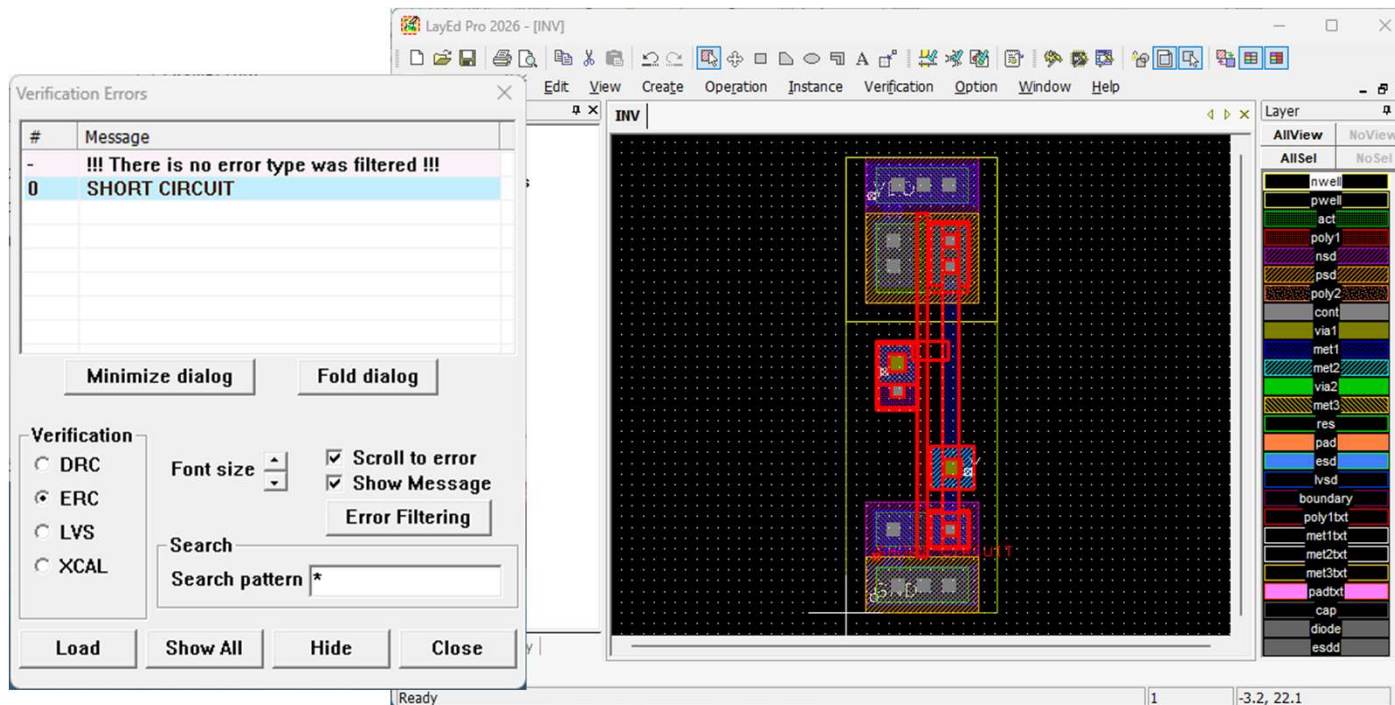
MyDRCPro : Design rule checking(FAB specific)



Design Rule Check result – 2 errors, poly1 width error (less than 500nm)

4. MyChipStationX - Verification

MyERCPro : Electrical rule checking (Detection of Electrical Rule Violations)



Electrical Rule Check result – 1 errors, short between input A and output Y

4. MyChipStationX - Verification

MyLvsPro : Layout versus Schematic checking

The screenshot displays the LayEd Pro 2026 interface. The main window shows a layout of an inverter cell. A 'Verification Errors' dialog box is open, showing a single error: '!!! There is no error type was filtered !!! DISCREPANCY 1'. The 'Verification' section on the left has 'LVS' selected. A netlist window titled 'Inverter.cir' is also open, showing the circuit definition for the inverter. The netlist includes parameters for NMOS and PMOS transistors, with the NMOS gate width highlighted as 'L=0.4U'.

Verification Errors

#	Message
0	!!! There is no error type was filtered !!! DISCREPANCY 1

Verification
☐ DRC
☐ ERC
☒ LVS
☐ XCAL

Font size: [dropdown]
☒ Scroll to error
☒ Show Message
Error Filtering

Search
Search pattern: *

Load Show All Hide Close

Layer
AllView
AllSel

netlist
nwell
pwell
act
poly1
nsd
psd
poly2
cont
via1
met1
met2
via2
met3
res
pad
esd
lvsd
boundary
poly1bxt
met1bxt
met2bxt
met3bxt
padbxt
cap
diode
esdd

KANC_NNFC_CMOS • Inverter.cir

파일 편집 보기

```
*****  
* MyAnalog V7.0 SPICE netlist generator  
* Cell Name :: Inverter  
* Flatten Extraction for Berkely SPICE 3  
* Generated Date :: 2026/4/23 10:7.45  
*****  
*.GLOBAL IN0 OUT0 VDD GND  
*****  
MT_I0 OUT0 IN0 GND GND NMOS L=0.4U W=1.6U  
MT_I1 VDD IN0 OUT0 VDD PMOS L=0.5U W=3.2U  
*****  
*====Spice Command Examples=====  
*.op : DC operating Point Analysis
```

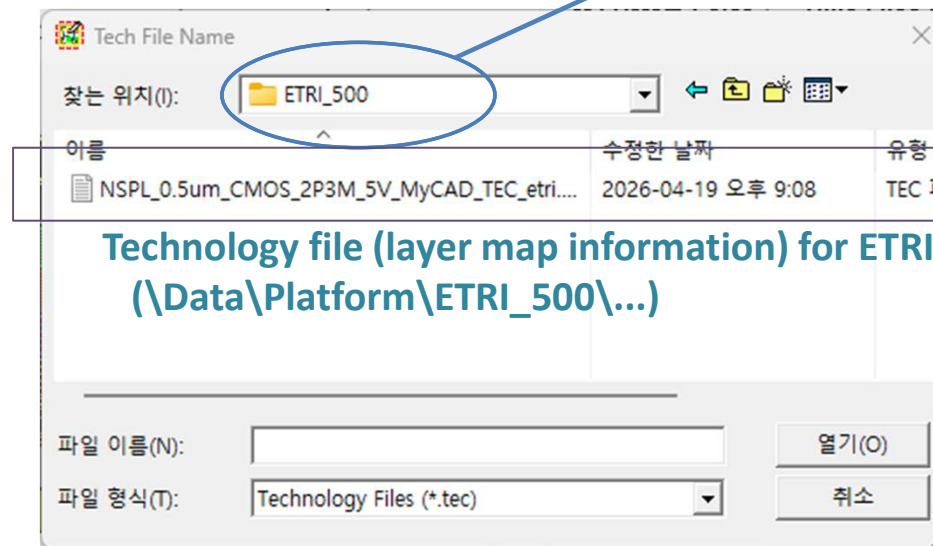
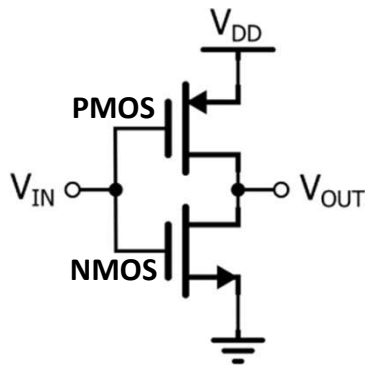
줄 10, 열 34, 8,783자 일반 텍스트 100% Windows (CRL UTF-8

LVS Check result – 1 errors, discrepancy between NMOS Gate width in circuit and Layout gate width

5. Cell design example - Inverter

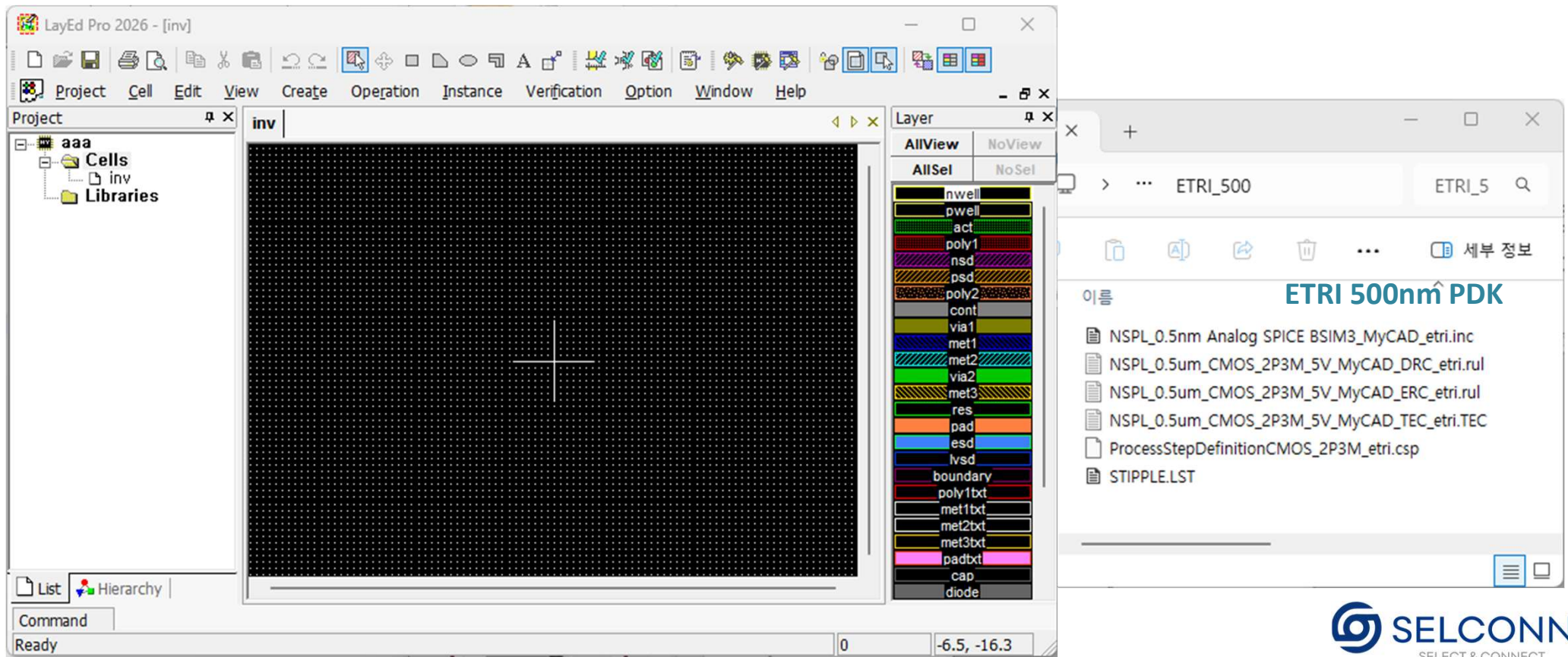
What You Need

1. Schematic
2. Platform PDK files (Technology, Rule, and Mapping Files for Semiconductor Fabrication)
3. Generating Project on LayEdPro.exe

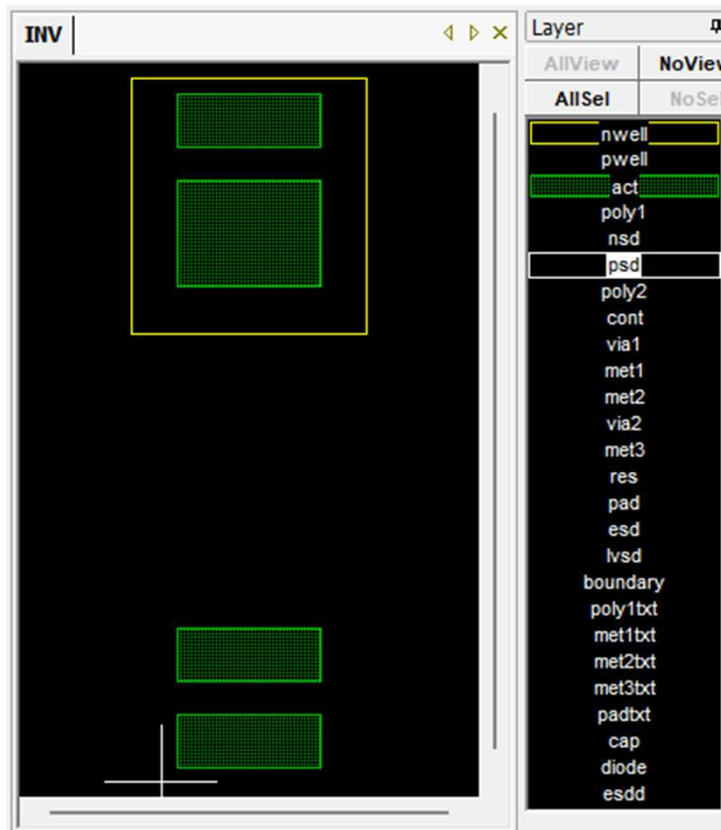


5. Cell design example - Inverter

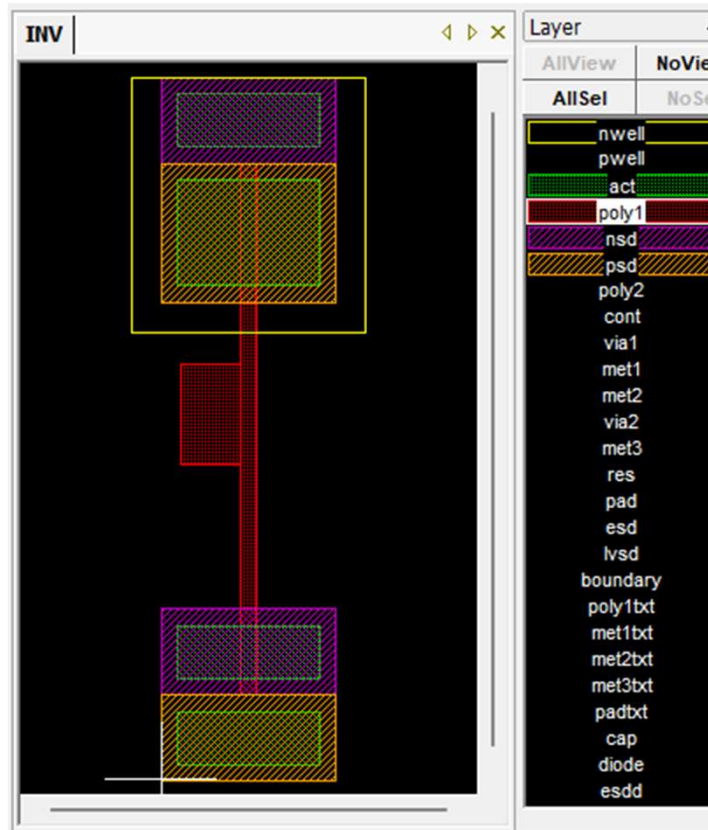
Ready for ETRI 500 process (Cell -> new cell -> generate 'inv cell')



