PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

ANNOUNCEMENT!

- Due to COVID-19 concerns, the syllabus for the course will need to be modified somewhat
- After spring break, the course will continue, but it will be *ONLINE*…
- Please refer to the course web page http://www.physics.purdue.edu/~mjones/phys53600_Spring2020/

  for upcoming announcements.

ANNOUNCEMENT!!!

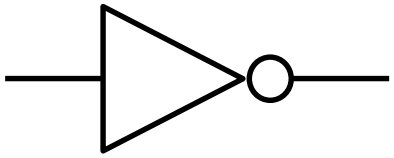
- What the rest of the course might look like:
  - Lecture notes will continue to be posted online
  - Communication with the class will be via e-mail
  - Perhaps we can do without the lectures?
  - Maybe one or two assignment questions associated with each lecture?
  - Skip the second mid-term (grading scheme will be modified accordingly)
  - Lab grade will be based on work performed so far
  - Still don't know what to do about the final… stay tuned.

Digital Electronics

- Fundamental unit of information is the Boolean digit (bit) which can either be 0 or 1
- Combinatorial logic circuits perform Boolean algebraic operations
  - Inverter: $Q = \overline{D}$
  - NOR gate: $Q = \overline{A + B}$
- Other functions can be constructed from these (or similar) elementary operations

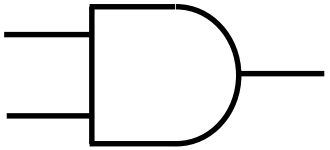
Basic Logic Gates

- Inverter:



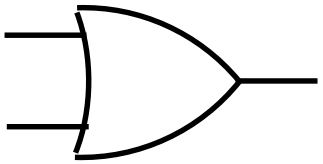
$$Q = \overline{D}$$

- AND:



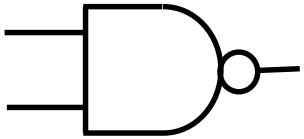
$$Q = A \cdot B$$

- OR:



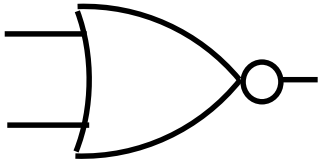
$$Q = A + B$$

- NAND:



$$Q = \overline{A \cdot B}$$

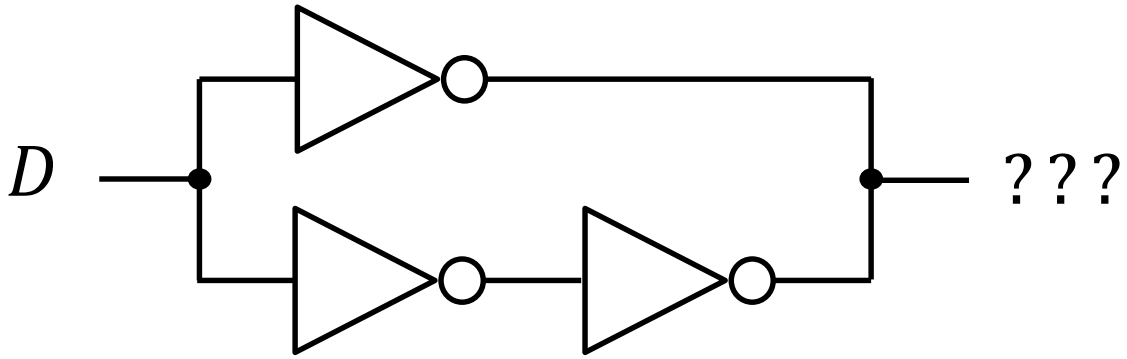
- NOR:



$$Q = \overline{A + B}$$

Other Logic Types

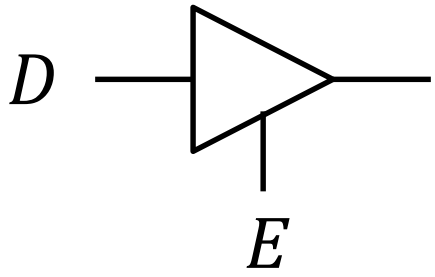
- Although Boolean digits can be represented by voltage levels, these are still fundamentally analog circuits
- Some configurations will simply not work:



- This will always have both logic levels connected together at the outputs

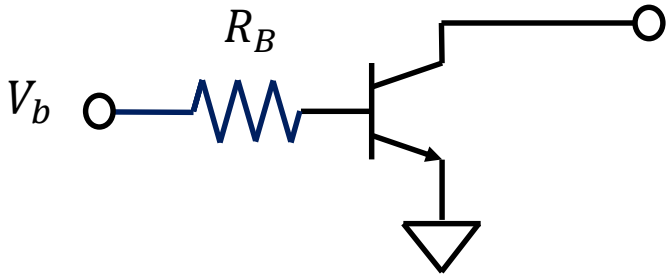
Other Logic Types

- We can exploit the analog nature of logic circuits to resolve these possible ambiguities
- Tri-state logic: 0, 1 or Z (high impedance)



