CS220 Quiz#5

General instructions: Please write brief explanation for your answers. If you submit multiple times, your last submission will be used for grading. Please provide an email address below where your responses can be sent.

Email address *

sohamg@iitk.ac.in

Your name *

Soham Ghosal

5/6

Good!

Your roll number *

180771

Q1. Consider a program that has 15% load/store instructions, 15% conditional branch instructions, 10% other types of control transfer instructions, and 60% arithmetic and logic instructions. The program is executed on a processor with average load/store CPI of 7, conditional branch CPI of 4, other types of control transfer instructions CPI of 3, and arithmetic and logic instructions CPI of 2. Suppose the implementation of only one of the aforementioned four categories of instructions could be optimized to bring the CPI of that category down to 1 while keeping everything else unchanged. What is the maximum speedup achievable by this optimization? [2 points]

Let no of instructions be N
Let cycle time be t
Base time= 1.05Nt+0.6Nt+0.3Nt+1.2Nt = 3.15Nt

If we bring down cpi of ALU instructions:
New time= 3.15Nt-0.6Nt=2.55Nt
If we bring down cpi of L/S instructions:
New time= 2.25Nt
If we bring down cpi of Branch instructions:
New time=2.7Nt
If we bring down cpi of other instructions:
New time=2.95Nt

2

Therefore, we should bring down cpi of L/S instructions to 1.

Maximum speedup=3.15/2.25= 1.4

40% speedup.

Q2. A certain portion P of a program has been optimized such that the execution time of that portion has become one-fourth of the original time this portion used to take. The execution time of P after the optimization is one-third of the total post-optimization execution time of the program. What is the overall speedup enjoyed by the program due to this optimization? [1 point]

Let x be execution time of the portion in question, and y be the time taken to execute the remaining part.
Total program execution time=x+y
New time for portion P=x/4
The other time has not changed.
Therefore, x/4=(1/3)(x/4+y)

Therefore, x=2y

1

Total speedup= (x+y)/((x/4)+y) = 3y/(3y/2) = 2

100% speedup.

Q3. 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 0xcddaabbc and 0xddaabbcc. Count the number of addition and subtraction operations. [2 points]

multiplicand = 11001101110110101010101110111100
multiplier = 1101 1101 1010 1010 1011 1011 1100 1100
The number of addition and subtraction operations will only depend on the multiplier.
We also need to append a zero in the least significant side.

Multiplier= 1101 1101 1010 1010 1011 1011 1100 1100 0
Number of additions=Number of transitions from 1 to 0
Number of subtractions=Number of transitions from 0 to 1

2

Number of additions=9
Number of subtractions=10

Q4. By inspecting the quotient of an unsigned division it is possible to infer the sequence of subtractions and additions that would have taken place if the division was done using the non-restoring division algorithm. Calculate the number of addition and subtraction operations if the quotient is 1110. [1 point]

The division must have taken 4 iterations, since the quotient has 4 bits.
The number of times the addition path was taken has to be the number of zeroes in the quotient(excluding the extra iteration). This is 1 in this case. For the other 3 iterations, we must have subtracted, since the remainder would have been positive and therefore we have a 1 in the quotient(the LSB hadn't been xor-ed)

Thus, without counting the extra iteration(if required), we have 1 addition and 3 subtractions,
Now, the last bit of quotient is 0, which means that the last remainder was negative, and therefore we had to add it with the current divisor(which was required divisor/2, since the next iteration has started). This implies that the remainder was still negative, and therefore the extra iteration IS required, where we have to perform one more addition.

In total, we have 2 additions and 3 subtractions. 0