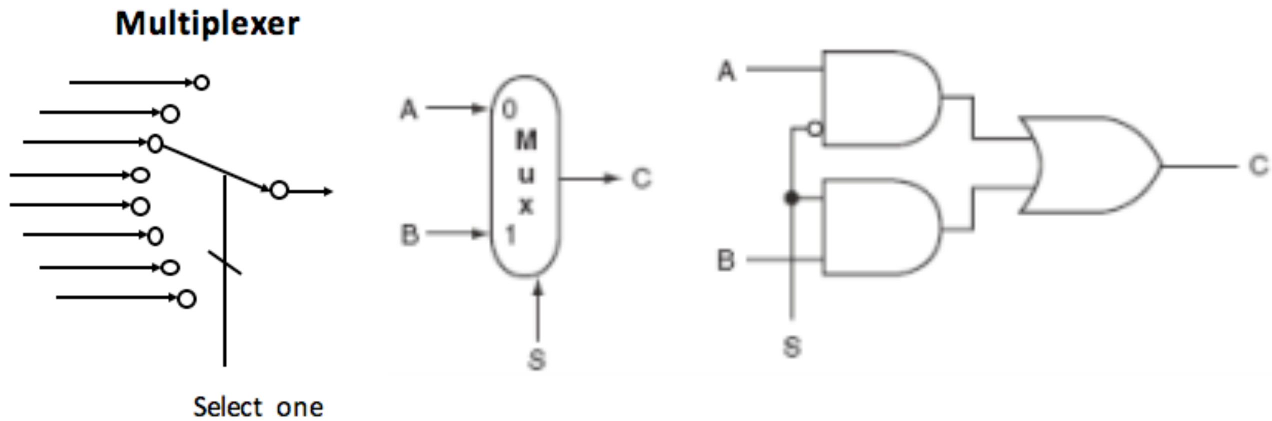
CS 240 Lab 3
Combinational and
Arithmetic Circuits

- **Multiplexer**
- **Decoder**
- **Adder**

Multiplexer

A multiplexer can be thought of as a **selection circuit**, which steers a single input from a set of inputs through to the output, based on the select line.



- n select lines
- 2^n input lines
- 1 output

One of the possible 2^n inputs is chosen by the n select lines, and gated through to the output of a multiplexer. The truth table for an 8x1 MUX is:

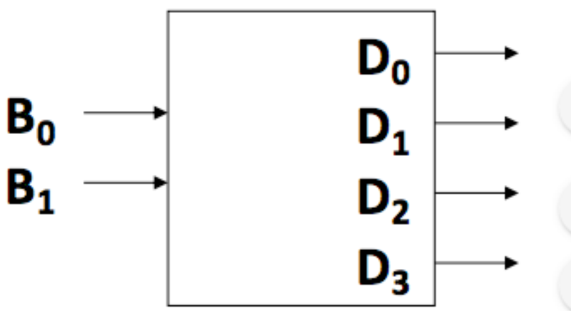
S2	**S1**	**S0**	**Q**
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

Decoder

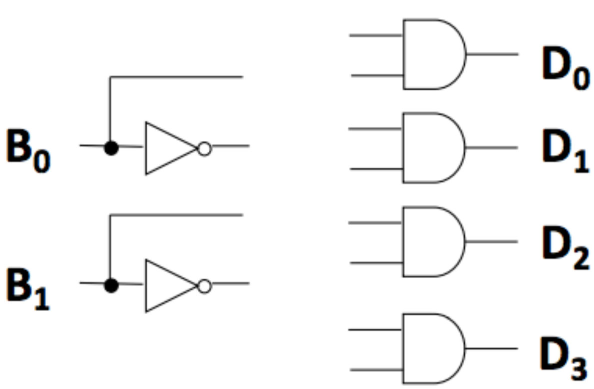
A decoder takes an n-bit binary number as an input, and asserts the corresponding numbered output from the set of 2^n outputs.