$$Q = \begin{cases} D \text{ if } E = 1 \\ Z \text{ if } E = 0 \end{cases}$$

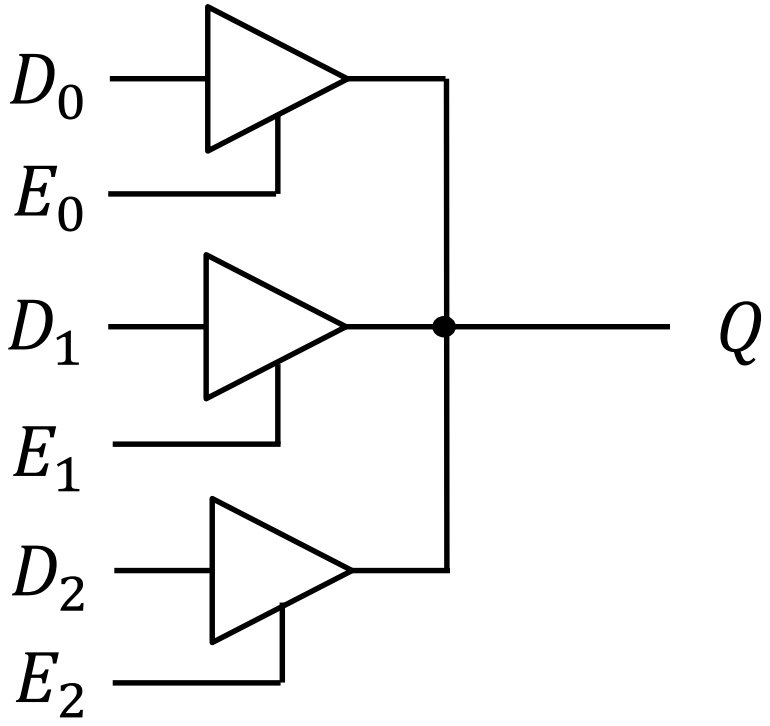
- Open-collector logic: 0 or Z (high impedance)



$$Q = \begin{cases} 0 \text{ if } V_b > V_{IH} \\ Z \text{ if } V_b < V_{IL} \end{cases}$$

Applications of other Logic Types

- Multiple drivers of a common net:

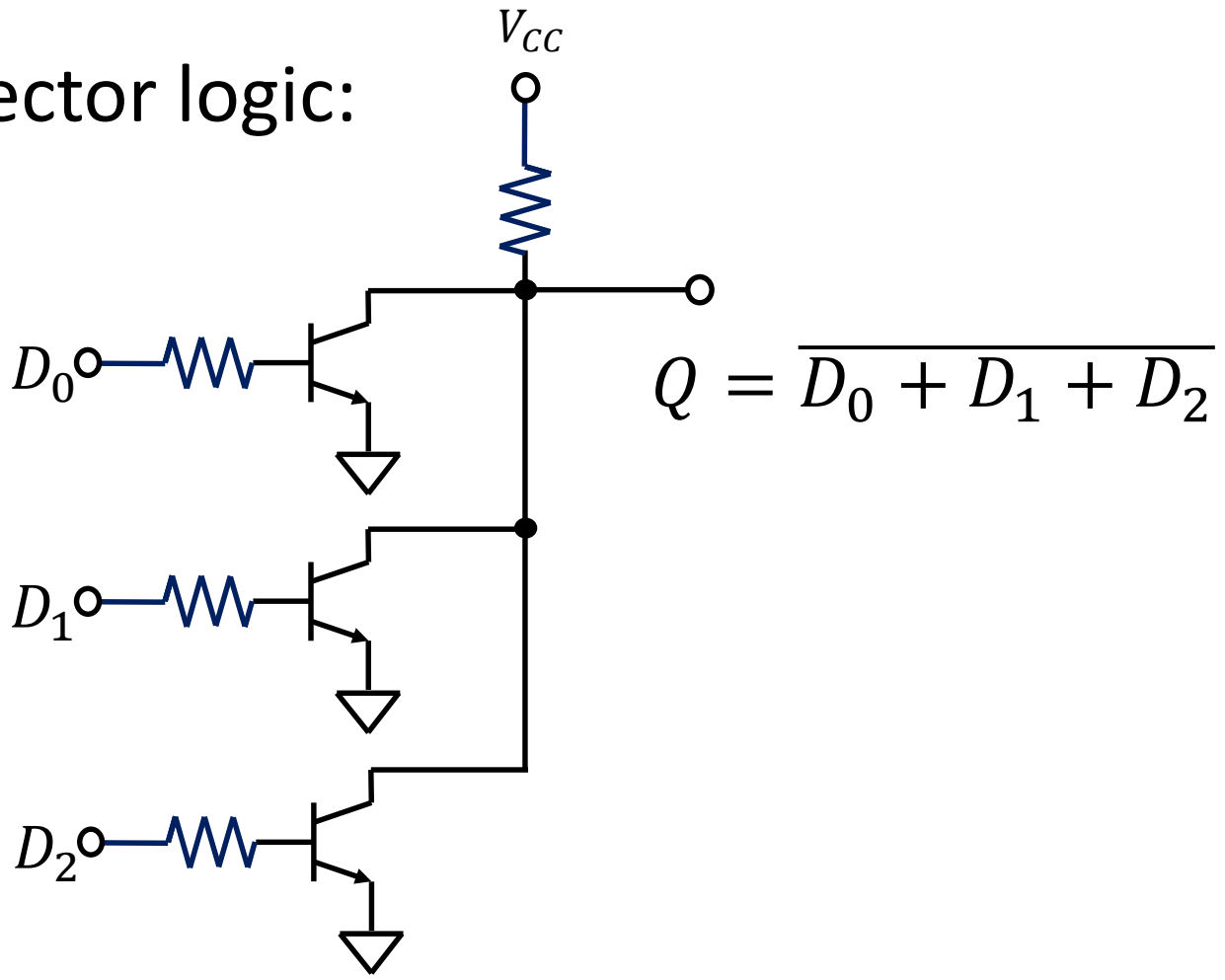


$Q = D_i$ when $E_i = 1$ and all other enables are 0

(Make sure that no two enables are 1 simultaneously)

Application of other Logic Types

- Open collector logic:



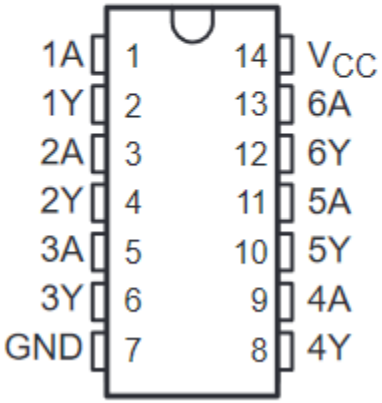
Discrete Logic Gates

- You can buy integrated circuits that contain these discrete logic gates
- They can be very useful when used individually
- They are not often used to construct complicated logic circuits

