Computers Fundamentals

Subject 3. Basic combinatorial blocks

Aims and objectives

- At the end of the subject the student should know how works the following combinatorial circuits
  - Decoders
  - Encoders
  - Multiplexors
  - Demultiplexors

Outline

1. Introduction
2. Decoders
    1. Binary decoders
    2. Composition of binary decoders
    3. Non binary decoders
3. Encoders
4. Multiplexors
    1. Composition of multiplexors
    2. Data multiplexors of *n* bits
5. Demultiplexors

Introduction

- The basic foundations of design and implementation of digital circuits using gates have been studied in subject 2.
- In this subject such foundations will be applied in order to understand how works and how are implemented the most used combinatorial circuits.

Introduction

- Basic combinatorial circuits implement simple circuits
  - They can be found implemented into commercial Integrated Circuits (ICs)

- Combinatorial circuits are very important because they are the basic building blocks from which the functional units of a computer are designed

Introduction

FCO

- Combinatorial circuit:
  - The relationship between the inputs and the outputs of combinatorial circuits can be expressed as a logic function
  - At any given time, the outputs depends only and exclusively on the value of the inputs
- Logic-gates generate delays between the inputs and the outputs (measured in nanosec.)
  - In real combinatorial circuits, changes in the inputs are manifested on the outputs after a delay
  - The delay depends on the technology used to implement logic, the level of the circuit and the kind of logic gates used.

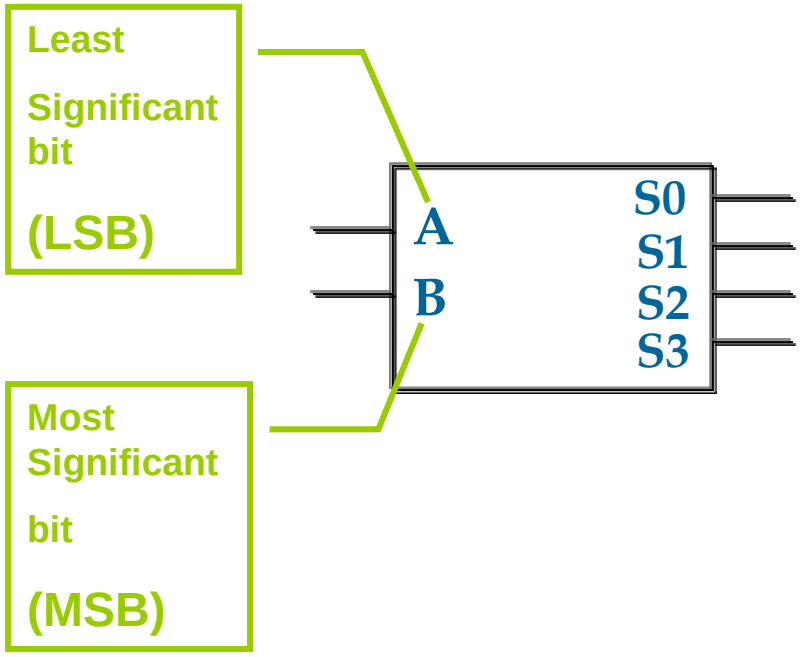
Decoders

FCO

- Binary decoders
  - m inputs, n = 2^m outputs (2 to 4, 3 to 8, 4 to 16)
  - They are very useful for enabling circuits
- Decoders from BCD to 7 segments LEDS
  - 4 inputs, 7 outputs
- Decoders from BCD to decimal
  - 4 inputs,10 outputs

Decoders

- Binary decoders



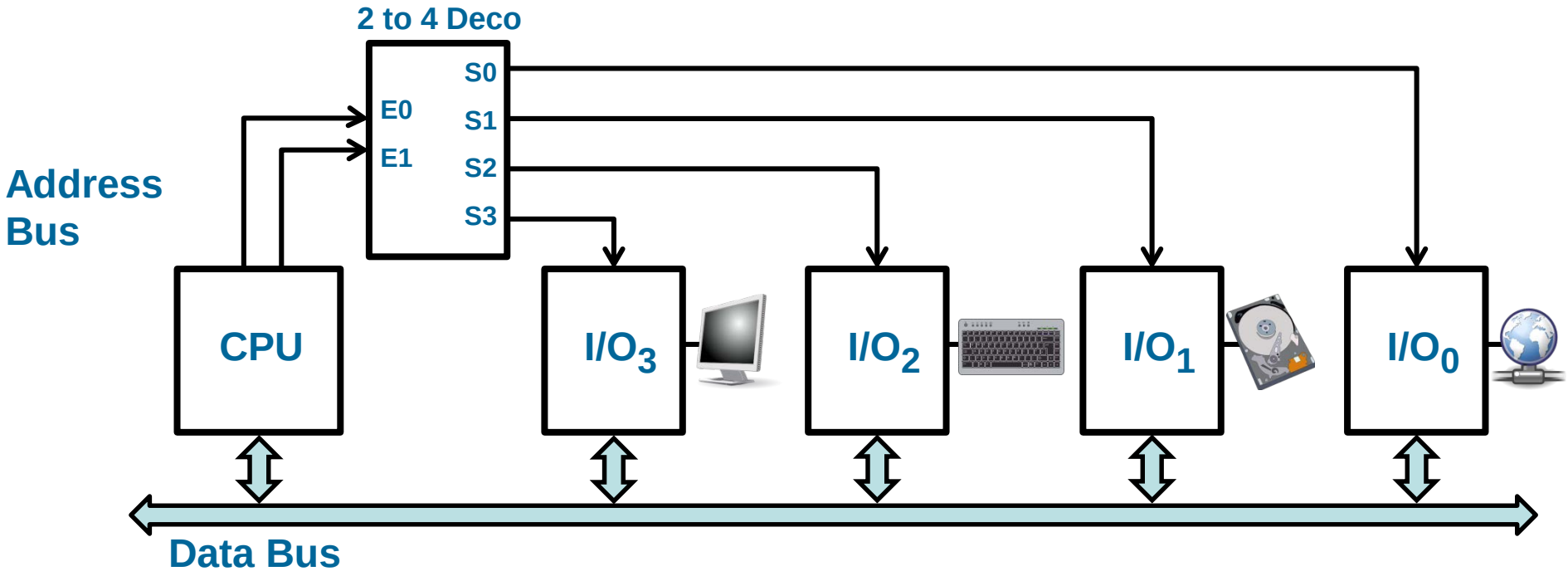
Inputs		Outputs			
B	A	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Ouputs are mutually exclusive

Decoders

FCO

– Binary decoders used to enable devices:



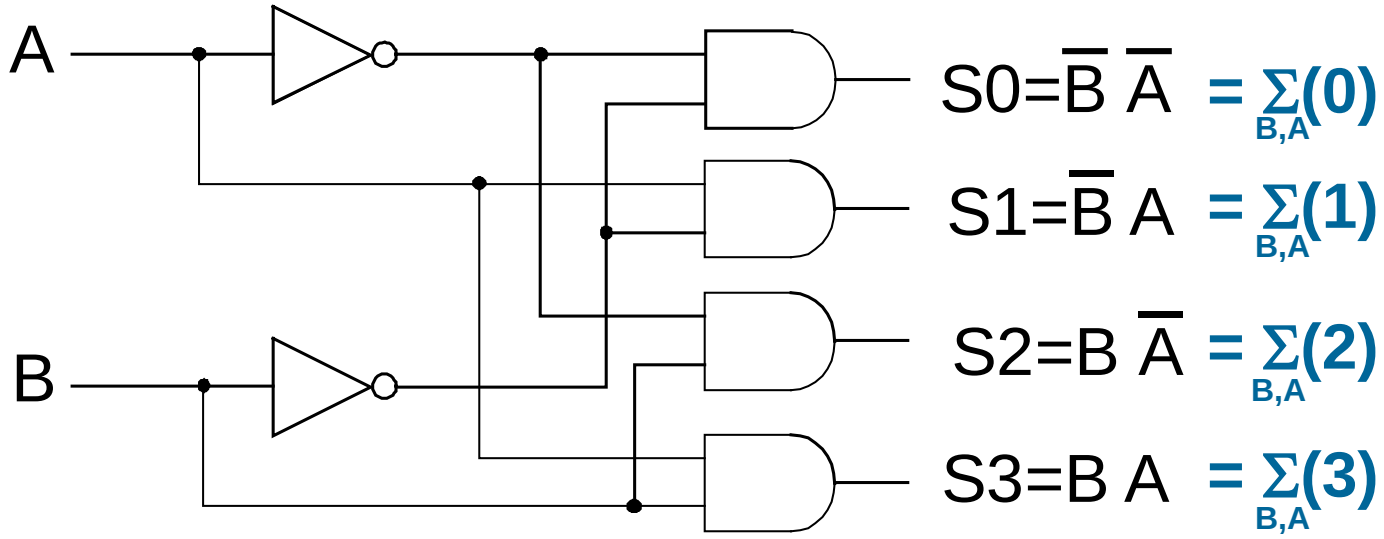
DISCA

Decoders

- Design of a binary decoder

Inputs		Outputs			
B	A	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

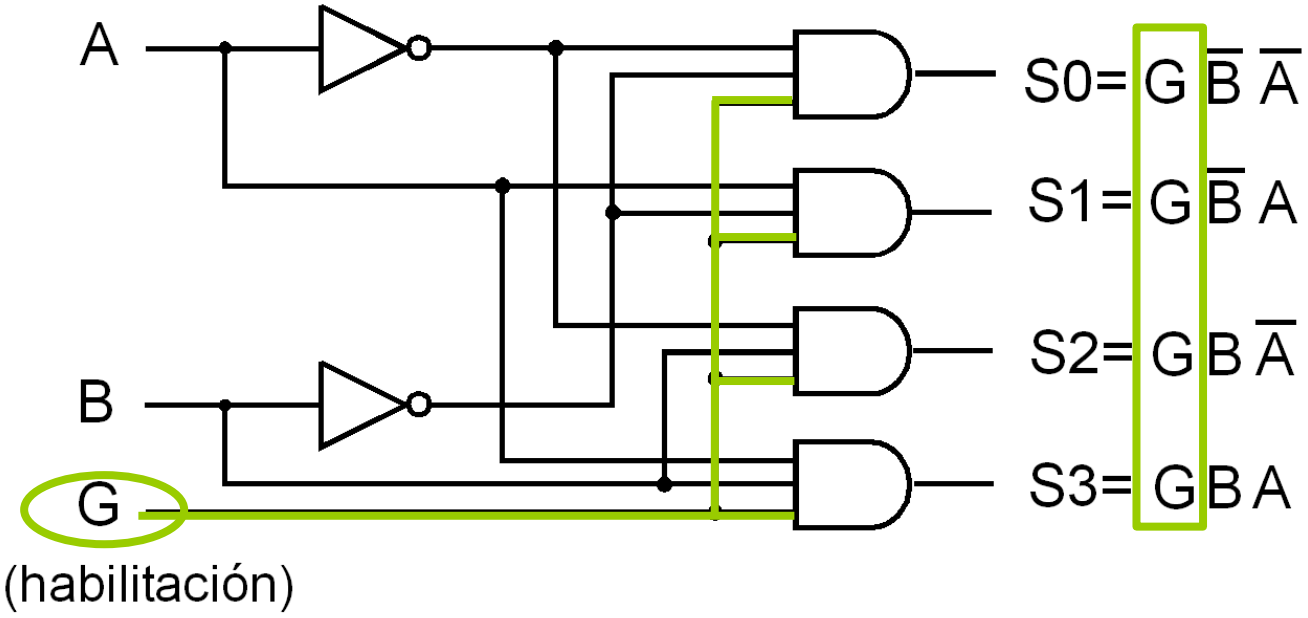
Note that each output only has one valuation equal to 1, so the minimal expression is obtained expanding the canonical form obtained from the sum of products



Decoders

- Decoder with enable input
  (Enable or strobe)

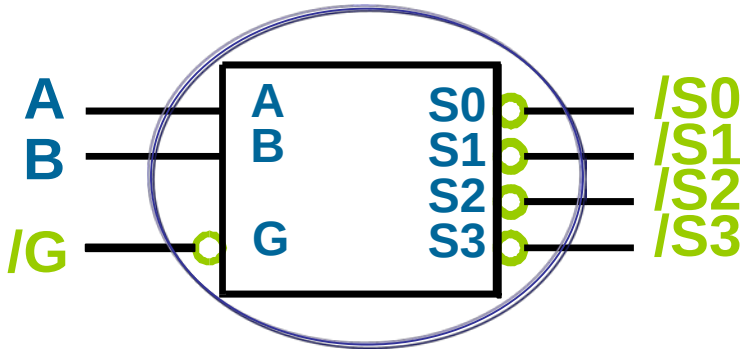
Inputs			Outputs			
G	B	A	S3	S2	S1	S0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



Decoders

FCO

- Integrated circuit 74LS139



Inputs			Outputs			
/G	B	A	/S3	/S2	/S1	/S0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

Note that the enable input and the outputs are low-level activated.

