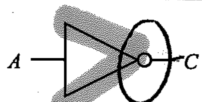
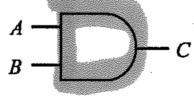
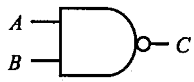
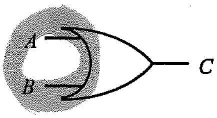
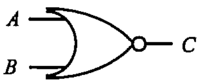
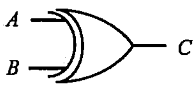
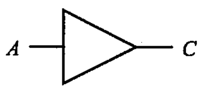
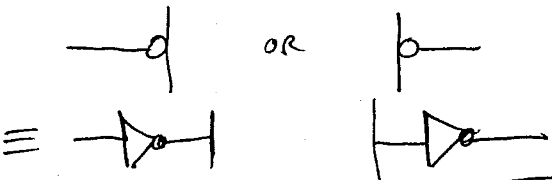
AVAIL ON IC DIP
(IN MULTIPLES OF
4 (ex: QUAD NAND IC)
OR 6 (ex: HEX INV IC)

SEE FIG. 6.25

Table 6.3 Combinational logic operations (p.203)

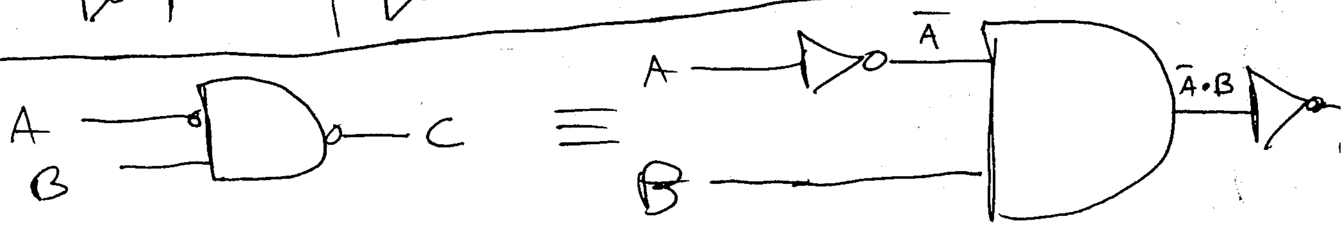
Gate	Operation	Symbol	Expression	Truth table
Inverter (IN[redacted] NOT) "V"	Invert signal (complement)		$C = \bar{A}$	A C 0 1 1 0
AN[redacted] gate "D"	AND logic		$C = A \cdot B$	A B C 0 0 0 0 1 0 1 0 0 1 1 1 — ON IF A AND B
NAND gate	Inverted AND logic		$C = \overline{A \cdot B}$	A B C 0 0 1 0 1 1 1 0 1 1 1 0
[redacted]R gate "O"	OR logic		$C = A + B$	A B C 0 0 0 0 1 1 1 0 1 1 1 1 — IF A OR B
NOR gate	Inverted OR logic		$C = \overline{A + B}$	A B C 0 0 1 0 1 0 1 0 0 1 1 0
XOR gate	Exclusive OR logic		$C = A \oplus B$ $= \bar{A}\cdot B + A \cdot \bar{B}$	A B C 0 0 0 0 1 1 1 0 1 1 1 0 — IF A OR B BUT NOT BOTH
Buffer	Increase output signal current		$C = A$	A C 0 0 1 1

THIS IS WHAT ALLOWS "SMART" DEVICES + SOFTWAR



⇒ INVERSION

Ex.)



TRUTH TABLE

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

nary
unt
eq.

