Jon Crawford

CDA3100

Assignment 1

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?

      Spim(parser) syntax error

   b) Does the program load and execute properly if you get rid of the “text” line completely?
      Yes
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? What value is in register 2?
      v0, a
   b) Register 15 goes by what other name? What value is in register 15?
      t7, 0
   c) Register 28 goes by what other name? What value is in register 28?
      gp, 10008000
   d) Register 31 goes by what other name? What value is in register 31?
      ra, 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? What instruction is this?

`00041080` **`sll $2, $4, 2`** *`$v0 $a0`* **`shift left logical`**

b)What value is stored at location 00400034? What instruction is this?

`3402000a` **`ori $2, $0, 10`** *`li $v0, 10`* **`load(or)immediate`**

`# this is the exit option`

c)What value is stored at location 00400038? What instruction is this?

`0000000c` **`syscall #Perform the system call`** d)What value is stored at location 80000184? What instruction is this?

```
3c019000  lui $1, -28672 sw $v0 s1 load upper imm.

# Not re-entrant and we can't trust
```

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  6f57206f  0a646c72  00000000  H e l l o   W o r l d . . . . .
```

b)What value is stored at location 90000160?

```
63614d5b  656e6968  65686320  005d6b63  [ M a c h i n e   c h e c k ] .
```

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. 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)

```
# Jonathan Crawford -- 01/22/18
# bio.s -- an introductory bio program
# Register use:
#       $v0     syscall parameter and return value
#       $a0     syscall parameter
        .text
        .globl main
main:
        la      $a0, msg1       # address of my name message
        li      $v0, 4          # this is the print_string option
        syscall         # perform the system call

        la      $a0, msg2
        li      $v0, 4          # this is the print_string option
        syscall         # perform the system call

        la      $a0, msg3
        li      $v0, 4          # this is the print_string option
        syscall         # perform the system call
```

```
	la	$a0, msg4
	li	$v0, 4		# this is the print_string option
	syscall		# perform the system call

	li	$v0, 10 # this is the exit option
	syscall		# perform the system call
# Here is the data for the program
	.data
msg1: .asciiz	"My name is Jon Crawford\n"
msg2: .asciiz	"I was born in Medina, Ohio, but grew up outside Detroit, Michigan\n"
msg3: .asciiz	"I am a factory design consultant who dreams of becoming a robotics engineer\n"
msg4: .asciiz	"I enjoy hiking with my schnauzer Ted, cooking bbq, and watching european soccer"
# end bio.s
```

7. Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024.

a) What is the minimum size in bytes of the frame buffer to store a frame?

1280x1024 = 1310720*(24/8) = 3932160bytes

b) How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?

3932160*8 = 31457280bits/100E6 = 0.3145728s = 315ms

8. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a) Which processor has the highest performance expressed in instructions per second?

P1 = 3GHz/1.5 = 2.000E9 inst/s, P2 = 2.5GHz/1 = 2.500E9 inst/s, P3 = 4GHz/2.2 = 1.818E9 inst/s

b)If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

P1*10 = 3E10 cycles,2E10 instr.

P2*10= 2.5E10 cycles, 2.5E10 instr.

P3*10 = 4E10 cycles, 1.818E10 instr.

c) We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

30% = .7Execution time = 7s

P1 CPI = 1.5*1.2 = 1.8, P2 CPI = 1*1.2 = 1.2, P3 CPI = 2.2*1.2 = 2.64

P1 = 2E10inst.*1.8/7 = 5.143GHz

P2 = 2.5E10*1.2/7 = 4.286GHz

P3 = 1.818E10*2.64/7 = 6.856GHz

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 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)

Class A = 10E5*.1 = 1E5 instr.

Class B = 10E5*.2 = 2E5 instr.

Class C = 10E5*.5 = 5E5 instr.

Class D = 10E5*.2 = 2E5 instr.

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)

P1 = (1E5*1 + 2E5*2 + 5E5*3 + 2E5*3) / 2.5E9 = .00104s =1040us

P2 = (1E5*2 + 2E5*2 + 5E5*2 + 2E5*2) / 3.0E9 = .000667s = 666.6 us

W1 = Assume a 15 cm diameter wafer has a cost of 12, contains 84 dies, and has 0.020 defects/cm2 .

W2 = Assume a 20 cm diameter wafer has a cost of 15, contains 100 dies, and has 0.031 defects/cm2 .

10. Find the yield for both wafers

W1 = A = 3.14(15/2)^2 = 176.625cm^2/84 =2.103cm^2

Yield W1 = 1/ (1+ .020*(2.103/2))^2 = 95.92%

W2 = A = 3.14(20/2)^2 = 314/100 =3.14cm^2

Yield W2 = 1/ (1+.031*(3.14/2))^2 = 90.93%

11. Find the cost per die for both wafers.

W1 = 12/84 * .9592 = 0.1370

W2 = 15/100 * .9093 = 0.1364