Project 2 – Cell-Based Layout Design

11110CS312000

Introduction of Integrated Circuit Design, NTHU

Outline

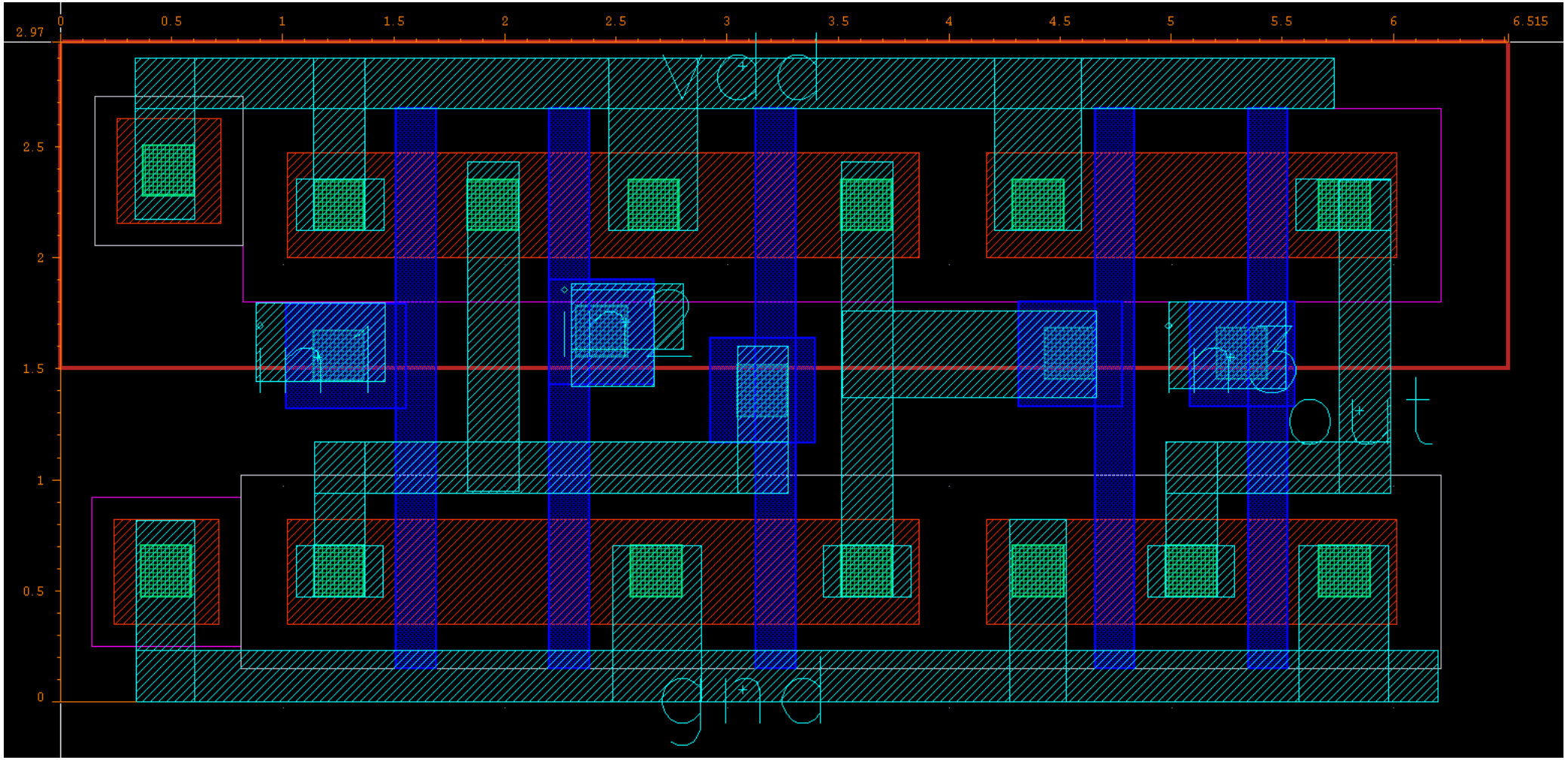
- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

