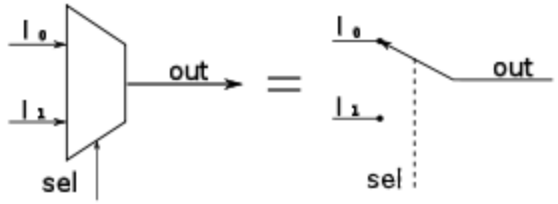
WIKIPEDIA
The Free Encyclopedia

Multiplexer

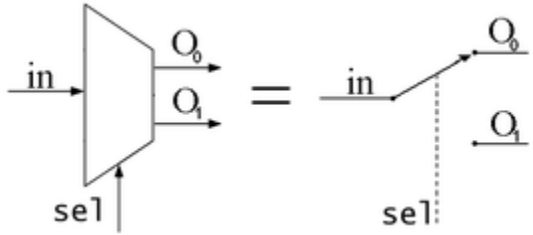
In electronics, a **multiplexer** (or **mux**; spelled sometimes as **multiplexor**), also known as a **data selector**, is a device that selects between several analog or digital input signals and forwards the selected input to a single output line.[1] The selection is directed by a separate set of digital inputs known as select lines. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output.[2]



Schematic of a 2-to-1 multiplexer. It can be equated to a controlled switch.

A multiplexer makes it possible for several input signals to share one device or resource, for example, one analog-to-digital converter or one communications transmission medium, instead of having one device per input signal. Multiplexers can also be used to implement Boolean functions of multiple variables.

Conversely, a **demultiplexer** (or **demux**) is a device taking a single input and selecting signals of the output of the compatible **mux**, which is connected to the single input, and a shared selection line. A multiplexer is often used with a complementary demultiplexer on the receiving end.[1]



Schematic of a 1-to-2 demultiplexer. Like a multiplexer, it can be equated to a controlled switch.

An electronic multiplexer can be considered as a multiple-input, single-output switch, and a demultiplexer as a single-input, multiple-output switch.[3] The schematic symbol for a multiplexer is an isosceles trapezoid with the longer parallel side containing the input pins and the short parallel side containing the output pin.[4] The schematic on the right shows a 2-to-1 multiplexer on the left and an equivalent switch on the right. The ***sel*** wire connects the desired input to the output.

Applications

Multiplexers are part of computer systems to select data from a specific source, be it a memory chip or a hardware peripheral. A computer uses multiplexers to control the data and address buses, allowing the processor to select data from multiple data sources

In digital communications, multiplexers allow several connections over a single channel, by connecting the multiplexer's single output to the demultiplexer's single input (Time-Division Multiplexing). The image to the right demonstrates this benefit. In this case, the cost of implementing separate channels for each data source is higher than the cost and inconvenience of providing the multiplexing/demultiplexing functions.

At the receiving end of the data link a complementary *demultiplexer* is usually required to break the single data stream back down into the original streams. In some cases, the far end system may have functionality greater than a simple demultiplexer; and while the demultiplexing still occurs technically, it may never be implemented discretely. This would be the case when, for instance, a multiplexer serves a number of IP network users; and then feeds directly into a router, which immediately reads the content of the entire link into its routing processor; and then does the demultiplexing in memory from where it will be converted directly into IP sections.

The basic function of a multiplexer: combining multiple inputs into a single data stream. On the receiving side, a demultiplexer splits the single data stream into the original multiple signals.

Often, a multiplexer and demultiplexer are combined into a single piece of equipment, which is simply referred to as a *multiplexer*. Both circuit elements are needed at both ends of a transmission link because most communications systems transmit in both directions.

In analog circuit design, a multiplexer is a special type of analog switch that connects one signal selected from several inputs to a single output.

Digital multiplexers

In digital circuit design, the selector wires are of digital value. In the case of a 2-to-1 multiplexer, a logic value of 0 would connect I_0 to the output while a logic value of 1 would connect I_1 to the output. In larger multiplexers, the number of selector pins is equal to $\lceil \log_2(n) \rceil$ where n is the number of inputs.

