Modern Computer Architecture

Lecture 4
Sequential Logic, Memories
Outline – Lecture 3

• State Machines
• Memory Types
  – SRAM
  – flash
  – DRAM
- stores 1 bit of information
- symbol: master-slave flip-flop

edge-sensitive

level-sensitive
• two types: SR-NOR latch and SR-NAND latch

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>Q'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Transparent Latch

- enable signal: **level sensitive**
- **transparent** if enable signal is 1
- enable: write-strobe

<table>
<thead>
<tr>
<th>D</th>
<th>E</th>
<th>Q</th>
<th>Q'</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
</tbody>
</table>
Master-Slave Flip-Flop

- clock signal: **edge-triggered**
- idea: chain two latches

![Diagram of master-slave flip-flop](image)

(negative edge-triggered flip-flop)

positive edge-triggered flip-flop from NAND gates
Flip-Flop Timing

- setup time: $t_{su}$
- hold time: $t_h$
- propagation delay: $t_{CO}$ (clock to output)

- determines clock speed
State Machines a.k.a. Synchronous Sequential Logic

- Input
- $f_T$ transfer function (combination circuit)
- $Z(k)$ state (flip flops)
- Output
- Current state
- New state
- Clock
• Moore Machine

• Mealy Machine
• One wire: one bit
• Multiple wires: one word (typically: 32, 64 bits)
• Used to be 8 bit: one byte

• Memory: input an address (number of word), stores a word of data
• Operations:
  – Read input address, output: data
  – Write input (address, data), output: none

• Even though accesses are of word size granularity, each byte in memory gets an individual address (byte-addressed machine). Thus the memory layout is independent of processor implementation. E.g. x86 16-bit → 32-bit → 64-bit
Random Access Memory (RAM)

- Address (bus)
- Data (bus)
  - read: output
  - write: input
- Control (Read/Write Mode)
- Bus: Word width
  - number of bits in parallel
Byte Addressing vs. Word Addressing

- **Address**
- **Data**
  - read: output
  - write: input
- **Control Read/Write Mode**
- **Word width**
  - number of bits in parallel
- **Byte adress**
  - word address
  - byte selection

<table>
<thead>
<tr>
<th>31</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>word address</td>
<td>byte sel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram:
- Memory with address/data flow
- 32-bit data
- 8-bit data
- Address flow

Dr. Gordon Cichon
• RAMs are implemented on 2D chips
• In grids of rows and columns
- Store information with one transistor and capacitor
- Structure photo
Flash Memory

Source Line

Word Line Control Gate

Bit Line

Float Gate

N

P

N