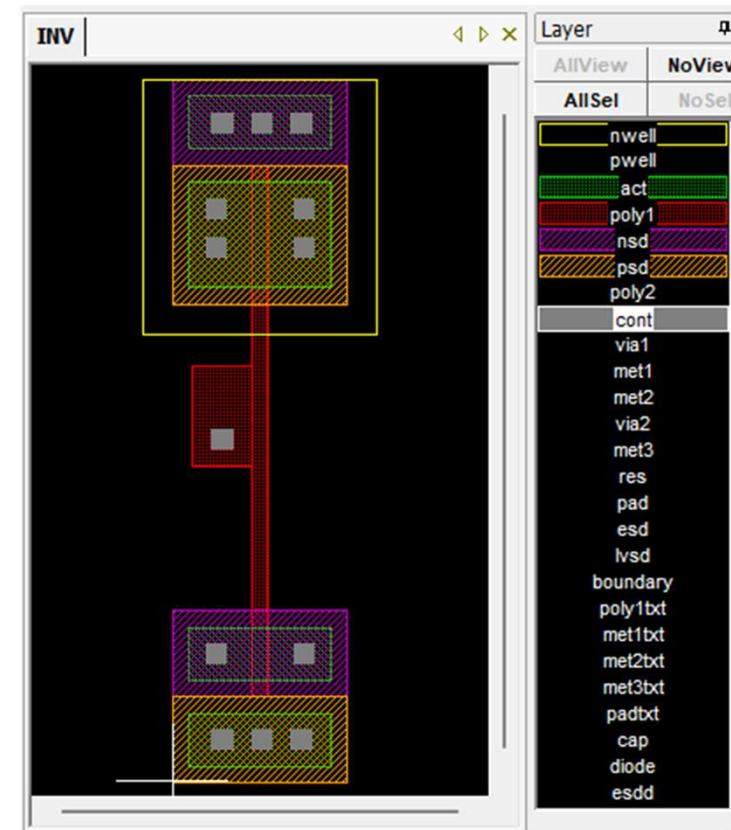
5. Cell design example - Inverter



n-well/act

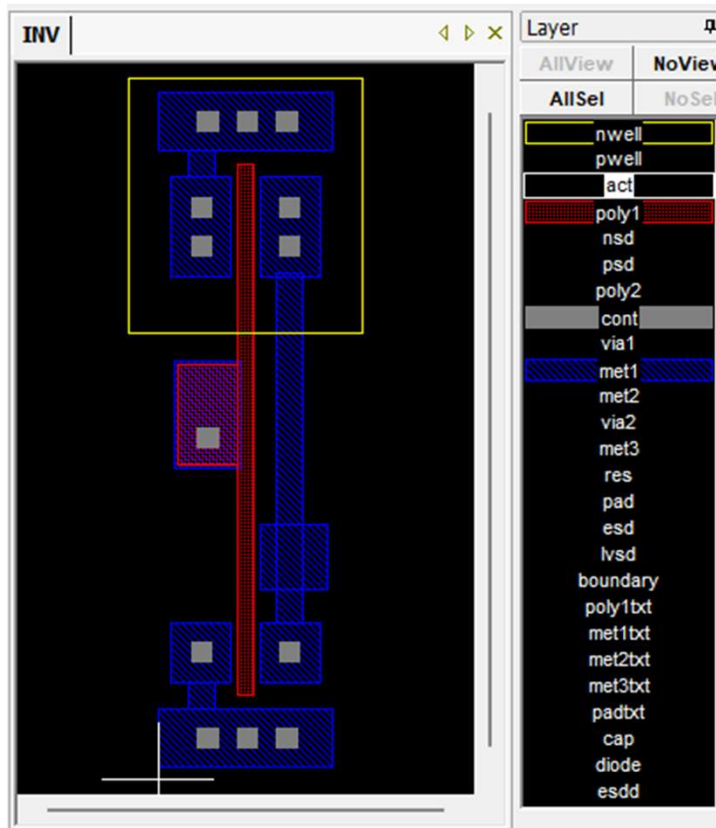


poly1/psd/nsd

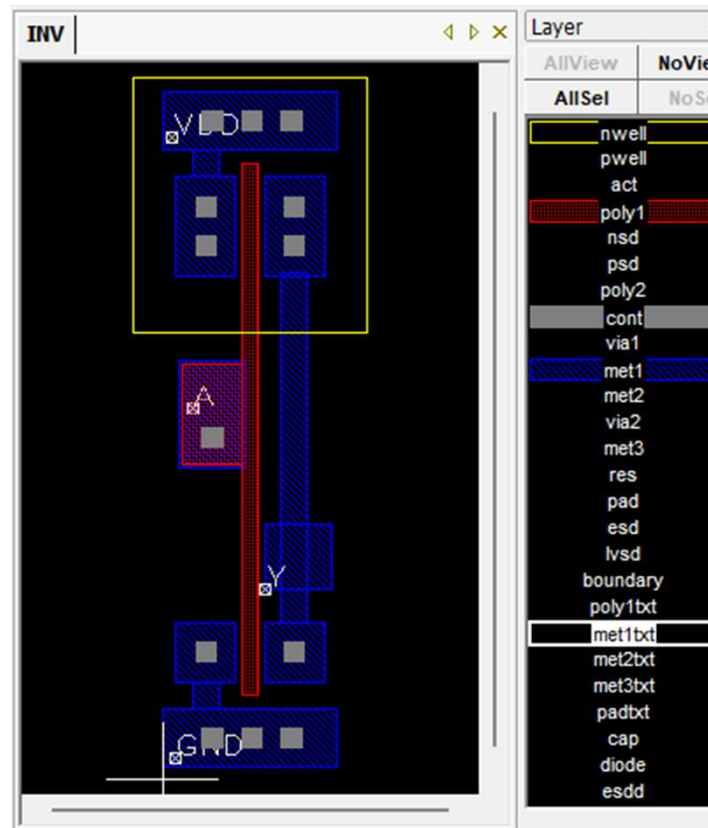


contact

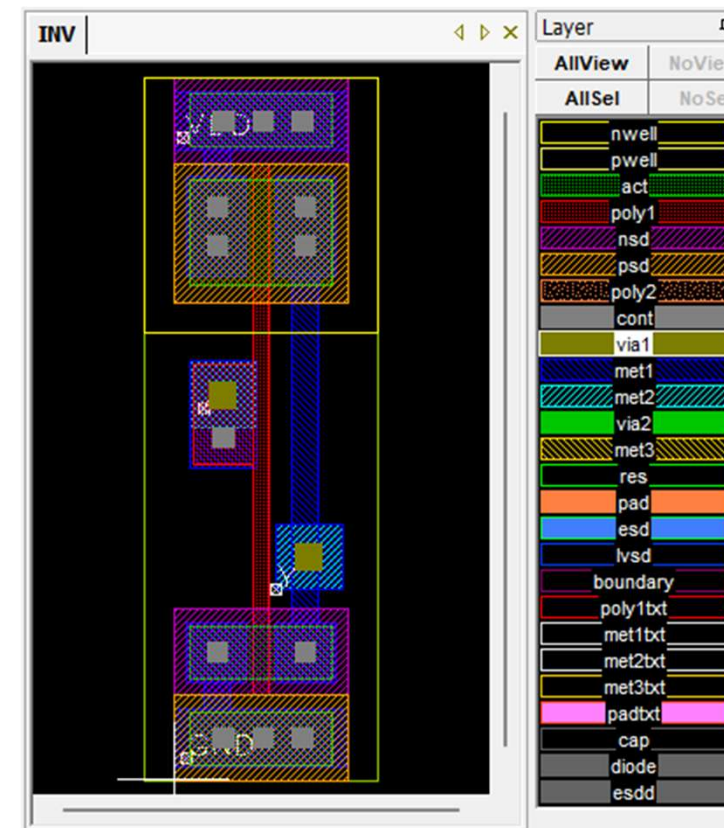
5. Cell design example - Inverter



metal1



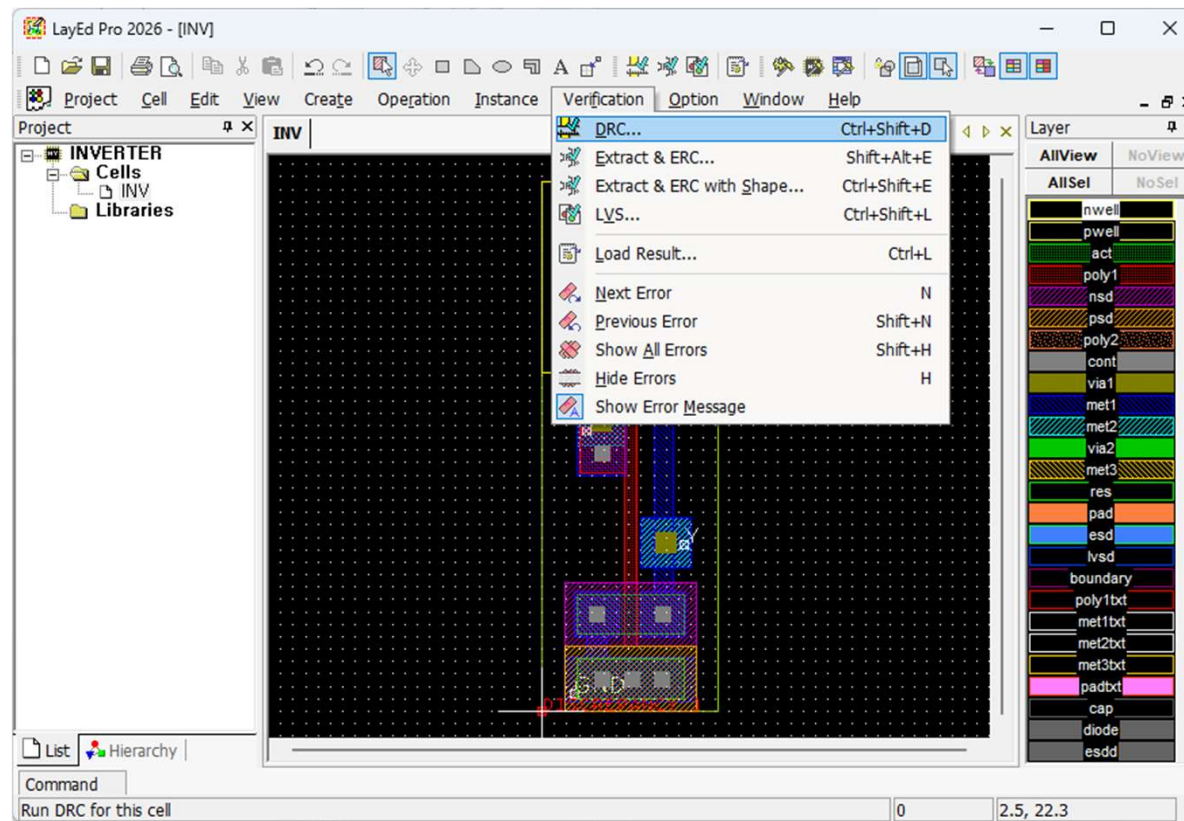
text1(label)



via1/metal2

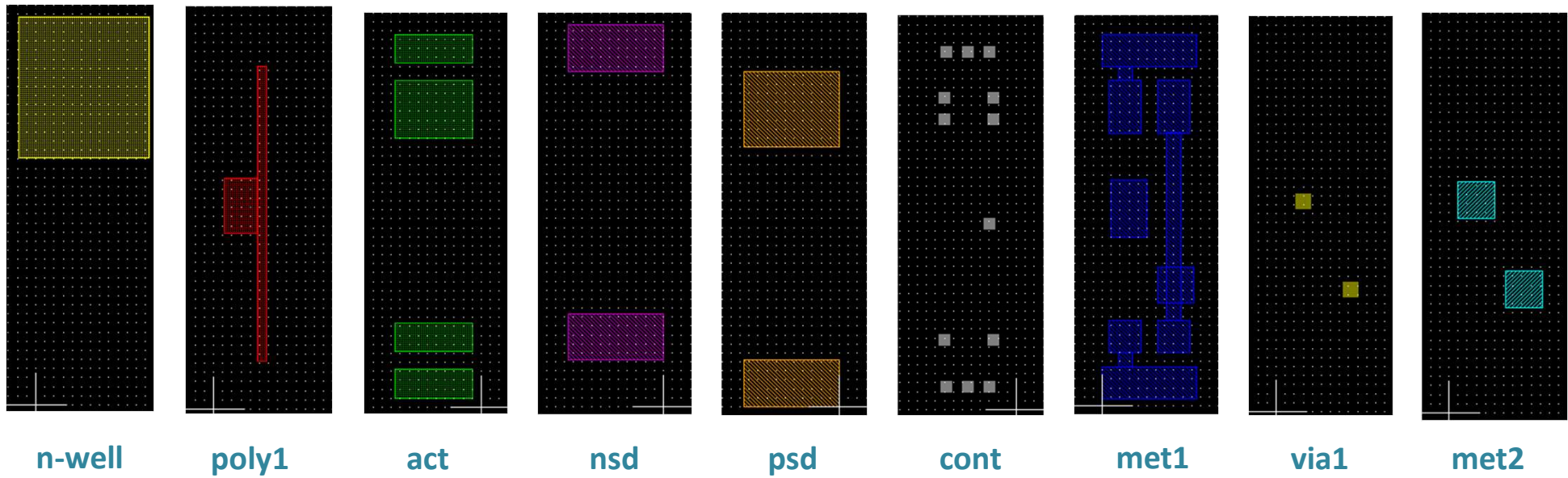
5. Cell design example - Inverter

Verification (DRC/ERC/LVS)