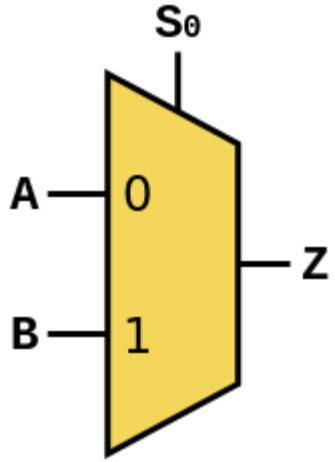
For example, 9 to 16 inputs would require no fewer than 4 selector pins and 17 to 32 inputs would require no fewer than 5 selector pins. The binary value expressed on these selector pins determines the selected input pin.

A 2-to-1 multiplexer has a boolean equation where A and B are the two inputs, S_0 is the selector input, and Z is the output:

$$Z = (A \wedge \neg S_0) \vee (B \wedge S_0)$$

Which can be expressed as a truth table:

S_0	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



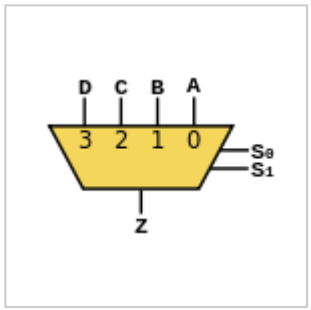
A 2-to-1 mux

Or, in simpler notation:

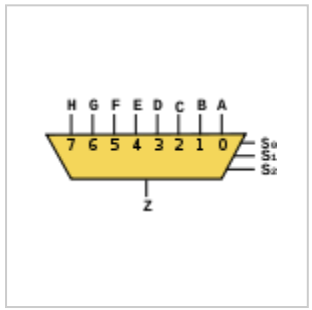
S_0	Z
0	A
1	B

These tables show that when $S_0 = 0$ then $Z = A$ but when $S_0 = 1$ then $Z = B$. A straightforward realization of this 2-to-1 multiplexer would need 2 AND gates, an OR gate, and a NOT gate. While this is mathematically correct, a direct physical implementation would be prone to race conditions that require additional gates to suppress.[5]

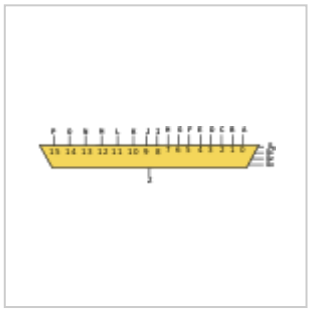
Larger multiplexers are also common and, as stated above, require $\lceil \log_2(n) \rceil$ selector pins for n inputs. Other common sizes are 4-to-1, 8-to-1, and 16-to-1. Since digital logic uses binary values, powers of 2 are used (4, 8, 16) to maximally control a number of inputs for the given number of selector inputs.



4-to-1 mux



8-to-1 mux

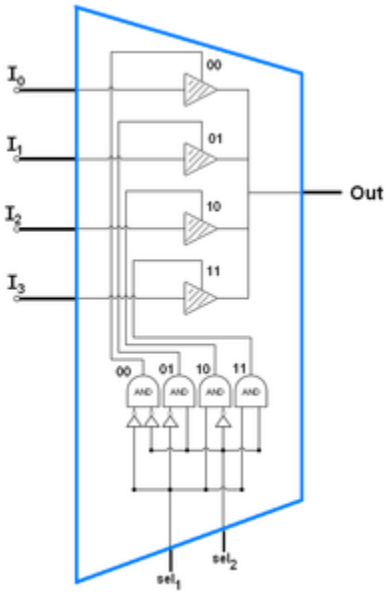


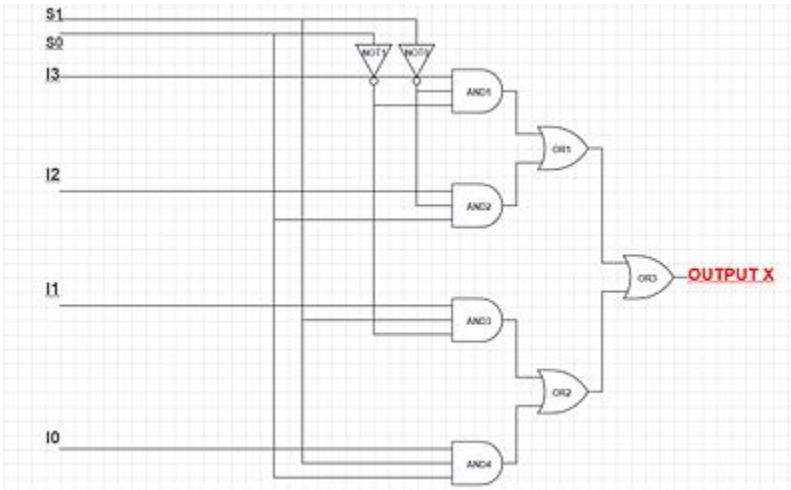
16-to-1 mux

The boolean equation for a 4-to-1 multiplexer is:

$$Z = (A \wedge \neg S_0 \wedge \neg S_1) \vee (B \wedge S_0 \wedge \neg S_1) \vee (C \wedge \neg S_0 \wedge S_1) \vee (D \wedge S_0 \wedge S_1)$$

The following 4-to-1 multiplexer is constructed from 3-state buffers and AND gates (the AND gates are acting as the decoder):





A 4:1 MUX circuit using 3 input AND and other gates

The subscripts on the I_n inputs indicate the decimal value of the binary control inputs at which that input is let through.

Chaining multiplexers

Larger Multiplexers can be constructed by using smaller multiplexers by chaining them together. For example, an 8-to-1 multiplexer can be made with two 4-to-1 and one 2-to-1 multiplexers. The two 4-to-1 multiplexer outputs are fed into the 2-to-1 with the selector pins on the 4-to-1's put in parallel giving a total number of selector inputs to 3, which is equivalent to an 8-to-1.

List of ICs which provide multiplexing

For 7400 series part numbers in the following table, "x" is the logic family.

IC No.	Function	Output State
74x157	Quad 2:1 mux.	Output same as input given
74x158	Quad 2:1 mux.	Output is inverted input
74x153	Dual 4:1 mux.	Output same as input
74x352	Dual 4:1 mux.	Output is inverted input
74x151A	8:1 mux.	Both outputs available (i.e., complementary outputs)
74x151	8:1 mux.	Output is inverted input
74x150	16:1 mux.	Output is inverted input