Low –level activated signals are shown:

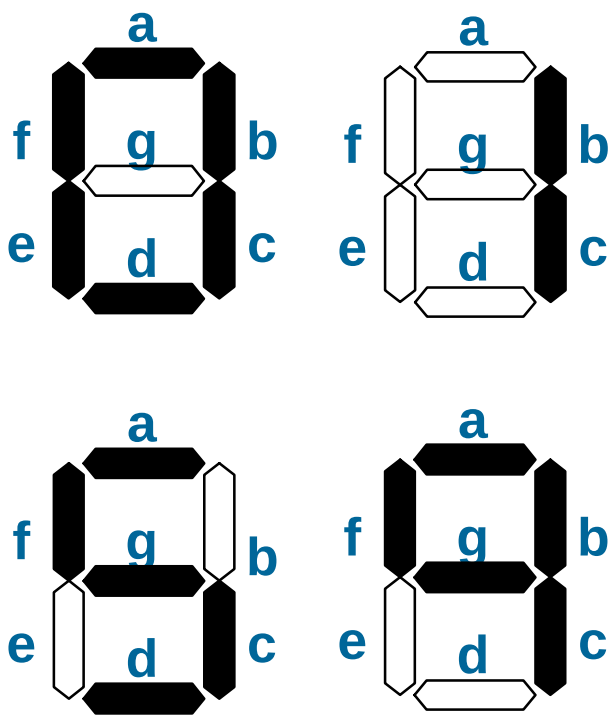
- **using a circle in the logic symbol**
- **Prefixing the signal name with the slash symbol '/'**

DISCA

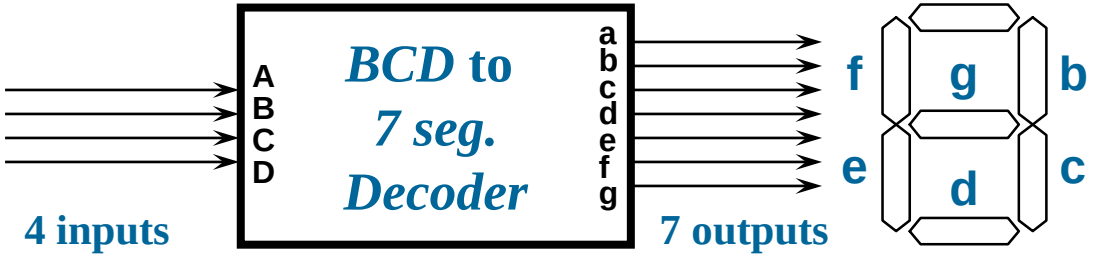
Non binary decoders

FCO

- BCD to 7-segments Decoders (outputs are mutually exclusive)



DECIMAL	Inputs				Outputs						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1



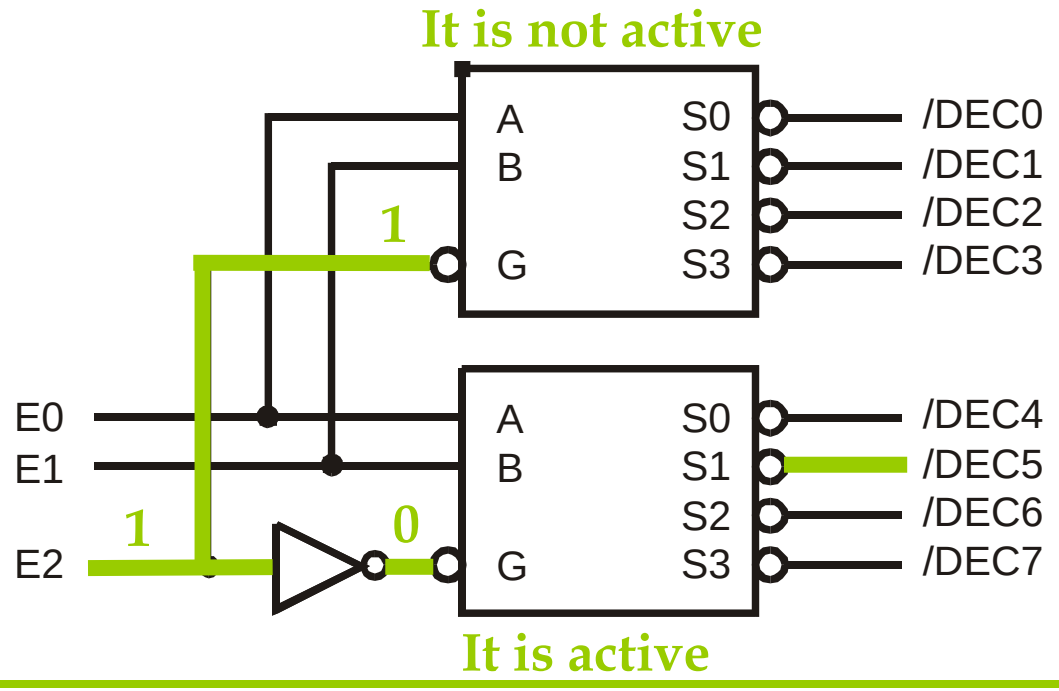
Composition of decoders

FCO

- The biggest commercial decoder circuit is: 4 to 16
- It is possible to implement decoders of bigger size composing smaller decoders

Example: 3 to 8 decoder (from 2 to 4 decoders)