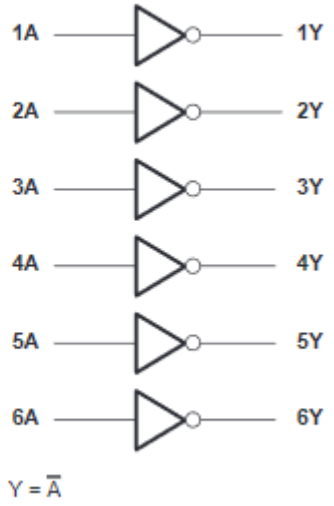
Discrete Logic Gates

- SN74LS04: Hex inverter

SN5404 . . . J PACKAGE
SN54LS04, SN54S04 . . . J OR W PACKAGE
SN7404, SN74S04 . . . D, N, OR NS PACKAGE
SN74LS04 . . . D, DB, N, OR NS PACKAGE
(TOP VIEW)



logic diagram (positive logic)



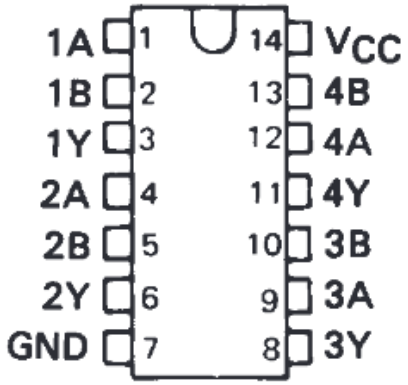
FUNCTION TABLE
(each inverter)

INPUT A	OUTPUT Y
H	L
L	H

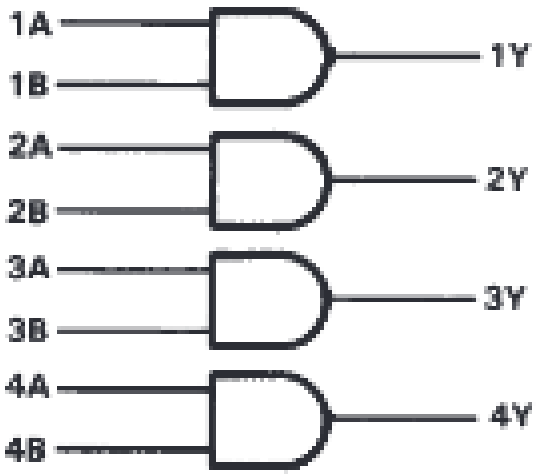
Discrete Logic Gates

- 74LS08: Quad AND gate

SN5408, SN54LS08, SN54S08 . . . J OR W PACKAGE
SN7408 . . . J OR N PACKAGE
SN74LS08, SN74S08 . . . D, J OR N PACKAGE
(TOP VIEW)



logic diagram (positive logic)



$Y = A \cdot B$ or $Y = \overline{\overline{A} + \overline{B}}$

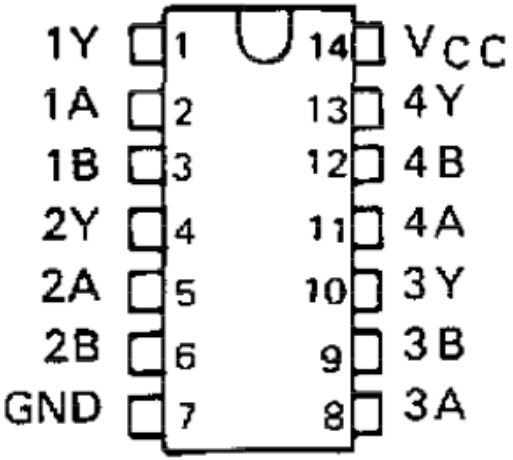
FUNCTION TABLE (each gate)

INPUTS		OUTPUT
A	B	Y
H	H	H
L	X	L
X	L	L

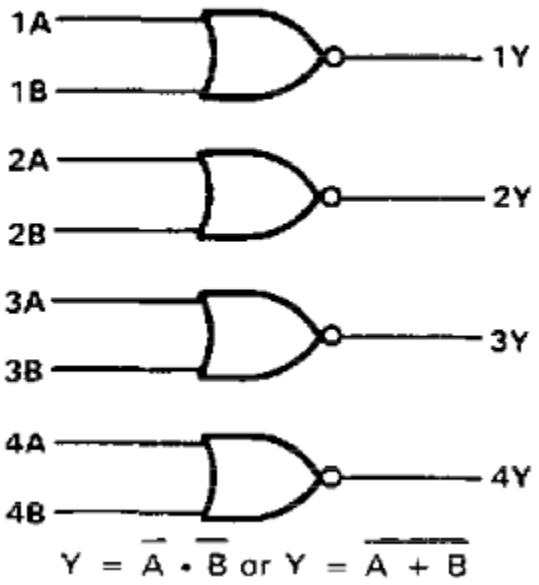
Discrete Logic Gates

- 74LS02: Quad NOR gate

SN5402 . . . J PACKAGE
SN54LS02, SN54S02 . . . J OR W PACKAGE
SN7402 . . . N PACKAGE
SN74LS02, SN74S02 . . . D OR N PACKAGE
(TOP VIEW)



logic diagram (positive logic)



