Lecture 9B MUX/DMUX Tree

- One application example of MUX/DMUX
  - One data line + 3 selection line (GND not shown)
  - Time-multiplexed transmission

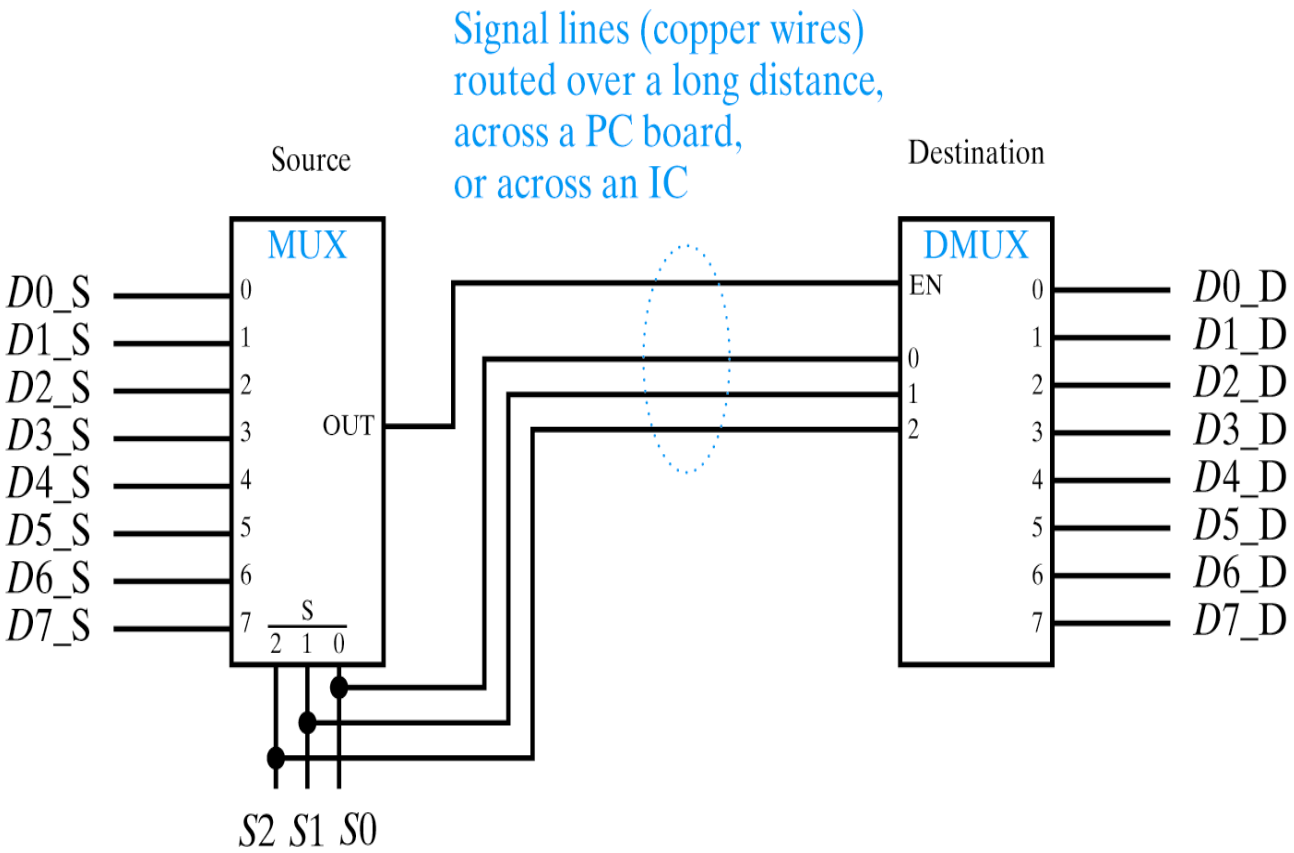


Figure 4.3.1 MUX/DMUX circuit for a communication link.

How to design a large MUX and DMUX

- Why large MUX, for instance 1024 inputs (needs 10 select lines)?
- Using truth tables (tedious)
- Iterative modular design
  - Use a smaller MUX to form a larger MUX or MUX tree.

MUXs

- Symbols
- Objectives: we want to design a larger MUX using smaller MUXs.
  - This increases the number of gate levels.

