CSCI 210: Computer Architecture
Lecture 14: Digital Logic

Stephen Checkoway
Slides from Cynthia Taylor

CS History: The Manchester Transistor Computer

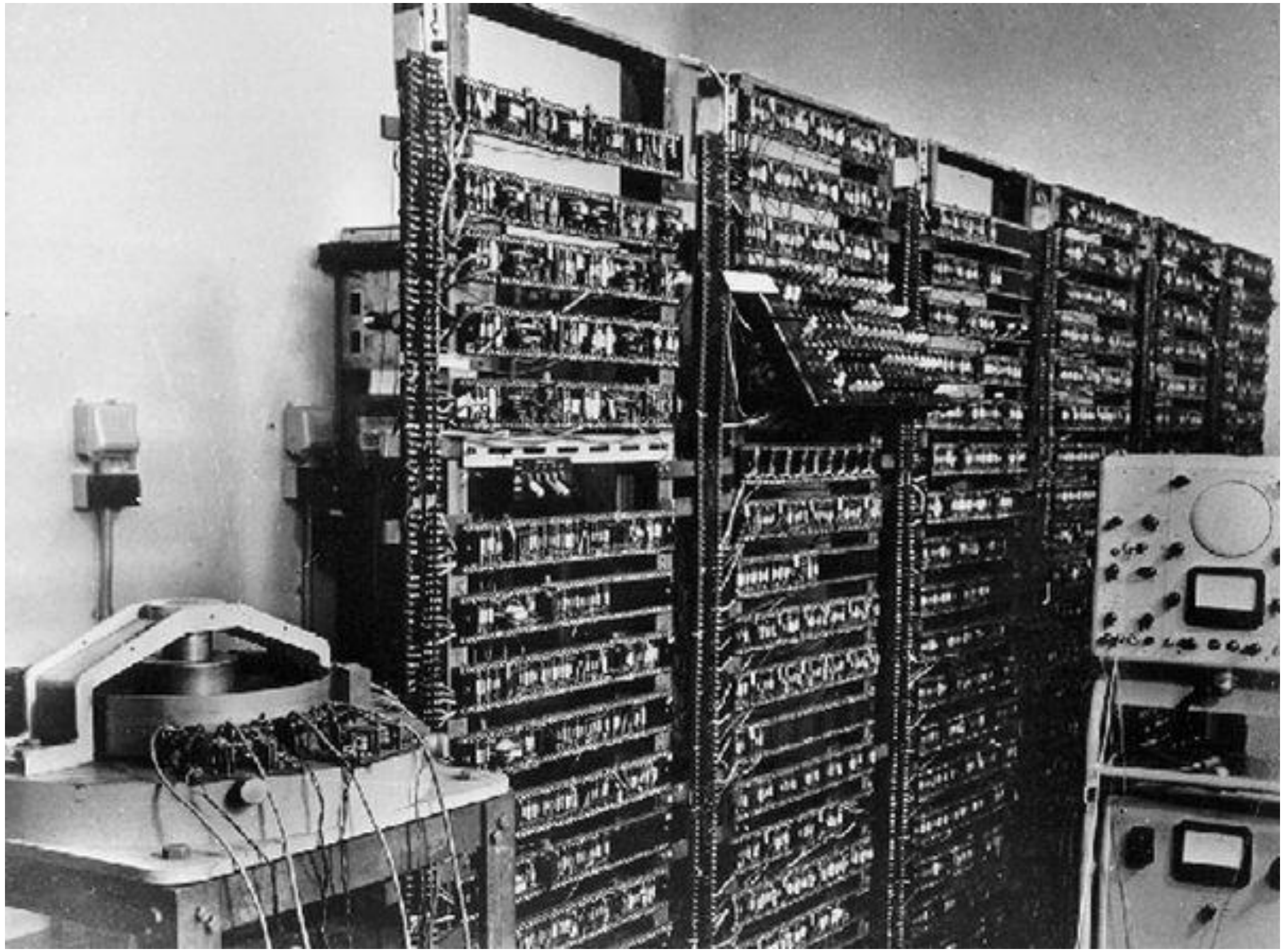


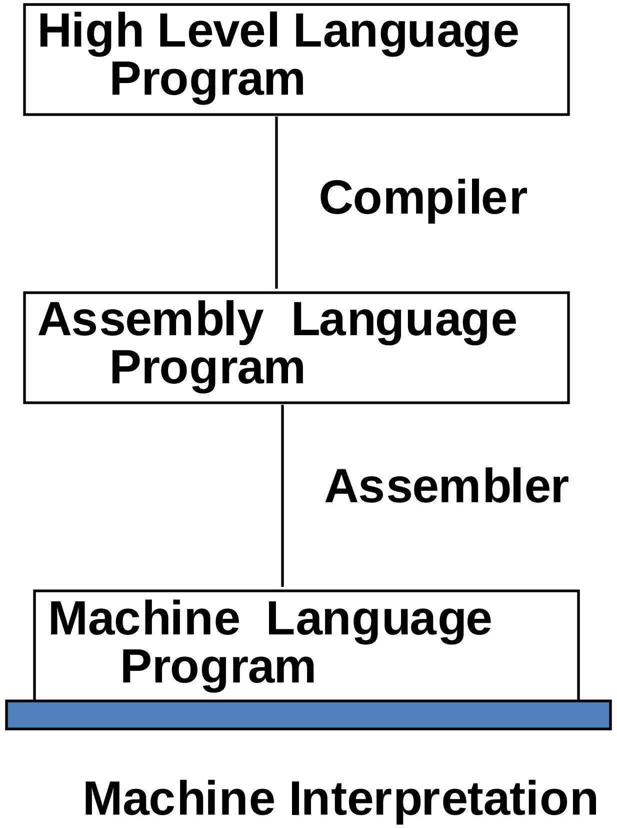
Image credit: The University of Manchester

- First computer to use transistors
- Developed at University of Manchester in 1953
- Problems with the reliability of early batches of transistors meant that its mean time between failures was about 90 minutes
- Still used valves for its clock and memory, so not fully transistorized

