

Assignment #1

(max = 80)

Work through the “Notes for Assignment #1” (under Course Notes). These notes will assist you in the installation of QtSpim on your computer and will give you an opportunity to experiment with QtSpim.

Next, read the first 6 sections of Chapter 1 in the *Computer Organization and Design* text. This reading will give you a glimpse of what we will be looking at during the term.

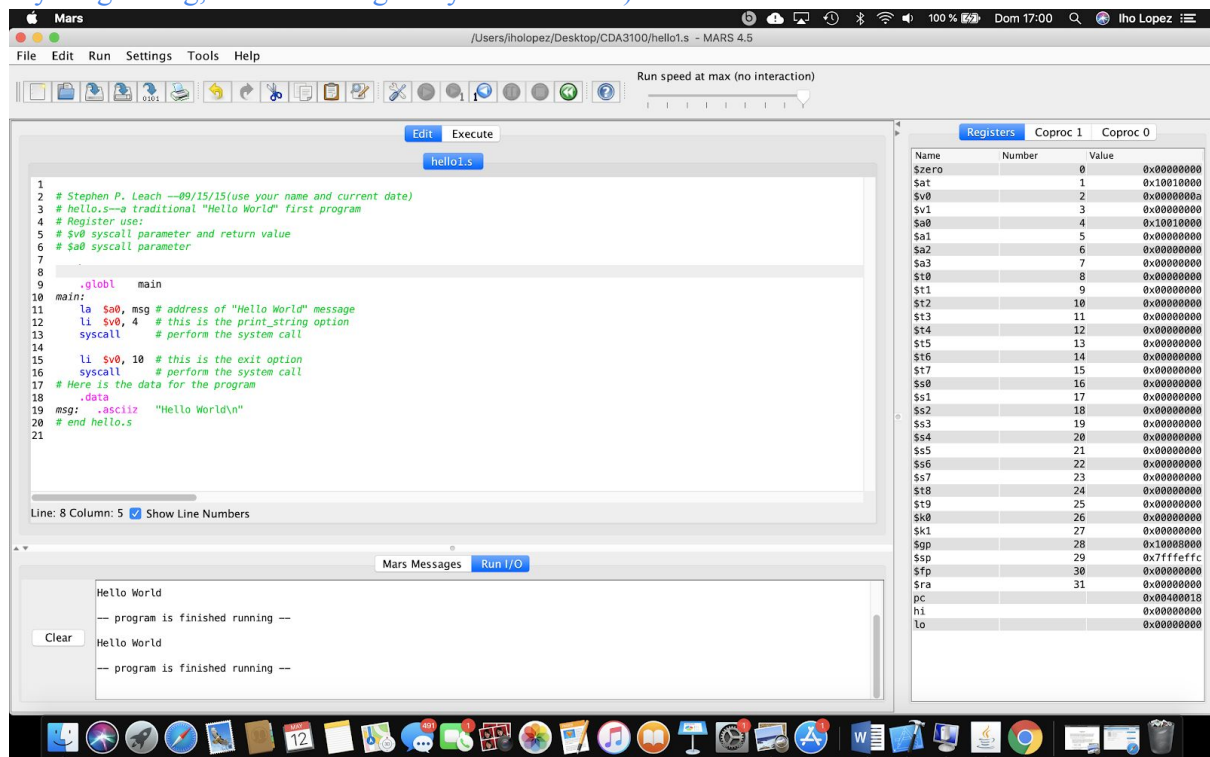
Afterwards, submit answers for the following problems (where appropriate, show the details of your calculations):

1. When working through the “Notes for Assignment #1”, you created a small assembly program called **hello.s**. Remove the period from the “.text” line in the program. Try loading this modified program into QtSpim. (2 points)

- a) What type of error do you get? **I did not get any error**
- b) Does the program load and execute properly if you get rid of the “text” line completely? **It loaded and ran the program without any issues**

QtSpim Version 9.1.20 of August 29, 2017
Copyright 1990-2017 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

I even ran the program with MARS (another assembly program to make sure that I was not doing anything wrong, but I did not get any error as well).



2. I assume that you have just executed **hello.s** and are still inside of QtSpim. There are actually several different sections in the QtSpim screen. On the left side you are able to view the contents of the various MIPS integer and floating point registers; we will be learning all about these registers in the coming weeks. Make sure that "Int Regs" is selected. Notice that the PC register contains the value 400038. The General Purpose Registers are numbered starting with 0. Register 1 (also called "at") contains the value 10010000. (8 points)
 - a) Register 2 goes by what other name? **v0** What value is in register 2? **a**
 - b) Register 15 goes by what other name? **t7** What value is in register 15? **0**
 - c) Register 28 goes by what other name? **gp** What value is in register 28? **10008000**
 - d) Register 31 goes by what other name? **ra** What value is in register 31? **400018**
3. Let's now look at the right side of the QtSpim screen. We can view our data and our code ("Text") on this side. Make sure that "Text" is selected. We are able to see the instructions as they are stored in memory. For example, location 00400000 contains the value 8fa40000, which is the instruction "lw \$4, 0(\$29)". (8 points)

