Devdhar Kodiatar

ME 333

02/19/19 Homework 11

Chapter 7:

1. False. The input pin can also read 'high' when disconnected if the pin is configured with an internal pull-up resistor.

2. b). AD1PCFG = 0x0000001E;
TRISB = 0x0000FFF9;
ODCB = 0x00000004;
CNPUE = 0x00000010;
CNCON = 0x00008000;
CNEN = 0x00000001;

a). Here pin 2 can have an external pull-up resistor.
We should not try to source or sink more than 10 mA.

$\therefore$ minimum resistance $= \frac{5\,V}{0.01\,A} = 500\,\Omega$

The voltage across this resistance should not be more the 9V otherwise, this may damage the PIC32,

$\therefore$ maximum resistance $= \frac{9\,V}{0.01\,A} = 900\,\Omega$

Chapter 8:

1. No. of ticks = $(PR+1) \times 64$

$\Rightarrow (PR+1) \times 64 = \frac{16 \times 10^6 \text{ ns}}{12.5 \text{ ns}}$

$\Rightarrow PR = 19999$

```
T3CON = 0x8060;
PR3   = 0x4E1F;
```

Chapter 9:

1. $f_{PWM} \geq 100\, f_c \Rightarrow f_c \leq \frac{f_{PWM}}{100}$, $f_c \geq 10 f_a \Rightarrow f_a \leq \frac{f_c}{10}$

$\Rightarrow f_a \leq \frac{f_{PWM}}{1000}$

Maximum value of $f_{PWM} = \frac{80 \text{ MHz}}{2^n}$

$\therefore$ maximum $f_a = \frac{80}{1000 \times 2^n}$ MHz

$= 2^{3-n} \times 10$ KHz

Also, $f_c = \frac{1}{2\pi RC}$

$\Rightarrow$ minimum $RC = \frac{1}{2\pi \times (10 f_a)} = \frac{2^{n-3}}{200\pi}$

Chapter 10 :-

1. The least possible value of TAD is 75 ns

∵ we don't need to configure the ADC to sample automatically, we can choose the sampling time = 132 ns.

∴ minimum read time = 132 ns + 12 × 75 ns = 1032 ns

```
AD1CON1bits.ON = 1;
AD1CON1bits.SSRC = 0b111;
AD1CON1bits.ASAM = 0;
AD1CON3bits.ADCS = 2;
```

2.
```
unsigned int adc_sample_convert(int pin){
    unsigned int elapsed = 0, finish_time = 0;
    AD1CHSbits.CH0SA = pin;
    AD1CON1bits.SAMP = 1;
    elapsed = CP0_GET_COUNT();
    finish_time = elapsed + 10; // 10 core ticks = 250 ns
    while(CP0_GET_COUNT() < finish_time){
        ;
    }
    AD1CON1bits.SAMP = 0;
    AD1CON1bits.SSRC = 0b111; // Auto conversion
    while(!AD1CON1bits.DONE){
        ;
    }
    return ADC1BUF0;
}
```