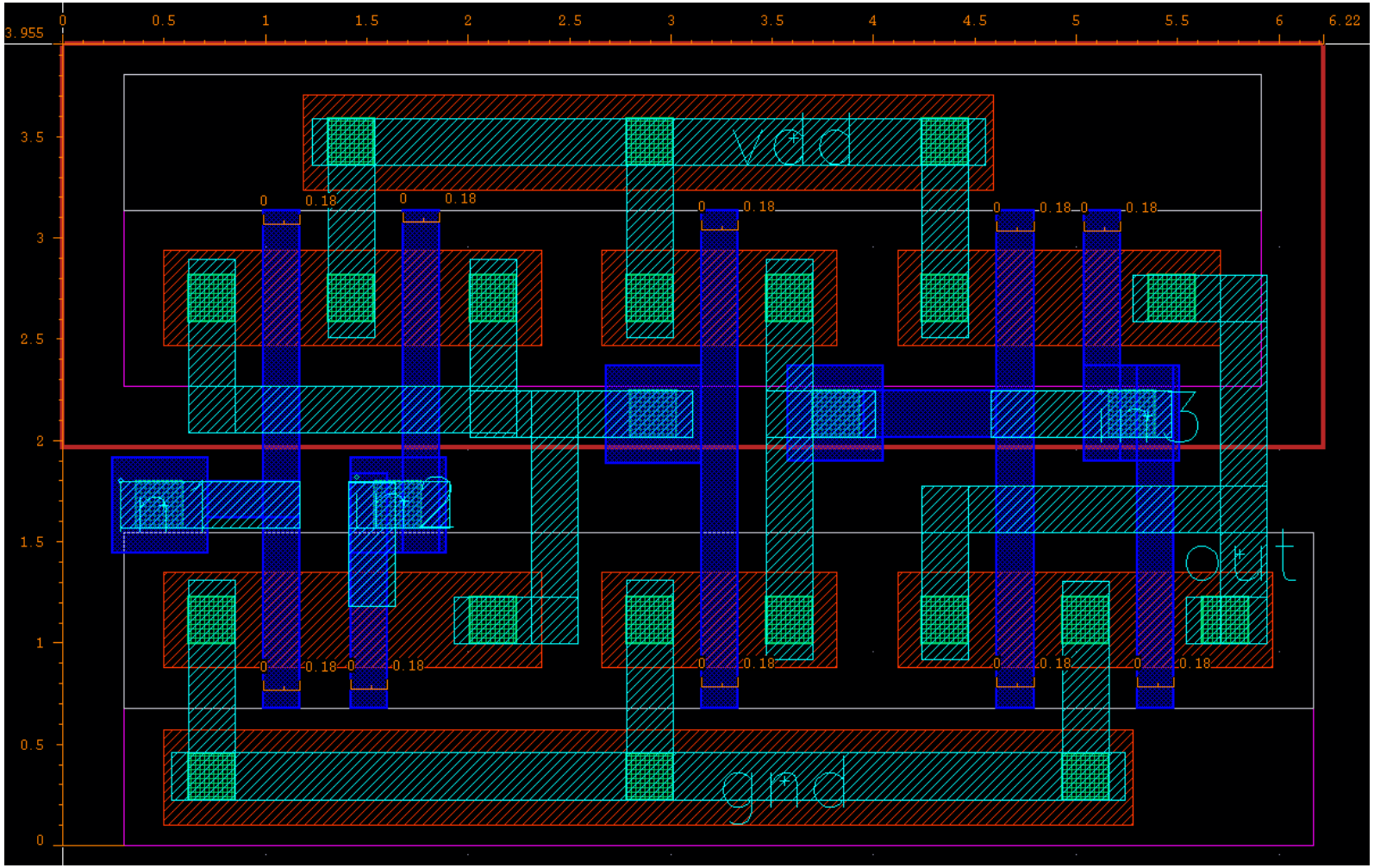
Project 1 - Reviewed

Top 3 Layout - 1st



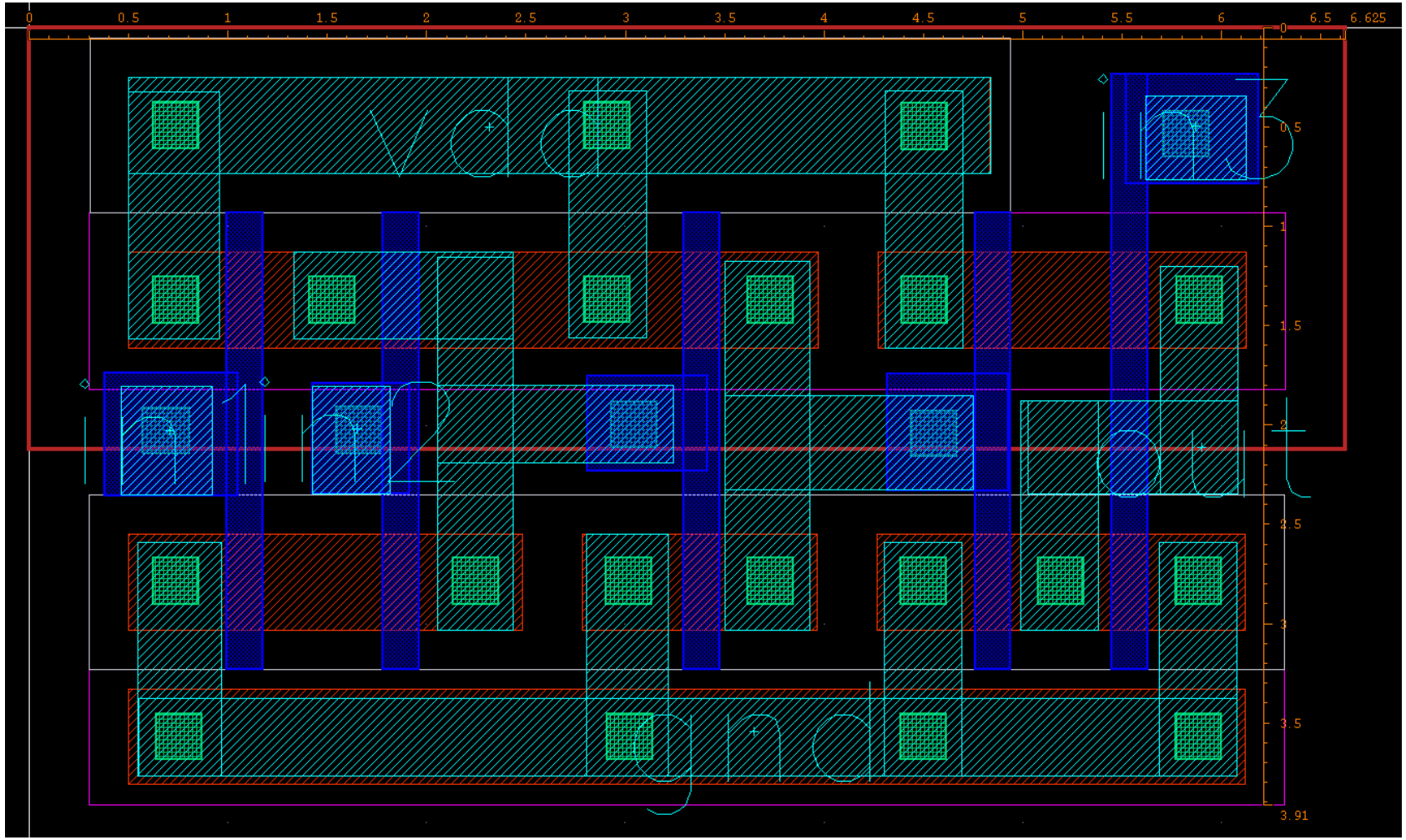
Project 1 - Reviewed

Top 3 Layout - 2nd



Project 1 - Reviewed

Top 3 Layout - 3rd



Project 1 - Reviewed

Reminds

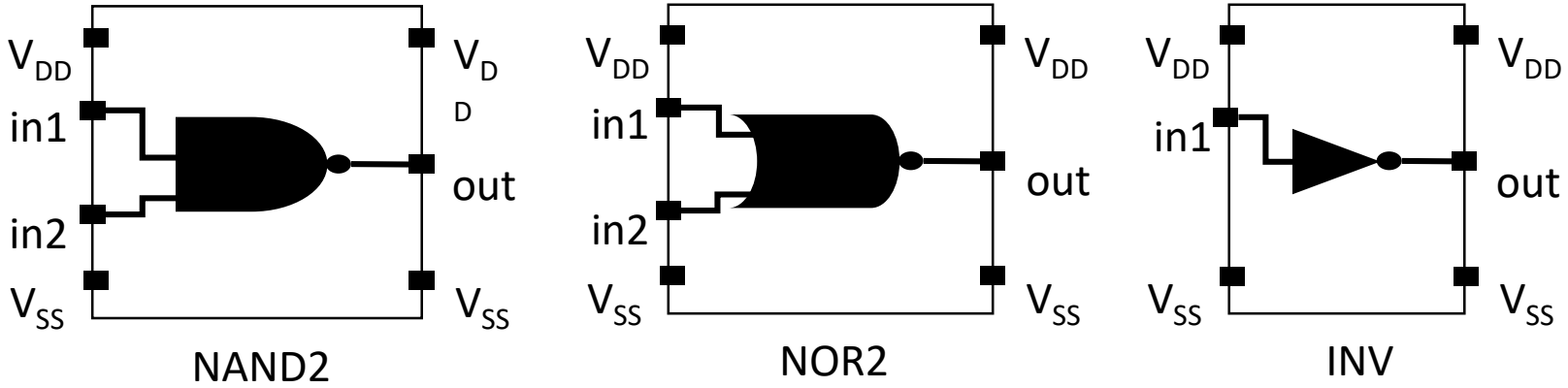
1. Please submit the correct file
2. Ruler annotation
3. Overwrite the result of DRC/LVS
4. FileZilla (Tool for transferring files between server and local storage)
   - Host: sftp://nthucad.cs.nthu.edu.tw

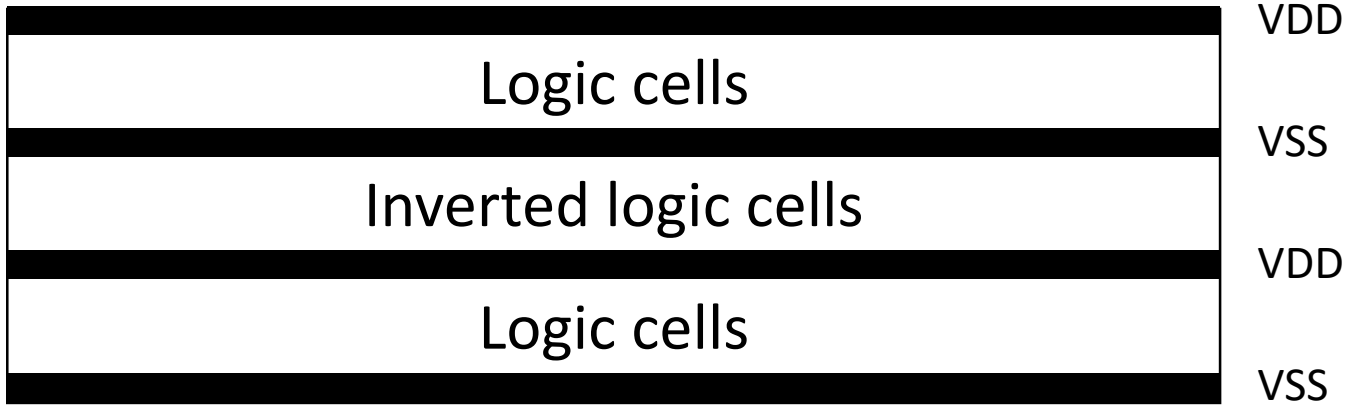
Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

Cell-Based Design

- Based on the idea of hierarchical design.
- Do not care about the internal details at the cell-level design.





Weinberger image array

Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

Standard Cell Setting

Library Prepared

- SSH Connection

  $ **`ssh -XY icXX (21, 51, 55)`**

- Go into the working directory **layout2**.

  $ **`tar zxvf layout2.tar.gz`** (for the first time)

  $ **`cd layout2`**

- You can use **ls** command to see all files in the directory.

  $ **`ls`**

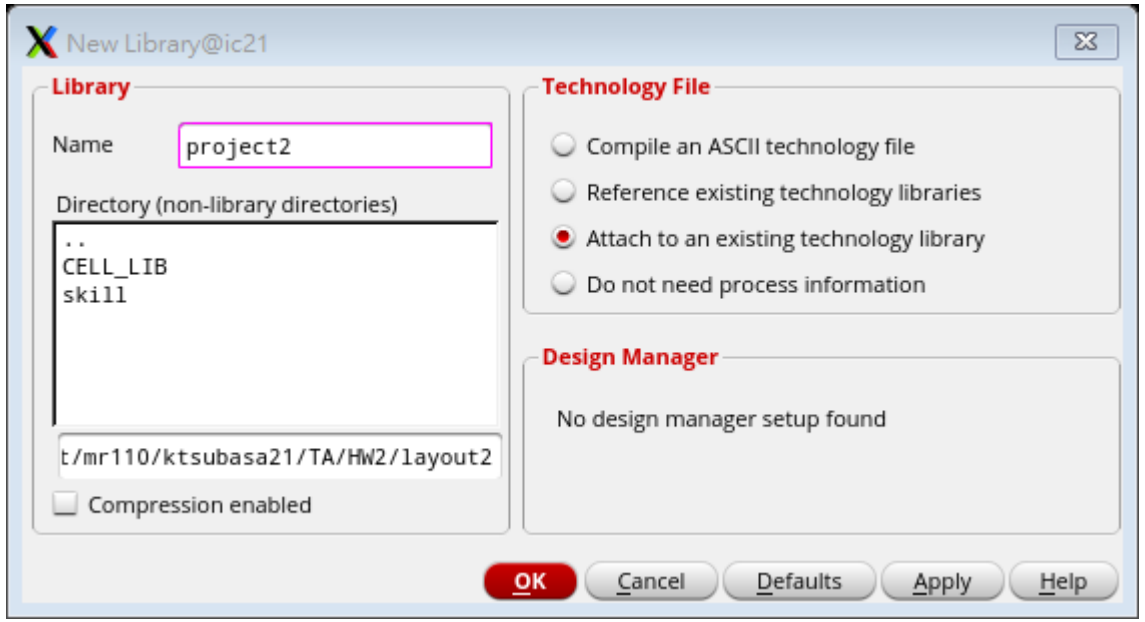
Files in the directory:

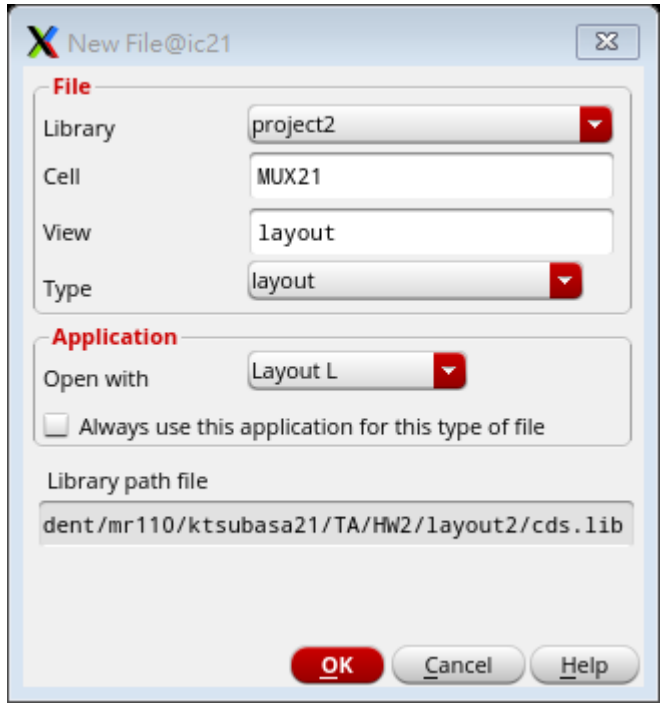
- CELL_LIB (new)
  - INV_2X
  - NAND_2X
  - NOR_2X
- CIC18_Calibre_DRC.drc
- CIC18_Calibre_LVS.lvs
- CIC_Lib
- cds.lib
- display.drf
- skill
- sourceMe

Standard Cell Setting (Cont.)

New Project Library Creation

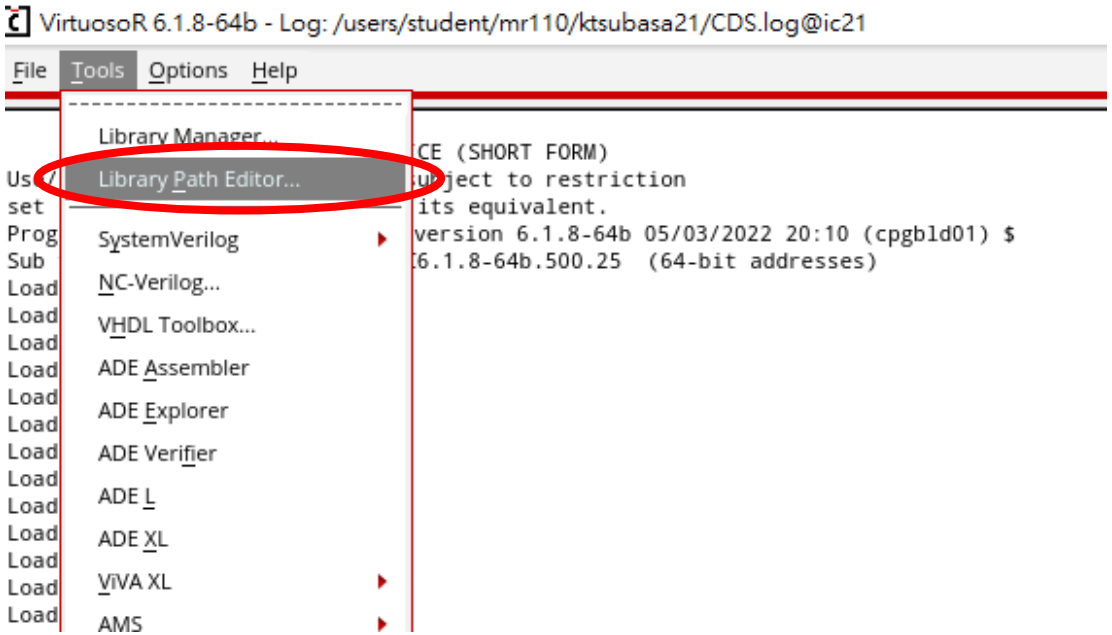
- Please refer the slide of Project1.
  - Library name: project2
  - Cell name: MUX21, FA, ADDER3

