

# NCTU-EE IC LAB – Fall 2023

## Lab01 Exercise

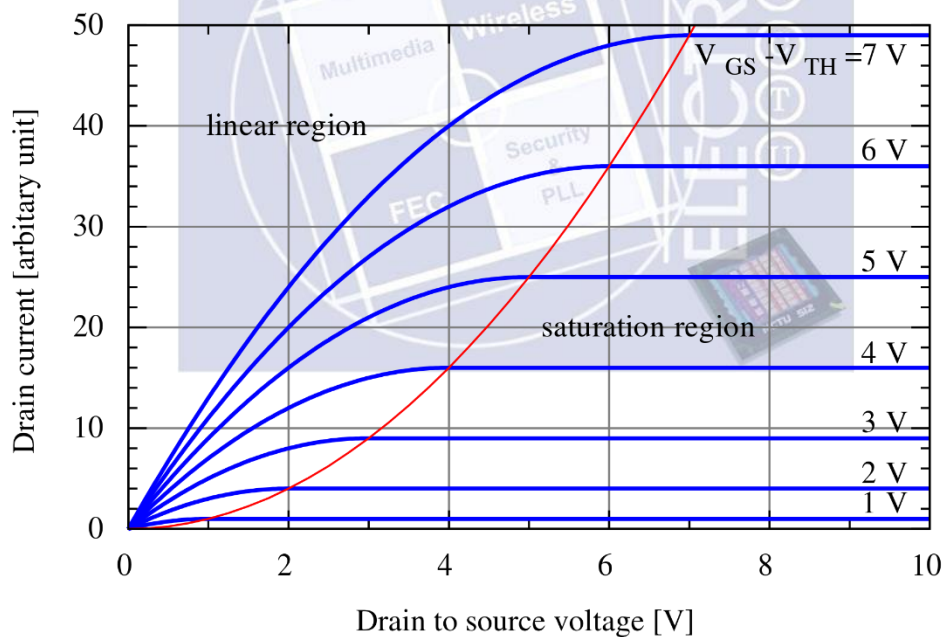
### Design: Supper MOSFET Calculator(SMC)

#### Data Preparation

1. Extract files from TA's directory:  
`% tar -xvf ~iclabTA01/Lab01.tar`

#### Design Description and Examples

One day after microelectronics class, you are excited about what the professor talked in class, the fantastic characteristics of Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET). Therefore, you want to design a Supper MOSFET Calculator to calculate the drain current  $I_D$  and transconductance  $g_m$  in a short time. Furthermore, you also want to find what if given numerous combinations of width,  $V_{GS}$  and  $V_{DS}$ , which one could get the maximum value? To satisfy your curiosity, now you are going to conquer this problem.



Before you start your work, here is a quick review of MOS for you.

## ✓ MOS's modes of operation

The operation of a MOSFET can be separated into four different modes, depending on the voltages at the terminals. Here we consider an NMOS model.

### 1. Triode region (Linear region)

- ✓ Condition:  $V_{GS} - V_{th} \geq 0$  &&  $(V_{GS} - V_{th}) > V_{DS}$
- ✓ Current:  $I_D = \frac{1}{2} \mu_n C_{OX} \left( \frac{W}{L} \right) [2(V_{GS} - V_{th})V_{DS} - V_{DS}^2] = K_n W [2(V_{GS} - V_{th})V_{DS} - V_{DS}^2]$
- ✓ Transconductance:  $g_m = \mu_n C_{OX} \left( \frac{W}{L} \right) V_{DS} = 2K_n W V_{DS}$