E2	E1	E0	
0	0	0	/DEC0
0	0	1	/DEC1
0	1	0	/DEC2
0	1	1	/DEC3
1	0	0	/DEC4
1	0	1	/DEC5
1	1	0	/DEC6
1	1	1	/DEC7

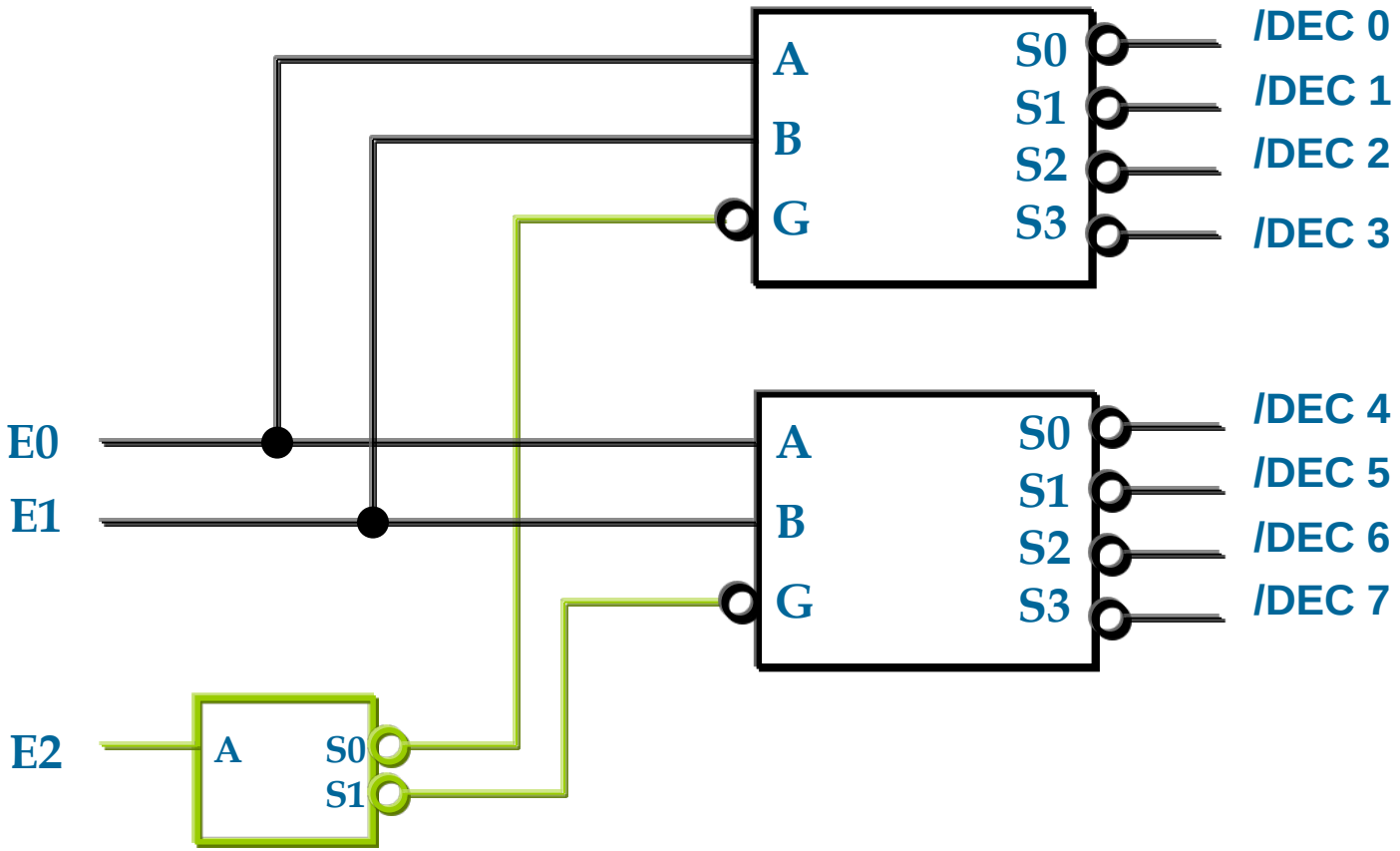


DISCA

Composition of decoders

FCO

Implementation of a 3 to 8 decoder from two 2 to 4 decoders and one 1 to 2 decoder.

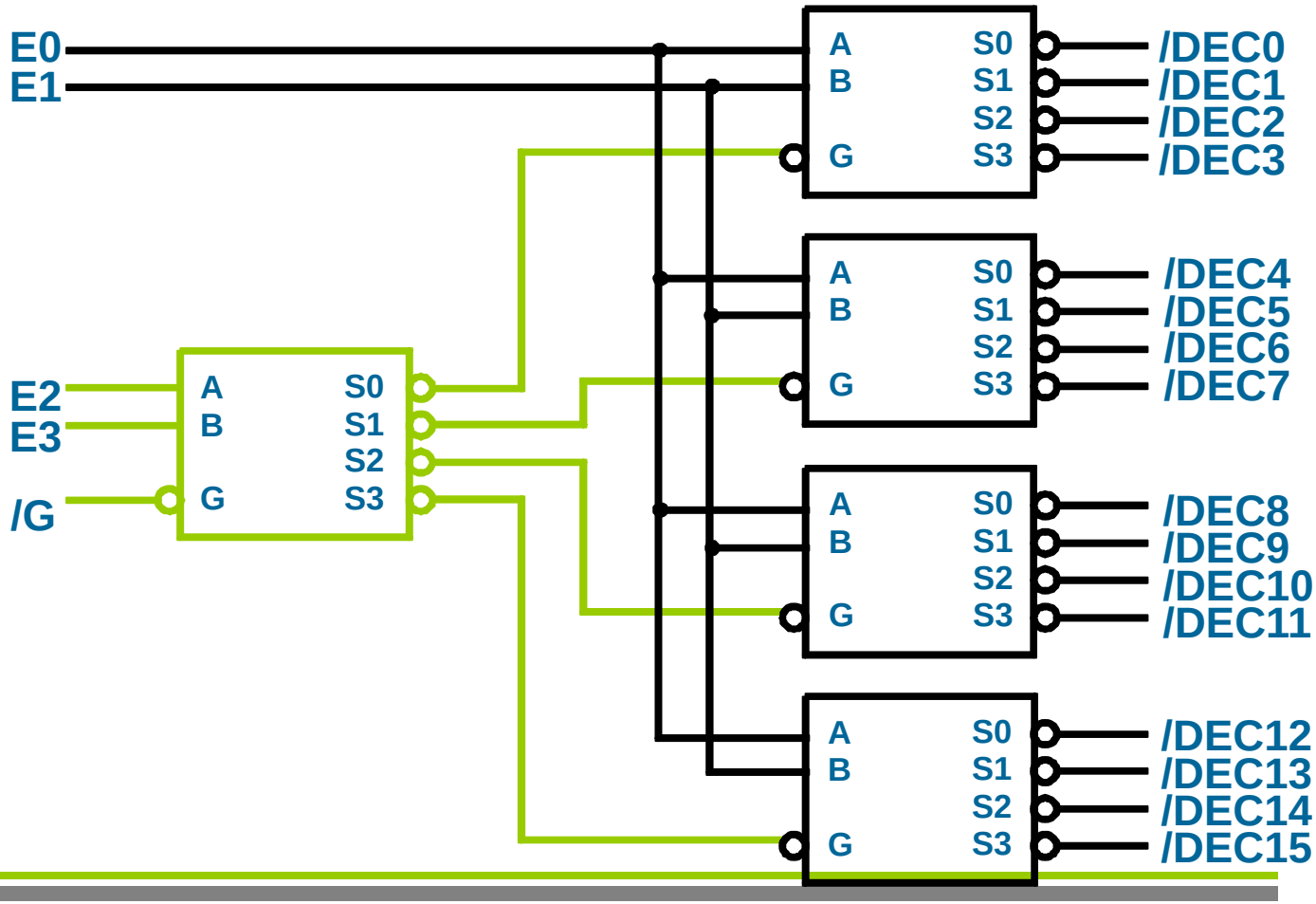


Composition of decoders

FCO

4 to16 decoder from 2to4 decoders (74LS139)

