Microcontrollers and Industrial Applications 1
KOM3722
Week 3

Asst. Prof. Bahadır Çatalbaş

TRISx Register

REGISTER 10-8: TRISx: PORTx TRI-STATE REGISTER(1)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx7	TRISx6	TRISx5	TRISx4	TRISx3	TRISx2	TRISx1	TRISx0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **TRISx<7:0>:** PORTx Tri-State Control bit

1 = PORTx pin configured as an input (tri-stated)

0 = PORTx pin configured as an output

Note 1: Register description for TRISA, TRISB, TRISC and TRISD.

ANSELB Register

REGISTER 10-4: ANSELB – PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **ANSB<5:0>:** RB<5:0> Analog Select bit

1 = Digital input buffer disabled

0 = Digital input buffer enabled

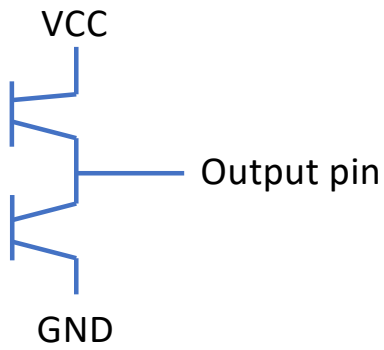
Blinking Led Program

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f45k22.h>
#include <htc.h>

#define _XTAL_FREQ 8000000

#pragma config FOSC = HSHP
#pragma config WDTEN = OFF

void main(void) {
   ANSELB = 0x00;
   TRISB = 0xFE;
   PORTB = 0;
   while(1){
      PORTBbits.RB0 = !PORTBbits.RB0;
      __delay_ms(500);
   }
   return;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f45k22.h>
#include <htc.h>

