Philip Lenox, Devlin Junker
November 25, 2013
CS 472 HW 5

(6.5) The time taken by Machines A, B and C to exectue a given task is:
A 16m, 9s : 9s / 16m = 0.5625
B 14m, 12s : 12s / 14m = 0.8571
C 12m, 47s : 47s / 12m = 3.916
relative to A:
A = 1 A
B = A/B = 0.6563 A
C = C/A = 0.1426 A

(6.6) Clock rate is a poor metric for performance because clock rates can be clocked arbitrarily quickly, but if it takes more than 1 clock cycle to execute an instruction, the performance can go down while increasing the clock rate. Clock speed, if correctly chosen for the processor, can indicate the speed with which one instruction will complete, so prior to multi-core and super scalar processors, it was an ok metric for performance of a processor.

(6.12) Would lowering the clock frequency by 15% be a good idea?

To answer this question, we will evaluate the performance as is, and with the 15% reduction
As is: $1(.45) + 3(.2) + 2(.1) + 2(.25) = 1.75$
15% reduction: $1(1(.45) + 2(.2) + 2(.1) + 2(.25))/(.85) = 1.823$
No, not worth it

(6.13) Determine cyles per conditional branch for 20% improvement

current performance: $1(.65) + 5(.1) + 2(.05) + 8(.2) = 2.85$
Target performance: current performance * 0.8 = 2.28
2.28 - [1(.65) + 5(.1) + 2(.05)] = x(.2)
x = 5.15
target cycles per conditional branch = 5

(6.17) $S = \frac{1}{f_s + \frac{1-f_s}{P}}$

(a) 10 processors , $f_s = 0.1 : P = 10 : S = \frac{1}{0.1+\frac{1-0.1}{10}} = 5.26$

(b) 100 processors , $f_s = 0.1 : P = 100 : S = \frac{1}{0.1+\frac{1-0.1}{100}} = 9.17$

(c) 5 processors , $f_s = 0.4 : P = 5 : S = \frac{1}{0.4+\frac{1-0.4}{5}} = 1.92$

(d) 100 processors , $f_s = 0.01 : P = 100 : S = \frac{1}{0.01+\frac{1-0.01}{100}} = 50.25$

(6.18) The optimal number of cores to purchase is 32. Purchasing additional processors increase's the ratio by less than 5% of the original speed. This would constitute a 306% increase in the cost.

Number of Processors	Speedup Ratio
1	1.00
2	1.82
3	2.50
4	3.08
5	3.57
10	5.26
20	6.90
30	7.69
31	7.75
32	7.80

(6.22) $\frac{1}{(1-x)+\frac{x}{3.5}} = 1.5$

$x = 47\%$ of the code must be using the string processing instructions.

(7.12) The memory, registers and ALU micro-operations could be performed for other or the next instructions while the conditional branch is being performed.

(7.15) Another adder would need to be added after the ALU_MPLX before the ALU. The adder would have to be conected to the IR to recieve the shift operand.