Digital Basics

•Analogue Communications over long distances often were hampered by noise, fluctuations, etc.

•Analogue communications can have high band width. AM and FM radio! (information transfer/sec)

•Digital communications are more robust but can suffer from band width limitations. Drums, Smoke Signals, Morse Code, Internet, etc.

•Analogue-to-digital & Digital-to-Analogue translators important for research and communications.

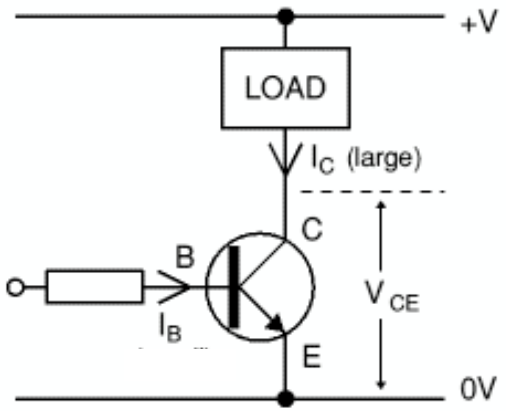
Transistor Switch and Comparator

In the transistor switch we can create a situation where a small input current can turn a transistor “On” or “Off”, the two states needed for digital electronics.

Early computers used the vacuum tubes, but in the 1950’s Silicon rectifiers and diodes were being developed for this purpose.

The “On” “Off” state is refered to as a “bit” short for “binary digit”.

Switch



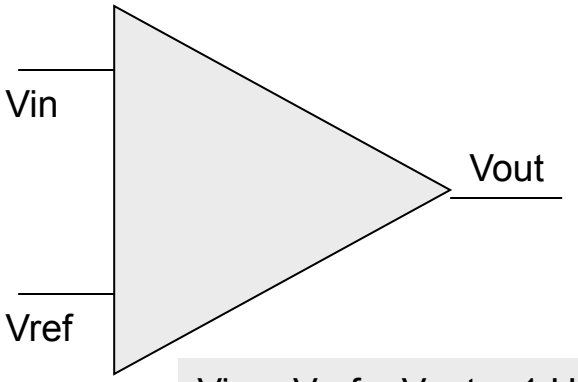
On

I_B high
saturating transistor

Off

I_B low

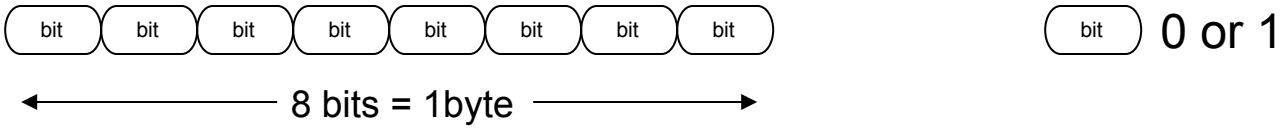
Digital Comparator



Vin > Vref Vout = 1 HIGH
Vin < Vref Vout = 0 LOW

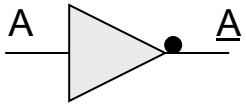
BITS and BYTES

- Digital information is sent over data packets.

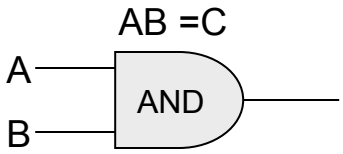


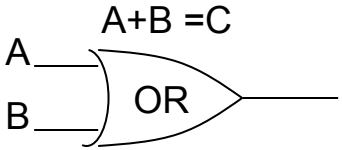
- Internet packet = 32 bits
- Computer word packet = 32 bits long word
  Computer word packet = 64 bits double precision
- Computer memory = 1 Gbyte = 10^9 x 8 bits
- Hard disk = >50 GB 2 TB (10^{12})
- CD = 700 MB (80 minutes of audio)
- DVD = 4.7 GB (2 hrs HR video)
- Blue Ray = 100-125 GB

Digital Gates



INERTER





TRUTH TABLE

A	B	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

TRUTH TABLE

A	B	OR	NOR	XOR
0	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

Boolean Algebra

$A \cdot B = C$ A .and. B = C	$\underline{A} \cdot \underline{B} = \underline{C}$ $\underline{A}$ .and. $\underline{B}$ = $\underline{C}$
A+B= C A .or. B = C	$\underline{A+B} = \underline{C}$ $\underline{A}$ .or. $\underline{B}$ = $\underline{C}$

