EE-222: Microprocessor Systems

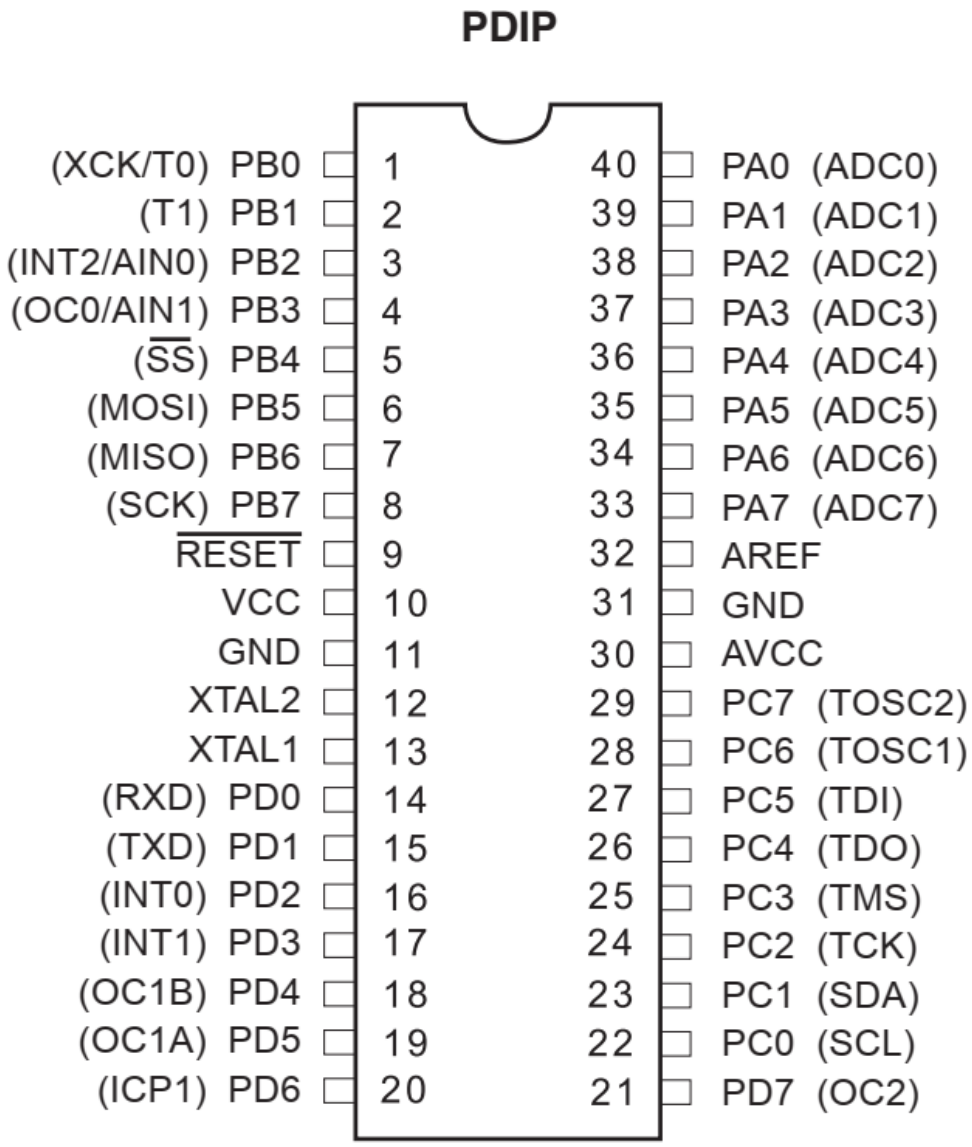
AVR Microcontroller:
I/O Port Programming