$C = \overline{\bar{A} \cdot B}$

ex: A=0, B=1

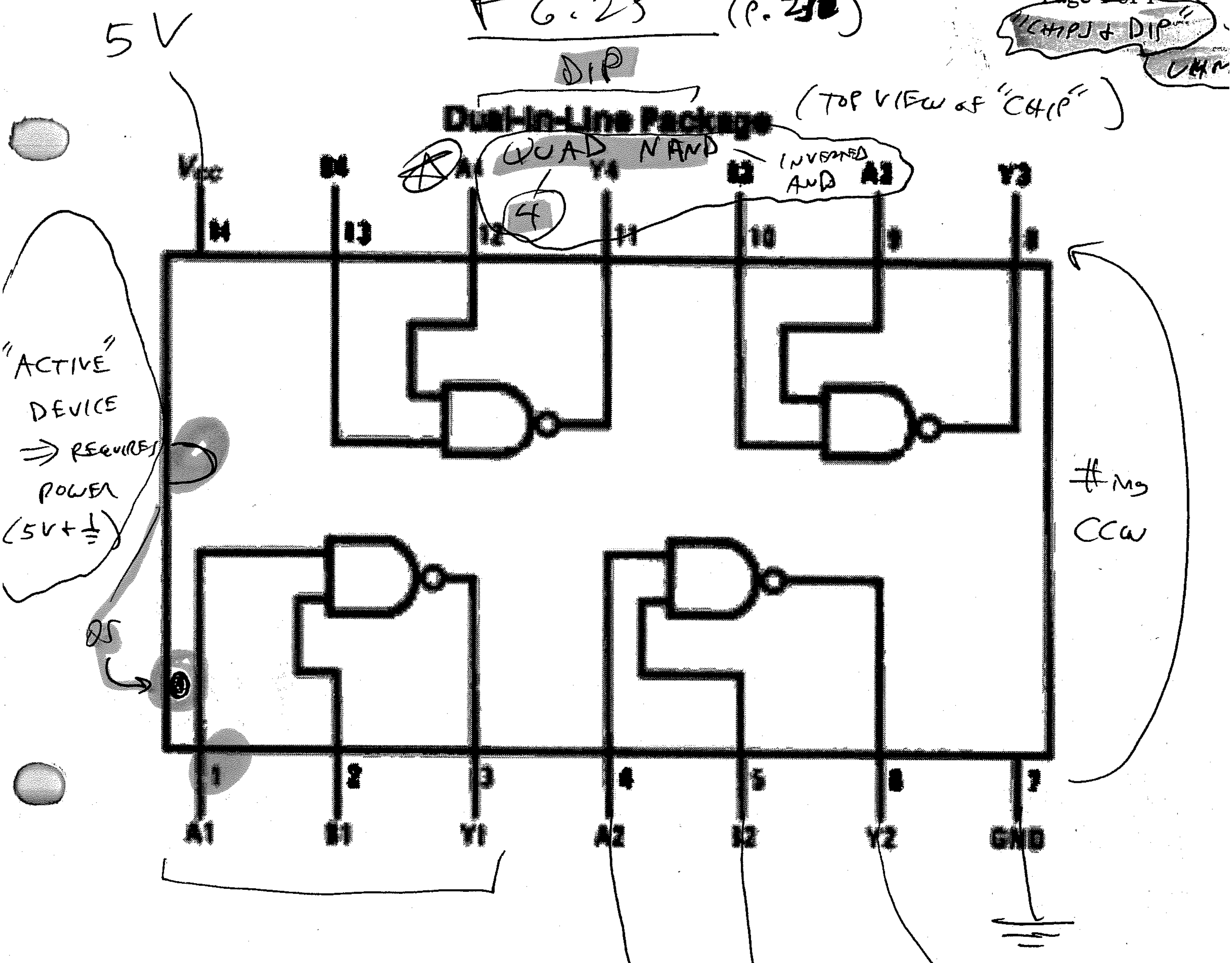
$C = \overline{\bar{0} \cdot 1} = \overline{1 \cdot 1} = \bar{1} = 0$

"CHIPS & DIP" UMM

F6.25 (p. 230)

