MSI COMBINATIONAL LOGIC CIRUITS

Combinational Logic Design Procedure

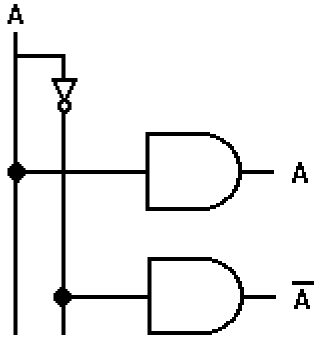
The design of combinational logic circuits starts from the verbal outline of the problem and ends in a logic circuit diagram or a set of Boolean functions from which the logic diagram can be easily obtained.

1. The problem is stated.
2. The number of available input variables and required output variables is determined.
3. The input and output variables are designed letter symbols.
4. The truth table that defines the required relationships between inputs and outputs is derived.
5. The simplified Boolean function for each output is obtained.
6. The logic diagram is drawn.

Example Design a combinational circuit that will multiply two 2-bit numbers.

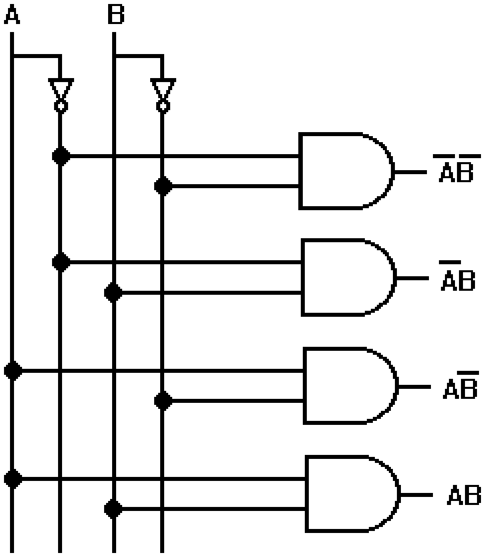
DECODERS/DEMULTIPLEXERS

A *decoder* is a combinational circuit that converts binary information from 8 inputs to a maximum of $\#^8$ unique output lines. If 8 bit decoded information has unused or don't care combination, the decoder output will have less than $\#^8$ outputs.



E	H!	H"
!	"	!
"	!	"

1:2 decoder



E	F	H!	H"	H#	H$
!	!	"	!	!	!
!	"	!	"	!	!
"	!	!	!	"	!
"	"	!	!	!	"

2:4 line decoder

Example.

"Ñ Design a 2-input, 4-output combinational logic circuit to decode the 2-bit output of the following function table:

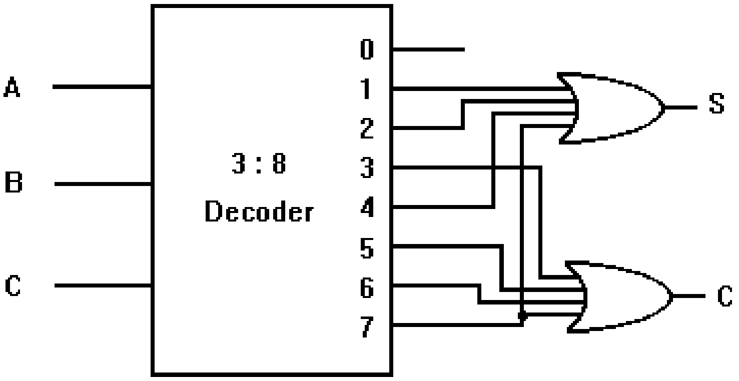
Function	Code
Add	00
Sub	01
Mul	10
Div	11

Inputs		Outputs			
\	]	E	W	Q	H
!	!	"	!	!	!
!	"	!	"	!	!
"	!	!	!	"	!
"	"	!	!	!	"

