Cody Martin

18-447 Homework 2

1. See microcode.csv

2. a)

Program A on Machine X:

ADD A, A, 1

ADD B, B, 2

ADD C, C, 3

ADD D, D, 4

Program B on Machine X:

ADD B, B, A

ADD C, B, A

SUB A, B, C

SUB D, A, C

Program A on Machine Y:

LD R1, A

ADD R1, R1, 1

ST R1, A

LD R1, B

ADD R1, R1, 2

ST R1, B

LD R1, C

ADD R1, R1, 3

ST R1, C

LD R1, D

ADD R1, R1, 4

ST R1, D

Program B on Machine Y:

LD R1, A

LD R2, B

ADD R2, R2, R1

ST R2, B

ADD R2, R2, R1

ST R2, C

LD R1, B

SUB R3, R1, R2

SUB R1, R3, R2

ST R3, A

ST R1, D

b) Machine X with Code A: 28 bytes of instructions, 8\*4 = 32 bytes of data transferred. Total of 60 bytes.

Machine X with Code B: 28 bytes of instructions, 12\*4 = 48 bytes of data transferred. Total of 76 bytes.

Machine Y with Code A: 48 bytes of instructions, 32 bytes of data transferred. Total of 80 bytes.

Machine Y with Code B: 44 bytes of instructions, 28 bytes of data transferred. Total of 72 bytes.

c) No, it does not. Immediate operands will favor Machine X because you don’t need to store and load the data like you do with Machine Y. Other operands favor Machine Y because instead of continuously pulling data from memory as with Machine X, Machine Y can use smaller registers to do the operations.

3. a) For ISA 1: 16 bytes

For ISA 2: 13 bytes

b) For ISA 1: 7 cycles

For ISA 2: 15 cycles

c) ISA 2 has the lower code size because it does not waste space for the third operand if it is not needed, whereas ISA 1 does.

d) ISA 1 has the lower execution time because it does not take the extra time to do the extra decoding that ISA 2 has to do with its optional operand.

4. For the first program, Auto increment would be the best because the array elements are each a byte, so the auto increment will increment the memory location by 1 byte each time. For the second program, Scale indexed would be best because each element is 4 bytes, so Auto increment won’t work well, and scale indexed will work well because the assembler can just keep i in a register and index easily that way. For the third program, register indirect would be the best because it would be one instruction taking the value in p and fetching the value at that memory location. For the fourth program, memory indirect would be best because you have to grab the address of the pointer to the memory, so going into memory to get the address of that pointer, then accessing the value pointed at by that pointer would all be done in one instruction.

5. a) (i) you need to be able to address 5.3688\*10^8 bits. This can be accomplished with 28 bits.

(ii) you need to be able to address 6.711\*10^7 bytes. This can be accomplished with 25 bits.

(iii) you need to be able to address 8.389\*10^6 8-byte words. This can be accomplished with 22 bits.

(iv) you need to be able to address 2097152 32-byte words. This can be accomplished with 20 bits.

b) NOTE: This code assumes the offset is given in nibbles.

LW $2, 0($5)

ADDIU $3, $zero, 4

MULT $4, $3

MFLO $3

SRL $2, $2, $3

LW $3, 0($7)

ADDIU $4, $zero, 4

MULT $6, $4

MFLO $4

SRL $3, $3, $4

ADD $2, $2, $3

6. a) The Microarchitecture is essentially the way instructions are executed on a machine, whereas the ISA is the convention for the instructions. Each instruction of the ISA can have multiple states or a single state in the Microarchitecture. The ISA manipulates the Microarchitecture in a user-friendly manner. A compiler needs to know how many cycles each instruction takes so that it can properly optimize the program. Also, a compiler should know how much memory is in the microarchitecture.

b) (i) ISA

(ii) Microarchitecture

(iii) Microarchitecture

(iv) ISA

(v) Microarchitecture

(vi) ISA

(vii) Microarchitecture

(viii) Microarchitecture

7.

