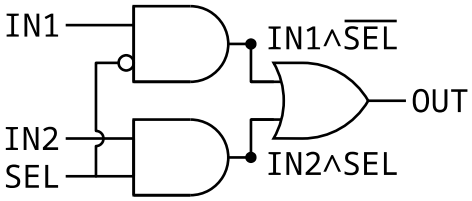
Worksheet: Decoders and Multiplexers

2 to 1 Multiplexers

Complete the truth table, *Table 1A*, to the right for the circuit diagram given below.



When SEL (select) is set low, the output follows IN1. When SEL is set high, the output follows IN2.

This circuit is a possible implementation of a one-bit multiplexer, also called a *2 to 1 multiplexer.* This circuit is often drawn as:

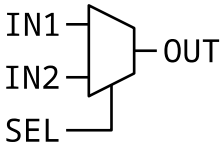


Table 1A

	SEL	IN1	IN2	IN2∧SEL	$\overline{SEL}$	IN1∧$\overline{SEL}$	OUT
1	0	0	0	0	1	0	0
2	0	0	1	0	1	0	0
3	0	1	0	0	1	1	1
4	0	1	1	0	1	1	1
5	1	0	0	0	0	0	0
6	1	0	1	1	0	0	1
7	1	1	0	0	0	0	0
8	1	1	1	1	0	0	1

Table 1B contains the first four rows of *Table 1A* – showing only the inputs and outputs, not intermediate values. The circles show that:

- When SEL and IN1 are both set to 0, OUT is set to 0. Changing the value of IN2 does not change the value of OUT.
- When SEL is set to 0 and IN1 is set to 1, OUT is set to 1. Changing the value of IN2 again does not change the value of OUT.

Table 1B

	SEL	IN1	IN2	OUT
1	0	0	0	**0**
2	0	0	1	**0**
3	0	1	0	**1**
4	0	1	1	**1**

Thus, we "don't care" what the value of IN2 is when SEL is set to 0. Only the value of IN1 matters to the value of OUT.

We can simplify *Table 1B* by introducing a truth table value called "*don't care*", which is represented by the letter X. We replace rows 1 and 2 with a single row, where X has the meaning that whether the value is 0 or 1 does not change OUT. We again do this, combining row 3 and 4 into a single row.

Table 1B Simplified

	SEL	IN1	IN2	OUT
1,2	0	0	X	**0**
3,4	0	1	X	**1**

Following the same logic for rows 5 and 7, and rows 6 and 8. (Note that the order of the rows was changed to more clearly show how changing `IN1` does not change the value of `OUT` when `SEL` is set to `1`).

Table 1C: Simplification of Rows 6-8

	SEL	IN1	IN2	OUT
5	1	0	0	**0**
7	1	1	0	**0**
6	1	0	1	**1**
8	1	1	1	**1**

→

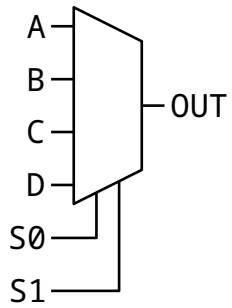
	SEL	IN1	IN2	OUT
5,7	1	X	0	**0**
6,8	1	X	1	**1**

The final, combined truth table for a 2 to 1 Multiplexer is given below. The row numbers from *Table A* have been left in so you can verify the values.

Table 1D: 2 to 1 Multiplexer Truth Table

	IN1	IN2	SEL	OUT
1,2	0	X	0	0
3,4	1	X	0	1
5,7	X	0	1	0
6,8	X	1	1	1

N to 1 Multiplexers



There will be times when the output will need to be selected from more than two inputs.