### 2. Saturation region

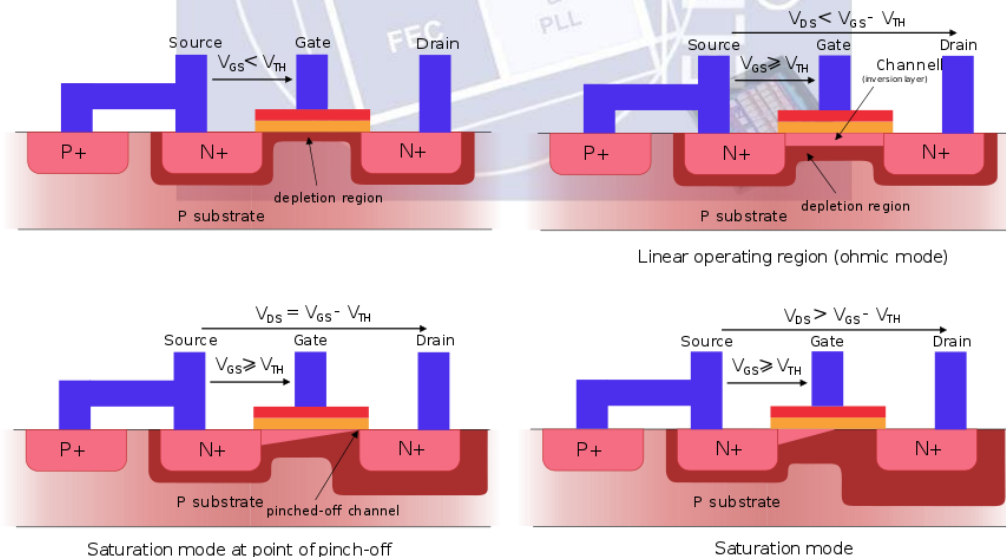
- ✓ Condition:  $V_{GS} - V_{th} \geq 0$  &&  $(V_{GS} - V_{th}) \leq V_{DS}$
- ✓ Current:  $I_D = \frac{1}{2} \mu_n C_{OX} \left( \frac{W}{L} \right) (V_{GS} - V_{th})^2 = K_n W (V_{GS} - V_{th})^2$
- ✓ Transconductance:  $g_m = \mu_n C_{OX} \left( \frac{W}{L} \right) (V_{GS} - V_{th}) = 2K_n W (V_{GS} - V_{th})$

### 3. Subthreshold region (cut-off)

- ✓ Condition:  $V_{GS} - V_{th} < 0$
- ✓ Current:  $I_D \propto e^{V_{GS}}$  (behavior like BJT, very small current)
- ✓ Will not happen in this lab since  $V_{GS}$  is range from 1~7 which won't smaller than  $V_{th} = 1$

### 4. Deep triode region

- ✓ Condition:  $V_{GS} - V_{th} \geq 0$  &&  $2(V_{GS} - V_{th}) \gg V_{DS}$
- ✓ Current:  $\mu_n C_{OX} \left( \frac{W}{L} \right) [(V_{GS} - V_{th})V_{DS}]$
- ✓ Will not happen in this lab (since  $V_{DS}$  is range from 1~7 which is not negligible level. )



- To simplify calculation, we assume  $K_n = \frac{1}{2} \mu_n C_{OX} \left( \frac{1}{L} \right) = \frac{1}{3}$ ,  $V_{th} = 1$  in this lab
- Also, we **don't** consider **body effect** and **channel length modulation** in this lab.
- No need to consider fixed-point division, **round-down to integer only**.

