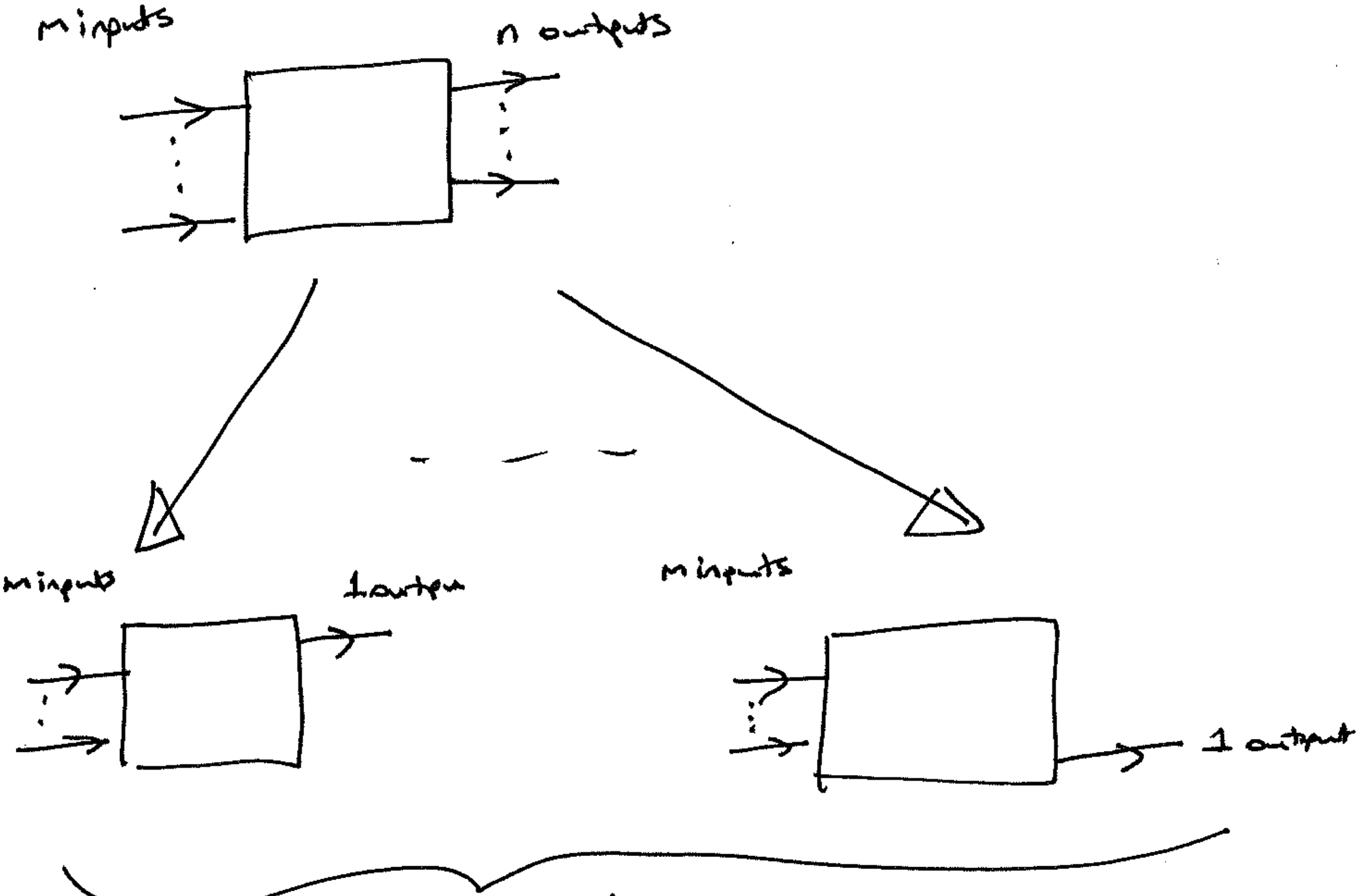
Lecture 24

<u>Goal</u> To show that every combinational circuit can be constructed from gates