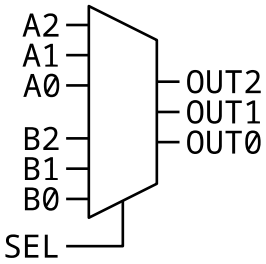
For example, the diagram to the left shows the circuit element for a *4 to 1 multiplexer*. There are 4 inputs, and the output is selected using two select lines, S1 and S0. The selection is based on the binary number representation (`00` selects input `A`, `01` selects input `B`, `10` selects input `C` and `11` selects input `D`). Examine and understand the truth table for the 4 to 1 multiplexer:

Table 2: 4 to 1 Multiplexer

	A	B	C	D	S1	S0	OUT
1	0	X	X	X	0	0	0
2	1	X	X	X	0	0	1
3	X	0	X	X	0	1	0
4	X	1	X	X	0	1	1
5	X	X	0	X	1	0	0
6	X	X	1	X	1	0	1
7	X	X	X	0	1	1	0
8	X	X	X	1	1	1	1

Worksheet: Decoders and Multiplexers

2N to N Multiplexers



To represent all but the simplest binary number requires more than a single bit. For example, to represent all numbers 0 to 7 requires 3 bits: $2^3 = 8$, and there are eight integers in the range [0, 7]. If we wish to select between two 3-bit numbers, we would use a *6 to 3 multiplexer*, shown in the diagram to the right. When `SEL` is set to `0`, the output, `OUT2:OUT0` is set to `A2:A0`. When `SEL` is set to `1`, the output, `OUT2:OUT0` is set to `B2:B0`.

For a device that uses byte-wide or longer numbers, one might want a *16 to 8 multiplexer* or perhaps wider. This part would have a large number of pins for such a simple task. We will shortly learn of an alternative to using a multiplexer, but first let's discuss decoders.

2 to 4 Decoder

Complete the truth table, *Table 1A*, to the right for the circuit diagram given below.

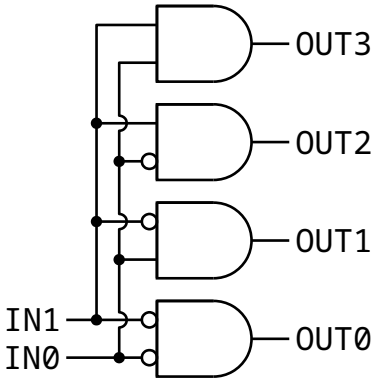
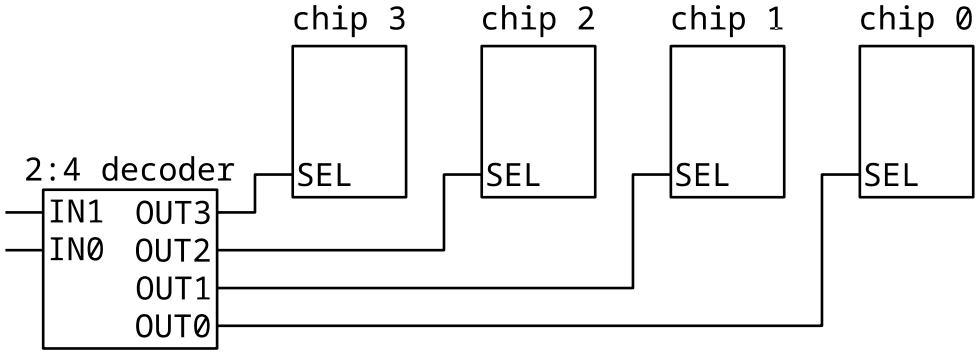


Table 1A: 2 to 4 Decoder

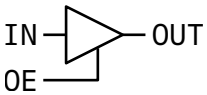
	IN0	IN1	OUT3	OUT2	OUT1	OUT0
1	0	0	0	0	0	1
2	0	1	0	0	1	0
3	1	0	0	1	0	0
4	1	1	1	0	0	0

Notice that only a single output is ever set to `1` at a time. This is called "***one-hot***" encoding. This type of circuit is commonly used when there are multiple chips, and we wish to select one chip only to perform an operation. For example, we may wish to save data to only one memory chip, or read data from only one chip. The figure below shows the circuit diagram of a 2 to 4 decoder chip providing chip select signals to four separate chips. Not all lines on the chips are shown.



Worksheet: Decoders and Multiplexers

Tristate Buffers

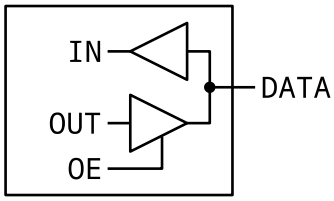


IN	OE	OUT
0	1	0
1	1	1
X	0	Z

To the right is the symbol and the truth table for a tristate buffer.