$Y = \overline{A} \cdot \overline{B}$ or $Y = \overline{A + B}$

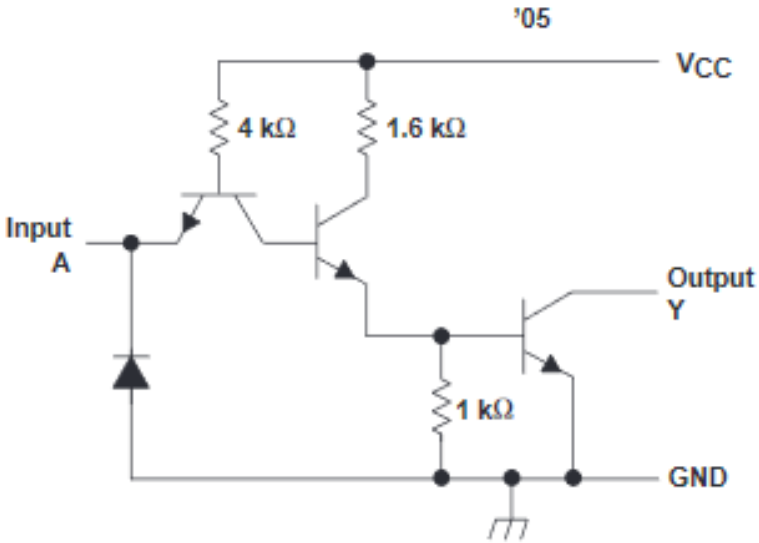
FUNCTION TABLE (each gate)

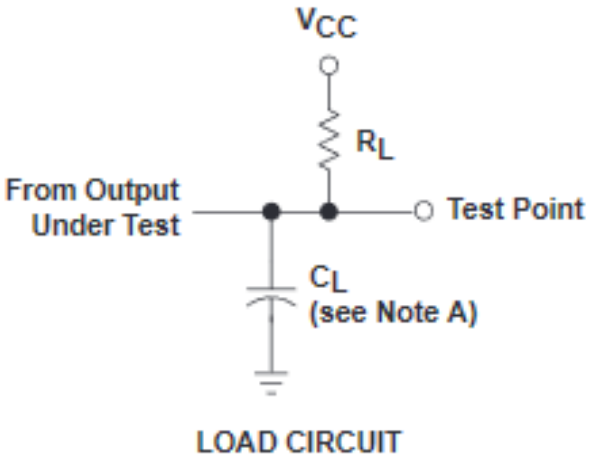
INPUTS		OUTPUT
A	B	Y
H	X	L
X	H	L
L	L	H

Discrete Logic Gates

- 74LS05: Hex inverter with open-collector outputs

schematic (each inverter)





R_L determines the switching speed.

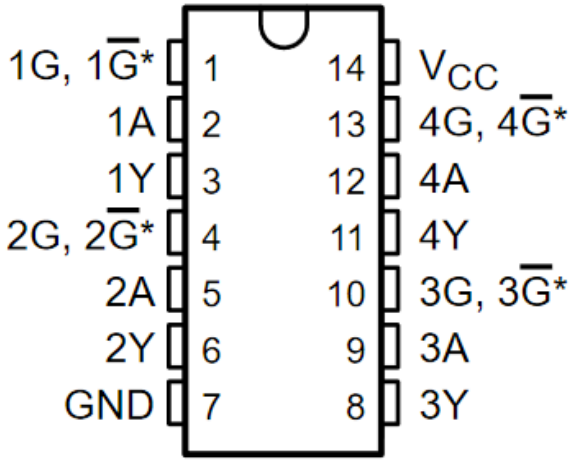
switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	R_L = 280 Ω	C_L = 15 pF	2	5	7.5	ns
t_{PHL}					2	4.5	7	
t_{PLH}				C_L = 50 pF		7.5		ns
t_{PHL}						7		

Discrete Logic Gates

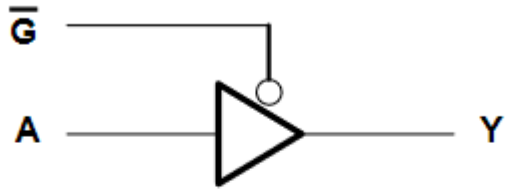
- 74LS125, 74LS126: Quad buffer with 3-state outputs

SN54125, SN54126, SN54LS125A,
SN54LS126A . . . J OR W PACKAGE
SN74125, SN74126 . . . N PACKAGE
SN74LS125A, SN74LS126A . . . D, N, OR NS PACKAGE
(TOP VIEW)

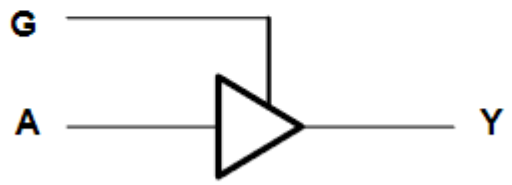


*$\overline{G}$ on ’125 and ’LS125A devices;
G on 126 and ’LS126A devices

’125, ’LS125A



’126, ’LS126A

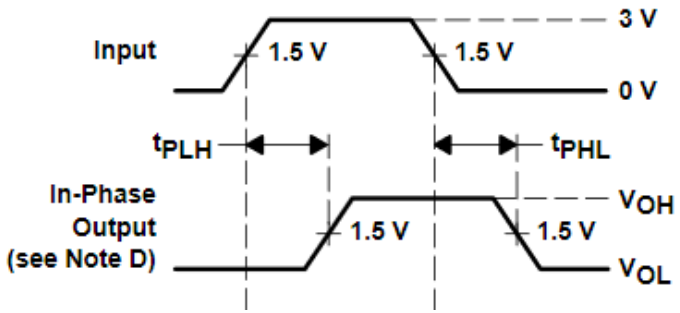


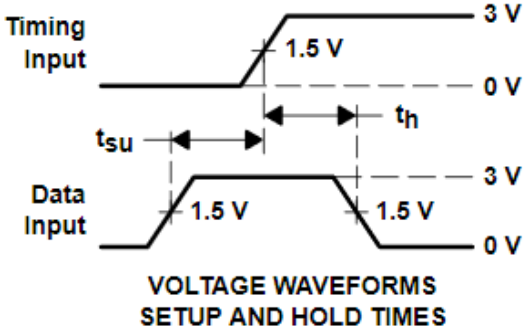
Y = A

Output is disabled when G=1 (LS125)
Output is disabled when G=0 (LS126)

