CSE2301
Digital Logic Design

Department of Computer Science and Engineering
University of Connecticut

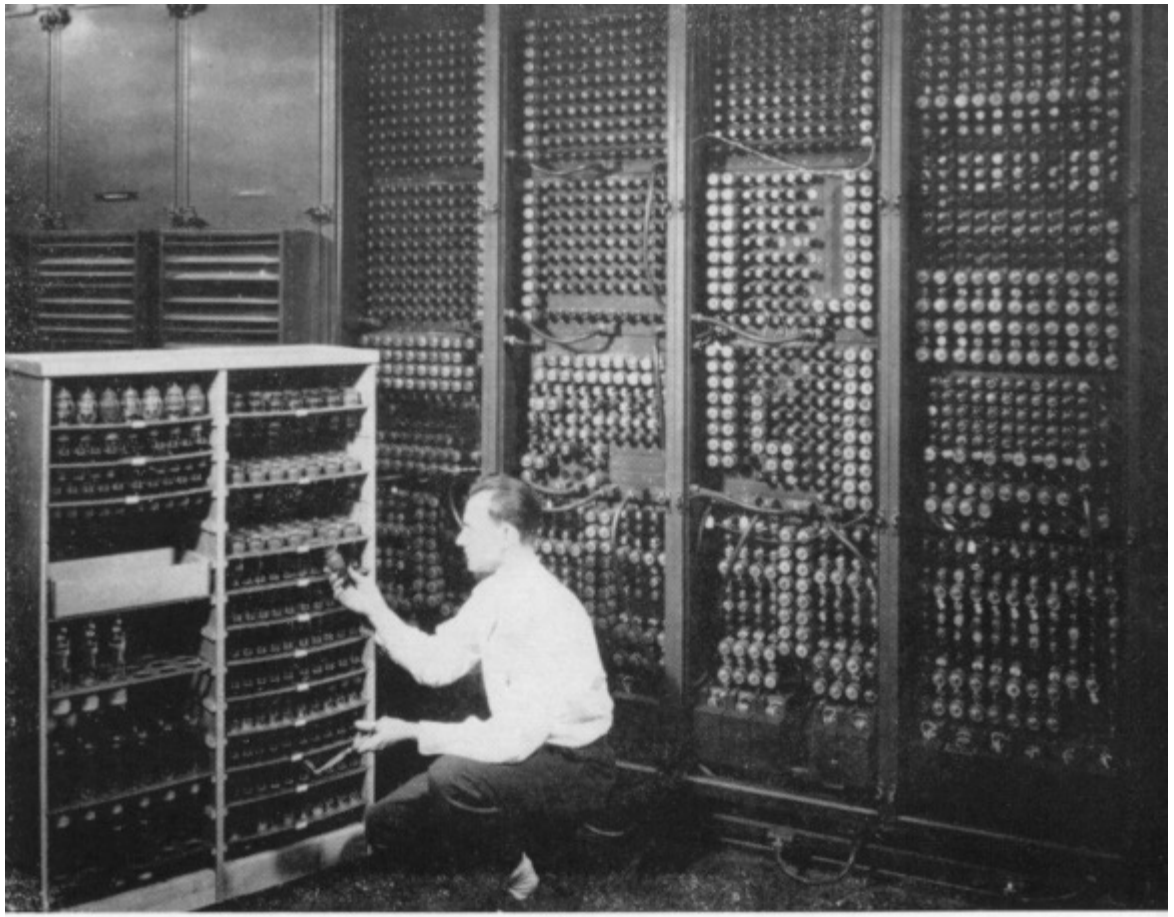
Why is this course important?

If you want to ….

- Design processor/motherboard/ Nintendo Wii
- Write operating system, device drivers
- Write software for embedded systems
- Understand how software interacts with hardware
- Do hardware security
- ……

This class is really important for you!

