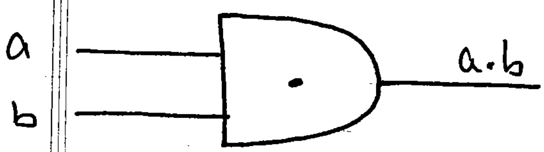
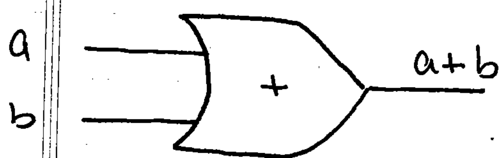
AND GATE



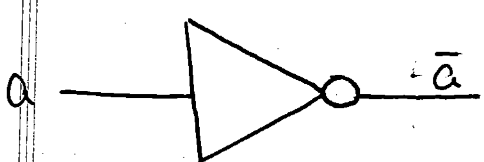
a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE



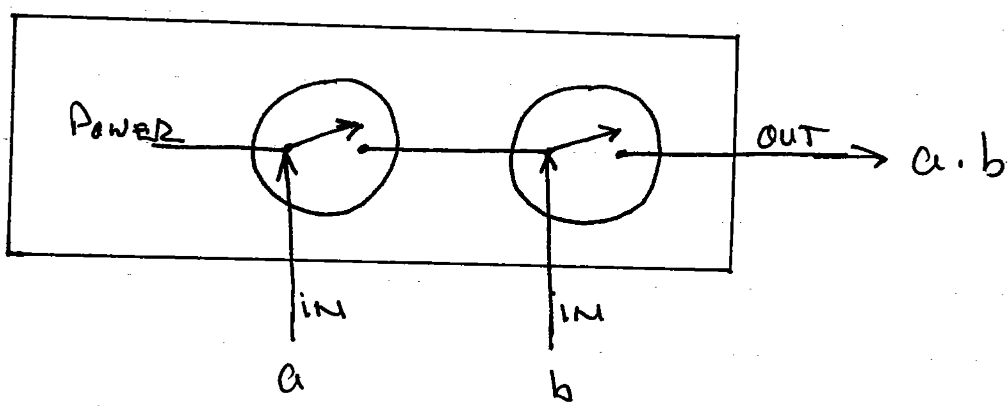
a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE



a	$\bar{a}$
0	1
1	0

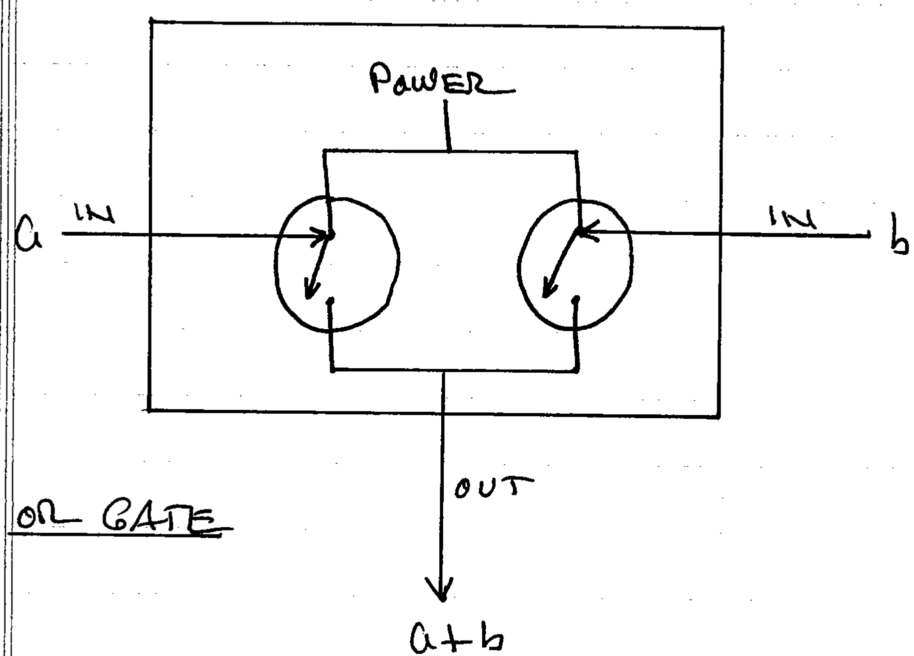
AN AND GATE CAN BE CONSTRUCTED FROM TWO TRANSISTORS BY PLACING THEM IN SERIES.



AND GATE

IF BOTH a AND b ARE ON (i.e. SET TO 1) BOTH TRANSISTORS ARE IN THE ON STATE, AND THE OUTPUT LINE IS CONNECTED TO THE POWER SUPPLY (i.e. IS 1). IF EITHER OF a OR b IS 0, THEN AT LEAST ONE TRANSISTOR IS IN THE OFF STATE, WHENCE THERE IS NO POWER TO OUT (i.e. 0).