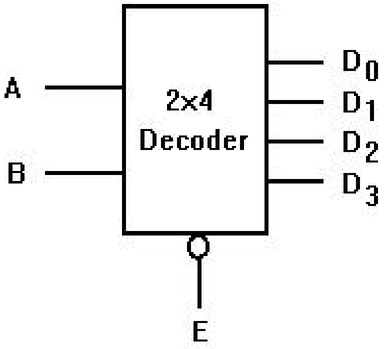
#Ñ Full adder

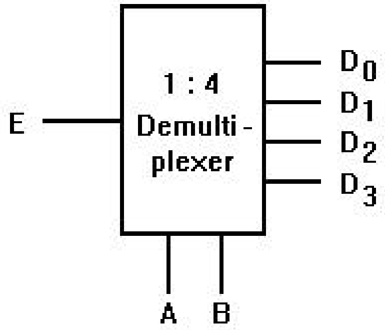
WÐEßFßGÑ œ ! 7Ð"ß#ß%ß(Ñ

GÐEßFßGÑ œ ! 7Ð$ß&ß'ß(Ñ

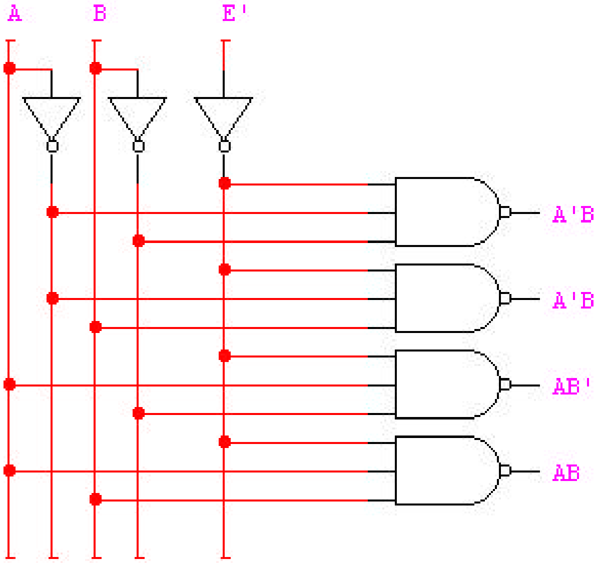


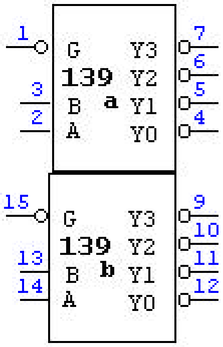
A decoder with an *enable* can function as a *demultiplexer*. A *demultiplexer* is a circuit that receives information on a single line and transmits this information on one of #^{8} possible lines. The selection of a specific output line is controlled by the bit values of 8 selection lines.