#define _XTAL_FREQ 8000000

#pragma config FOSC = HSHP
#pragma config WDTEN = OFF

void main(void) {
    ANSELB = 0x00; //0000 0000 ANSELB =
    TRISB = 0xFE;  //1111 1110 RB0 is ou
    PORTB = 0;
    while(1){
        PORTBbits.RB0 = !PORTBbits.RB0;
        __delay_ms(500); // requires fre
    }
    return;
}
```

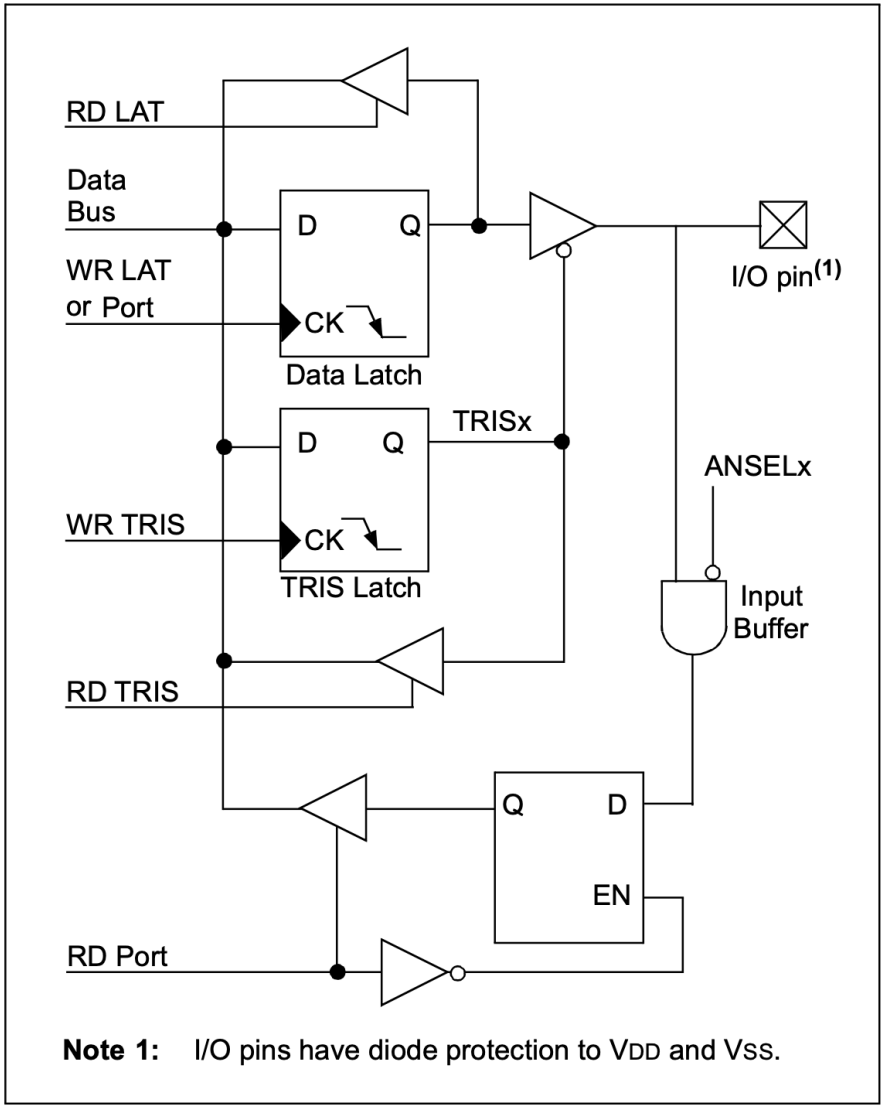
I/O Ports

- TRISx –> 0: Output
  1: Input
- ANSELx –> 0: Digital
  1: Analog
- Output pin configuration



Problematic for TRISx: 0 and ANSELx: 1

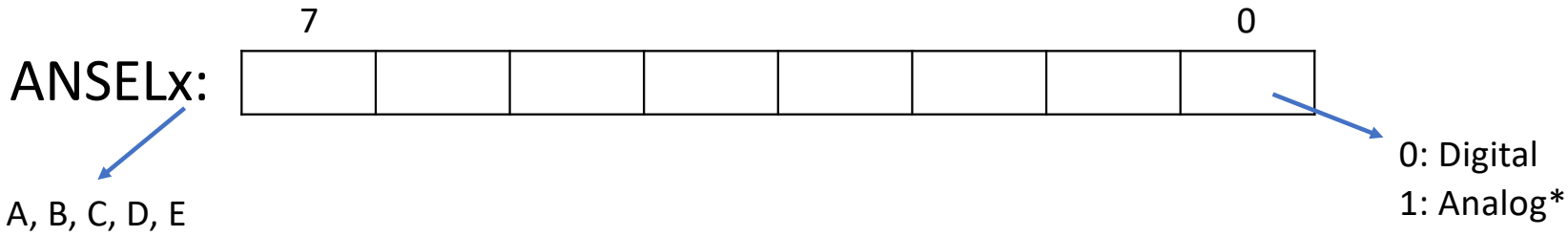
FIGURE 10-1: GENERIC I/O PORT OPERATION



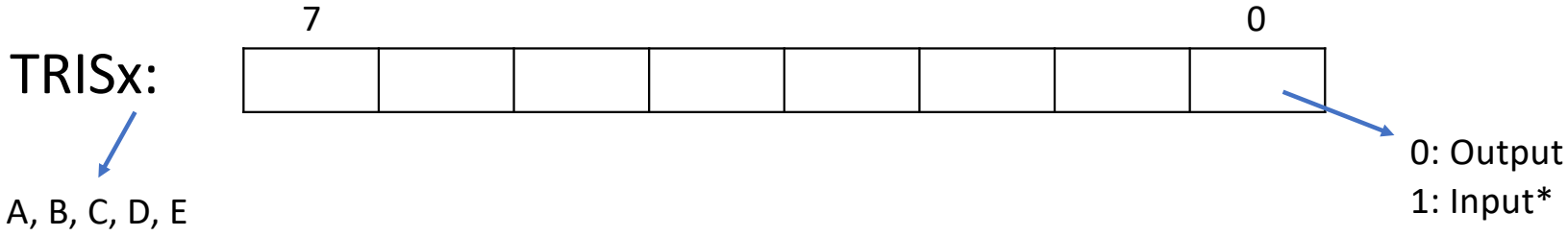
Note 1: I/O pins have diode protection to VDD and VSS.

Port Registers

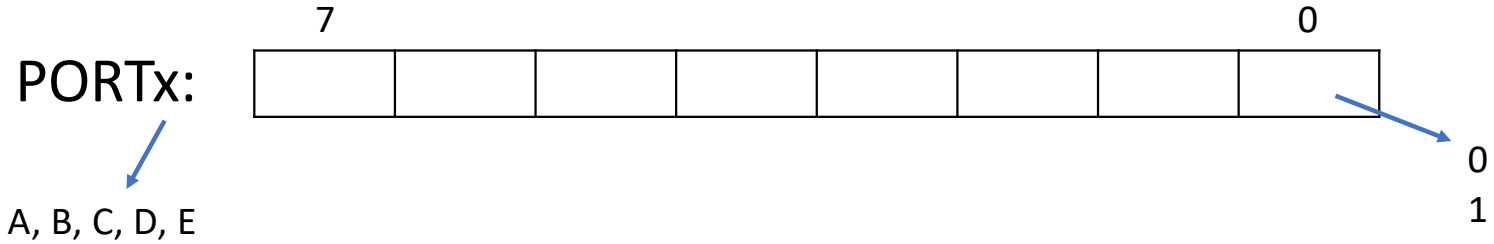
- ANSELx:



- TRISx:



- PORTx:



Operators

- ~: bytewise reverse
  - PORTB = 0b11110011
  - PORTB = ~0x0C

They are same

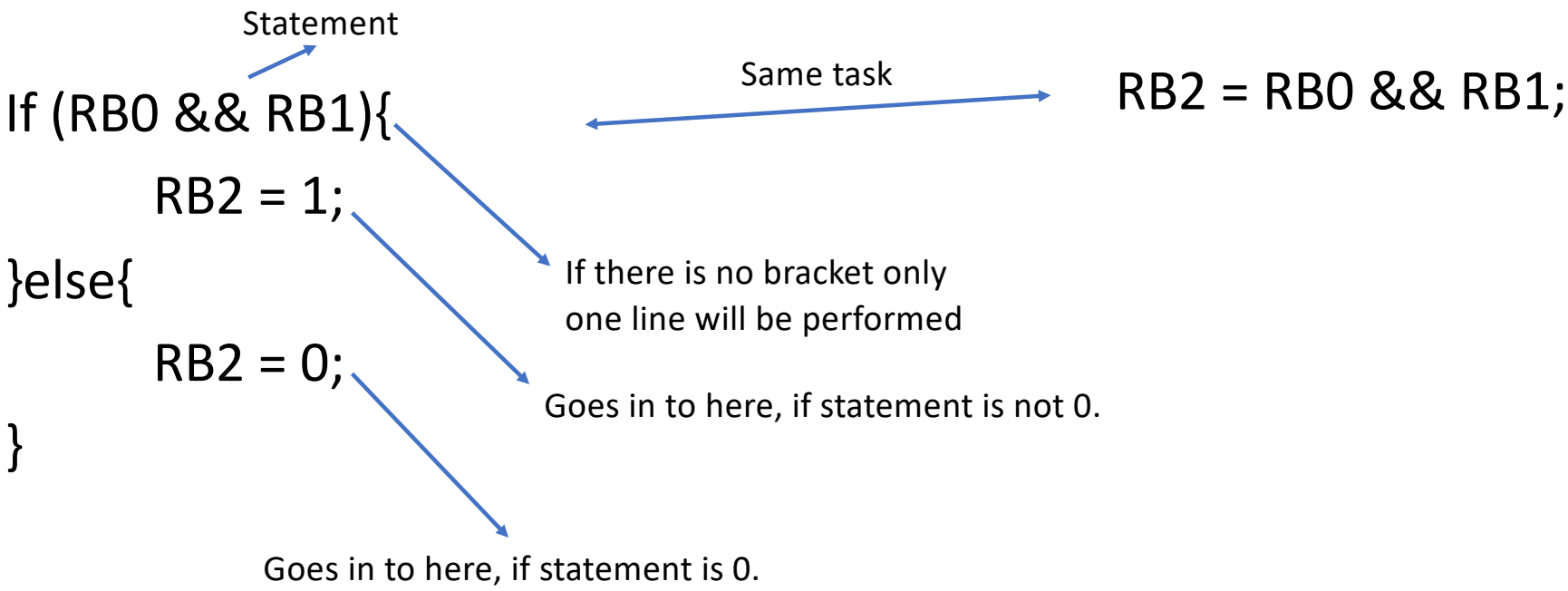
- !: bitwise reverse
  - RB0 = !RB0;
- Operator
  - Btyewise