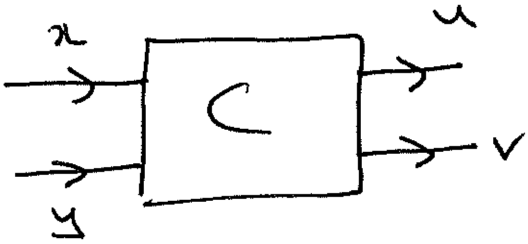


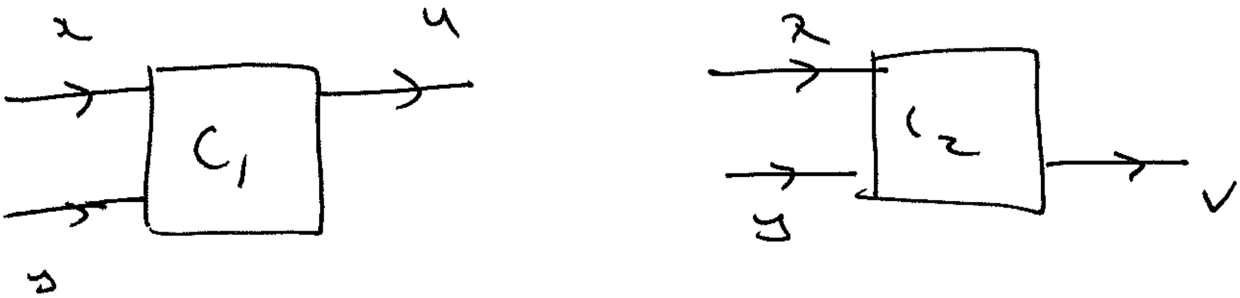
n circuits ~~representing~~ producing each of the n outputs

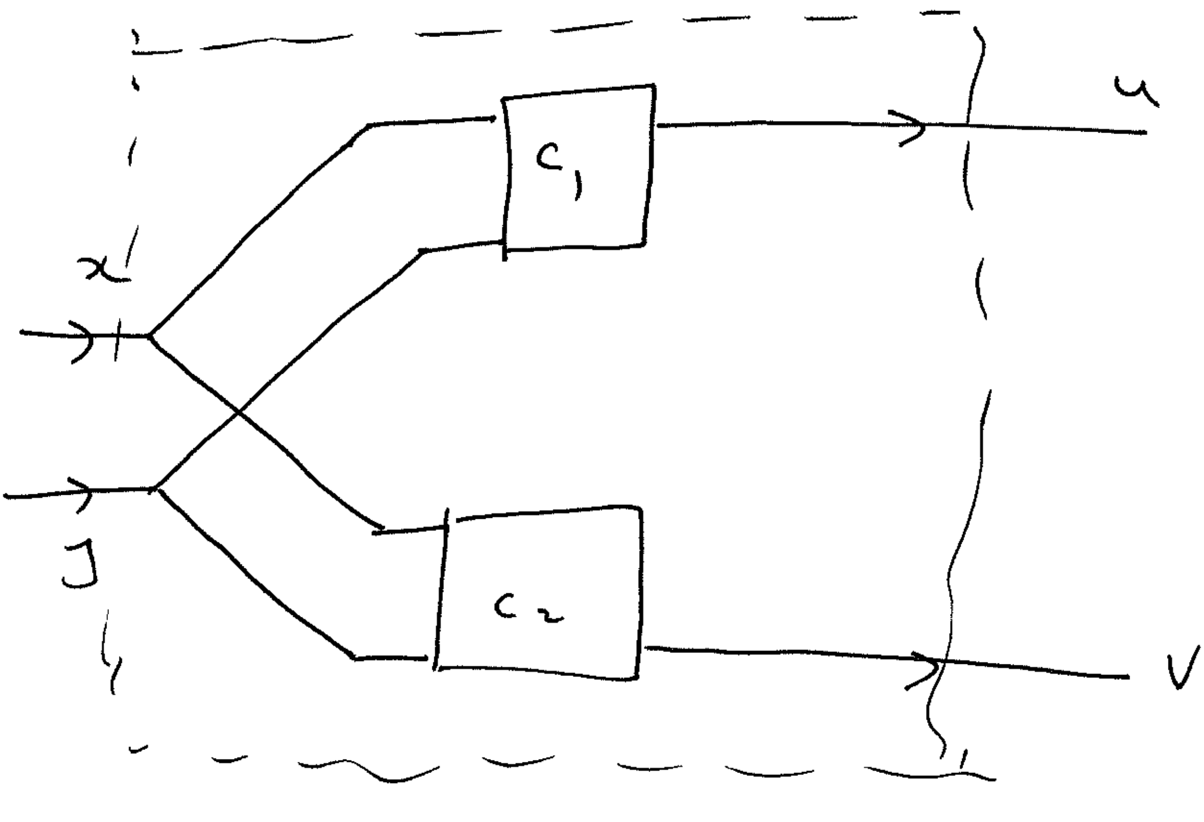
~~Thus here we restrict our attention to machines with m inputs and 1 output~~.