5. Cell design example - Inverter

Generated Masks

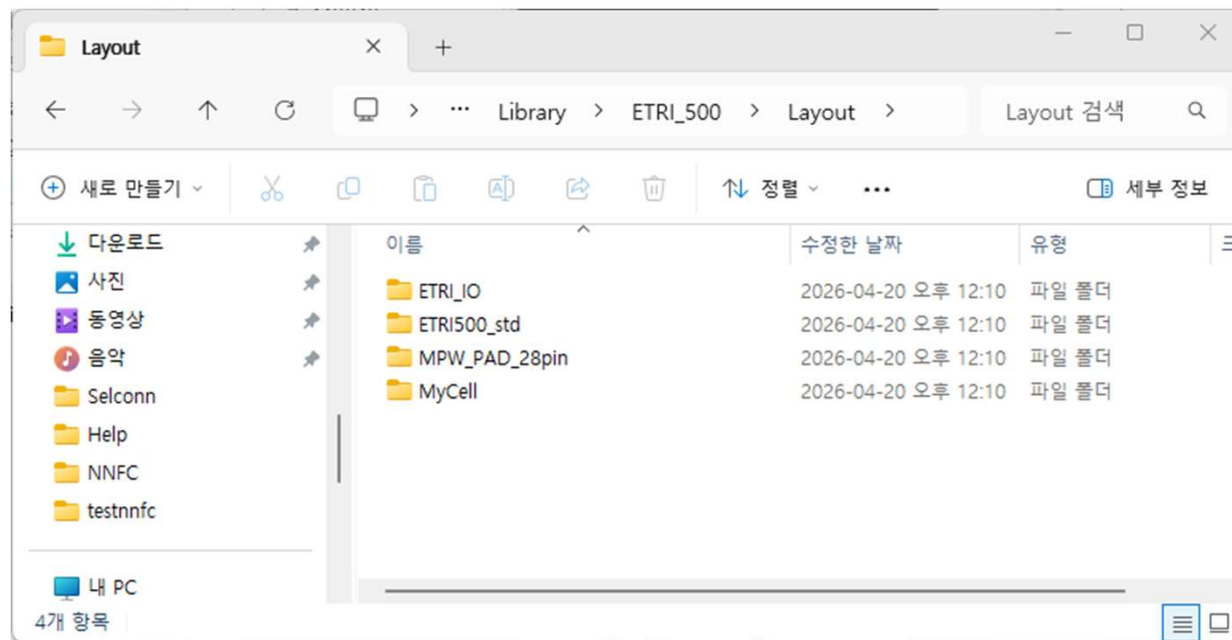


6. Design Stopwatch : Circuit (bottom-up)

ETRI500 MPW(multi-project on wafer) Platform

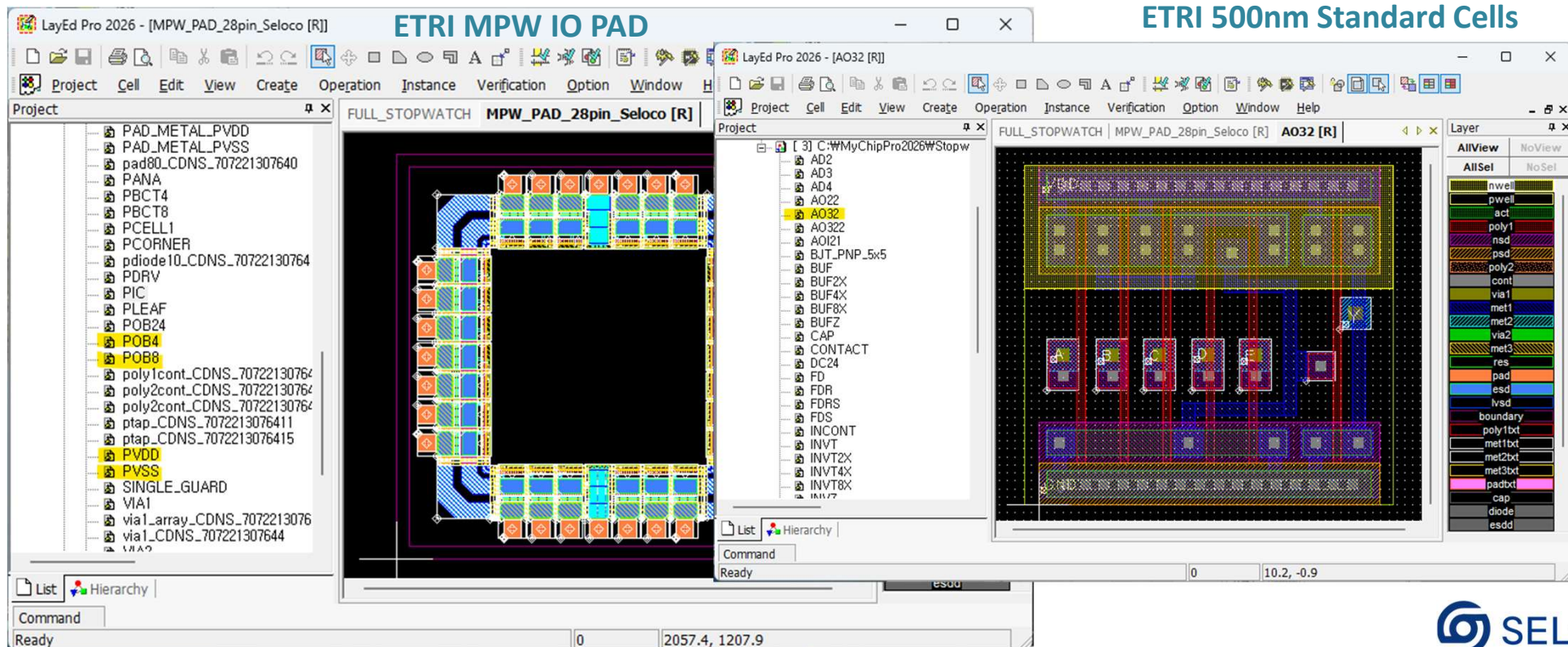
ETRI500 Standard Cell library

ETRI500 IO, PAD Library



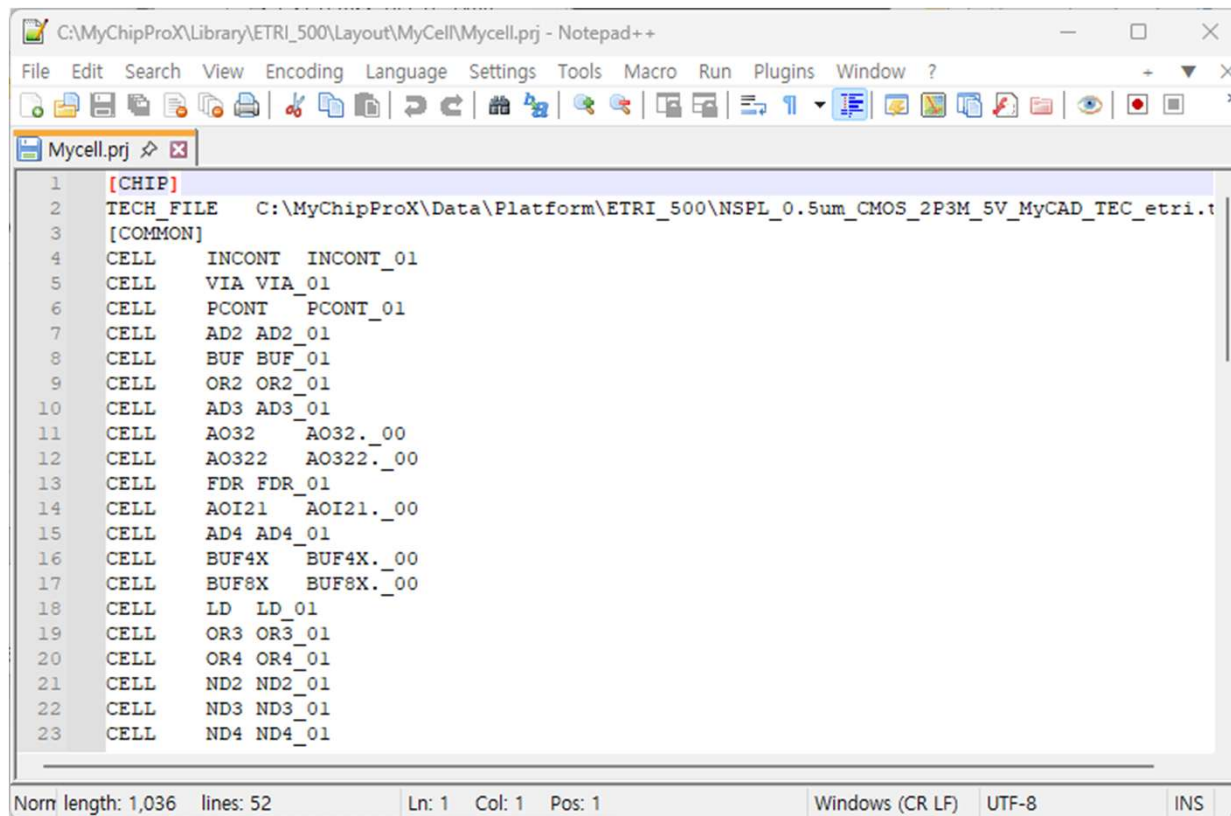
6. Design Stopwatch : Circuit (bottom-up)

ETRI500 IO, PAD Library – PIC(input pad), POB4(output pad, PVDD, PVSS)
ETRI500 MPW(multi-project on wafer)



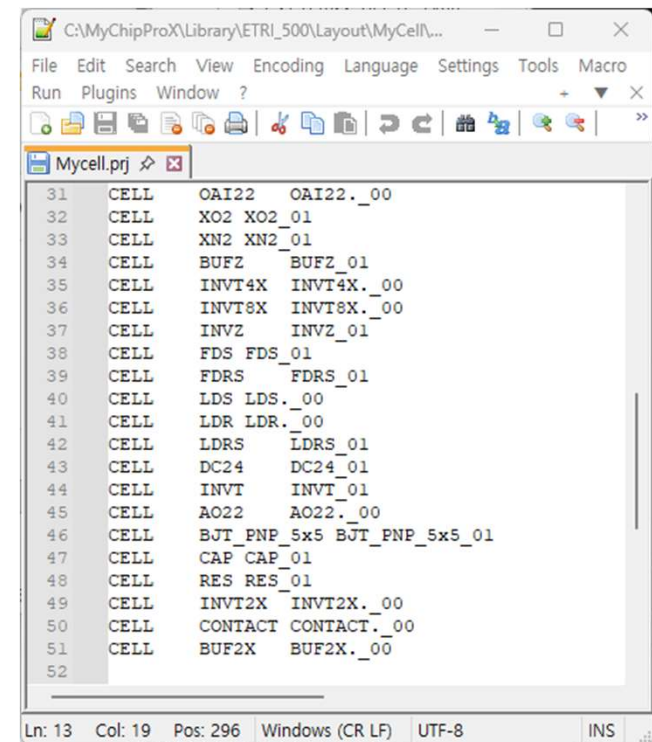
6. Design Stopwatch : Circuit (bottom-up)

ETRI500용 Standard Cell – Layout/Circuit, total 48 logic/contact/other cells



```
C:\MyChipProX\Library\ETRI_500\Layout\MyCell\Mycell.prj - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
MyCell.prj
1 [CHIP]
2 TECH_FILE C:\MyChipProX\Data\Platform\ETRI_500\NSPL_0.5um_CMOS_2P3M_5V_MyCAD_TEC_etri.t
3 [COMMON]
4 CELL INCONT INCONT_01
5 CELL VIA VIA_01
6 CELL PCONT PCONT_01
7 CELL AD2 AD2_01
8 CELL BUF BUF_01
9 CELL OR2 OR2_01
10 CELL AD3 AD3_01
11 CELL AO32 AO32_00
12 CELL AO322 AO322_00
13 CELL FDR FDR_01
14 CELL AOI21 AOI21_00
15 CELL AD4 AD4_01
16 CELL BUF4X BUF4X_00
17 CELL BUF8X BUF8X_00
18 CELL LD LD_01
19 CELL OR3 OR3_01
20 CELL OR4 OR4_01
21 CELL ND2 ND2_01
22 CELL ND3 ND3_01
23 CELL ND4 ND4_01
```

Norm length: 1,036 lines: 52 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS



```
C:\MyChipProX\Library\ETRI_500\Layout\MyCell\...
File Edit Search View Encoding Language Settings Tools Macro
Run Plugins Window ?
MyCell.prj
31 CELL OAI22 OAI22_00
32 CELL XO2 XO2_01
33 CELL XN2 XN2_01
34 CELL BUF2 BUF2_01
35 CELL INVT4X INVT4X_00
36 CELL INVT8X INVT8X_00
37 CELL INVZ INVZ_01
38 CELL FDS FDS_01
39 CELL FDRS FDRS_01
40 CELL LDS LDS_00
41 CELL LDR LDR_00
42 CELL LDRS LDRS_01
43 CELL DC24 DC24_01
44 CELL INVT INVT_01
45 CELL AO22 AO22_00
46 CELL BJT_PNP_5x5 BJT_PNP_5x5_01
47 CELL CAP CAP_01
48 CELL RES RES_01
49 CELL INVT2X INVT2X_00
50 CELL CONTACT CONTACT_00
51 CELL BUF2X BUF2X_00
52
```

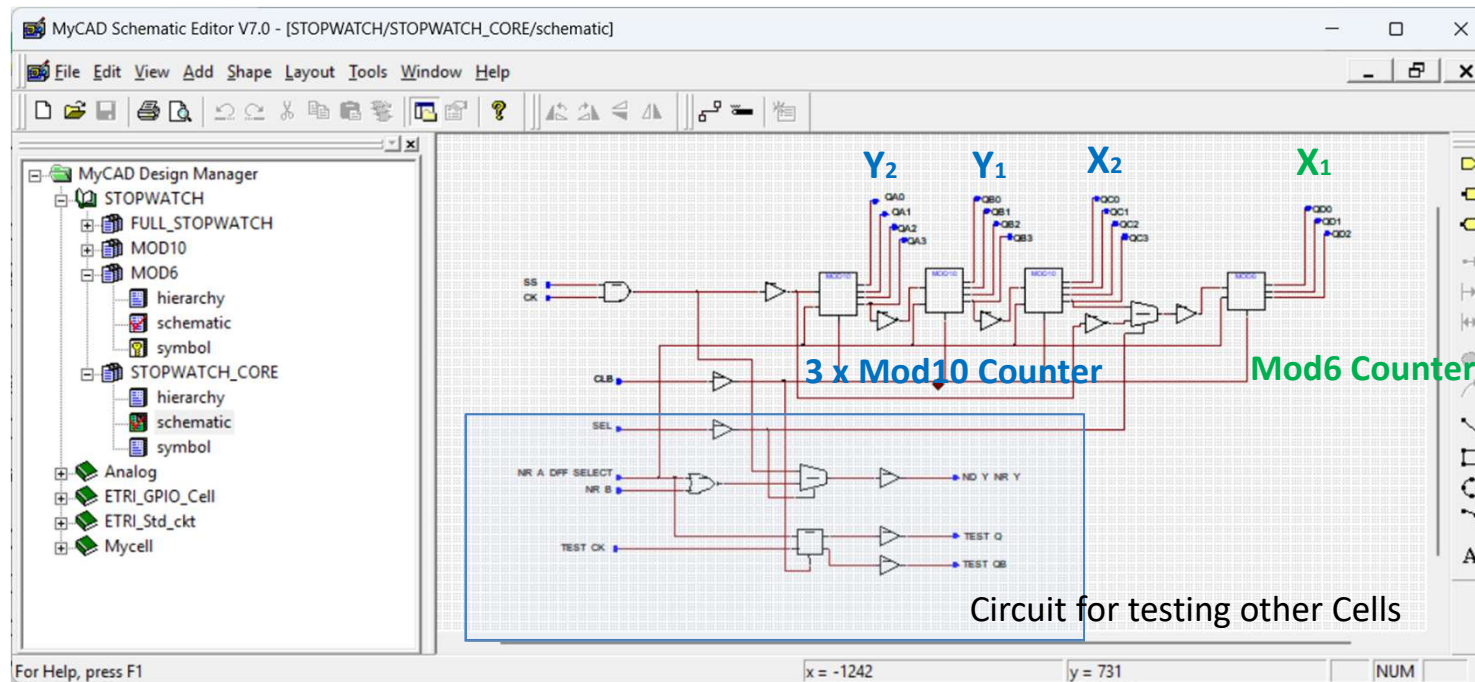
Ln: 13 Col: 19 Pos: 296 Windows (CR LF) UTF-8 INS

6. Design Stopwatch : Circuit (bottom-up)

Modular circuit design on ScheEd.exe

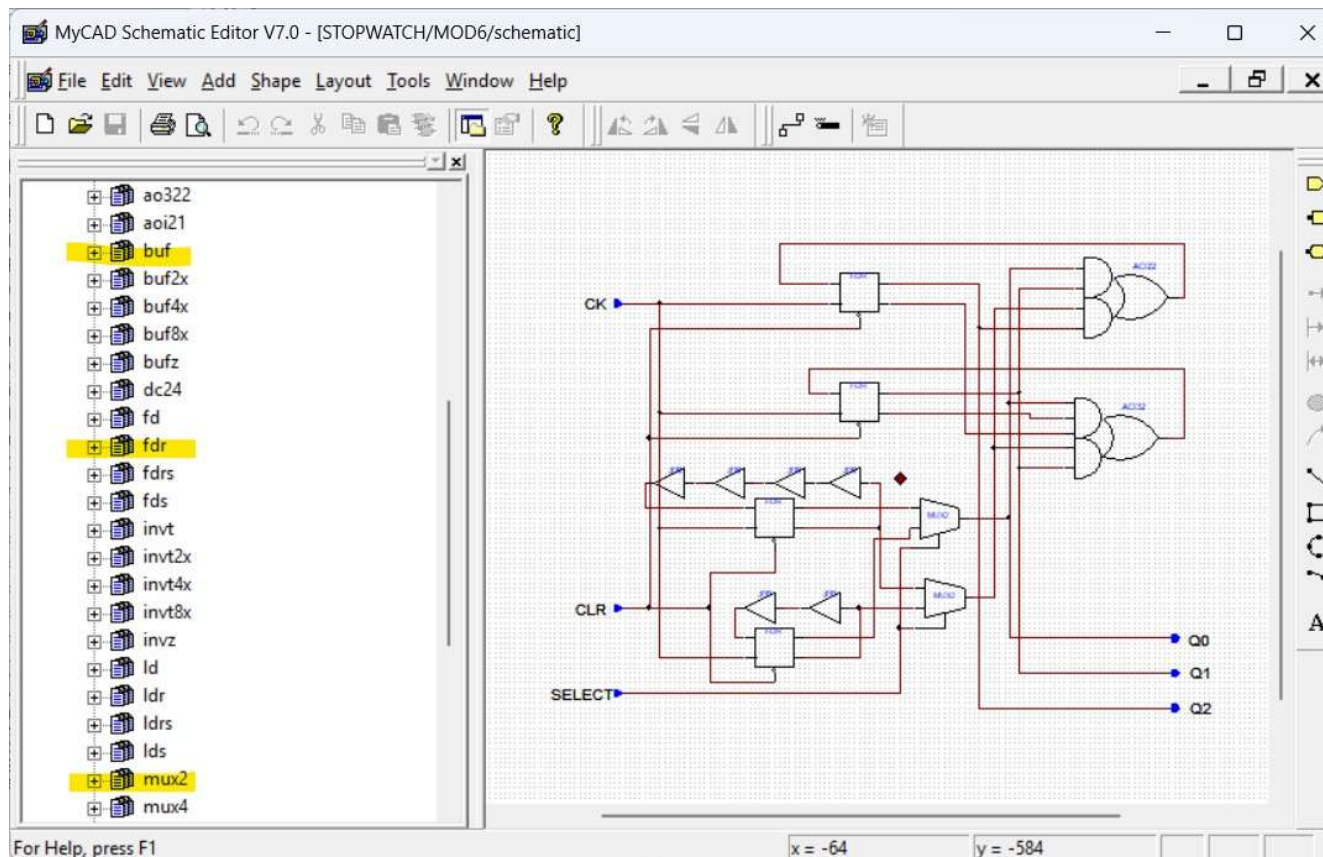
1. 2 digits for SECOND (0~59), 2 digits for CENTI-SECOND (0~99)
2. Start/Pause key, Reset key, Clock(100Hz)
3. 3 BCD counter (0~9 count), 4-pin BCD output
4. 1 mod 6 counter (0~5 count), 3-pin output

$X_1X_2:Y_1Y_2$



6. Design Stopwatch : Circuit (bottom-up)

Mod6 counter



Standard Cell Circuit

- 4 FDR
flip-flop with Reset
- 6 BUF
buffer
- 2 MUX2
2-input mux
- 1 AO22 cell
OR (2-input AND) (2-input AND)
- 1 AO32 cell
OR (2-input AND) (3-input AND)

6. Design Stopwatch : Circuit (bottom-up)

Check Circuit, Spice Net List Extract

Tools -> Check Circuit

Tools -> Spice Netlist

The screenshot displays the MyCAD Schematic Editor V7.0 interface. The main window shows a circuit diagram with various components and connections. The left pane shows the MyCAD Design Manager tree, which includes a project named 'STOPWATCH' containing sub-projects like 'FULL_STOPWATCH', 'MOD10', 'MOD6', and 'STOPWATCH_CORE'. The 'STOPWATCH_CORE' sub-project is expanded, showing its hierarchy and schematic. The right pane shows the 'Check circuit' dialog box, which lists errors and warnings. The 'Errors' list includes: Invalid net/bus name, Net disconnect, Bus has no name, Invalid bus width, Invalid net index, Bus size mismatch, Short, Floating input, In-port disconnect, Port mismatch, Port duplicate, and Net Input(Q). The 'Warnings' list includes: Out-port disconnect, Symbol not found, Net Feedback(K), and Floating output. The 'Check circuit' dialog box also has buttons for 'Begin', 'Finish(Z)', '<<', '>>', and 'Close'. Below the 'Check circuit' dialog box is the 'Logic2SPICE V7.0' dialog box, which shows the context file path, library, cell, result file, and include file. It also has buttons for 'Run', 'View', 'Close', and 'Option...'. The status bar at the bottom shows 'For Help, press F1', 'x = -1242', 'y = 731', and 'NUM'.