Instructor: Dr. Arbab Latif

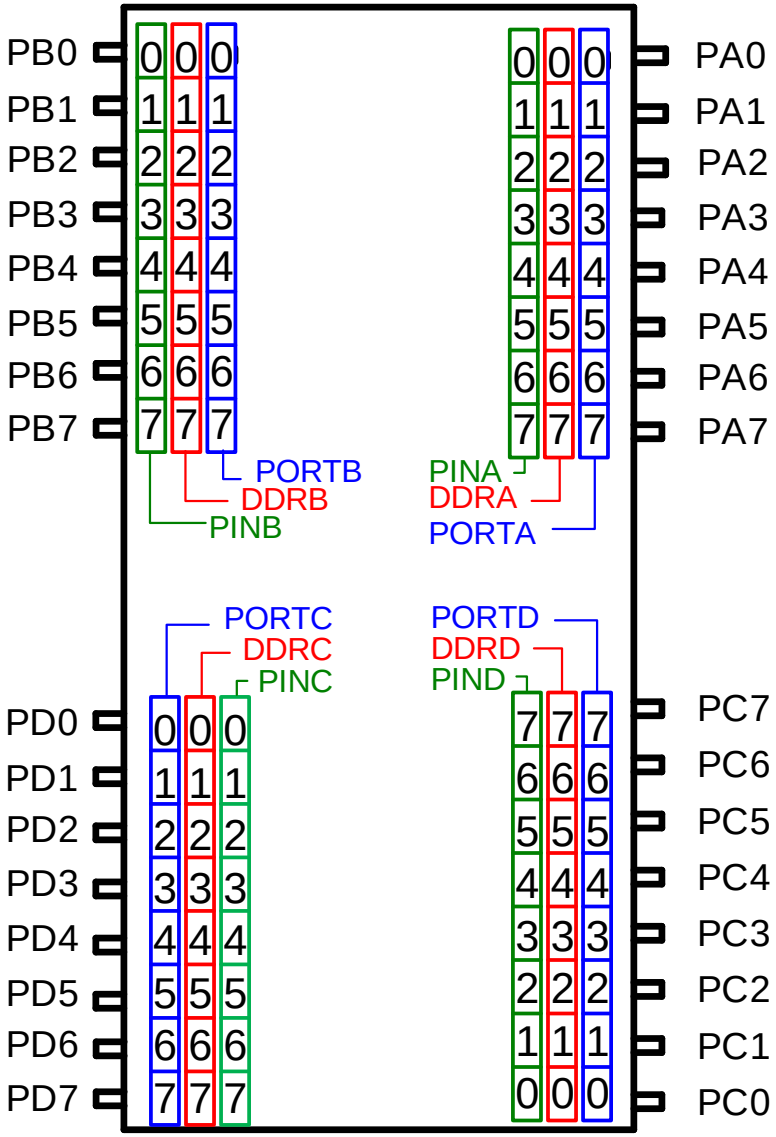
NUST
SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