We can combine these n circuits using fanout.

Example







Thus we can restrict our attention to circuits with m-inputs and 1-output.

Example / Method

Input / output table of a combinational circuit.

input			output	
x	y	z	u	
1	1	1	0	
1	1	0	1	①
1	0	1	0	
1	0	0	1	②
0	1	1	0	
0	1	0	1	③
0	0	1	0	
0	0	0	0	

Look at the places where the output is 1.

Write down what are called <u>minterms</u>

① $x y \bar{z}$

② $x \bar{y} \bar{z}$

③ $\bar{x} y \bar{z}$

It follows that

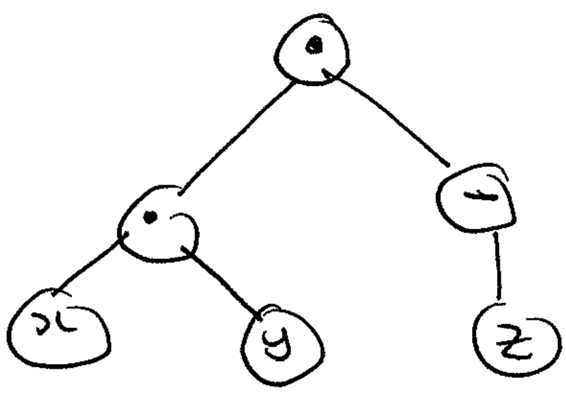
$$u = xy\overline{z} + x\overline{y}\overline{z} + \overline{x}y\overline{z}$$

This is a Boolean expression that describes exactly the input/output behaviour.

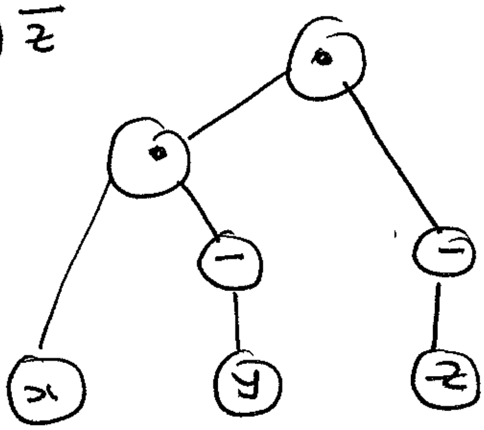
We now convert this expression into a circuit — we shall need to introduce brackets.