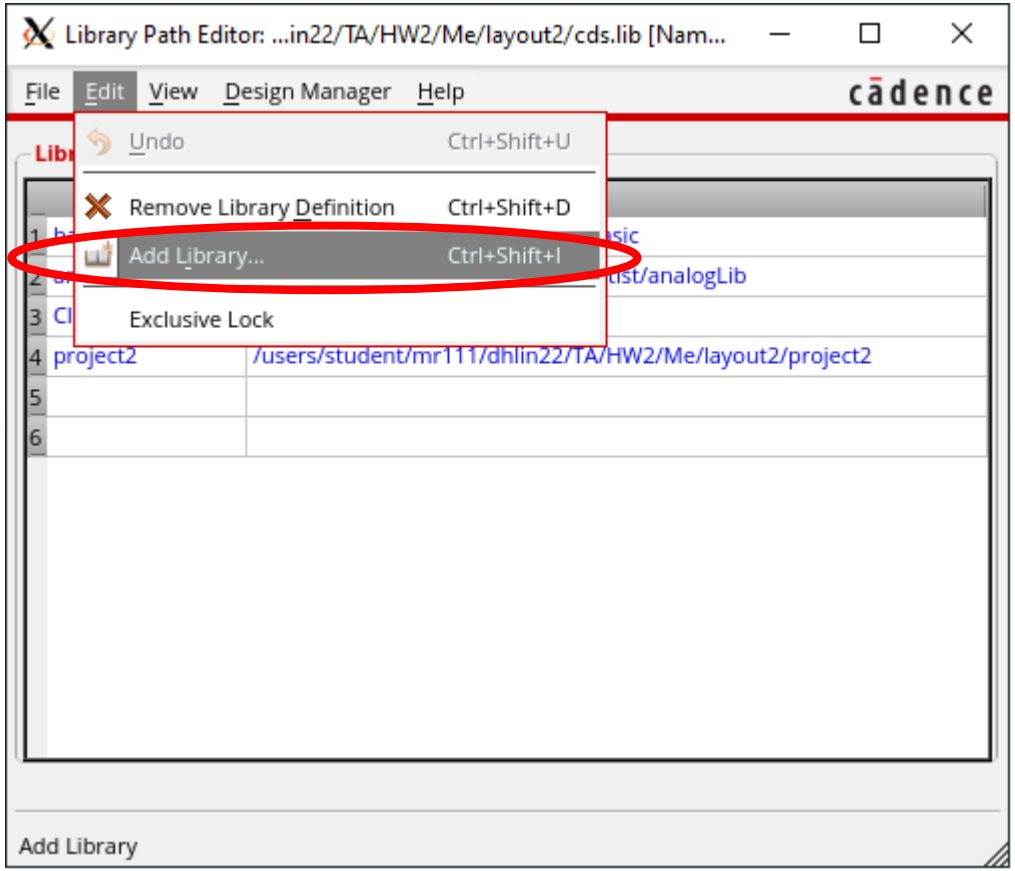




Standard Cell Setting (Cont.)

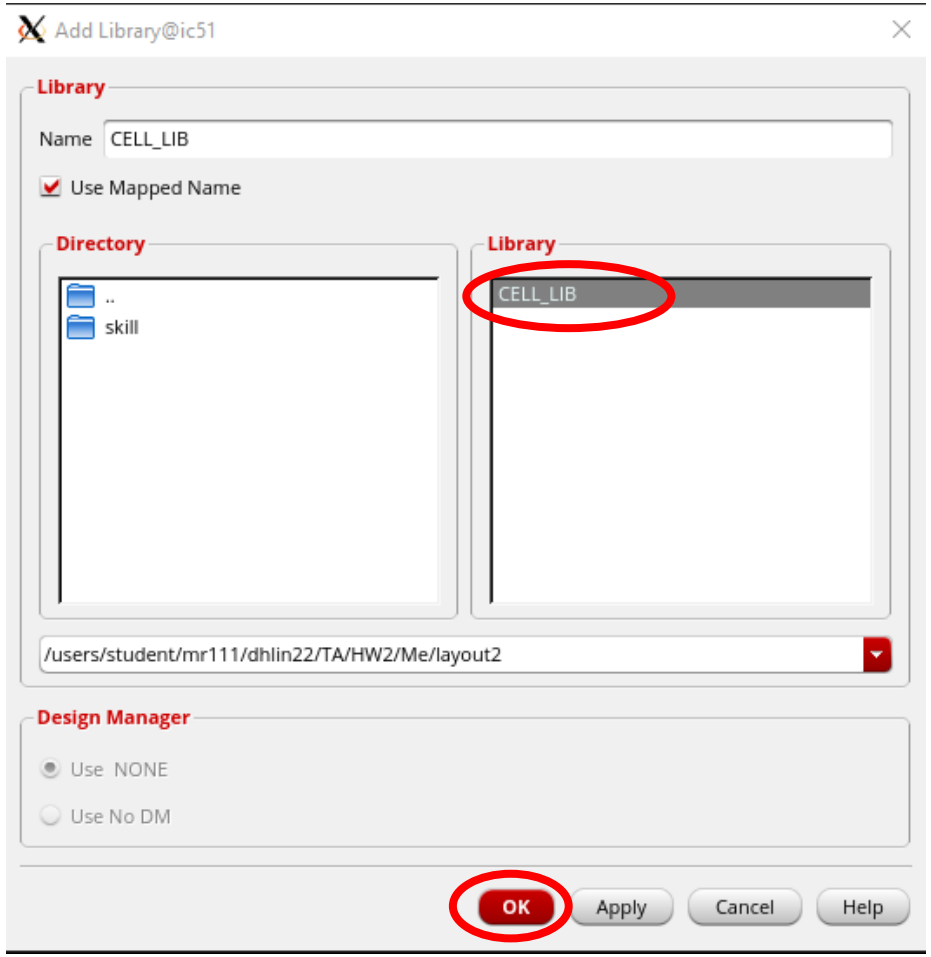
Set Library Path

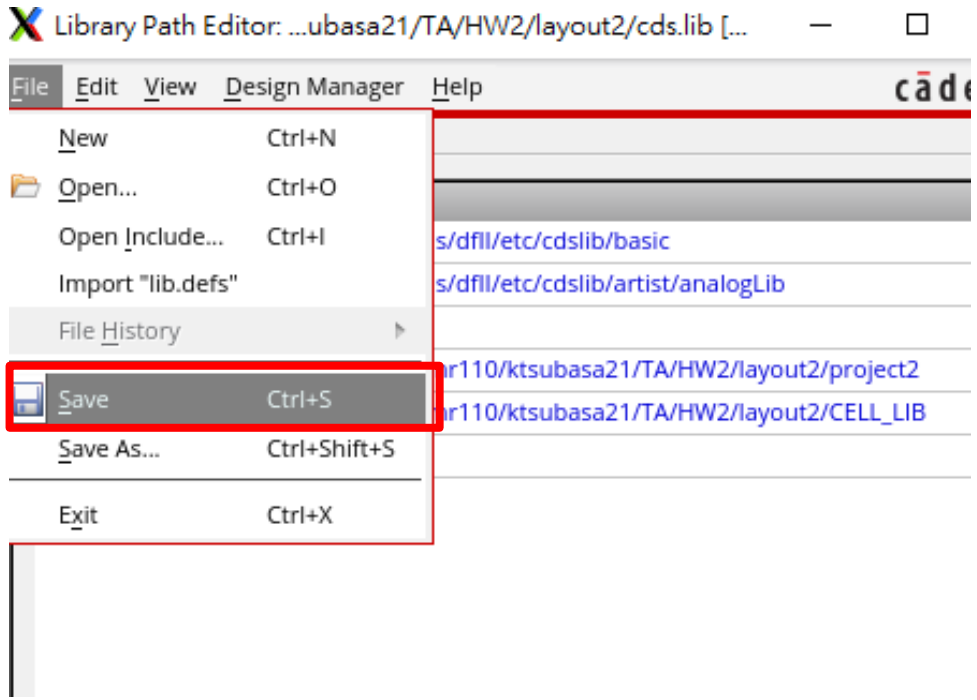




Standard Cell Setting (Cont.)

Set Library Path (Cont.)