MyCAD Schematic Editor V7.0 - [STOPWATCH/STOPWATCH_CORE/schematic]

File Edit View Add Shape Layout Tools Window Help

MyCAD Design Manager

- STOPWATCH
 - FULL_STOPWATCH
 - MOD10
 - MOD6
 - hierarchy
 - schematic
 - symbol
 - STOPWATCH_CORE
 - hierarchy
 - schematic
 - symbol
- Analog
- ETRI_GPIO_Cell
- ETRI_Std_ckt
- MyCell

Check circuit

Errors

- ☒ Check all errors
- ☒ Invalid net/bus name
- ☒ Net disconnect
- ☒ Bus has no name
- ☒ Invalid bus width
- ☒ Invalid net index
- ☒ Bus size mismatch
- ☒ Short
- ☒ Floating input
- ☒ In-port disconnect
- ☒ Port mismatch
- ☒ Port duplicate
- ☒ Net Input(Q)

Warnings

- ☐ Check all warnings
- ☐ Out-port disconnect
- ☐ Symbol not found
- ☐ Net Feedback(K)
- ☐ Floating output

Begin Finish(Z) << >> Close

Electrical Rule Checking...

0 Error(s), 0 Warning(s)

Logic2SPICE V7.0

Context File : C:\MyChipPro2026\Demo\Stopwatch\mde Browse...

Library : STOPWATCH

Cell : STOPWATCH_CORE

Result File : C:\MyChipPro2026\Demo\Stopwatch\ Result File Browse...

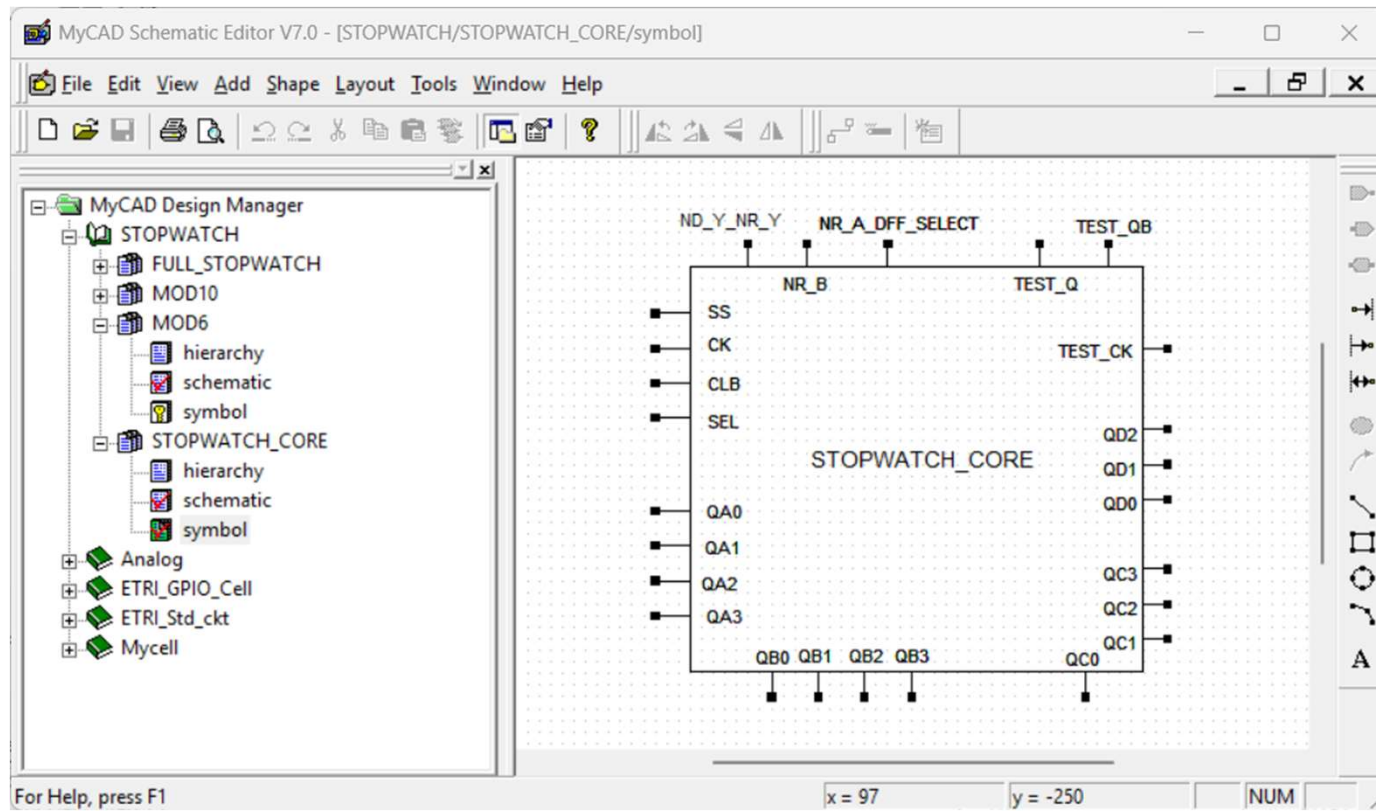
Include File : C:\MyChipPro2026\Data\Platform\ Include File Browse...

Run View Close Option...

For Help, press F1 x = -1242 y = 731 NUM

6. Design Stopwatch : Circuit (bottom-up)

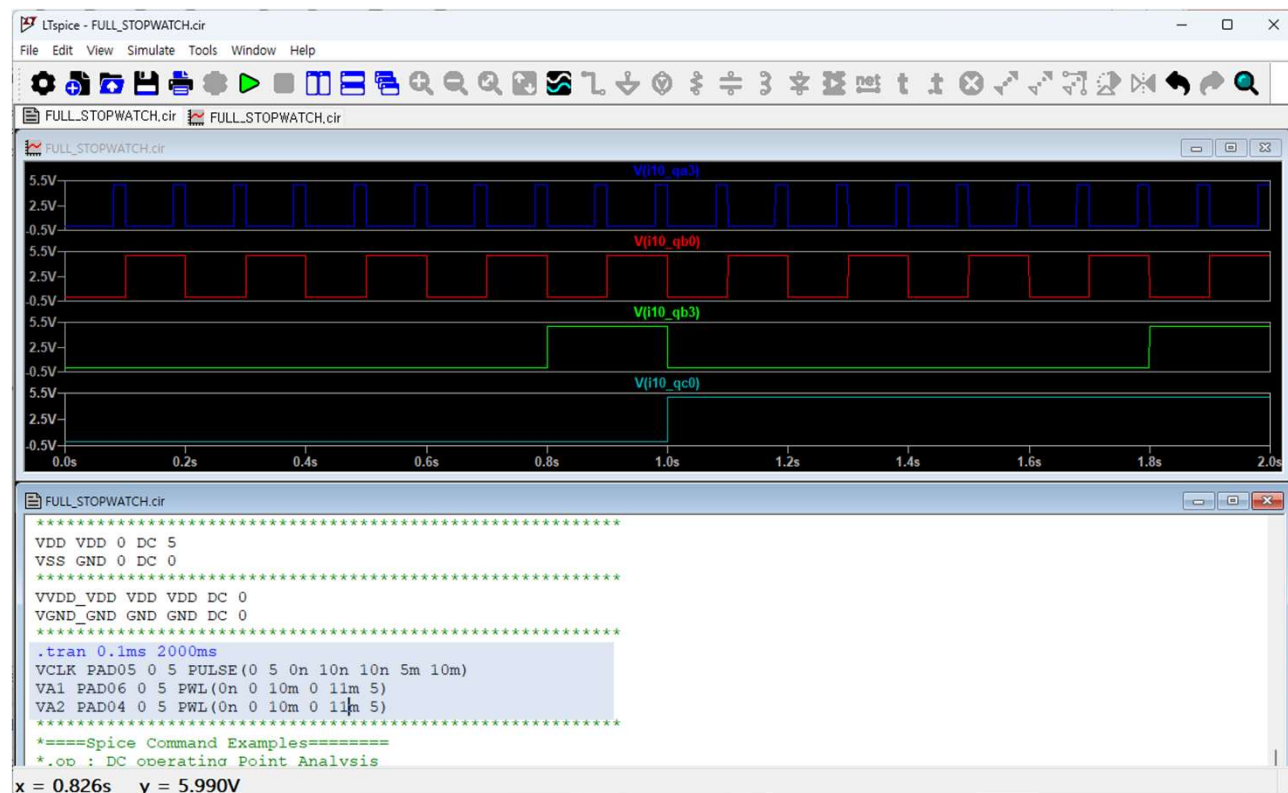
Make Symbol of Stopwatch_core



6. Design Stopwatch : Circuit (bottom-up)

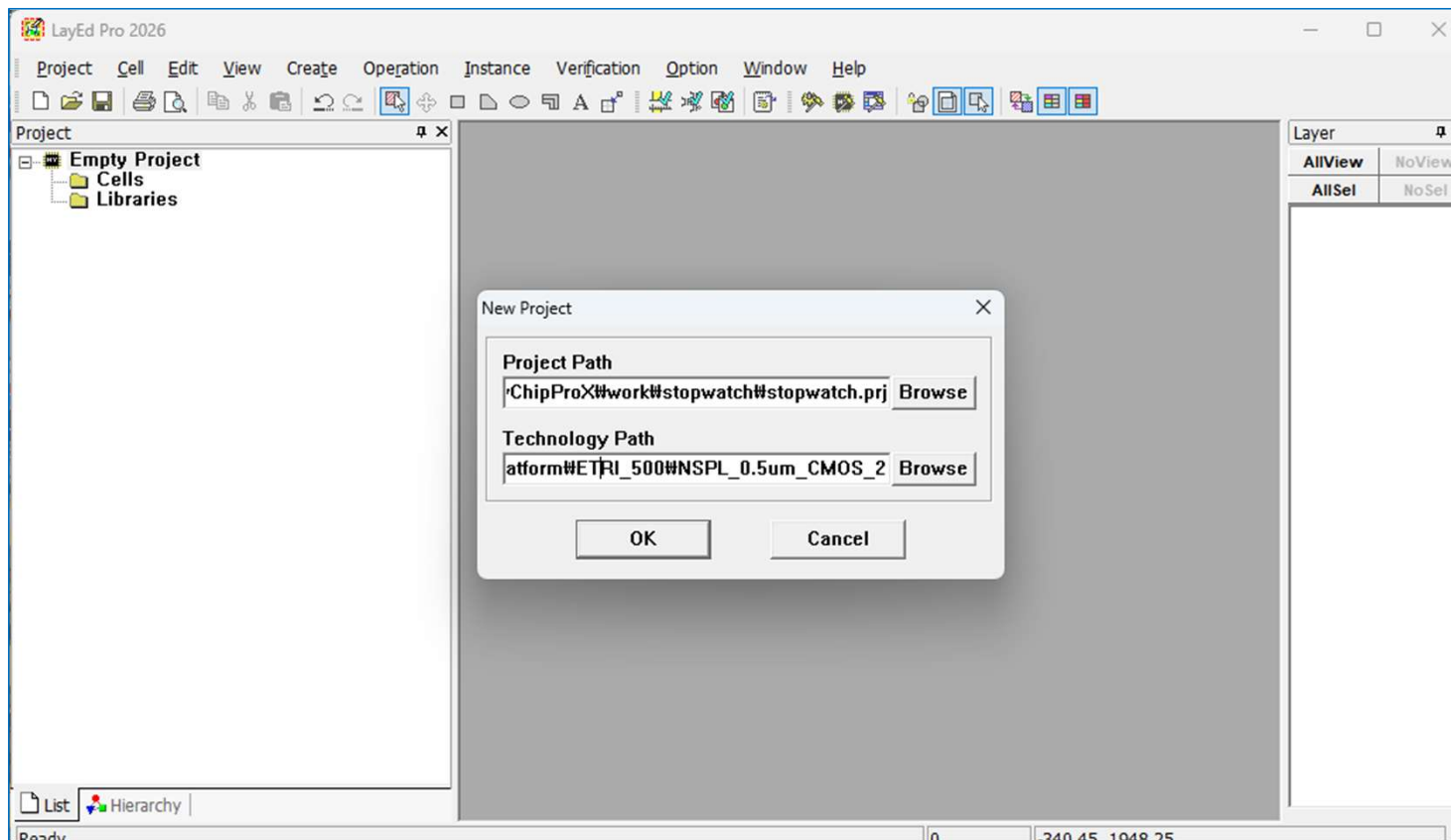
Spice Simulation using extracted spice netlist

- ☐ Simulation using netlist from Schematic (LTspice)
- ☐ SPICE Simulation
 - ☐ PAD05 -> CK
 - ☐ PAD06-> CLB
 - ☐ PAD11 -> QA3
 - ☒ PAD12 -> QB0
 - ☐ PAD15 -> QB3
 - ☐ PAD16 -> QC0
 - ☐ .99s -> 1s



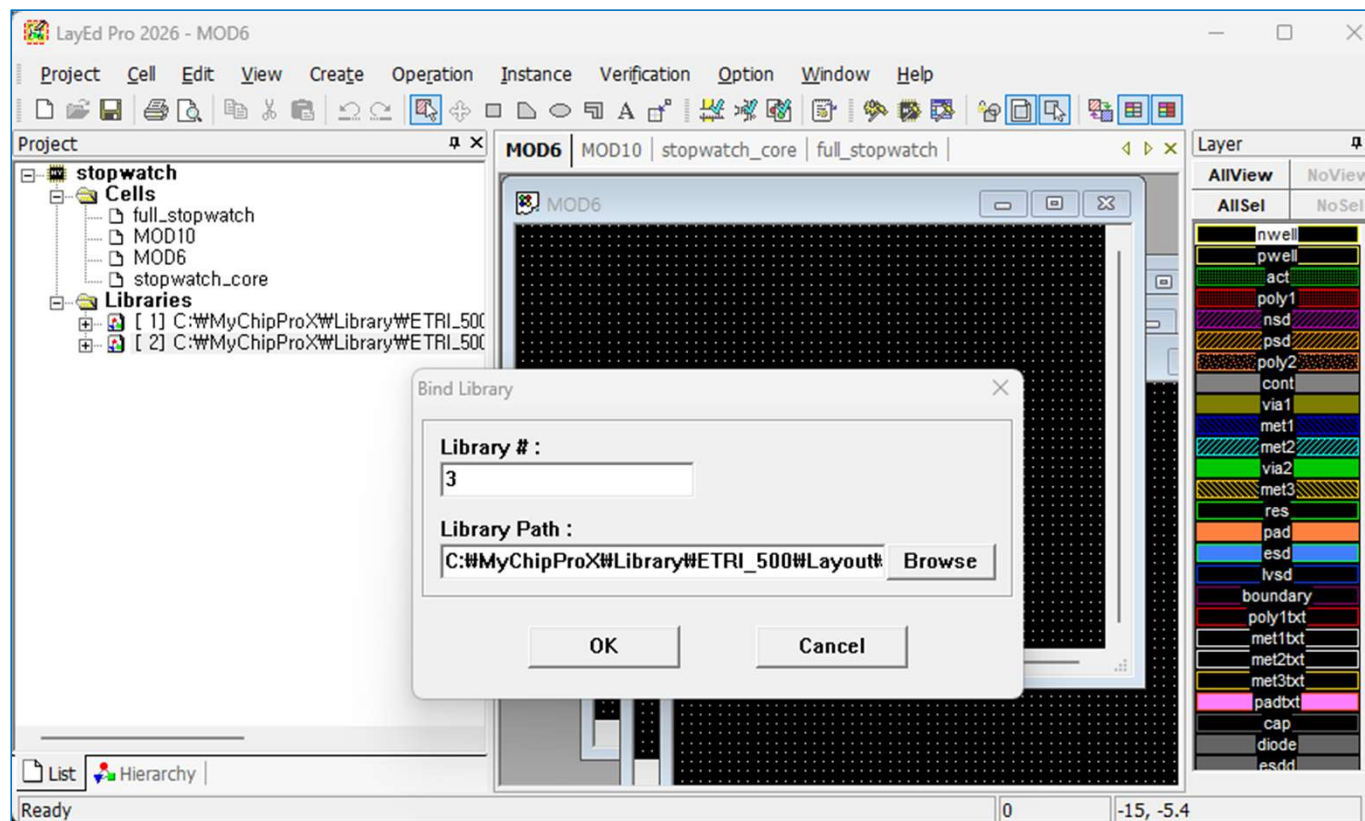
6. Design Stopwatch : Layout (bottom-up)

Stopwatch project generation using ETRI 500 platform – LayEdPro.exe



6. Design Stopwatch : Layout (bottom-up)

Create Cells and Binding Libraries



Project -> Bind Library...