Creating the Universe from 1 and 0

- We have seen how to build programs from assembly
- Now we’ll learn how we implement assembly language instructions using circuits

Machine Interpretation




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw  $15,   0($2)
lw  $16,   4($2)
sw  $16,   0($2)
sw  $15,   4($2)
```

```
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
```

Machine does something!

A digital circuit is comprised of signals, gates, and wires

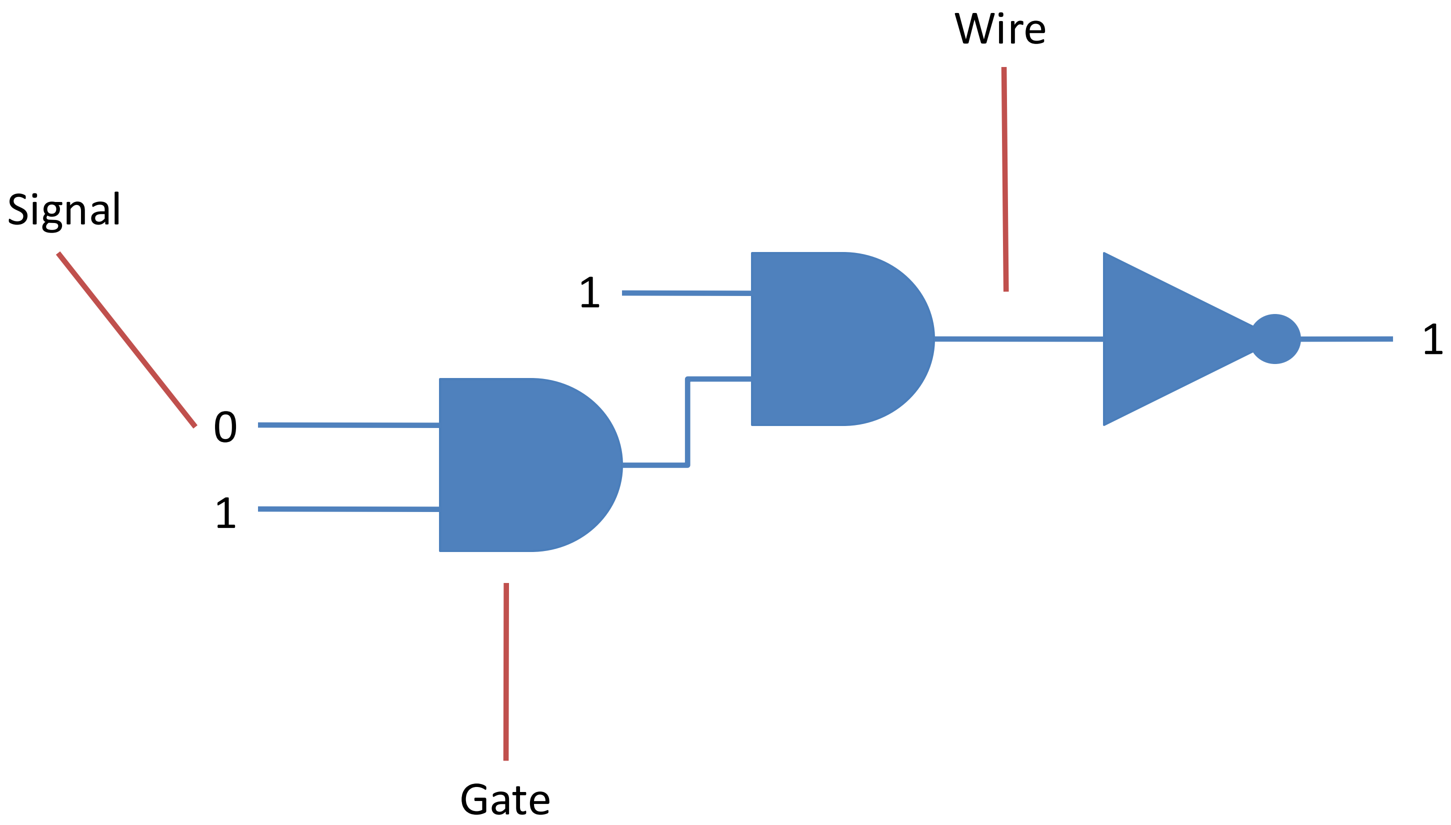
- Signals
  - Voltages applied to wires which generate electric current