I/O Port Programming

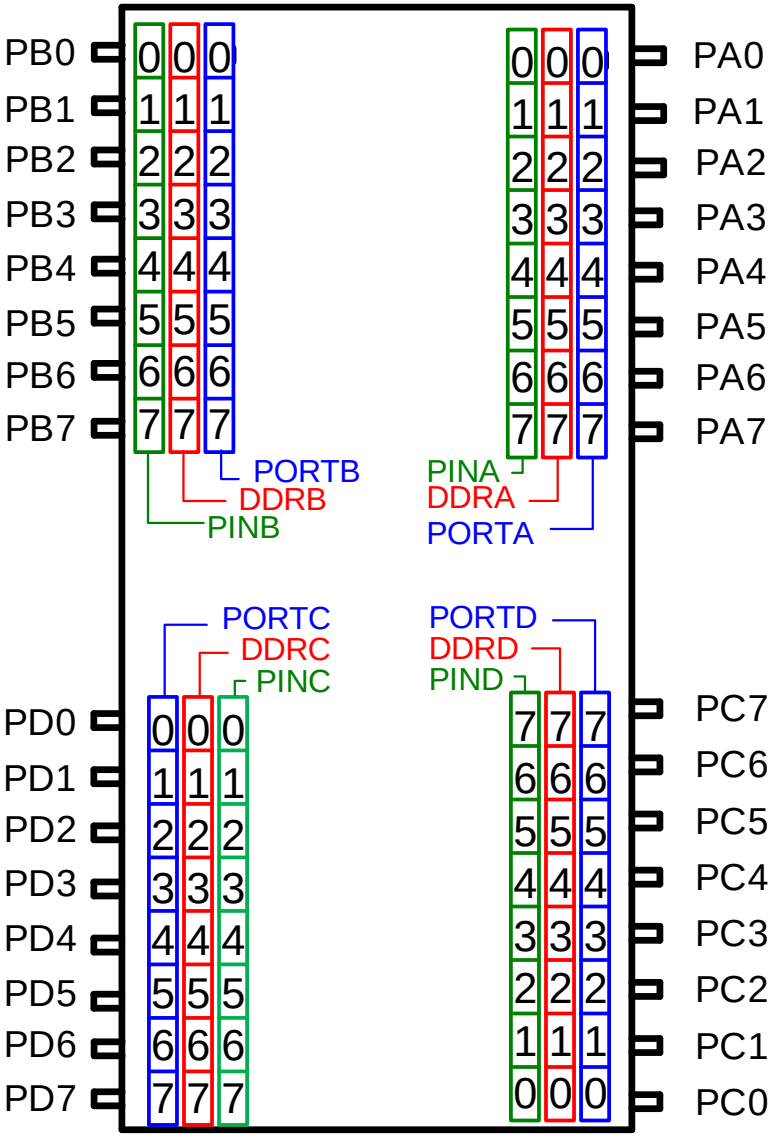
AVR I/O Ports

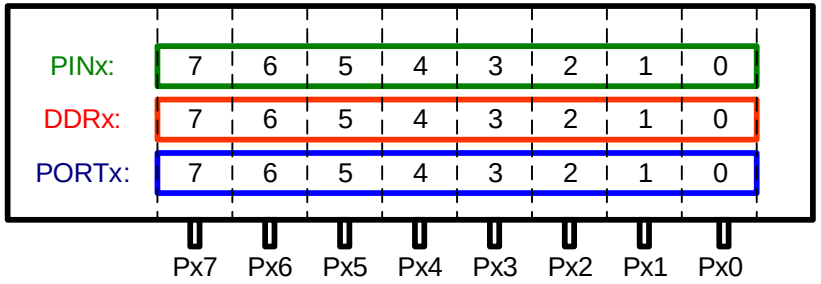

PDIP
(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP1) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

