COMP 311 Exam 2

ALU Component Design

$\gg$ is Logical Right Shift
$\ggg$ is Arithmetic Right Shift

Operations that only take 1 cycle:
AND, OR, XOR, NOT, NOR

Clock Period is the time it takes for one cycle
$f = \frac{1}{T}$, Where T is the period

Latches & Flip-Flops

Latches are essentially 2-to-1 MUX w/ Feedback
Reminder: NO CLOCK!!

- When G = 0, it is opaque, meaning that it is stable
- When G = 1, it is transparent, meaning that it is unstable

In other words, when G is transparent, then G follows D. And when G is opaque, D will be stable.

Flip-flops

With Flip-flops, there are reliable timing with clocks

D Flip-flop

D flip-flops is just one neg. connected to one pos.
It holds the value of D when there is rising edge until the next rising edge

If used in State Machine Truth Table, it is 1 when D is 1, refer to table.
Moreover, we can simply look at the current bits and the next bits and figure out what D input is.

T Flip-flop

Toggle flip-flop
If used in State Machine Truth Table, it is 1 when the bit switches

J-K Flip-flop

Set (J) and Reset (K) flip flop, having T and D capabilities

Memory/Register Design

N-bit registers need N flip-flops, generally using Rising-edge enabled D flip-flops
Each clock cycle, ALL the flip-flops will be updated

Clock events applies only to registers and NOT ALUs

State machines

States

States = $2^{\#\text{ of bits per state}}$

Transitions and Bits

One transition bit = two possible transitions
# of transitions = $2^{\#\text{ of transition bits}}$ ALL transitions must be defined

State Truth Tables

- Convert Diagram to Truth Table
- Select flip-flop storage Component
- Define flip-flop inputs

Miscellaneous

Two's Complement

Negative Number Steps in 2's Complement

1. Invert the Digits
2. Add 1
3. Then add two values together if needed

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

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
- **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

T
Q
CLK
J
K
Q
CLK

T	Q_p	Q
0	0	0
0	1	1
1	0	1
1	1	0

Rising-Edge Enabled

T=0, Q stays the same

T=1, Q toggles (0->1 or 1->0)

J	K	Q_p	Q
0	x	0	0
1	x	0	1
x	1	1	0
x	0	1	1

Rising-Edge Enabled

J=1, K=0, Q=1 (set)

K=1, J=0, Q=0 (reset)

J=K=1, Q toggles

J=K=0, Q stays the same

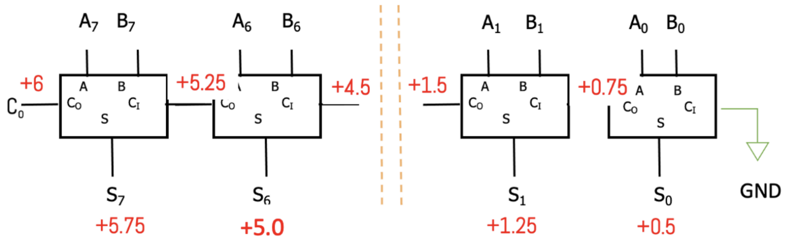
D	Q_p	Q
0	0	0
0	1	0
1	0	1
1	1	1

Figure 1: T Flip-flop, J K Flip-flop, and D flip-flop truth tables respectively

8-bit Full Add Circuit

If the clock period is 1 ns, then how many clock cycles (worst case) are needed?

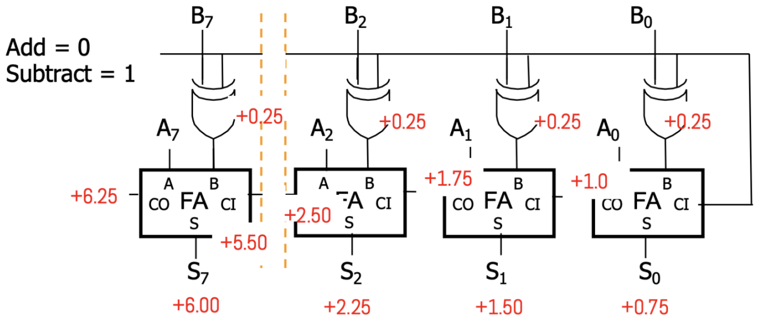
- Carry out delay (t_d) = 0.75 ns
- Total delay = t_d x number of bits = 0.75 x 8 = 6 ns (**6 clock cycles are needed**).



8-bit Full Add and Subtract Circuit

If the clock period is 1 ns, then how many clock cycles (worst case) are needed?

- LSB (C_0) carry-out delay (t^0_d) = 1.0 ns
- FA carry-out delay (t_d) = 0.75 ns
- Total delay = $t^0_d + t_d$ x (number of bits – 1) = 1.0 + 0.75 x 7 = 6.25 ns (**7 clock cycles are needed**)


Add = 0
Subtract = 1
B_7
B_2
B_1
B_0
A_7
A_2
A_1
A_0
+0.25
+0.25
+0.25
+0.25
+6.25
+1.75
+1.0
A B
CO FA CI
S
+5.50
+2.50
S_7
S_2
S_1
S_0
+6.00
+2.25
+1.50
+0.75