Digital Logic Design

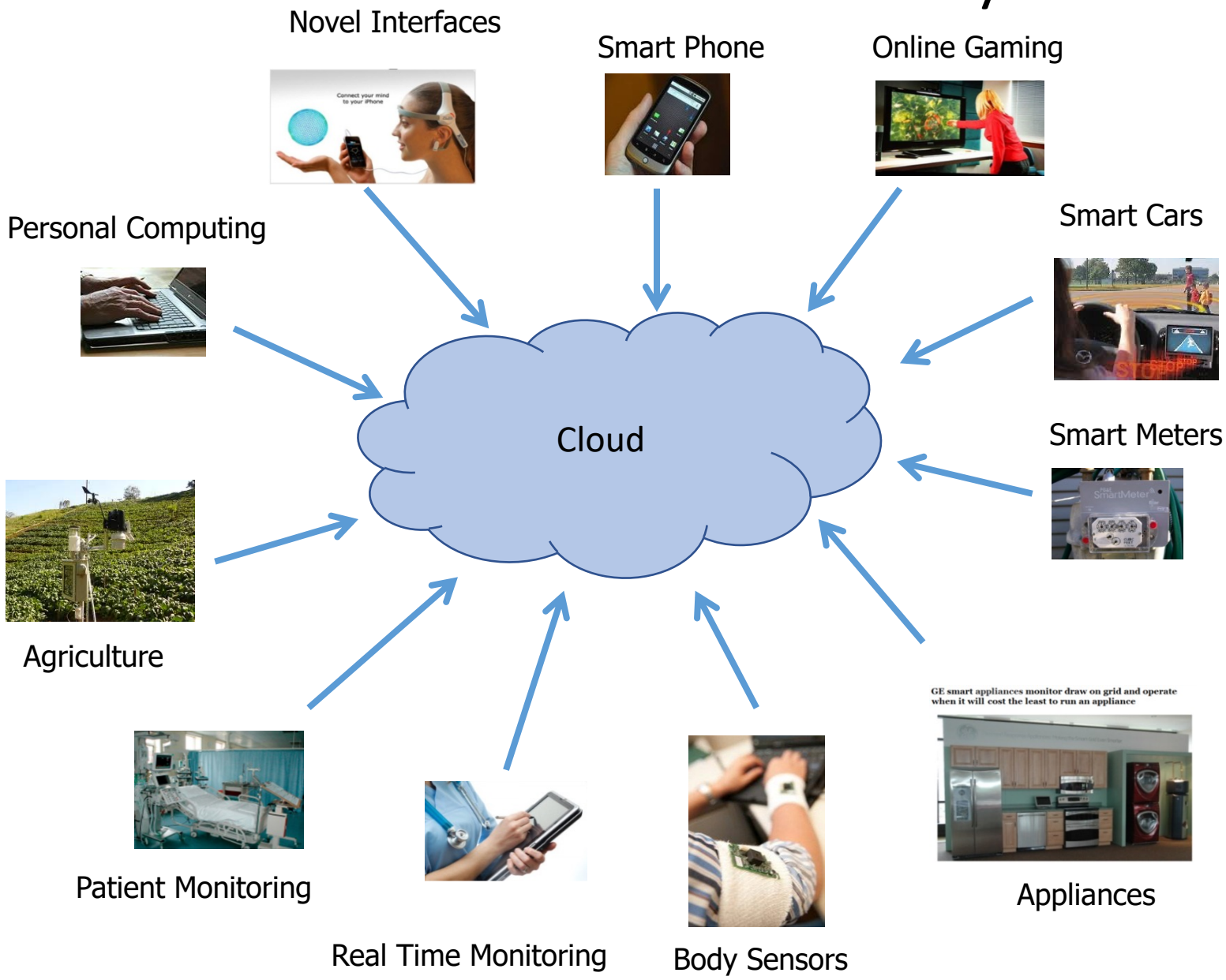


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Source:
http://www.computersciencelab.com/ComputerHistory/HistoryPt4.htm

John Bardeen, William Shockley and Walter Brattain at
Bell Labs, 1948.

Miniaturization was the key ……



Now, what are the main ideas?

- In our everyday world, we have many symbols
  - A,B,C,….0,1,2,3,…..$,@,……

- In digital world, there are only two symbols in our system
  - 0 and 1 (Welcome to the Binary system!)
  - You can only communicate using these two symbols!
  - You can use multiple of these symbols!

Now, wait a second ….

- How do you represent 0 or 1? There is no such thing in physical world!

Now, wait a second ….