The Structure of I/O pins



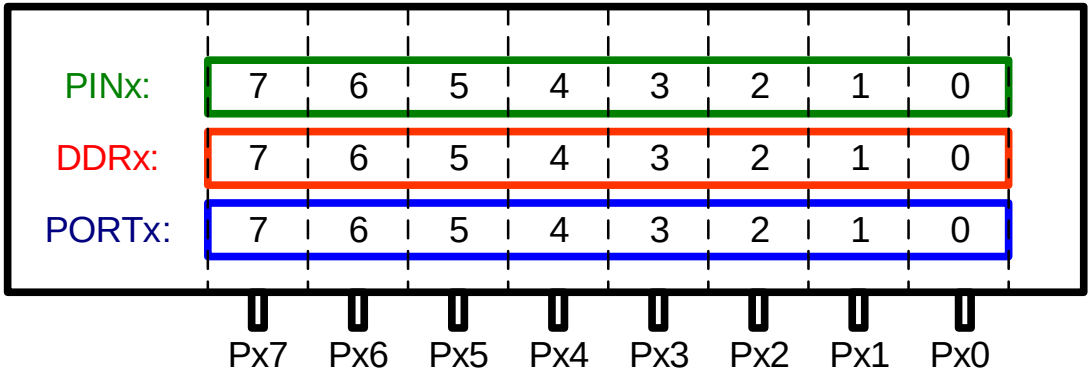
The Structure of I/O pins



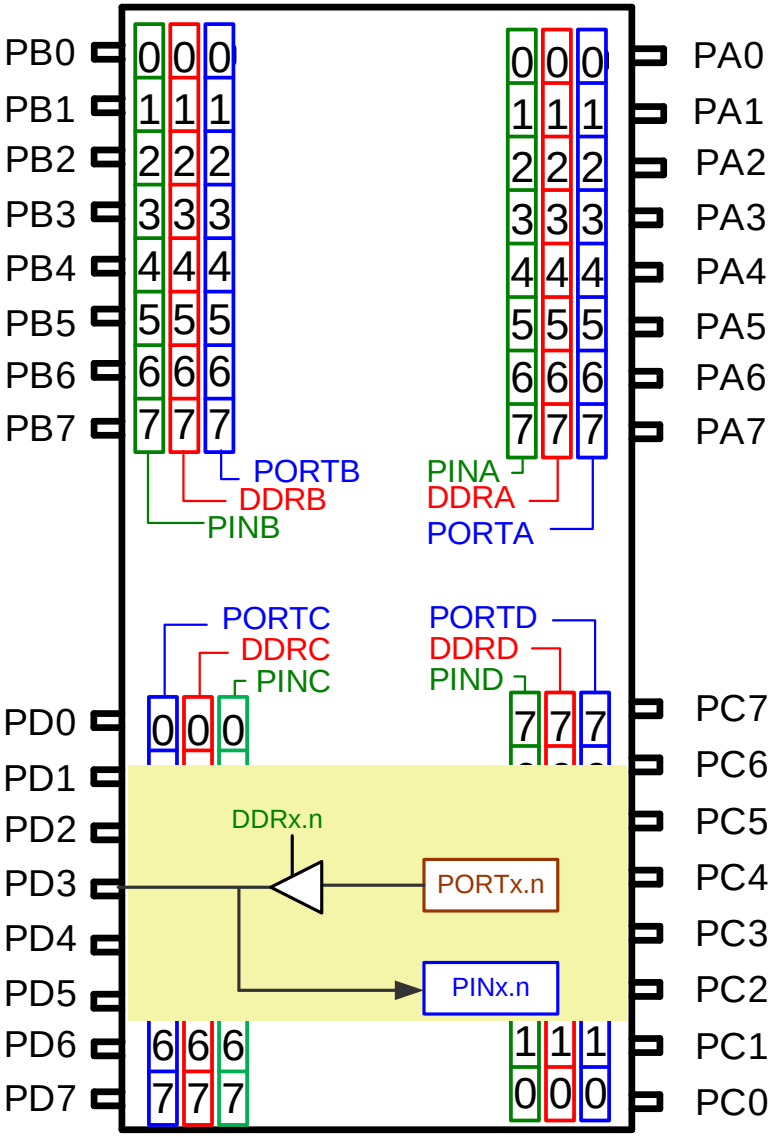


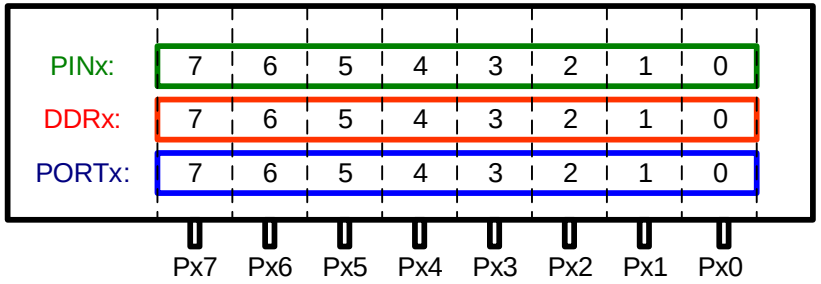
I/O Port Programming

- Each port has three (3) I/O registers associated with it:
  - PORTx [Data Register][Read/Write]
  - DDRx [**D**ata **D**irection **R**egister] [Read/Write]
  - PINx [**P**ort **IN**put] [Read-Only]
  - For example: for PORTB, we'll have:
    - PORTB
    - DDRB
    - PINB