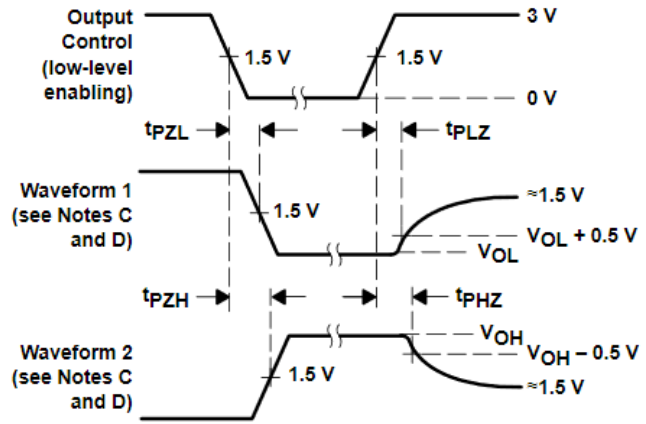
Switching Characteristics

- The speed of logic gates can be defined using several timing measures:





VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

Important timing parameters:

- Propagation delay
- Setup time
- Hold time

Timing Characteristics

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Figure 1)

PARAMETER	TEST CONDITIONS		SN54125 SN74125			SN54126 SN74126			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	R_L = 400 Ω,	C_L = 50 pF		8	13		8	13	ns
t_{PHL}				12	18		12	18	
t_{PZH}	R_L = 400 Ω,	C_L = 50 pF		11	17		11	18	ns
t_{PZL}				16	25		16	25	
t_{PHZ}	R_L = 400 Ω,	C_L = 5 pF		5	8		10	16	ns
t_{PLZ}				7	12		12	18	

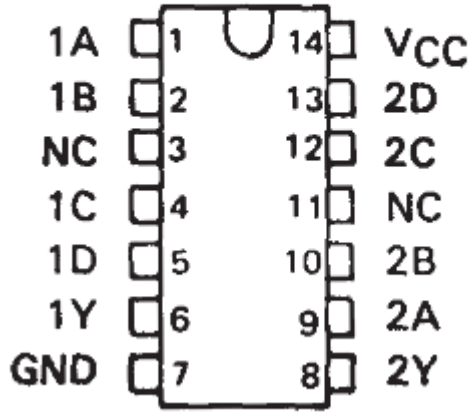
t_{PLH} is the propagation delay for a low-to-high transition
t_{PHL} is the propagation delay for a high-to-low transition
t_{PZH} is the propagation delay for a high-impedance to high transition
t_{PZL} is the propagation delay for a high-impedance to low transition
t_{PHZ} is the propagation delay for a high to high-impedance transition
t_{PLZ} is the propagation delay for a low to high-impedance transition

t_{SU} is the “setup time”: time input must remain stable before an event
t_H is the “hold time”: time input must remain stable after an event

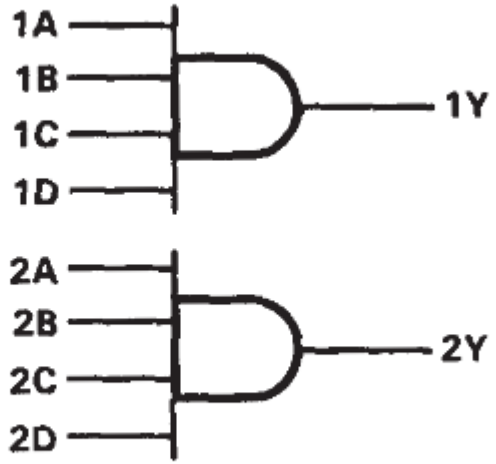
Other Logic Functions

- Logic with more than 2 inputs:

SN54LS21 . . . J OR W PACKAGE
SN74LS21 . . . D OR N PACKAGE
(TOP VIEW)



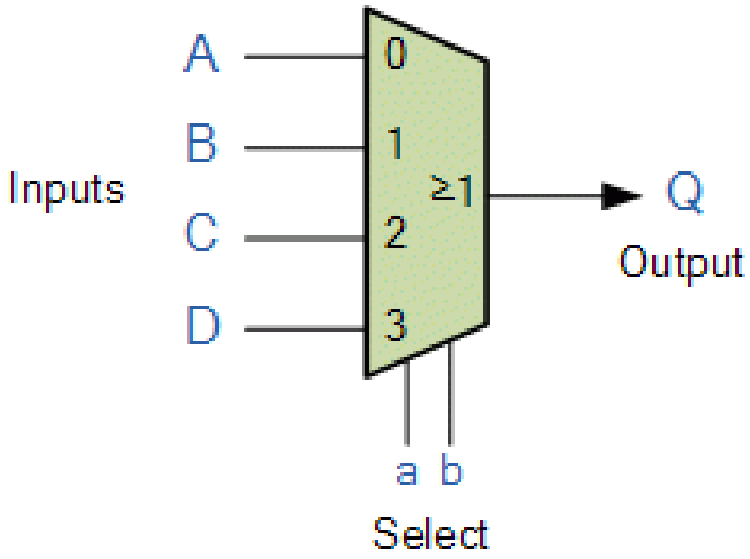
logic diagram



(positive logic) $Y = A \cdot B \cdot C \cdot D$ or $Y = \overline{\overline{A} + \overline{B} + \overline{C} + \overline{D}}$

Selectors/Multiplexers

- Beyond the basic Boolean algebraic operations, several logic functions can be described by their applications
- A selector will output one (out of many) inputs based on a set of control inputs

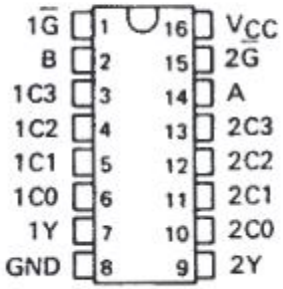


a	b	Q
0	0	A
0	1	B
1	0	C
1	1	D

Selectors/Multiplexers

- 74LS253: Dual 4-to-1 line data selectors (3-state outputs)