- Binary signals are represented by different voltages:
  - 0: 0-1 volts
  - 1: 2-5 volts

A digital circuit is comprised of signals, gates, and wires

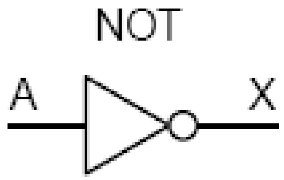
- Gates
  - Devices which perform operations on signals corresponding to basic logic operations: and, or, not, nand, nor, xor
  - Made out of transistors

- Wires
  - Lines over which signals are transmitted between gates


Signal
0
1
1
Wire
1
Gate

Representation of Logic Gates

- Symbol



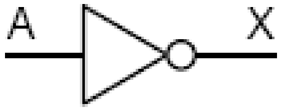
- Truth Table

A	X
0	1
1	0

- Algebraic Representation

$$\bar{A}$$

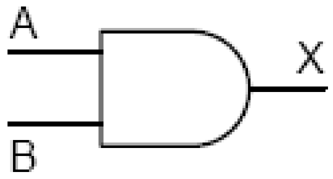
Not



A	X
0	1
1	0

- Inverts the input
- Algebraic representation: $\bar{A}$

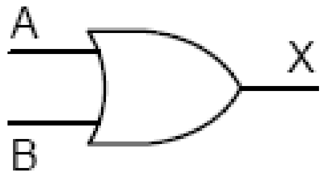
And



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

- Algebraic representation: AB or $A \cdot B$

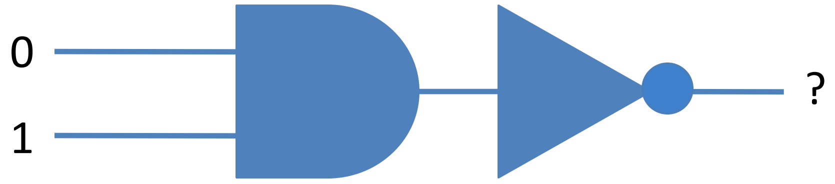
Or



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

- Algebraic representation: A+B

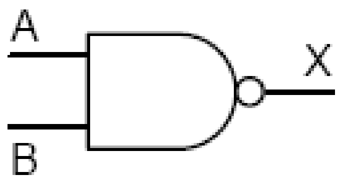
And and Not



A. 0

B. 1

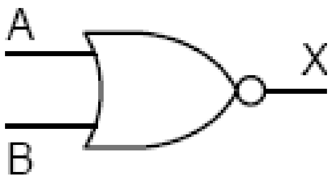
Nand



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