The Structure of I/O pins

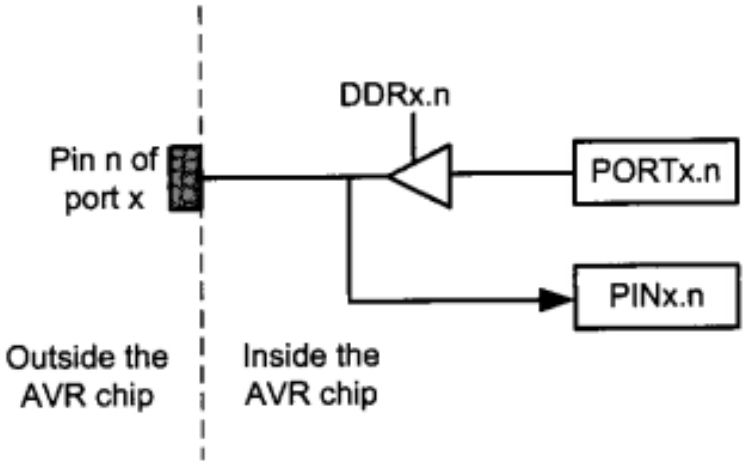




PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

I/O Registers

- DDRx
  - Determines port direction i.e Input or Output
  - 1 = output mode, 0 = input mode
  - For example: to configure Port B as output load 0xFF in DDRB
  - Note: Upon reset
- PORTx
  - Value send through this register
- PINx
  - To read the value

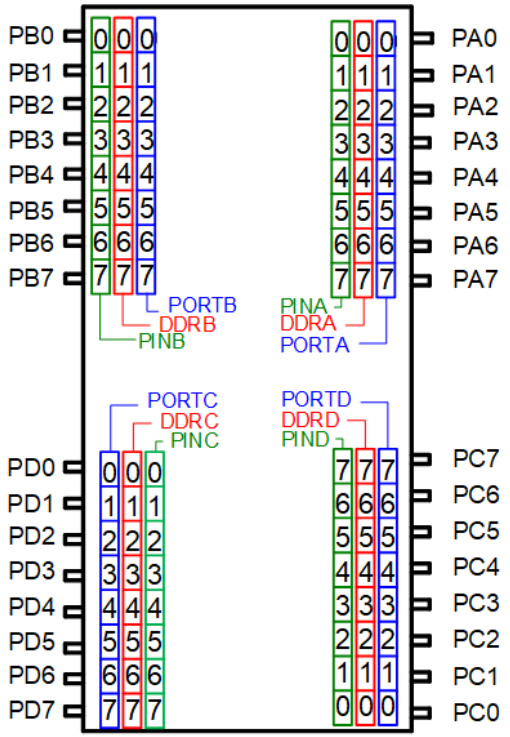


Example: Outputting Data

- Write a program that makes all the pins of PORTB one.

DDRB:	1	1	1	1	1	1	1	1	
PORTB:	1	1	1	1	1	1	1	1	

```
LDI   R20,0xFF   ;R20 = 11111111 (binary)
OUT   DDRB,R20   ;DDRB = R20
OUT   PORTB,R20  ;PORTB = R20
```



