

ME333 – Homework 3 – Yael Ben Shalom

Chapter 3, Exercise 1:

virtual address	physical address	Cacheable?	resides in RAM, flash, SFRs, or boot flash?
0x80000020	00000020	KSEG0	RAM
0xA0000020	00000020	KSEG1	RAM
0xBF800001	1F800001	KSEG1	SFR
0x9FC00111	1FC00111	KSEG0	boot flash
0x9D001000	1D001000	KSEG0	flash

Chapter 3, Exercise 2:

The bootloader install your program in virtual memory at the address 0xBD001970.

Chapter 3, Exercise 3:

- a) The bits that can be used as input/outputs for each of Ports B through G are:

PORTB - 31/15 – 16/0

PORTC - 31/15 – 28/12

PORTD - 27/11 – 16/0

PORTE - 23/7 – 16/0

PORTF - 21/5 – 19/3, 17/1 – 16/0

PORTG - 25/9 – 22/6, 19/3 – 18/2

Pin 60 corresponds to bit 0 of port E.

- b) unimplemented - 31/15 – 29/13

MVEC - 28/12

unimplemented - 27/11

TPC - 26/10 – 24/8

unimplemented - 23/7 – 21/5

INT#EP - 20/4 – 16/0

Chapter 3, Exercise 4:

In order to make both lights turn on or off at the same time, we need to change line 9 for 1 to 0:

```
LATFbits.LATF1 = 0; // on the NU32, so "high" (1) = "off" and "low" (0) = "on"
```

(The modified code is on folder code/Ch3Q4)

Chapter 3, Exercise 6:

SFR	virtual address	reset value
I2C3CON	BF805000	1000
TRISC	BF886080	F000

Chapter 3, Exercise 7:

The processor.o file linked with the simplePIC project is much larger than the final .hex file, because the processor.o file contains all of the virtual addresses of the SFRs. The .hex file does not have all the functions and information that the bootloader handles, so it is shorter.

Chapter 3, Exercise 8:

- a) The lines that call the user's main function when the C runtime startup completes are 512-518:

```
and    a0,a0,0
and    a1,a1,0
la      t0,_main_entry
jr      t0
nop
.end _startup
```

- b) The names and addresses of the five SFRs with the highest addresses are:

```
bf88cb3c A C2FIFOUA31INV
bf88cb40 A C2FIFOCI31
bf88cb44 A C2FIFOCI31CLR
bf88cb48 A C2FIFOCI31SET
bf88cb4c A C2FIFOCI31INV
```

- c) There are 10 bit fields are defined:

- SPIRBF – 1 bit
- SPITBF – 1 bit
- SPITBE – 1 bit
- SPIRBE – 1 bit
- SPIROV – 1 bit
- SRMT – 1 bit
- SPITUR – 1 bit
- SPIBUSY – 1 bit

- TXBUFELM – 5 bits
- RXBUFELM – 5 bits

These coincide with the Data Sheet (Table 14-4).

Chapter 3, Exercise 9:

Three C commands that set bits 2 and 3 of TRISD to 1, clear bits 1 and 5, and flip bits 0 and 4:

1. TRISDSET = 0b1100 or 0xC
2. TRISDCLR = 0b100010 or 0x22
3. TRISDINV = 0b10001 or 0x11

Chapter 4, Exercise 1:

The private and public functions, constants, and global variables in NU32.c are:

Public functions: NU32_Startup (meant to be used in other C files)
 NU32_ReadUART3
 NU32_WriteUART3
Private constant: NU32_DESIRED_BAUD

Chapter 4, Exercise 2:

- a) The modified code is on folder code/Ch4Q2a.
- b) The modified code is on folder code/Ch4Q2b.

The helper.h function contains the function prototype of all the non-main() functions from invest.c, the Investment datatype definition and the constant variable MAX_YEARS. The helper.c contains the function themselves and the main.c file contains only the main functions, and includes the helper.h file for the necessary functions.

- c) The modified code is on folder code/Ch4Q2c.

In this part, I separated the main.c and the other part of the code in the same way, only this time I added a new file called calculate.c for the calculation function calculateGrowth() (and its function prototype on calculate.h).

Chapter 4, Exercise 4:

A function that clears a single line of the LCD:

```
Void LCD_ClearLine(int ln) {  
    char space = " ";  
    LCD_Move(ln, 0);                       // Moves the cursor to column 0 on line 'ln'  
    for (int i=1; i<=16; i++) {  
        LCD_WriteChar(space);             // Writes " "  
    }  
}
```