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 *

palad@iitk.ac.in

Your name *

Adarsh Pal

6/6
Excellent!

Your roll number *

180032

Q1. Consider a program that has 15% load/store instructions, 25% conditional branch instructions, 20% other types of control transfer instructions, and 40% arithmetic and logic instructions. The program is executed on a processor with average load/store CPI of 6, conditional branch CPI of 3, other types of control transfer instructions CPI of 5, 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]

Suppose total no of instructions is N.
Initially,
total No of cycles reqd by:
load/store instructions= 0.15*6*N=.9N
conditional branch instructions=0.25*3*N=.75N
other types of control transfer instructions=0.20*5*N=N
ALU Instructions=0.4*2=.8N
Total cycles=(.9+.75+1+.8)N=3.45N

Total cycles if CPI of .... is reduced to 1:
load/store instructions= (3.45-.9+.15)N=2.7N
conditional branch instructions=(3.45-.75+.25)N=2.95N
other types of control transfer instructions=(3.45-1+.2)N=2.65N
ALU Instructions=(3.45-.8+.4)N=3.05N

So, maximum speedup achieved=3.45/2.65=1.30 (approx.)

2

Q2. A certain portion P of a program has been optimized such that the execution time of that portion has become one-third 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]

Suppose x=fraction of time taken initially by the program
and total Time taken initially=t
Final time taken=xt/3+(1-x)*t
So, xt/3=1/3*(xt/3+(1-x)*t)
So, x comes out to be 3/5=0.6
So, Speedup=t/(xt/3+(1-x)*t)=1/(x/3+(1-x))=1/(.2+.4)=5/3=1.67

1

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 0xdaabbccd and 0xaabbccdd. Count the number of addition and subtraction operations. [2 points]

The multiplier is B=0xaabbccdd=1010 1010 1011 1011 1100 1100 1101 1101
The total number of addition and subtraction operations depends upon the total i's such that i=0,1,...,32 and B[i] is not the same as B[i-1] where B[0] is LSB of B, with B[-1]=0
So, for 1010 1010 1011 1011 1100 1100 1101 1101 and 0 (B[-1]) we have total reqd i's = 0, 1, 2, 5, 6, 8, 10, 12, 14, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31. Total=21
So, the number of addition and subtraction operations comes out to be 21.
Individually, the number of addition operations is: 10 (B[i] is 0 while B[i-1] was 1)
Individually, the number of subtraction operations is: 11 (B[i] is 1 while B[i-1] was 0)

2

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 10001. [1 point]

The steps that would have taken place were:
Initially,
R=Dividend
R>0 is true
Q=1
(subtraction)
R=R-Divisor
Divisor=Divisor>>1
R>0 is true
Q=11
(subtraction, since remainder was >0)
R=R-Divisor
Divisor=Divisor>>1
R>0 is false
R=R+Divisor (addition)
Q=101
Divisor=Divisor>>1
R>0 is false
R=R+Divisor (addition)
Q=1001
Divisor=Divisor>>1
R>0 is false
R=R+Divisor (addition)
Q=10001
Stop
The total number of subtraction operations done is 2
and addition operations is 3

1

This content is neither created nor endorsed by Google.

Google Forms