SN54LS253, SN54S253 . . . J OR W PACKAGE
SN74LS253, SN74S253 . . . D OR N PACKAGE
(TOP VIEW)



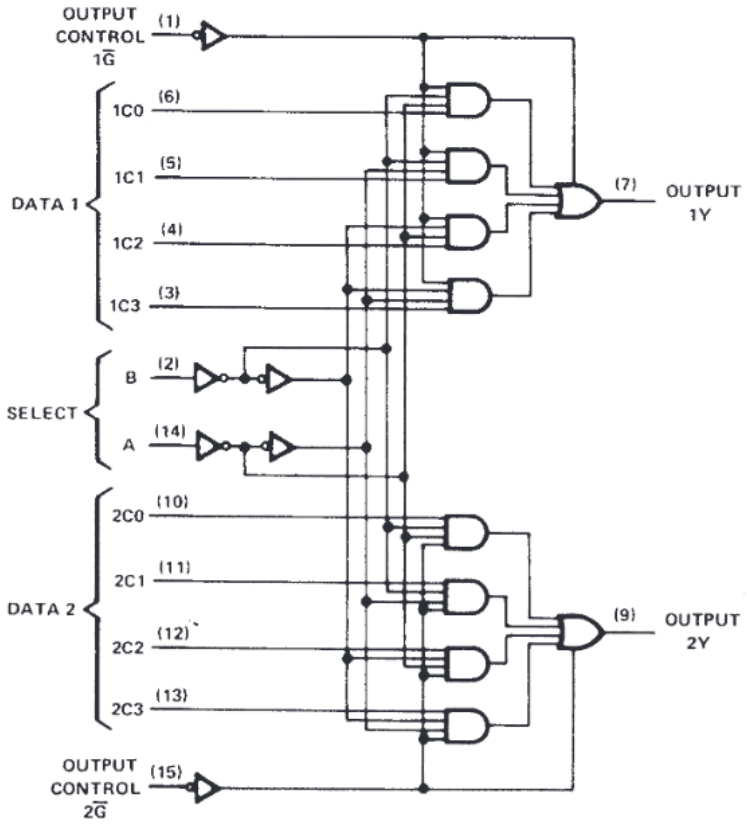
FUNCTION TABLE

SELECT INPUTS		DATA INPUTS				OUTPUT CONTROL	OUTPUT
B	A	C0	C1	C2	C3	$\bar{G}$	Y
X	X	X	X	X	X	H	Z
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

Address inputs A and B are common to both sections.

H = high level, L = low level, X = irrelevant, Z = high impedance (off)

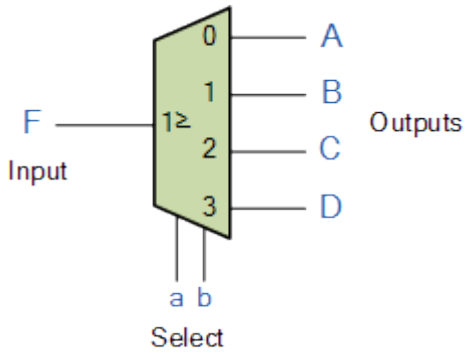
logic diagram (positive logic)



bers shown are for D, J, N, and W packages.

Decoders/Demultiplexers

- Decoders route one input to exactly one (of several) possible outputs
- Inputs are the binary address of the selected output

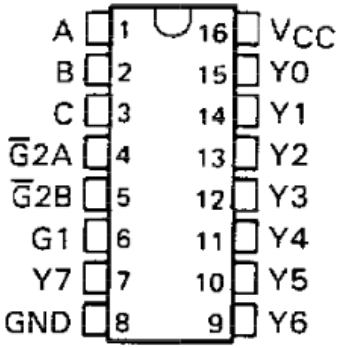


a	b	A	B	C	D
0	0	F	0	0	0
0	1	0	F	0	0
1	0	0	0	F	0
1	1	0	0	0	F

Decoders/Demultiplexers

- 74LS138: 3 to 8-line decoder/demultiplexer

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138A . . . D OR N PACKAGE
(TOP VIEW)



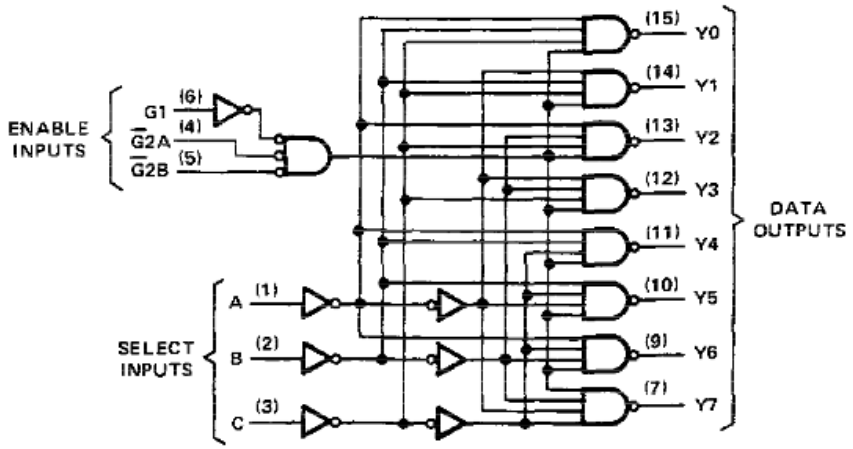
'LS138, SN54138, SN74S138A
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*$\overline{G2} = \overline{G2A} + \overline{G2B}$
H = high level, L = low level, X = irrelevant

