Time

Until now: we have essentially ignored the issue of time

• We have assumed that our digital logic circuits perform their computations instantaneously

• Our digital logic circuits have been “stateless”
  – Once you present a new input, they forget everything about previous inputs
  – We call this type of digital system **combinatorial logic**
Time

In reality, time is an important issue:

- Even our logic gates induce a small amount of delay (on the order of a few nanoseconds)
- For much of what we do – we actually want our circuits to have some form of memory
Timing Notation

X

In transition (undetermined)

high

low

time
Timing Notation

Either high or low (but well defined and constant)

In transition (undetermined)

X

time

low
D Flip Flops

Data Input

[Diagram of D Flip Flop]
D Flip Flops

Stored value (output)
D Flip Flops

NOT of stored value
D Flip Flops

Clock input
D Flip Flops

When the clock transitions from high to low: the value of D is stored
D Flip Flop

What happens to Q and Q'?
What happens to Q and Q’?
What happens to Q and Q’?
D Flip Flop

D

C

Q

Q'

No change in state
An Application of D Flip Flops

What does this circuit do?
Shift Register

On each clock transition from high to low:

- X0 takes on the current value of D
- X1 <- X0
- X2 <- X1
Another D Flip Flop Circuit

How does this circuit behave?

Andrew H. Fagg: Embedded Real-Time Systems: Sequential Logic
Frequency Divider

How does this circuit behave?

CLK

Q

CLK → D → Q → Q' → Q

Andrew H. Fagg: Embedded Real-Time Systems: Sequential Logic 71
Frequency Divider

Q flips state on every downward edge of the clock

CLK

Q

CLK

D Q

Q'}
A Bit About Binary Encoding

If a boolean variable can only encode two different values, how do we represent a larger number of values?
Binary Encoding

How do we represent a larger number of values?
Binary Encoding

How do we represent a larger number of values?

- As with our decimal number system: we concatenate binary digits (or “bits”) into strings
Binary Encoding

- The first (rightmost) bit is the 1’s digit
- The second bit is the 2’s digit
- The ith bit is the $2^{i-1}$ ’s digit
How do we convert from binary to decimal in general?

<table>
<thead>
<tr>
<th>B2</th>
<th>B1</th>
<th>B0</th>
<th>decimal</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>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>
Last Time

Sequential Logic
• D Flip Flops
• Shift registers
• Binary number system

There are only 10 types of people in the world: Those who understand binary and those who don't.
Today

• A little more on number systems
• Use of flip-flops
• Microprocessor basics
  – Memory
  – Arithmetic Logical Units
  – Instructions and execution
Administrivia

- Homework 1 due today at 5:00
- Homework 2 available tonight
Binary to Decimal Conversion

\[ \text{value} = B_0 + B_1 \times 2^1 + B_2 \times 2^2 + B_3 \times 2^3 + \ldots \]

\[ \text{value} = \sum_{i=0}^{N-1} B_i \times 2^i \]

How do we convert from decimal to binary?
Decimal to Binary Conversion

\[ \forall i : B_i \leftarrow 0 \]

\[ \text{while}(\text{value} \neq 0) \]
\[
\{ \\
\text{Find } i \text{ such that } 2^{i+1} > \text{value} \geq 2^i \\
B_i \leftarrow 1 \\
\text{value} \leftarrow \text{value} - 2^i \\
\}
\]
How would we build a circuit that counts the number of clock ticks that have gone by?

<table>
<thead>
<tr>
<th>B2</th>
<th>B1</th>
<th>B0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Binary Counter

How would we build a circuit that counts the number of clock ticks that have gone by?

Insight:
- B1 changes state at half the frequency that B0 does
- B2 changes state at half the frequency of B1

<table>
<thead>
<tr>
<th></th>
<th>B2</th>
<th>B1</th>
<th>B0</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>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<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>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Ripple Counter

The carry “ripples” down the chain …
Ripple Counter Notes

• The bits do not change state at the same time
• This can be repaired with a more sophisticated circuit design
  – We will experiment with this in hw2
Flip-Flop Notes

• Means of storing ‘bits’ of data
• Have now seen two circuits that operate on sets of ‘bits’ (or binary numbers)
  – Counter
  – Shift register
    • What arithmetic operation does shifting perform?

• These are examples of operations that are performed by the “Arithmetic Logical Unit”