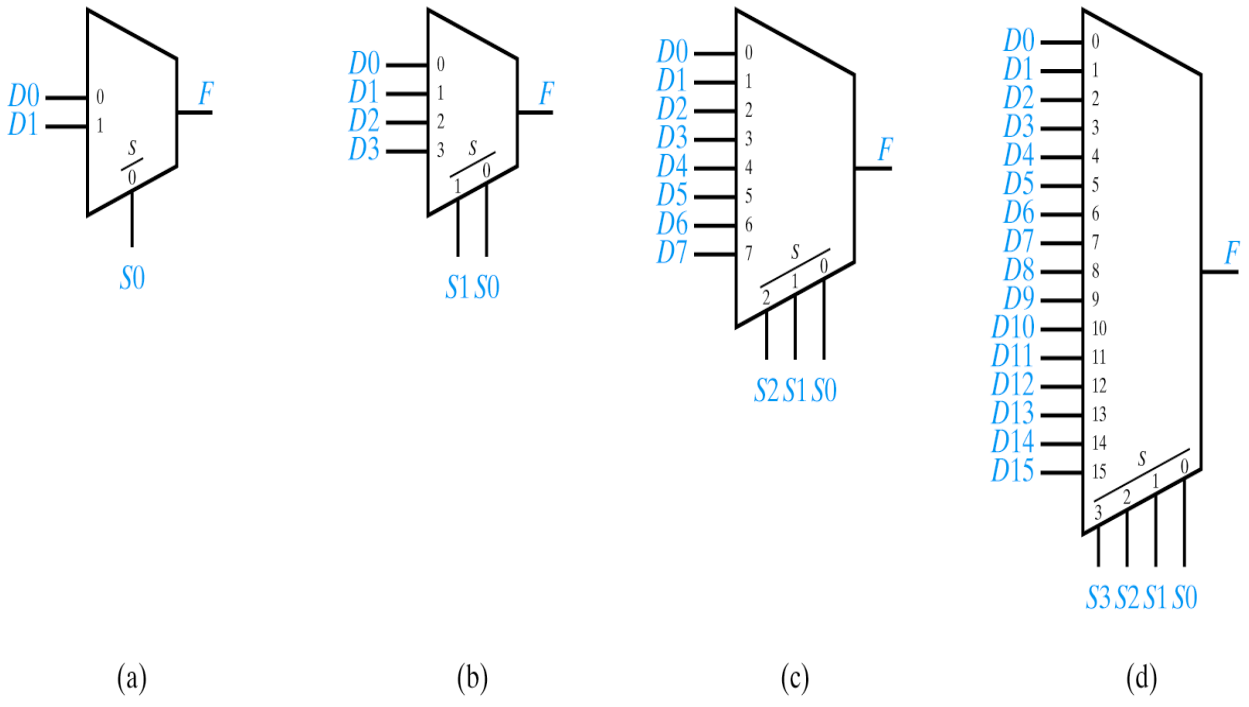
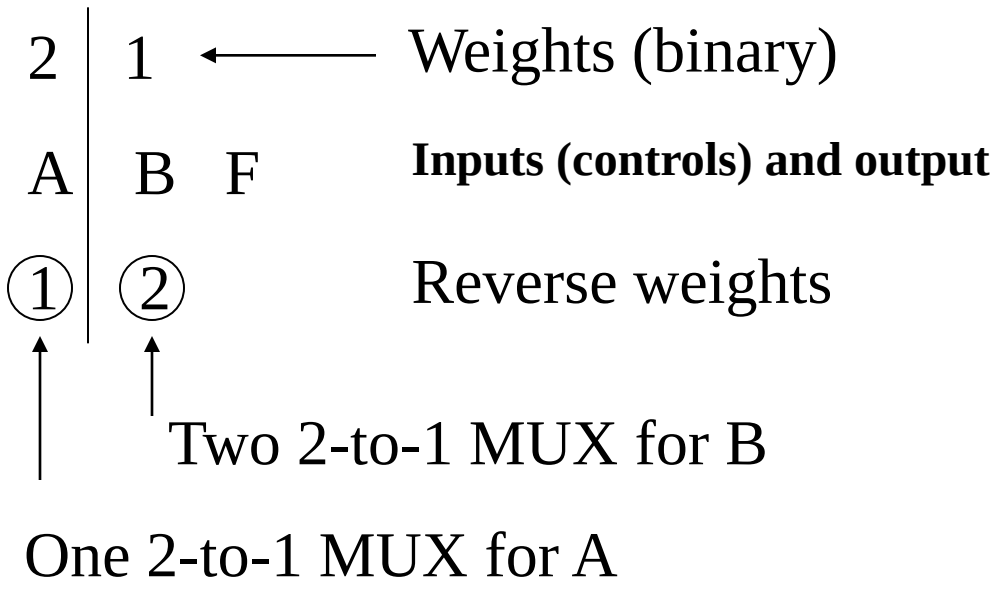


Figure 4.3.2 Graphic symbols for MUXs: (a) 2-to-1 MUX, (b) 4-to-1 MUX, (c) 8-to-1 MUX, and (d) 16-to-1 MUX.