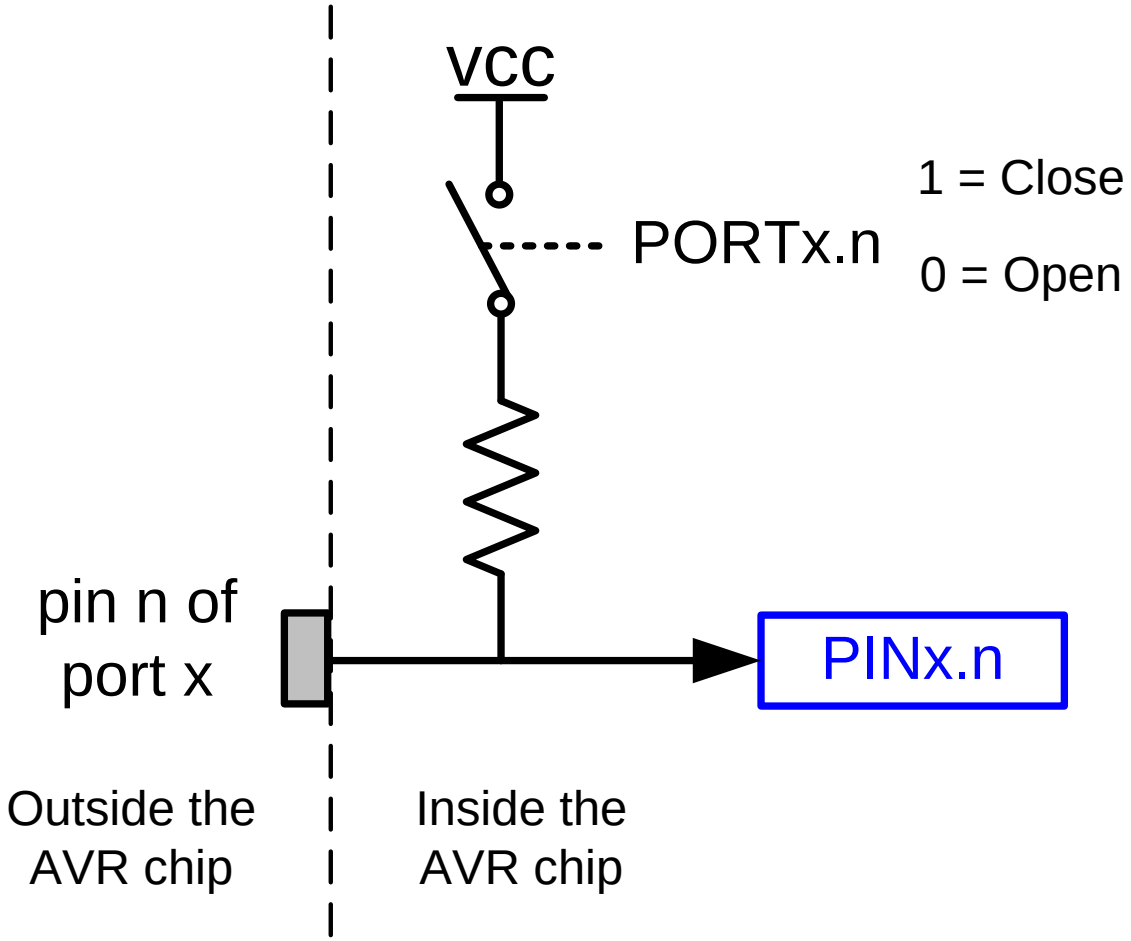
PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example: Outputting Data