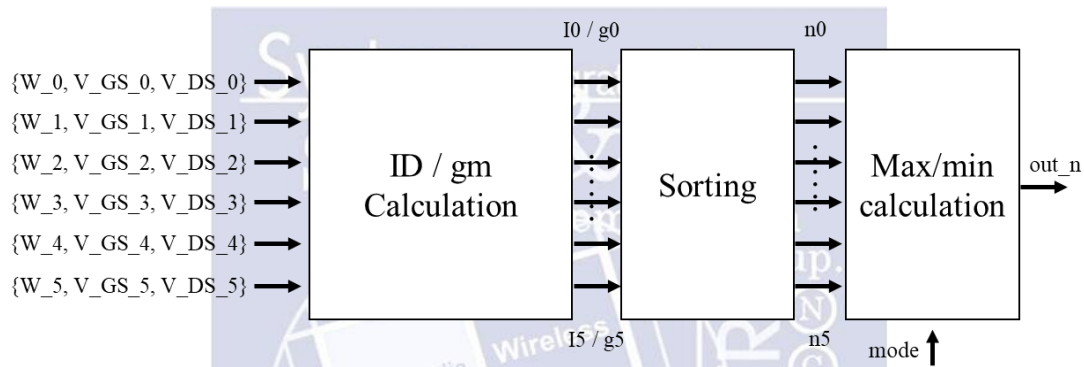
## ✓ Description of this lab

### ➤ Input:

You will receive a sequence with 6 combinations of **width**,  $V_{GS}$  and  $V_{DS}$ , each input value would range from 1~7 (3 bit). Each name is as following.

- $W_0, V_{GS\_0}, V_{DS\_0}$
- $W_1, V_{GS\_1}, V_{DS\_1}$
- ...
- $W_5, V_{GS\_5}, V_{DS\_5}$

Also, you will receive 2-bits mode code. In this lab, one bit **mode[0]** indicates that you would calculate whether the drain current or the transconductance. And, another one bit **mode[1]** shows that your output should be largest or smallest value.



### ➤ Calculate

You need to determine which mode would the MOS operate based on the given  $V_{GS}$  and  $V_{DS}$ . Next, you should calculate the drain current or the transconductance with corresponding equation. Below is the equation you would use in this lab.

- If  $(V_{GS\_n} - 1 > V_{DS\_n})$  // Triode mode
  - $I_{Dn} = K_n W [2(V_{GS} - V_{th})V_{DS} - V_{DS}^2] = \frac{1}{3} \{W [2(V_{GS\_n} - 1)V_{DS\_n} - V_{DS\_n}^2]\}$
  - $g_{mn} = 2K_n W V_{DS} = \frac{2}{3} [W V_{DS\_n}]$
- If  $(V_{GS\_n} - 1 \leq V_{DS\_n})$  // Saturation mode
  - $I_{Dn} = K_n W (V_{GS} - V_{th})^2 = \frac{1}{3} [W (V_{GS\_n} - 1)^2]$
  - $g_{mn} = 2K_n W (V_{GS} - V_{th}) = \frac{2}{3} [W (V_{GS\_n} - 1)]$

For example,  $V_{GS} = 3$  and  $V_{DS} = 1 \Rightarrow (3 - 1) > 1$ , which means that you need to use  $I_D$  or  $g_m$  equation in triode mode. Assume  $W = 7$ , and thus, you would get

- $I_{Dn} = \frac{1}{3} \{W [2(V_{GS\_n} - 1)V_{DS\_n} - V_{DS\_n}^2]\} = \frac{1}{3} * 7 * [2(3 - 1) * 1 - 1^2] = 7$
- $g_{mn} = \frac{2}{3} [W V_{DS\_n}] = \frac{2}{3} [7 * 1] = 4$
- **Round-down the answer if it is not integer**

Another example,  $V_{GS} = 3$  and  $V_{DS} = 5 \Rightarrow (3 - 1) \leq 5$ , which means that you need to use  $I_D$  or  $g_m$  equation in saturation mode. Assume  $W = 7$ , and thus, you would get

- $I_{Dn} = \frac{1}{3}[W(V_{GS\_n} - 1)^2] = \frac{1}{3}[7 * (3 - 1)^2] = 9$
- $g_{m_n} = \frac{2}{3}[W(V_{GS\_n} - 1)] = \frac{2}{3}[7 * (3 - 1)] = 9$
- Round-down the answer if it is not integer

### ➤ Sort

After calculation, you would have  $I_{D_0} \sim I_{D_5}$  or  $g_{m_0} \sim g_{m_5}$ . To find the maximum and minimum total current or transconductance, now you need to preprocess your result according to **mode[0]** code. If **mode[0] == 1'b1**, you would sort the sequence  $I_{D_0} \sim I_{D_5}$  as new sequence  $n_0 \sim n_5$ . As for **mode[0] == 1'b0**, you would sort the sequence  $g_{m_0} \sim g_{m_5}$  as new sequence  $n_0 \sim n_5$ . Notice that  $n_0$  is the largest one, and  $n_5$  is the smallest one.