First, write down parse trees

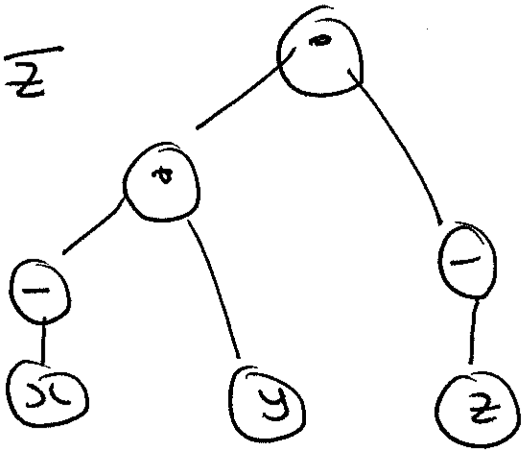
$(xy)\overline{z}$



$(x\bar{y})\bar{z}$

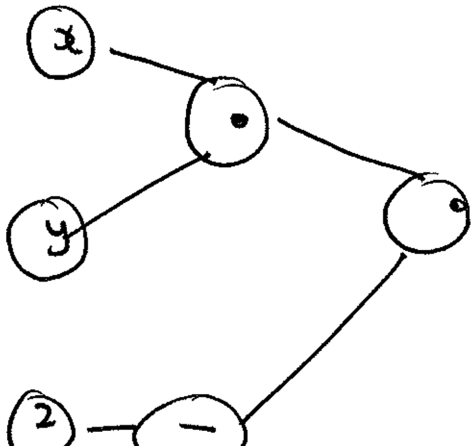


$(\bar{x}y)\bar{z}$

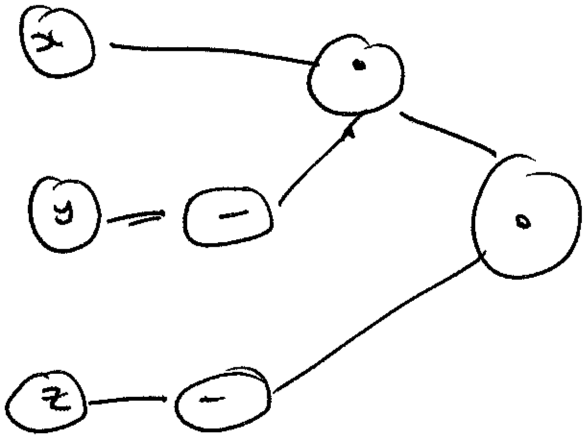


Next, rotate each parsetree 90° clockwise.

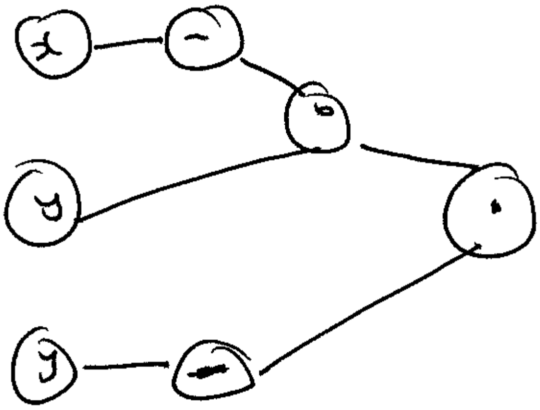
1



(2)

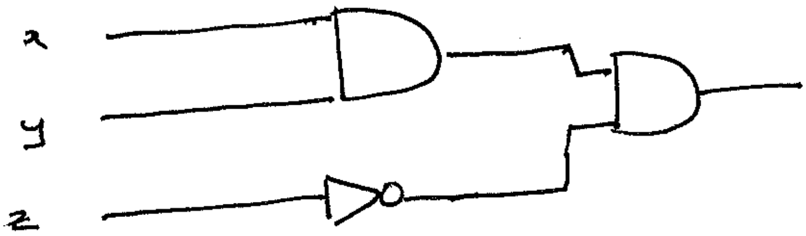


(3)

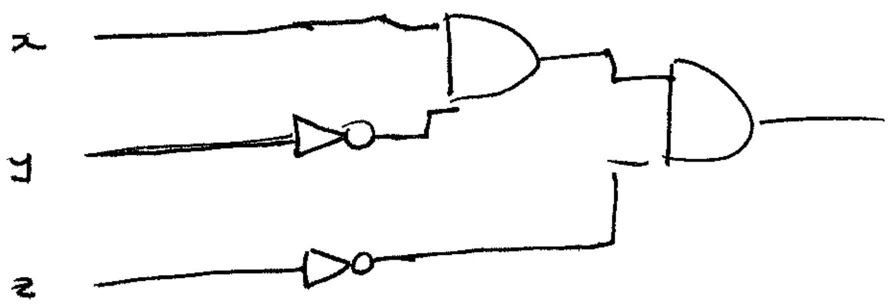


Now replace each Boolean operation by an appropriate gate.