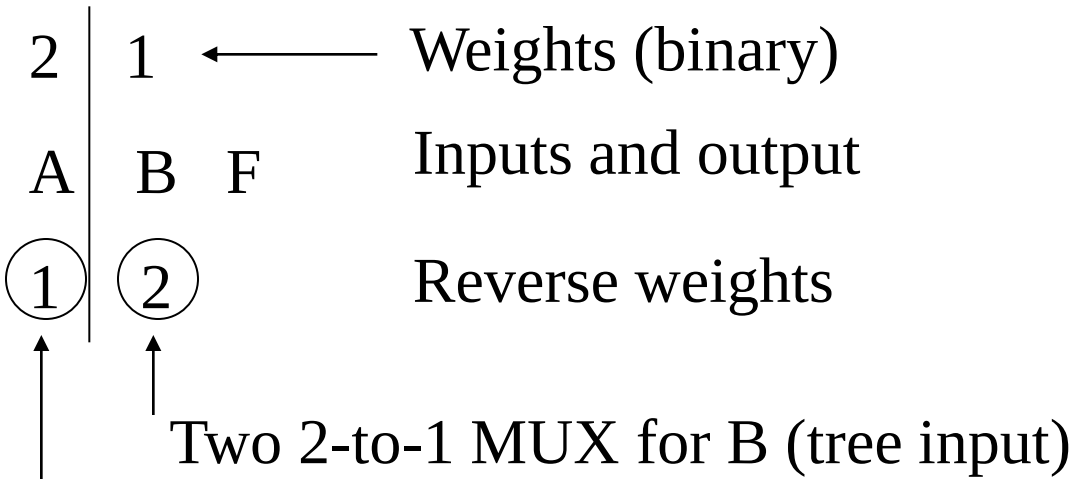
MUX Tree

- Example: design a 4-to-1 MUX using only 2-to-1 MUXs
  - Determine the number of select inputs. $2^n = 4$, n = 2.
  - Input (control) A and B, output F
  - Construct this table:



MUX Tree

- Example



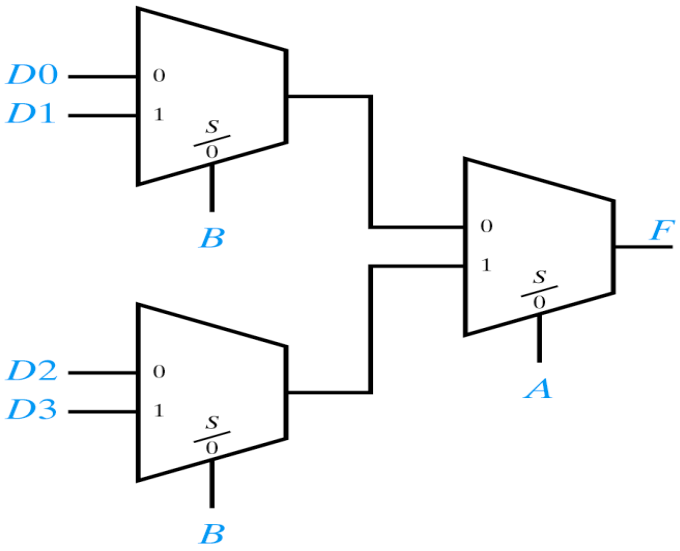
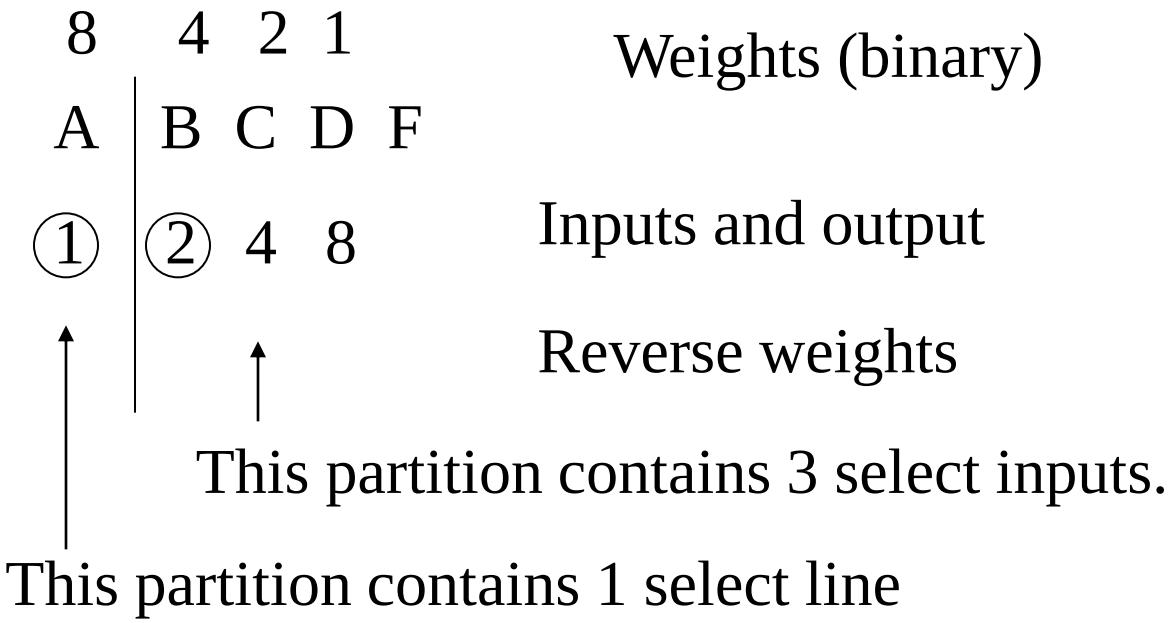


Figure 4.3.3 Modular logic circuit for a 4-to-1 MUX tree implemented with 2-to-1 MUXs.