For instance, **mode[0] == 1'b1**, now your own  $\{I_{D_0}, \dots, I_{D_5}\} = \{25, 33, 27, 5, 6, 0\}$ , then after sorting you would get  $\{n_0, \dots, n_5\} = \{33, 27, 25, 6, 5, 0\}$

Another example, **mode[0] == 1'b0**, now your own  $\{g_{m_0}, \dots, g_{m_5}\} = \{11, 5, 27, 10, 3, 1\}$ , then after sorting you would get  $\{n_0, \dots, n_5\} = \{27, 11, 10, 5, 3, 1\}$

### ➤ Calculate & Output

- **mode[0] = 1 (Weighted Average)**
  - Larger:  $I_{avg} = (3 * n_0 + 4 * n_1 + 5 * n_2)/12$  (**mode[1] = 1**)
  - Smaller:  $I_{avg} = (3 * n_3 + 4 * n_4 + 5 * n_5)/12$  (**mode[1] = 0**)
- **mode[0] = 0 (Average)**
  - Larger:  $gm_{avg} = (n_0 + n_1 + n_2)/3$  (**mode[1] = 1**)
  - Smaller:  $gm_{avg} = (n_3 + n_4 + n_5)/3$  (**mode[1] = 0**)

➤ Round-down the answer if it is not integer

Example 1:

$$mode = 2'b11, \{n_0, \dots, n_5\} = \{33, 27, 25, 6, 5, 0\}$$

$$out\_n = I_{avg} = (3 * 33 + 4 * 27 + 5 * 25)/12 = 8'd27$$

Example 2:

$$mode = 2'b00, \{n_0, \dots, n_5\} = \{27, 11, 10, 5, 3, 1\}$$

$$out\_n = gm_{avg} = (5 + 3 + 1)/3 = 8'd3$$

The summary of the description and specifications are as followings:

<b>Input Signal</b>	<b>Bit Width</b>	<b>Description</b>															
W_0, V_GS_0, V_DS_0	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
W_1, V_GS_1, V_DS_1	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
W_2, V_GS_2, V_DS_2	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
W_3, V_GS_3, V_DS_3	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
W_4, V_GS_4, V_DS_4	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
W_5, V_GS_5, V_DS_5	3 / per signal	ranged from 1~7 <b>unsigned integer</b>															
mode	2	<p>Mode indicates different operations. The operation will be encode as following:  mode [0]: 1: current, 0: transconductance  mode [1]: 1: larger eq., 0: smaller eq.</p> <table> <tr> <th>mode</th><th colspan="2">calculation</th></tr> <tr> <td>2'b00</td><td>smaller</td><td>transconductance</td></tr> <tr> <td>2'b01</td><td>smaller</td><td>current</td></tr> <tr> <td>2'b10</td><td>larger</td><td>transconductance</td></tr> <tr> <td>2'b11</td><td>larger</td><td>current</td></tr> </table>	mode	calculation		2'b00	smaller	transconductance	2'b01	smaller	current	2'b10	larger	transconductance	2'b11	larger	current
mode	calculation																
2'b00	smaller	transconductance															
2'b01	smaller	current															
2'b10	larger	transconductance															
2'b11	larger	current															

<b>Output Signal</b>	<b>Bit Width</b>	<b>Description</b>
out_n	8	The answer. Ranged from 0~256

## Inputs

---

1. The input signals W\_n, V\_GS\_n, V\_DS\_n, for n = 0~5, are 3-bit inputs
2. The input signal mode is a 2-bit input indicates whether to do the operations and which equation to use to get the final result.

## Outputs

---

The output signal **out\_n** is a unsigned number ranged from **0~256**. This represents the correct result.

