Arithmetic/Logic Unit & Shifter
Arithmetic/Logic Unit (ALU)

- A combinational circuit that performs a set of basic arithmetic and logic operations
  - A number of selection lines used to determine the operation to be performed
  - $K$ selection lines $\rightarrow 2^k$ distinct operations
- Arithmetic circuit + Logic circuit

Diagram:

- Data input A
- Data input B
- Carry input
- Operation select \{S_0, S_1, S_2\}
- Mode select
- Output data G
- Carry output

$$2^n$$
### Arithmetic Circuit

- **Input:**
  - $B$ (input), $S_0$, $S_1$
  - $C_{in}$

- **Output:**
  - $S_0$, $S_1$
  - $C_{out}$
  - $G$

- **Logic:**
  - **Sum**
  - **Cout**

- **n-bit adder:**
  - $G = X + Y + C_{in}$

#### Truth Table

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>$Y$</th>
<th>$C_{in} = 0$</th>
<th>$C_{in} = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>00...0</td>
<td>$G = A$</td>
<td>$G = A + 1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$B$</td>
<td>$G = A + B$</td>
<td>$G = A + B + 1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$B'$</td>
<td>$G = A + B'$</td>
<td>$G = A + B' + 1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>11...1</td>
<td>$G = A - 1$</td>
<td>$G = A$</td>
</tr>
</tbody>
</table>

**Diagram:**

- $B$ input logic
- $S_0$, $S_1$ inputs
- $C_{in}$ input
- $X$, $Y$ inputs to sum
- $C_{out}$ output
- $G$ output

**Equations:**

- $G = A + 1$ when $S_1 = 1$ and $S_0 = 0$,
- $G = A$ when $S_1 = 1$ and $S_0 = 1$.
Arithmetic Circuit (contd.)

\[ \begin{align*}
C_{in} & \quad S_0 \quad S_1 \\
A_0 & \\
B_0 & \\
A_1 & \\
B_1 & \\
\vdots & \\
A_{n-1} & \\
B_{n-1} & \\
\end{align*} \]
Logic Circuit

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>output</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$G = A \land B$</td>
<td>AND</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$G = A \lor B$</td>
<td>OR</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$G = A \oplus B$</td>
<td>XOR</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$G = A'$</td>
<td>NOT</td>
</tr>
</tbody>
</table>

Computer Principles
Combination of Arithmetic and Logic Circuits

- B’ - one’s complement of B
- Flip each bit in B

<table>
<thead>
<tr>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>Cin</th>
<th>Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>G=A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>G=A+1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>G=A+B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>G=A+B+1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>G=A+B’</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>G=A+B’+1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>G=A-1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>G=A</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>G=A∧B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>G=A∨B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>G=A⊕B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>G=A’</td>
</tr>
</tbody>
</table>

n-bit ALU

B’ - one’s complement of B
Flip each bit in B
Shift Operations

- Logical shift
  - A logical shift moves bits to the left or right
  - The bits which fall off the end of the word are discarded and the word is filled with 0's from the opposite end

- Arithmetic shift
  - The arithmetic left shift is the same as the logical left shift
  - In the arithmetic right shift, the leftmost bits are filled with the sign bit of the original number

<table>
<thead>
<tr>
<th></th>
<th>Logical shift</th>
<th>Arithmetic shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>x &gt;&gt; 3</td>
<td>10011101</td>
<td>11101000</td>
</tr>
<tr>
<td>x &lt;&lt; 3</td>
<td>10011101</td>
<td>11101000</td>
</tr>
</tbody>
</table>

Examples:
- Logical shift: `x >> 3`
- Arithmetic shift: `x << 3`
Shift Operations (contd.)

- On positive integers, a logical left shift is equivalent to multiplication by 2
  - \( (00001011_2 \ll 1) = 00010110_2 \)

- A logical right shift is equivalent to division by 2 (i.e., the quotient)
  - \( (00001011_2 \gg 1) = 00000101_2 \)

- Take floor value if the result is not an integer
  - Floor of \( x \) (\( \lfloor x \rfloor \)) is the greatest integer less than or equal to \( x \)
    - \( \lfloor 11/2 \rfloor = 5, \lfloor -3/2 \rfloor = -2 \)
Shift Operations (contd.)

- The arithmetic shift extends the notion (i.e., shifts for multiplication and division) to 2's complement numbers:
  - $(1100_2 << 1) = 1000_2$
  - $(1100_2 << 2) = 0000_2 \Rightarrow \text{Underflow}$
  - $(1100_2 >> 1) = 1110_2$
  - $(1100_2 >> 2) = 1111_2$
  - $(0100_2 << 1) = 1000_2 \Rightarrow \text{Overflow}$
Sign Extension

- Converting from smaller to larger integer in 2's complement representation
  - Given w-bit signed integer x, convert it to w+k-bit integer with same value
Rotate Operations

- A rotate operation is a circular shift in which no bits are discarded.
- A right rotation by n bits of an n bit word returns the original word unchanged.
- A right rotation by n-1 bits is equivalent to a left rotation of 1 bit.
  - The left rotation operation is redundant because a left rotation of j bits is equivalent to a right rotation of n-j bits.
Shifter

<table>
<thead>
<tr>
<th>S</th>
<th>D₃</th>
<th>D₂</th>
<th>D₁</th>
<th>D₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Rᵢn</td>
<td>B₃</td>
<td>B₂</td>
<td>B₁</td>
</tr>
</tbody>
</table>

Right shift

<table>
<thead>
<tr>
<th>S</th>
<th>D₃</th>
<th>D₂</th>
<th>D₁</th>
<th>D₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B₂</td>
<td>B₁</td>
<td>B₀</td>
<td>Lᵢn</td>
</tr>
</tbody>
</table>

Left shift

MUX

S

Rᵢn

D₃

D₂

D₁

D₀

B₃

B₂

B₁

B₀

Lin
Rotator

<table>
<thead>
<tr>
<th>S₁</th>
<th>S₀</th>
<th>D₃</th>
<th>D₂</th>
<th>D₁</th>
<th>D₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>B₃</td>
<td>B₂</td>
<td>B₁</td>
<td>B₀</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>B₂</td>
<td>B₁</td>
<td>B₀</td>
<td>B₃</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>B₁</td>
<td>B₀</td>
<td>B₃</td>
<td>B₂</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>B₀</td>
<td>B₃</td>
<td>B₂</td>
<td>B₁</td>
</tr>
</tbody>
</table>