- Algebraic representation: $\overline{(A \cdot B)}$

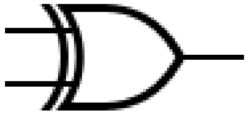
Nor



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

- Algebraic representation: $\overline{(A+B)}$

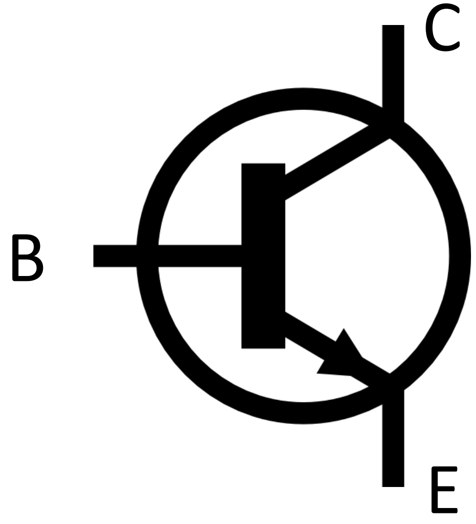
Xor

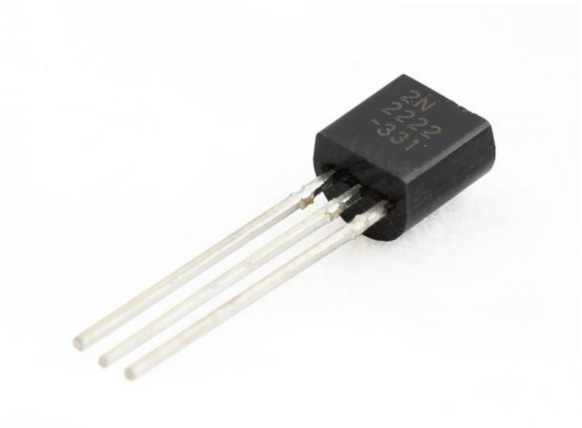


A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

- Algebraic representation: A^B or $A \oplus B$

Our Friend the Transistor

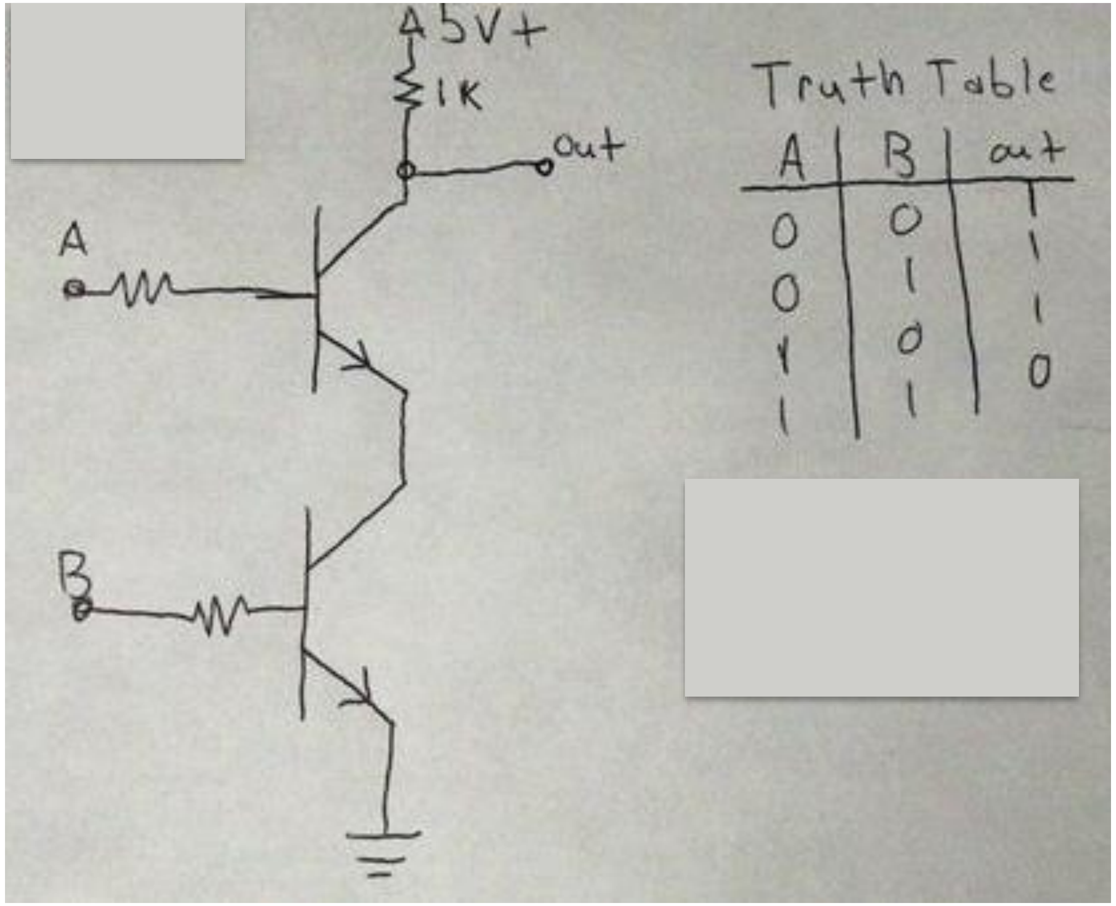




- The basic electronic component from which all gates are created; there are many types, this is an NPN transistor