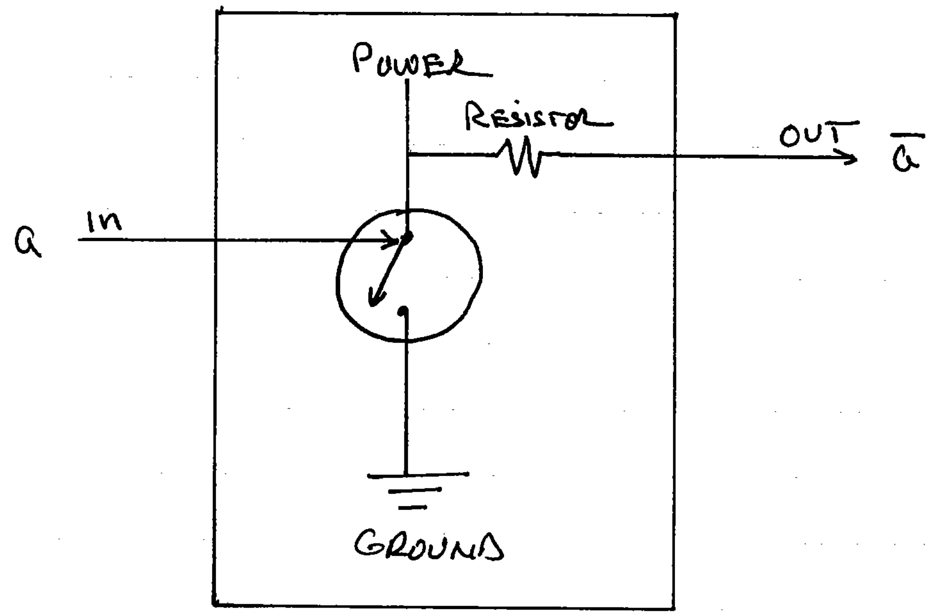
AN <u>OR</u> GATE CAN BE CONSTRUCTED BY PLACEING TWO TRANSISTORS IN PARALLEL.



<u>OR GATE</u>

IF EITHER a OR b IS 1, THEN AT LEAST ONE TRANSISTOR IS ON AND POWER IS CONNECTED TO OUT, GIVING 1. IF BOTH a AND b ARE 0, THEN BOTH TRANSISTORS ARE OFF AND POWER IS NOT CONNECTED TO OUT, GIVING 0.

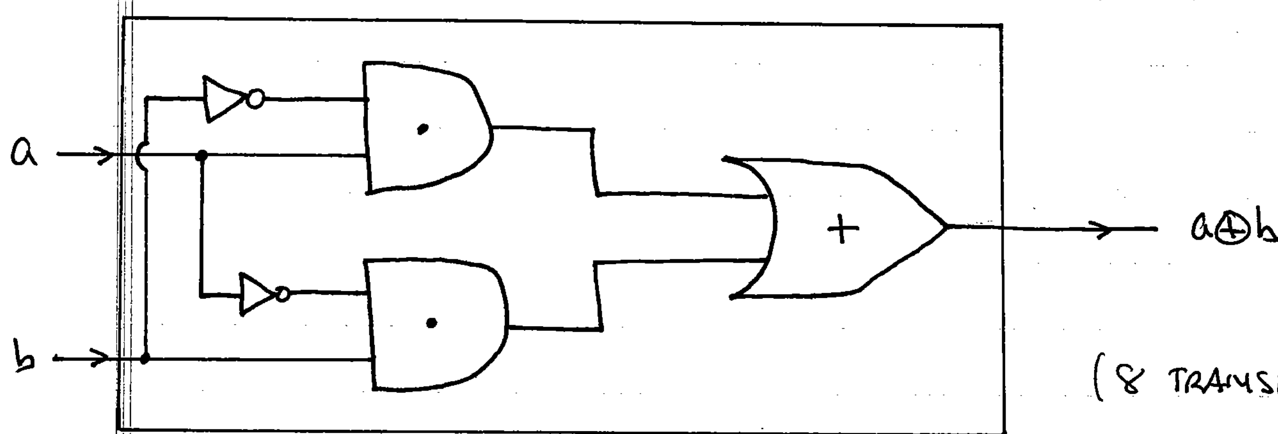
A <u>NOT</u> GATE CAN BE CONSTRUCTED FROM A SINGLE TRANSISTOR AND ANOTHER COMPONENT CALLED A <u>RESISTOR</u>.



