

# Session 9 - Microcontroller logic gates

🕒 Created	@February 22, 2022 2:14 PM
▼ Class	Embedded Systems
▼ Type	
🔗 Materials	
☑ Reviewed	<input type="checkbox"/>

## Number System

- Way of representing numbers
- Types:

	Base	Number Range
Decimal	10	0-9
Binary	2	0,1
Octal	8	0-7
Hexadecimal	16	0-9, A to F

128 64 32 16 8 4 2 1  
 $2^7$   $2^6$   $2^5$   $2^4$   $2^3$   $2^2$   $2^1$   $2^0$



1 1 1 1 1 1 1 1

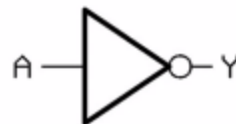
$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

1 0 0 1 1 0 1 1  
 $128 + 16 + 8 + 2 + 1 = 155$

0 0 0 0 1 0 1 0  
 $8 + 2 = 10$

## NOT Gate (Inverter)

- Purpose: To perform negation on all bits
- Symbol: ~
- Logic Diagram:



- Truth Table:

Input	Output
0	1
1	0

## AND Gate

- Purpose: To RESET particular bit/s
- Symbol: &
- Logic Diagram:



- Truth Table:

A	B	Y = A&B
0	0	0
0	1	0
1	0	0
1	1	1

---

## OR Gate

- Purpose: To SET particular bit/s
- Symbol: |
- Logic Diagram:



- Truth Table:

A	B	Y = A   B
0	0	0
0	1	1
1	0	1
1	1	1

## Masking:

## What are Ports?

ATmega 2560 is a 100 pin micro-controller

- 86 pins can be used as Input/Output pins
- Pins are grouped together and are called as **Ports**

ATmega 2560 has ten 8-bit Ports

- Port x; x = A to F and H, J, K, L

ATmega 2560 has one 6-bit Port

- Port G

Each Port has three associated registers with it:

- DDRx x = A to H and J, K, L
- PORTx x = A to H and J, K, L
- PINx x = A to H and J, K, L

these are  
used  
for accessing  
the port pin

## Need for masking

- AVR is not bit addressable. It is only bit accessible.
- No 'address' to a specific bit.
- Sometimes we need to change the state of one or more pins of the **port** keeping the rest of the pins unchanged.

## NOT operator

① Purpose: To perform negation on all bits.

② Symbol:  $\sim$

③ Example:

A =

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

$\sim A =$

B7	B6	B5	B4	B3	B2	B1	B0
0	1	1	1	1	1	0	0

## Shift Operators

- ❶ Purpose: To shift all bits by specified bit position.
- ❷ Types: Left Shift and Right Shift
- ❸ Symbol: Left shift ( $\ll$ ) and right shift ( $\gg$ )
- ❹ Example:

$$A =$$

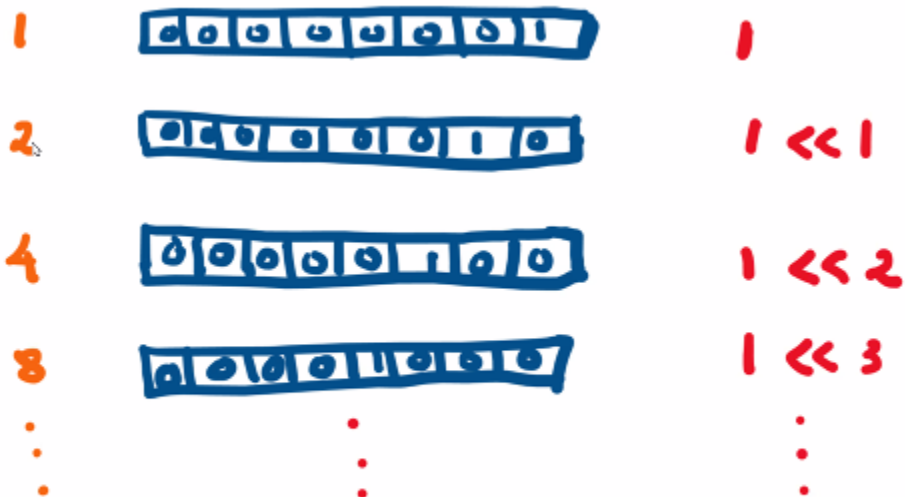
B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

$$A \ll 2 =$$

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	1	1	0	0

$$A \gg 2 =$$

B7	B6	B5	B4	B3	B2	B1	B0
0	0	1	0	0	0	0	0



① Purpose: To SET particular bit/s.

