Multiplexer (MUX): A logic block that selects 1 out of n inputs to make it available to an output. (important abstraction)

Logic diagram

\[ f = S'd_0 + Sd_1 \]

Operation:
1) \( S = 0 \), \( f = d_0 \)
2) \( S = 1 \), \( f = d_1 \)

Symbolic form

Labels indicate the value of select input for which adjacent data input is copied to output.
Generalize Mux:
1. Have multiple select inputs
   e.g.: 4:1 will have 2 select inputs S, S0
\[ S_1 S_0 = 10 \]

2 Building larger MUX using smaller muxes
2 Building larger MUX using smaller muxes
   e.g. Building a 4:1 MUX using 2:1 muxes

   Need:
   4 data inputs
   2 select inputs.

\[ S_0, S_1 \]

\[ d_0, d_1, d_2, d_3 \]

4:1 MUX

\[ d_2 \text{ as output} \]
\[ S_1S_0 = 10 \]
\[ f = d_2 \]

Use:
   \( f \) function of \( k \) variables can be implemented using a MUX.
Use:

Any function of $k$ variables can be implemented using a mux.

(i) $k$-variable function has $2^k$ rows in truth table
(ii) Each row is a 1 or a 0
(iii) Use a $2^k:1$ mux to implement function

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<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>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

$S_2S_1S_0 = ABC \ 011$
Combine available logic blocks using MUX

<table>
<thead>
<tr>
<th>AU Function</th>
<th>$k_1, k_0$</th>
<th>LU function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>00</td>
<td>A AND B</td>
</tr>
<tr>
<td>A+B+1</td>
<td>01</td>
<td>A OR B</td>
</tr>
<tr>
<td>A</td>
<td>10</td>
<td>A NAND B</td>
</tr>
<tr>
<td>A-B</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

![ALU Diagram](image-url)
Decoder
With n inputs selects 1 of 2^n lines as output.

- Exactly one of the outputs is 1 for each input

Symbol

\[
\begin{array}{c|c}
I_0 & 1 \downarrow \\
I_1 & 2 \downarrow \\
I_2 & 3 \downarrow \\
\end{array}
\]

Input | Output
--- | ---
00 | 1 0 0 0
01 | 0 1 0 0
10 | 0 0 1 0
11 | 0 0 0 1

Functions as a minterm generator
* Use it to implement any k-variable function