- How do you represent 0 or 1? There is no such thing in physical world!

- You are right! These are just two numbers!
- In physical world, voltage and current are real!
- In digital world, we use 5V to represent 1 and 0v to represent 0 (Different voltage level may be used as well!)

American Standard Code for Inform CII)

Table 1.7
American Standard Code for Information Interchange (ASCII)

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	–	o	DEL

Some Example Problems

- Design a circuit with the following specification-
  - Takes two one-bit input from two users
  - (a) If either of the input is 1, output 1
  - (b) If both input are 1, output 1
  - (c) If both input are same, output 1

Some Example Problems

- Assume that you are designing a circuit that beeps if any of the door in your car is not properly locked. Assume that each door is equipped with a sensor that outputs 1 if the door is not properly locked. You have 4 doors in your car.

- It is all about playing with voltage and switches!
- We just need to come up with the logic behind this switching.

Hence, the term “Digital Logic” Design

Some Example Problems

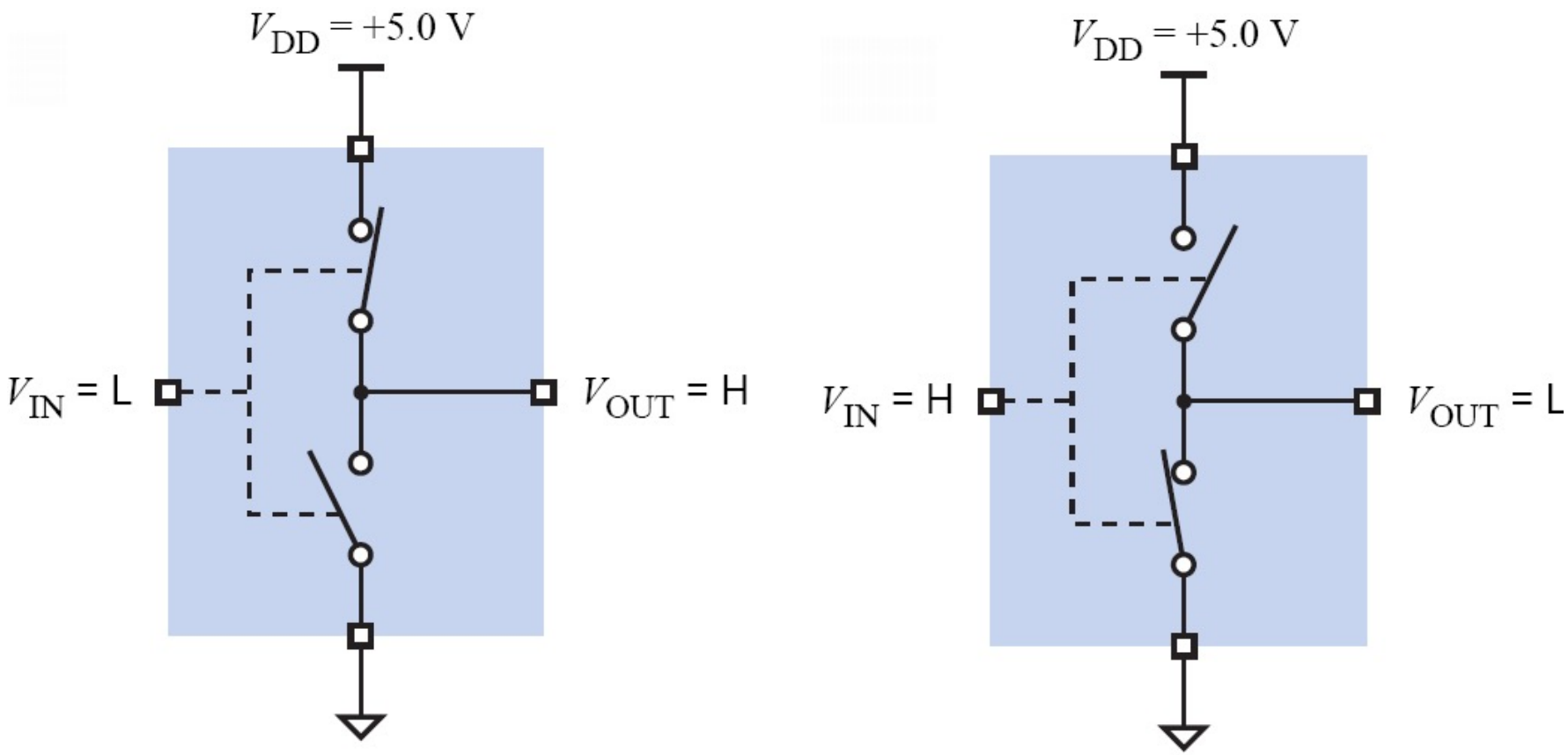
- Design a digital clock.
- How is this problem different than the first two examples?

