Homework

• Reading
  – Tokheim Chapter 9.1 – 9.6

• Machine Projects
  – Continue on mp3

• Labs
  – Continue in labs with your assigned section
Sequential Circuits

- A *sequential* circuit is constructed using a *combinational* circuit with memory circuits
- Similar to a C function with static internal variables (state memory)
- One additional input is a clock signal
**Simple Memories (Flip-Flops)**

- Simplest is Reset-Set (R-S type)
- Note the inverted signal inputs
- Can buy a standard TTL R-S flip-flop (279)

<table>
<thead>
<tr>
<th>( \overline{S} )</th>
<th>( \overline{R} )</th>
<th>Q</th>
<th>( \overline{Q} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>no change</td>
<td>prohibited</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![NAND gate diagram]

![NAND gate diagram]
Simple Memories (Flip-Flops)

- **Timing Diagram**

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Set Hold</td>
<td>No Effect Reset Hold No Effect Hold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Synchronous Flip-Flops

- Use of a clock to make the circuit synchronous
- Syn (with) + chronous (clock) => with a clock
- Level-triggered (changes state while clock high)

TRUTH TABLE

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Clock</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td>No Chg</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>Prohibited</td>
</tr>
</tbody>
</table>
The D-Type Flip-Flop

- Single data input and edge-triggered clock
- Also called a “Delay” flip-flop (D-type)
- Changes state on either rising or falling edge

**Truth Table**

<table>
<thead>
<tr>
<th>D</th>
<th>Clock</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**NAND R-S Flip Flop**

**Edge To Pulse Converter**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>(Due to delay through inverter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Actual D-Type Flip-Flop

- Has preset (PR) and clear (CLR) inputs which can be set asynchronously (but not both at same time)
- Nomenclature use > for an edge-triggered input
Reset Circuitry

- When power is turned on, there is a time delay for power to reach each part of the circuit and to stabilize at the rated voltage.
- We need to apply a reset signal for longer than that time delay to all memories.
- Reset signal presets (to 1) or clears (to 0) every flip-flop in the system as needed.
- Reset signal is released after a time delay.
- Reset button causes reset signal to be asserted and released again after time delay.
Example Reset Circuitry

Power

R

C

Ground

Reset Button

Time Constant = R * C

V

Time

To All Flip-Flops
Timing Diagrams for D Flip-Flop

(c) The general form of the timing diagram

(d) An alternative form of the timing diagram
Clock - Divide by Two Counter

• Connect $\overline{Q}$ output back to D input

• Timing Diagram (after starting with $Q = 0$)

Q is an output clock at $\frac{1}{2}$ the frequency of the input clock
Shift Registers

• Serial in, Parallel out:

Four Bit Parallel Output Available After Four Clocks/Shifts

Serial Data In → FF3 → FF2 → FF1 → FF0

Clock In
Shift Registers

• Parallel in, Serial out:

Four Bit Parallel Input Presented for One Clock Edge while “Load” Signal is True
The J-K or Universal Flip-Flop

- Named for Jack Kilby (TI Engineer / Inventor of IC)
- Three synchronous inputs (plus preset and clear)
- J-K flip-flop is available as a 7473 chip
- Can be edge-triggered or level-triggered
- Example shown is falling “edge triggered”

TRUTH TABLE

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Clock</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td>Stays same</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>Toggles</td>
</tr>
</tbody>
</table>
J-K Flip-Flop Internals

- Avoids an invalid input such as 0 0 to the R-S Flip Flop
Using J-K Flip-Flops

- Primary use is for storage registers and counters
- Mod-16 counter also known as a ripple counter

\[ X_3 X_2 X_1 X_0 \] counts 0x0 ... 0xF (Hexadecimal) sequentially
Timing diagram for Mod-16 Counter

Note that the counter actually serves to divide down the input clock!
Can we make it count to something different than $2^N$?

Ans. Yes, using a combinational logic (a NAND Gate in this case)
Counts: 0, 1, 2, 3, 4, 5, 0, 1, ...
Synchronous BCD up/down counter

- BCD Up/Down Counter is available as a 74192
- BCD stands for Binary Coded Decimal
- Counts 0000 - 1001, then carries
Describing Sequential Circuits

• In general,
  – Next state = \( f(\text{inputs, current state}) \)
  – Outputs = \( f(\text{inputs, current state}) \)

• Example:

  State diagram:

  Truth Table

  State = (Q1, Q2) [2 bits]
  4 states: (0,0), (0,1), (1,0), (1,1)

  A | Q1 | Q2 | Last State | Next State | X
  --|----|----|------------|------------|---
  0 | 0  | 0  | 0 0        | 0 1        | 0
  1 | 0  | 0  | 1 0        | 1 0        | 0
  0 | 0  | 1  | 0 0        | 0 0        | 0
  . | .  | .  | .           | .          | .
  . | .  | .  | .           | .          | .
  . | .  | .  | .           | .          | .
  0 | 1  | 1  | 0 1        | 0 1        | 1
  1 | 1  | 1  | 1 1        | 1 1        | 1
Digital Logic Summary

• Combinational circuits:
  – Made from gates without feedback
  – Have no internal states
  – Outputs depend only on current inputs
  – Fully defined by truth table on the inputs
  – Passes clocks (if any) as wave trains
  – Output states constantly change with inputs
Digital Logic Summary

• Sequential circuits:
  – Have feedback among the gates
  – Can have internal states
  – Outputs depend on inputs and past inputs (via values of internal states)
  – Not completely described by pure truth table on inputs
  – Usually one input is a clock signal
  – Outputs usually change on one clock edge only