BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

FALL 2019

PROF. IRIS BAHAR

SEPTEMBER 30, 2019

LECTURE 8: CMOS GATES & INTRODUCTION TO VERILOG

CLARIFICATION ABOUT DEADLINES

- Labs are due in groups:
  - Group 1: labs 0-3 due by **Sunday, Oct. 6 (lab 1 & 2 both required)**
  - Group 2: labs 4-9 due by **Sunday, Nov. 17 (lab 9, and either 7 or 8 required)**
  - Group 3: labs A and B due by **Friday, Dec. 6**
  - Group 4: labs C and D due by **Thursday, Dec. 13 (one of lab B, C, D required)**
- The last week before the final exam can be used as a grace period:
  - *Up to one additional lab from group 1-3 can be checked off between Dec. 7-13*
  - Labs may be checked off earlier, but if completed after the group deadline, they will count as being checked off during exam week
  - *Only one lab a day credited in exam week*
- **Plan ahead! Don't wait until exam week to finish all your labs!**

IBM STUDENT DESIGN WORKSHOP

WHEN: October 12, 2019

TIME: 9:00am-5:00pm

- **Open to all students, but geared mostly toward engineering students in their junior year.**
- **IBM facilitators will guide students through design for tackling a current pressing problem (e.g., within the or for a course)**

- Advanced registration is required. Please reserve your spot by *Thursday, Oct. 3* at https://forms.gle/ynUg4rixsJbTy8Qj6
- Email Elizabeth Austin or Jennifer Casasanto for additional questions.

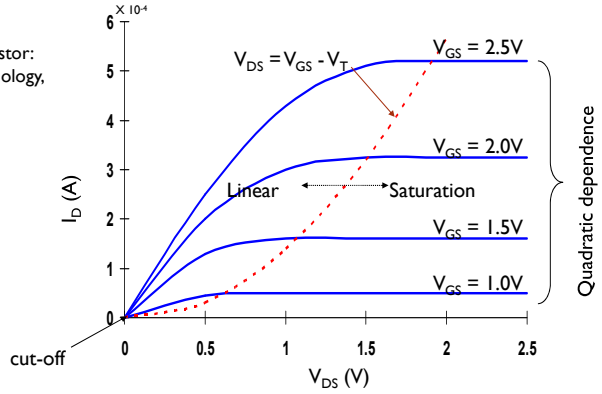
NMOS I-V SUMMARY

- *Shockley* 1st order transistor models

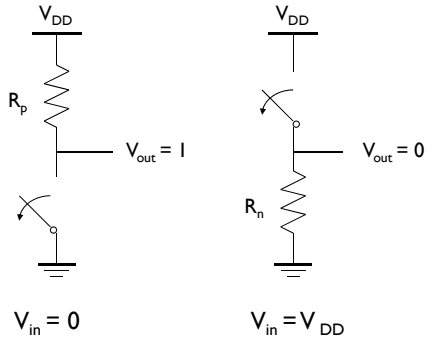
$$I_{ds} = \begin{cases} 0 & V_{gs} < V_{th} \quad \text{cutoff} \\ \beta\left(V_{gs} - V_{th} - {}^{V_{ds}}/_{2}\right)V_{ds} & V_{ds} < V_{dsat} \quad \text{linear} \\ \frac{\beta}{2}\left(V_{gs} - V_{th}\right)^2 & V_{ds} > V_{dsat} \quad \text{saturation} \end{cases}$$

LONG CHANNEL I-V PLOT (NMOS)

NMOS transistor:
0.25um technology,
L_d = 10um,
W/L = 1.5,
V_{DD} = 2.5V,
V_T = 0.4V



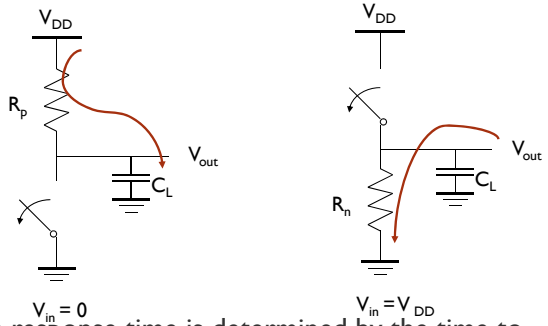
CMOS INVERTER: STEADY STATE RESPONSE



CMOS PROPERTIES

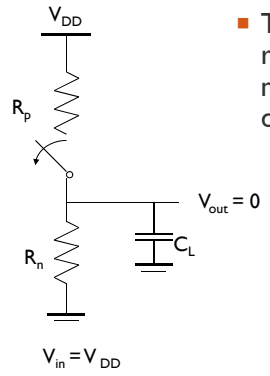
- Full rail-to-rail swing ➔ high noise margins
  - Logic levels independent of device sizes ➔ *ratioless*
- Always a path to V_{dd} or GND in steady state ➔ less sensitive to noise
- nearly zero steady-state input current
- No direct path steady-state between power and ground
  - no static power dissipation
- Propagation delay is a function of load capacitance and resistance of transistors

CMOS INVERTER:
SWITCH MODEL OF DYNAMIC BEHAVIOR



- Gate response time is determined by the time to charge C_L through R_p (discharge C_L through R_n)

CMOS INVERTER: DYNAMIC BEHAVIOR

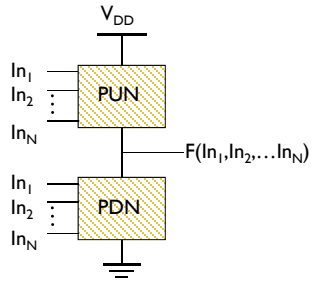


- Transient, or dynamic, response determines the maximum speed at which a device can be operated.

