9/10

good

CS220: Computer Organization
Quiz#2

Name: Sahil Dhull
Roll No.: 160607

1. Consider a single-channel DIMM card with a 128-bit channel. The channel has two ranks. The DIMM card uses x8 chips and each chip has sixteen banks. If each bank has 8192 rows and 1024 columns, what is the total capacity of the DIMM card in bits? **(2 points)**

128-bit channel & x 8 chips $\Rightarrow \frac{128}{8} = 16$ chips in a rank.

$\Rightarrow$ Total 32 chips. $\Rightarrow$ Total 16×32 banks $= 2^9$ banks

Each bank $\rightarrow$ 8192 rows x 1024 columns $= 2^{13} \times 2^{10}$ bits

$= 2^{23}$ bits

$\Rightarrow$ Total capacity $= 2^9 \times 2^{23}$ bits

$= \underline{2^{32} \text{ bits}}$

$2^5 \times 2^4 \times 2^{13} \times 2^{10}$

2. Suppose the following five access requests have come to a particular bank of a DRAM module, where each request is listed as (row number, column number): (19, 5), (7, 5), (11, 11), (19, 2), (11, 2). Show the sequence in which the requsts must be sent to the bank such that it takes the minimum amount of time to complete all the requests. **(2 points)**

(7,5) , (11,2) , (11,11) , (19,2) , (19,5)

The request must be sent in this sequence because changing a row ~~several~~ requires much more time.

2

3. Write down the two's complement binary representation of the decimal integer -79 assuming that twelve bits are used for the representation. **(2 points)**

$79 = 1001111$

2's complement of $(-x) = (2^n - 1) - x + 1$

79 (12 bit representation) = 000001001111

$\Rightarrow$ −79 (in 2's complement) = 111110110000 + 1

= 111110110001

2

Rough

64 = 1000000

15 = 1111

79 = 1001111

4. Write down the binary representation of the decimal fraction 14.6875 in normalized scientific notation. **(4 points)**

$14.6875_{10} = 1110_2 + 0.6875_{10}$

$0.6875_{10} = 1.375_{10} \times 2^{-1} = 2.75_{10} \times 2^{-2}$

$= 5.5_{10} \times 2^{-3}$

$= 11_{10} \times 2^{-4}$

$= 1011_2 \times 2^{-4}$

$\Rightarrow 14.6875_{10} = 1110 + 1011 \times 2^{-4}$

$= 1110 + 0.1011$

$= 1110.1011$

$14.6875_{10} = 1.1101011 \times 2^3$

4

Rough

14 = 1110

0.6 = $\frac{6}{10}$ $\frac{3}{5}$

0.5 + 0.1 $\quad 1 \times 10^{-1}$

0.1

.6875

1.3750

1.375×2^{-1}

$2.750 \quad 2^{-2}$

5.50

11

1011

101.1×2^{-3}

10.11

1.011

2/10

CS220: Computer Organization
Quiz#3

Name: Sahil Dhull
Roll No.: 160607

General instructions: In all the questions, you will assume 32-bit big-endian MIPS ISA.

1. Consider translating the following for loop into MIPS. Assume that `i` and `N` allocated in registers \$t0 and \$t1, respectively.

```
for (i=0; i<N; i++) { loop body }
```

The skeleton of the MIPS translation is shown below. The `bne` instruction starts at address 0x00603000. What is the minimum possible address of the first instruction of the loop situated at the label `start`? Express your answer in hexadecimal. **(2 points)**

```
        ...
start:  ...
        loop body
        ...
        slt $t2, $t0, $t1
        bne $t2, $0, start
```

bne has 16 bit immediate ~~and it is sign extended~~. and it is sign extended.

so, 0x8000 will be 0xffff8000 ~~which is most~~

adding 0x00603000 to 0xffff8000

0x00603000
0xffff8000
0x005fb000

So, minimum possible address of first instruction is 0x005fb000

2. Consider the following MIPS instruction sequence. What is the final hexadecimal value in \$t0? **(2 points)**

```
lui  $t0, 0x62
addi $t0, $t0, 0xaabc
sra  $t0, $t0, 0x4
```

0x 61aab is stored in \$t0 finally.

after lui → {0062, 0000}
+ ffff aabc
after addi → 0061aabc
after sra → 00061aab

2

Rough
f|10
1111
0101
0110 0010
1111 1111
0110 0001
00603000
ffff - -
100f
00 (11)
5(6)
~~00000001~~
~~0000~~
8000
603000
ffff8000