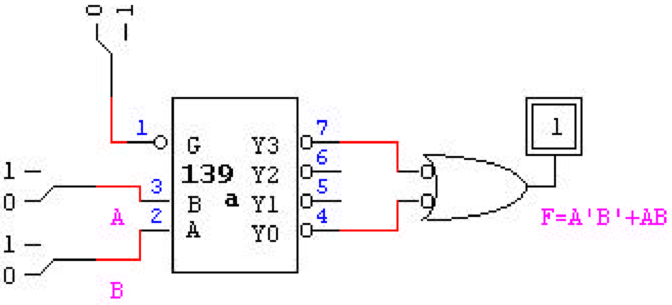


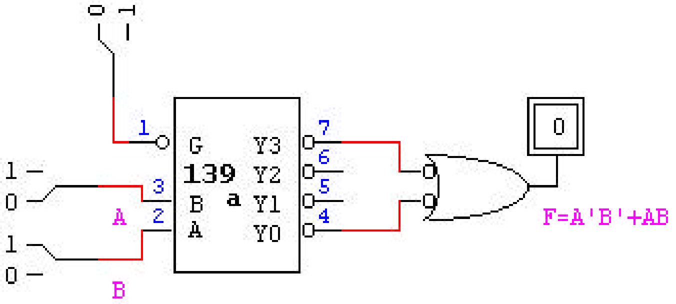
Standard MSI Decoders

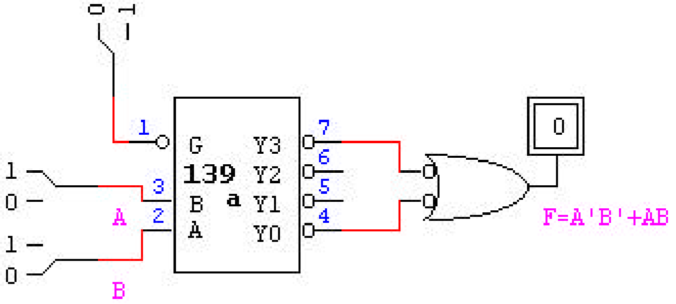


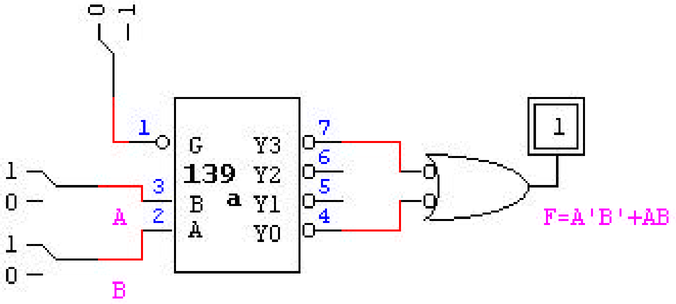


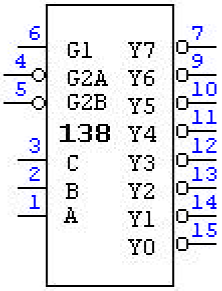
Example: Realize Boolean function 0 œ E"F" € EF with a 2:4 decoder 74139.