Boolean Theorems

(1) $A+A = A$

(2) $A \cdot A = A$

(3) $A + 1 = 1$

(4) $A \cdot 1 = A$

(5) $A + 0 = A$

(6) $A \cdot 0 = A$

(7) $A + \underline{A} = A$

(8) $A \cdot \underline{A} = 0$

(9) $\underline{\underline{A}} = A$

(10) $A + A \cdot B = A$

(11) $A \cdot (A + B) = A$

(12) $A \cdot (\underline{A} + B) = A \cdot B$

(13) $A + \underline{A} \cdot B = A + B$

(14) $\underline{A} + A \cdot B = \underline{A} + B$

(15) $\underline{A} + A \cdot \underline{B} = \underline{A} + \underline{B}$

(16) $\underline{A \cdot B} = \underline{A} + \underline{B}$

(17) $\underline{A + B} = \underline{A} \cdot \underline{B}$

DeMorgan's Theorems

Boolean Proofs by Truth Table

(1) A+A = A

A	A	A+A
0	0	0
1	1	1

(2) A · A = A

A	A	AA
0	0	0
1	1	1

(8) A · $\overline{A}$ = 0

A	$\overline{A}$	A$\overline{A}$
0	1	0
1	0	0

(11) A (A+B) = A

A	B	A+B	A(A+B)
0	0	0	0
1	0	1	1
0	1	1	0
1	1	1	1

(16) $\overline{A\,B}$ = $\overline{A}$ + $\overline{B}$

A	B	AB	$\overline{AB}$
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

$\overline{A}$	$\overline{B}$	$\overline{A}$+$\overline{B}$
1	1	1
0	1	1
1	0	1
0	0	0

Boolean Proofs by DeMorgan's Theorems

•F = A(A+B)
=AA+AB
= 0 + AB -8
= AB -5

•#6(a)
•F = ABCD + ABCD
= A(B+B)CD
= A(1)CD -7
= ACD

•#6(b)
•F = AB + AC + ABC(AB+C)
= AB + AC + ABCAB+ABCC)
= AB + AC + (AA)(BB)C+ABCC) 2,8
= AB + AC + (A)(0)C +ABC
= A(B+BC) + AC
= A(B+C) + AC 13
= AB +AC +AC =
= AB

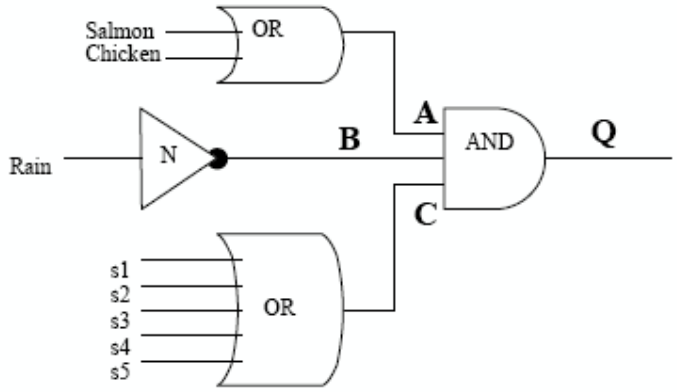
BOOLEAN LOGIC PROBLEMS

George Boole
(1815-1864) was an English mathematician and logician who devised a system for representing logical symbolic relationships now known as ***Boolean algebra****. The logical relationships, called* ***Boolean expressions****, use the logical operators AND, OR, and NOT between entities. These expressions have application in computer circuit design, information retrieval strategies, and logic problems such as this.*

#1- I want to go to a picnic. There are conditions to be met before I can go. Construct a logic gate circuit to determine if I may go on to the picnic. Gates with more than 2 inputs cay be used. 1 = yes 0 = no

- Krogers must have either baked chicken or salmon on sale.
- It must not be raining
- At least one of my five students has to go with me.

$$Q = \{\text{Chicken} \oplus \text{Salmon}\} \bullet \{\overline{rain}\} \bullet \{s1 \oplus s2 \oplus s3 \oplus s4 \oplus s5\}$$



