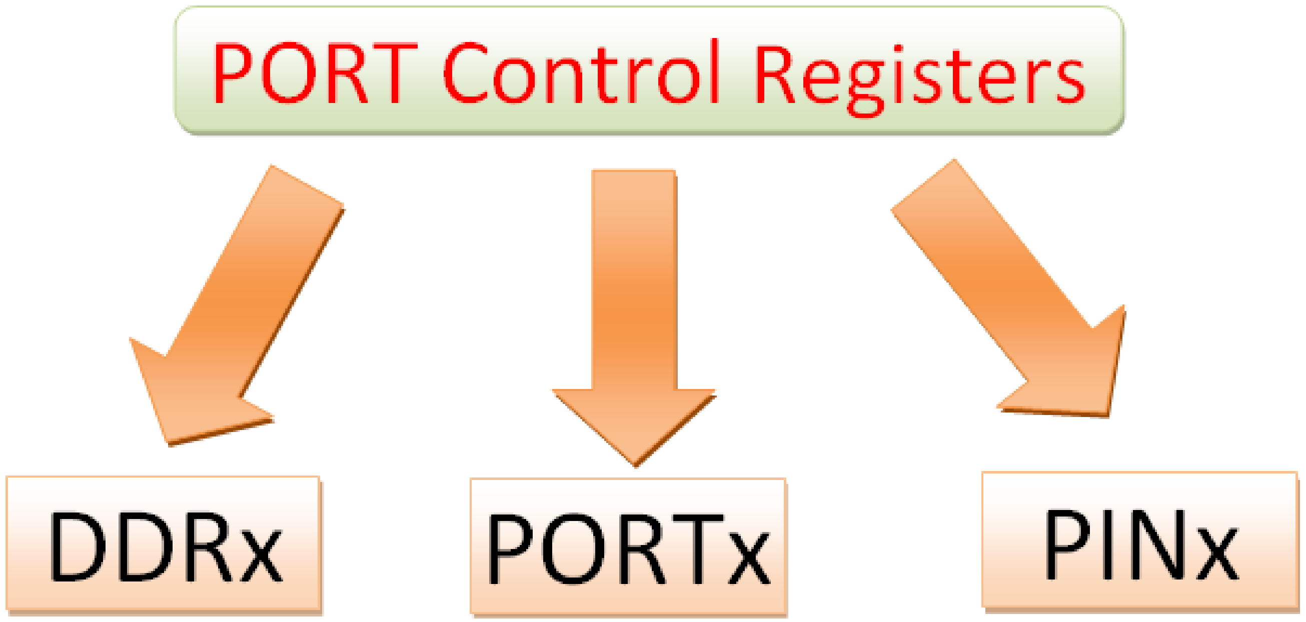
AVR Programming

AVR Registers



Controls the I/O status of the Port pins

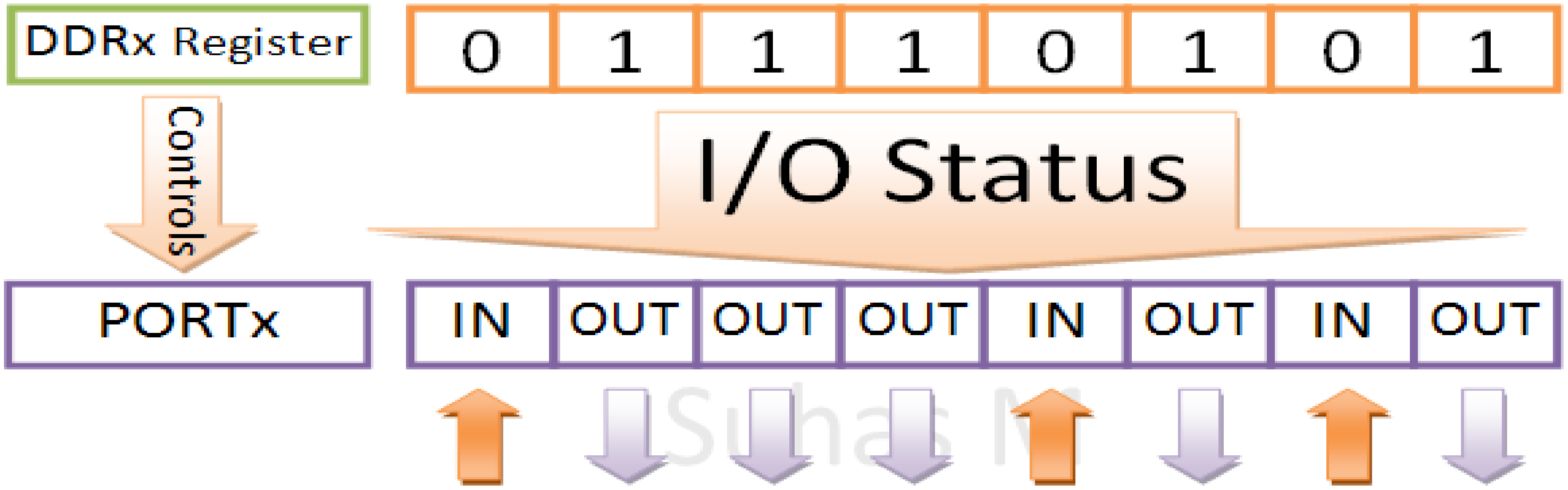
Controls the output status of pins.

