COMP 311 Exam 1

Gates, expressions, circuits, & analysis

Sample question 1

Clock Design

Properties

- **Clock Period (T)** → one complete cycle, generally 1ns
- **Clock Frequency (F)** → how many cycles in one sec,
  - $F = \frac{1}{T}$, measured in Hertz (Hz)
- Events, thinking the part going up and down
  - Rising-edge, logic 0 to logic 1
  - Falling-edge, logic 1 to logic 0
- Active high and low, think the stable parts once up or down
  - Active High, signal is 1
  - Active Low, signal is 0

Oscillators

Ring oscillator

- Frequency $= 1/(2 * \#$ of inverters $* t_d)$
- t_d is the propagation delay for 1 inverter

Quartz Crystal

- Very thin piece of cut quartz
- Two parallel metallic surfaces for electrical connection
- Crystal physical size and thickness determines frequency

CMOS gate design & analysis

Transistors

Conceptually similar to a DC voltage switch

- (+) Voltage can be many values such as 1.5 VDC or 9 VDC
- (-) Ground can be GND or 0 VDC

Metal Oxide Semiconductor (MOS) Transistor

- **nMOS** → "n-channel" switch is open when GND, aka need voltage to "turn on"
- **pMOS** → "p-channel" switch is open when positive VDC value, aka voltage needs to be "turn off", this one has the inverter before it

Continuous to Discrete Conversion

- Logic 1 → Voltage range from 5 to 2 VDC
- Logic 0 → Voltage range from 0 to 0.8 VDC
- Invalid → Less than 2 VDC, Greater than 0.8 VDC, this is unstable and unreliable

Gate Delay

- There exists gate delay, where we can assume that the time delay is the same as the time it takes to go from low to high voltage and high to low voltage
- t_d → propagation delay

Common component & analysis

Multiplexer & Demultiplexer

S bits to "select" which inputs (A or B) becomes the output

- For the Multiplexer **MUX**
  - If S = 1, then Y = B
  - If S = 0, then Y = A
  - # of inputs $= 2^{\#\text{ of select bits}}$
  - For circuit, think of the two ANDs into 1 OR
- For the Demultiplexer **DeMUX**
  - If S = 1, then Y = A
  - If S = 0, then Y = B
  - # select bits $= \log_2(\#$ output bits$)$

Decoder & Encoder

- **Decoder**, where S input bits are used to "select" which outputs (A) are turned on (logic 1)
  - Usually 2-to-4 as we need two bits to dictate which of the 4 will be on
  - # of outputs $= 2^{\#\text{ of select bits}}$
  - Example: $S_2S_1 = 00 \rightarrow A_1 = 1$ $A_2, A_3, A_4 = 0$
  - Think the 4 ANDs that need that specific case in order to be one
- **Encoder**, where input bits (A) are used to select which output bits (S) are turned on (logic 1)
  - Usually 4-to-2, as we need 4 bits to dictate the output of 2
  - # of outputs $= \log_2(\#$ of select bits$)$
  - Example: $A_4A_3A_2A_1 = 0001_2 \rightarrow S_2S_1 = 00_2$
  - Think the two ORs

ALU component design and analysis

Bitshifts Reminder

- **Left Shift**: Shifts in a 0 from the right end.
  - if $X = 01010$ then $X << 1$ is 10100
- **Logic Right Shift**: Shifts 0 from left end.
  - if $X = 01010$ then $X >> 1$ is 00101
- **Arithmetic Right Shift**: maintains sign bit.
  - if $X = 11010$ then $X >>> 1$ is 11101

Binary Addition and Subtraction

Binary Addition:

- Not a single operation → $A+B =$ Sum & Carry-out
- Each are one bit binary values

Binary Half Adder vs. Full Adder

- **Binary Half Adder** has 2 input, A and B, and 2 output, C and S.
- **Binary Full Adder** has 3 input, C_{in}, A, and B, and 2 outputs C_o and S
- **8-bit Full Adder Circuit** → needs total delay $= t_d *$ number of bits
- **8-bit Full Add and Subtract Circuit** → needs total delay $= t_d * ($number of bits $- 1)$

