Emmanuel Vargas

CSE-140

February 4, 2021

Homework 1

Exercise

1. Partners: Vincent Lee Stanford
2. Downloaded the sheet.
3. Loaded proc1.s into MARS.
4. Studied text segment.
5. A machine instruction contains 32 bits.
6. There are three types of instructions, R, J and I type instructions.

R format: add, jr

J format: j, jal

I format: addi, beq

1. .
   1. Slt is an R type instruction. It has 6 fields: Opcode, rs, rt, rd, shamt, and funct.
   2. Opcode of slt in hex: 0x0. Register rs is register 17 in hex 0x11. Register rt is 16 in hex 0x10. Register rd is 8 in hex 0x08. The function field of this instruction in hex is 0x42.
   3. Machine code of line 14 in binary and hex:

000000 10001 10000 01000 00000 101010 (R spacing)

0000 0010 0011 0000 0100 0000 0010 1010 (hex spacing)

0x0230402A (hex)

1. .
   1. The code at this address is 0x34240000

In binary 0011 0100 0010 0100 0000 0000 0000 0000

* 1. This is an I type instruction, we can tell because of its opcode, there are 4 fields in this type of instruction: opcode, rs, rt, and immediate.
  2. Opcode: 0x0D

Rs: 0x01

Rt: 0x04

Immediate: 0x0000

* 1. This instruction is ori, we can tell by looking up the opcode value, the mapping of registers used in this instruction is as follows ori rt, rs, immediate.
  2. Final instruction:

ori $4, $1, 0x0000 – Matches source code from text window.

1. .
   1. This is an I format instruction.
   2. Opcode: 0x5

Rs: 0x8

Rt: 0x0

* 1. The name of the label is LESS, the address of this label is: 0x0040001C.
  2. We cannot put the address directly into the immediate field as there are only 16 bits available and the address is 32 bits total.
  3. We find the value of the immediate field by counting how many instructions after the next line of the bne we are branching to, in this case 1.
  4. Machine instruction in binary:

0001 0101 0000 0000 0000 0000 0000 0001

Machine instruction in hex:

0x15000001 (matches the text segment code)

1. .
   1. This is a J type instruction, there are two fields.
   2. Opcode: 0x2
   3. This label jumps to label GREQ at address 0x00400030.
   4. We can use 26 bits in the address field. We can squeeze the address into the field by excluding the last 2 bits of the address and by using the first 4 bits of the current address instead of storing those bits. We can ignore the last two bits because they are always 00, because of how the addresses are spaced out they always end in 00 so those bits don’t need to be stored in the instruction.

Address field in binary: 0000 0100 0000 0000 0000 0011 00

* 1. Machine code in binary:

0000 1000 0001 0000 0000 0000 0000 1100

Machine code in hex:

0x0810000C (matches text segment code)

Individual Assignment

1. addu $17, $0, $2

R type

Opcode: 000000

Rs: 00000

Rt: 00010

Rd: 10001

Shamt: 00000

Funct: 100001

Code binary:

0000 0000 0000 0010 1000 1000 0010 0001

Code hex:

0x00028821

0000 0000 0000 0010 1000 1000 0010 0001

Opcode: 000000 – function

Rs: 00000 - $0

Rt: 00010 - $2

Rd: 10001 - $17

Shamt: 00000 - none

Funct: 100001 – addu – R type

MIPS:

addu $17, $0, $2

1. beq $8, $0, 0x00000006

I type.

Opcode: 000100

Rs: 01000

Rt: 00000

Immediate: 0000 0000 0000 0110

Code binary:

0001 0001 0000 0000 0000 0000 0000 0110

Code hex:

0x11000006

0001 0001 0000 0000 0000 0000 0000 0110

Opcode: 000100 – beq I type

Rs: 01000 - $8

Rt: 00000 - $0

Immediate: 0000 0000 0000 0110 - 6

MIPS:

beq $8, $0, 6

1. j END

J type

Opcode: 000010

Current address: 0x0040002C

Label address: 0x00400044 – 0000 0000 0100 0000 0000 0000 0100 0100

Address field: 0000 0100 0000 0000 0000 0100 01

Code binary:

0000 1000 0001 0000 0000 0000 0001 0001

Code hex:

0x08100011

0000 1000 0001 0000 0000 0000 0001 0001

Opcode: 000010 – j type – jump

Address field: 0000 0100 0000 0000 0000 0100 01

Current address: 0x0040002C

Label address: 0000 0000 0100 0000 0000 0000 0100 0100 – 0x00400044

Label: END

MIPS:

j END

1. lui $1, 0x00001001

I type

Opcode: 001111

Rs: 00000

Rt: 00001

Immediate: 0001 0000 0000 0001

Code binary:

0011 1100 0000 0001 0001 0000 0000 0001

Code hex:

0x3C011001

0011 1100 0000 0001 0001 0000 0000 0001

Opcode: 001111 – lui I type

Rs: 00000 - none

Rt: 00001 - $1

Immediate: 0001 0000 0000 0001 – 0x1001

MIPS:

lui $1, 0x1001