Toby Chappell

CPSC 250

October 5, 2019

Homework 2

Language of the Computer

1. Convert the following binary numbers to their decimal equivalents (5 pts):

a. 001100 b. 000011 c. 011100 d. 111100 e. 101010

1. 12
2. 3
3. 28
4. 60
5. 42
6. Convert the following decimal numbers to their binary equivalents (5 pts):

a. 64 b. 100 c. 111 d. 145 e. 255

1. 01000000
2. 01100100
3. 01101111
4. 10010001
5. 11111111
6. Convert the following hexadecimal numbers to their decimal equivalents (5 pts):
7. C b. 9F c. D52 d. 67E e. ABCD
8. 12
9. 159
10. 3410
11. 1662
12. 43981
13. List a couple disadvantages as well as advantages of assembly language compared to high-level languages. (4 pts)
14. Disadvantages:

* Simple expressions in higher level languages become far longer in assembly
* Harder to write in comparison
* Assembly program on one processor will not run on a different processor

1. Advantages:

* Same efficiency as machine code
* Each instruction is basic making it easier to debug code
* Does not require an interpreter or compiler to run

1. List and briefly define four different kinds of assembly language statements (8 pts).

* Arithmetic Operations: Preforms addition and subtraction (pulls from two sources and outputs to one destination)
* Logical Operations: Used for bitwise manipulation (can be used to extract/insert groups of bits in a word)
* Conditional Operations: Jumps to a labeled instruction if a condition is met, otherwise proceeds sequentially (ie beq, bnq, etc.)
* Byte/Halfword Operations: Use bitwise operations; can load and store

1. Write a MIPS assembly language equivalent for the following C code fragments (12 pts).

1. *if (a == b) c++; else c--;*

#$s0 = a, $s1 = b, $s2 i=c

*beq $s0, $s1, Plus*

*addi $s2, $s2, -1*

*j End*

*Plus: addi $s2, $s2, 1*

*End:*

1. *if ((a >= b) && (c > d)) e++; else e--;*

#$s0 = a, $s1 = b, $s3 = c, $s4 = d, $s5 = e

*beq $s0, $s1, Plus*

*slt $t0, $s1, $s0*

*beq $t0, 1, Plus*

*slt $t0, $s4, $s3*

*beq $t0, 1, Plus*

*addi $s5, $s5, -1*

*j End*

*Plus: addi $s5, $s5, 1*

*End:*

1. *while (c >= 0) d++;*

#$s0 = c, $s1 = d

*Loop: slt $t0, $s0, 0*

*beq $t0, 1, End*

*addi $s1, $s1, 1*

*j Loop*

*End:*

1. Consider the following C program: (6 pts)

/\* a simple C program to average 3 integers \*/

main ()

{

int avg;

int i1 = 20;

int i2 = 13;

int i3 = 82;

avg = (i1 + i2 + i3)/3;

}

Write a MIPS version of this program.

#$t0 = i1, $t1 = i2, $t2 = i3

#$s0 = values, $s1 = avg

.text

.globl main

main: la $s0, values

lw $t0, 0($s0)

lw $t1, 4($s0)

lw $t2, 8($s0)

add $s1, $t0, $t1

add $s1, $s1, $t2

div $s1, $s1, 3

li $v0, 10

syscall

.data

values: .word 20, 13, 82

1. Find the binary equivalent codes for the following MIPS program segment. (5 pts)

*addi $t1, $0, -12*

*back:   addi $t1, $t1, 4*

*bne $t1, $0, back*

001000 00000 01001 1111111111110100

001000 01001 01001 0000000000000100

000101 01001 00000 0000000000000001