- It is all about playing with voltage and switches!
- We just need to come up with the logic behind this switching.

Hence, the term “Digital Logic” Design

What do I mean by “switches”?

Inverter



That Car Example…

- Let us assume that we have only two doors in our car. If one of the two doors are open, set the alarm signal to 1.

That Car Example…

- 2 door

Door	Open	Close
X	1	0
Y	1	0

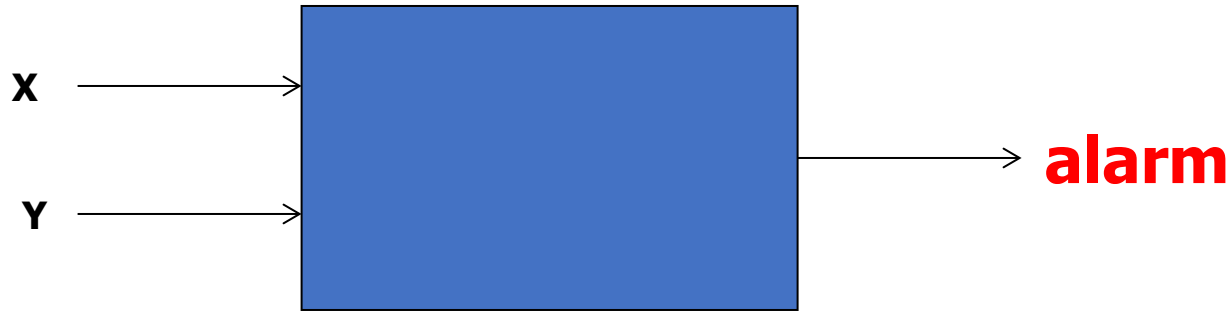
How would you express the logic?

That Car Example…

- If (X= =1 **or** Y = = 1) then **alarm** = 1

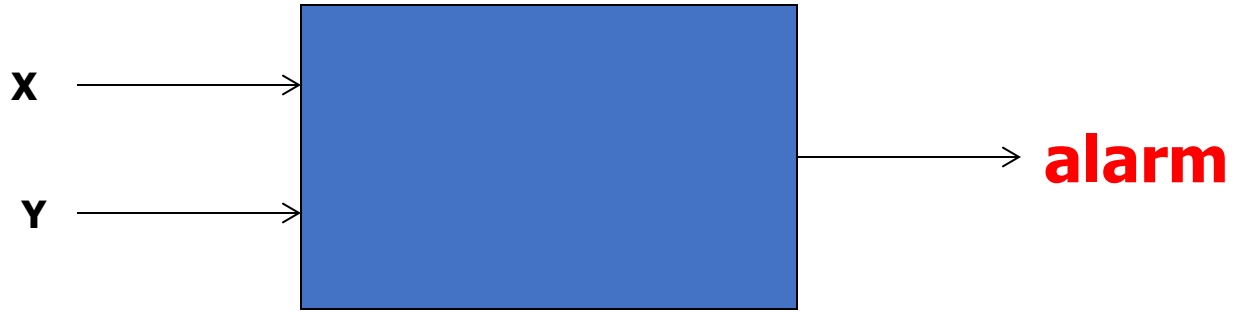
That Car Example…

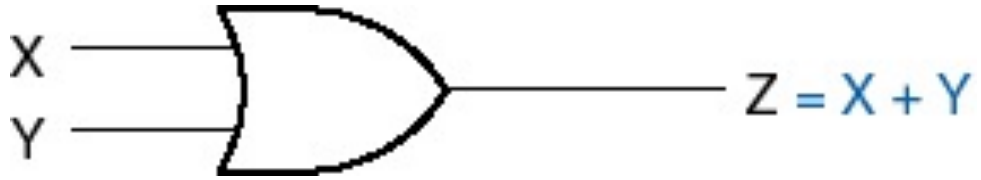
- If (X= =1 **or** Y = = 1) then **alarm** = 1