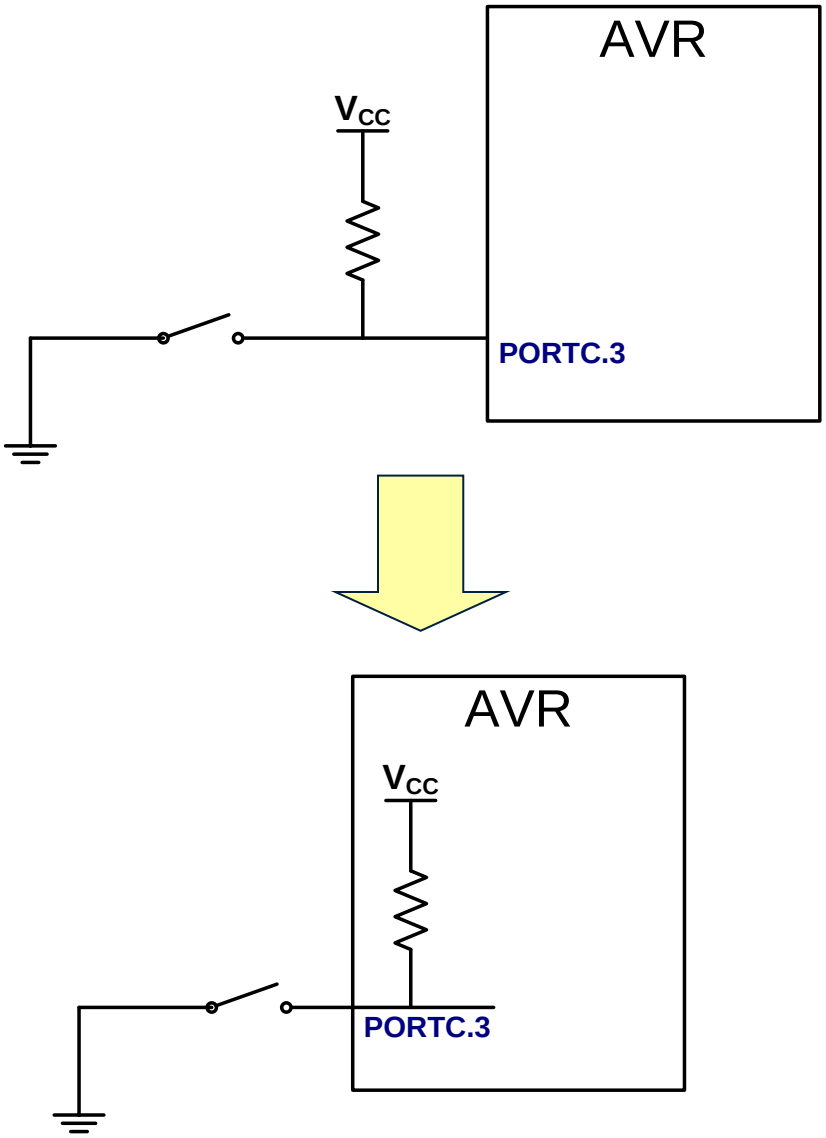
- The following code will toggle all 8 bits of Port B forever with some time delay between "on" and "off" states:

```
    LDI   R16,0xFF    ;R16 = 0xFF = 0b11111111
    OUT   DDRB,R16    ;make Port B an output port (1111 1111)
L1: LDI   R16,0x55    ;R16 = 0x55 = 0b01010101
    OUT   PORTB,R16   ;put 0x55 on port B pins
    CALL   DELAY
    LDI   R16,0xAA    ;R16 = 0xAA = 0b10101010
    OUT   PORTB,R16   ;put 0xAA on port B pins
    CALL DELAY
    RJMP L1
```

Pull-up resistor



Pull-up resistor



Example: Inputting Data

- The following code gets the data present at the pins of port C and sends it to port B indefinitely, after adding the value 5 to it:

```
     LDI   R16,0x00     ;R16 = 00000000 (binary)
     OUT   DDRC,R16     ;make Port C an input port
     LDI   R16,0xFF     ;R16 = 11111111 (binary)
     OUT   DDRB,R16     ;make Port B an output port(1 for Out)
L2:  IN    R16,PINC     ;read data from Port C and put in R16
     LDI   R17,5
     ADD   R16,R17      ;add 5 to it
     OUT   PORTB,R16    ;send it to Port B
     RJMP  L2           ;jump L2
```

PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Reading

- The AVR Microcontroller and Embedded Systems: Using Assembly and C by Mazidi et al., Prentice Hall
  - Chapter-4:
  - Go through all the examples carefully and make sure you run them on Atmel Studio for firm understanding.

THANK YOU