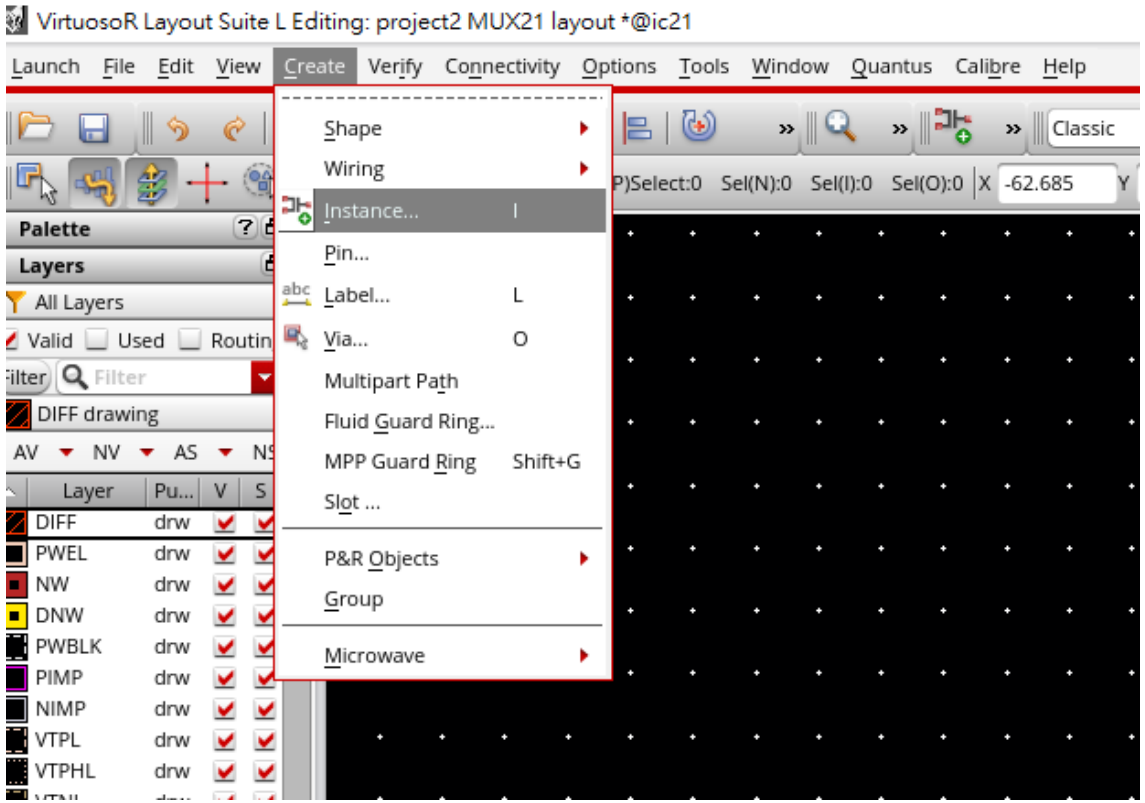


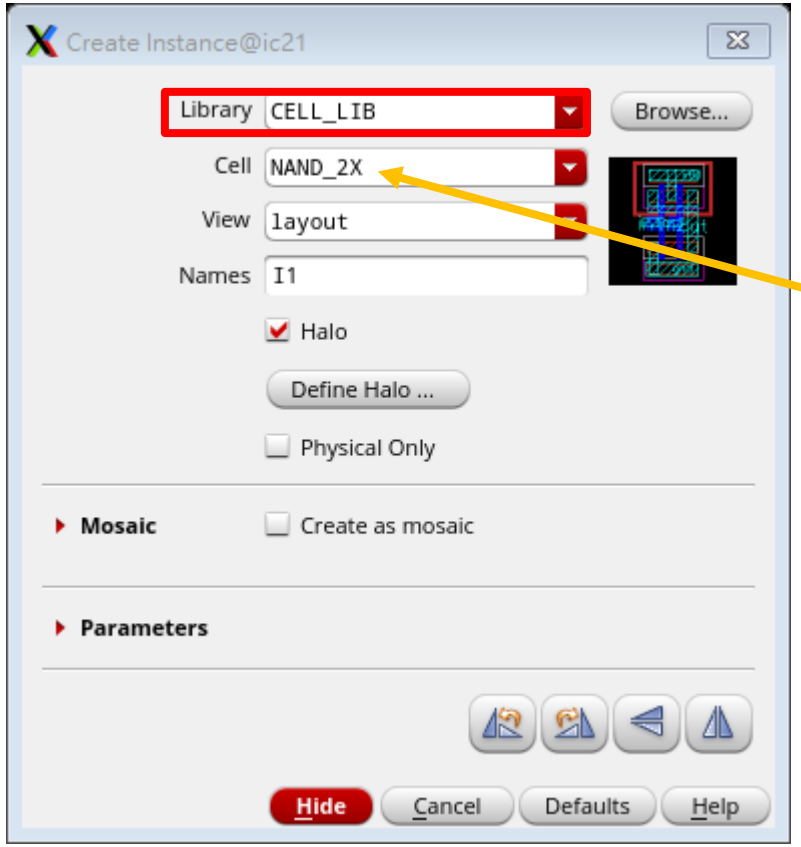
Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

Cell Duplication for Logic Function

Instance Creation

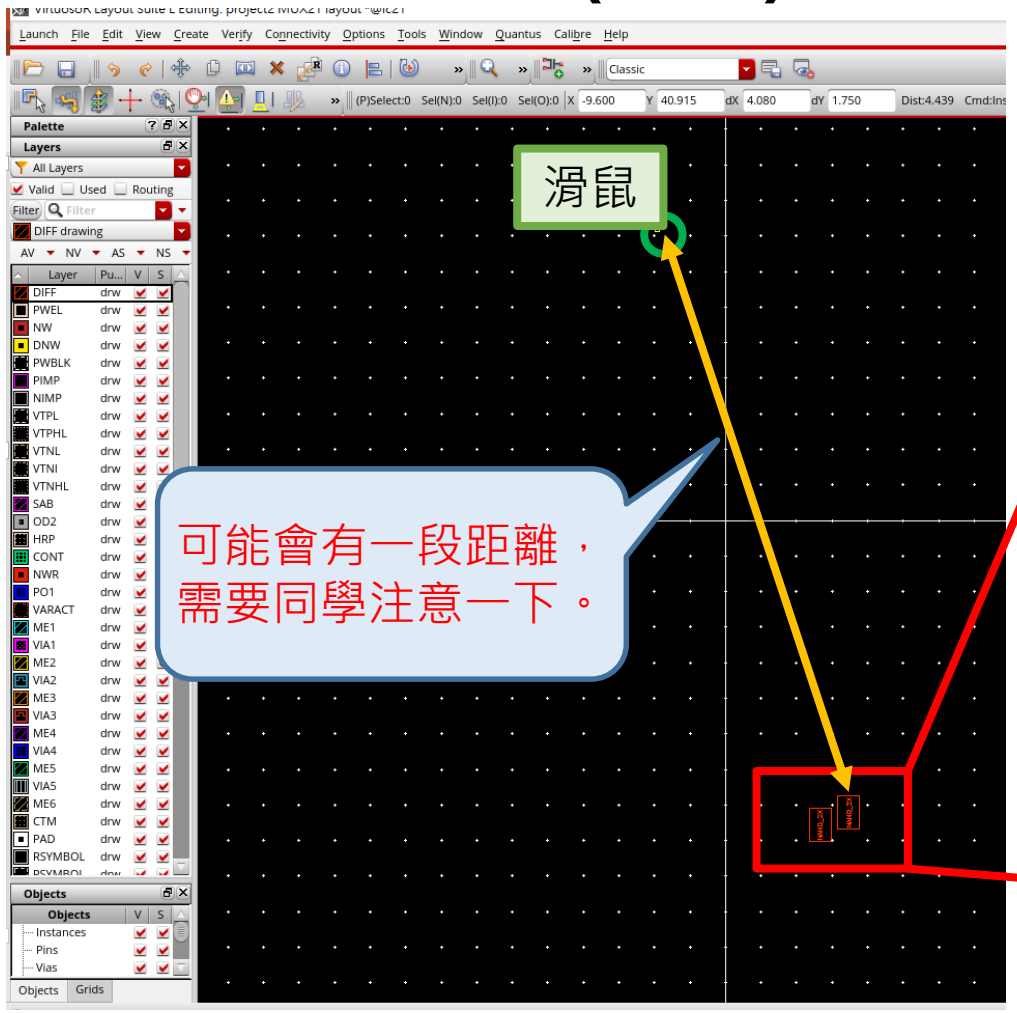




NAND_2X
NOR_2X
INV_2X

Cell Duplication for Logic Function

Instance Creation (Cont.)

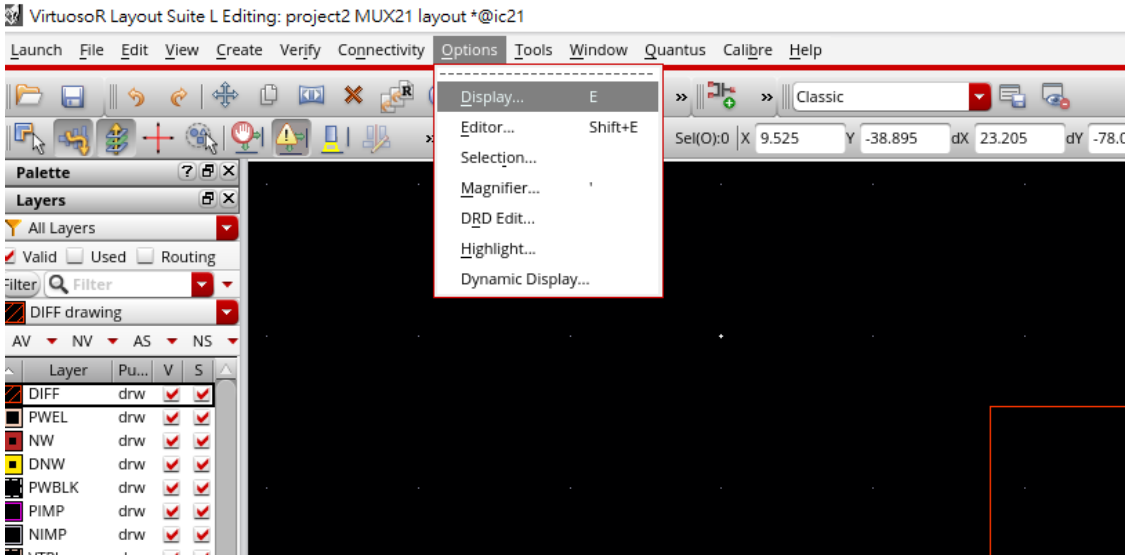


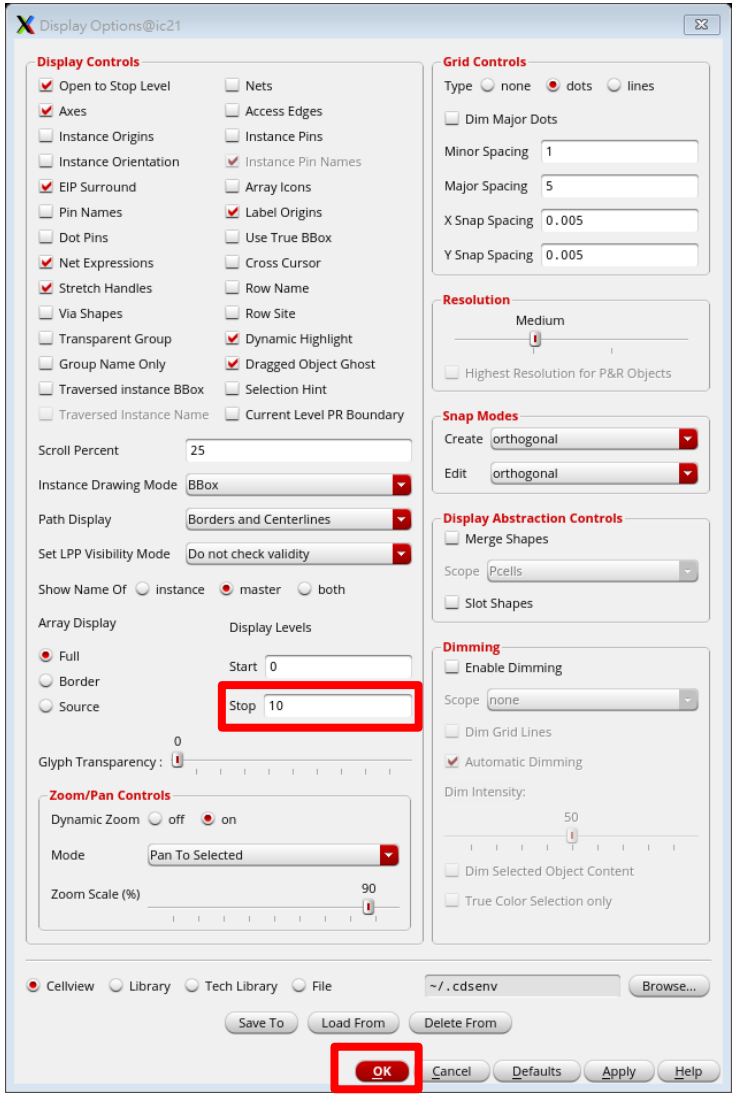
Left click to place the cell.