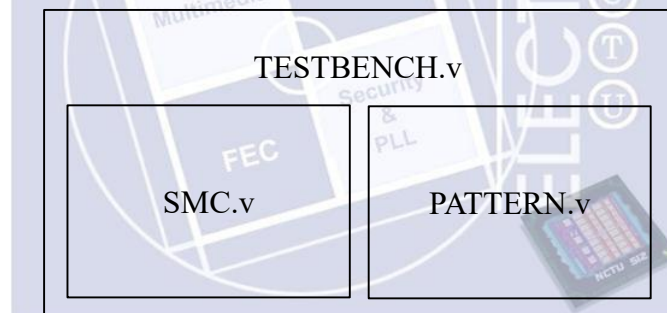
## Specifications

---

1. Top module name : SMC(File name: SMC.v)
2. After synthesis, check the “SMC.area” and “SMC.timing” in the folder “Report”.  
**The area report is valid only when the slack in the end of “SMC.timing” is “MET”.**
3. The synthesis result **cannot** contain any **latch**.  
**Note:** You can check if there is a latch by searching the keyword “**Latch**” in 02\_SYN/syn.log

## Block Diagram

---



## Grading Policy

---

The performance is determined by the area of your design. The less area your design has, the higher grade you get. Try to reach better performance by thinking your architecture before coding.

Function Validity: 70%

Performance: area 30%

If you fail Lab01 at first demo, and pass at second demo, you will get 30% off of your original score. Get no score if you fail both first and second demo. Note that you will get **0 score** if you are **found plagiarism** at your code.



## Note

---

1. Tar all your design by run the command **Lab01/09\_SUBMIT/00\_tar**
2. Submit your design through **Lab01/09\_SUBMIT/01\_submit**
  - a. 1st\_demo deadline: **2023/09/25(Mon.) 12:00:00**
  - b. 2nd\_demo deadline: **2023/09/27(Wed.) 12:00:00**
3. If your file **violates the naming rule**, you will **lose 5 points**.
4. Don't use any wire/reg/submodule/parameter name called **\*error\***, **\*Congratulations\***, **\*latch\*** or **\*FAIL\*** otherwise you will fail the lab. Note: \* means any char in front of or behind the word. e.g: error\_note is forbidden.

Be careful about all details!

Template folders and reference commands:

In demo, the reference commands is:

1. 01\_RTL (RTL simulation):  
**./01\_run\_vcs\_rtl**
2. 02\_SYN/ (Synthesis):  
**./01\_run\_dc\_shell**  
(Check **latch** by searching the keyword "**Latch**" in 02\_SYN/syn.log)  
(Check the design's timing in /Report/ SMC.timing)  
(Check the design's area in /Report/ SMC.area)
3. 03\_GATE/ (Gate-level simulation):  
**./01\_run\_vcs\_gate**
4. **09\_SUBMIT/ (submit your files):**  
**./00\_tar**  
**./01\_submit**  
**./02\_check**

You can key in **./09\_clean\_up** to clear all log files and dump files in each folder