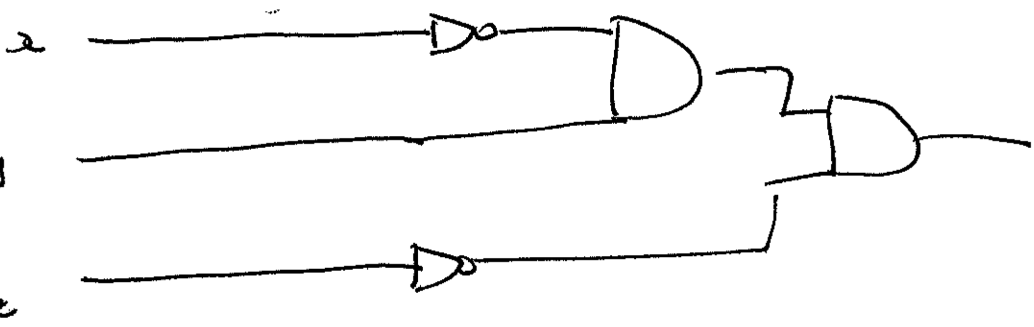
1



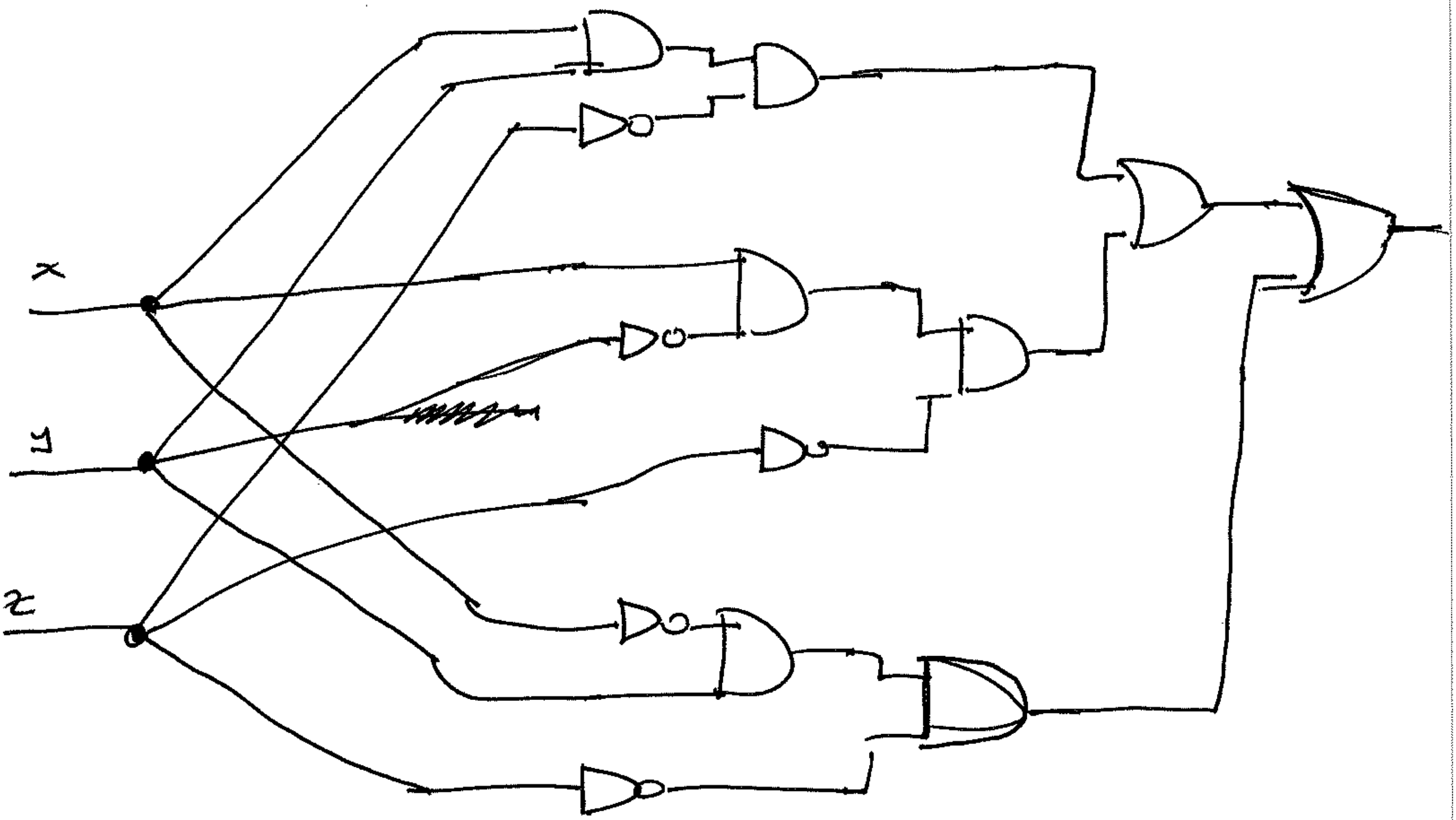
2



3



Now combine these three circuits using fanout and or-gates $\left((xy)\overline{z} + (x\overline