Feedback and Flip-Flops

Philipp Koehn

19 September 2016
The Story So Far

- We can encode numbers
- We can do calculation
- ... but it’s all a bit static

- How about a counter?
  → this requires "memory"
feedback
A Strange Contraption
Let’s Turn It On

Electricity is on → this opens the normally closed key
Let’s Turn It On

Electricity is off → this closes the normally closed key
What Do We Have?

- A Buzzer
- A Clock
- An Oscillator

(symbol)
Oscillator

- **Period** of oscillator

- **Frequency**: cycles per second

- **Unit**: 1 cycle per second: 1 Hertz

- Modern computes: Billions of Hertz = Gigahertz (GHz)
flip flop
Another Contraption
Closing Upper Key
Opening Upper Key

Same key configuration as initially

But: Now OUT is on --- we \textit{remembered} the key turn
Closing Lower Key

\[ \text{V} \quad \text{NOR} \quad \text{V} \quad \text{NOR} \quad \text{OUT} \]
Opening Lower Key

Back to initial state
• We have memory -- called **Reset-Set Flip-Flop**

• Truth table

<table>
<thead>
<tr>
<th>UPPER</th>
<th>LOWER</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>OUT</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Illegal</td>
</tr>
</tbody>
</table>

• **UPPER = SET**

• **LOWER = RESET**
Symmetric

\[ S \quad \text{NOR} \quad \overline{Q} \]

\[ R \quad \text{NOR} \quad Q \]
## Truth Table

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>$\bar{Q}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Q</td>
<td>$\bar{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Illegal</td>
<td></td>
</tr>
</tbody>
</table>
d-type flip flop
• Control bit ("clock")
  - on = write to memory
  - off = read from memory

• Data bit
  - data item to be written

• Output
  - current state of the memory
Replace Set/Reset with Data
Add Control Bit ("Clock")

Diagram:
- DATA
- CLOCK
- AND
- NOR
- Q
- \( \overline{Q} \)
D-Type Flip-Flop

- Also called D-type latch
- Circuit latches on one bit of memory and keeps it around
- Truth table

<table>
<thead>
<tr>
<th>Data</th>
<th>Clock</th>
<th>Q</th>
<th>Q̅</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>Q</td>
<td>Q̅</td>
</tr>
</tbody>
</table>

- Can also build these for multiple data bits
accumulative adder
Design Goal

• Adder has initially value 0

• Adding a number
  → value increases

• Resetting
  → value goes back to 0
Ingredients

8-BIT ADDER

8
A
B
S
CO CI

8-BIT LATCH

8
D
Q
CLK
Building an Accumulative Adder

- Latch: current sum
- Clock on → set it to 0
Building an Accumulative Adder

- Adder

- Combines
  - current value
  - selected input
Building an Accumulative Adder

- Can we pass output directly to latch?

- Concerns
  - select between 0 and sum
  - only stored when clock on
Building an Accumulative Adder

- 2-1 selector
- Either uses 0 or sum
- Built with AND gates
- Still have runaway feedback loop...
Building an Accumulative Adder

- Two Latches
  - one to store the sum
  - one to store input to adder

- Clock 1
  - carry out addition
  - store result

- Clock 2
  - transfer to set up next addition
Building an Accumulative Adder

- Combine the clocks

- Pressing the add key
  - carry out addition
  - store result in upper latch

- Release the add key
  - transfer to lower latch
  - set up next addition
• Remember the oscillator?

\[
\begin{array}{c}
\text{NOT} \\
\end{array}
\]

---

What Else?
- Each cycle of oscillator:
  keeps adding
What Else?

- We have something interesting here
edge triggered flip-flop
D-Type Latch

- When clock is on, save data
- "Level-triggered"
• "Edge-triggered": changes value, when switched from 0 to 1
Edge Triggered D-Type Latch

Symbol
### Truth Table

<table>
<thead>
<tr>
<th>Data</th>
<th>Clock</th>
<th>Q</th>
<th>( \bar{Q} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>↑</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>Q</td>
<td>( \bar{Q} )</td>
</tr>
</tbody>
</table>

Philipp Koehn  
Computer Systems Foundation: Feedback and Flip-Flops  
19 September 2016
ripple counter
Oscillator and Latch

![Oscillator and Latch diagram]

<table>
<thead>
<tr>
<th>Data</th>
<th>Clock</th>
<th>Q</th>
<th>$\bar{Q}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>↑</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Oscillator and Latch

<table>
<thead>
<tr>
<th>Data</th>
<th>Clock</th>
<th>Q</th>
<th>Ō</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>↑</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
# Halving of Frequency

<table>
<thead>
<tr>
<th>Data</th>
<th>Clock</th>
<th>Q</th>
<th>Q̅</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>↑</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**IN**

![Input waveform]

**OUT**

![Output waveform]
Multiple Bits

CLK

OUT0

OUT1

OUT2

OUT3

NOT

D → CLK

Q

D → CLK

Q

D → CLK

Q

D → CLK

Q

NOT

OUT0

OUT1

OUT2

OUT3
Ripple Counter

<table>
<thead>
<tr>
<th>CLK</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OUT2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OUT2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>