That Car Example…

- If (X= =1 **or** Y = = 1) then **alarm** = 1



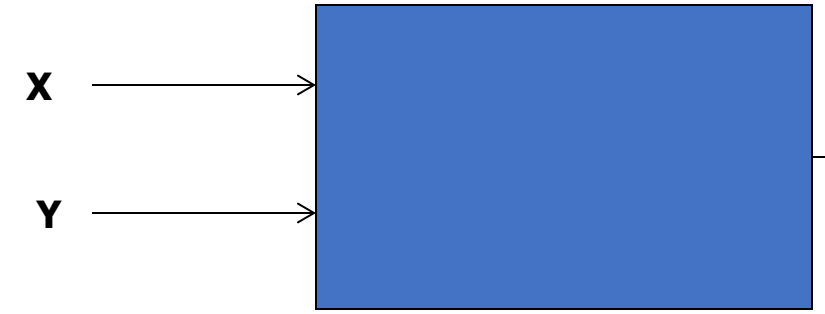


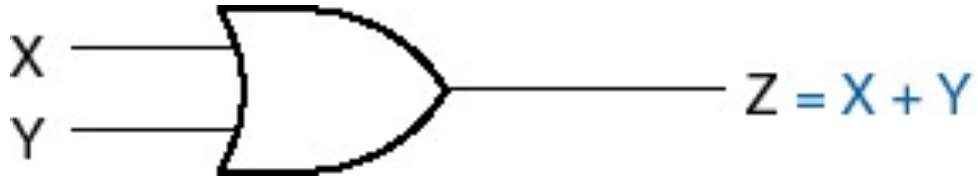
OR Gate

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

That Car Example…

- If (X= =1 **or** Y = = 1) then **alarm** = 1





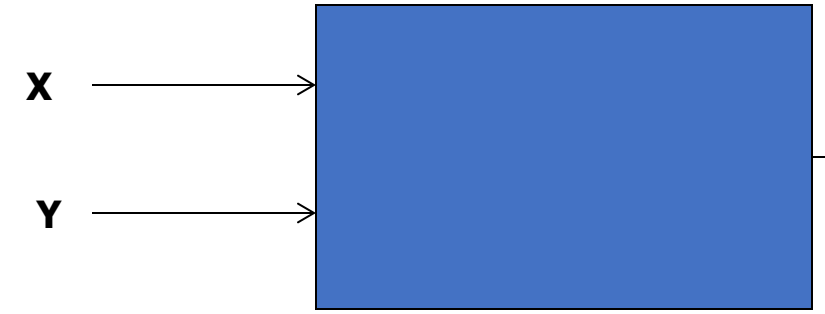
OR Gate

Truth Table

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

That Car Example…

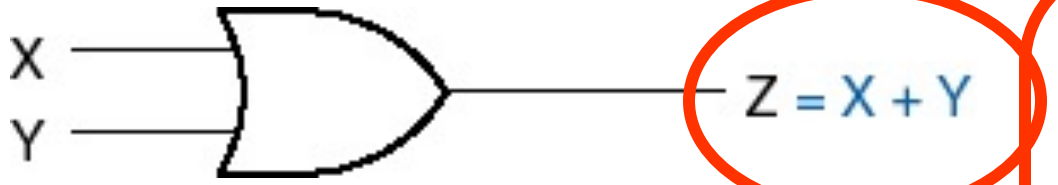
- If (X= =1 **or** Y = = 1) then **alarm** = 1



alarm

Truth Table

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1



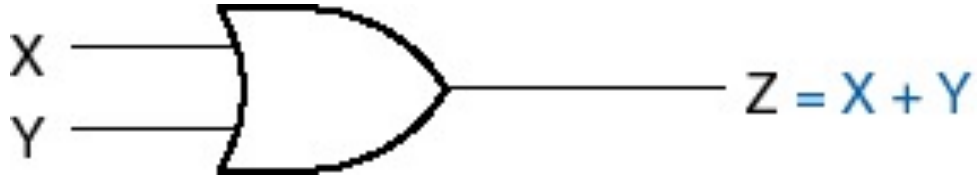
OR Gate

OR is denoted using the "+" symbol

That Car Example…

- If (X= =1 **or** Y = = 1) then **alarm** = 1

Congratulations!
I just introduced you to your first logic gate!