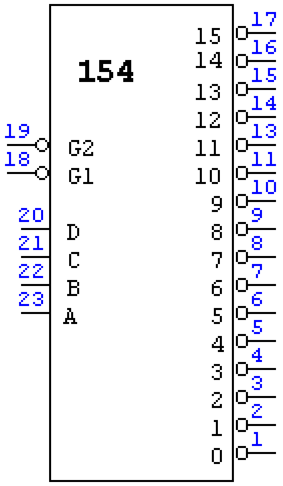






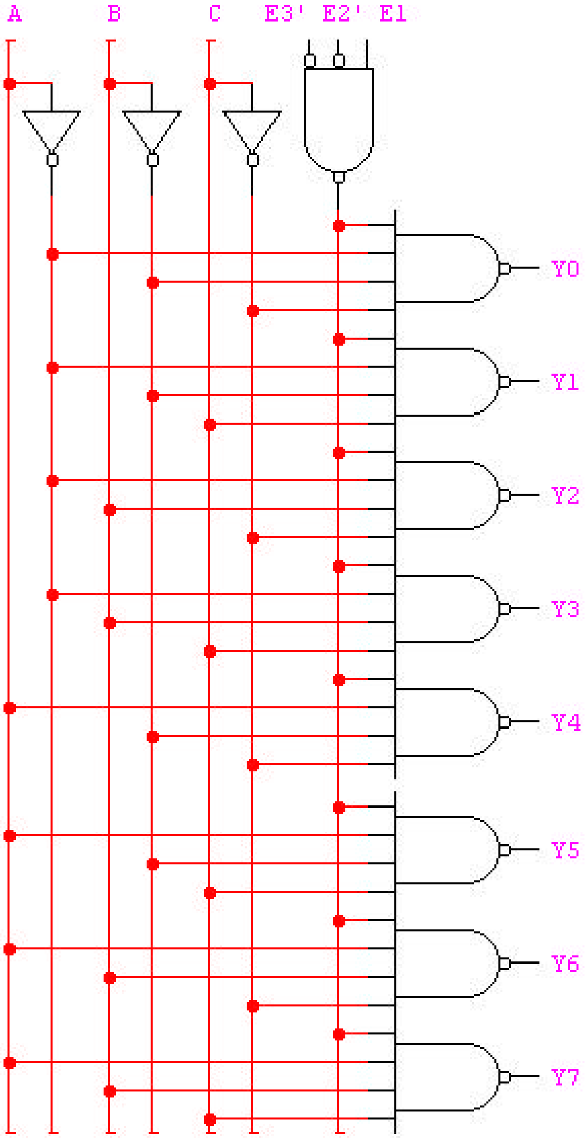


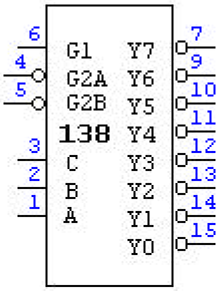
3 to 8 decoder



4 to 16 decoder

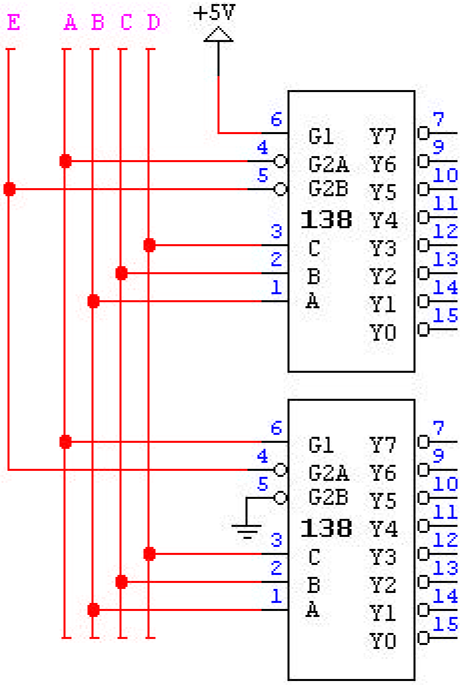
3:8 decocoder





Cascading decoders

Build a 4:16 decoder using two 74138 decoders.



ENCODERS

An *encoder* is a digital circuit that performs the inverse operation of a decoder. An *encoder* has #^{8} (or fewer) input lines and 8 output lines. The output lines generate the binary code corresponding to the input value.

Example.

A calculator has four function keys (add, subtract, multiply, divide). Only one of the function keys can be pressed at a time. When a particular key is pressed, that key is encoded according to the following function tab

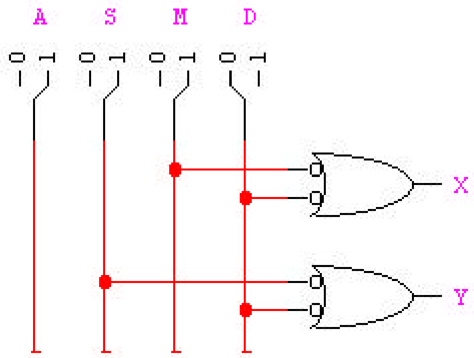
Function	Code
Add	00
Sub	01
Mul	10
Div	11

	Inputs				Outputs	
Line	E	W	Q	H	\	]
1	"	!	!	!	!	!
2	!	"	!	!	!	"
3	!	!	"	!	"	!
4	!	!	!	"	"	"

\ œ E"W"

] œ E"Q"

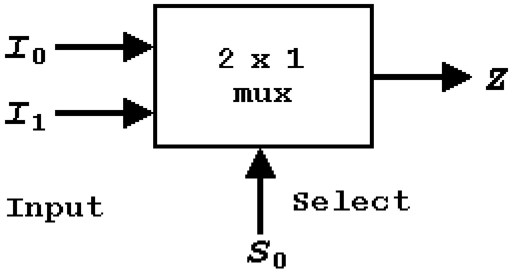
Line	Inputs				Outputs	
	E	W	Q	H	\	]
1	!	"	"	"	!	!
2	"	!	"	"	!	"
3	"	"	!	"	"	!
4	"	"	"	!	"	"



MULTIPLEXERS/SELECTORS

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A *digital multiplexer* is a combination circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^8 input lines and 8 selection lines whose bit combinations determine which input is selected.