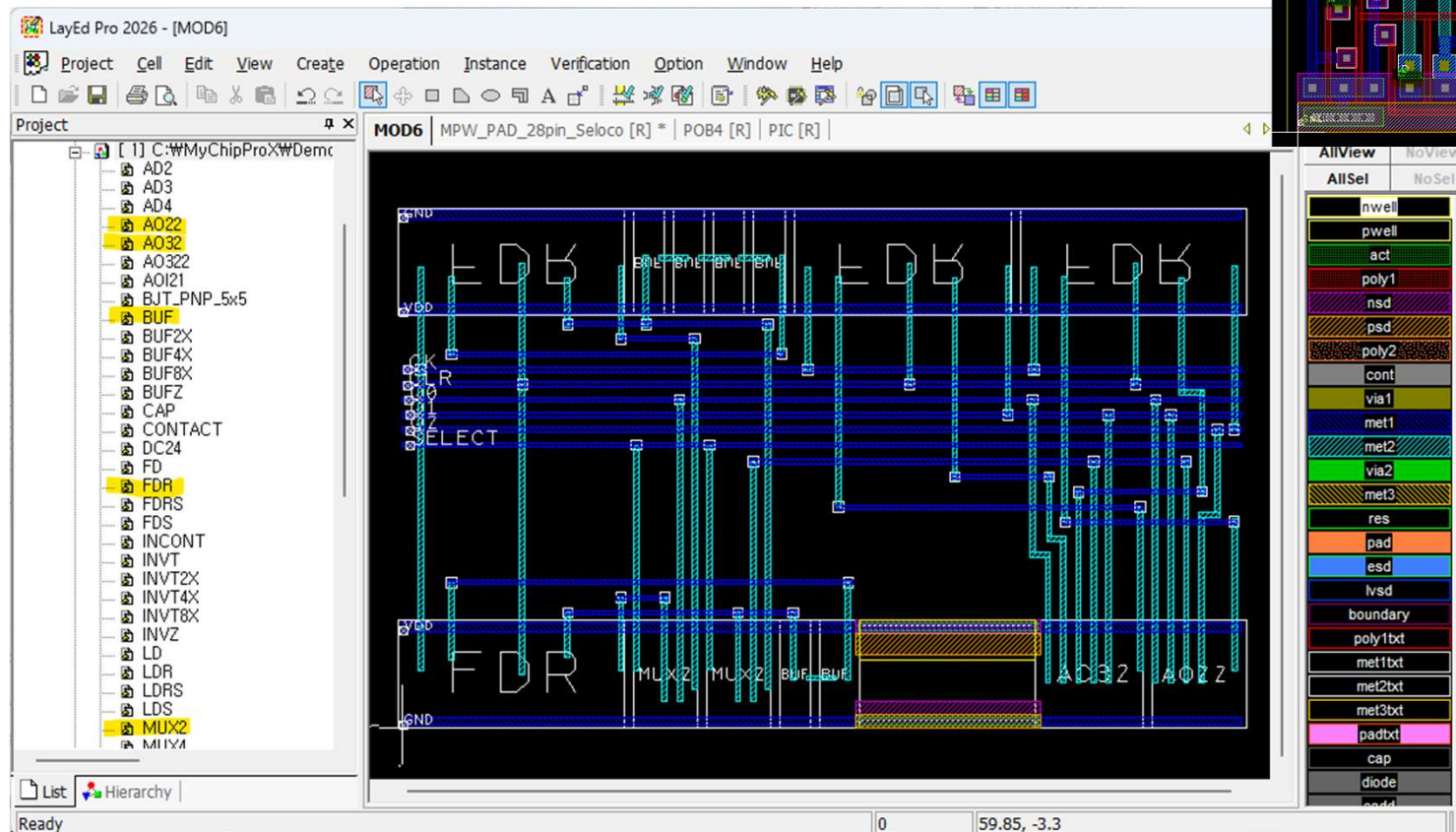
Library\ETRI_500\Layout

- EYTRI_IO
- MyCell
- MPW_PAD_28pin_Seloco

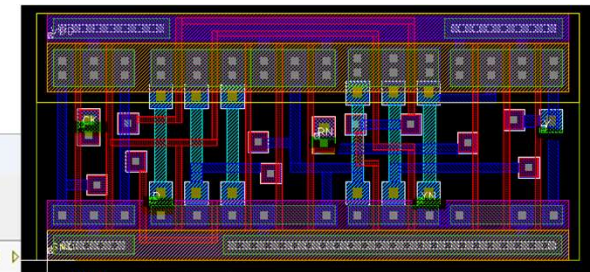
6. Design Stopwatch : Layout (bottom-up)

Design mod6 Layout on LayEdPro.exe

Place Cells and connect each cell based on Mod6 counter Circuit



FDR cell



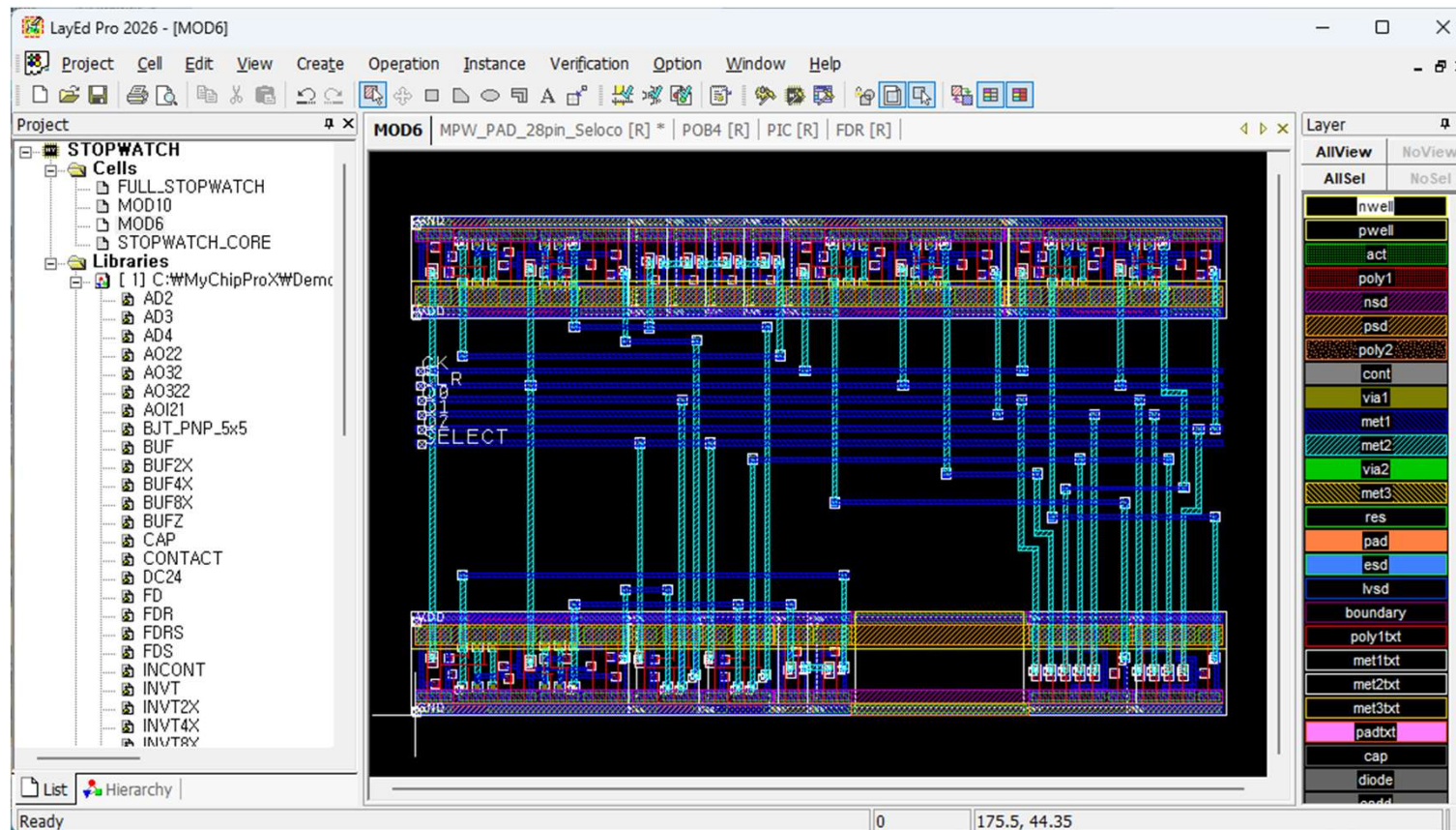
- 4 FDR cells
flip-flop with Reset
- 6 BUF cells
buffer
- 2 MUX2 cells
2-input mux
- 1 AO22 cell
OR (2-input AND) (2-input AND)
- 1 AO32 cell
OR (2-input AND) (3-input AND)

Connect each cell using metal/via
after placement

**Run DRC Frequently
During Layout Design**

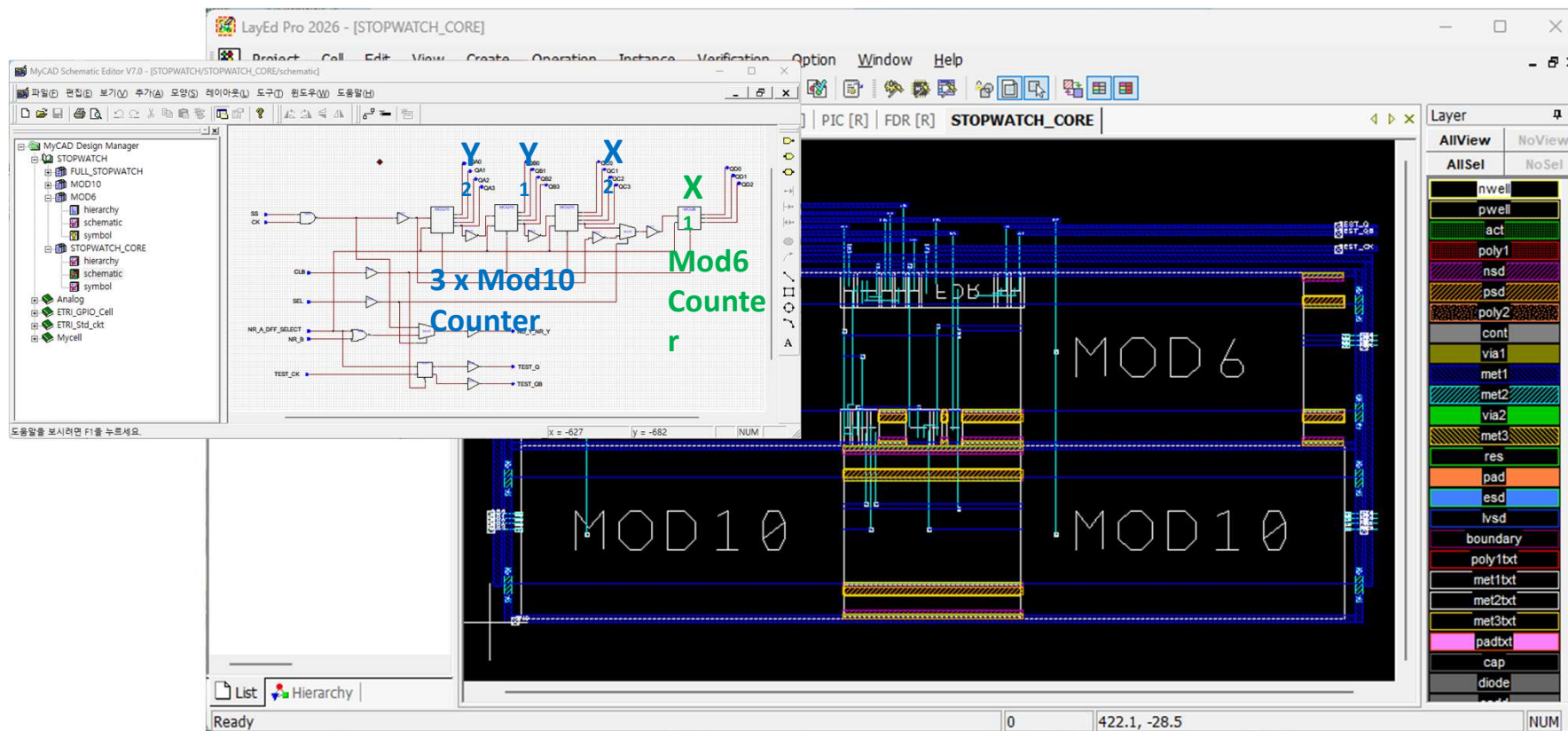
6. Design Stopwatch : Layout (bottom-up)

Designed Mod 6 layout - Flattened View



6. Design Stopwatch : Layout (bottom-up)

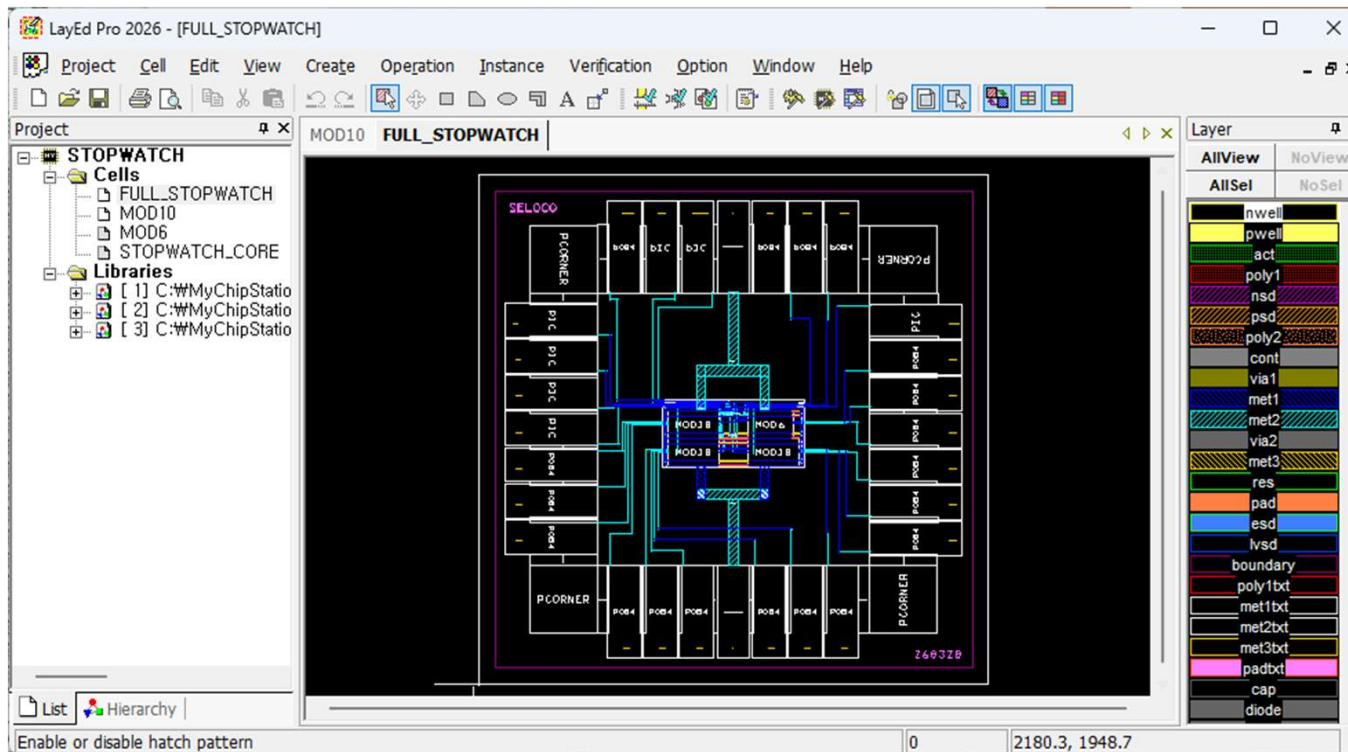
Design Stopwatch core using mod10 and mod6 based on stopwatch_core circuit