GROUND CAN BE THOUGHT OF HERE AS A SPONGE WHICH SOAKS UP ELECTRIC CURRENT. IF GIVEN A CHOICE, CURRENT FROM THE POWER SUPPLY WOULD RATHER FLOW TO GROUND THAN THROUGH THE RESISTOR. IF a IS 1, THE TRANSISTOR IS ON AND POWER IS CONNECTED TO GROUND, SO NO CURRENT FLOWS TO OUT, GIVING 0. IF a IS 0, THE SWITCH IS OPEN FORCING CURRENT THROUGH THE RESISTOR TO OUT, GIVING 1.

YOU MAY WONDER WHY WE HAVE NO XOR GATE. IT IS POSSIBLE TO BUILD AN XOR GATE FROM AND, OR, AND NOT GATES. RECALL THE TRUTH TABLE FOR XOR:

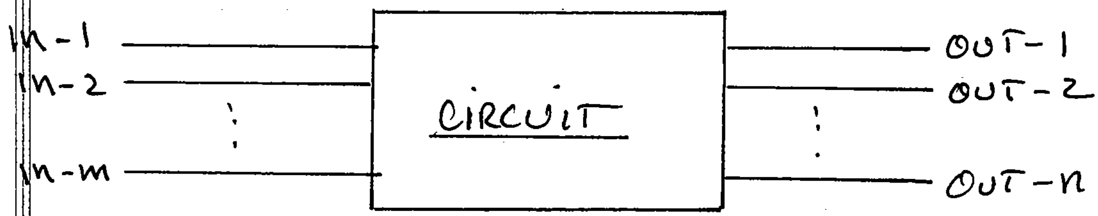
a	b	$a \oplus b \equiv (a \cdot \bar{b}) + (\bar{a} \cdot b)$
0	0	0
0	1	1
1	0	1
1	1	0



(8 TRANSISTORS)

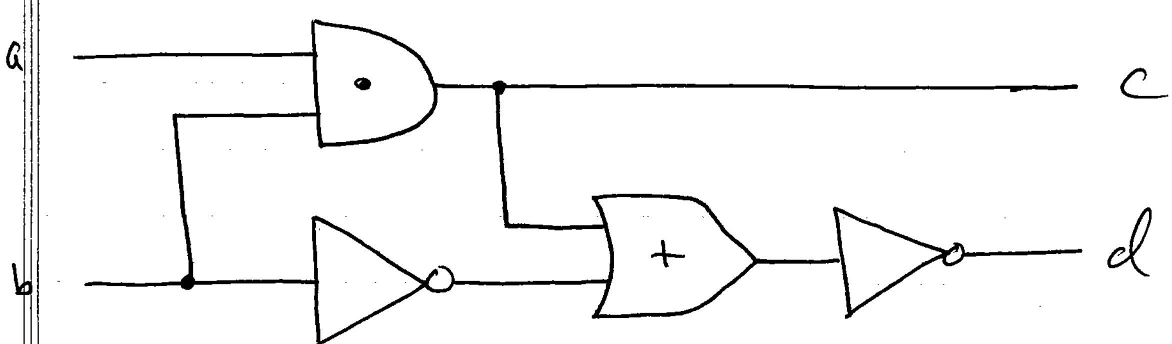
XOR GATE

A (COMBINATIONAL) CIRCUIT IS A COLLECTION OF LOGIC GATES WHICH TRANSFORMS A SET OF BINARY INPUTS INTO A SET OF BINARY OUTPUTS.



IT IS REQUIRED THAT EACH OUTPUT DEPENDS ONLY ON THE CURRENT VALUES OF THE INPUTS. (A SEQUENTIAL CIRCUIT IS ONE IN WHICH SOME OUTPUTS MAY DEPEND ON PREVIOUS VALUES OF THE INPUTS.)

EX.



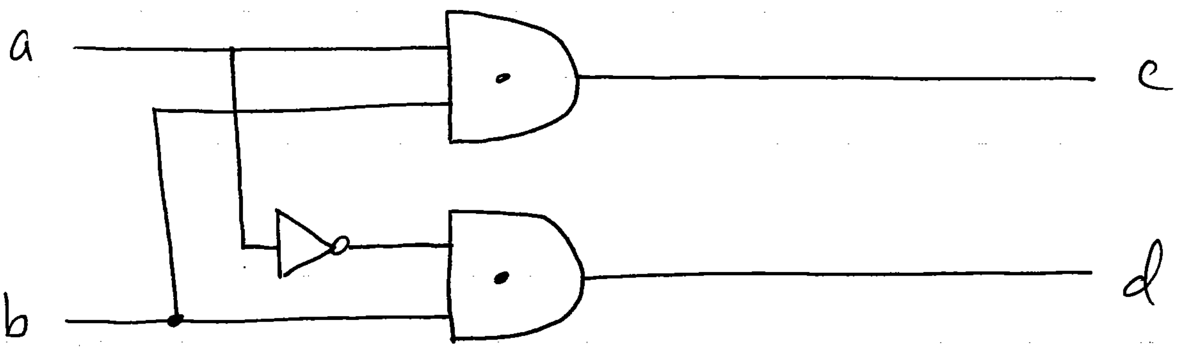
THE TRUTH TABLE FOR THIS CIRCUIT IS

INPUTS		OUTPUTS	
a	b	c	d
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

WE SEE THAT c AND d ARE THE LOGICAL EXPRESSIONS

$$c \equiv a \cdot b \quad \text{AND} \quad d \equiv \bar{a} \cdot b \, .$$

THIS SUGGESTS AN EQUIVALENT CIRCUIT DESIGN:



THUS DIFFERENT CIRCUITS CAN ACCOMPLISH THE SAME TASK. LOOKING AT THE ORIGINAL CIRCUIT WE SEE

$$c \equiv a \cdot b \quad \text{AND} \quad d \equiv \overline{(a \cdot b) + \bar{b}} \, .$$

EQUIVALENCE OF THE ABOVE CIRCUITS IMPLIES THE EQUIVALENCE OF THE LOGICAL EXPRESSIONS FOR d:

$$\overline{(a \cdot b) + \bar{b}} \equiv \bar{a} \cdot b$$

EXERCISE: PROVE THIS

IN GENERAL WE HAVE THE CORRESPONDENCES:

$$\begin{tikzcd}
\underline{\text{CIRCUITS}} \arrow[rr, leftrightarrow, "\text{1 TO 1}"] \arrow[dr, "\text{MANY TO 1}"'] & & \underline{\text{LOGICAL EXPRESSIONS}} \arrow[dl, "\text{MANY TO 1}"] \\
& \underline{\text{TRUTH TABLES}} &
\end{tikzcd}$$

<u>CIRCUIT OPTIMIZATION</u> IS THE PROCESS OF FINDING A CIRCUIT WITH THE FEWEST COMPONENTS (i.e. TRANSISTORS) TO PERFORM A GIVEN TASK (i.e. WITH A GIVEN TRUTH TABLE.)

ANY (COMBINATIONAL) CIRCUIT CAN BE CONSTRUCTED BY COMBINING AND, OR, AND NOT GATES.

ORDINARILY WE START WITH A TRUTH TABLE, THEN WRITE A CORRESPONDING BOOLEAN EXPRESSION AND DESIGN A CIRCUIT.

<u>EX.</u> CONSTRUCT A 1-BIT COMPARE FOR EQUALITY CIRCUIT: GIVEN INPUTS a, b:

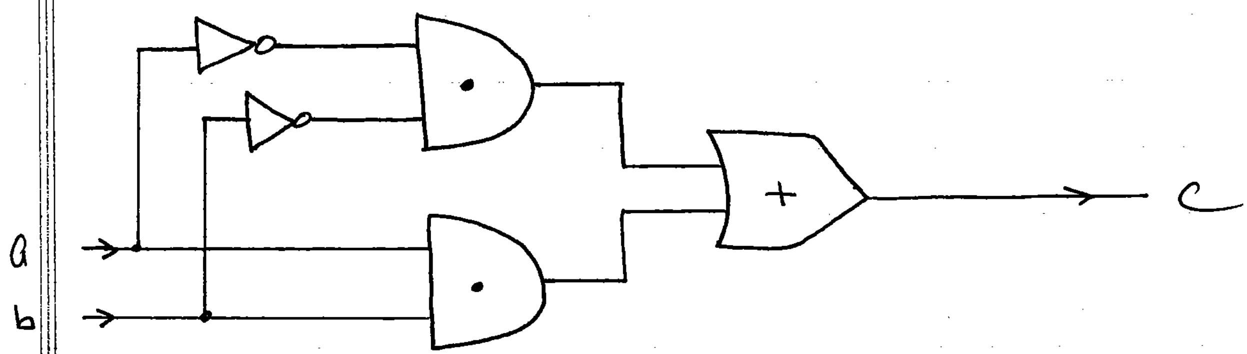
INPUT a	INPUT b	OUTPUT c
0	0	1
0	1	0
1	0	0
1	1	1

OUTPUT c IS 1 IF THE TWO INCOMING BITS ARE EQUAL, 0 OTHERWISE

OBSERVE THAT c is EQUIVALENT TO:

$$c \equiv (\bar{a} \cdot \bar{b}) + (a \cdot b)$$

ONE CHECKS THIS BY TRUTH TABLE ANALYSIS. THIS "SUM OF PRODUCTS" GIVES THE FOLLOWING CIRCUIT DIAGRAM.



EX. CONSTRUCT A CIRCUIT WHICH COMPARES TWO n-BIT BINARY NUMBERS AND RETURNS 1 IF THEY ARE EQUAL, 0 OTHERWISE.

WE USE THE RESULT OF THE LAST EXAMPLE AND SYMBOLIZE THAT CIRCUIT BY