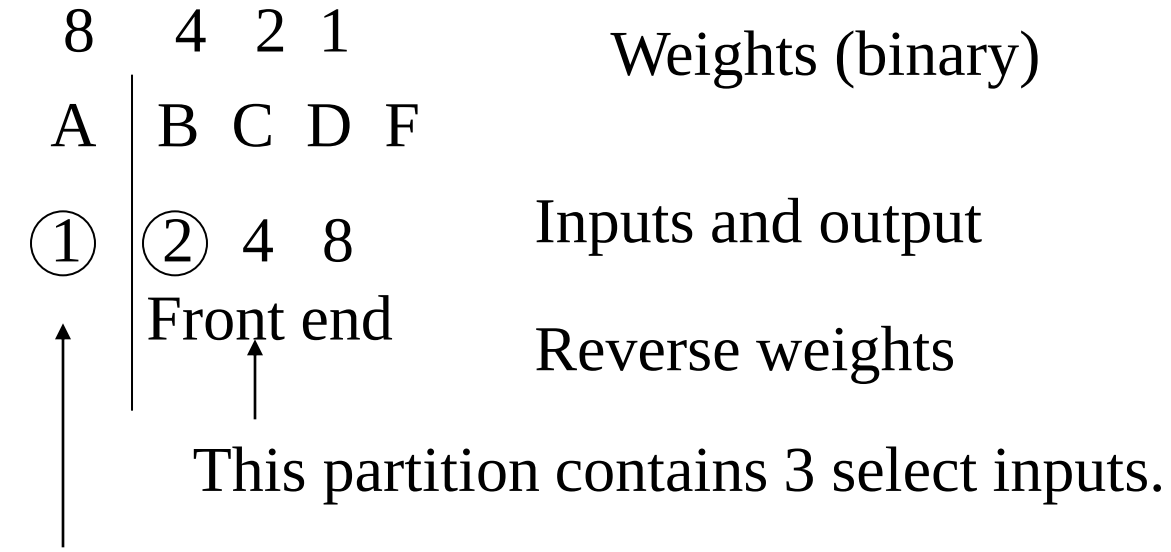
16-to-1 Mux

- Example: design a MUX tree for 16 select lines (data inputs)
- The circled numbers indicate the number of muxs for each partition.
- The number of input (control) lines determines the size of the Mux required.



16-to1 Mux

- The first partition contains input select line A. So this is a 2-to-1 Mux.
- The second partition contains 3 inputs (control). So this is a 8-to-1 Mux



16-to-1 Mux

- The first partition contains input select line A. So this is a 2-to-1 Mux.
- The second partition contains 3 input select lines. So this is a 8-to-1 Mux

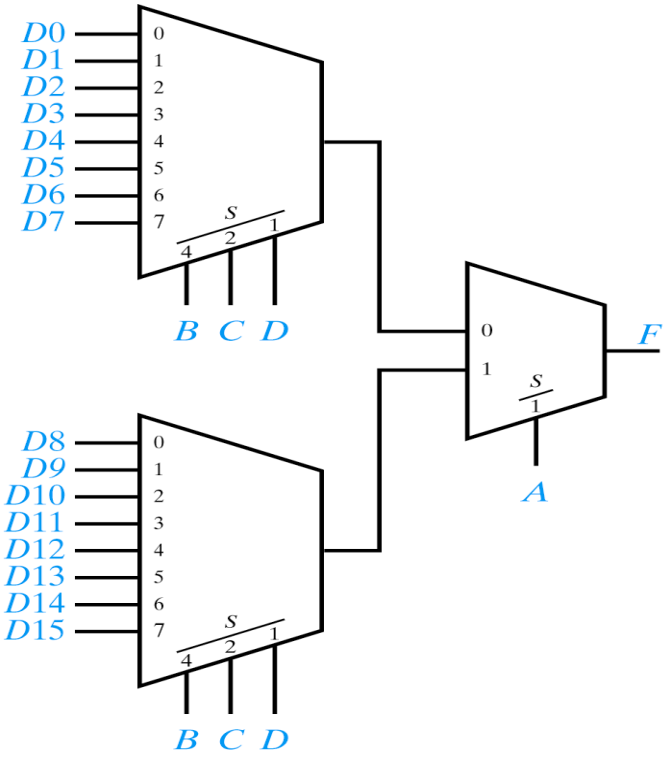


Figure 4.3.4 Modular logic circuit for a 16-to-1 MUX tree implemented with one 2-to-1 MUX and two 8-to-1 MUXs.