Cell Duplication for Logic Function

Cell Display

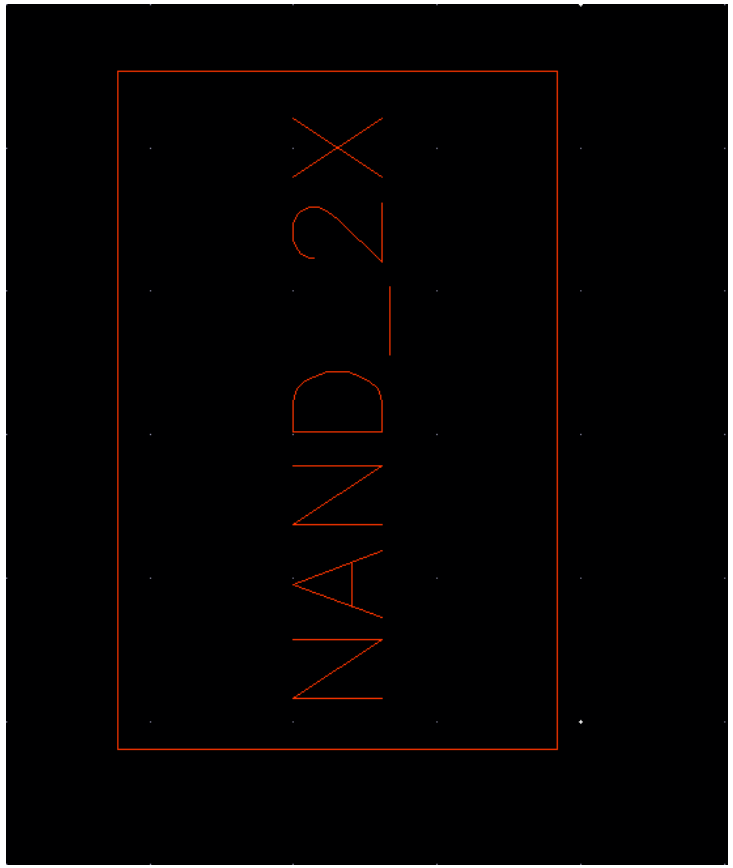
- Display Level
  - Level 0: Only top level
  - Level 10: all levels



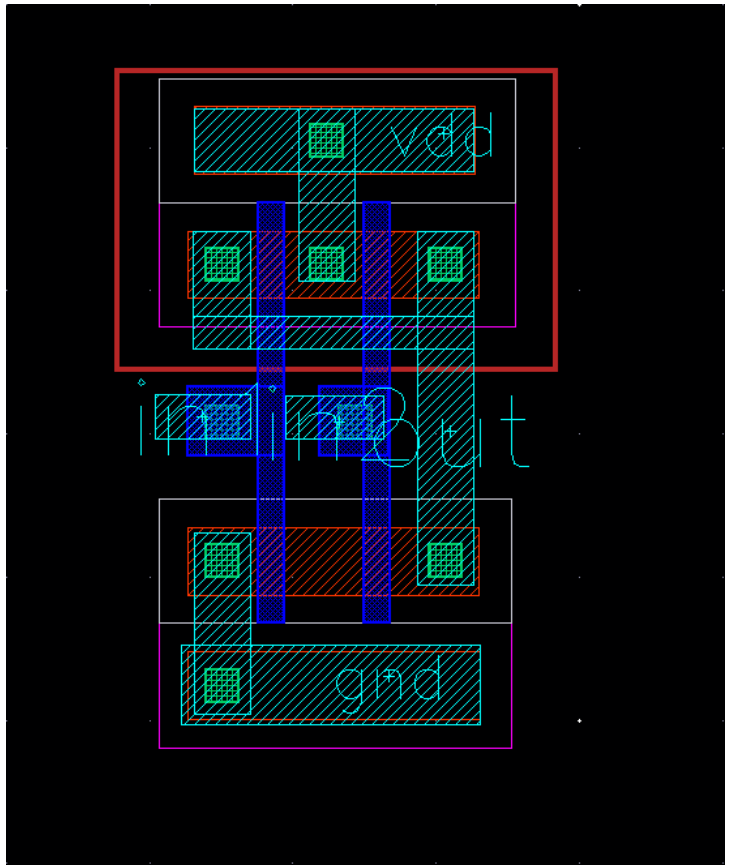


Cell Duplication for Logic Function

Cell Display (Cont.)



Display Level 0



Display Level 10

Cell Duplication for Logic Function

Virtuoso Layout Hotkeys

	Category	Action	Hotkey
VIEW	ZOOM	IN	cntl+z
		OUT	shift+Z
	FIT	Fit whole layout to exiting window	f
	VIEW HIERARCHY	MORE DETAIL	shift+F
		LESS DETAIL	cntl+f
	REDRAW	---	cntl+r
EDIT	STRETCH	---	s
	MOVE	---	m
	COPY	---	c
	UNDO	---	u
	SELECT ALL	Select all objects on the window	cntl +a
	DESELECT ALL	Deselects all selected objects	cntl +d
MISC	PROPERTIES	Show properties of selected object	q
	RULER	CREATE	k
		DELETE ALL	shift+K
	HIERARCHY	DESCEND	shift+X
		RETURN	shift+B

Cadence Virtuoso Layout Hotkeys

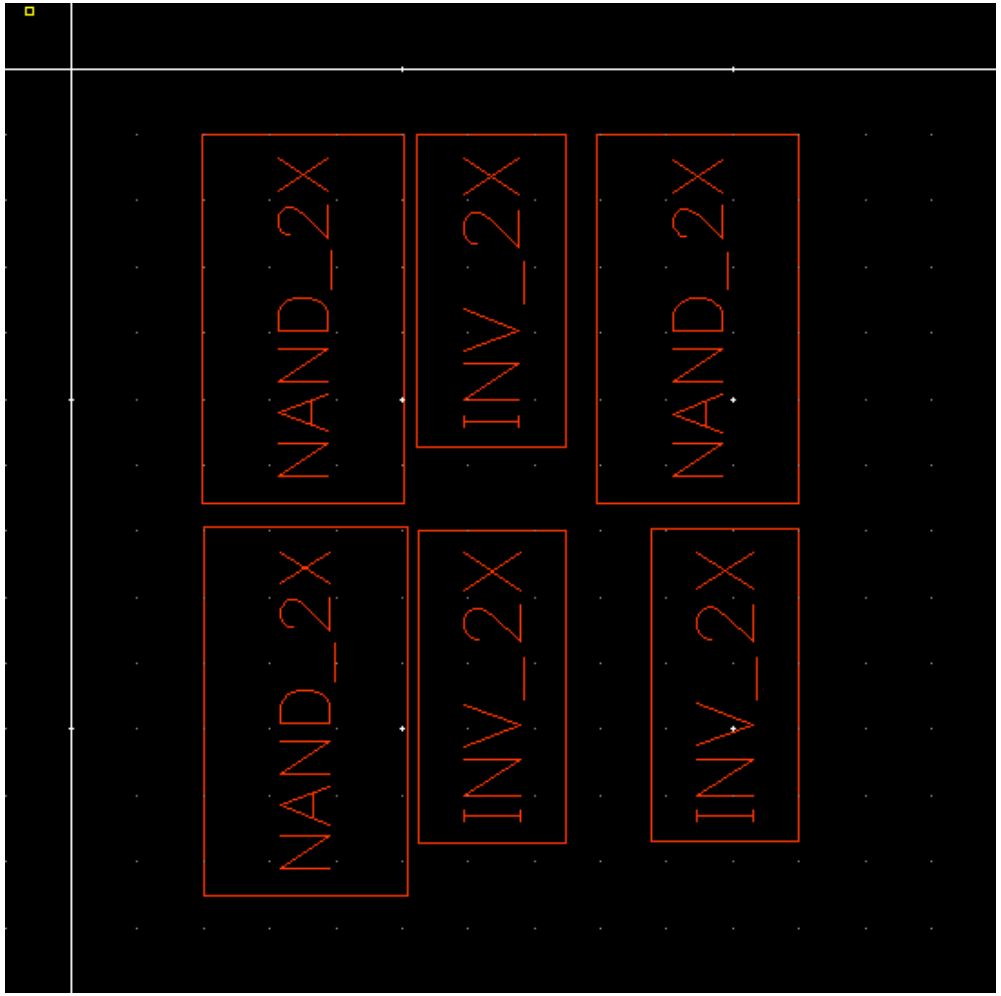
Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