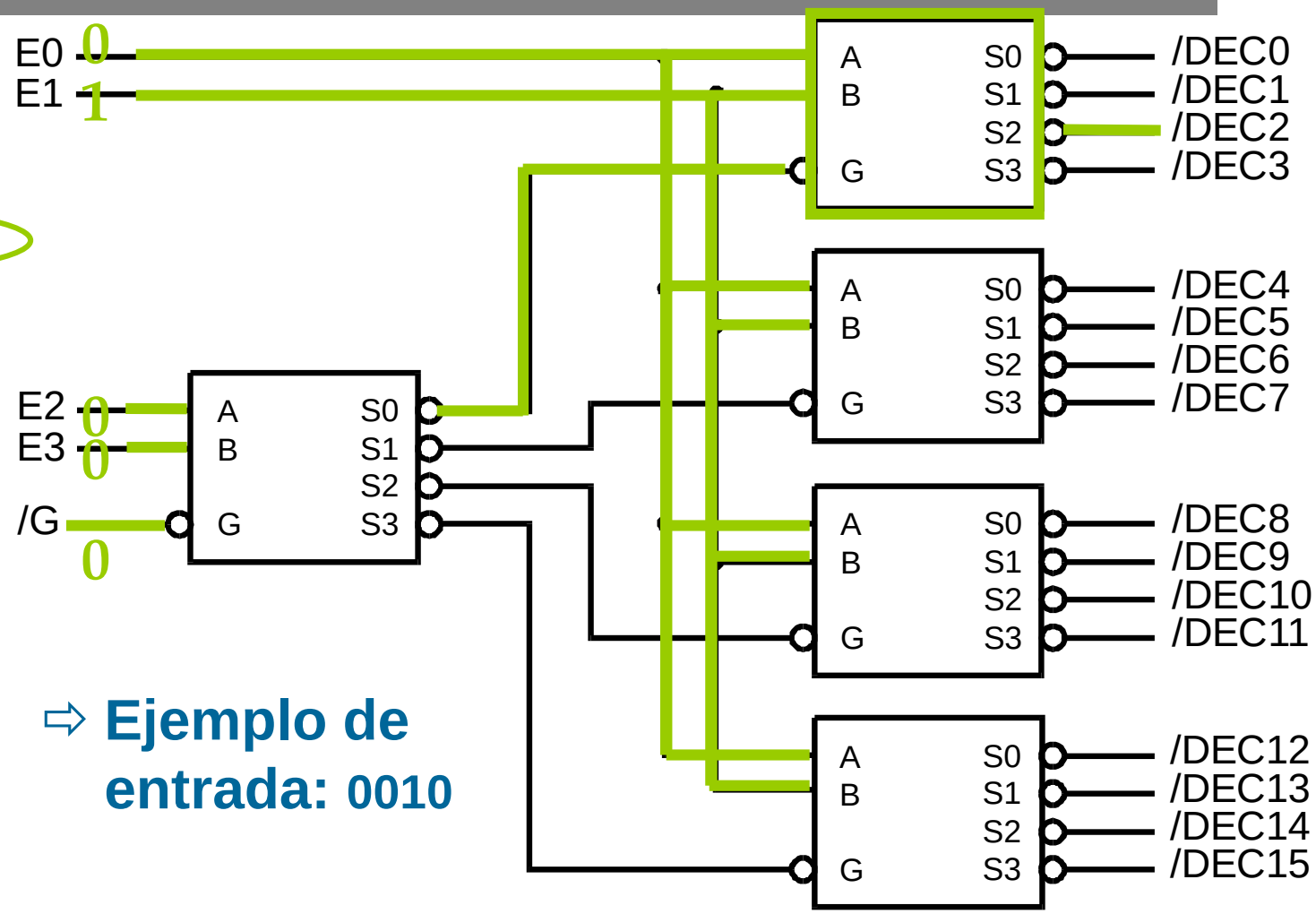
	E3	E2	E1	E0	
(0)	0	0	0	0	/DEC0
(1)	0	0	0	1	/DEC1
(2)	0	0	1	0	/DEC2
(3)	0	0	1	1	/DEC3
(4)	0	1	0	0	/DEC4
(5)	0	1	0	1	/DEC5
(6)	0	1	1	0	/DEC6
(7)	0	1	1	1	/DEC7
(8)	1	0	0	0	/DEC8
(9)	1	0	0	1	/DEC9
(10)	1	0	1	0	/DEC10
(11)	1	0	1	1	/DEC11
(12)	1	1	0	0	/DEC12
(13)	1	1	0	1	/DEC13
(14)	1	1	1	0	/DEC14
(15)	1	1	1	1	/DEC15



Composition of decoders

FCO

	E3	E2	E1	E0	
(0)	0	0	0	0	/DEC0
(1)	0	0	0	1	/DEC1
(2)	0	0	1	0	/DEC2
(3)	0	0	1	1	/DEC3
(4)	0	1	0	0	/DEC4
(5)	0	1	0	1	/DEC5
(6)	0	1	1	0	/DEC6
(7)	0	1	1	1	/DEC7
(8)	1	0	0	0	/DEC8
(9)	1	0	0	1	/DEC9
(10)	1	0	1	0	/DEC10
(11)	1	0	1	1	/DEC11
(12)	1	1	0	0	/DEC12
(13)	1	1	0	1	/DEC13
(14)	1	1	1	0	/DEC14
(15)	1	1	1	1	/DEC15

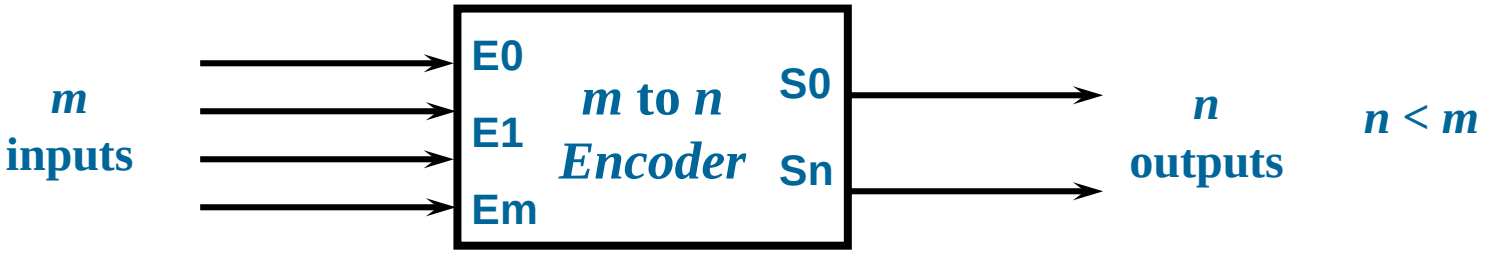


⇨ **Ejemplo de entrada: 0010**

Encoders

FCO

- Encoders realize the opposite function of decoders



– Binary encoder

- $m = 2^n$ inputs and n outputs
- The output string represents the number of the active input in binary format
- Encoders are used in input/output systems

– Example: The output string identifies the device which is demanding the processor attention

- When many devices demands the processor attention it is necessary to establish priorities

DISCA

Encoders

FCO

- Binary encoder (with priority)

Inputs				Outputs	
E3	E2	E1	E0	S1	S0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

