Lecture 9 - Adders
- Half-adders
- Full-adders
- Carry-ripple Adder

Functional Requirements:
- Design a circuit that will add two 2-bit binary numbers
  - Input: A1A0, B1B0
  - Output: S1S0: sum of inputs

Inputs | Outputs
-------|-------
```c
a1 a0 b1 b0 c s1 s0
```

16-bit Adder: 16-bits * 2-operands = 32 inputs
Over 4,000,000,000 rows

*Exponential Growth for Two-Level Adder Implementation*
**Datapath Components: Adders: Carry-Ripple**

**Functional Requirements:**
- Design a circuit that will add two bits
  - Input: $A, B$
  - Output: $S$: sum of inputs
  - $C$: carry bit

**Datapath Components: Adders: Carry-Ripple: Half-Adder**

1) Capture the Function

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$b$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2) Convert to Equations

- $c_o = ab$
- $s = a'b + ab' = a \ xor \ b$

**Datapath Components: Adders: Carry-Ripple: Full Adder**

**Functional Requirements:**
- Design a circuit that will add three bits
  - Input: $A, B, C_{in}$
  - Output: $S$: sum of inputs
  - $C$: carry bit

3) Create the Circuit

- $c_o = ab$
- $s = a'b + ab' = a \ xor \ b$
1) Capture the Function

Input: A, B, Cin
Output: S: sum of inputs, C: carry bit

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>a  b  c</td>
<td>c  s</td>
</tr>
<tr>
<td>0  0  0</td>
<td>0  0</td>
</tr>
<tr>
<td>0  0  1</td>
<td>0  1</td>
</tr>
<tr>
<td>0  1  0</td>
<td>1  0</td>
</tr>
<tr>
<td>0  1  1</td>
<td>1  1</td>
</tr>
<tr>
<td>1  0  0</td>
<td>0  1</td>
</tr>
<tr>
<td>1  0  1</td>
<td>1  0</td>
</tr>
<tr>
<td>1  1  0</td>
<td>0  1</td>
</tr>
<tr>
<td>1  1  1</td>
<td>1  1</td>
</tr>
</tbody>
</table>

2) Convert to Equations

\[ c_0 = b \cdot c + a \cdot c + a \cdot b \]
\[ s = a \oplus b \oplus c \]

3) Create the Circuit

For a 4-bit adder composed of full adders, what would be the soonest time you could expect a result? A) .9 ns  B) 2.3 ns  C) 3.6 ns

4-bit Adder:

Inputs (9-bits): a3a2a1a0, b3b2b1b0, cin
Outputs (5-bits): c0, s3s2s1s0

NOTE: Adder will exhibit temporarily incorrect (spurious) results until the carry bit from the rightmost bit has had a chance to propagate (ripple) all the way through to the leftmost bit.
Datapath Components: Adders: Carry-Ripple

8-bit Carry-Ripple Adder:
- Inputs (16-bits): a7a6a5a4a3a2a1a0, b7b6b5b4b3b2b1b0
- Outputs (5-bits): c0, s3s2s1s0

8-bit Calculator Functional Requirements:
- Design a circuit that will add two eight-bit inputs controlled by DIP switches and output the result using eight LEDs
- Input: A (8-bits), B (8-bits)
- Output: S (8-bits): sum of inputs

Functional Requirements:
- Design a circuit that will weigh an object, and add an adjustment from a user
  - Input: 8-bit Weight Reported by Sensor
  - 3-bit User Adjustment
  - Output: S7..0: sum of inputs
8-bit DIP switch-based adding calculator. The addition 2+3=5 is shown.

8-bit DIP switch-based adding calculator, using a register to block spurious LED outputs. The LEDs only get updated after the button is pressed, which loads the output register.

8-bit Calculator Functional Requirements:
- Design a circuit that will add two eight-bit inputs controlled by DIP switches and output the result using eight LEDs. The output should only be updated with the user presses a calculate button
  - Input: A (8-bits), B (8-bits), e: input from calculate button
  - Output: S (8-bits): sum of inputs