OR Gate

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

Now, a changed Car Example…

- **If and only if** both of the two doors are open, set the alarm signal to 1.

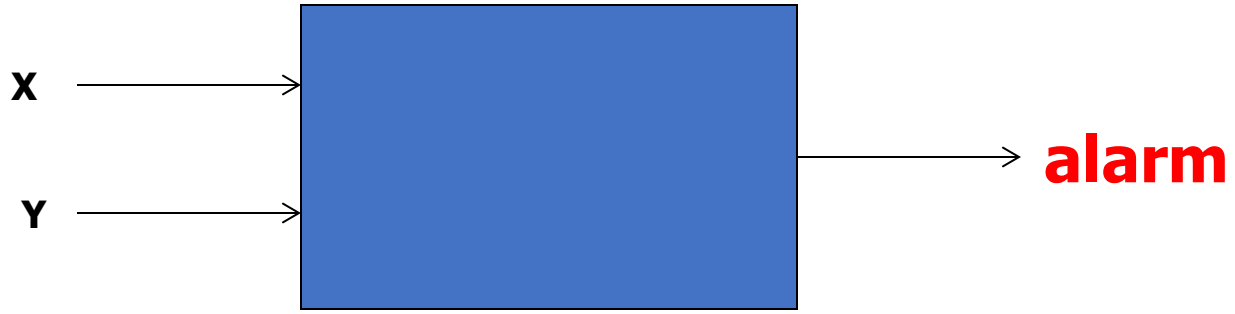
How would you express the logic?