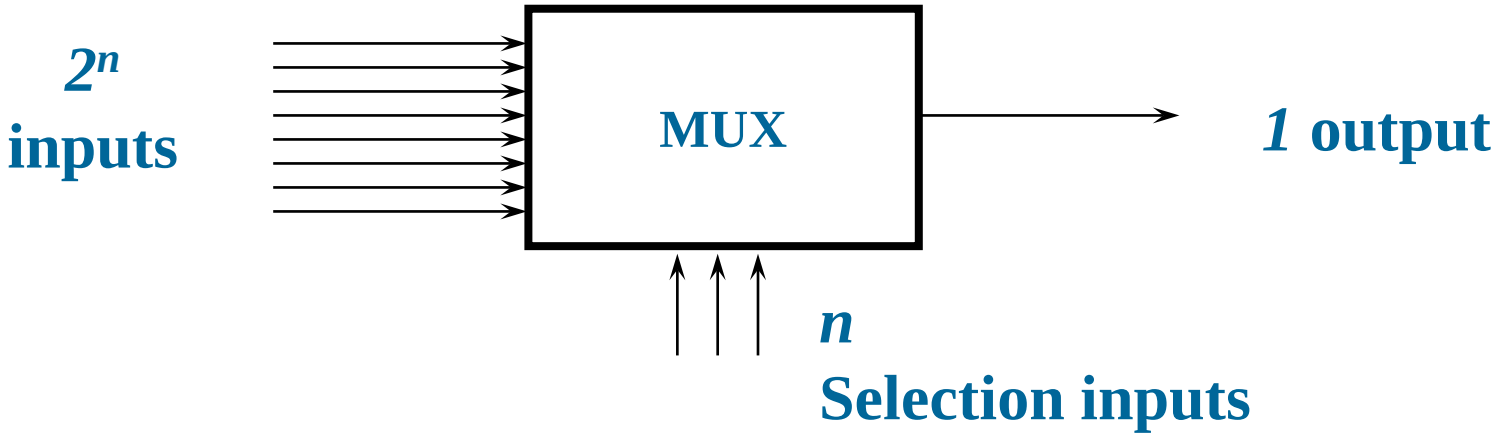
The input of higher weight has priority on those of lower weight

DISCA

Multiplexors

FCO

- Selection inputs are used to connect **one and only one** of the inputs with the output
- They are widely used in data-paths used in computers systems

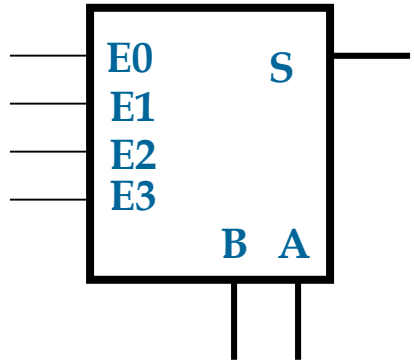


DISCA

Multiplexor

FCO

INPUTS OF SELECTION		OUTPUT
B	A	S
0	0	E0
0	1	E1
1	0	E2
1	1	E3



Inputs of Selection		Inputs of Data				Output
B	A	E3	E2	E1	E0	S
0	0	X	X	X	0	0
0	0	X	X	X	1	1
0	1	X	X	0	X	0
0	1	X	X	1	X	1
1	0	X	0	X	X	0
1	0	X	1	X	X	1
1	1	0	X	X	X	0
1	1	1	X	X	X	1

Extended truth table

DISCA

Multiplexors

FCO

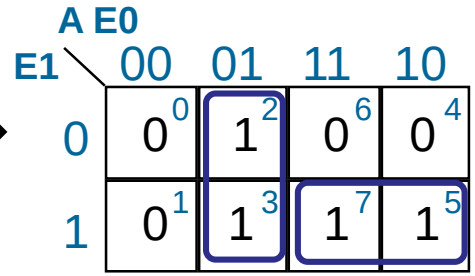
– Example of the design of a 2 data-inputs MUX

A	E0	E1	S
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

Reduced truth table

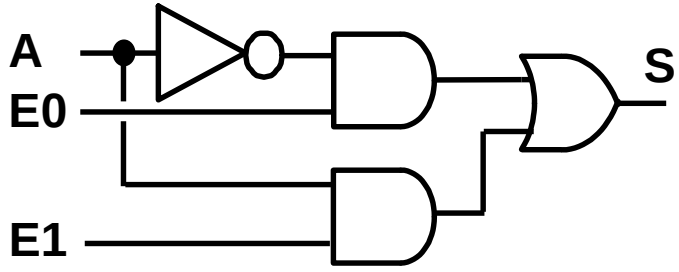
	A	E0	E1	S
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Extended truth table