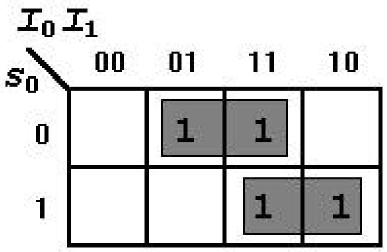
2 , 1 multiplexers

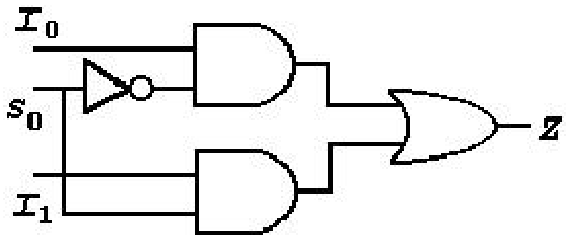


=!	D
!	M!
"	M"

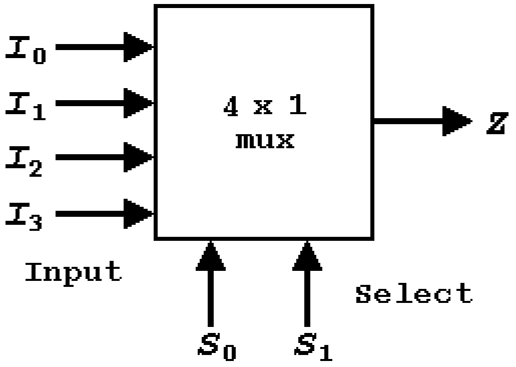
^ œ ≐!M! € =!M"

M"	M!	=!	J"
!	!	!	!
!	!	"	!
!	"	!	"
!	"	"	!
"	!	!	!
"	!	"	"
"	"	!	"
"	"	"	"