Logic gates & boolean algebra

Stateless Digital Logic Circuit

- Combination Logic → combination of logic gates
- Change input values → immediate changes in output values
- No memory
- No Feedback

Karnaugh Maps

Rules:

- Every Rectangle has to be as Large as possible
- Goes $00, 01, 11, 10$ per row/col

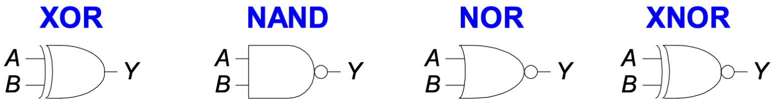
Don't Cares

Rules

- X is treated as a 0 or 1
- Circle an X only if it helps us cover 1's with larger rectangle
- We don't care about X if it's not helpful for covering 1's

Basic Identities of Boolean Algebra

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			
10.	$X + Y = Y + X$	11.	$XY = YX$	Commutative
12.	$X + (Y + Z) = (X + Y) + Z$	13.	$X(YZ) = (XY)Z$	Associative
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's



$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$Y = \overline{A + B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$Y = \overline{A \oplus B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

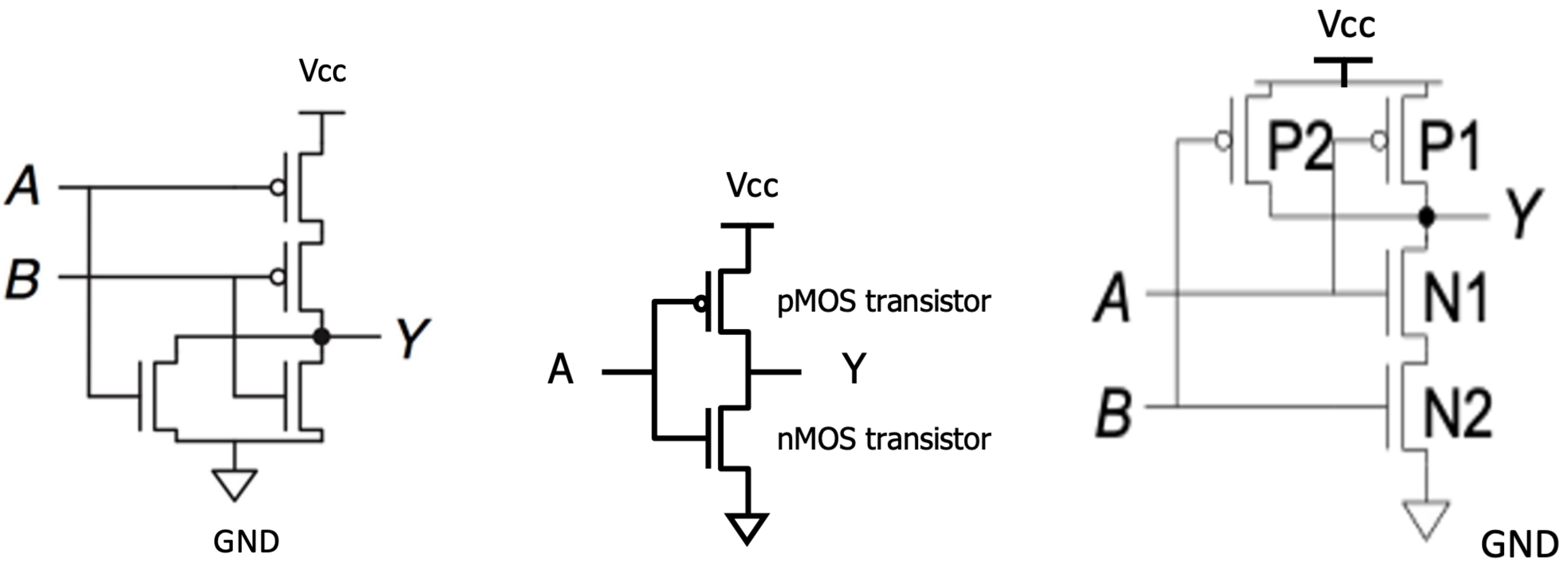


Figure 1: These are the nor, not, and nand gates respectively