- a) What value is stored at location 0040000c? [00041080](#) What instruction is this? `sll $2, $4, 2`
 - b) What value is stored at location 00400034? [3402000a](#) What instruction is this? `ori $2, $0, 10`
 - c) What value is stored at location 00400038? [0000000c](#) What instruction is this? `syscall`
 - d) What value is stored at location 80000184? [3c019000](#)
What instruction is this? `lui $1, -28672`
4. Let's move on to the data portion of the QtSpim screen (still assuming that we just executed **hello.s**). We are able to see the data (and the stack) as it is stored in memory. For example, location 90000030 contains the value 2000205d. (2 points)
- a) What value is stored at location 10010000? [6c6c6548](#)
 - b) What value is stored at location 90000160? [63614d5b](#)
5. The bottom of the QtSpim screen contains some copyright information. (2 points)
- a) Who developed the SPIM product (who has the original copyright)? [James Larus](#)
 - b) What year was the original copyright established? [1990](#)
6. I have created another small assembly program called **bio.s** that gives a little bit of information about yours truly. Here is the output that appears in the Console window when I execute the program:

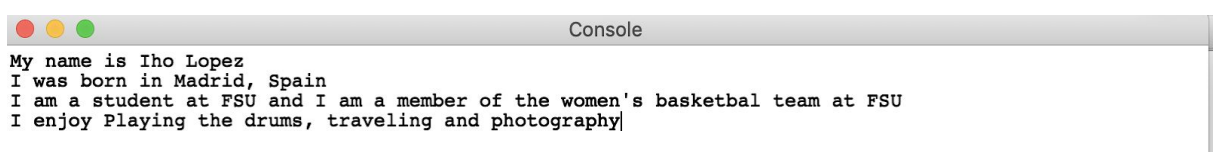
My name is Steve Leach

I was born in Key West, Florida

I am a faculty member at FSU's Panama City Campus

I enjoy reading, singing karaoke and playing with my grandchildren

Create a **bio.s** program of your own that tells me a little about yourself. Make sure that your program is documented using a style similar to that used in **hello.s**. Submit a separate file called **bio.s** as well as placing your code in this assignment submission; the Mentor will clarify what I mean by this. (15 points)



```
My name is Iho Lopez
I was born in Madrid, Spain
I am a student at FSU and I am a member of the women's basketball team at FSU
I enjoy Playing the drums, traveling and photography
```

7. Do Exercise 1.4 on Page 55 in the textbook. For part (b), give answer to nearest millisecond. (4 points ... 2 points each)

a) $\text{Number of bytes per pixel} = 8 \text{ bits} * 3 \text{ colors} / 8 = 3 \text{ bytes}.$

$\text{Therefore } 1280 * 1024 \text{ pixels} = 1,310,720 \text{ pixels}.$

$\text{Then } 1,310,720 * 3 \text{ bytes/frame} = 3,932,160 \text{ bytes/frame which is the minimum size}.$

b) $\text{Minimum time required to take the frame to be sent over a 100 Mbit/s network}$
 $(3,932,160 \text{ bytes} * (8 \text{ bits/byte})) / 100 \text{E6 bits/second}$

$\text{Therefore } (3,932,160 * 8) / 100,000,000 = 0.3145728 \text{ seconds. It will take } 0.3145728$
 $\text{seconds at a minimum for the frame to be sent over a 100 Mbit/s network}.$

8. • Do Exercise 1.5 on Page 55 in the textbook. For part (a), simply compute the instructions/second (four significant digits) for each of the three processors (3 points). Part (b) will have six answers: the number of cycles and the number of instructions (four significant digits) for each of the three processors (6 points). For part (c), compute the new execution time and the new CPI for each of the three processors (4 points) and then compute the new required clock rate for each of the three processors, expressing these rates in GHz (four significant digits) (6 points). (19 points total)

a) $P1: 3\text{GHz} / 1.5 = 2 * 10^9 \text{ instructions per second}$

$P2: 2.5\text{GHz} / 1.0 = 2.5 * 10^9 \text{ instructions per second}$

$P3: 4\text{GHz} / 2.2 = 1.82 * 10^9 \text{ instructions per second}$

With P2 having the highest performance.

b) Number of Cycles:

$P1: 3\text{GHz} * 10 = 3 * 10^{10} \text{ cycles}$

$P2: 2.5\text{GHz} * 10 = 2.5 * 10^{10} \text{ cycles}$

$P3: 4\text{GHz} * 10 = 4 * 10^{10} \text{ cycles}$

Number of instructions:

$P1: 3\text{GHz} * 10 / 1.5 = 2 * 10^{10} \text{ instructions}$

$P2: 2.5\text{GHz} * 10 / 1.0 = 2.5 * 10^{10} \text{ instructions}$

$P3: 4\text{GHz} * 10 / 2.2 = 1.82 * 10^{10} \text{ instructions}$

c)

New Clock:

$$P1: 3\text{GHz} * 1.71 = 5.13 \text{ GHz}$$

$$P2: 2.5\text{GHz} * 1.71 = 4.27 \text{ GHz}$$

$$P3: 4\text{GHz} * 1.71 = 6.84 \text{ GHz}$$

So the clock rate will now be $1.71 * \text{Clock rate}$

$$\text{Execution time} = (\text{Number of instructions} * \text{CPI}) / (\text{Clock rate})$$

So if we want to reduce the execution time by 30%, and CPI increases by 20%, we have:

$$\text{Execution time} * 0.7 = (\text{Number of instructions} * \text{CPI} * 1.2) / (\text{New Clock rate})$$

$$\text{New Clock rate} = \text{Clock rate} * 1.2 / 0.7 = 1.71 * \text{Clock rate}$$

9. • This is an adaptation of Exercise 1.6 on Page 55 in the textbook. We assume that there are two different implementations, P1 and P2, of the same instruction set architecture. The instructions are divided into four classes according to their CPIs (classes A, B, C and D). P1 has a clock rate of 2.5 GHz and CPIs of 1, 2, 3 and 3 for the corresponding classes of instructions. Likewise, P2 has a clock rate of 3 GHz and CPIs of 2, 2, 2 and 2

We next assume that we have a program that executes a million (10^6) instructions such that 10% of these instructions are of class A, 20% class B, 50% class C and 20% class D. Finally, the question: (8 points total)

- a) For our program, how many instructions of each class are executed? (2 points)

$$\text{Class A} = 10\% \text{ of } 10^6 = 100000$$

$$\text{Class B} = 200000$$

$$\text{Class C} = 500000$$

$$\text{Class D} = 200000$$

- b) Compute the execution time of the program using P1 and using P2. Express the answer to the nearest microsecond. (6 points ... 3 points each)

Execution time formula: number of CPI*class+next CPI*class... all the way to last

$$P1 \text{ cycles: } 1 * 100000 + 2 * 200000 + 3 * 500000 + 3 * 200000 = 2600000$$

$$\text{Execution time} = 2600000 / 2.5 * 10^9 = 1040\text{ms}$$

$$P2 \text{ cycles: } 2 * 100000 + 2 * 200000 + 2 * 500000 + 2 * 200000 = 2000000$$

$$\text{Execution time} = 2000000 / 3 * 10^9 = 670\text{ms}$$

10. •Do Exercise 1.10.1 on Page 57 in the textbook. As part of this problem, you will want to derive a formula for die area (see page 28 of the textbook; note that this will be an approximation). For both wafers, give your calculated value (use 3.14 for π) of die area (in cm^2 , two decimal places) and the value of yield to four decimal places. (8 points ... 2 points each)

INFO:

First Wafer:

Diameter of the wafer=15

Cost of wafer=12

Number of Dies=84

Defect=0.020 defect/ cm^2

Second Wafer:

Diameter of the wafer=20cm

Cost of wafer=15

Number of dies=100

Defect=0.031 defect/ cm^2

FORMULAS:

Cost per die= cost per wafer/(Dies per Wafer)*(yield)

Dies per wafer= wafer area/die area

Yield= $1/(1+(\text{defects per area} * \text{die area}/2))^2$

Radius of wafe= Diameter/2

Area of the wafer= $\text{PI} * \text{radius}^2$

Area of the die= Area of the Wafer/Dies per Wafer

SOLUTION:

First wafer:

Radius of wafer= $15/2=7.5\text{cm}$

Area of the wafer= 176.625 cm^2

Area of die= $176.625/84= 2.103$

Yield= 0.0959apx.

Second Wafer:

Radius of wafer= $20/2=10\text{cm}$

Area of the wafer= 314m^2

Area of die= $314/100= 3.14$

Yield= 0.909apx.

11. •Do Exercise 1.10.2 on Page 57 in the textbook. The problem does not specify the unit of cost (it is simply 12 and 15). Therefore, your answer will not have a specific unit either. Give answer to four decimal places. (4 points ... 2 points each)

INFO:

First Wafer:

Diameter of the wafer=15

Cost of wafer=12

Number of Dies=84

Defect=0.020 defect/cm²

Second Wafer:

Diameter of the wafer=20cm

Cost of wafer=15

Number of dies=100

Defect=0.031 defect/cm²

FORMULAS:

Cost per die= cost per wafer/(Dies per Wafer)*(yield)

SOLUTION:

First wafer:

Cost per die: $12/(84*0.959)=0.149$ apx.

Second wafer:

Cost per die: $15/(100*0.909)=0.165$ apx.

Your assignment is due by 11:59 PM (Eastern Time) on the assignment due date (consult Course Calendar on course website).