Outline

1. Memory
2. Latches
3. Clocks
4. Homework
Combinational logic outputs a function of inputs, only
Sequential logic outputs a function of inputs and state
State is remembered
Consider a sequential vending machine
Flip-flop introduction

- Stores, and outputs, a value
- Puts a special clock signal in charge of timing
- Allows output to change in response to clock transition
- More on this later
  - Timing and sequential circuits
Feedback and memory

- Feedback is the root of memory
- Can compose a simple loop from NOT gates
Feedback and memory

- Feedback is the root of memory
- Can compose a simple loop from NOT gates
- However, there is no way to switch the value
Outline

1. Memory
2. Latches
3. Clocks
4. Homework
TG and NOT-based memory

- Can break feedback path to load new value
- However, potential for timing problems
Ring oscillators

Like pulse shaping circuits with memory
Ring oscillators

- Period of Repeating Waveform ($tp$)
- Gate Delay ($td$)

<table>
<thead>
<tr>
<th></th>
<th>A (=X)</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Reset/set latch
Reset/set timing

Unstable state

Unstable state
RS latch state diagram

output = Q  Q
input = R  S
Outline

1. Memory
2. Latches
3. Clocks
4. Homework
Clocking terms

- Clock – Rising edge, falling edge, high level, low level, period
- Setup time: Minimum time before clocking event by which input must be stable ($T_{SU}$)
- Hold time: Minimum time after clocking event for which input must remain stable ($T_H$)
- Window: From setup time to hold time
Gated RS latch
Gated RS latch

\( \overline{S} \)  \( \overline{R} \)  \( \overline{ENB} \)  \( Q \)  \( \overline{Q} \)
# Memory element properties

<table>
<thead>
<tr>
<th>Type</th>
<th>Inputs sampled</th>
<th>Outputs valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclocked latch</td>
<td>Always</td>
<td>LFT</td>
</tr>
<tr>
<td>Level-sensitive latch</td>
<td>Clock high $(T_{SU}$ to $T_H$) around falling clock edge</td>
<td>LFT</td>
</tr>
<tr>
<td>Edge-triggered flip-flop</td>
<td>Clock low-to-high transition $(T_{SU}$ to $T_H$) around rising clock edge</td>
<td>Delay from rising edge</td>
</tr>
</tbody>
</table>
Clocking conventions

Active-high transparent

Active-low transparent

Positive (rising) edge

Negative (falling) edge
Timing for edge and level-sensitive latches
Falling edge-triggered D flip-flop

- Use two stages of latches
- When clock is high
  - First stage samples input w.o. changing second stage
  - Second stage holds value
- When clock goes low
  - First stage holds value and sets or resets second stage
  - Second stage transmits first stage
- $Q^+ = D$
- One of the most commonly used flip-flops
Falling edge-triggered D flip-flop

Clock high
Falling edge-triggered D flip-flop

Clock switching
Inputs sampled on falling edge, outputs change after falling edge
Falling edge-triggered D flip-flop

Clock low
Summary

- Memory
- Latches
- Flip-flops (more on these later)
Outline

1. Memory
2. Latches
3. Clocks
4. Homework
Reading assignment

- Sections 5.1–5.7
- Sections 6.1–6.4
Computer geek culture reference

Computer security
- PGP
- (Open)SSH
- (Type II) remailers
- Wireshark
- Crack