- Applying a voltage to the base (B) allows current to flow from the collector (C) to the emitter (E)
- This creates an on/off switch

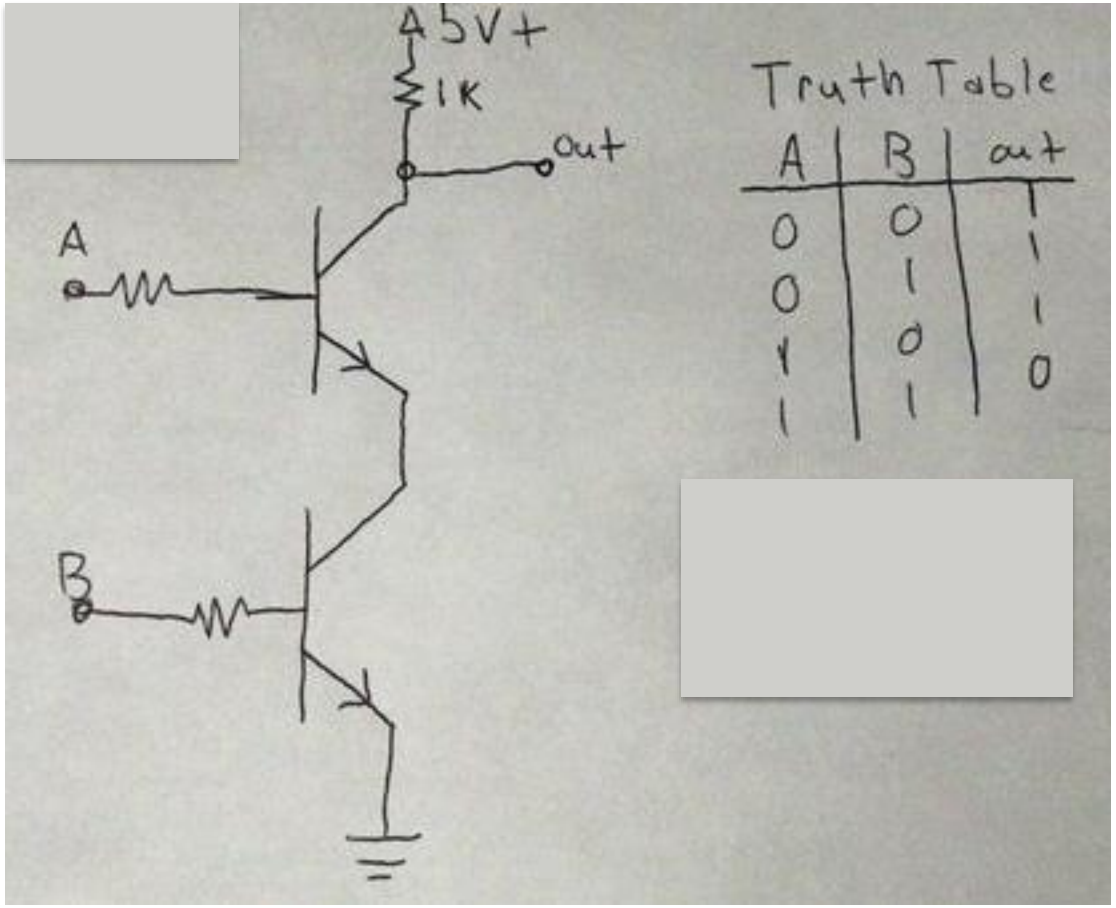
Building gates out of switches

- Two inputs labeled A and B
- One output labeled out
- When A or B is 1, the other two electrodes (collector and emitter) are connected
- When A and B are both 1, out is connected to ground (logic value 0)
- When either A or B is 0, out is not connected to ground and current can flow from 5V to out

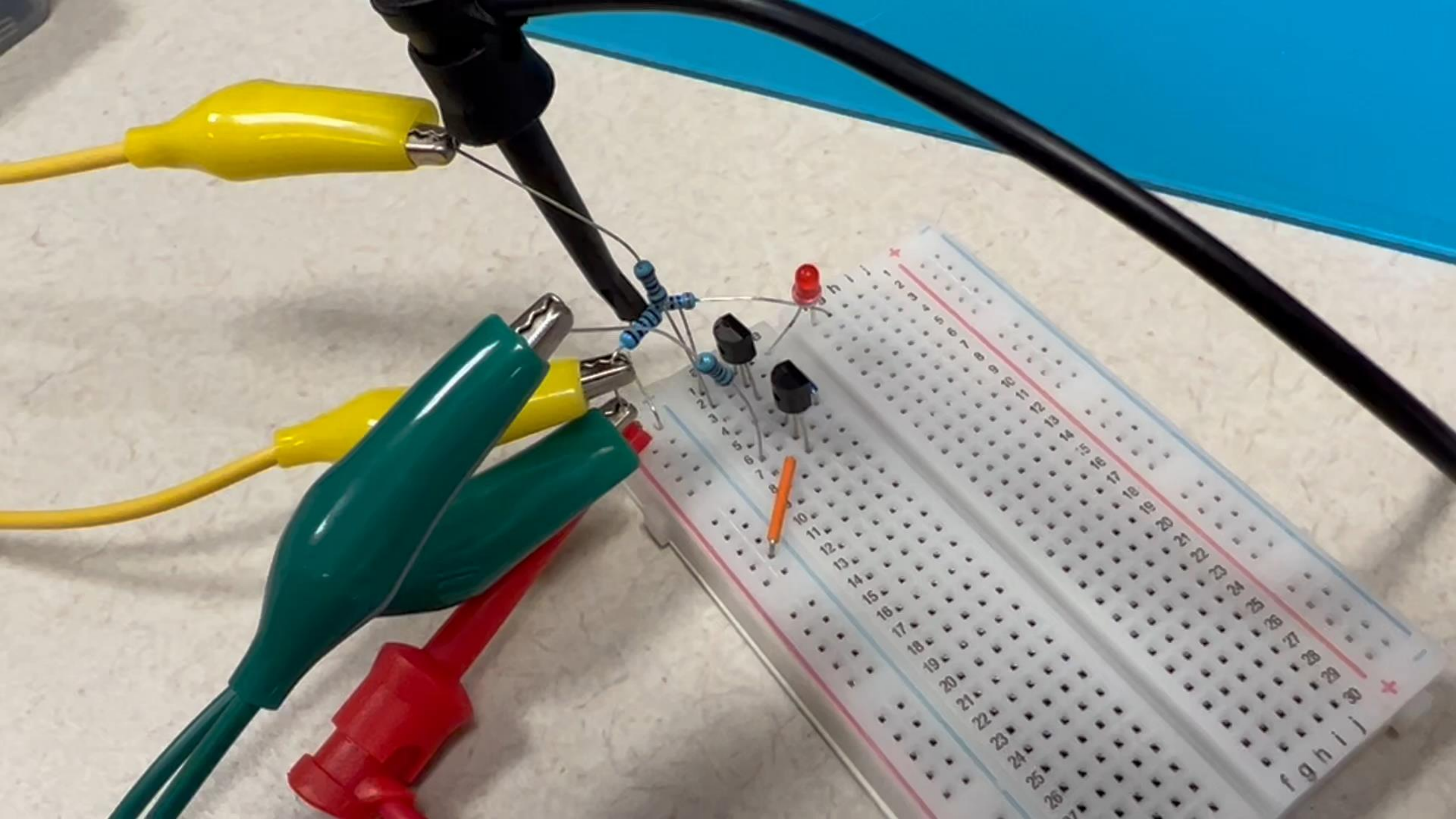


What Gate Does This Match?