6. Design Stopwatch : Layout (bottom-up)

Stopwatch Completion – Connecting the Stopwatch Core to the I/O Pads for the ETRI MPW

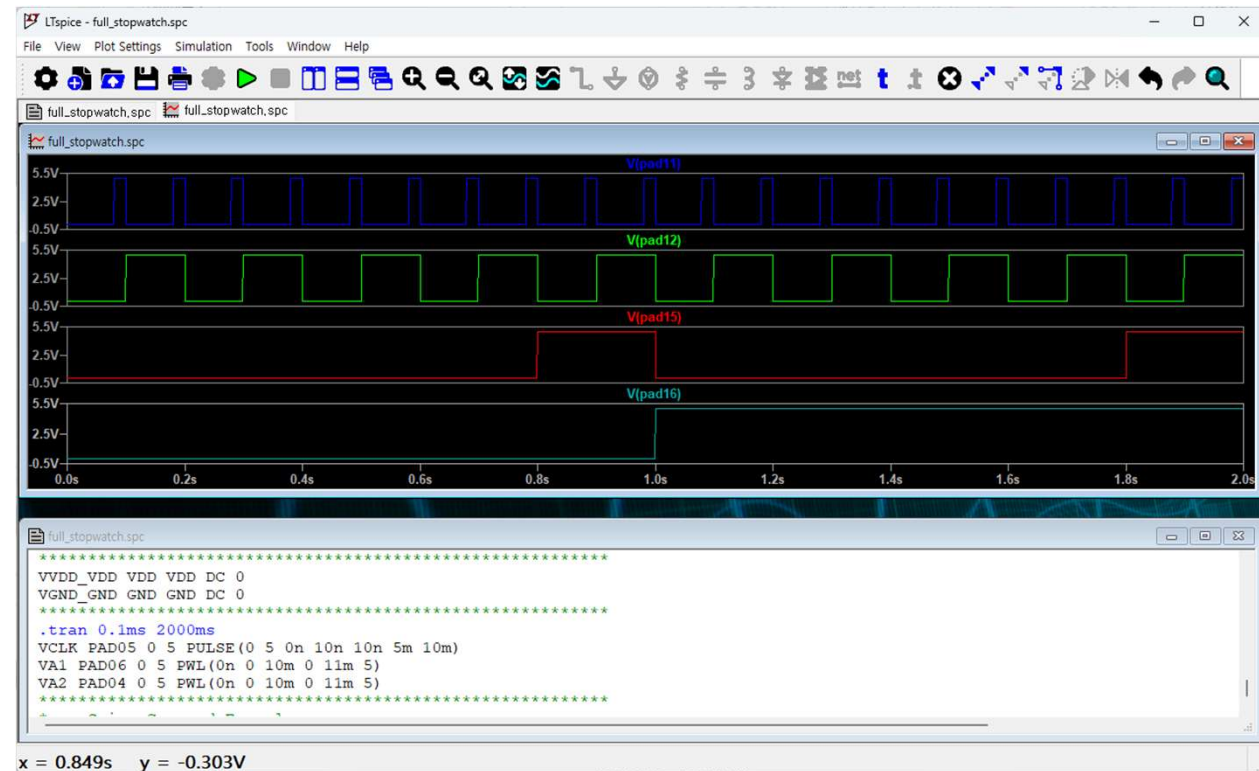
- Create a new cell named **Full_stopwatch**. Load the **28-pin pad frame** and **Stopwatch_core** from the library.
- Connect the I/Os of the **Stopwatch_core** to the I/Os of the **28-pin MPW pad frame**
- Replace each I/O pad with an **input pad** or **output pad** as needed.



6. Design Stopwatch : Layout (bottom-up)

Check the result of Layout Spice Simulation (EXTRACT.1 file under design folder, extracted by MyNetPro)

- ☐ Simulation using netlist extracted from layout
- ☐ Spice Simulation (LayEdPro->MyNetPro)
 - ☐ PAD05 -> CK
 - ☐ PAD06-> CLB
 - ☐ PAD11 -> QA3
 - ☐ PAD12 -> QB0
 - ☐ PAD15 -> QB3
 - ☐ PAD16 -> QC0
 - ☐ .99s -> 1s

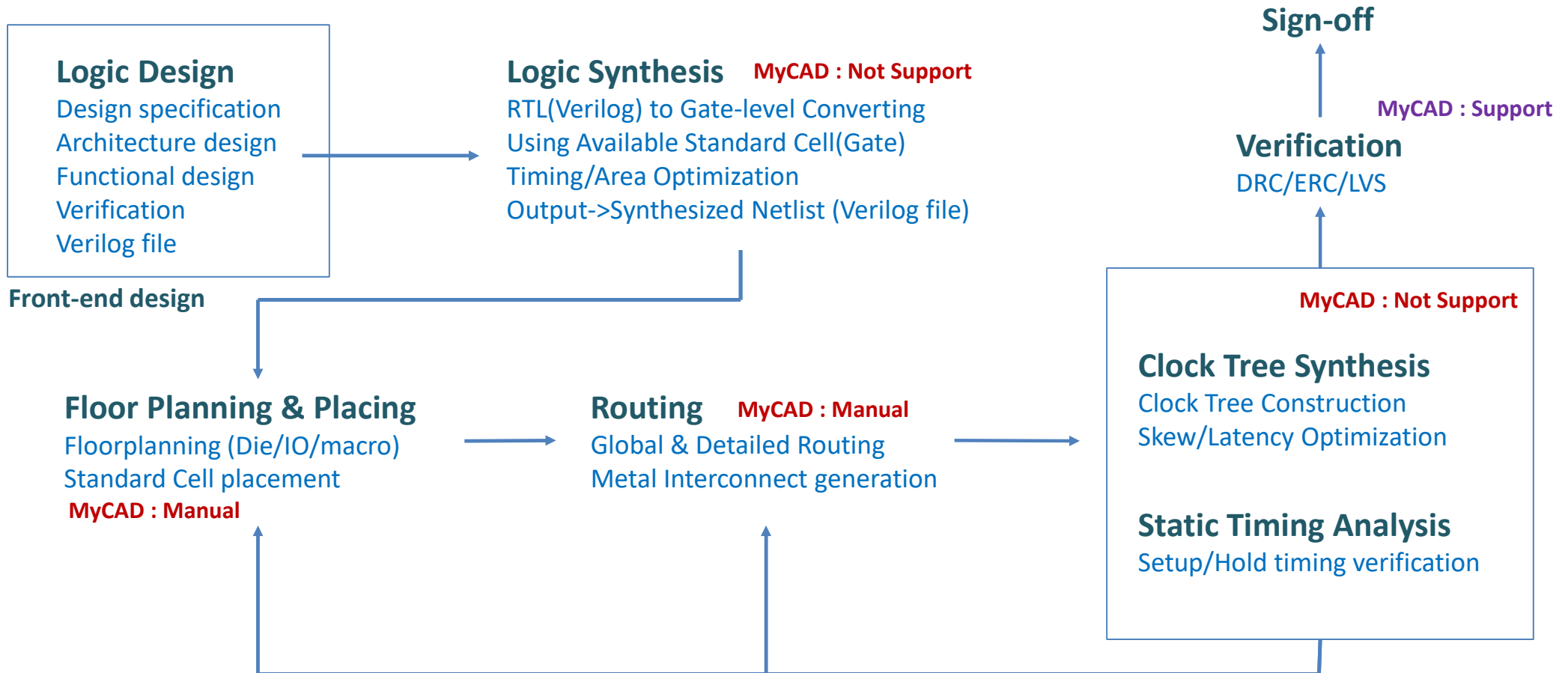


6. Design Stopwatch : Layout (bottom-up)

Manual (bottom-up) Layout Design Flow Summary

- 1. Select the Target Fabrication Platform** : Choose the semiconductor process (platform) for chip fabrication.
- 2. Verify the Available Standard Cell Library** : Confirm the available standard cells (layout and circuit).
- 3. Design and Verify the Chip Functionality**
 1. Implement the chip using standard cells (**modular design**).
 2. Verify the circuit schematic.
- 4. Create a New Layout Project** : Create a project in the Layout Editor using the selected platform.
- 5. Create Blank Cells and Bind the Library**
 1. Create a blank cell for each module.
 2. Bind the platform-specific library.
- 6. Design the Layout of Each Module**
 1. Place the required standard cells.
 2. Connect them using metal layers and vias.
- 7. Perform DRC During Layout Design** : Run DRC frequently to detect design rule violations.
- 8. Design the logic core using designed modules**
- 9. Verify the Logic Core** : Perform **DRC, ERC, and LVS** after completing the logic core.
- 10. Integrate the I/O Pads** : Connect the **ETRI MPW I/O pad frame** to the **Stopwatch core**.
- 11. Final Verification** : Perform the final **DRC, ERC, and LVS** before tape-out.

7. Top-down Chip Design Process



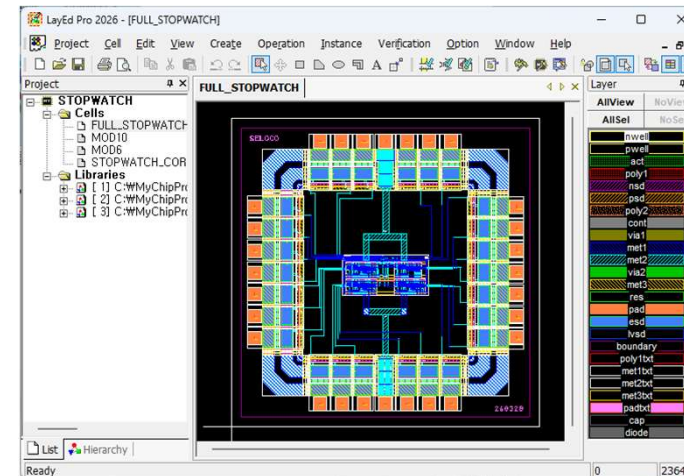
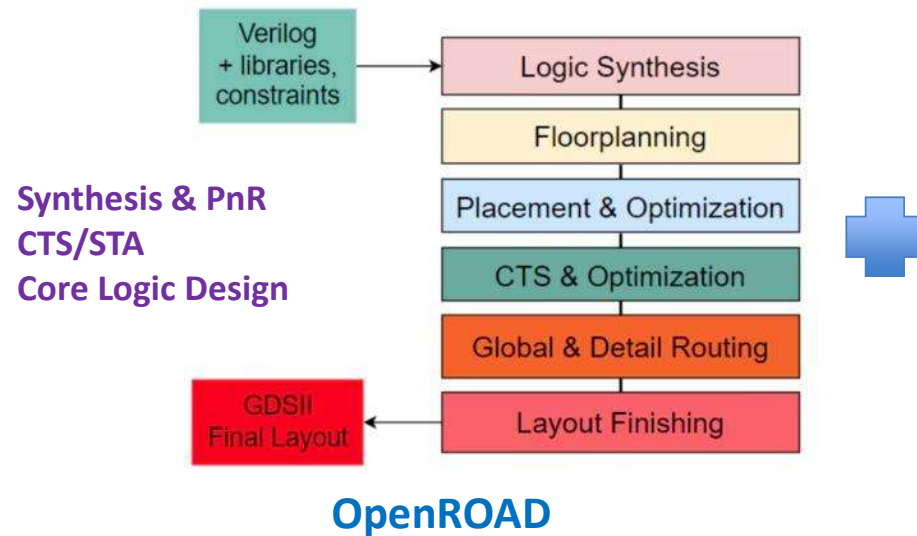
*Major EDA tool은 Logic Synthesis 부터 전 과정을 제공

Back-end design

7. Top-down Chip Design Process – OpenROAD & MyChipStationX

OpenROAD + MyCAD: Best Combination

- **OpenROAD**: Open-source RTL-to-GDS ASIC design automation toolchain
 - **Focus**: Physical design / backend design
 - **Limitations**: Linux-based, script-driven, limited GUI layout editing
- **MyChipStationX**: Windows-based layout editing, verification, and visualization
- **Integration**: Seamless connection between Linux-based OpenROAD and Windows-based MyChipStationX
- **Result**: Automated digital design flow with powerful layout editing capability



MyChipStationX

Layout Editing
Verification
Add IO PAD
Mixed Signal Design

The diagram illustrates the IC design flow, highlighting the integration of OpenROAD and MyChipStationX. The flow is as follows:

- Front-end design** (Logic Design):
 - Design specification
 - Architecture design
 - Functional design
 - Verification
 - Verilog file
- Logic Synthesis** (OpenROAD):
 - RTL(Verilog) to Gate-level Converting
 - Using Available Standard Cell(Gate)
 - Timing/Area Optimization
 - Output->Synthesized Netlist (Verilog file)
- Floor Planning & Placing** (OpenROAD):
 - Floorplanning (Die/IO/macro)
 - Standard Cell placement
- Routing** (OpenROAD):
 - Global & Detailed Routing
 - Metal Interconnect generation
- Verification** (MyChipStationX):
 - DRC/ERC/LVS
 - 수정/IO 추가
 - Integrate Multiple Modules
- Clock Tree Synthesis** (OpenROAD):
 - Clock Tree Construction
 - Skew/Latency Optimization
- Static Timing Analysis** (OpenROAD):
 - Setup/Hold timing verification
- Sign-off**

Arrows indicate the flow between these stages. OpenROAD is used for Logic Synthesis, Floor Planning & Placing, Routing, Clock Tree Synthesis, and Static Timing Analysis. MyChipStationX is used for Verification. The flow starts with Front-end design, moves to Logic Synthesis, then to Floor Planning & Placing, Routing, and Verification. Clock Tree Synthesis and Static Timing Analysis are also part of the flow, leading to the final Sign-off.