{y})\overline{z}\right) + (\overline{x}y)\overline{z}$

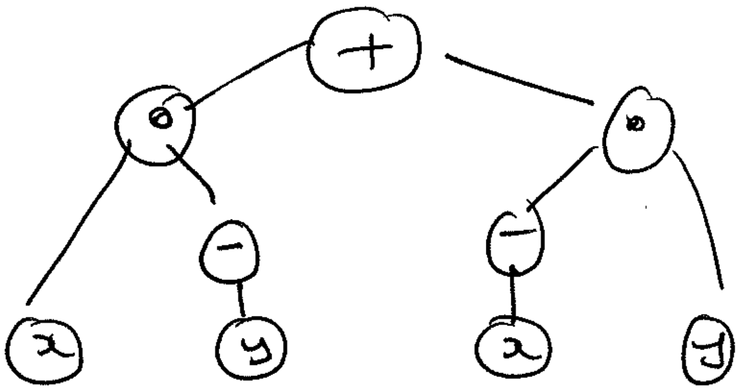


Example

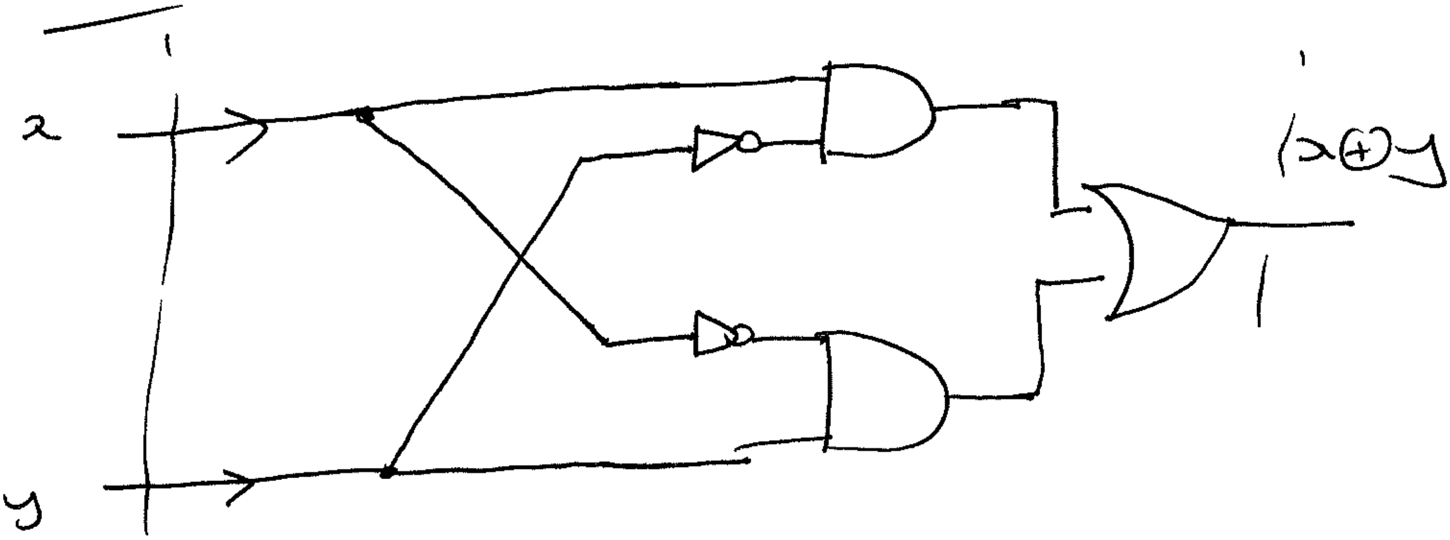
x	y	$x \oplus y$
1	1	0
1	0	1
0	1	1
0	0	0