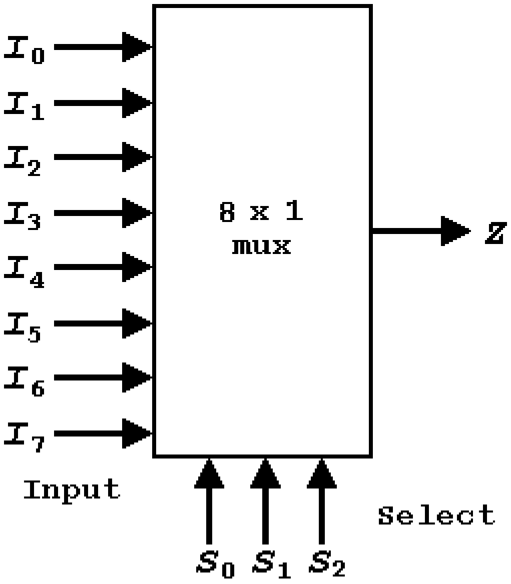
4 , 1 multiplexers



="	=!	^
!	!	M!
!	"	M"
"	!	M#
"	"	M$

^ œ ≐"≐!M! € ≐"=!M" € ="≐!M# € ="=!M$

8 , 1 multiplexers

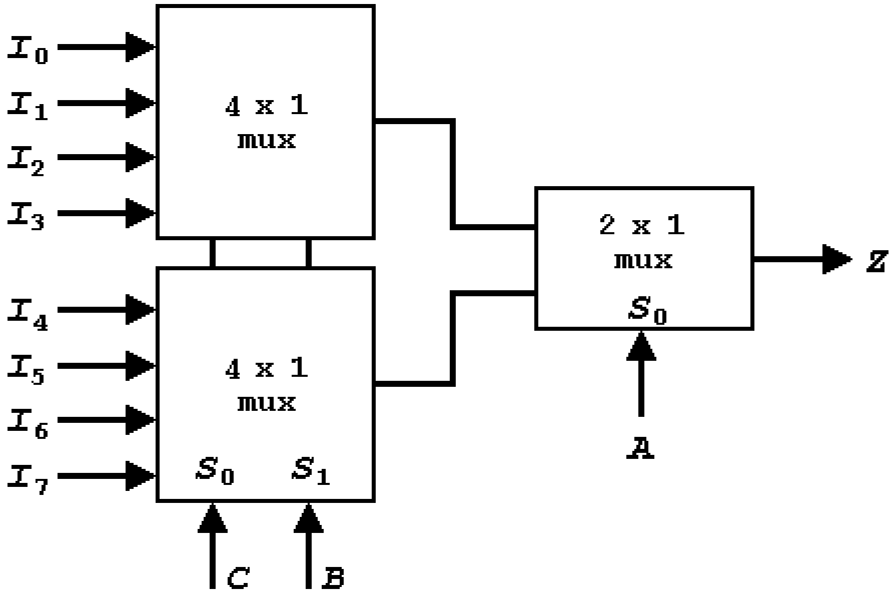


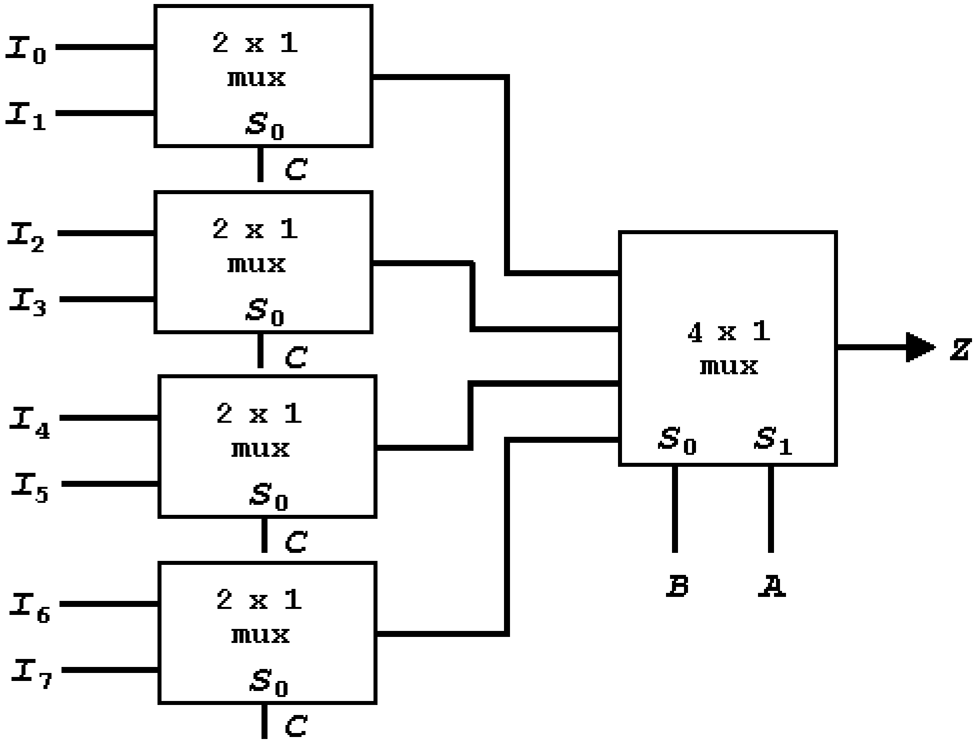
S#	="	=!	^
!	!	!	M!
!	!	"	M"
!	"	!	M#
!	"	"	M$
"	!	!	M%
"	!	"	M&
"	"	!	M'
"	"	"	M(

^ œ ≐#≐"≐!M! € ≐#≐"=!M" € ≐#="≐!M# € ≐#="=!M$
€ =#≐"≐!M% € =#≐"=!M& € =#="≐!M' € =#="=!M(

Alternative Implementation

The following diagrams show how to construct a 8 , 1 multiplexer using either 4 , 1 multiplexers or 2 , 1 multiplexers.