64-to-1 Mux

- The input select lines = 6.
- We want to do it in three parts.
  - First part has input U V (this is on the output of the tree) and needs 1 4-to-1 Mux.
  - Second has inputs W X and needs 4 4-to-1 Muxs.
  - Third part has input Y Z and needs 16 4-to-1 Muxs.

32	16	8	4	2	1		Weights (binary)
U	V	W	X	Y	Z	F	Inputs and output
1	2	4	8	16	32		Reverse weights

64-to-1 Mux

32	16	8	4	2	1	Weights (binary)
U	V	W	X	Y	Z	F	Inputs and output
(1)	2	(4)	8	(16)	32	Reverse weights

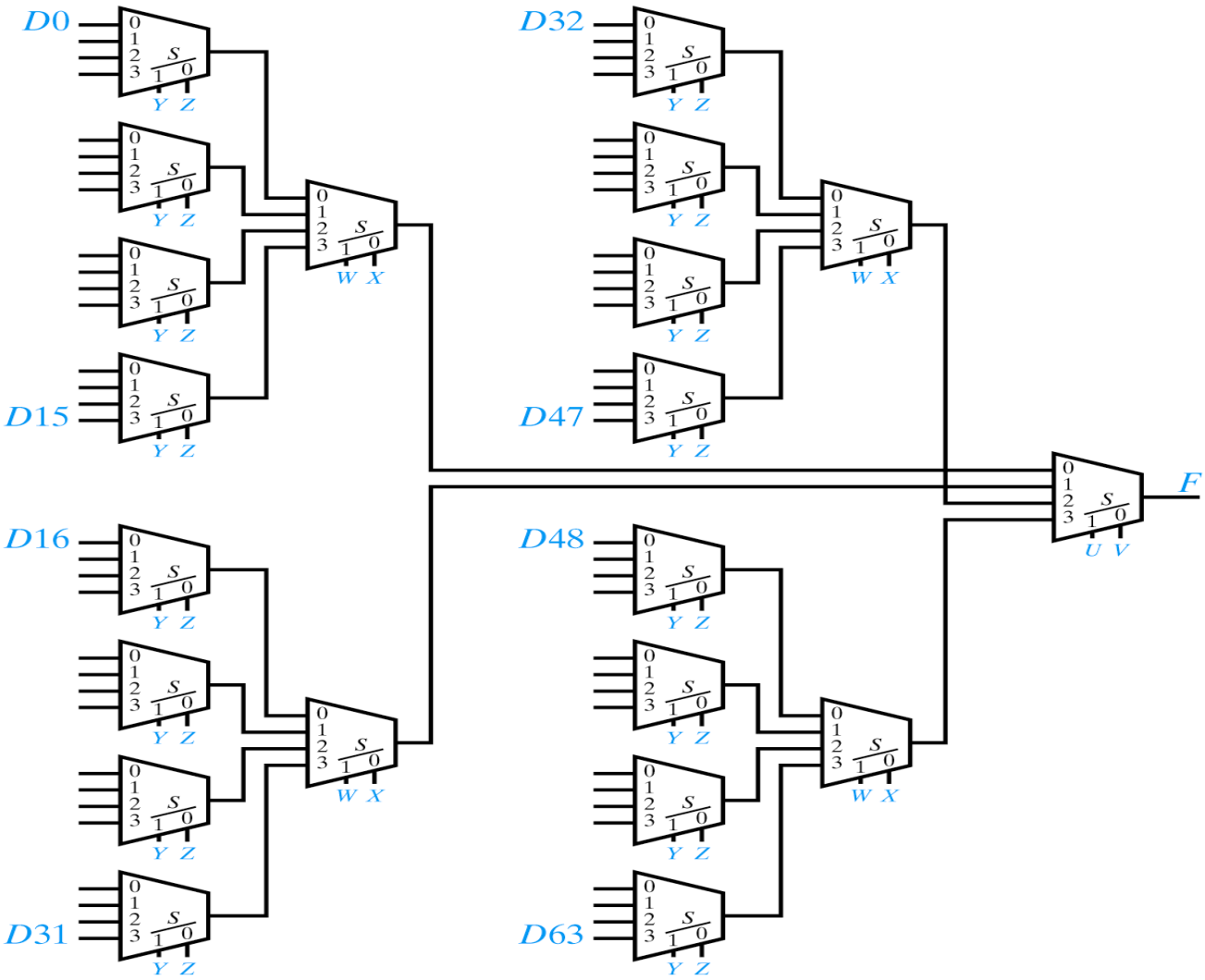


Figure 4.3.5 Modular logic circuit for a 64-to-1 MUX tree implemented with 4-to-1 MUXs.