Example

f = a*b*c*d

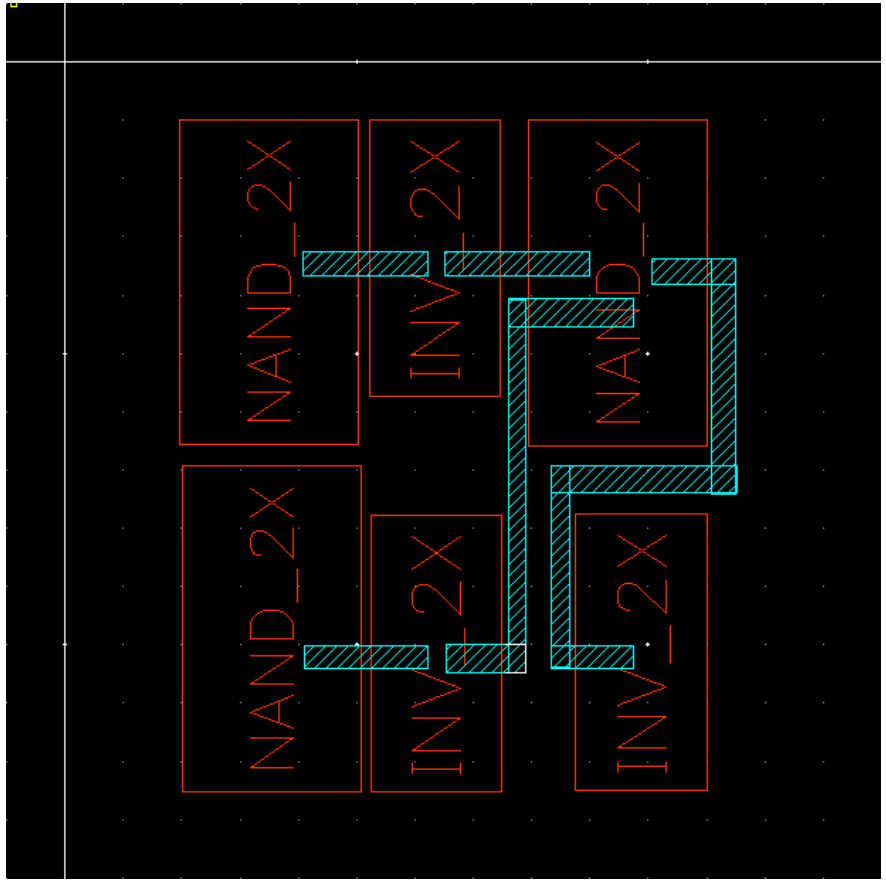
Cell Duplication



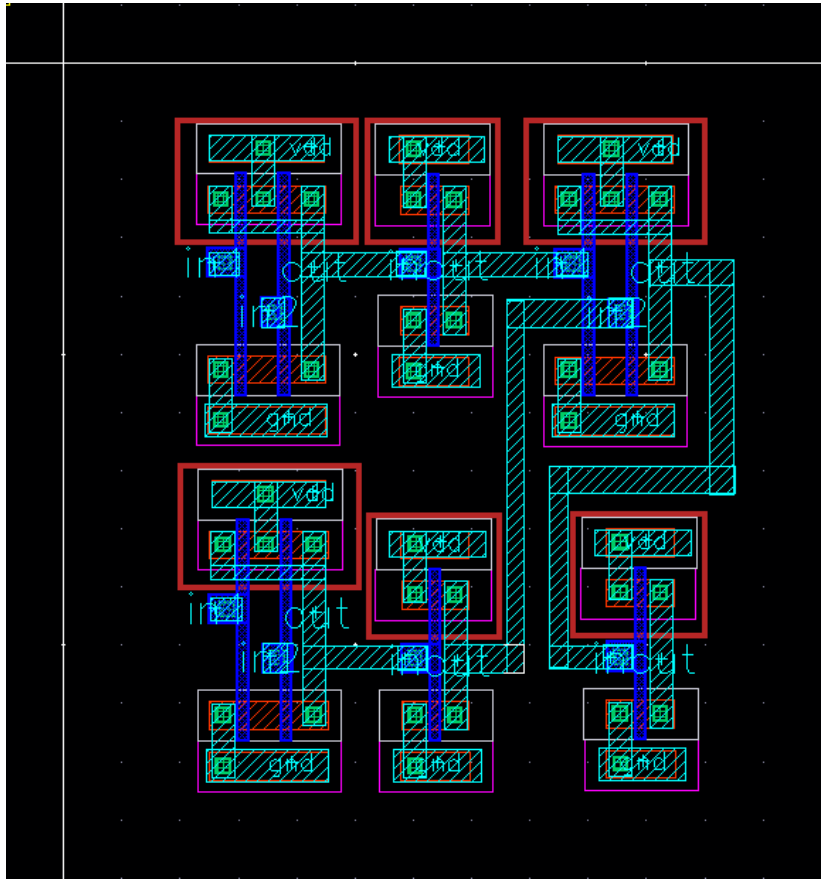
Example (Cont.)

f = a*b*c*d

Metal Wire Connection



Display Level 0

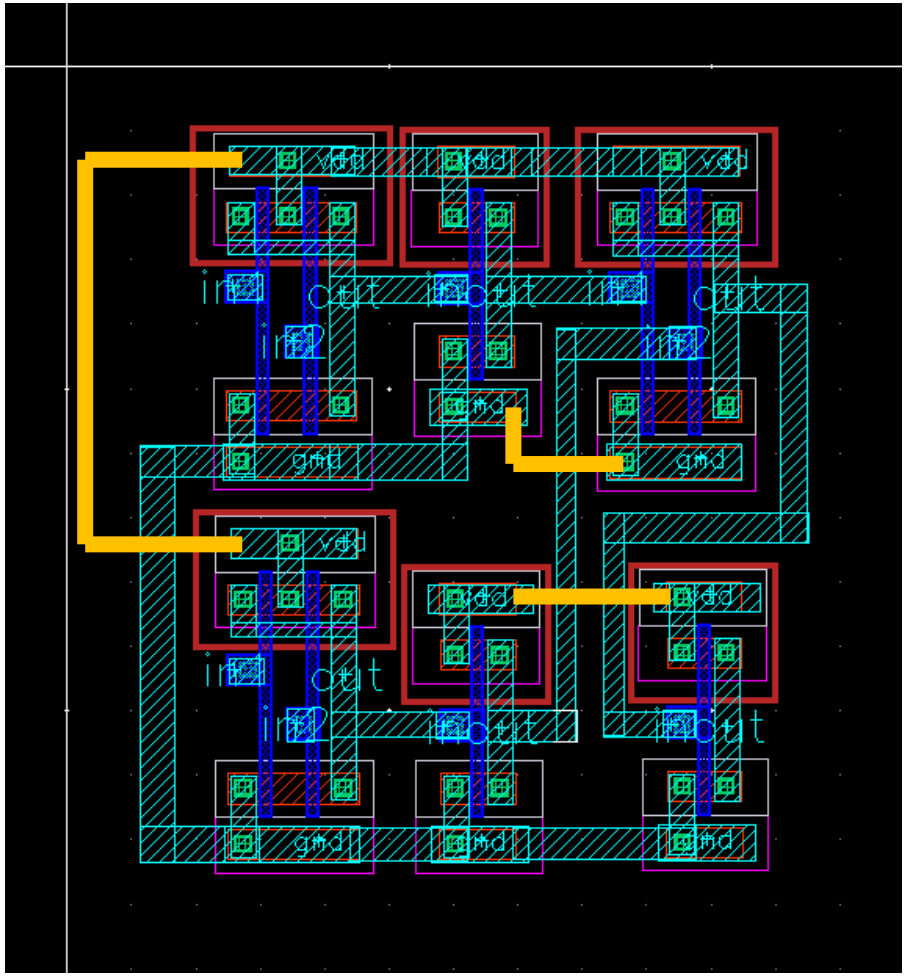


Display Level 10

Example (Cont.)

f = a*b*c*d

vdd/gnd connection by metal 1

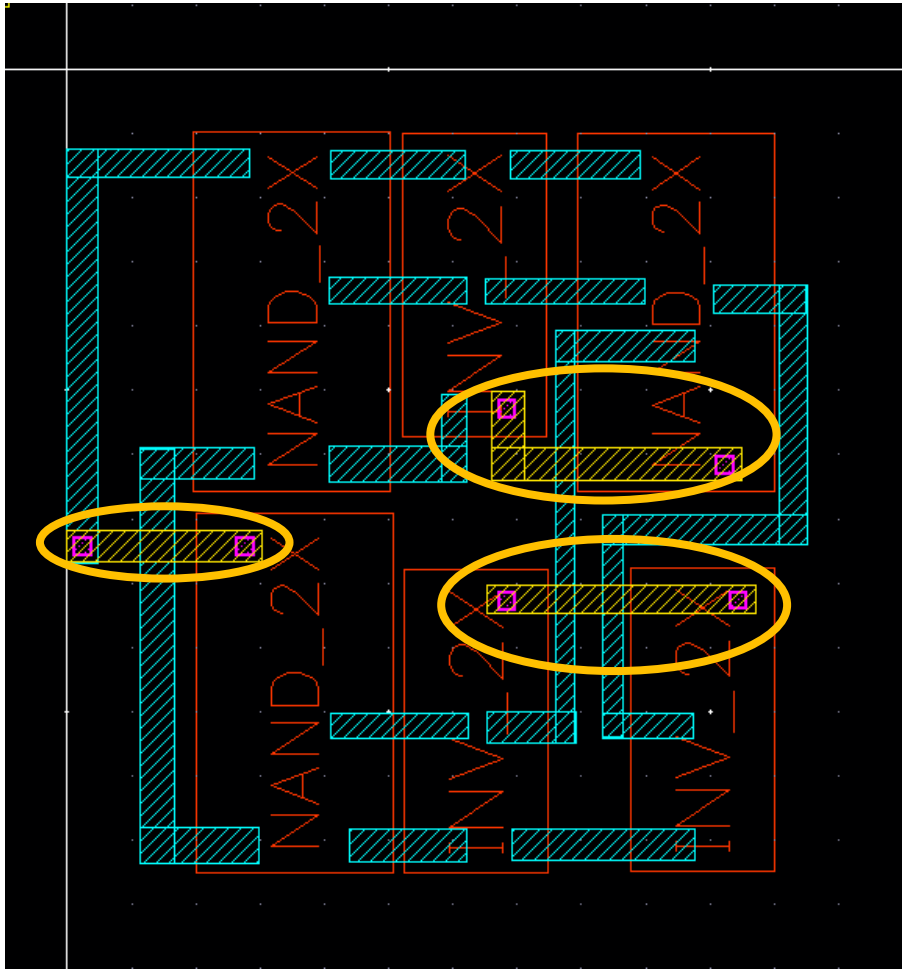


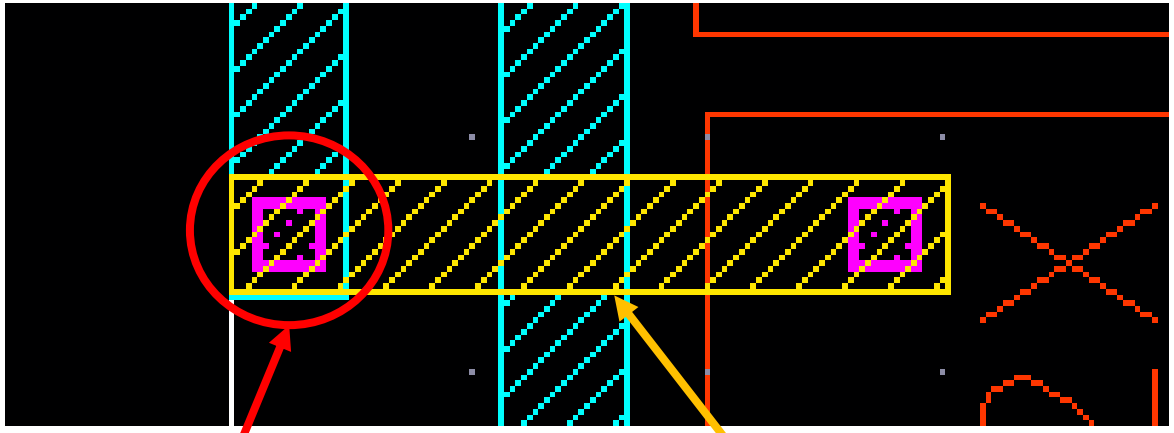
- You cannot just use metal 1 to connect orange segments.
- It will cause unexpected shorts to other metal 1.

