How Circuit Elements are Implemented

- CMOS logic gates
- Where circuit delays come from
- Implementation of latches and flip flops

CMOS Logic Gates

- CMOS integrated circuits are built using two types of Field Effect Transistors (FET), n-type & p-type
- Logic gates are constructed by combining transistors in complementary arrangements

Circuit Delays in CMOS Circuits

- Gate inputs and interconnecting wires act as capacitances
- Driving gate must charge/discharge this capacitance to change voltage level
- Response to instantaneous change at X is gradual change in voltage at Y
- Voltage at Y must pass logic threshold level before change is "seen"
- Effect can be viewed as delay in propagation of logic values through gates
- Delay increases with fanout and wire length
- Every gate input adds capacitance, increasing delay
- Capacitance of interconnecting wires increases with length

Closer Look at CMOS Circuit Delays

- When X goes high, pull-up turns off and pull-down turns on
- Decrease of voltage at Y requires transfer of charge from capacitor to ground
- Time for transfer depends on size of capacitance and on-resistance of transistors
- With large fanout must charge/discharge larger capacitance, hence slower
- NAND2 has 2 pull-downs in series, hence twice the resistance and twice the delay
- NAND3 has three times the delay

From AND/OR to NAND/NOR

- Right-hand circuit requires 10 fewer transistors (50 vs. 60) and has 20% smaller delay
Gate Delays & Circuit Performance

- Circuit performance determined by sum of delays on input-to-output paths
  - U to F path has 2 gates
  - V to F path has 3 gates
  - X to F path has 2 gates
  - X to G path has 3 gates
  - Y to F path has 4 gates
  - V to G path has 3 gates
  - Z to G path has 3 gates

- so, worst-case is 4 gates – can easily reduce to 3
- more careful analysis will account for varying delays of different types of gates, effect of fanout and wire length

Review Questions

1. Show that the two gates shown below are equivalent

[Diagram of two gates]

1. In CMOS circuits, n-FETs have a smaller resistance when they are turned on than similar p-FETs. Explain how this affects the rise and fall times of a CMOS inverter. Which has the largest overall propagation delay, \( t_{PD} \), a NAND gate, or a NOR gate?

2. Find the maximum delay path for the circuit on the right side of page 6, assuming all gates have the same delay. Now find the maximum delay path assuming that an inverter has a delay of 1, a NAND gate has a delay of 2, a NOR gate a delay 3 and an XOR gate a delay of 5.

Question

1. Assume NOT gates have 1 ns delay and AND and OR gates have 2 ns delay. What is the maximum delay through the circuit below?

[A circuit diagram]

- A. 4 ns
- B. 5 ns
- C. 6 ns
- D. 7 ns
- E. 8 ns

Edge-Triggered D Flip Flop

- D flip flop stores value at D input when clock rises
- Most widely used storage element for sequential circuits
- Propagation time is time from rising clock to output change
- If input changes when clock rises, new value is uncertain
  - output may oscillate or may remain at intermediate voltage (metastability)
- Timing rules to avoid metastability
  - D input must be stable for setup time before rising clock edge
  - must remain stable for hold time following rising clock edge

The SR Latch

- Pair of inverters provides stable storage
- To enable stored value to be changed, use cross-coupled NOR gates
  - equivalent to inverter pair when both inputs are low
- SR latch is key building block for flip flops
  - when \( S=1, R=0 \) latch is set
  - when \( S=0, R=1 \) latch is reset
  - when \( S=0, R=0 \) latch retains value
  - when \( S=1, R=1 \) latch state is undefined

S-R Latch Behavior

- Note that when \( S=R=1 \), both outputs are low
  - outputs are not complements of each other in this case
- When \( S, R \) drop together, latch output is undefined
  - may remain at intermediate voltage
  - or, may oscillate between low and high values
- Latch metastability can cause unpredictable circuit behavior
- For these reasons, avoid \( S=R=1 \) condition
More on SR Latches