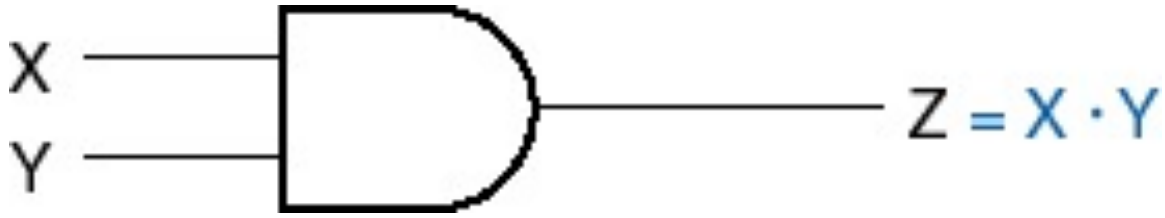
That Car Example…

- If (X= =1 **and** Y = = 1) then **alarm** = 1

That Car Example…

- If (X= =1 **and** Y = = 1) then **alarm** = 1



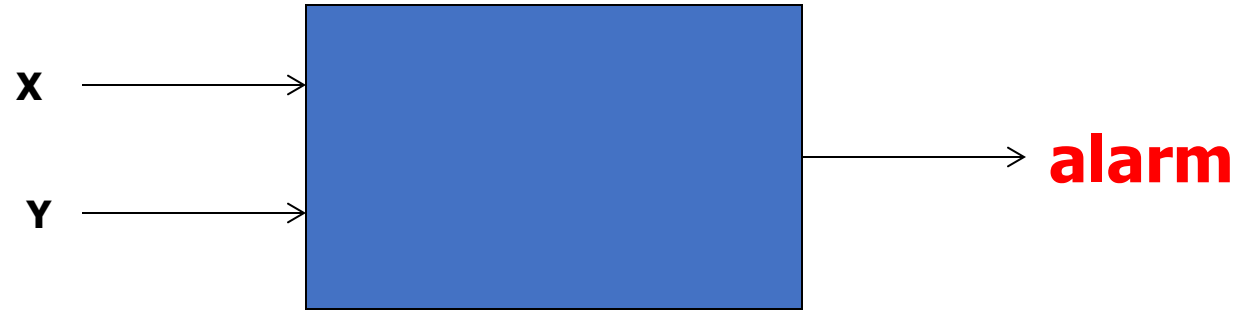


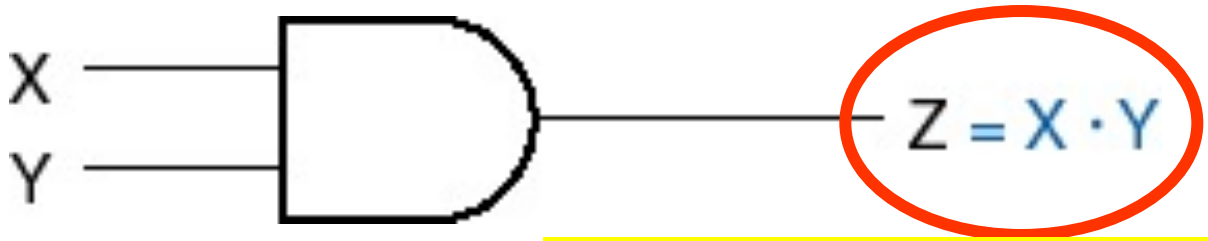
AND Gate

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

That Car Example…

- If (X= =1 **and** Y = = 1) then **alarm** = 1





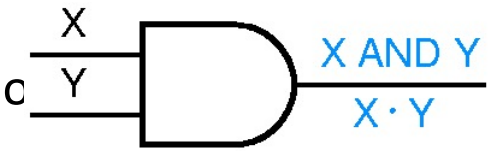
X	Y	X AND Y
0	0	0
1	0	0
1	1	1

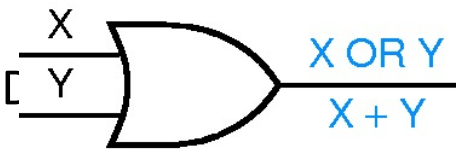
AND is denoted using the " . " symbol

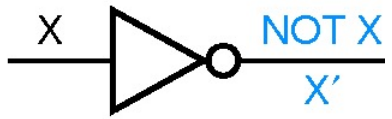
AND Gate

Gates

- Gates are basic digital devices
- A gate takes one or more inputs and produces an output
  - Can be considered as a function
  - Inputs are either 0 or 1

Example o 

: AND 



X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	NOT X
0	1
1	0

Boolean operators

- Complement: X′ (opposite of X)
- AND: X · Y
- OR: X + Y

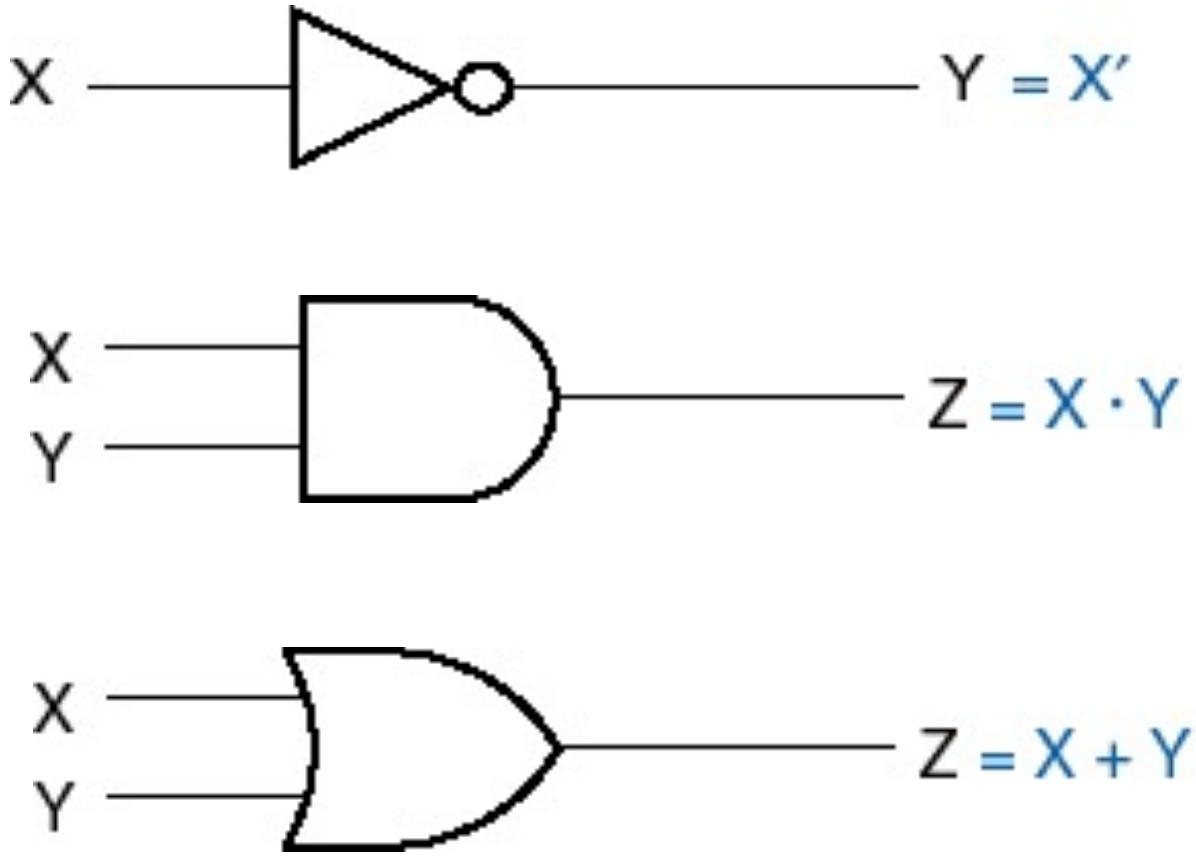
Function described with truth table.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	NOT X
0	1
1	0