    - &, |
  - Bitwise
    - &&, ||

- ^: bitwise XOR:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- >>: shift to right
  - A = A >> 2 → A = A/4
- <<: shift to left
  - A = A << 1 → A = A*2

Programing With Operators



PORT B

10.3 Additional PORTB Pin Functions

PORTB pins RB<7:4> have an interrupt-on-change option. All PORTB pins have a weak pull-up option.

10.3.1 WEAK PULL-UPS

Each of the PORTB pins has an individually controlled weak internal pull-up. When set, each bit of the WPUB register enables the corresponding pin pull-up. When cleared, the $\overline{\text{RBPU}}$ bit of the INTCON2 register enables pull-ups on all pins which also have their corresponding WPUB bit set. When set, the $\overline{\text{RBPU}}$ bit disables all weak pull-ups. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

TABLE 10-6: REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	150
ECCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		202
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
ECCP3AS	CCP3ASE	CCP3AS<2:0>			PSS3AC<1:0>		PSS3BD<1:0>		202
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	110
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	111
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	153
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	152
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	148
SLRCON	—	—	—	SLRE(1)	SLRD(1)	SLRC	SLRB	SLRA	153
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		167
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSCEN	$\overline{\text{T3SYNC}}$	T3RD16	TMR3ON	166
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		167
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	152

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for PORTB.

Note 1: Available on PIC18(L)F4XK22 devices.

REGISTER 9-2: INTCON2: INTERRUPT CONTROL 2 REGISTER

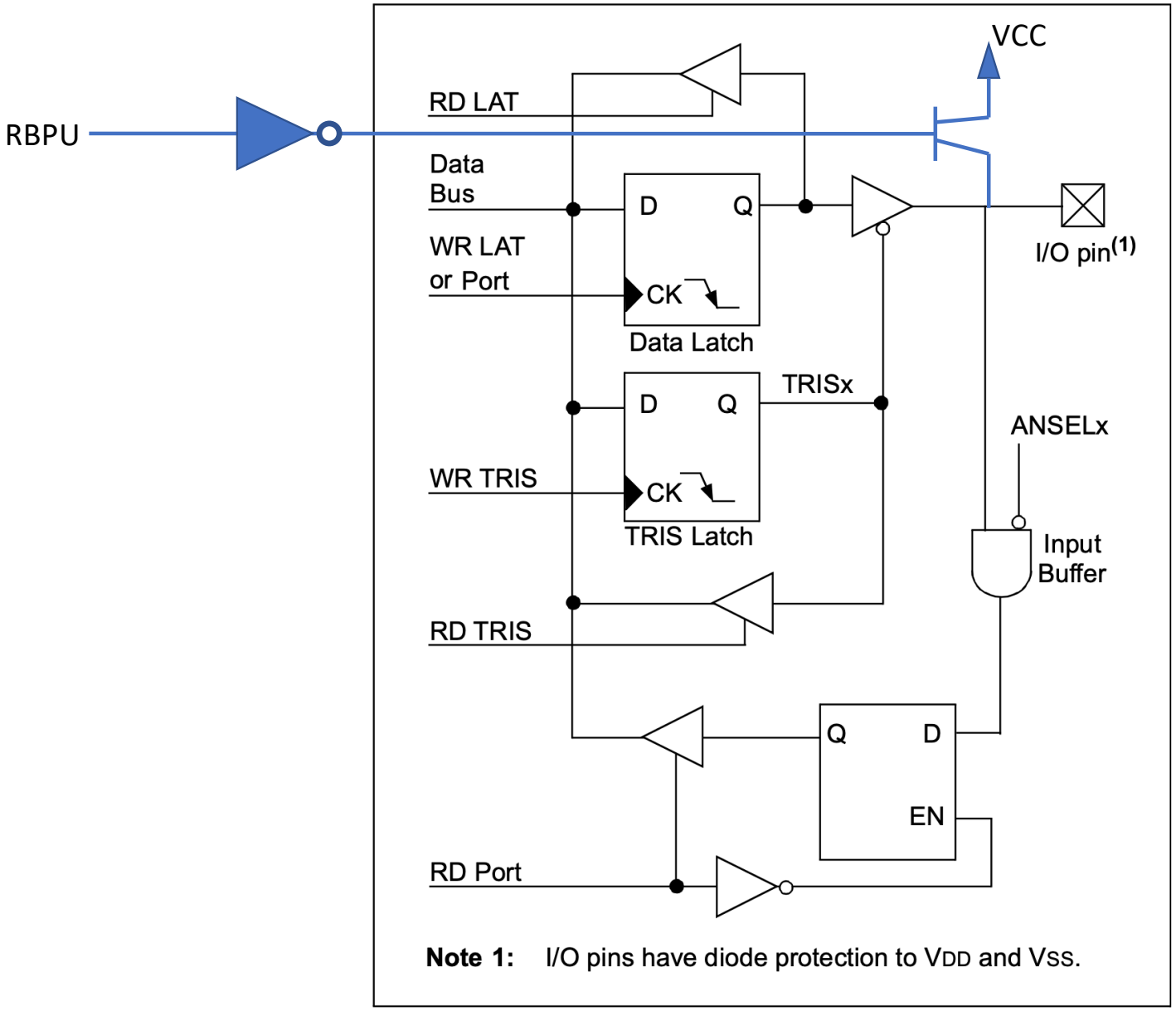
R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit

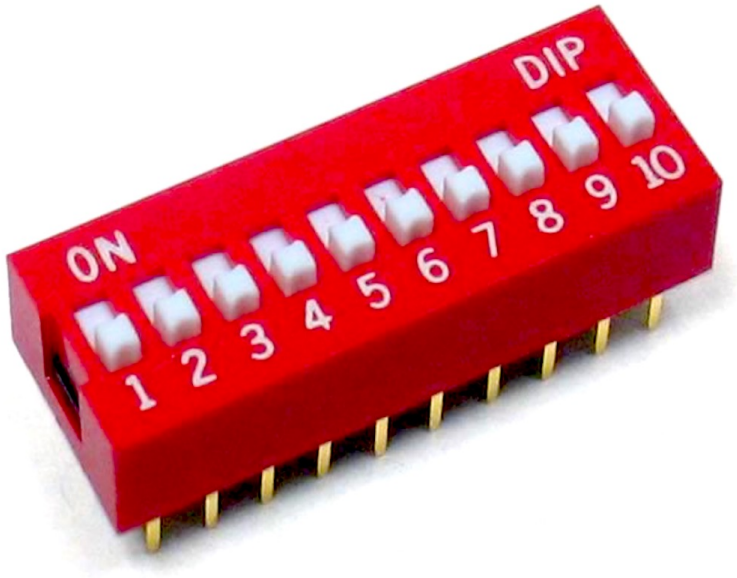
1 = All PORTB pull-ups are disabled

0 = PORTB pull-ups are enabled provided that the pin is an input and the corresponding WPUB bit is set.



Note 1: I/O pins have diode protection to VDD and VSS.

Use of Weak Pull-Up



https://www.hobidevre.com/10pin-dip-switch

Dip Switch Application

```
#include <stdio.h>
#include <stdlib.h>
#include <pic18f45k22.h>
#include <htc.h>

#define _XTAL_FREQ 8000000

#pragma config FOSC = HSHP
#pragma config WDTEN = OFF

#define RB0 PORTBbits.RB0
#define RB1 PORTBbits.RB1
#define RB2 PORTBbits.RB2
#define RB3 PORTBbits.RB3
#define RB4 PORTBbits.RB4
#define RB5 PORTBbits.RB5
#define RB6 PORTBbits.RB6
#define RB7 PORTBbits.RB7

void main(void) {
   ANSELB = 0x00; //all is digital
   TRISB = 0xF0;  //4 high pit is input
   PORTB = 0;
   RBPU = 0;
   while(1){
      RB0 = !RB4;
      RB1 = !RB5;
      RB2 = !RB6;
      RB3 = !RB7;
      // or
      //PORTB = (~PORTB & 0xF0)>>4;
   }
   return;
}
```