Example (Cont.)

f = a*b*c*d

vdd/gnd connection by metal 2





VIA1
size must be
0.25um * 0.25um

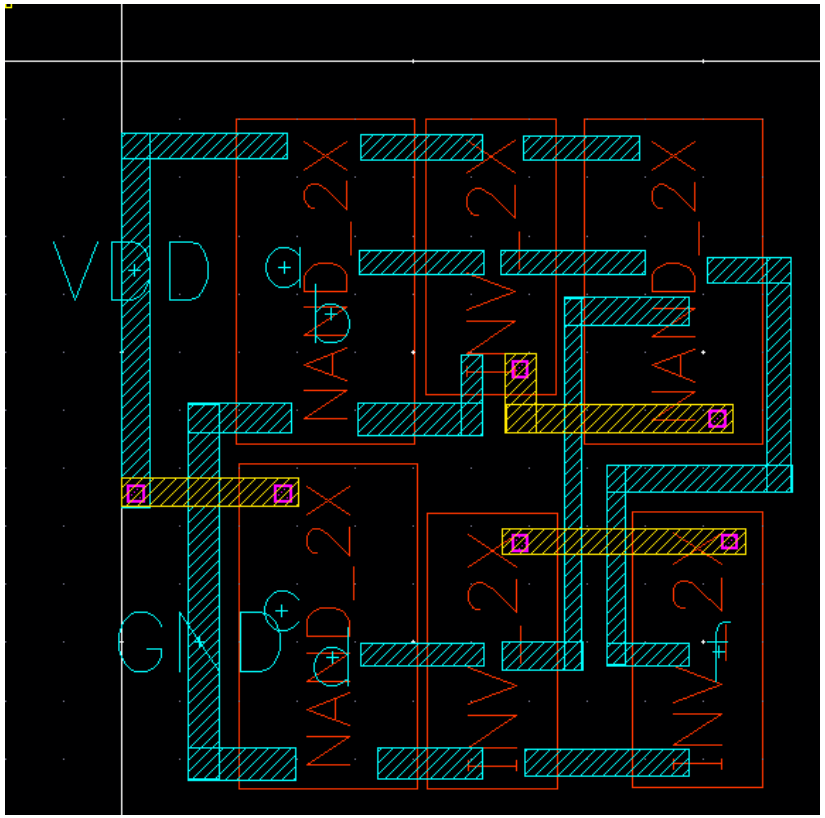
ME2 (metal 2)

Example (Cont.)

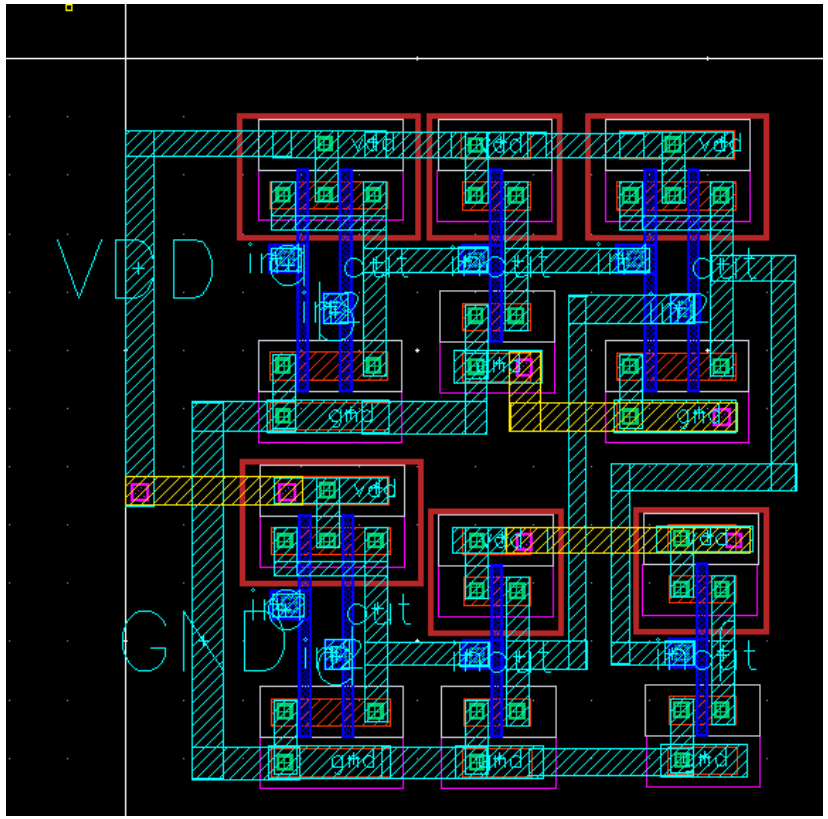
f = a*b*c*d

Labels in standard cells are different from current cell.

*P.S. You can display **level zero** for checking.*



Display Level 0



Display Level 10

Example (Cont.)

f = a*b*c*d

Schematic File

- “x” represents the cell
  - x_[name_as_you_like] [Pin 1] [Pin 2] … [Pin N] [cell_type]

- E.g.

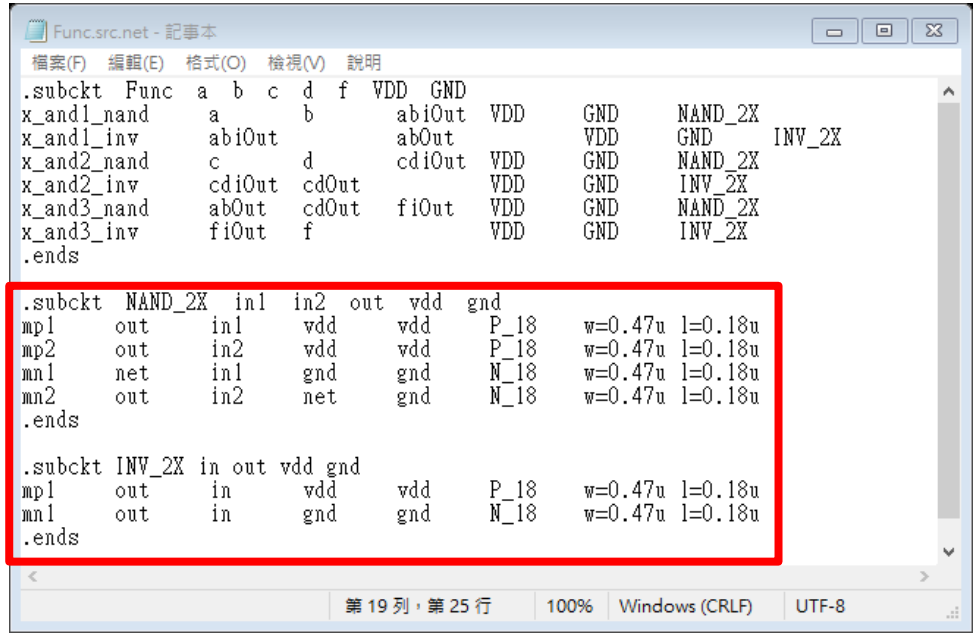
```
.subckt  Func  a  b  c  d  f  VDD  GND
x_and1_nand     a        b        abiOut   VDD    GND    NAND_2X
x_and1_inv      abiOut   abOut             VDD    GND    INV_2X
x_and2_nand     c        d        cdiOut   VDD    GND    NAND_2X
x_and2_inv      cdiOut   cdOut             VDD    GND    INV_2X
x_and3_nand     abOut    cdOut    fiOut    VDD    GND    NAND_2X
x_and3_inv      fiOut    f                 VDD    GND    INV_2X
.ends
```

Example (Cont.)

f = a*b*c*d

Schematic File (Cont.)

- Definition of standard cells also needs to be put into schematic file.
  - The standard definition is at **`layout2/CELL_LIB/[cell_name].`**
  - E.g. `layout2/CELL_LIB/NAND_2X/NAND_2X.src.net`



Func.src.net - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
.subckt  Func  a  b  c  d  f  VDD  GND
x_and1_nand     a       b       abiOut  VDD     GND     NAND_2X
x_and1_inv      abiOut          abOut           VDD     GND     INV_2X
x_and2_nand     c       d       cdiOut  VDD     GND     NAND_2X
x_and2_inv      cdiOut  cdOut           VDD     GND     INV_2X
x_and3_nand     abOut   cdOut   fiOut   VDD     GND     NAND_2X
x_and3_inv      fiOut   f               VDD     GND     INV_2X
.ends

.subckt  NAND_2X  in1  in2  out  vdd  gnd
mp1     out     in1     vdd     vdd     P_18    w=0.47u l=0.18u
mp2     out     in2     vdd     vdd     P_18    w=0.47u l=0.18u
mn1     net     in1     gnd     gnd     N_18    w=0.47u l=0.18u
mn2     out     in2     net     gnd     N_18    w=0.47u l=0.18u
.ends

