Andy Alvarenga

CSE031

Lab 9

TPS Activity #1:

1. A machine code instruction contains 32 bits.
2. There are 3 types of formats: R, I, and J format. R: add, jr; I: addi, beq; J: j, jal
3. a.) Addi is in the I format and has 4 fields, which are named: opcode, rs, rt, and immediate.

b.) The opcode in hex is 0x8. $zero is the rs register and in hex, zero is represented as 0x00. $s0 is the rt register and in hex, $s0 is represented as 16 = 0x10. The value of immediate is 25 and in hex, 25 is 0x0019

c.) Hex: 0x20100019

Binary: 0010 0000 0001 0000 0000 0000 0001 1001

1. a.) Binary: 0000 0010 0011 0000 0100 0000 0010 1010

Machine Code,: 0x0230402A

b.) The instruction type is in the R format, you can tell by looking at the first 6 bits, which is the opcode. There are 6 fields: opcode, rs, rt, rd, shamt, funct.

c.) opcode: 0x0 rs: 0x11 rt: 0x10 rd: 0x08 shamt: 0x0 funct: 0x2a

d.) The instruction is: slt. To determine this, you combine the opcode and the funct. The mapping for the registers is: opcode, rs, rt, rd, shamt, funct

e.) slt $t0, $s1, $s0, yes it is the same as in source window.

1. a.) I format

b.) opcode: 0x05, rs: 0x08, rt: 0x00

c.) LESS, Address: 0x0040001c

d.) No, since bne is in the I-format and the address has 32 bits , it can not fit in the machine code.

e.) The immediate field is the number of lines from the next line to the label. The value is 1.

f.) Binary: 0001 0101 0000 0000 0000 0000 0000 0001

Hex: 0x15000001

Yes, it matches the code in the text segment.

1. a.) J format, there are only two fields.

b.) The opcode for j is: 0x2.

c.) It jumps to label GREQ, which is at the address 0x00400030.

d.) We can fit 26 bits in the address field. We do this by removing the first 4 bits and last 2 bits of the address in binary. We can do this because the 2 right-most bits are always 00, which allows us to shift the binary. The address field is: 0000 0100 0000 0000 0011 00.

e.) In binary: 0000 1000 0001 0000 0000 0000 0000 1100.

In hex: 0x0810000C

Yes, it is the same as the code in the text segment.

TPS Activity #2:

1. addi $s0, $zero, -15 (I-format)

opcode: addi = 0x8 = 001000

rs: $zero = 0x0 = 00000

rt: $s0 = 16 = 10000

immediate: -15 = 0000 0000 0000 1111 🡪 2’s compliment = 1111 1111 1111 0001

In binary: 0010 0000 0001 0000 1111 1111 1111 0001

In hex (machine code): 0x2010FFF1

Starting from: 0x2008FFF1

Convert to binary: 0010 0000 0001 0000 1111 1111 1111 0001

Check first 6 bits: 001000 = addi, therefore we know it’s in I-format.

Rs: 00000 = $zero

Rt: 10000 = $s0

Immediate: 1111 1111 1111 0001 🡪 2’s compliment = 0000 0000 0000 1111 = -15

Therefore, addi $s0, $zero, -15

1. slt $t0, $s0, $s1 (R-format)

opcode: slt = 0x0 = 000000

rs: $s0 = 16 = 10000

rt: $s1 = 17 = 10001

rd: $t0 = 8 = 01000

shamt: 0x0 = 00000

funct: 0x2a = 101010

In binary: 0000 0010 0001 0001 0100 0000 0010 1010

In hex (machine code): 0x0211402A

Starting from: 0x0211402A

Convert to binary: 0000 0010 0001 0001 0100 0000 0010 1010

Check first 6 bits: 000000, Since they equal 0, check last 6 bits = 101010 = slt, therefore R-format

Rs: 10000 = 16 = $s0

Rt: 10001 = 17 = $s1

Rd: 01000 = 8 = $t0

Shamt: 00000 = 0

Therefore, slt $t0, $s0, $s1

1. beq $t0, $zero, LEEQ (I-format)

opcode: beq = 0x4 = 000100

rs: $t0 = 8 = 01000

rt: $zero = 0x0 = 00000

immediate: LEEQ = 6

6 = 0000 0000 0000 0110

In this case we do not perform 2’s compliment on negative binaries.

In binary: 0001 0001 0000 0000 0000 0000 0000 0110

In hex (machine code): 0x11000006

Starting from: 0x11000006

Convert to binary: 0001 0001 0000 0000 0000 0000 0000 0110

Check first 6 bits: 000100 = 0x4 = beq, therefore I-format

Rs: 01000 = 8 = $t0

Rt: 00000 = $zero

Immediate: 0000 0000 0000 0110 = 6

Therefore, beq $t0, $zero, LEEQ

1. j GRT (J-format)

opcode: j = 0x2 = 000010

address: GRT = 0x0040001C = 0000 0000 0100 0000 0000 0000 0001 1100

Remove the 4 left-most and 2 right-most bits of the address:

0000 0100 0000 0000 0000 0001 11

In binary: 0000 1000 0001 0000 0000 0000 0000 0111

In hex (machine code): 0x08100007

Starting from: 0x08100007

Convert to binary: 0000 1000 0001 0000 0000 0000 0000 0111

Check first 6 bits: 000010 = 0x2 = j, therefore J-format

Address: 00 0001 0000 0000 0000 0000 0111

Add 4 bits to the left and 2 to the right of the address:

0000 0000 0100 0000 0000 0000 0001 1100

Convert address into Hex: 0x0040001C

Therefore, j 0x0040001C