Logic symbols of NOT, AND, and OR



Now, another changed Car Example…

- **If and only if** both of the two doors are open (or close), set the alarm signal to 1.

How would you express the logic?

That Car Example…

If ((X= =1 **and** Y = = 1) or (X= = 0 **and** Y = = 0))

then **alarm** = 1

Can you generate the truth table for this?

That Car Example…

If ((X= =1 **and** Y = = 1) or (X= = 0 **and** Y = = 0))

then **alarm** = 1

X	*Y*	Alarm
0	0	1
0	1	0
1	0	0
1	1	1

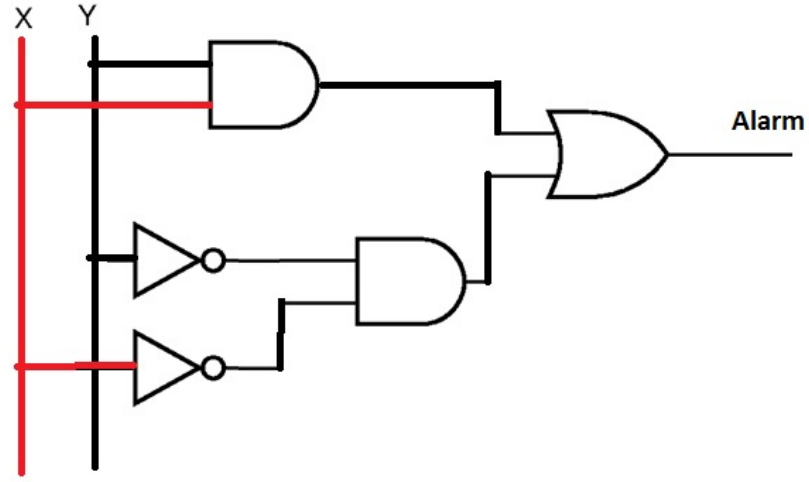
Can you implement this logic using the gates we have seen so far?

That Car Example…

If ((X= =1 **and** Y = = 1) or (X= = 0 **and** Y = = 0))

then **alarm** = 1

X	*Y*	Alarm
0	0	1
0	1	0
1	0	0
1	1	1



Can you implement this logic using the gates we have seen so far?

- So, the idea is to –
  - First, express the circuit design problem in terms of input binary variables
  - Second, identify the output binary variables
  - Third, build a truth table where you need to assign a value to the output variable for each combination of input variables
  - Finally, express the output variable(s) in terms of input variables
  - **We will see how to do the last step soon!**