L4: Sequential Building Blocks
(Flip-flops, Latches and Registers)

Acknowledgements:

Materials in this lecture are courtesy of the following sources and are used with permission.


Combinational logic circuits are memoryless
No feedback in combinational logic circuits
Output assumes the function implemented by the logic network, assuming that the switching transients have settled
Outputs can have multiple logical transitions before settling to the correct value
A Sequential System

- Sequential circuits have memory (i.e., remember the past)
- The current state is “held” in memory and the next state is computed based on the current state and the current inputs
- In a synchronous systems, the clock signal orchestrates the sequence of events
A Simple Example

Adding N inputs (N-1 Adders)

Using a sequential (serial) approach

Current_Sum
Implementing State: Bi-stability

Point C is Metastable

Points A and B are stable (represent 0 & 1)
**NOR-based Set-Reset (SR) Flipflop**

Flip-flop refers to a bi-stable element (edge-triggered registers are also called flip-flops) – this circuit is not clocked and outputs change “asynchronously” with the inputs.
Making a Clocked Memory Element: Positive D-Latch

- **A Positive D-Latch**: Passes input D to output Q when CLK is high and holds state when clock is low (i.e., ignores input D)
- **A Latch is level-sensitive**: Invert clock for a negative latch
2:1 Multiplexor

\[ \text{Out} = \text{sel} \times \text{in}_1 + \overline{\text{sel}} \times \text{in}_0 \]

Positive Latch

Negative Latch

"remember" "load"

"data" "stored value"
74HC75 (Positive Latch)

Operating Modes

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{LE}_{n-n}$</td>
<td>$nD$</td>
</tr>
<tr>
<td>Data Enabled</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td>Data Latched</td>
<td>L</td>
</tr>
</tbody>
</table>

Figures by MIT OpenCourseWare.
Master-Slave Register

- Use negative clock phase to latch inputs into first latch
- Use positive clock to change outputs with second latch

View pair as one basic unit
- master-slave flip-flop twice as much logic
**Latches vs. Edge-Triggered Register**

*Edge triggered* device sample inputs on the event edge.

*Transparent latches* sample inputs as long as the clock is asserted.

**Timing Diagram:**

- **Edge triggered** register:
  - Positive edge-triggered register
  - Level-sensitive latch

- **Level-sensitive latch**:
  - Bubble here for negative edge-triggered register

Behavior the same unless input changes while the clock is high.
Important Timing Parameters

Clock:
Periodic Event, causes state of memory element to change

memory element can be updated on the: rising edge, falling edge, high level, low level

Setup Time ($T_{su}$)
Minimum time before the clocking event by which the input must be stable

Hold Time ($T_h$)
Minimum time after the clocking event during which the input must remain stable

Propagation Delay ($T_{cq}$ for an edge-triggered register and $T_{dq}$ for a latch)
Delay overhead of the memory element

There is a timing "window" around the clocking event during which the input must remain stable and unchanged in order to be recognized.
Eliminate the forbidden state of the SR Flip-flop

Use output feedback to guarantee that R and S are never both one
J-K Master-Slave Register

Is there a problem with this circuit?
Ways to design an edge-triggered sequential cell:

- **Master-Slave Latches**
- **Pulse-Based Register**

- Pulse registers are widely used in high-performance microprocessor chips (Sun Microsystems, AMD, Intel, etc.)
- The can have a negative setup time!
D Flip-Flop vs. Toggle Flip-Flop

D Flip-Flop

<table>
<thead>
<tr>
<th>D</th>
<th>Q&lt;sub&gt;N&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

T (Toggle) Flip-Flop

<table>
<thead>
<tr>
<th>T</th>
<th>Q&lt;sub&gt;N&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Q&lt;sub&gt;N-1&lt;/sub&gt;</td>
</tr>
<tr>
<td>1</td>
<td>Q&lt;sub&gt;N-1&lt;/sub&gt;</td>
</tr>
</tbody>
</table>
Realizing Different Types of Memory Elements

**Characteristic Equations**

- **D:**  \( Q^+ = D \)
- **J-K:**  \( Q^+ = J \overline{Q} + K Q \)
- **T:**  \( Q^+ = T \overline{Q} + T Q \)

E.g.,
- \( J=K=0, \) then \( Q^+ = Q \)
- \( J=1, K=0, \) then \( Q^+ = 1 \)
- \( J=0, K=1, \) then \( Q^+ = 0 \)
- \( J=1, K=1, \) then \( Q^+ = Q \)

**Implementing One FF in Terms of Another**

- **D implemented with J-K**
- **J-K implemented with D**
Excitation Tables: What are the necessary inputs to cause a particular kind of change in state?

<table>
<thead>
<tr>
<th>Q</th>
<th>Q+</th>
<th>J</th>
<th>K</th>
<th>T</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Implementing D FF with a J-K FF:

1) Start with K-map of $Q^+ = f(D, Q)$

2) Create K-maps for J and K with same inputs $(D, Q)$

3) Fill in K-maps with appropriate values for J and K to cause the same state changes as in the original K-map

E.g., $D = Q = 0$, $Q^+ = 0$ then $J = 0$, $K = X$
Implementing J-K FF with a D FF:

1) K-Map of $Q^+ = F(J, K, Q)$

2,3) Revised K-map using D's excitation table

It's the same! That is why design procedure with D FF is simple!

$$Q^+ = D = J\overline{Q} + K\overline{Q}$$

Resulting equation is the combinational logic input to D to cause same behavior as J-K FF. Of course it is identical to the characteristic equation for a J-K FF.
System Timing Parameters

Register Timing Parameters

- $T_{cq}$: worst case rising edge clock to $q$ delay
- $T_{cq, cd}$: contamination or minimum delay from clock to $q$

Logic Timing Parameters

- $T_{logic}$: worst case delay through the combinational logic network
- $T_{logic, cd}$: contamination or minimum delay through logic network

Tsu: setup time
Th: hold time
System Timing (I): Minimum Period

\[ T > T_{cq} + T_{\text{logic}} + T_{su} \]
System Timing (II): Minimum Delay

Combinational Logic

\[ T_{eq,cd} + T_{\text{logic,cd}} > T_{\text{hold}} \]
Typical parameters for Positive edge-triggered D Register

all measurements are made from the clocking event that is, the rising edge of the clock

Shift-register

Shift-Register