- n input/select lines
- 2^n outputs
- only one of the outputs is active at any given time, based on the value of the n select lines.

2x4 Decoder



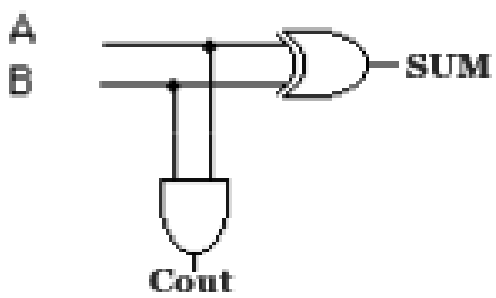
Built with code detectors:



Truth table for an 3x8 decoder

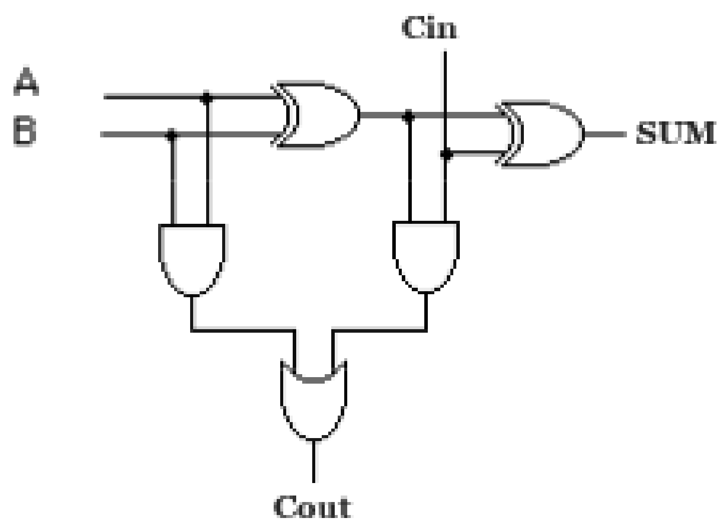
S2	S1	S0	|	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	|	1	0	0	0	0	0	0	0
0	0	1	|	0	1	0	0	0	0	0	0
0	1	0	|	0	0	1	0	0	0	0	0
0	1	1	|	0	0	0	1	0	0	0	0
1	0	0	|	0	0	0	0	1	0	0	0
1	0	1	|	0	0	0	0	0	1	0	0
1	1	0	|	0	0	0	0	0	0	1	0
1	1	1	|	0	0	0	0	0	0	0	1

Half-Adder – adds two one-bit values



A	**B**	**Cout**	**Sum**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Full Adder – uses two half-adders and incorporates a carry-in