Karnaugh's map

$S = \overline{A}\,E0 + A\,E1$

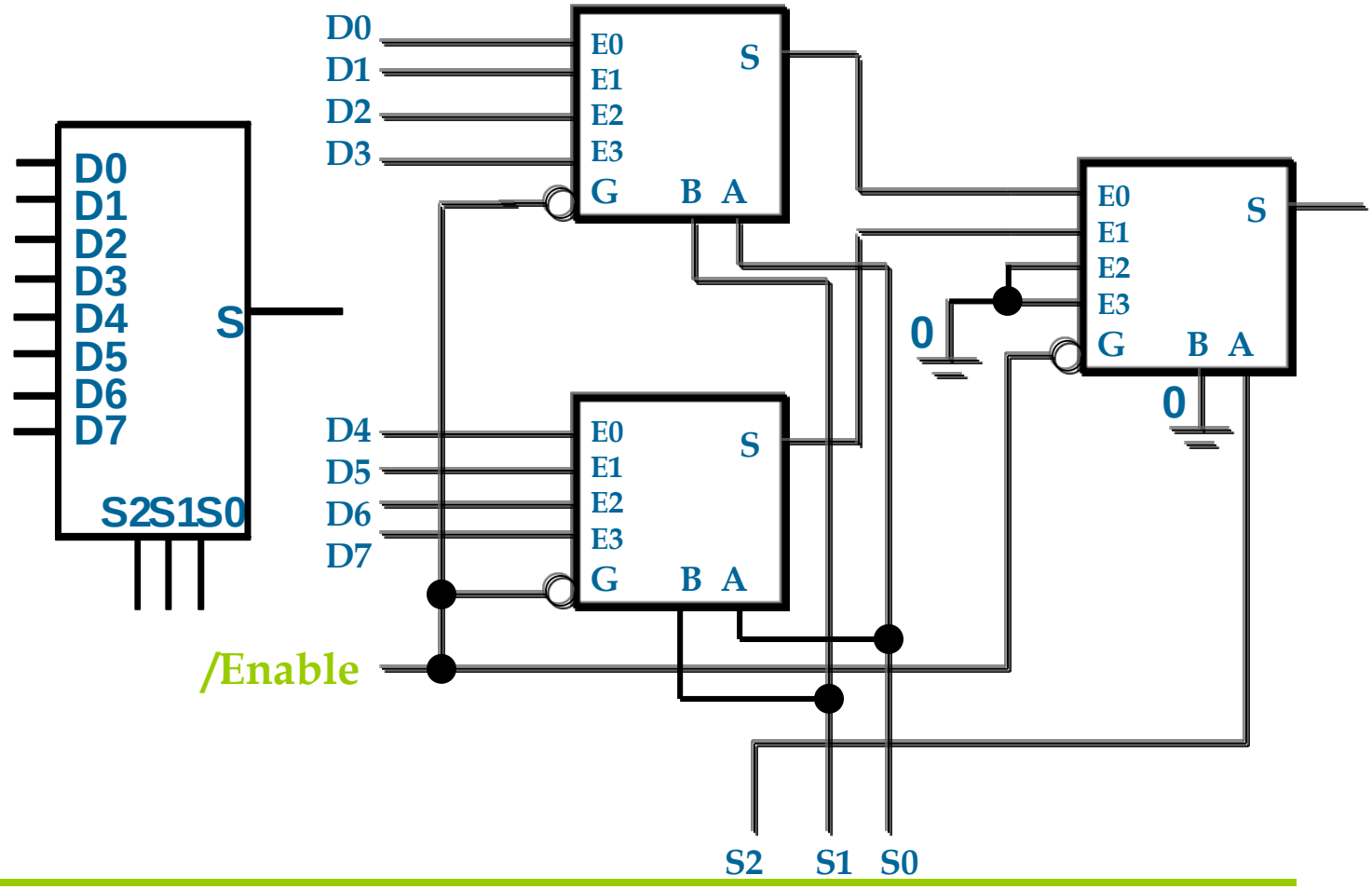


Composition of multiplexers

FCO

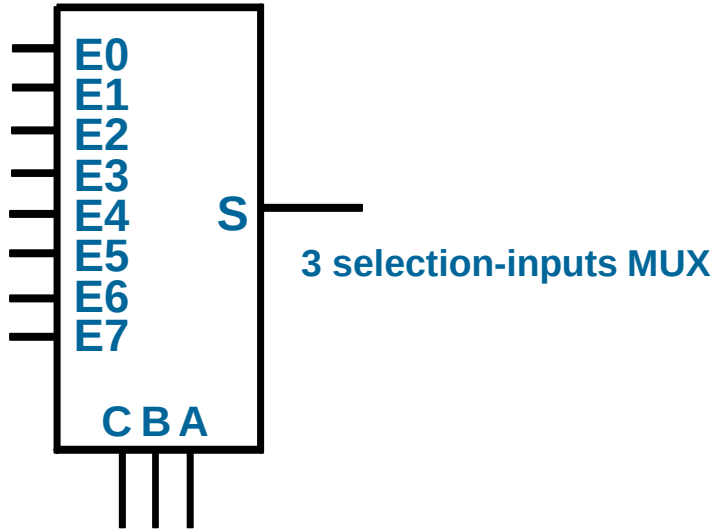
8 data-inputs MUX built from 4 data-inputs MUX´s

S2	S1	S0	S
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



Composition of multiplexers

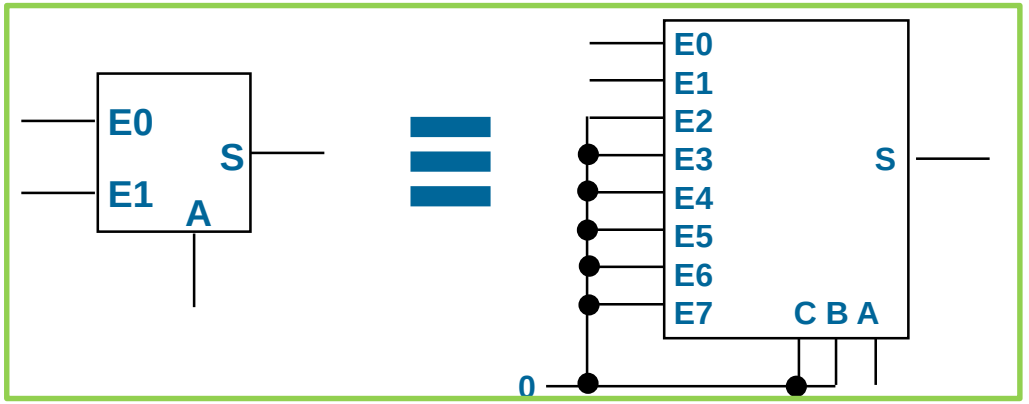
- Method:
  - It is desired to implement a 1024 data inputs MUX. It can be used 3 selection-inputs MUX as many as necessary.
  - How many 3 selection-inputs MUX are necessary?
  - How the 3 selection-inputs MUX should be connected?



Composition of multiplexers FCO

- To cover 1024 data-inputs MUX from 8 data-inputs MUX
  we need 1024 / 8 = 128 8 data-inputs MUX in the first level.
- To cover the 128 outputs of the Muxes of the first level
  we need 128 / 8 = 16 128 8 data-inputs MUX in the second level.
- To cover the 16 outputs of the Muxes of the second level
  we need 16 / 8 = 2 8 data-inputs MUX in the third level.
- To cover the outputs of the muxes of the third level
  we only need one 1 mux.
  - The last mux can be configures as follows:

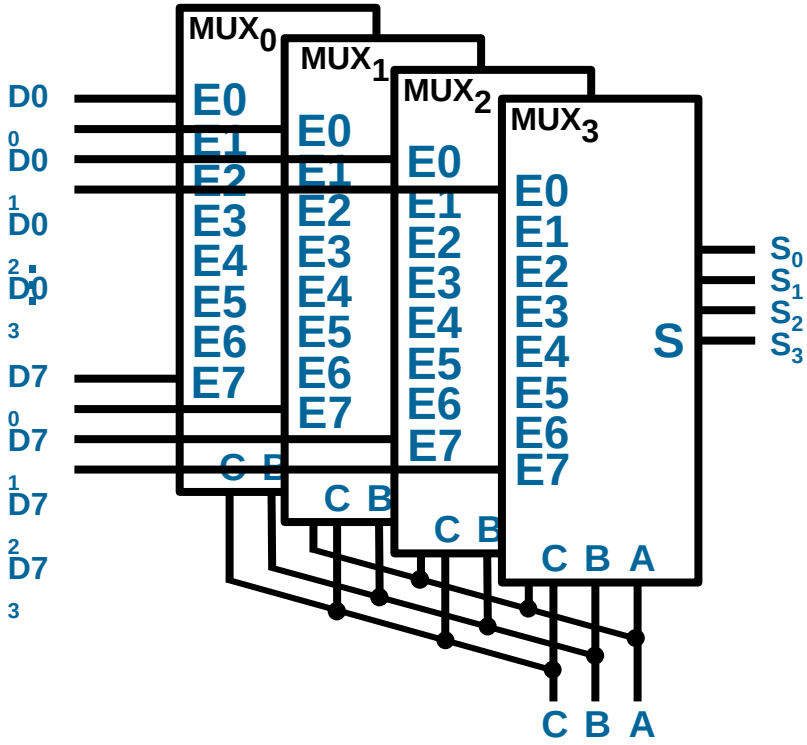
128 + 16 + 2 + 1 =
147 8 data-inputs MUX

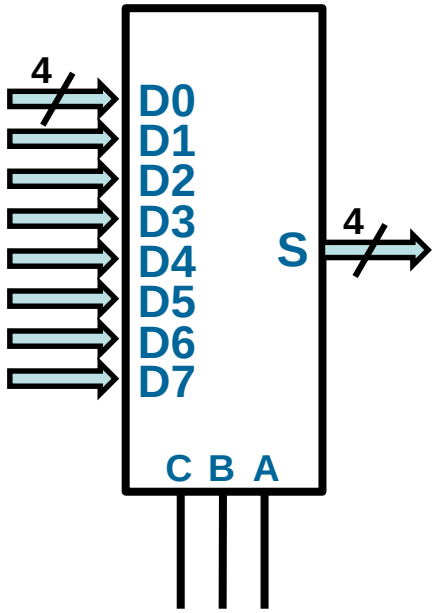


Multiplexors for data of *n*-bits

FCO

- Multiplexors with data-inputs of many bits.
  - Example: 8 data-inputs MUX each input of 4 bits