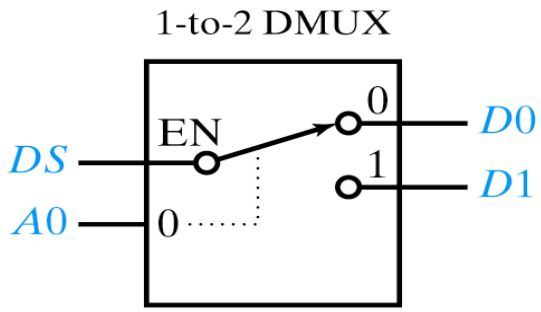
64-to-1 Mux

- Using 8-to-1 Muxs.
- Tree back end needs one 8-to-1 Mux.
- Tree front end needs eight 8-to-1 Muxs.

32	16	8	4	2	1		Weights (binary)
U	V	W	X	Y	Z	F	Inputs and output
(1)	2	4	(8)	16	32		Reverse weights

DMUX

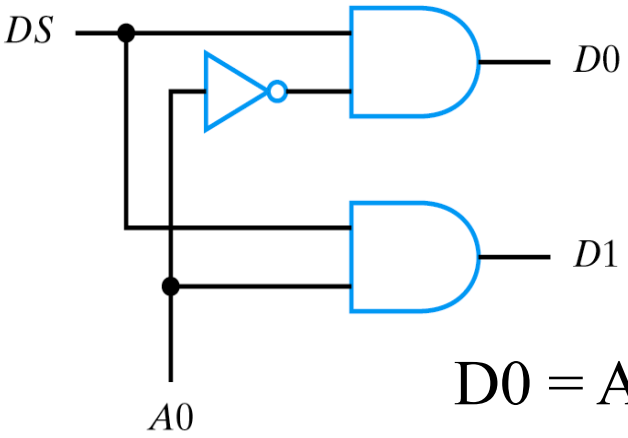
- Demultiplexer is a decoder with an enable input.



A0	D0	D1
0	DS	0
1	0	DS

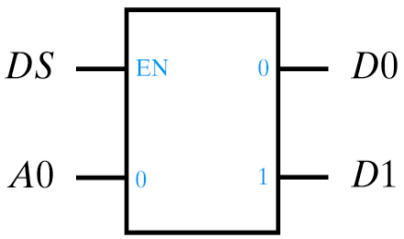
(a) (b)

Figure 4.3.6 A 1-to-2 DMUX: (a) simplified switch circuit, and (b) compressed truth table.



D0 = A0' DS

D1 = A0 DS



(a) (b)

Figure 4.3.7 A 1-to-2 DMUX: (a) gate-level circuit, and (b) graphic symbol.

DMUX

- Demultiplexer is a decoder with an enable input.