TRUTH TABLE

A	B	C	Q
1	1	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Binary - base2

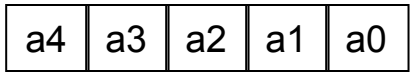
$$N_2 = a_0\,2^0 + a_1\,2^1 + a_2\,2^2 + a_3\,2^3 + \ = (a_3\,a_2\,a_1\,a_0)_2$$

Decimel	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^o
0	0	0	0	0	0	0	0	**0**
1	0	0	0	0	0	0	0	**1**
2	0	0	0	0	0	0	**1**	**0**
3	0	0	0	0	0	0	**1**	**1**
4	0	0	0	0	0	**1**	**0**	**0**
5	0	0	0	0	0	**1**	**0**	**1**
6	0	0	0	0	0	**1**	**1**	**0**
7	0	0	0	0	0	**1**	**1**	**1**
8	0	0	0	0	**1**	**0**	**0**	**0**
	MSB							LSB

Other Systems

Binary 2-digi 0,1

$N_2 = a_0\,2^0 + a_1\,2^1 + a_2\,2^2 + a_3 2^3 +$ …



Octal 8-digi 0,1,2,3,4,5,6,7

$N_8 = a_0\,8^0 + a_1\,8^1 + a_2\,8^2 + a_3 8^3 +$

Decimal 10-digi 0,1,2,3,4,5,6,7,8,9

$N_{10} = a_0\,10^0 + a_1\,10^1 + a_2\,10^2 + a_3 10^3 +$

Hexadecimal 16-digi 0,1,2,3,4,5,…..A,B,C,D,E,F

$N_{16} = a_0\,16^0 + a_1\,16^1 + a_2\,16^2 + a_3 16^3 +$

BCD binary coded decimal

$99_{10} = 1001\ 1001_{BCD}$

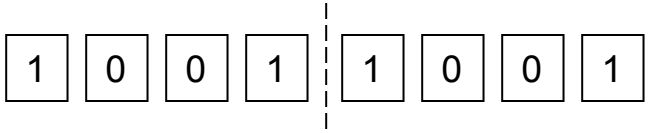


TABLE 11.13 Comparison of Numbering Systems

Decimal	Binary	BCD		Octal	Hexadecimal
0	0		0000	0	0
1	1		0001	1	1
2	10		0010	2	2
3	11		0011	3	3
4	100		0100	4	4
5	101		0101	5	5
6	110		0110	6	6
7	111		0111	7	7
8	1000		1000	10	8
9	1001		1001	11	9
10	1010	0001	0000	12	A
11	1011	0001	0001	13	B
12	1100	0001	0010	14	C
13	1101	0001	0011	15	D
14	1110	0001	0100	16	E
15	1111	0001	0101	17	F
16	10000	0001	0110	20	10
17	10001	0001	0111	21	11
18	10010	0001	1000	22	12
19	10011	0001	1001	23	13
20	10100	0010	0000	24	14
21	10101	0010	0001	25	15
22	10110	0010	0010	26	16
23	10111	0010	0011	27	17
24	11000	0010	0100	30	18
25	11001	0010	0101	31	19
99	110011	1001	1001	143	63

ASCII	Hex	Symbol
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	TAB
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI

ASCII	Hex	Symbol
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

ASCII	Hex	Symbol
32	20	(space)
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/

ASCII	Hex	Symbol
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

ASCII	Hex	Symbol
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O

ASCII	Hex	Symbol
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

ASCII	Hex	Symbol
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o

ASCII	Hex	Symbol
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	□

LOGIC FAMILIES

TTL Logic- Transistor-Transistor Logic gates
- Transistors are fully turned on.
- Excellent noise margin
- High power (1-20)mW/gate
- Single Vcc supply voltage.
- High>3-5V Low~0.2V

ECL Logic- Emitter Coupled Logic
- High speed by not saturating transistors.
- Td < 1ns (gate speed)
- Low Power (0.5-1.0) mW/gate
- Multiple supply voltages
- Higher fan-in capability

CMOS- Complementary Metal Oxide Logic
- Very Low Power
- Excellent Noise Margin
- Inexpensive
- Slower than TTL Td>10ns
- Fragile

