# Digital IC Design

## Exercise 2
## Combinational circuits

**Professor Po-Tsang Huang**

**International College of Semiconductor Technology**
**National Yang Ming Chiao Tung University**

# 2-1: 1-bit Full Adder standard cell [20%]

- Implement 1-bit FA by XOR and MAJ gates in the cell library (asapt_75t_R_24.cdl)

- Settings:
  - ◆ Inputs: A, B, Cin
    - ➢ The driving ability of inputs is one unit size of INV.
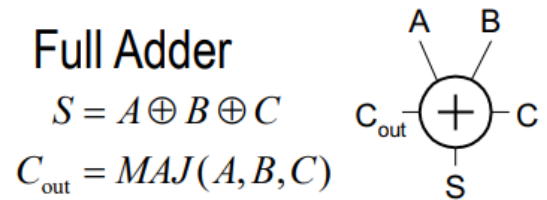    - ➢ Specify all-case switching patterns
  - ◆ Outputs
    - ➢ Output: carry_out, sum
    - ➢ The output loading of each output is **5fF.**
  - ◆ Using 7nm FinFET model.

- Measure the power, delay of a 1-bit FA
  - ◆ Delay: worst case delay

**Full Adder**

$S = A \oplus B \oplus C$

$C_{out} = MAJ(A, B, C)$



| A | B | C | $C_{out}$ | S |
|---|---|---|-----------|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# 2-2: Different logic family of 1-bit Full Adders [30%]

- Analyze and compare the power, performance and area (PPA) of 1-bit FA by different logic families
  - ◆ CMOS logic family
  - ◆ CPL-type logic family
  - ◆ DCVS-type logic family
- Settings:
  - ◆ Inputs
    - ➢ A, B, Cin (Specify all-case switching patterns)
  - ◆ Outputs
    - ➢ Output: carry_out, sum
    - ➢ The output loading of each output is **5fF.**
- PPA measurement: transistor counts for area
- **Compare and analyze these three logic families**

# 2-3: Design a 4-Bit Adder [30%]

- Design a 4-bit adder in Verilog
  - ◆ Input: a[3:0], b[3:0]
  - ◆ Output: {carry_out, sum[3:0]}
  - ◆ The sampling rate of inputs is 5GHz
- Synthesis using ASAP 7nm Technology
  - ◆ **Find out the critical paths**
- Verify your design in gate-level simulation
  - ◆ Used the pattern provided by TA

# 2-4: CMOS Logics for 4-Bit Adder [20%]

- Covert the Verilog gate-level netlist to HSPICE netlist
  - ◆ Adder_4bit_SYN.v to Adder_4bit_SYN.sp.

- Measure the delay and power of the 4-bit adder in HSPICE
  - ◆ The tested patterns (vectors) should cover the worst-case delay and power consumption.
  - ◆ **Please indicate which pattern covered the worst-case delay.**

# Vector for HSPICE

- Use high level language (C, python, script) to generate vector files for HSPICE

```
radix    1 1 1 1 1 44 44444444444444444444444444444444444
vname clk reset wen_in ren_in cen_in wl_in[[7:0]] data_in[[127:0]]
io      i i i i i ii iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

tunit ns
slope  0.1
tdelay 0.1
vih 1.0
vil 0

0.00         1 1 0 0 0 00 00000000000000000000000000000000
1.20         1 0 0 0 0 00 00000000000000000000000000000000
2.00         0 0 0 0 0 00 00000000000000000000000000000000
4.00         1 0 0 0 0 00 00000000000000000000000000000000
6.00         0 0 1 0 0 10 ffe6dba4adae6b963e23751c19c40236
8.00         1 0 1 0 0 10 ffe6dba4adae6b963e23751c19c40236
10.00        0 0 1 0 0 20 9afde6d8e5c528b5e9dc22300da2fe6a
```

**.vec test.vec**

**.vec test.txt**

6

# Submission on e3 platform

- Please compress your **report** & **source codes in a single compressed file (.zip)** and upload this single file on E3 platform

- Due date: 10/30 PM 23:55