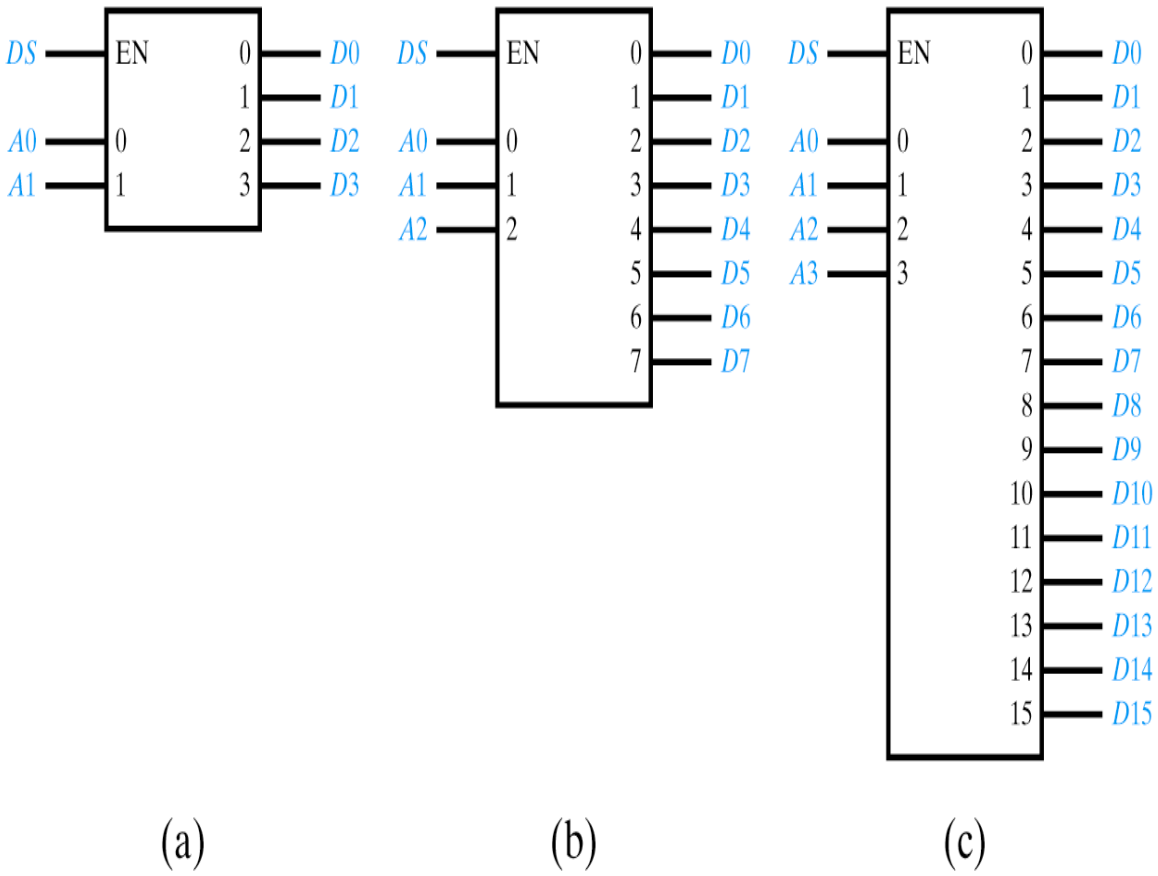
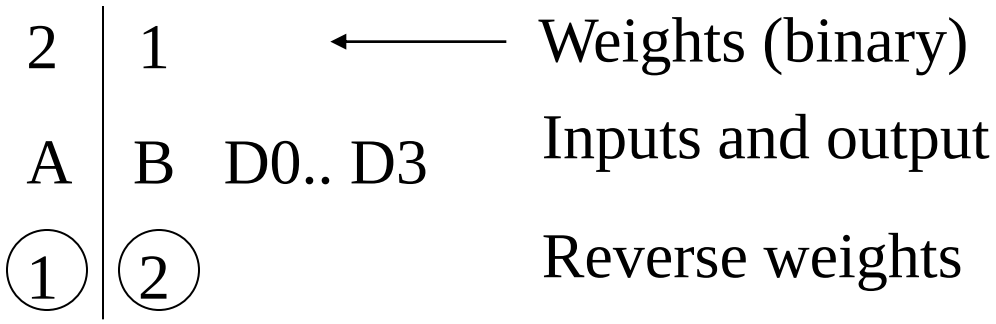


Figure 4.3.8 Graphic symbols for DMUXs: (a) 2-to-4 DMUX, (b) 3-to-8 DMUX, and (c) 4-to-16 DMUX.

DMUX tree

- 2-to-4 DMUX using 1-to-2 DMUXs
- Two parts
  - The circle number indicates the number of DMUXs required for each part.
  - The number of address inputs determines the size of the DMUX used.