3. Consider the following MIPS instruction sequence. Assume that initially $t0 contains 0x10000000 and $t1 contains 0x10000004. Initially, the word stored at address 0x10000000 is 0x12fe43ba and the word stored at address 0x10000004 is 0xab34ef21. What is the final hexadecimal value of the word stored at address 0x10000004? **(3 points)**

```
lb $t0, 1($t0)
sh $t0, 0($t1)
```

addr 0x10000000 → 0x12fe43ba → 32bit

~~0x100000~~.

lb $t0, 1($t0) ~~gives~~ stores 0xfe in $t0.

sh $t0, 0($t1) stores 0x00fe (16bits) in address 0x10000004

So, word at address 0x10000004 is

0x00feef21

4. Consider translating the following C statement where the value of `label` is 0x0 and `label1` is 2^{28} instructions away. This information is available at the time of compilation of the statement. Show the MIPS translation of this C statement using minimum number of instructions. **(3 points)**

```
label: goto label1
```

~~label : goto labelx~~

~~labelx: goto label1~~

~~label1:~~

0x0 label: j labelx — 2^{26} instruction

labelx: j labely — 2^{26} instruction

labely: j label z — 2^{26} instruction

labelz: j label1 — 2^{26} instructions

label1

∵ j has 26 bit target, ie max ~~2^{28}~~ 2^{28} bytes can be traversed [∵ lower 2bits are 0 ie << 2]

but each instruction is 4 bytes ⇒ 2^{26} instruction can be traversed.

Moreover ~~H~~ 4 MSB of PC are 0 in all instruction addresses

Rough

26 bits 28

28 bits

4 bits PC = 0

$2^{28}-1$

Scanned by CamScanner

3.5/10

CS220: Computer Organization
Quiz#4

Name: Sahil Dhull
Roll No.: 160607

1. Suppose Booth's algorithm is used in a multiplication where the multiplicand and the multiplier are represented in two's complement and their respective values are 0xabcdef01 and 0xcdef01ab. Count the number of addition and subtraction operations. **(1+1 points)**

~~Multiplier → Both multiplier and multiplicand are -ve. So, we first convert them to their +ve absolute value.~~

Multiplier → 1100 1101 1110 1111 0000 0001 1010 1011

No. of addition = No. of times transition from 1 to 0. (lower to higher)
= 8

No. of subtraction = No. of transition from 0 to 1 (lower to higher)
= 7

Rough
8 → 1000
9 → 1001
a → 1010
b → 1011
c → 1100
d → 1101
e → 1110
f → 1111

2. Consider a program that has 15% load/store instructions, 25% conditional branch instructions, 10% other types of control transfer instructions, and 50% arithmetic and logic instructions. The program is executed on a processor with average CPI of load/store 10, of conditional branch 4, of other types of control transfer instructions 3, and of arithmetic and logic instructions 2. Rank these four categories of instructions from most important to least important for optimizing the overall performance of the program. Assume that the clock frequency of the processor which the program runs on is kept constant during the optimization process. **(2 points)**

Arithmetic & Logic ~~of~~ instruction
Conditional Branch Instruction
Load/Store Instruction
Control Transfer.

0.5

∵ Using Amdahl's law, Optimization of part of program that is most common is best.

3. Suppose the variables x, y, z are of signed type of length 32 bits and we would like to compute $z = x - y$. If x is 0xffff0abc, what is the range of permissible values of y so that no overflow occurs in the subtraction used to compute z? Express the upper and lower bounds in hexadecimal of length 32 bits. **(3 points)**

x → 0x ffff0abc

Max value:

let y = 0x7 ffff fff , in this case overflow occurs.

But if there is at least one 1 in upper 16 bits of 2's complement of y , then overflow will not occur

⇒ y = 0x7ffe ffff is upper bound.

Lower bound:

0x ffff ffff is lower bound.

Rough

7ffE ffff

8001 0001

4. Suppose 1001010 (in binary) is divided by 1001 (in binary). We would like to calculate the number of additions and subtractions when using the restoring, non-performing restoring, and non-restoring algorithms. Fill out the six entries in the table below. **(0.5×6 points)**

Table 1. Count of additions and subtractions in division algorithms

Algorithm	Number of additions	Number of subtractions
Restoring	3	4
Non-performing restoring	0	4
Non-restoring	3	2

3

Rough

1001010

1001

74

9

Scanned by CamScanner