A tristate buffer, as the name suggests, has three possible states. When OE, the output enable, is set to 1, the output follows the input. If OE is set to 0, the buffer will output be in a ***high-impedance*** state (also labeled a high-Z state, and in the truth table to the left represented with letter Z). Here, high impedance means the output will neither be a 0 nor a 1. From a perspective of logic, it is as though the buffer output has been electrically disconnected from the circuit.

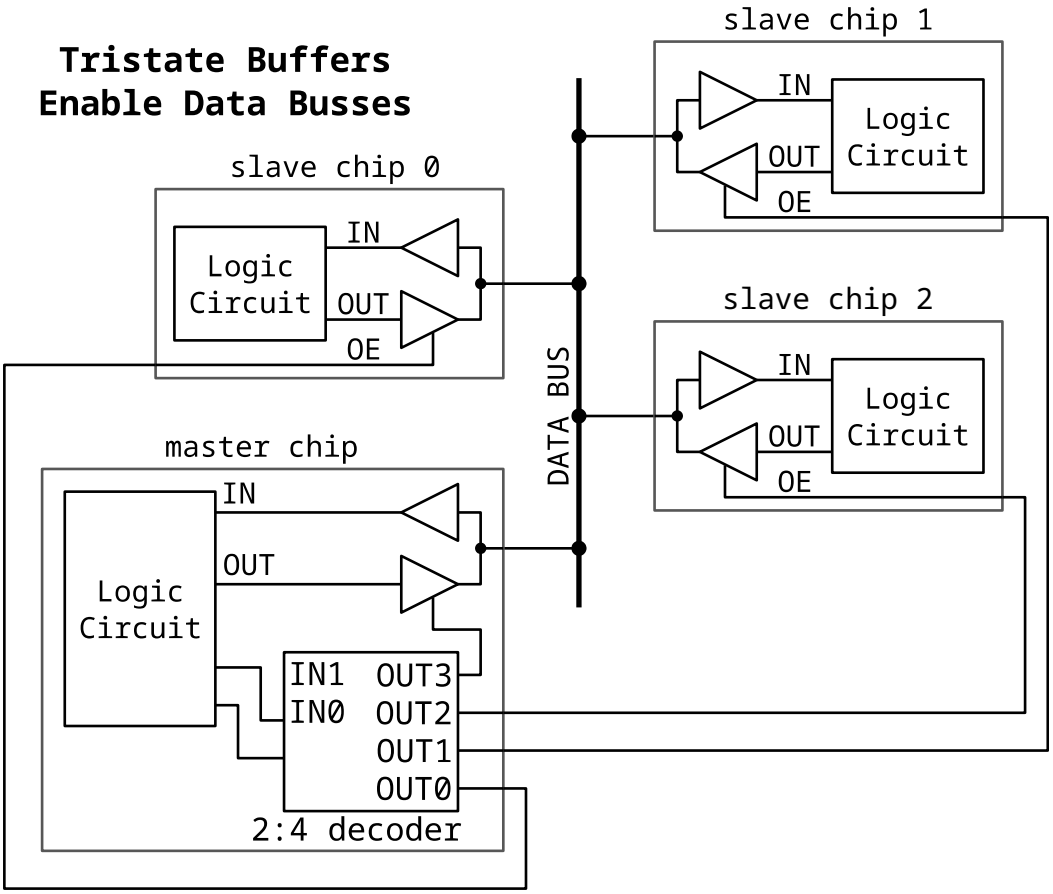
Tristate Buffers and the Data Bus



Tristate buffers are the essential to the operation of a data bus. A data bus is one or more electrical wire that is shared between multiple chips to allow communication between them. The figure to the left shows a simple diagram of the circuit connected to a data pin of a chip.

The top part of the circuit is an input buffer that can detect the logic level on the data pin. The lower part of the circuit is an output buffer with tristate that only drives a logic level if the output is enabled by setting a 1 value to OE.

Connecting a number of pins to a single wire creates a data bus. Examine the figure below.



Each chip can sense the logic level of the bus, but only one chip is allowed to drive the bus at a time. The decoder in the *master chip* controls this because only one of the outputs is 1 at a time. In actual circuits, such as a computer, the data bus will be a number of lines – likely 64 data lines for a 64-bit processor.