Case of Binary Addition A+B=C

Carry	0	0	0	0	0	1	1	0	
A register	0	0	0	0	1	0	1	1	$A = 11_{10}$
+									
B register	0	0	0	0	0	0	0	1	$B = 01_{10}$
=	0	0	0	0	1	1	0	0	$C = 12_{10}$
	registers								

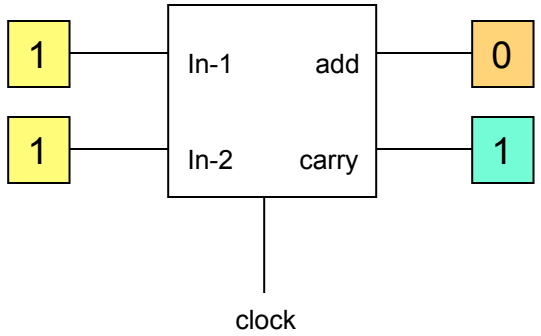
Base 2 notation

$1011_2 + 0001_2 = 1100_2$

Base 10 notation

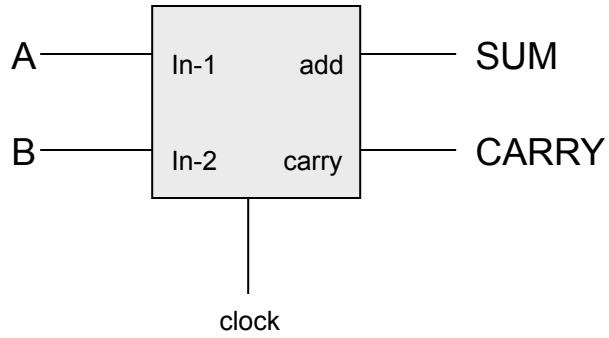
$11_{10} + 01_{10} = 12_{10}$

GATE/FLIP-FLOP



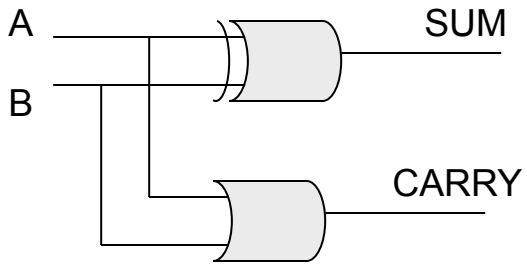
TRUTH TABLE

A	B	AND	NAND	OR	NOR	XOR
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

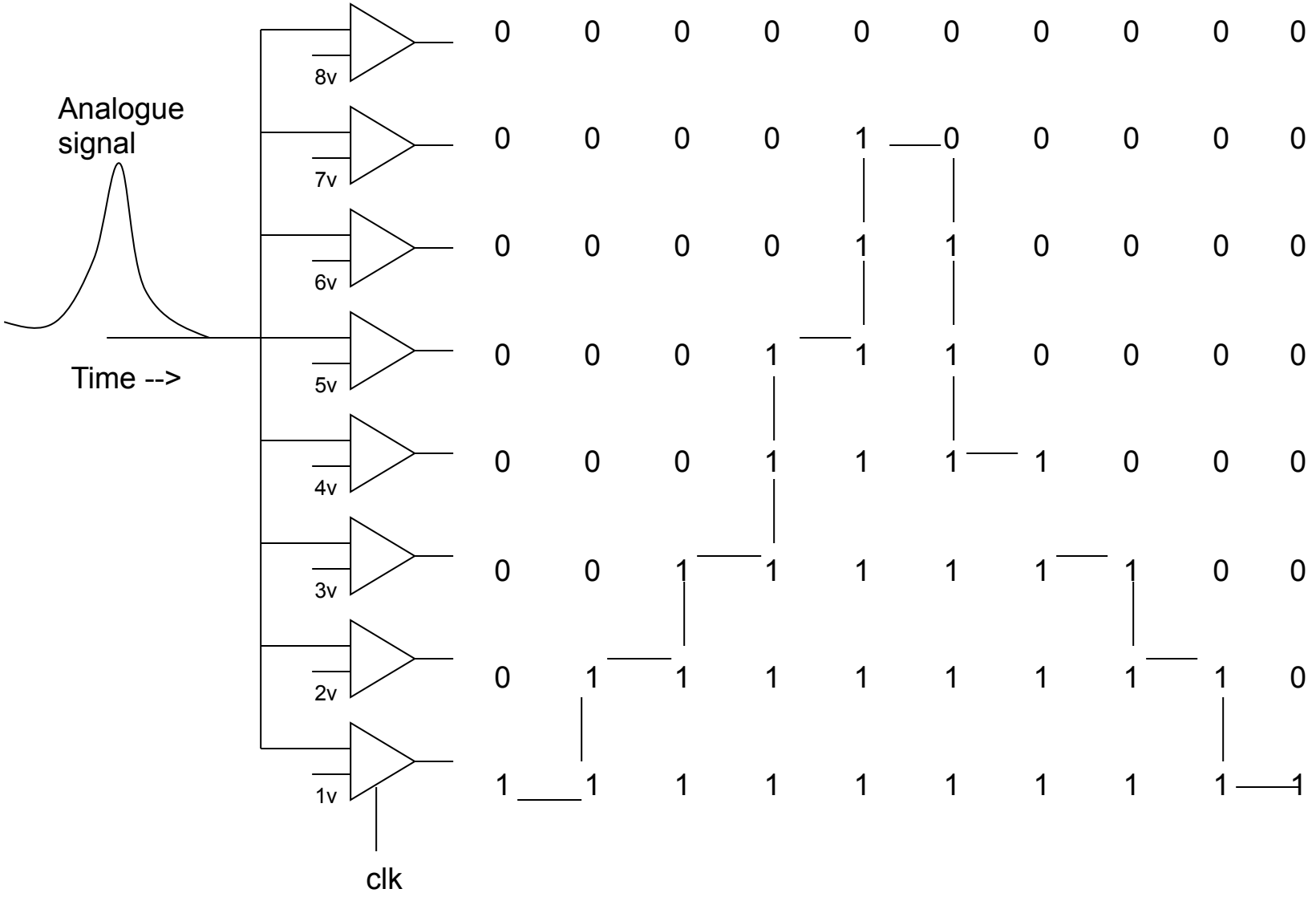


ADDITION

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
		XOR	OR



Analogue-to-Digital