$t_{pHL} = f(R_n, C_L)$

STATIC COMPLEMENTARY CMOS

- Pull-up network (PUM) and pull-down network (PDN)

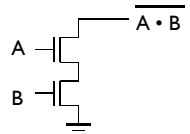


- PMOS transistors only
- pull-up: make a connection from V_{DD} to F when $F(In_1,In_2,\ldots In_N) = 1$
- NMOS transistors only
- pull-down: make a connection from F to GND when $F(In_1,In_2,\ldots In_N) = 0$

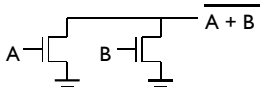
PUN and PDN are dual logic networks

CONSTRUCTION OF PDN

- NMOS devices in series implement a NAND function



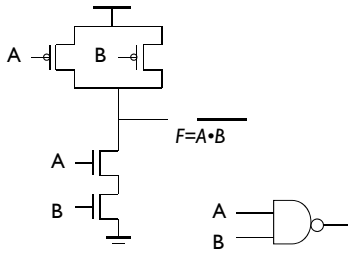
- NMOS devices in parallel implement a NOR function



DUAL PUN AND PDN

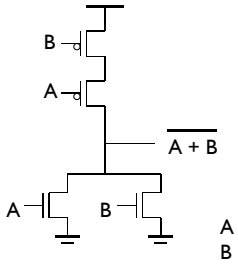
- PUN and PDN are dual networks
  - DeMorgan's theorems
    - $\overline{A + B} = \overline{A} \cdot \overline{B}$ [!(A + B) = !A • !B or !(A | B) = !A & !B]
    - $\overline{A \cdot B} = \overline{A} + \overline{B}$ [!(A • B) = !A + !B or !(A & B) = !A | !B]
  - a parallel connection of transistors in the PUN corresponds to a series connection of the PDN
- Complementary gate is naturally inverting (NAND, NOR, AOI, OAI)
- Number of transistors for an N-input logic gate is 2N

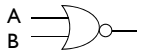
CMOS NAND



A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

CMOS NOR





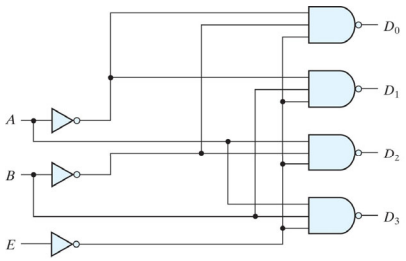
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

PULL UP RESISTORS AND TRISTATE DRIVERS

- Lab 2 discusses use of pull-up resistors and tristate drivers
- Why use them?

INTRODUCTION TO THE VERILOG HARDWARE DESCRIPTION LANGUAGE

2:4 DECODER WITH ENABLE



(a) Logic diagram

- Complemented enable input
- 0-hot output encoding

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

2:4 DECODER: STRUCTURAL

```
// Gate-level (structural) description of 2-to-4 decoder
module decoder_2x4_gates(D, A, B, enable_);
  output      [3:0] D;
  input       A, B, enable_;
  wire     A_not, B_not, enable_not;

  not G1(A_not, A);
  not G2(B_not, B);
  not G3(enable_not, enable_);
  nand G4(D[0], A_not, B_not, enable_not);
  nand G5(D[1], A_not, B, enable_not);
  nand G6(D[2], A, B_not, enable_not);
  nand G7(D[3], A, B, enable_not);
endmodule
```

multi-bit output

2:4 DECODER: BEHAVIORAL (DATAFLOW)

```
// Behavioral (dataflow) description of 2-to-4 decoder
module decoder_2x4_df(D, A, B, enable_);
  output      [3:0] D;
  input       A, B, enable_;

  assign D[0] = ~((~A) & (~B) & (~enable_));
  assign D[1] = ~((~A) & B & (~enable_));
  assign D[2] = ~(A & (~B) & (~enable_));
  assign D[3] = ~(A & B & (~enable_));

endmodule
```

left hand side must be a wire (or output)

2:4 DECODER: BEHAVIORAL 2 (DATAFLOW)

```
// Behavioral description of 2-to-4 decoder
//  inputs A and B replaced with I[1:0]
module decoder_2x4_beh2(D, I, enable_);
  output      [3:0] D;
  input       [1:0] I;
  input       enable_;

  wire     [3:0] Di;

  assign Di = (I == 2'b11) ? 4'b0111:
              (I == 2'b10) ? 4'b1011:
              (I == 2'b01) ? 4'b1101: 4'b1110;

  assign D = (enable_ == 1'b0) ? Di : 4'b1111;

endmodule
```

conditional continuous assignment statement