.subckt INV_2X in out vdd gnd
mp1     out     in      vdd     vdd     P_18    w=0.47u l=0.18u
mn1     out     in      gnd     gnd     N_18    w=0.47u l=0.18u
.ends
```

第 19 列，第 25 行 100% Windows (CRLF) UTF-8

Outline

- Project 1 - Reviewed
- Cell-Based Design
- Standard Cell Setting
- Cell Duplication for Logic Function
- Example
- Project

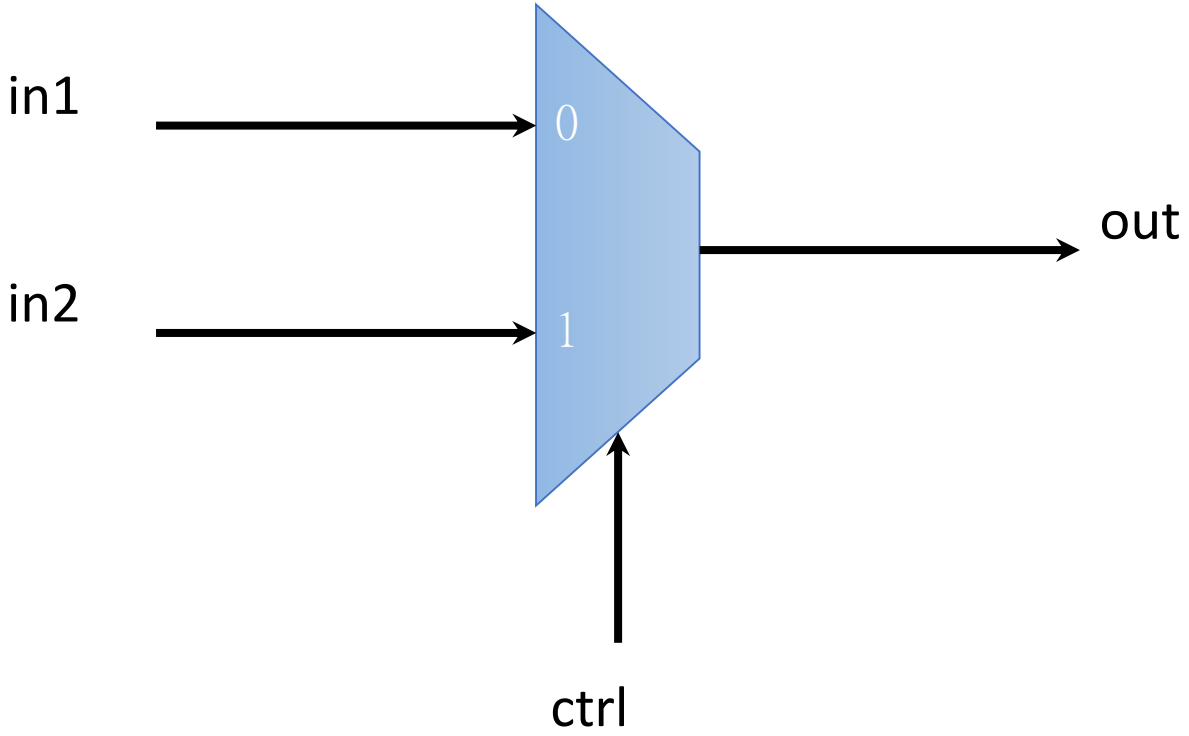
Project Requirements

- Step 1: Design a 2x1 multiplexer (MUX21)
  - Use cell-based design (using NAND/NOR/INV given by TA)
  - Pass DRC and LVS
- Step 2: Design a 1-bit full adder (FA)
  - Use cell-based design (using NAND/NOR/INV given by TA and MUX designed by you)
  - Pass DRC and LVS
- Step 3: Design a 3-bit carry select adder (ADDER3)
  - Draw cell-based layout (***only*** reuse your own MUX21 & FA)
  - Pass DRC and LVS

Project Requirements (Cont.)

2x1 Multiplexer

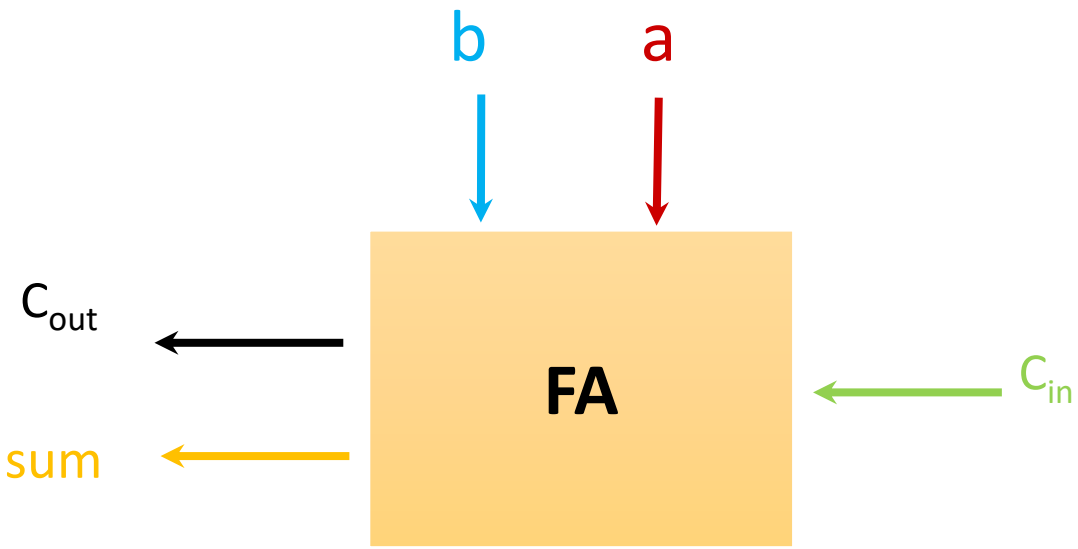
- Cell name: MUX21
- Input
  - in1, in2, ctrl
- Output
  - out
- Function
  - out = (ctrl==0) ? in1 : in2



Project Requirements (Cont.)

1-bit Full Adder

- Cell name: FA
- Input
  - a, b, c_in
- Output
  - c_out, sum
- Function
  - {c_out, sum} = a + b + c_in

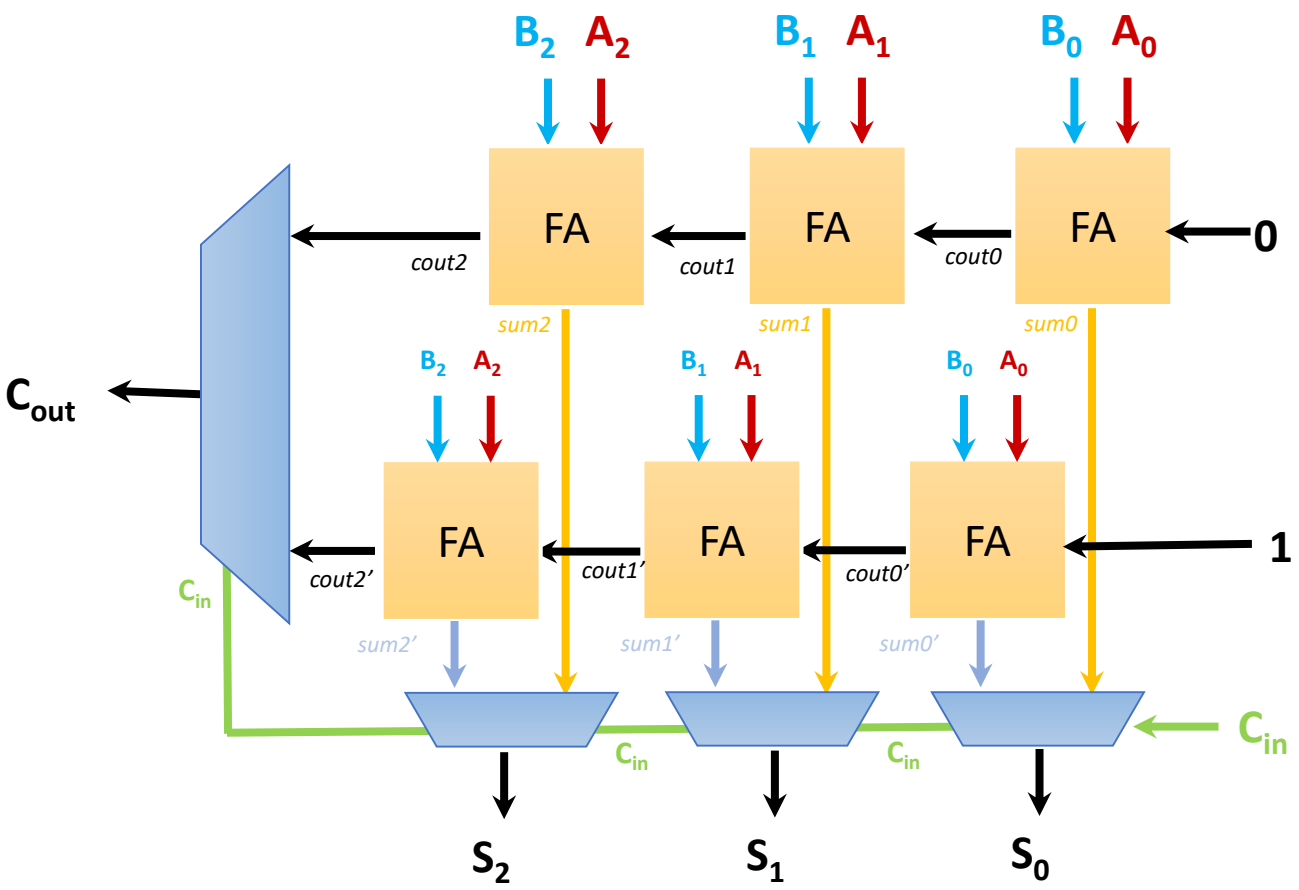


Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Project Requirements (Cont.)

3-bit Carry Select Adder

- Cell name: ADDER3
- Input
  - A[0:2]: A0, A1, A2
  - B[0:2]: B0, B1, B2
  - CIN
- Output
  - S[0:2]: S0, S1, S2
  - COUT
- Function
  - {COUT, S[0:2]} = A[0:2] + B[0:2] + CIN



Project Requirements (Cont.)

Transistor Schematic

- Schematic format
  - Cell and port names should be named as follows.
  - I/O port must be **in order**.

```
.subckt ADDER3 A0 A1 A2 B0 B1 B2 CIN S0 S1 S2 COUT VDD GND
    ...
.ends

.subckt FA a b c_in sum c_out VDD GND
    ...
.ends

.subckt MUX21 in1 in2 ctrl out VDD GND
    ...
.ends
    ...
```

Project Requirements (Cont.)

Layout

- Labels should be **STRICTLY** named as follows.
  - Cell name
    - ADDER3
  - Input
    - A0, A1, A2, B0, B1, B2, CIN, VDD, GND
  - Output
    - S0, S1, S2, COUT
  - No naming constraint to internal signals.
  - You can use only metal layer 1 and 2 (ME1 & ME2).
  - Using rulers to show the width and height of your three layouts.

Report

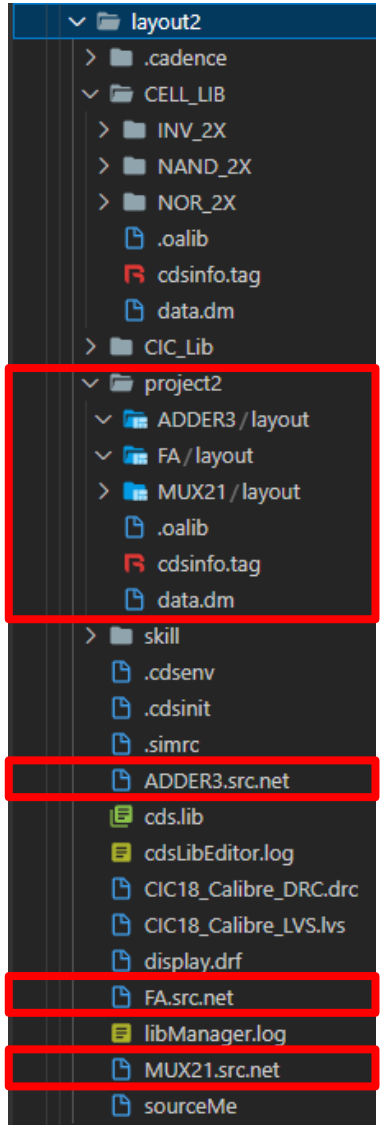
- Your report should contain the following content, and you can add more as you wish.
  - Your name and student ID
  - **NINE** screenshots:
    - Layout of MUX21, FA, ADDER3 with rulers (**BOTH** display level 0 & 10, height & width)
    - DRC summary report of ADDER3
    - The message of passing LVS of ADDER3
    - LVS schematic of **ADDER3.src.net** (screenshot of the text file)
  - What else did you do to enhance your layout quality?
  - What have you learned from this homework?
  - What problem(s) have you encountered in this homework?

Grading

- Deadline: 2022/12/25 (Sun.) 23:59
- Files to upload:
  1. **.pdf** format report
  2. **.tar.gz** file includes all files in directory **layout2**

  You can use the following command to compress your directory on a workstation:

  ```
  $ tar zcvf StudentID_project2.tar.gz layout2/
  ```
- Upload 1. and 2. to eeclass.
  – Name the files as "StudentID_report2.pdf" and "StudentID_project2.tar.gz", respectively.



Grading (Cont.)

- Assessment
  - Report: 10%
  - Layout: 20%
  - Layout Area (By Increasing Order): 30%

  (1st ~ 25th: 30 points, 26th ~ 50th: 20 points, 51st ~: 10 points)
  - DRC pass: 20%
  - LVS pass + LVS schematic: 20%

- Early bird bonus: Extra 10 points for submission before 2022/12/18 (Sun.) 23:59
- Penalty for late submission: -20% per day.
- No Ruler (must not be too large or too small): -10 points
- Naming Rule Violation: -5 points

Notice

- File Naming
  - Library name: project2
  - Cell name: MUX21, FA, ADDER3
  - LVS netlist name: [cell_name].src.net (e.g. ADDER3.src.net)

- Plagiarism → 0 point
- Dishonest contents in the report → 0 point