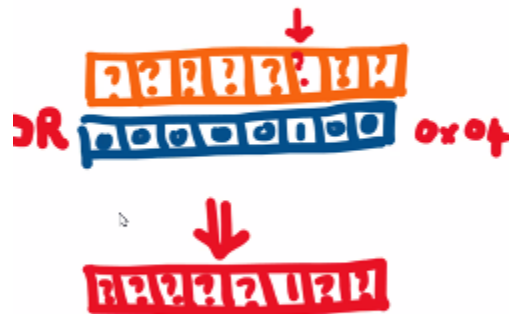
② Symbol: |

③ Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

### Example

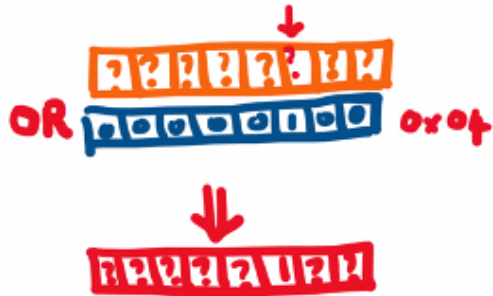
Setting B2



- Whichever bit you want to set only there you are putting a 1, other than that the others are zero, as the value will remain the same

## Example

Setting B2



B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

OR

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	1	0	0

Output same as Expected output:

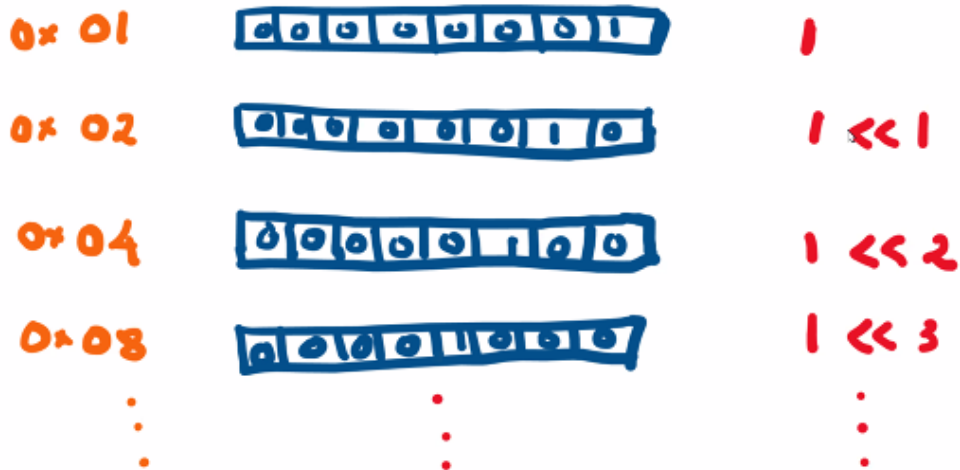
B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

1 register\_name = register\_name | 0x04;

2 register\_name |= 0x04;

www.o-pentra.org Finished ATmega2560 Robotics Research Platform

the simple name to check the above equation





## Example of Masking with Shift Operator

Sets B2 →

Using shift operator

For setting multiple bits →

① `register_name |= 0x04;`

② 0x04 can also be written as `1<<2`

③ In general, statement can be written as:

`Register_name |= (1 << pin_no)`

④ For setting multiple bits at once the statement can be written as:

`Register_name |= ((1 << pin_no1) | (1 << pin_no2))`

To set pin/s:

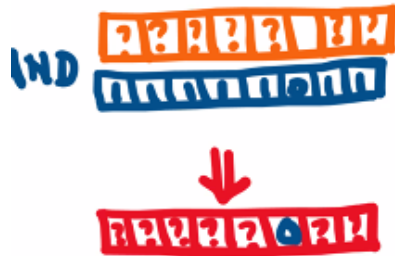
In general, statement can be written as:

Register name  $j = (1 \ll \text{pin no})$

For setting multiple bits at once the statement can be written as:

Register name  $j = ((1 \ll \text{pin no1}) | (1 \ll \text{pin no2}))$

## Example



Example

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

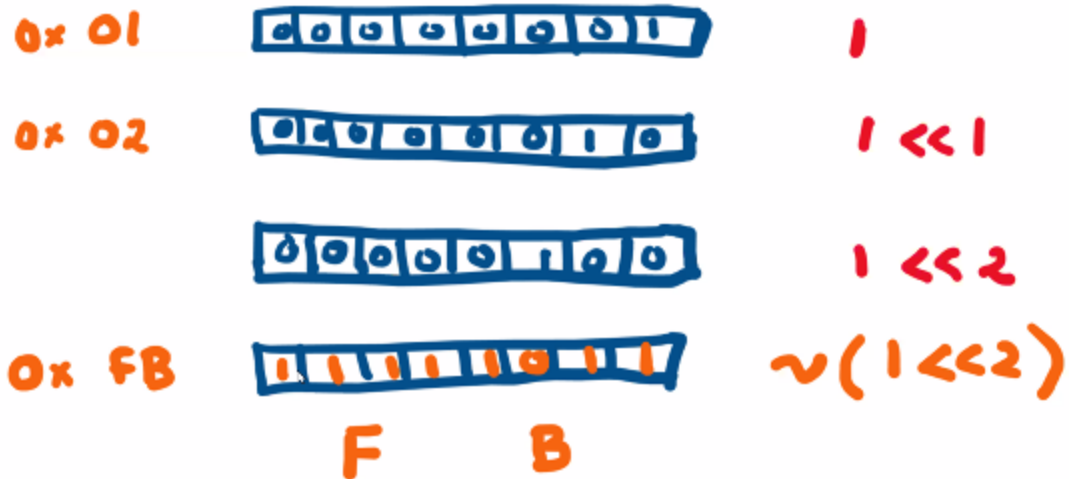
AND

B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	1	1	0	1	1

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

- 1 register\_name = register\_name & 0xFB;
- 2 register\_name &= 0xFB;

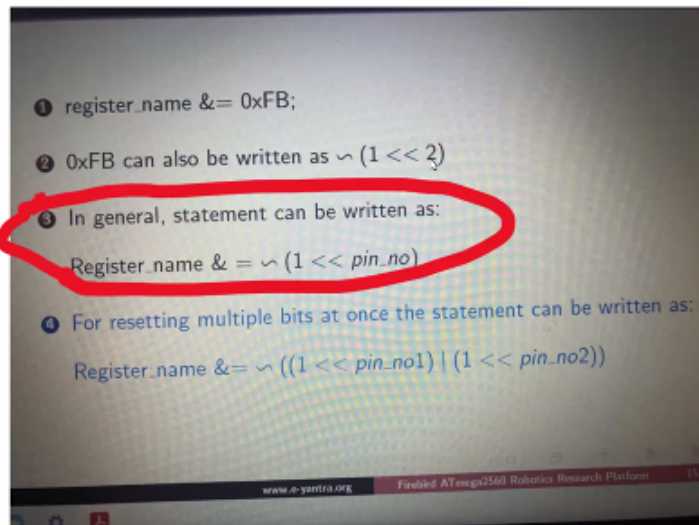


## Example of Masking with Shift Operator

Resetting  $B_2 \rightarrow$

using shift operator  $\rightarrow$

For resetting multiple bits  $\rightarrow$



To Reset pin/s:

In general, statement can be written as:

**Register name &= ~ (1 << pin no)**

For resetting multiple bits at once the statement can be written as:

**Register name &= ~ ((1 << pin no1) | (1 << pin no2))**

## EOR Operator

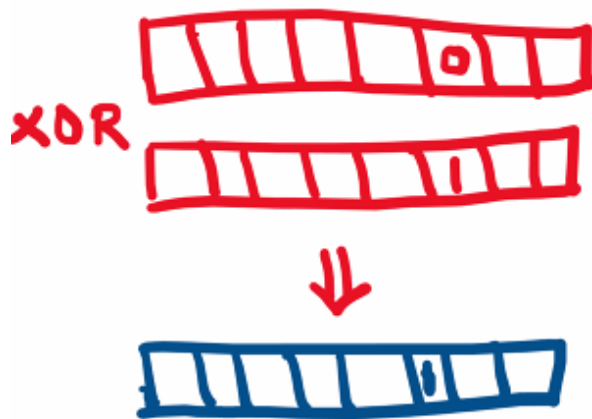
Purpose: To TOGGLE particular bit

Symbol: ^

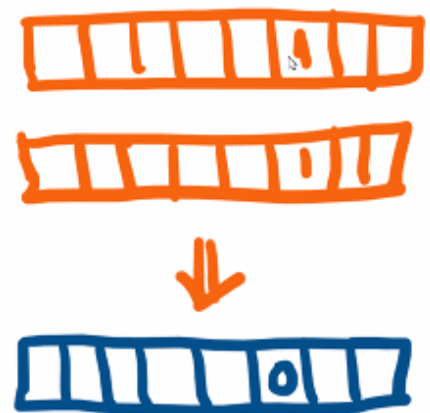
Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Case 1 :



Case 2 :



- For one bit : Register name ^= (1<<pin no)
-