logic diagram and function table

'LS138, SN54S138, SN74S138A

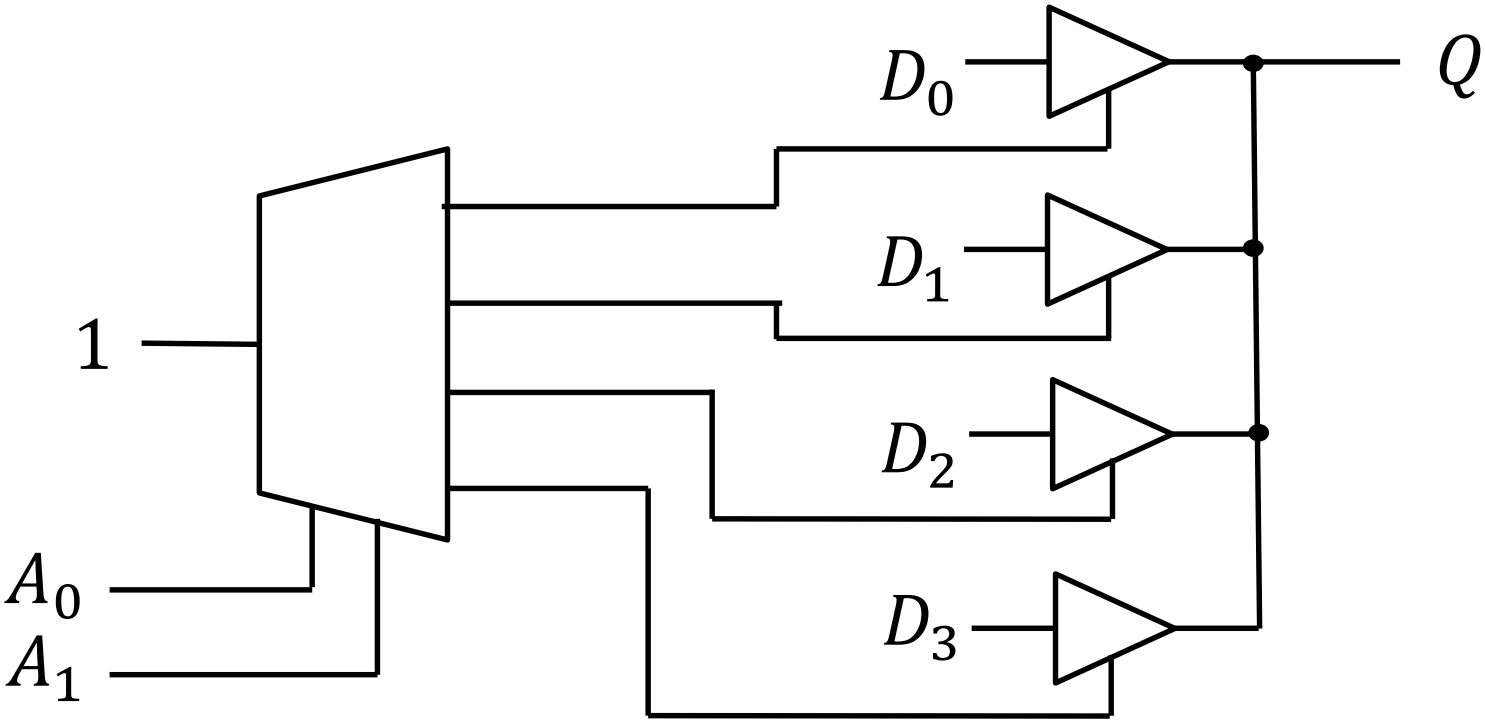


Pin numbers shown are for D, J, N, and W packages.

This device sets one input LOW based on the select and enable inputs. Otherwise, the inputs are HIGH.

Example Applications

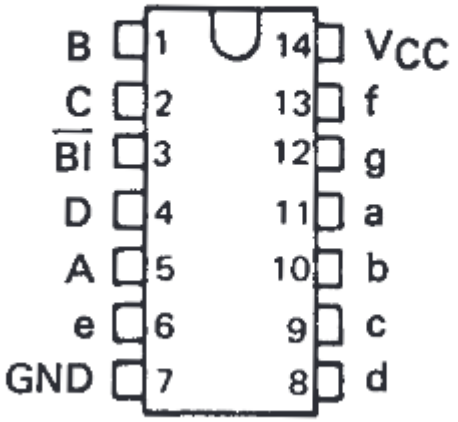
- Outputs of tri-state logic can be connected together, but only one output should be driven at a time

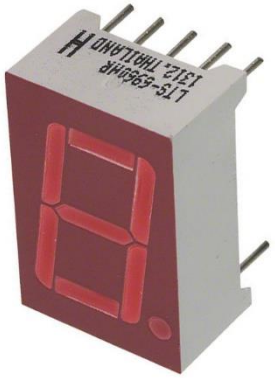


Special Applications

- 74LS47: Binary to 7-segment LED driver

SN54LS49 . . . J OR W PACKAGE
SN74LS49 . . . D OR N PACKAGE
(TOP VIEW)





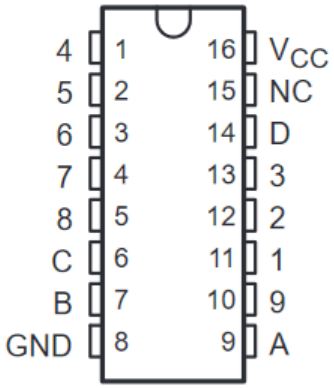
SEGMENT IDENTIFICATION

NUMERICAL DESIGNATIONS AND RESULTANT DISPLAYS

Priority Encoder

- 74LS147: 10 to 4-line priority encoder

SN54147, SN54LS147 . . . J OR W PACKAGE
SN74147, SN74LS147 . . . D OR N PACKAGE
(TOP VIEW)



FUNCTION TABLE – '147, 'LS147

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

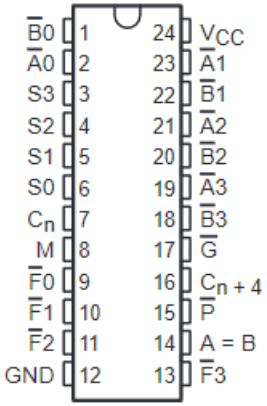
H = high logic level, L = low logic level, X = irrelevant