DIP

(TOP VIEW of "CHIP")



$$Y1 = \overline{A1 \cdot B1}$$

NOT INVERSION HERE

Ex.) ⏚ (0V ≡ 0) ○ 5V (≡ 1)

$$\overline{0 \cdot 1} = \overline{0} = 1$$

⇒ ~5V

S6.5

$\cdot$: AND
$+$: OR
$\overline{}$: INV

P. 206

Boolean Algebra Laws and Identities

Fundamental Laws

TALK THROUGH EACH

OR	AND	NOT	
$A+0=A$	$A\cdot 0=0$		
$A+1=1$	$A\cdot 1=A$	$\overline{\overline{A}}=A$ (double inversion)	(6.10)
$A+A=A$	$A\cdot A=A$		
$A+\overline{A}=1$	$A\cdot\overline{A}=0$		

Commutative Laws

$$A+B = B+A \tag{6.11}$$

$$A\cdot B = B\cdot A \tag{6.12}$$

Associative Laws

$$(A+B)+C = A+(B+C) \tag{6.13}$$

$$(A\cdot B)\cdot C = A\cdot(B\cdot C) \tag{6.14}$$

SAME AS NORMAL ALGEBRA

Distributive Laws

$$A\cdot(B+C) = (A\cdot B)+(A\cdot C) \tag{6.15}$$

$$A+(B\cdot C) = (A+B)\cdot(A+C) \tag{6.16}$$

NOT

De Morgan's Laws

$$A+B+C+\cdots = \overline{\overline{A}\cdot\overline{B}\cdot\overline{C}\cdots} \tag{6.27}$$

$$A\cdot B\cdot C\cdots = \overline{\overline{A}+\overline{B}+\overline{C}+\cdots} \tag{6.28}$$

IMPORTANT

JUST SAY

ALLOWS CONVERTING BETWEEN ORs + ANDs (USU. TO REDUCE # OF REQUIRED "CHIPS")

ICs

S6.5 ex:

$\overline{1\cdot\overline{0}} + 1\cdot(0+1) + \overline{\overline{0}}\cdot(1+$

$= \overline{1\cdot 1} + 1\cdot 1 + 0\cdot 1$

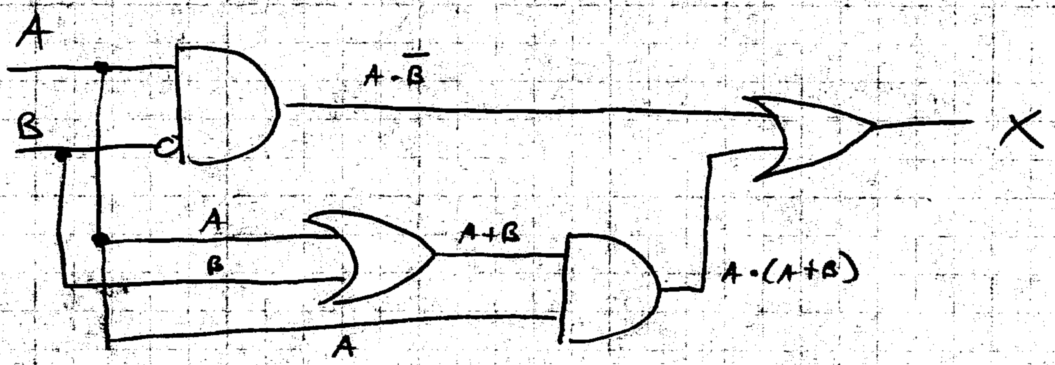
$= \overline{1} + 1 + 0$

$= 0 + 1 + 0$

$= 1$

Ex Simplifying a Boolean Expression

Find the simplest possible Boolean expression representing X in the circuit below:



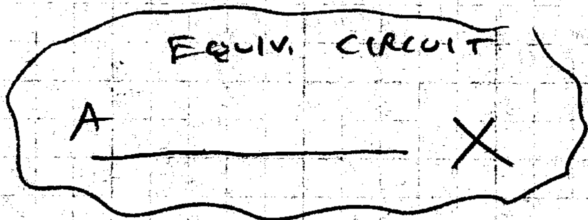
$$X = A \cdot \overline{B} + A \cdot (A+B)$$

factor out A $\Rightarrow$ $= A(\overline{B} + A + B)$

$B + \overline{B} = 1$ $\Rightarrow$ $= A(A+1)$

$A+1 = 1$ $\Rightarrow$ $= A \cdot 1$

$A \cdot A = A$ $\Rightarrow$ $\boxed{X = A}$



$\Rightarrow$ SAVE LOTS of \$ (ESP. IF MAKING 100,000)

Ex. 1 USING DE MORGAN'S LAWS

QUAD AND IC (4 ANDS USED)
QUAD OR IC (4 ORS, only 2 USED)
6 ~~HEX~~ INV IC (ONLY 1 INV USED)

FROM EXAMPLE 6.3

$$X = (A \cdot B \cdot C) + (B \cdot C) + (\bar{A} \cdot B)$$

4 ANDS
2 ORS
1 INV
(3 ICs, lots of wiring)

SIMPLIFY

$$X = B \cdot (C + \bar{A})$$

1 AND
1 OR
1 INV
(STILL 3 ICs!)

BUT LESS WIRING

CONVERTING TO ALL "AND"

$$\alpha + \beta = \overline{\bar{\alpha} \cdot \bar{\beta}} \Rightarrow X = B \cdot (\overline{\bar{C} \cdot \bar{\bar{A}}})$$

$$= B \cdot (\overline{\bar{C} \cdot A}) \quad (1)$$

(2 AND, 2 INV)

CONVERTING TO ALL "OR"

$$\alpha \cdot \beta = \overline{\bar{\alpha} + \bar{\beta}} \Rightarrow X = \overline{\bar{B} + \overline{C + \bar{A}}} \quad (2)$$

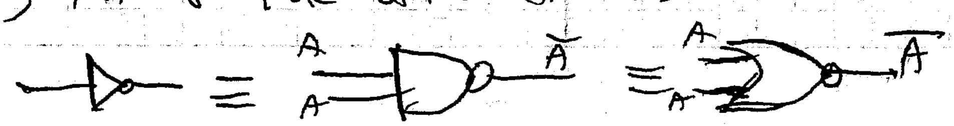
(2 OR, 4 INV)

(1) + (2) BOTH REQUIRE ONLY 2 ICs (QUAD AND or QUAD OR and ~~HEX~~ INV)

$\Rightarrow$ SAVE MONEY

(1) REQUIRES LESS WIRING THAN (2) $\Rightarrow$ SAVE MORE MONEY

ALSO, NAND & NOR GATES CAN BE USED TO MAKE INVERTERS



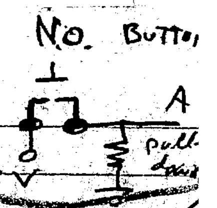
SKIP

SHOW EACH CIRCUIT

Up (open) ⇒ A = 0 (0V)

down (closed) ⇒ A = 1 (5V)

N.O. BUTTON



Q

Logic Circuit

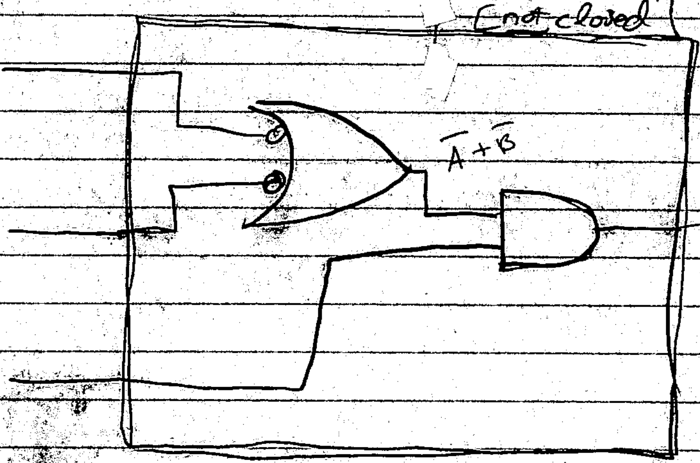
Automobile door / seatbelt buzzer

If ignition on and (door open [not closed] or seat belt not fastened), THEN sound buzzer