$$x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y$$

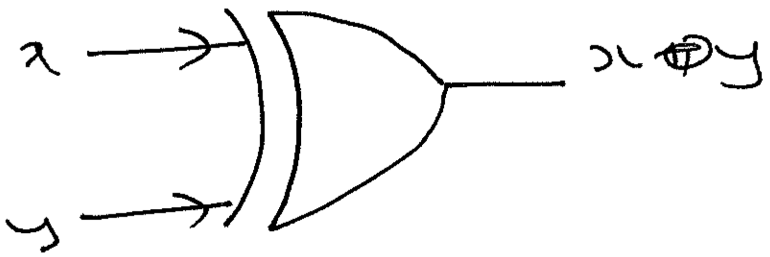
parse tree



Circuit.

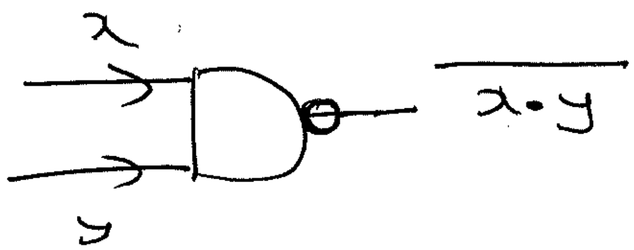


This is <u>exclusive-or</u> and has its own symbol

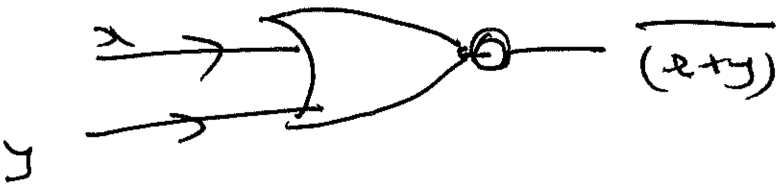


Other gates

nand-gates



nor gates



<u>[illegible] or work in PL</u>

Theorem

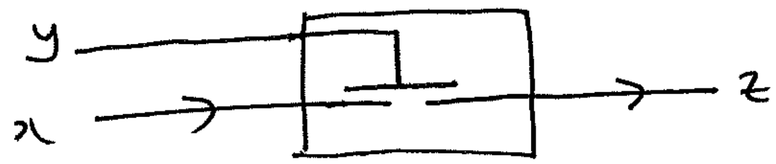
(1) Every combinational circuit can be constructed from and-gates, or-gates and not-gates

(2) Every combinational circuit can be constructed from <u>nand</u>-gates (resp. nor-gates).

Transistors

These are electronic switches

Idea



If $y = 0$ then $z = x$.

If $y = 1$ then $z = 0$.

Input/output table

ad hoc symbol

x	y	$x \,\square\, y$
1	1	0
1	0	1
0	1	0
0	0	0

$$x \,\square\, y = x \cdot \overline{y}$$

<u>Theorem</u> Every combinational circuit can be constructed from transistors.

<u>Proof.</u> Enough to show that transistors can be used to construct and-gates, or-gates, not-gates

- $1 \square x = 1 \cdot \overline{x} = \overline{x}$. not-gates.

- $x \square \overline{y} = x \cdot \overline{\overline{y}} = x \cdot y$ and gates

- ~~$((1 \square x)) \square y$~~

$$1 \square \left[(1 \square x) \square y \right]$$

$$= 1 \; \overline{\left((1 \square x) \square y \right)}$$

$$= \overline{\left((1 \square x) \square y \right)}$$

$$= \overline{\left(\left(1\, \overline{x} \right) \square\, y \right)}$$

$$= \overline{\left(\overline{x} \,\square\, y \right)}$$

$$= \overline{\left(\overline{x}\, \overline{y} \right)} \quad = \quad x + y \quad \text{wyctes} \quad \blacksquare$$