Binary output encodes the index of the highest-numbered input channel that is in a low state

Arithmetic/Logic Units

- 74AS181: Performs various logic operations

SN54AS181B . . . JT OR JW PACKAGE
SN74AS181A . . . N OR NT PACKAGE
(TOP VIEW)



SELECTION				ACTIVE-HIGH DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	C_n = H (no carry)	C_n = L (with carry)
L	L	L	L	$F = \overline{A}$	F = A	F = A PLUS 1
L	L	L	H	$F = \overline{A + B}$	F = A + B	F = (A+ B) PLUS 1
L	L	H	L	$F = \overline{A}B$	$F = A + \overline{B}$	$F = (A + \overline{B})$ PLUS 1
L	L	H	H	F = 0	F = MINUS 1 (2's COMPL)	F = ZERO
L	H	L	L	$F = \overline{AB}$	$F = A$ PLUS $A\overline{B}$	$F = A$ PLUS $A\overline{B}$ PLUS 1
L	H	L	H	$F = \overline{B}$	$F = (A + B)$ PLUS $A\overline{B}$	$F = (A + B)$ PLUS $A\overline{B}$ PLUS 1
L	H	H	L	$F = A \oplus B$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	$F = A\overline{B}$	$F = A\overline{B}$ MINUS 1	$F = A\overline{B}$
H	L	L	L	$F = \overline{A} + B$	F = A PLUS AB	F = A PLUS AB PLUS 1
H	L	L	H	$F = \overline{A \oplus B}$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	$F = (A + \overline{B})$ PLUS AB	$F = (A + \overline{B})$ PLUS AB PLUS 1
H	L	H	H	F = AB	F = AB MINUS 1	F = AB
H	H	L	L	F = 1	F = A PLUS A†	F = A PLUS A PLUS 1
H	H	L	H	$F = A + \overline{B}$	F = (A + B) PLUS A	F = (A + B) PLUS A PLUS 1
H	H	H	L	F = A + B	$F = (A + \overline{B})$ PLUS A	$F = (A + \overline{B})$ PLUS A PLUS 1
H	H	H	H	F = A	F = A MINUS 1	F = A

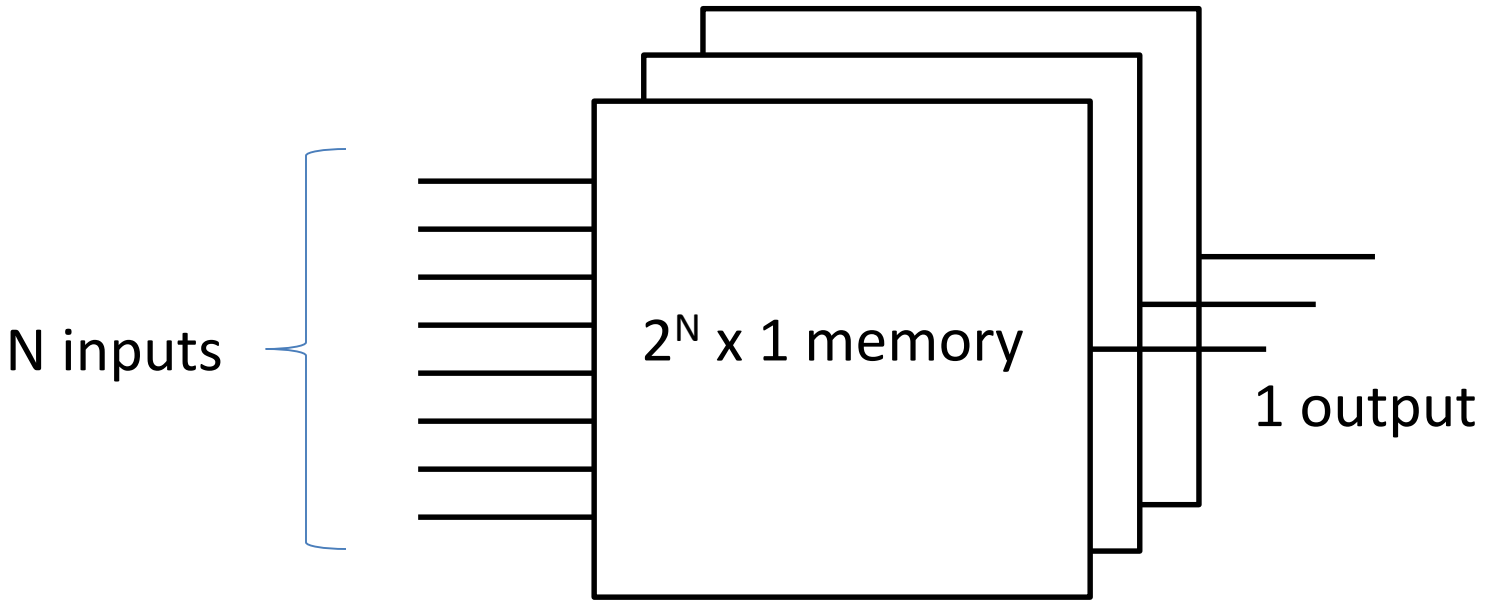
† Each bit is shifted to the next more significant position.

(re-)Programmable Logic Arrays

- Devices with many inputs and/or outputs that can be re-configured to perform arbitrary logic functions.
- Configuration can be programmed once, or re-programmed when required
- Requires synthesis design tools to generate configuration data
- Examples: Xilinx, Altera (now Intel)

(re-)Programmable Logic

- A simple way to think of these is like a read-only memory device:



- There needs to be a way to configure the memory before the device is put into use.

Combinatorial vs Sequential Logic

- So far, all the devices we have considered perform Boolean algebraic operations.
- The output is simply a function of the input
- They do not store information
- Their output does not depend on the past history of the inputs
- Sequential logic is based on the ability to store information and change it as required
- The combination of combinatorial and sequential logic is the basis for most digital architectures of arbitrary complexity

Wash your hands!