7. Top-down Chip Design Process – Setup OpenROAD Environment

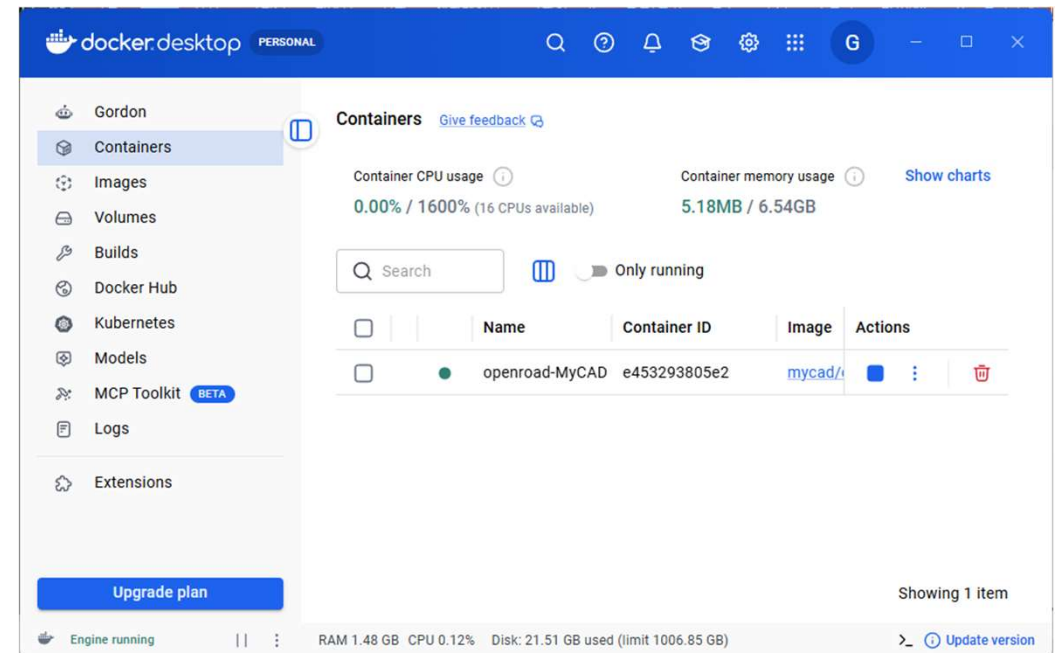
1. Install WSL (Ubuntu) from Microsoft Store
2. Install Docker Desktop from Microsoft Store
(Please check Hyper-V is enabled on BIOS)
3. Make an account for Docker and log-in
4. Run Docker Desktop on Windows
5. Make a Docker container as below to share
C:\MyChipStationX\openroad and **/flow/MyCAD**
on OpenROAD

on Windows Power Shell

```
PS C:\Users\User> docker run -it --name openroad-MyCAD -v C:\MyChipStationX\openroad:/OpenROAD-flow-scripts/flow/MyCAD mycad/openroad:v1 bash
```

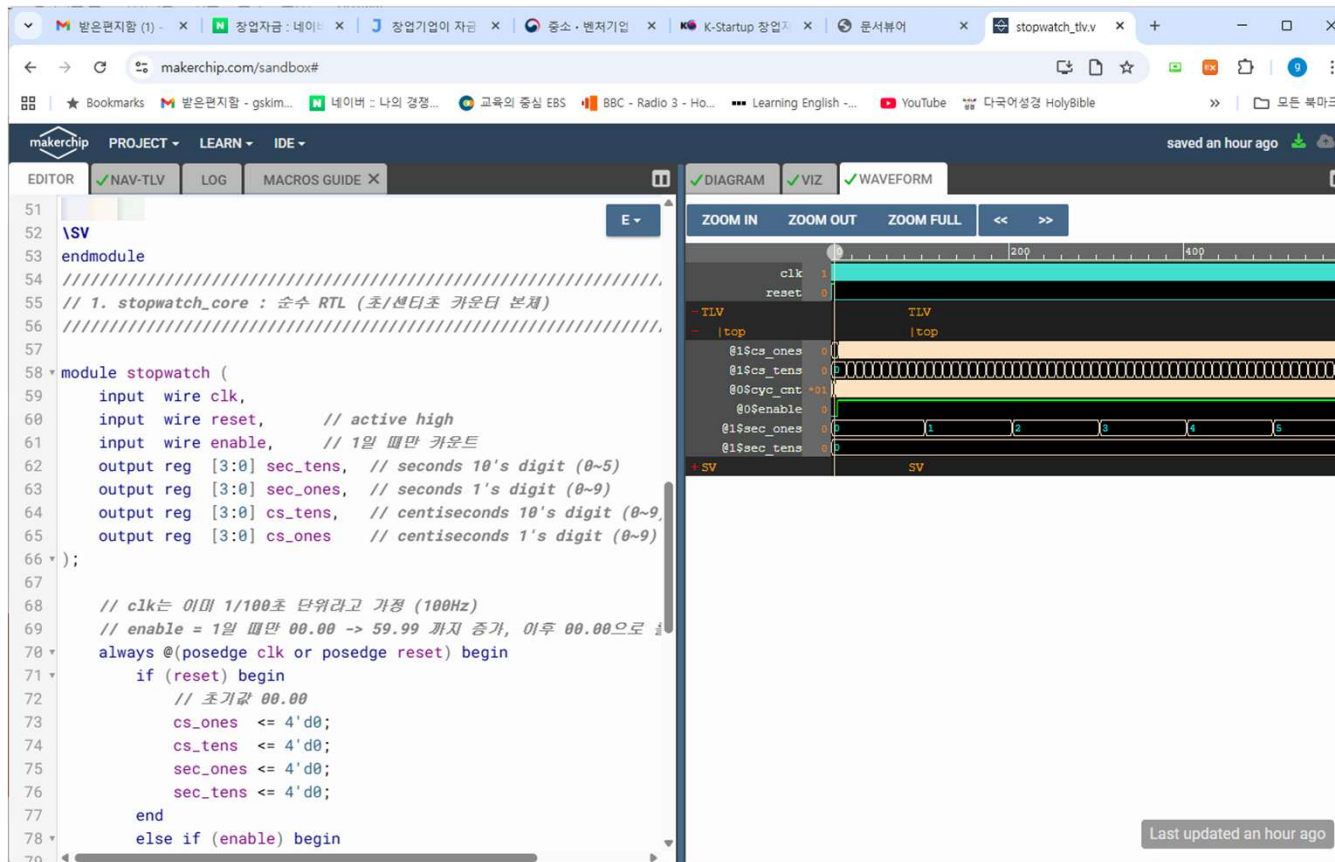
Now You Are ready to use MyOpenROAD.exe to use OpenROAD for synthesis and PnR

You need to Run Docker Desktop and openroad-MyCAD must be active



8. Stopwatch Design (Front-end) - Stopwatch logic design(verilog)

Verilog design and Simulation Example(Makerchip solution & ChatGPT)



The screenshot shows the Makerchip website interface. On the left is the Verilog code editor, and on the right is the simulation waveform viewer.

```
51 \SV
52 endmodule
53
54 //////////////////////////////////////////////////
55 // 1. stopwatch_core : 순수 RTL (초/센티초 카운터 본체)
56 //////////////////////////////////////////////////
57
58 module stopwatch (
59     input wire clk,
60     input wire reset, // active high
61     input wire enable, // 1일 때만 카운트
62     output reg [3:0] sec_tens, // seconds 10's digit (0~5)
63     output reg [3:0] sec_ones, // seconds 1's digit (0~9)
64     output reg [3:0] cs_tens, // centiseconds 10's digit (0~9)
65     output reg [3:0] cs_ones, // centiseconds 1's digit (0~9)
66 );
67
68 // clk는 이미 1/100초 단위라고 가정 (100Hz)
69 // enable = 1일 때만 00.00 -> 59.99 까지 증가, 이후 00.00으로 리셋
70 always @(posedge clk or posedge reset) begin
71     if (reset) begin
72         // 초기값 00.00
73         cs_ones <= 4'd0;
74         cs_tens <= 4'd0;
75         sec_ones <= 4'd0;
76         sec_tens <= 4'd0;
77     end
78     else if (enable) begin
79
```

The waveform viewer on the right shows the simulation results for the stopwatch. It displays signals for `clk`, `reset`, `enable`, and the BCD outputs `sec_tens`, `sec_ones`, `cs_tens`, and `cs_ones`. The `enable` signal is shown as a pulse, and the BCD outputs show the corresponding time values during the pulse.

ChatGPT Input and Makerchip Simulation

•Enter the design requirement into **ChatGPT**:

- Create a stopwatch in Verilog with:

- **100 Hz clock input**
- **Enable signal** for start/pause control
- **Reset signal**
- **BCD outputs** for:
 - Two-digit centiseconds
 - Two-digit seconds

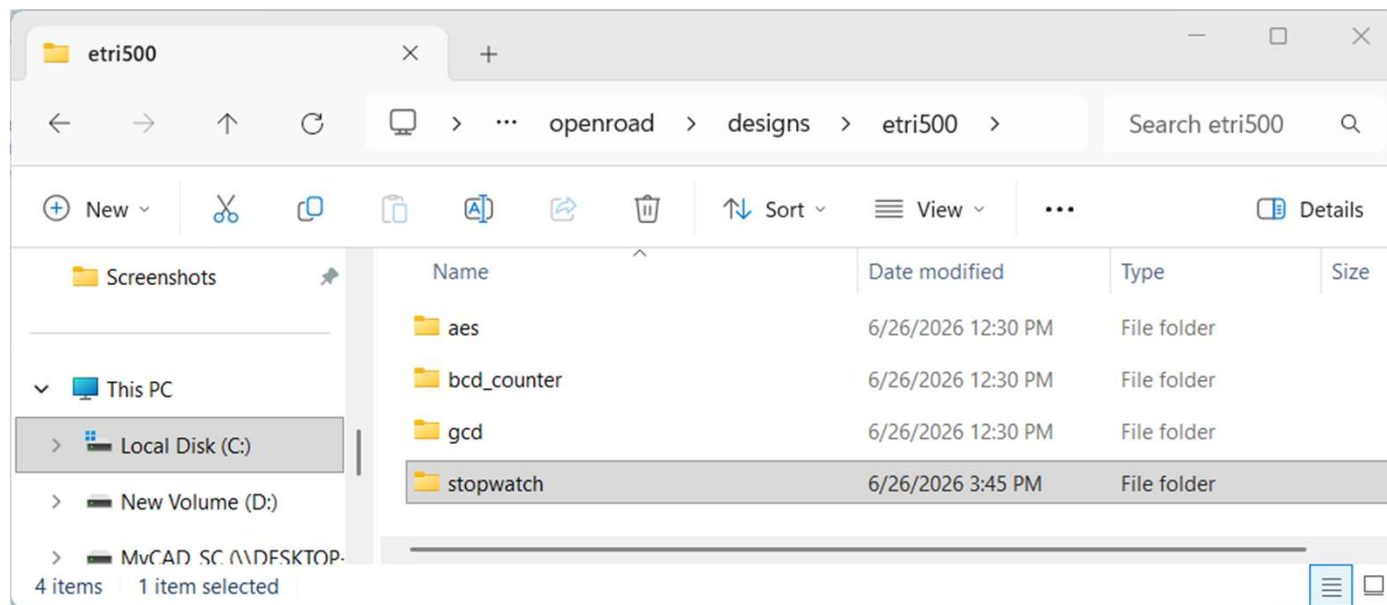
•Ask ChatGPT to generate a **Verilog-TLV version** by adding a **TLV header** to the Verilog code.

•Load the generated **verilog.tlv** file on www.makerchip.com.

•Run the simulation on the **Makerchip** website and verify the stopwatch operation.

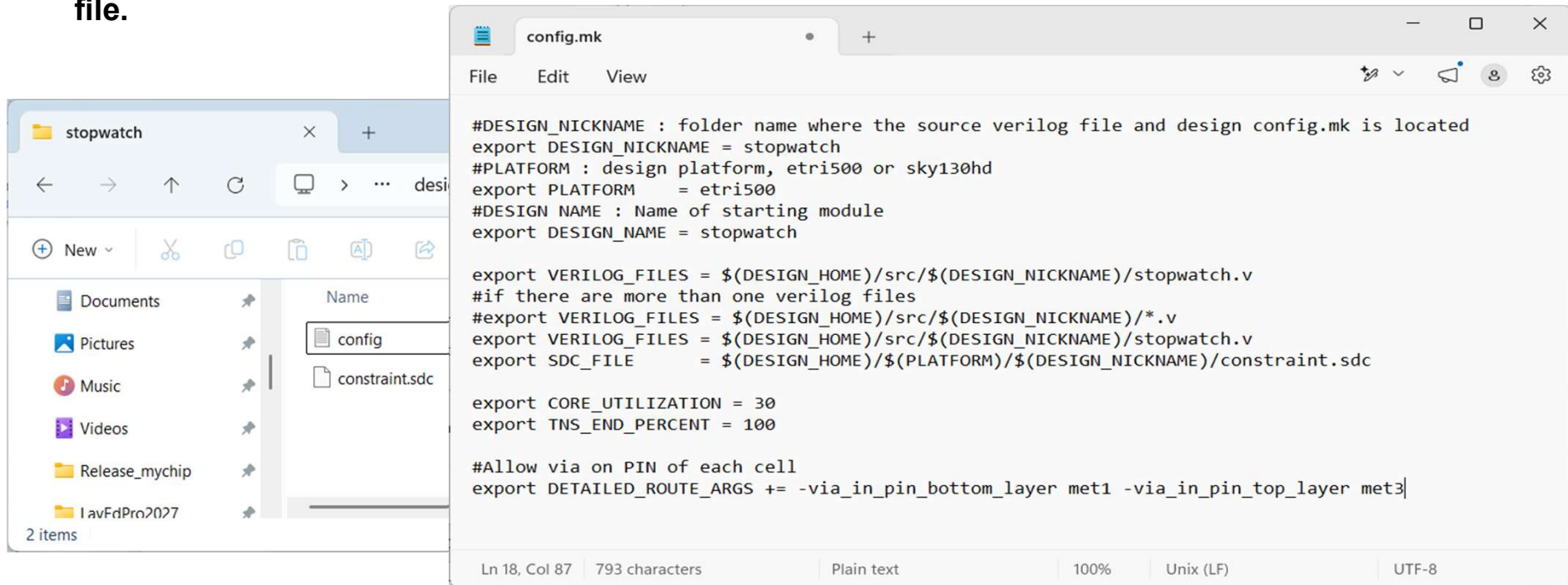
8. Stopwatch Design (Top-down) – Configuring Design Source file

1. Make design source folder(stopwatch) under C:\MyChipStationX\openroad\designs\src
2. Copy verilog source file(stopwatch.v) into this new folder.



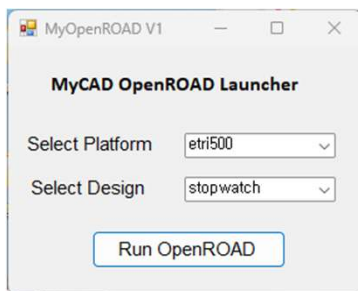
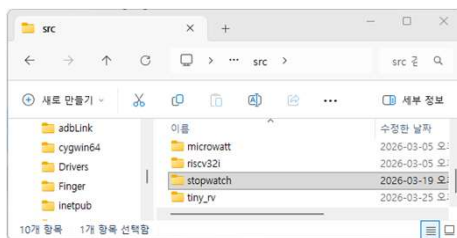
8. Stopwatch Design (Top-down) – Configuring Design Source file

1. Create ne folder(stopwatch) under C:\MyChipStationX\openroad\designs\etri500 (you may copy one of existing folder and rename it).
2. Create or edit config.mk file as below. And change the design name to stopwatch in constraint.sdc file.