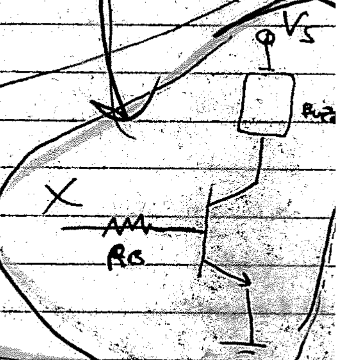
Door switch (ON ⇒ closed)

Seat Belt switch (ON ⇒ fastened)

ignition switch (ON ⇒ active)



Logic Circuit



BOOLEAN EXPRESSION

$$X = C \cdot (\bar{A} + \bar{B})$$

SAY IN WORDS

Truth Table

COVER SOME OF THIS NEXT CLASS IF NO TIME

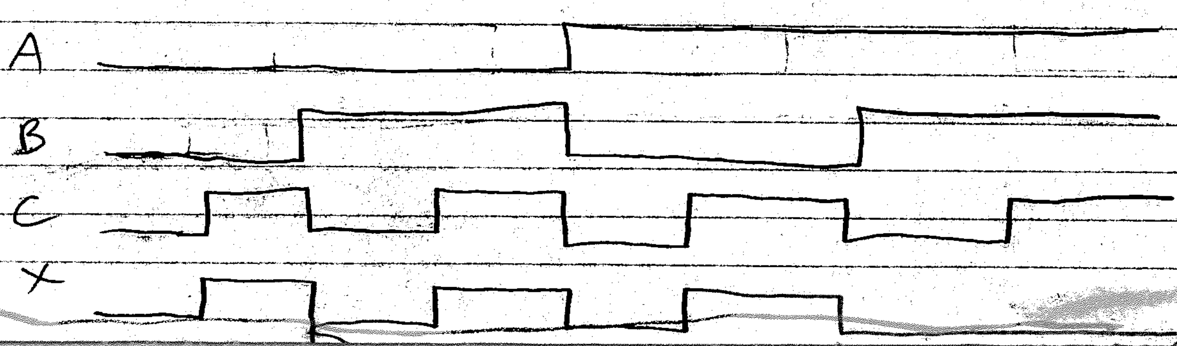
A	B			X
0	0			0
0	0			1
0	1	0		0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

8 combinations (permutations)

BINARY COUNTING SEQUENCE

SKIP IF NO TIME

Timing Diagram



Subject: add to 307 website and as Internet Link in book
From: David Alciatore <David.Alciatore@colostate.edu>
Date: Fri, 02 Oct 2009 14:51:27 -0600
To: David Alciatore <David.Alciatore@colostate.edu>

David Alciatore wrote:

> MECH307,
>
> Check out this cool website, where you can easily create and simulate logic circuits (see below).
>
> Dr. Dave
>
> A student wrote:
>
> > http://joshblog.net/projects/logic-gate-simulator/Logicly.html
> >
> > Saves time by allowing you to test a logic array before you build it!
>
> ---
>
> David G. Alciatore, PhD, PE ("Dr. Dave")
> Department of Mechanical Engineering
> Colorado State University
> 1374 Campus Delivery
> Fort Collins, CO 80523-1374
>
> ph: (970) 491-6589
> David.Alciatore@colostate.edu
> www.engr.colostate.edu/~dga
>
> ---

SHOW CIRCUIT OPERATING HERE

--

David G. Alciatore, PhD, PE ("Dr. Dave")
Department of Mechanical Engineering
Colorado State University
1374 Campus Delivery
Fort Collins, CO 80523-1374

ph: (970) 491-6589
David.Alciatore@colostate.edu
www.engr.colostate.edu/~dga

DOWNLOAD TRIAL VERSION IF NOT WORKING IN BROWSER