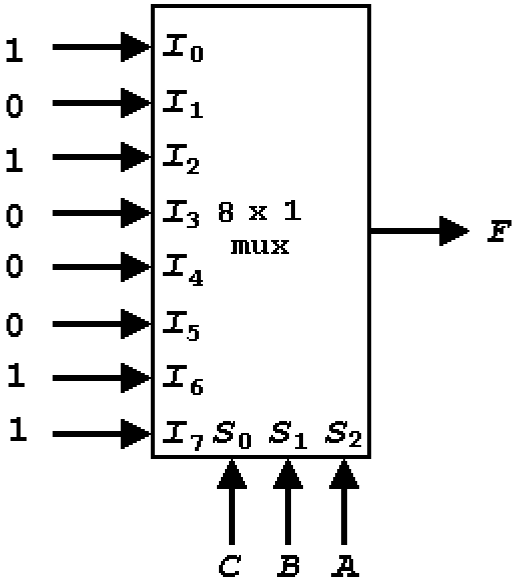


A multiplexer can implemented a truth table but it can do better than that.

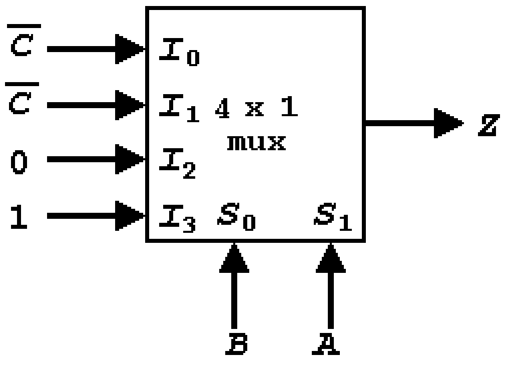
Example

1) J ÐEß F ßGÑ œ ! 7 Ð!ß#ß 'ß (Ñ

E	F	G	J
!	!	!	"
!	!	"	!
!	"	!	"
!	"	"	!
"	!	!	!
"	!	"	!
"	"	!	"
"	"	"	"



E	F	G	J	
!	!	!	"	Ġ
!	!	"	!	
!	"	!	"	Ġ
!	"	"	!	
"	!	!	!	!
"	!	"	!	
"	"	!	"	"
"	"	"	"	



C \ AB	00	01	11	10
0	1	1	1	0
1	0	0	1	0

#Ñ Implement a 2 , 1 multiplexer to J ÐEß F ßGÑ œ ! 7Ð"ß$ß&ß 'Ñ

$Ñ Implement a 4 , 1 multiplexer to J ÐEß F ßGß HÑ œ ! 7Ð!ß "ß$ß%ß)ß *ß "&Ñ

Multiplexer devices

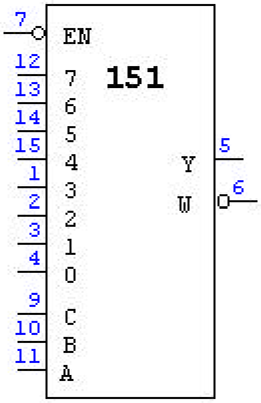
There are multiplexers devices in 7400 series,

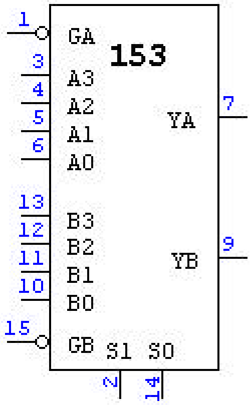
74150 16-to-1 multiplexer (24 pins)

74151 8-to-1 multiplexer (16 pins)

74153 4-to-1 multiplexers

74157 2-to-1 multiplexers





Assignment p.319 4.3, 4.4, 4.5, 4.9, 4.15, 4.16, 4.18, 4.19, 4.20, 4.21, 4.23, 4.25