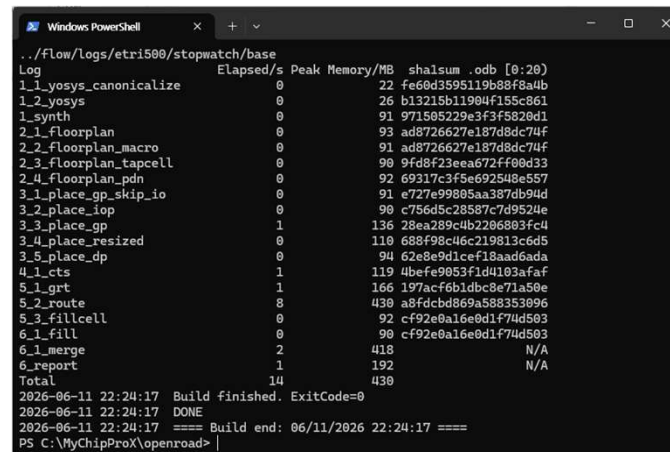
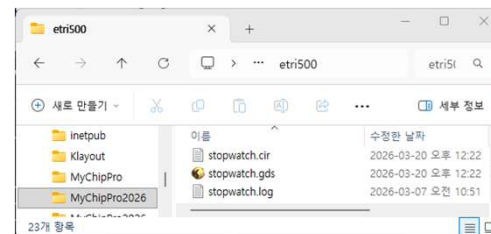
8. Stopwatch Design (Top-down) – Design flow

1. Design (Verilog) & Scripts (On Windows)

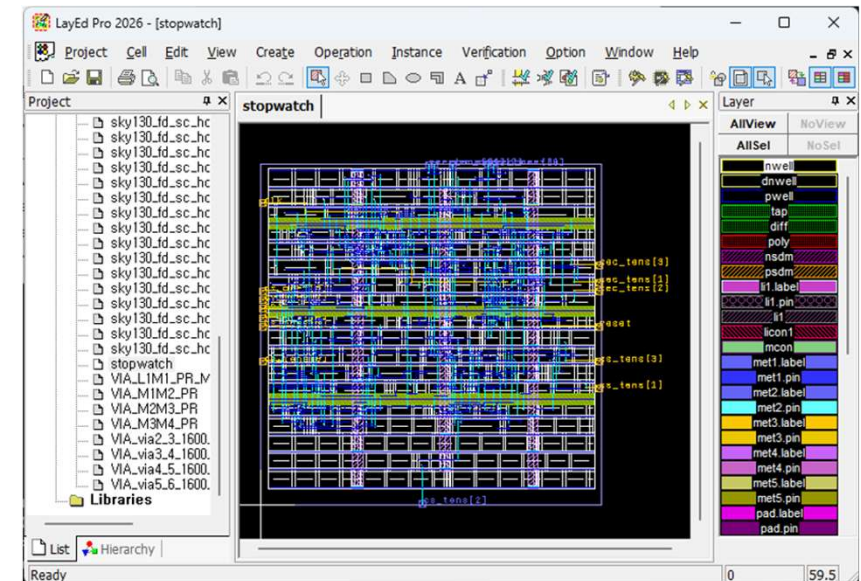


2. Execute OpenROAD

4. Final results(GDSII, netlist..) on Windows



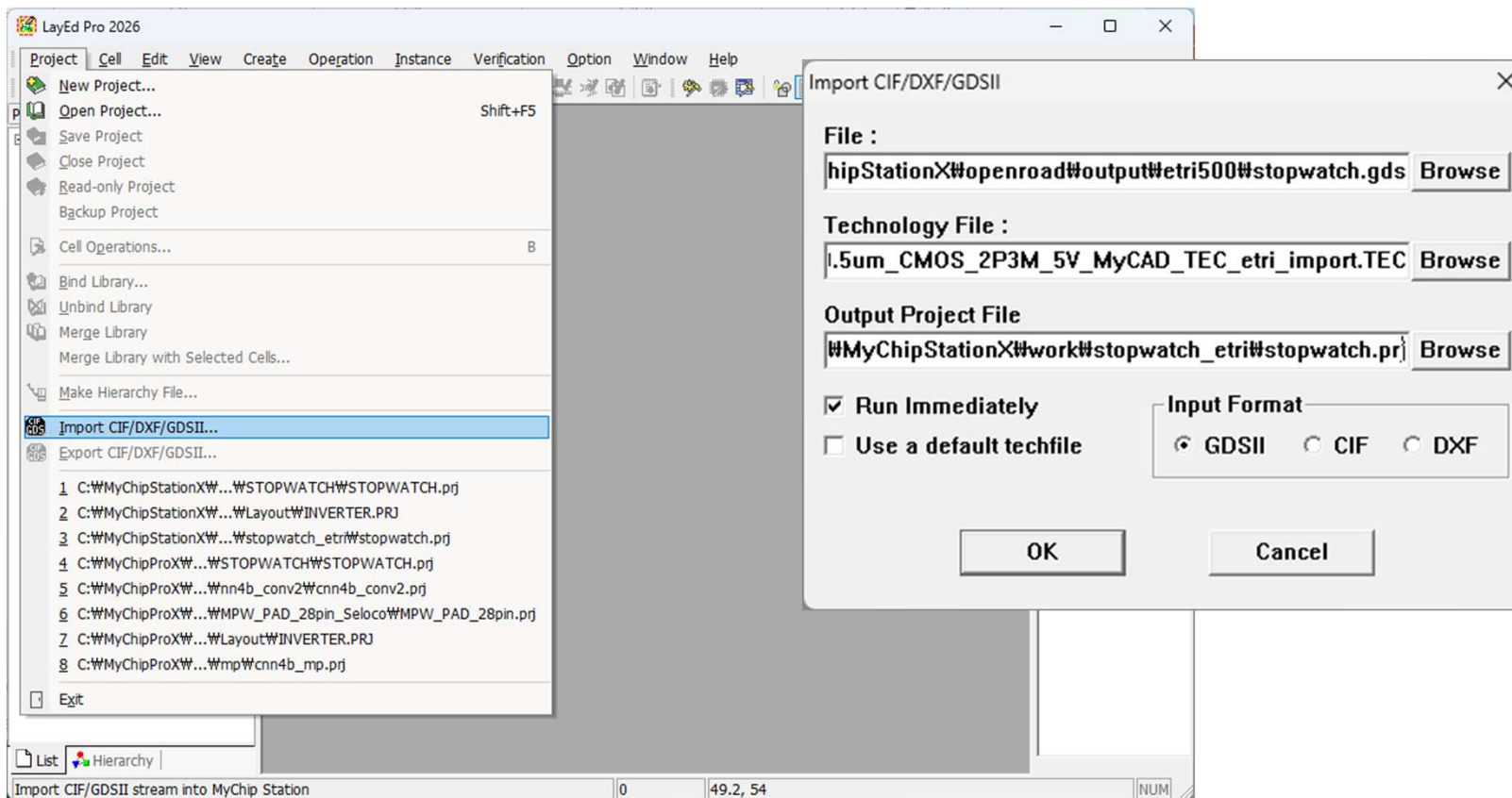
3. OpenROAD Result (Synthesis & PnR)



5. Editing & Verification on MyChipStationX

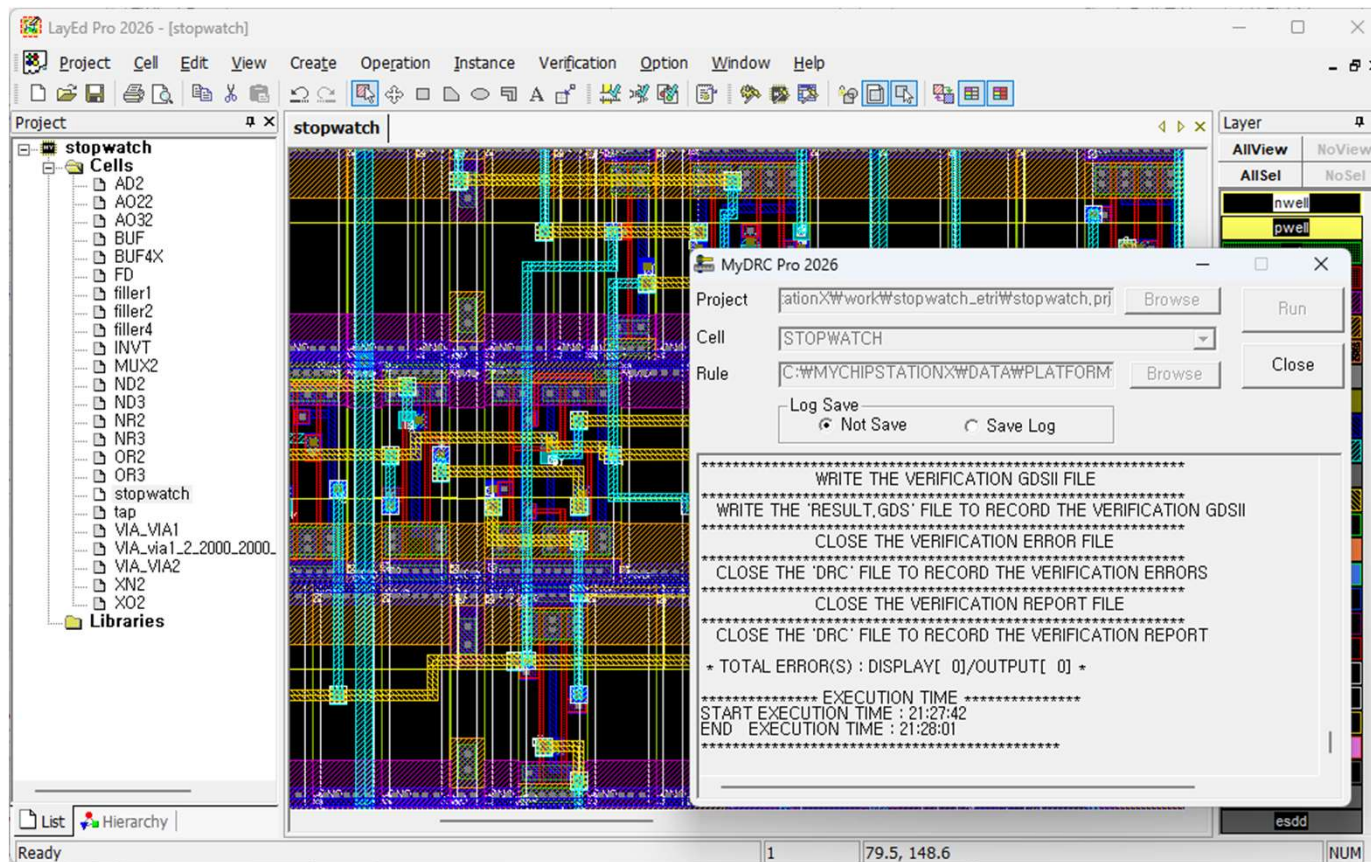
8. Stopwatch Design (Top-down) – Import Result into MyChipStationX

Import result of OpenROAD(at openroad\output\etri500\) into MyChipStationX



8. Stopwatch Design (Top-down) – verification of layout(DRC/ERC)

DRC/ERC the Synthesized/PnR by OpenROAD on ETRI MPW platform, Fix Errors



***Warning**
If there is DRC error,
you need to fix it manually.

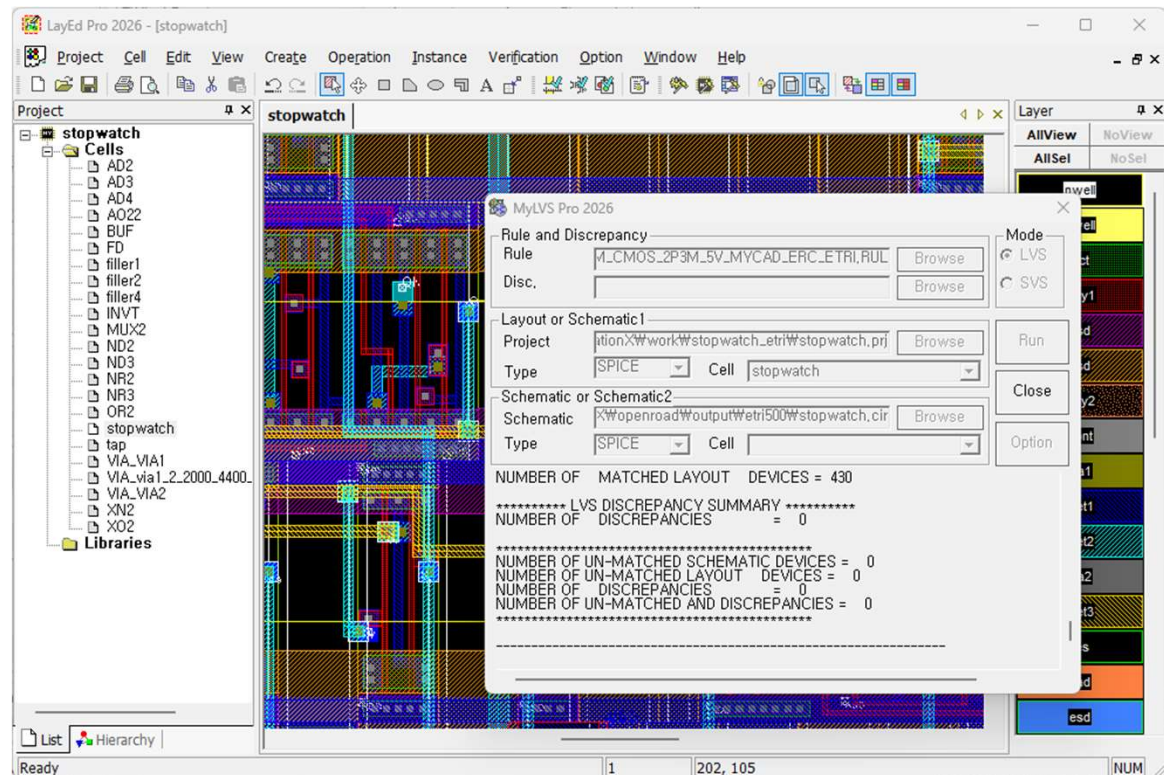
Before checking ERC,
you need to add VDD and
GND label correctly
to avoid 'no power connected'
error.

8. Stopwatch Design (Top-down) – verification of layout(LVS check)

(CDL → Verilog → Flatten → MOS Extract → Net Rename → Reduce → Compare → LVS Clean)

```
*****  
*** REDUCE (LAYOUT) SUMMARY REPORT  
*****  
----- TOTAL LAYOUT DEVICES -----  
MOS : 852  
----- REDUCED LAYOUT DEVICES -----  
MOS : 176  
INV : 191  
NAND : 35  
NOR : 25  
AOI2 : 3  
  
*****  
*** REDUCE (SCHEMATIC) SUMMARY REPORT  
*****  
----- TOTAL SCHEMATIC DEVICES -----  
MOS : 852  
----- REDUCED SCHEMATIC DEVICES -----  
MOS : 176  
INV : 191  
NAND : 35  
NOR : 25  
AOI2 : 3
```

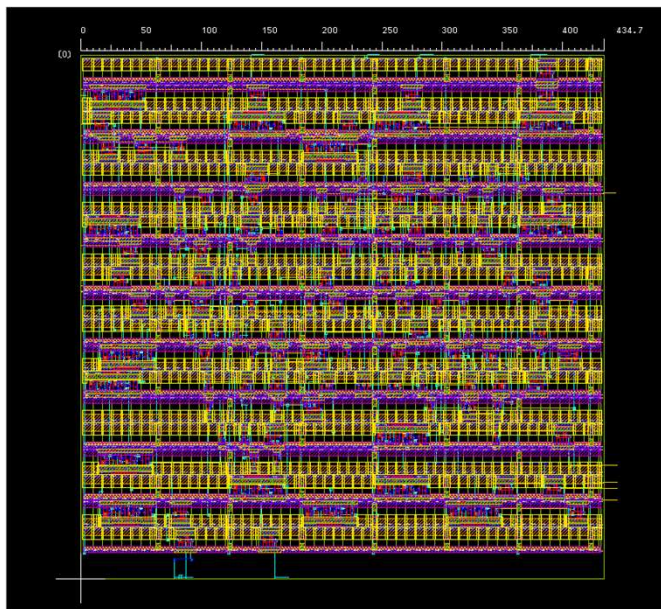
100% logic matches



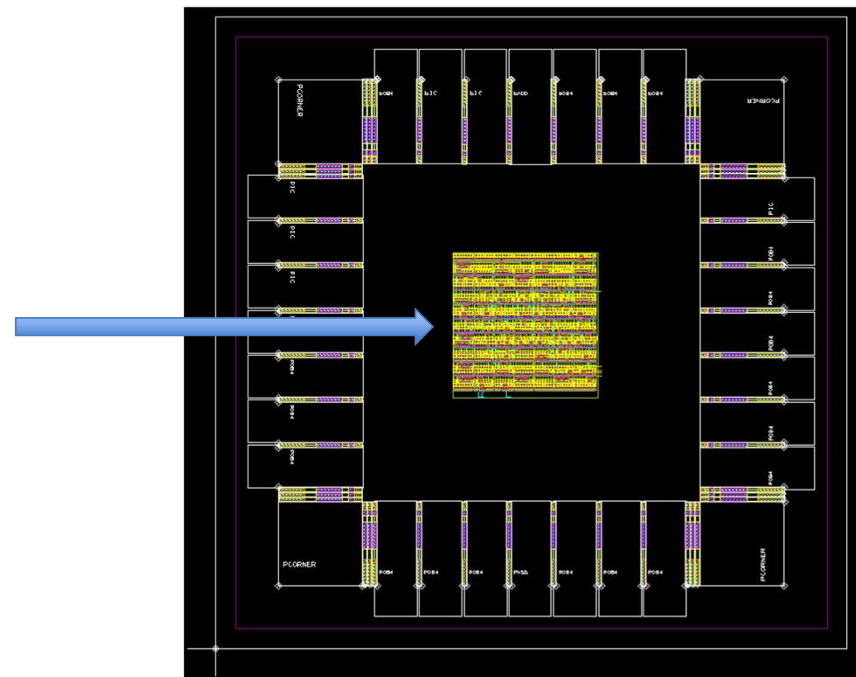
No Mismatches in LVS

8. Stopwatch Design (Top-down) – MPW chip layout on MyChipStationX

Verilog → OpenROAD(Yosis, Openroad, klayout) → Synthesis/Placement/Routing → Import on MyChipStationX -> MyChipStationX(IO/PAD + Stopwatch_core) -> MyChipStationX(DRC, ERC, LVS) -> Tapeout



Synthesized stopwatch core in OpenROAD
Edited and Verified on MyChipStationX



Merged with IO/PAD in MyChipStationX

Thank You!