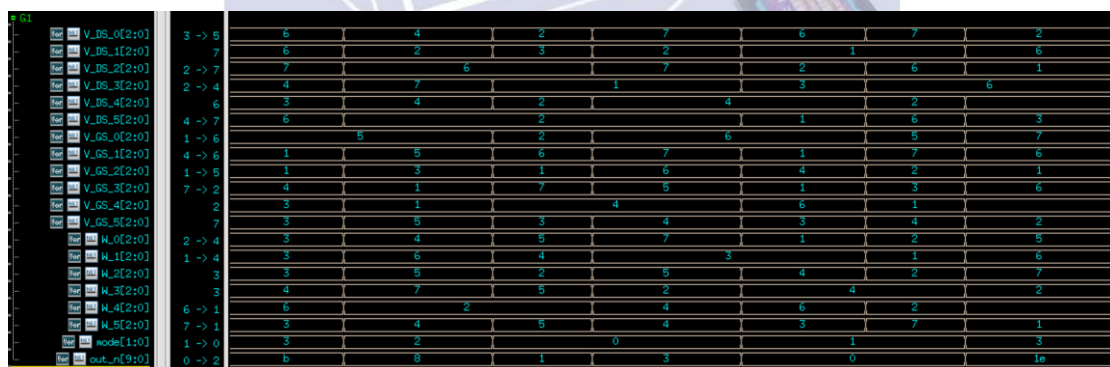
```

-- 00_TESTBED
|-- PATTERN.v
|-- TESTBED.v
|-- Test_data_gen_ref.py
|-- filelist.f
|-- input.txt
|-- makefile
|-- output.txt
-- 01_RTL
|-- 01_run_vcs_rtl
|-- 02_irun_rtl
|-- 03_xrun_rtl
|-- 04_verdi
|-- 05_nWave
|-- 08_check
|-- 09_clean_up
|-- PATTERN.v → ../00_TESTBED/PATTERN.v
|-- SMC.v
|-- TESTBED.v → ../00_TESTBED/TESTBED.v
|-- filelist.f → ../00_TESTBED/filelist.f
|-- makefile → ../00_TESTBED/makefile
-- 02_SYN
|-- 01_run_dc_shell
|-- 02_run_design_vision
|-- 03_read_dv_rtl
|-- 04_read_ddc
|-- 08_check
|-- 09_clean_up
|-- Netlist
|-- Report
|-- makefile → ../00_TESTBED/makefile
|-- syn.tcl
-- 03_GATE
|-- 01_run_vcs_gate
|-- 02_irun_gate
|-- 03_xrun_gate
|-- 04_verdi
|-- 05_nWave
|-- 08_check
|-- 09_clean_up
|-- PATTERN.v → ../00_TESTBED/PATTERN.v
|-- SMC_SYN.sdf → ../02_SYN/Netlist/SMC_SYN.sdf
|-- SMC_SYN.v → ../02_SYN/Netlist/SMC_SYN.v
|-- TESTBED.v → ../00_TESTBED/TESTBED.v
|-- filelist.f → ../00_TESTBED/filelist.f
|-- makefile → ../00_TESTBED/makefile
-- 09_SUBMIT
|-- 00_tar → /RAID2/COURSE/BackUp/2023_Spring/iclab/iclabta01/AutoSubmitScript/Lab01/09_SUBMIT/00_tar
|-- 01_submit → /RAID2/COURSE/BackUp/2023_Spring/iclab/iclabta01/AutoSubmitScript/Lab01/09_SUBMIT/01_submit
|-- 02_check → /RAID2/COURSE/BackUp/2023_Spring/iclab/iclabta01/AutoSubmitScript/Lab01/09_SUBMIT/02_check

```

## Example Waveform

Input and output signal:





## Hint

---

**Hint1:** Try to use **behavior modeling description** instead of gate level description.

**Hint2:** Try to use **submodule** rather than copy and paste to simplify your design. (not necessary in this lab)

```
// -----  
// Example for using submodule  
// BBQ bbq0(.meat(meat_0), .vagetable(vagetable_0), .water(water_0),.cost(cost[0]));  
// -----
```

**Hint3:** Try to think if there is any possible **hardware** that can be **shared** with different mode operation. You can use command dc\_shell-gui to examine your design.(not necessary in this lab)

**Hint4:** Pattern provided by TA will cover **only some simple cases**, you can try to write **your own input / output file** by yourself. Here is the format how TA will read in PATTERN:

```
/* input.txt format  
1. [PATTERN_NUM]  
  
repeat(PATTERN_NUM)  
    1. [mode]  
    2. [W_0 V_GS_0 V_DS_0]  
    3. [W_1 V_GS_1 V_DS_1]  
    4. [W_2 V_GS_2 V_DS_2]  
    5. [W_3 V_GS_3 V_DS_3]  
    6. [W_4 V_GS_4 V_DS_4]  
    7. [W_5 V_GS_5 V_DS_5]  
*/  
  
/* output.txt format  
1. [out_n]
```

You can check input.txt and PATTERN.v in 00\_TESTBED as a reference, and choose to write either c++/python or Verilog code for generating corner cases.