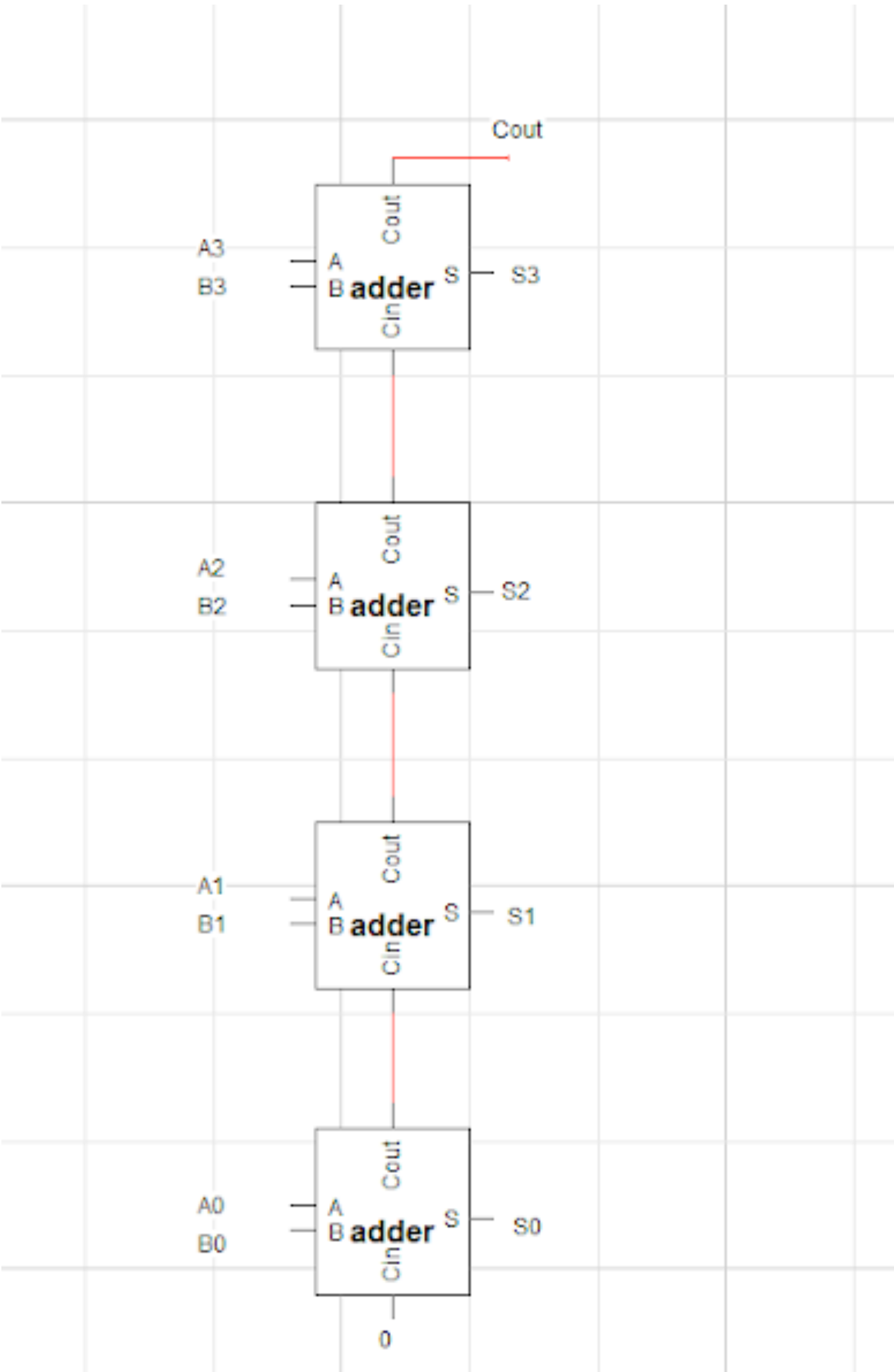


Cin	**A**	**B**	**Cout**	**Sum**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Sum = $A \oplus B \oplus Cin$

Cout = $AB+(A \oplus B)Cin$

4-bit Ripple-Carry Adder



Overflow when Adding

An overflow occurs when adding two n-bit numbers if the result will not fit in n bits.

An overflow can be detected when:
-Two positive numbers added together yield a negative result, or
-Two negative numbers added together yield a positive result.

Overflow can also be detected when:
-The Cin and Cout bits to the most significant pair of bits being added are not the same.

An overflow cannot result if a positive and negative number are added.

Example: given 4 bits:

$$
\begin{array}{r}
0111_2 \\
+\ 0001_2 \\
\hline
1000_2
\end{array}
$$

1000_2 = overflow **NOTE:** there is not a carry-out!

In two's complement representation, a carry-out does not indicate an overflow, as it does in unsigned representation.

Example: given 4 bits,

$$
\begin{array}{r}
1001_2\ (-7_{10}) \\
+\ 1111_2\ (-1_{10}) \\
\hline
1\ 1000_2\ (-8_{10})
\end{array}
$$

no overflow, even though there is a carry-out