NAME:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ECE-C304 Lab – Assignment # 1

(1A,1B due 2nd Week, 1C due 3rd Week (Mon for Sec060, Tue for Sec061))

The lab is based on the handout supplied

*Note: Always use the part no CY8C29466-24PXI*

Lab 1A - CPU

1A-1 What does this code do?

1. Takes the value on Port 1 and stores it in the Accumulator.
2. Increments the Accumulator by 1.
3. Outputs the value of the accumulator on Port 1.
4. Jumps back to step 1.

1A-2 What is the loop cycle time? Show details of calculation

SysClk/4 = 24Mhz/4 = 6Mhz

Inverse of frequency gets us a period of 1/6 of microsecond per cycle.

Every instruction requires a certain number of clock cycles to complete.

|  |  |
| --- | --- |
| Instruction | Clock Cycles |
| INC | 4 |
| JMP | 5 |
| MOV A,Reg[expr] | 6 |
| MOV Reg[expr],A | 5 |

4+5+6+5 = 20 Clk Cycles

20\*1/6 = **3 & 1/3** microseconds per loop.

1A-3 For 12 MHz CPU clock what is the output frequency of each pin?

12Mhz/4 = 3 Mhz

Period = 1/3 seconds

20\*1/3 = 6 & 2/3 microseconds per loop

Frequency per loop = 1/(20/3 seconds) = 0.15 Mhz

First pin’s frequency is the frequency of the loop

Loop frequency / 2 for each subsequent pin because of the increment instruction.

|  |  |
| --- | --- |
| Pin | Frequency kHz |
| 0 | 75 |
| 1 | 36 |
| 2 | 19 |
| 3 | 9.4 |
| 4 | 4.7 |
| 5 | 2.3 |
| 6 | 1.2 |
| 7 | 0.6 |

1A\_4 What are the measured values of the frequencies above?

|  |  |
| --- | --- |
| Pin | Frequency kHz |
| 0 | 150 |
| 1 | 75 |
| 2 | 36 |
| 3 | 19 |
| 4 | 9.4 |
| 5 | 4.7 |
| 6 | 2.3 |
| 7 | 1.2 |

1A-5 If CPU clock is set at 3 MHz, how will these values change?

The value compared to 24 MHz would be 1/8 the frequency for each pin.

The way 12 compared to 24 is half the frequency for each pin.

1A-6 What are the measured values?

|  |  |
| --- | --- |
| Pin | Frequency kHz |
| 0 | 150 |
| 1 | 75 |
| 2 | 36 |
| 3 | 19 |
| 4 | 9.4 |
| 5 | 4.7 |
| 6 | 2.3 |
| 7 | 1.2 |

1A-7 How many ‘nop’s need to be inserted in the loop to reduce the frequencies by half?

Modify the code and list the measured frequencies (for CPU clock of 12 MHz)

**(Submit modified code)**

NOP costs 4 clock cycles, but to reduce the frequency by half we need to double the number of cycles in the loop. Since the number of cycles is 20 the number of NOPs we need is 5 to make 40 cycles per loop.

;----------------------------------------------------------------------

; Assembly main line

;----------------------------------------------------------------------

include "m8c.inc" ; part specific constants and macros

include "memory.inc" ; Constants & macros for SMM/LMM and Compiler

include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export \_main

\_main:

loop:

mov A,reg[PRT1DR] ; 4 clk cycles

inc A ; 4

mov reg[PRT1DR],A ; 4

jmp loop ; 5

nop ; 4

nop

nop

nop

nop

1A-8 Comment out the line

include “m8c.inc”

and compile main.asm. What error do you get and why?

!ERROR {linker} file 'main.o': undefined symbol 'PRT1DR'

This error occurs because the compiler doesn’t know what certain variables mean unless it imports the definitions that are found in m8c.inc

1A-9 Find the address of PRT1DR in the I/O register space and modify the code in such a way that

the code still runs correctly **(Submit modified code)**

;-----------------------------------------------------------------------------

; Assembly main line

;-----------------------------------------------------------------------------

//include "m8c.inc" ; part specific constants and macros

include "memory.inc" ; Constants & macros for SMM/LMM and Compiler

include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export \_main

\_main:

loop:

mov A,reg[04h] ; 4 clk cycles

inc A ; 4

mov reg[04h],A ; 4

jmp loop ; 5

nop ; 4

nop

nop

nop

nop

.terminate:

jmp .terminate

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab 1B- GPIO Output

1B-0 Use the code in Fig 1.7 and make Port1 pins “Pull-up”. Plot the waveform observed

on the scope and explain why it is so.

1B-1 Now “Wire-AND” bit 0 and bit 6 as suggested. What waveform do you expect to see? Explain

1B-2 Do your observations agree with your predictions?

1B-3 Now break the input-output interaction by changing the code using bShadow as in Fig 1.10.

while still wire-ANDing bit 0 and bit 6. What do the waveforms look like? Draw them.

1B-4 What is loop cycle time in CPU cycles for the code in Fig 1.10? Show details of work.

1B-5 What RAM location is allocated to bShadow? (Hint: Find this from the map file for Lab1B)

1B-6 Convert the assembly code for Lab 1B to machine code manually. Compare with the machine

code given in the listing file for Lab 1B.

1B-7 Modify the code in such a way that bShadow is in location 40h **(Submit modified code)**

Confirm from listing file.