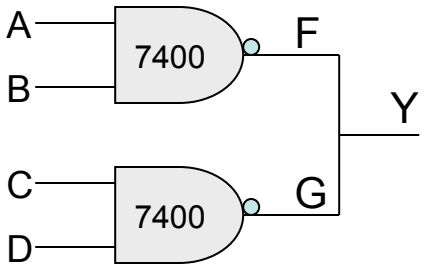


Open Collector and Tri-state Logic

•In standard ttl or cmos logic there is a problem with concatenating gates for an output. Consider Y=ABCD formed below from 2 NAND Gates.

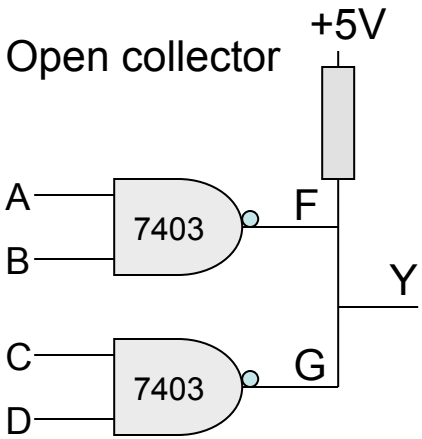
• When A=B=C=D=0 F=G=1 and Y=1
When A=0 B=C=D=1 F=1 G=0 and Y=?? indeterminant

Standard



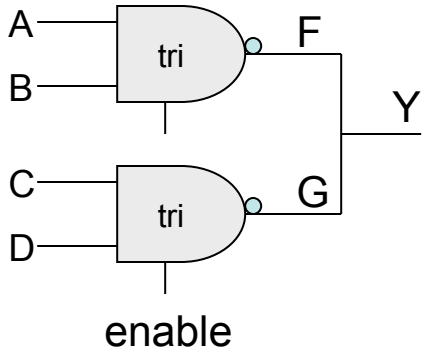
Output can be indeterminant!

Open collector



Outputs are pulled up to +5V when hi and virtual ground when low.

Tri-state



Tri-state chips can be enabled/disabled, effectvely setting them to infinite ground, *virtually* removing them from the circuit.

Epitaxial Gates