- SR latch most often implemented with NAND gates.
  - inputs are active low (negative logic inputs)
  - when both inputs are low, both outputs high
  - when inputs rise together, outputs can become metastable

SR latch with control input

- SR latch with control input changes state only when control input is high.
  - inputs are active high
  - forbidden input condition is \( C=S=R=1 \)
  - change \( S, R \) inputs when \( C=0 \)

NAND-based SR Latch

D Latch

- Adding inverter to SR latch with control input yields \( D \)-latch
- Stores the value on the \( D \) input when the control input is raised
  - no forbidden input combinations
  - but, input must be stable when the control input drops
  - if not, outputs may become metastable

Implementing D Flip Flops

- When clock rises, value in \( D \)-latch propagates to \( SR \)-latch and outputs
- New value determined by \( D \) input at time clock rises
- Flip flop setup and hold time conditions designed to prevent metastability in latches
- Propagation delay determined primarily by \( SR \)-latch delay

Question

1. Assume all gates have 1 ns delay. What is the maximum propagation delay from the rising edge of the clock? Don't forget to count the feedback path in the cross-coupled NANDs.

   - A. 4 ns
   - B. 5 ns
   - C. 6 ns
   - D. 7 ns
   - E. 8 ns

Question

1. Assume all gates have 1 ns delay. What is an appropriate value for the setup time? Don't forget to count the feedback path in the cross-coupled NANDs.

   - A. 4 ns
   - B. 5 ns
   - C. 6 ns
   - D. 7 ns
   - E. 8 ns

SR Master-Slave Flip Flop

- The SR master-slave flip flop uses two SR latches with complementary enables
- First stage follows all changes while clock is high, but second stage only "sees" value after the clock drops
  - not the same as a negative edge-triggered flip flop
- Forbidden input combination causes metastability
- Recommended usage: change \( S, R \) only when \( C=0 \)
Other Types of Flip Flops

- **Toggle flip flop**
  - change state when T input high

- **J-K flip flop**
  - set when J high, reset when K high, toggle when both high

\[
\begin{array}{c|c|c}
\text{T} & \text{Q(t+1)} \\
\hline
0 & 0 \\
1 & 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{J} & \text{K} & \text{Q(t+1)} \\
\hline
0 & 0 & Q(t) \\
0 & 1 & 0 \\
1 & 1 & Q(t) \\
\end{array}
\]

Exercises

1. Give an equivalent AND/OR circuit, corresponding to the NAND/NOR circuit shown below.

2. How many transistors are used by the original circuit in the previous problem? How many are used by the circuit in your solution (assuming each AND gate is implemented using a NAND followed by an inverter and each OR gate is implemented using a NOR followed by an inverter)?

3. Consider the circuit shown below is implemented directly using CMOS, with each AND gate implemented using a NAND and an inverter (similarly for OR gates), and that the multiplexor is implemented using AND and OR gates. What is the worst-case delay for the circuit, assuming that NAND and NOR gates have a delay of 3 ns each, and inverters have a delay of 0.5 ns? Highlight the path through the circuit that accounts for the worst-case delay. Show an alternate implementation, using NANDs, NORs and inverters that reduces the delay by at least 2.5 ns.

4. Define each of the following terms, with respect to a positive edge-triggered D flip flop.
   - (a) propagation delay
   - (b) setup time
   - (c) hold time

5. Complete the timing diagram for the circuit shown below.

Solutions

1. 16 transistors are needed for the original circuit. 10 are needed for the revised version.

3. The path from A through the top AND gate, the mux control input and the OR gate to X has a delay of 8 ns. This is the worst-case delay. The alternate implementation shown below has a worst-case delay of 5.5 ns (start at A, take path through the top gate, then top right inverter, then through the three gates leading to X).

4. (a) The propagation delay of a flip flop is the time from when the clock goes high until the time that the output changes.
   (b) The setup time of a flip flop is the period of time before the clock goes high during which the D input is required to be stable (not changing).
   (c) The hold time of a flip flop is the period of time after the clock goes high during which the D input is required to be stable (not changing).