Reads the I

of pins.

Data Direction Register

The DDRx register controls if a pin is in Input mode or Output mode



Data Direction Register

DDRA = 0xaa;

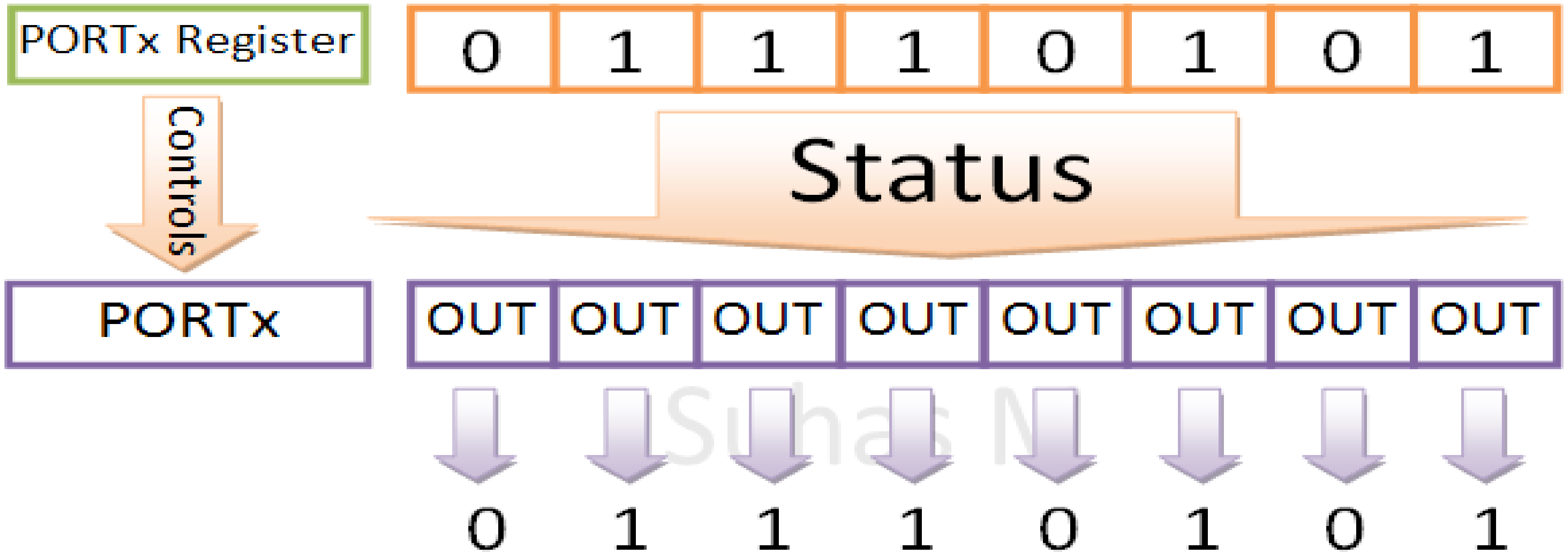
= 0b10101010

Will define pins 1,3,5,7 of Port A as output since bits 1,3,5,7 are 1 &

the rest of the pins are input as those bits are 0.

PORTx Instruction

If all the pins in PORTx are configured as output:



PORTx Instruction

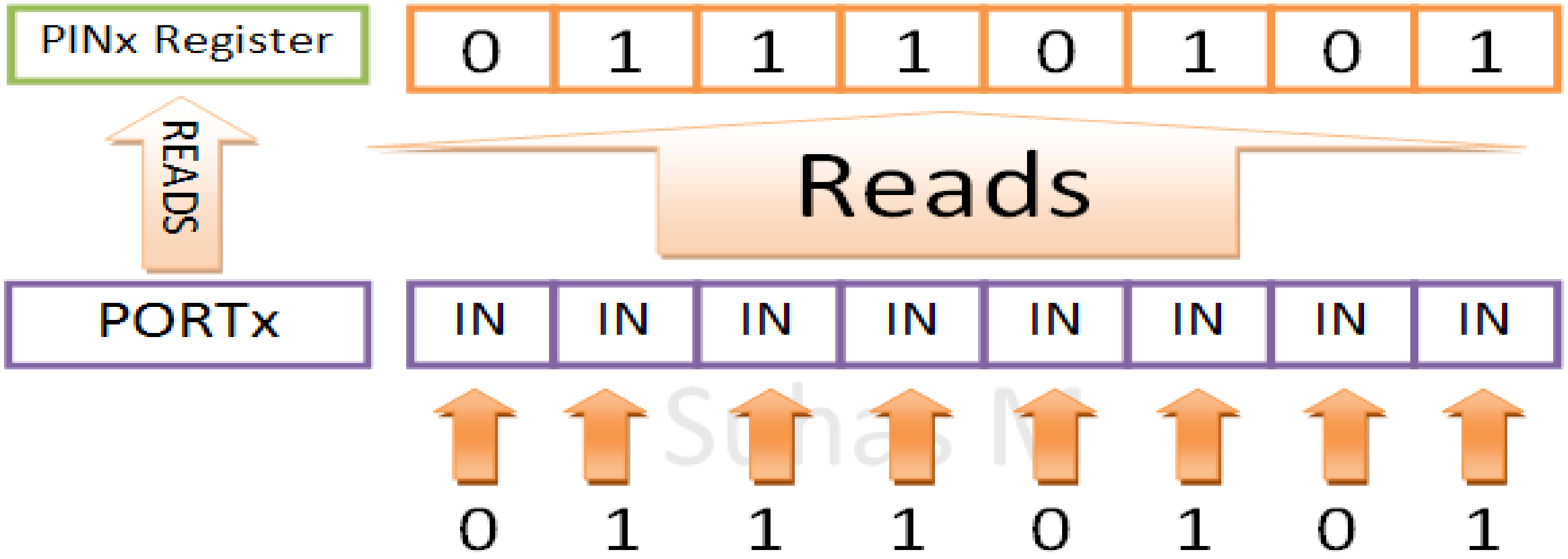
```
DDRA = 0xFF;

PORTA=0xAA;
```

DDRA defines all the pins as output.

PORTA will make bits 1,3,5,7 high i.e. 1. and the rest low i.e. 0.

PINx Instruction

PINx register reads the value of input pins



Setting and clearing bits

- To **set** a given bit (say bit 5) of a given register (say reg):
  Reg |=0b0010000
  Alternatively , if the bit has a name (say Enable) then
  Reg |=(1<<Enable)
  does the same.

where << is the **Left Shift Operator.**

- Similarly for clearing the bit
  Reg&=~(1<<Enable)
  is used.

Contd...

An easier way of doing this is using the following functions:

- **sbi** : set a bit

```
        void sbi(u08 register , u08 bit) ;
```

- **cbi** : clear a bit

```
        void cbi(u08 register , u08 bit);
```

Other useful functions

- **bit_is_set**(PINx, y) returns a '1' if the specified bit is set(logical 1) and zero otherwise .
- Similarly , **bit_is_clear**(PINx , y) can be used to check if the bit is clear.
- **_delay_ms(time)** gives a delay of time milliseconds.

AVR Headers

- avr/io.h : contains appropriate IO definitions for the device.
- stdlib.h : declares some basic C macros and functions plus some AVR-specific extensions.
- compat/deprecated.h : contains several items that used to be available in previous versions of this library, but have eventually been deprecated over time.
- util/delay.h: Contains convenience functions for busy/wait delay loops.

BLINK!!

Code :

```
#include<avr/io.h>
#include<stdlib.h>
#include <compat/deprecated.h>
#include <util/delay.h>
```

Header Files

```
void port_init(void)
{  PORTA = oxoo;     //input port
   DDRA  = oxoo;                     //no pull-up
   PORTB = oxoo;     //input port
   DDRB  = oxoo;                     //no pull-up
   PORTC = oxoo;
   DDRC  = oxFF;    // PORTC all output
   PORTD = oxoo;     //input port
   DDRD  = oxoo;                     //no pull-up
}
```

Port Initialization

Code :

```
void init_devices(void)
{

 port_init();

}
```

Device Initialization

MAIN FUNCTION

Main function:

```
void main(void)
{
   init_devices();
    while(1)
          {
               PORTC=0xFF;          // all LEDs glow initially

               delay(0);            //you can add delay in seconds here
               _delay_ms(50);       //delay in  milli seconds
               _delay_us(0);                        //delay in micro seconds

               PORTC=0x00;          //all LEDs are off after the set time delay
               delay(0);
               _delay_ms(50);
               _delay_us(0);
          }
}
```