Signetics S54S157 quad 2:1 mux

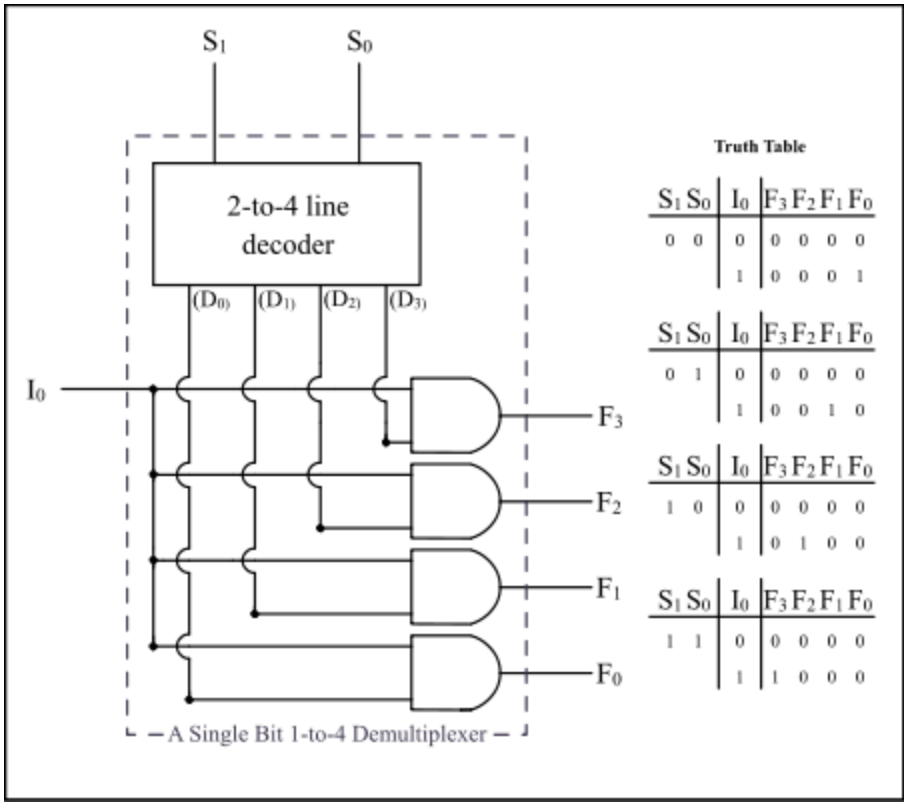
Digital demultiplexers

Demultiplexers take one data input and a number of selection inputs, and they have several outputs. They forward the data input to one of the outputs depending on the values of the selection inputs. Demultiplexers are sometimes convenient for designing general-purpose logic because if the demultiplexer's input is always true, the demultiplexer acts as a binary decoder. This means that any function of the selection bits can be constructed by logically OR-ing the correct set of outputs.

If X is the input and S is the selector, and A and B are the outputs:

$$A = (X \wedge \neg S)$$

$$B = (X \wedge S)$$



Example: A Single Bit 1-to-4 Line Demultiplexer

List of ICs which provide demultiplexing

For 7400 series part numbers in the following table, "x" is the logic family.

IC No. (7400)	IC No. (4000)	Function	Output State
74x139		Dual 1:4 demux.	Output is inverted input
74x156		Dual 1:4 demux.	Output is open collector
74x138		1:8 demux.	Output is inverted input
74x238		1:8 demux.	
74x154		1:16 demux.	Output is inverted input
74x159	CD4514/15	1:16 demux.	Output is open collector and same as input

Fairchild 74F138 1:8 demultiplexer

Multiplexers as PLDs

Multiplexers can also be used as programmable logic devices, to implement Boolean functions. Any Boolean function of *n* variables and one result can be implemented with a multiplexer with *n* selector inputs. The variables are connected to the selector inputs, and the function result, 0 or 1, for each possible combination of selector inputs is connected to the corresponding data input. If one of the variables (for example, *D*) is also available inverted, a multiplexer with *n*-1 selector inputs is sufficient; the data inputs are connected to 0, 1, *D*, or ~*D*, according to the desired output for each combination of the selector inputs.[6]

See also

- Digital subscriber line access multiplexer (DSLAM)
- Inverse multiplexer
- Multiplexing
  - Code-division multiplexing
  - Frequency-division multiplexing
  - Time-division multiplexing
  - Wavelength-division multiplexing
  - Statistical multiplexing
  - Charlieplexing
- Priority encoder
- Rule 184, a cellular automaton in which each cell acts as a multiplexer for the values from the two adjacent cells

- Statistical multiplexer

References

1. Dean, Tamara (2010). *Network+ Guide to Networks* (https://books.google.com/books?id=UD0h_GqgbHgC&q=network%2B+guide+to+networks). Delmar. pp. 82–85. ISBN 978-1423902454.
2. Debashis, De (2010). *Basic Electronics* (https://books.google.com/books?id=mT_j4F1bJx4C&q=Basic+Electronics+By+De+Debashis). Dorling Kindersley. p. 557. ISBN 9788131710685.
3. Lipták, Béla (2002). *Instrument engineers' handbook: Process software and digital networks* (https://books.google.com/books?id=KPjLAyA7HgoC&q=instrument+engineers'+handbook:+Process+software+and+digital+networks+By+B%C3%A9la+G.+Lipt%C3%A1k). CRC Press. p. 343. ISBN 9781439863442.
4. Harris, David (2007). *Digital Design and Computer Architecture* (https://books.google.com/books?id=5X7JV5-n0FIC&q=Digital+design+and+computer+architecture+By+David+Money+Harris,+Sarah+L.+Harris). Penrose. p. 79. ISBN 9780080547060.
5. Crowe, John; Hayes-Gill, Barrie (1998). "The multiplexer hazard" (https://www.google.com.au/books/edition/Introduction_to_Digital_Electronics/97w8luwEIAsC?hl=en&pg=PA111). *Introduction to Digital Electronics*. Elsevier. pp. 111–3. ISBN 9780080534992.
6. Lancaster, Donald E. (1974). *The TTL Cookbook*. H.W. Sams. pp. 140–3. ISBN 9780672210358.

Further reading

- Mano, M. Morris; Kime, Charles R. (2008). *Logic and Computer Design Fundamentals* (4th ed.). Prentice Hall. ISBN 978-0-13-198926-9.

External links

- The dictionary definition of *multiplexer* at Wiktionary