- If both A and B are high voltage (logical 1), out will be low voltage (logical 0)
- Otherwise, out is high voltage



A. AND
B. OR
C. NAND
D. NOR



All Other Gates Can Be Created From NAND

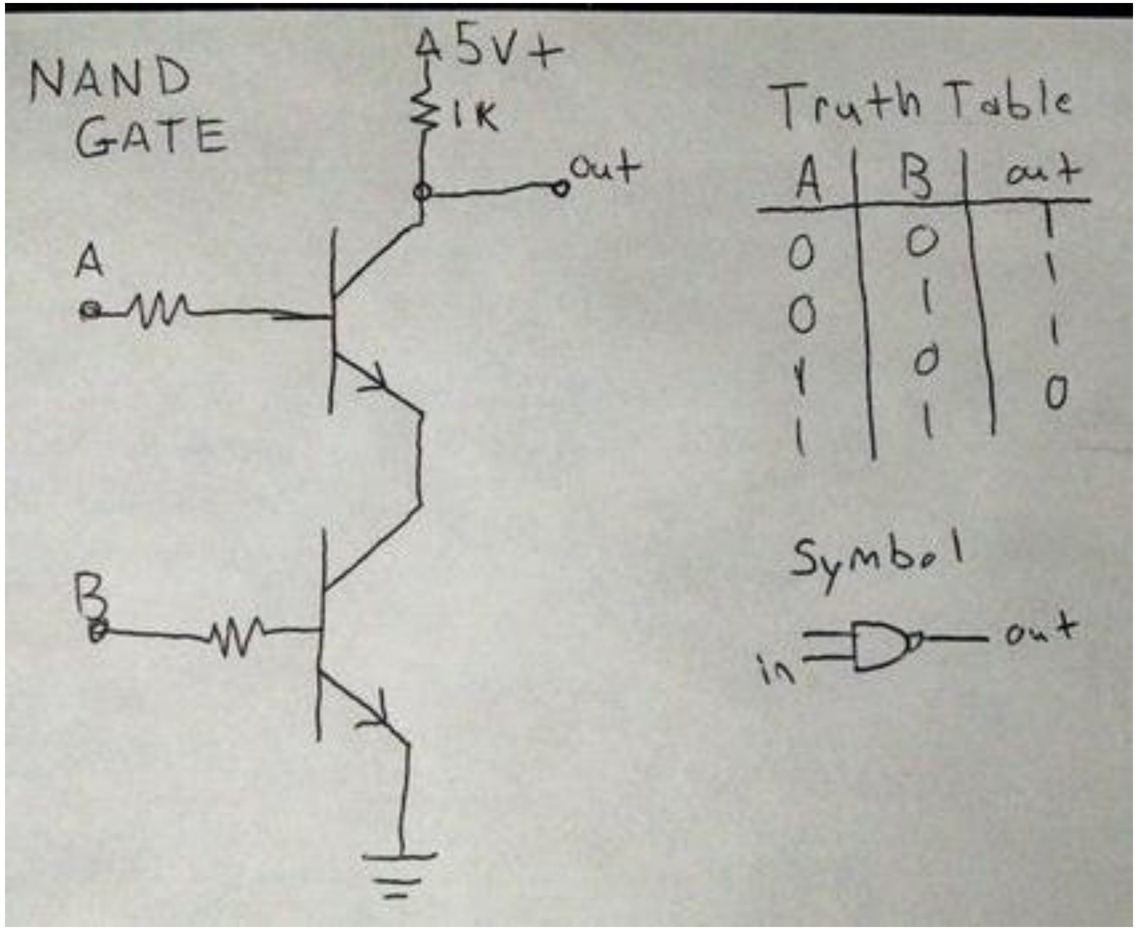
Not

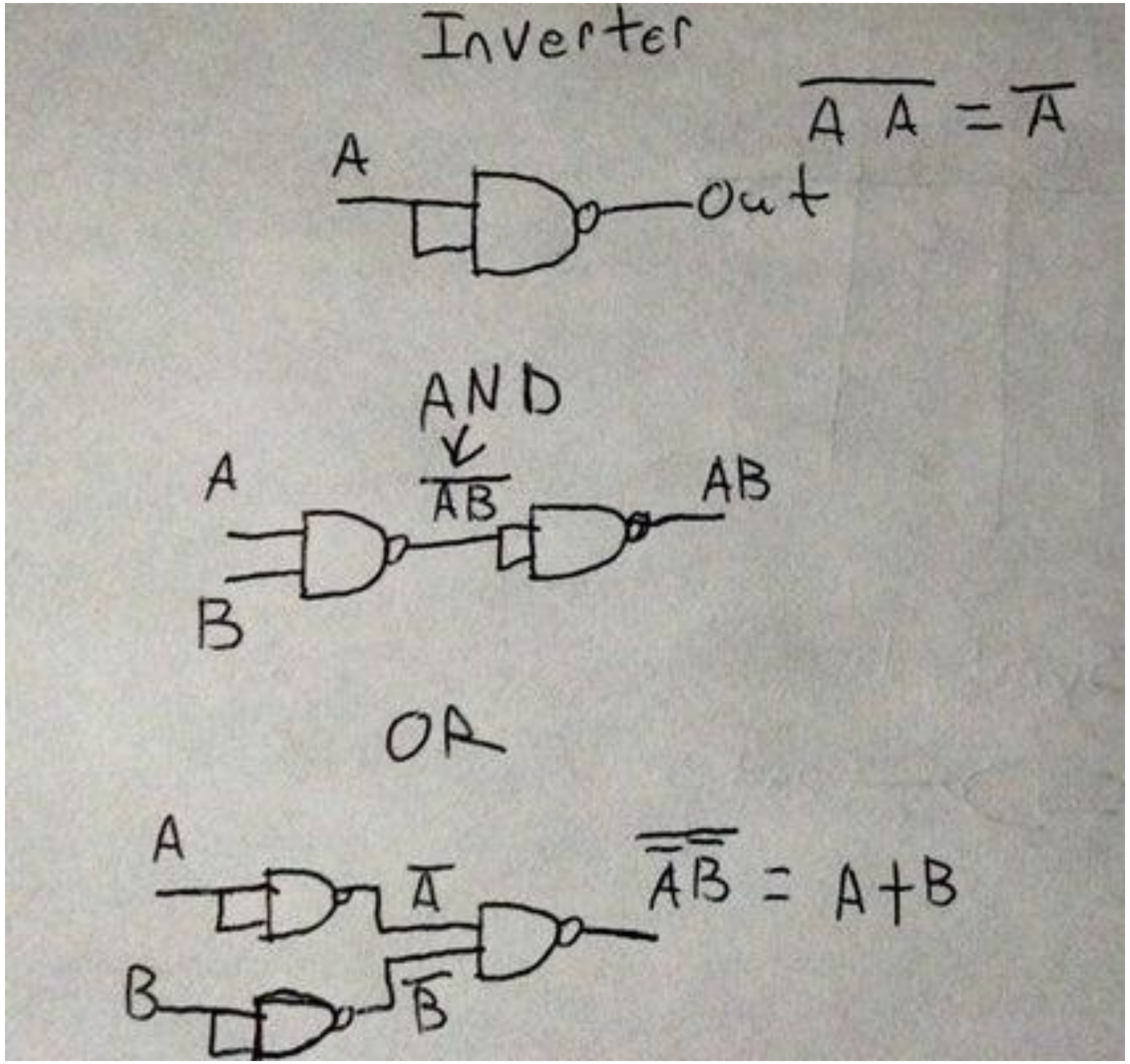
And

Which is equivalent to A OR B?

A. A NAND B
B. NOT (A NAND B)
C. (NOT A) NAND (NOT B)
D. NOT ((NOT A) NAND (NOT B))
E. None of the above

Putting them together





Images from: https://www.instructables.com/Build-a-NAND-gate-from-transistors/

All Gates Can Also Be Created from NOR

- NOR and NAND are universal gates
  - All gates can be created from them

- You will show this in Problem Set 5

Which column completes the truth table for

$$F = \overline{X} \cdot (Y + Z)\,?$$

X	Y	Z	A	B	C	D
0	0	0	0	0	1	1
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

Groups: Draw circuit diagram for

$$F = \overline{X} \cdot (Y + Z)$$

$$F = \overline{A} + (B(AC + \overline{AB}))$$

Truth Table

A	B	C	AC	$\overline{AB}$	$B(AC + \overline{AB})$	F
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Reading

- Next lecture: Boolean Algebra
  - 3.3

- Problem Set 4 due Friday

- Lab 3 due Monday