DMUX tree

- 2-to-4 DMUX using 1-to-2 DMUXs

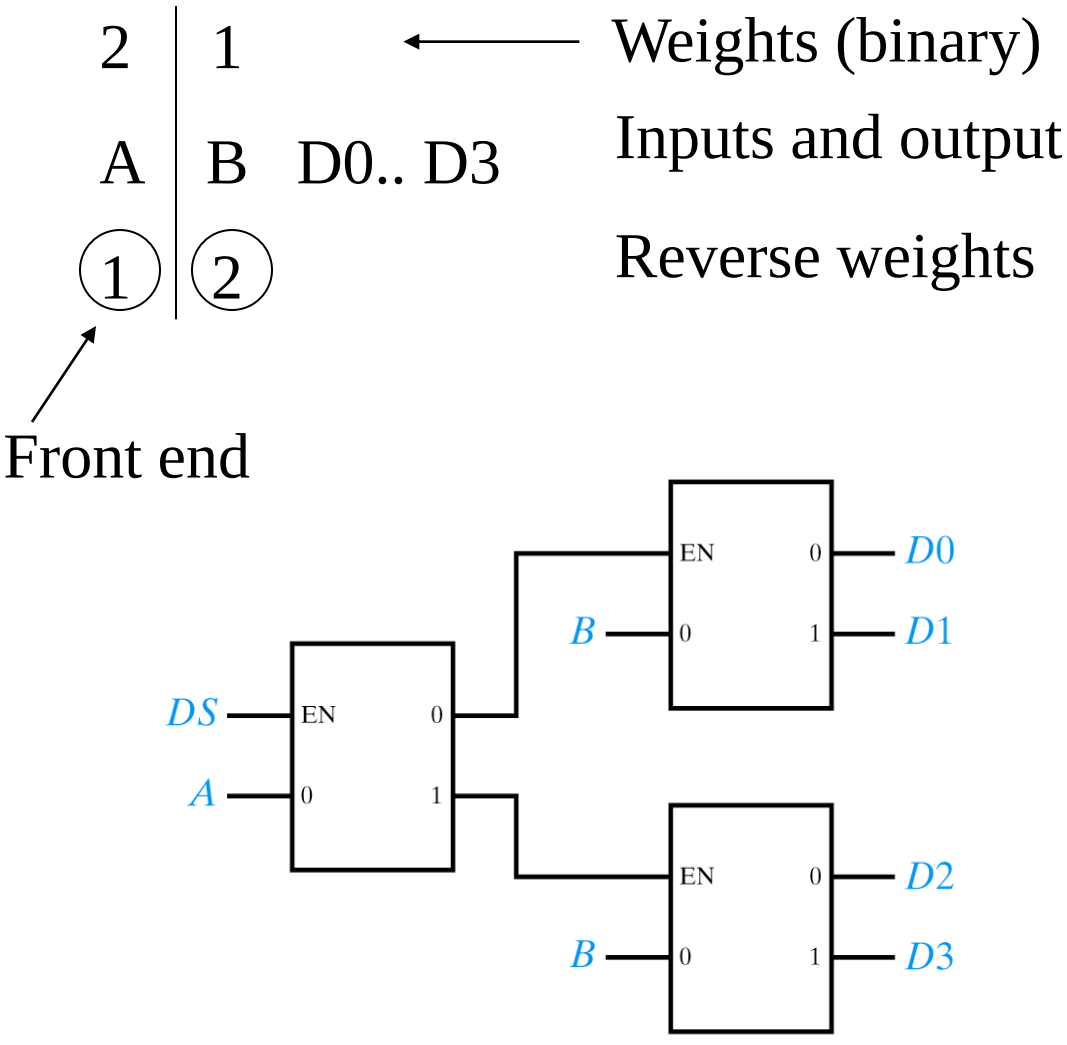


Figure 4.3.9 Modular logic circuit for a 2-to-4 DMUX tree implemented with three 1-to-2 DMUXs.

DMUX tree 16 outputs

- 3 inputs for the front end.

8	4	2	1	←	Weights (binary)
A	B	C	D	D0.. D15	Inputs and output
(1)	2	4	(8)		Reverse weights

Front end

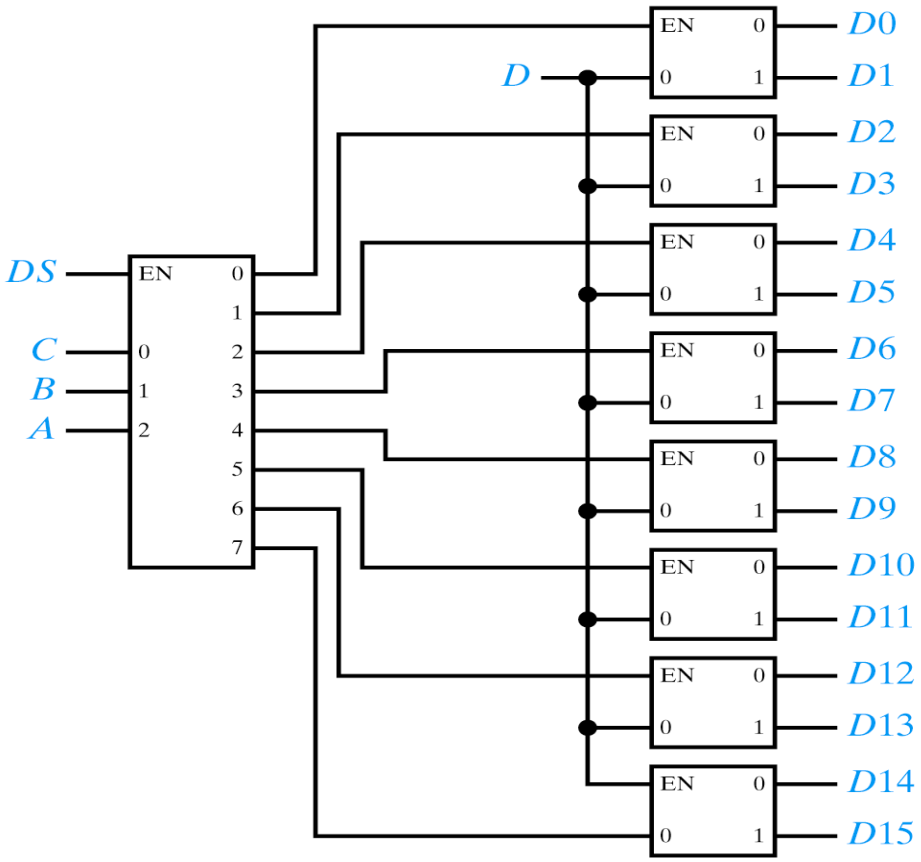


Figure 4.3.10 Modular logic circui for a 4-to-16 DMUX tree implemented with one 3-to-8 DMUX and eight 1-to 2 DMUXs.