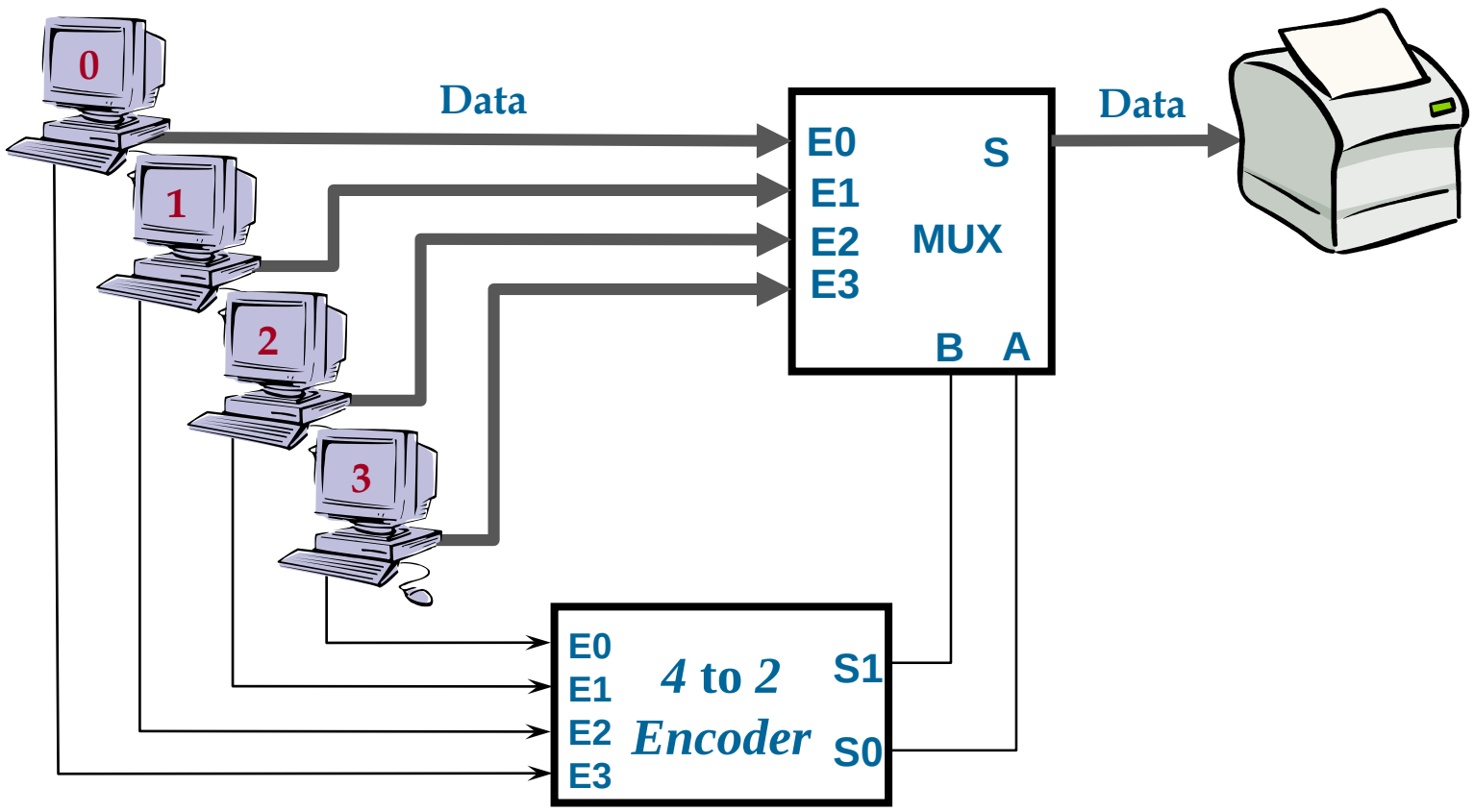




Multiplexors for data of *n*-bits

FCO

– Example of the use of 4 data-inputs MUX

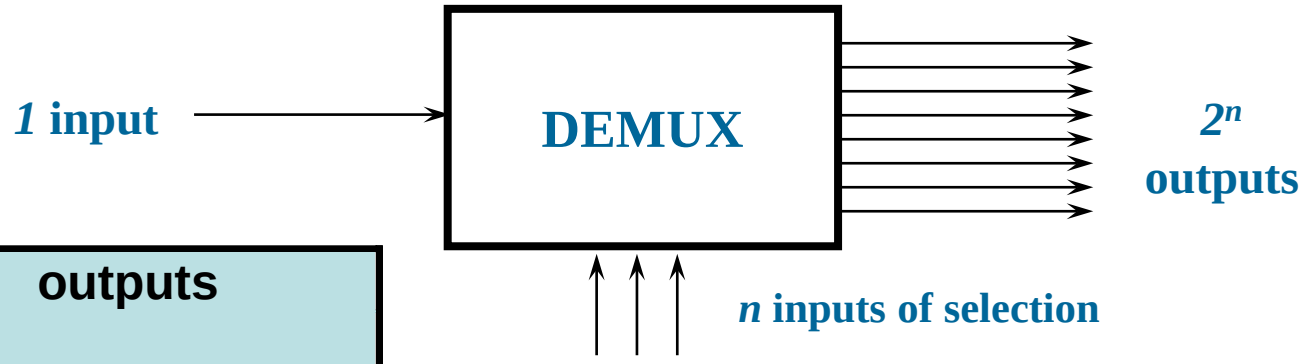


DISCA

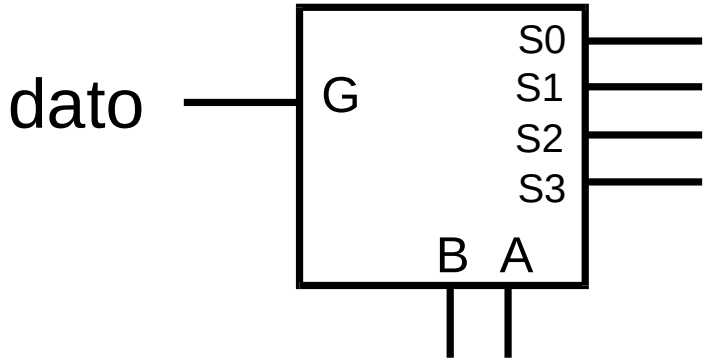
Demultiplexors

FCO

- Demux can be built from decoders
- Demux are used to enable devices



Inputs			outputs			
	selection					
data	B	A	S3	S2	S1	S0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



DISCA

Computers Fundamentals

Subject 3. Basic combinatorial blocks