;-----------------------------------------------------------------------------

; Assembly main line

;-----------------------------------------------------------------------------

include "m8c.inc" ; part specific constants and macros

include "memory.inc" ; Constants & macros for SMM/LMM and Compiler

include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export bShadow

area myArea(ROM,ABS,CON)

ORG 40h

bShadow:

BLK 1; Reserve a RAM location named bShadow

area text(ROM,REL)

export \_main

\_main:

mov [bShadow],0

loop:

inc [bShadow]

mov A,[bShadow]

mov reg[PRT1DR],A

jmp loop

.terminate:

jmp .terminate

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab 1C - GPIO Input

**(Submit fully commented code for parts 1C-1 and 1C-2)**

1C-0 (Without debouncing)

Run the following code (SW on P1[0], LEDs on P[1]-P1[4])

mov {bShadow], 0

begin: tst reg[PRT1DR],01h

jz begin

add [bShadow],2

mov A,[bShadow]

mov reg[PRT1DR], A

jmp begin

As you repeatedly press and release SW what happens and why?

**The lights light up somewhere in the sequence, but not in the order you would expect it to iterate. This is because as you hold down the switch it iterates multiple times and when you release it will be somewhere in the sequence, but not where you’d expect.**

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1C-1

Code the switch debouncing program in Assembler and show correct functioning

;-----------------------------------------------------------------------------

; Assembly main line

;-----------------------------------------------------------------------------

include "m8c.inc" ; part specific constants and macros

include "memory.inc" ; Constants & macros for SMM/LMM and Compiler

include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export bShadow

area MyArea(RAM) ;Create storage location for variables

bShadow: BLK 1

area text(ROM,REL)

export \_main

\_main:

mov [bShadow], 0

begin:

tst reg[PRT1DR],01h

jz begin

add [bShadow],2

mov A,[bShadow]

mov reg[PRT1DR], A

jmp waitTilRelease

waitTilRelease:

tst reg[PRT1DR],01h

jnz waitTilRelease

jmp begin

.terminate:

jmp .terminate

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1C-2

Code the switch debouncing program in C and show correct functioning.

//----------------------------------------------------------------------------

// C main line

//----------------------------------------------------------------------------

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

BYTE bShadow;

void main()

{

bShadow = 0x00;

PRT1DR = bShadow;

while(1)

{

while(!((PRT1DR & 0x01) == 0x01)){

// loop here until button is pushed

}

bShadow += 0x02;

PRT1DR = bShadow;

while((PRT1DR & 0x01) == 0x01){

// loop here until button is released

}

}

}

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1C-3

Modify the flowchart and the assembly language code in 1C-1 so that the release of the switch is also debounced. The count on the LEDs should increment every time the switch is pressed or released.

;-----------------------------------------------------------------------------

; Assembly main line

;-----------------------------------------------------------------------------

include "m8c.inc" ; part specific constants and macros

include "memory.inc" ; Constants & macros for SMM/LMM and Compiler

include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export bShadow

area MyArea(RAM) ;Create storage location for variables

bShadow: BLK 1

area text(ROM,REL)

export \_main

\_main:

mov [bShadow], 0

begin:

tst reg[PRT1DR],01h

jz begin

add [bShadow],2

mov A,[bShadow]

mov reg[PRT1DR], A

jmp waitTilRelease

waitTilRelease:

tst reg[PRT1DR],01h

jnz waitTilRelease

add [bShadow],2

mov A,[bShadow]

mov reg[PRT1DR], A

jmp waitTilPressed

waitTilPressed:

tst reg[PRT1DR],01h

jz begin

.terminate:

jmp .terminate

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Familiarization with Assembly Language

Reference: PSoC Designer Assembly Language User Guide

1 What are the different CPU registers in the microcontroller and what is the size of the different registers?

|  |  |  |
| --- | --- | --- |
| Register | Shorthand | Size (bits) |
| Accumulator | A | 8 |
| Index | X | 8 |
| Program Counter | PC | 16 |
| Stack Pointer | SP | 8 |
| Flag | F | 8 |

2 Classify the following instructions as (i) data movement (ii) data processing or (iii) control flow

**Data Movement:**  MVI, POP

**Data Processing:** ADD, DEC,MVI

**Control Flow:**  JNC, LJMP, RETI

3 Get familiar with the addressing modes. What do the following instructions do? What is the addressing mode for each?

|  |  |
| --- | --- |
| Instruction | What it does |
| mov A, 50h | Sets A = 0x50 |
| mov A, [50h] | Sets A to the value at RAM address 0x50 |
| mov [50h],50h | Sets the value at RAM address 0x50 to the value in A |
| mov reg[50h],A | Sets the value at register 0x50 to the value in A |
| mov [40h],[50h] | Set the value at RAM address 0x40 to the value at RAM address 0x50 |

4 What is the machine language encoding for each of the above? How many clock cycles does each take to execute?

|  |  |  |
| --- | --- | --- |
| Instruction | Cycles | Machine Language Encoding |
| mov A, 50h | 4 | 0x5050 |
| mov A, [50h] | 5 | 0x5150 |
| mov [50h],50h | 8 | 0x555050 |
| mov reg[50h],A | 5 | 0x6050 |
| mov [40h],[50h] | 10 | 0x5F4050 |