Problem Set–5 Solutions
CS 230, Autumn 2023

1.a) CPI = $\sum^{3}_{i=0} CPI_i * p_i$

MIPS = $f_{CPE}/(CPI*10^6)$

Computer R1:

CPI = $\sum^{3}_{i=1} CPI_i * p_i = 0.16 * 6 + 0.1 * 8 + 0.08 * 10 + 0.66 * 3 = 4.54$

Computer R1 needs an average of 4.54 clock periods for one instruction

MIPS = $f_{CPE} / CPI*10^6 = 400*10^6 / 4.54*10^6 = 88.1$

Computer R1 executes an average of 88 100 000 instructions per second

Computer R2:

CPI = $\sum CPI_i * p_i = 0.16 * 20 + 0.1 * 32 + 0.08 * 66 + 0.66 * 3 = 13.66$ Computer R2 needs an average of 13.66 clock periods for one instruction MIPS = f_{CPE} /CPI$*10^6$ = $400*10^6 / 13.66*10^6 = 29.28$

Computer R2 executes an average of 29 280 000 instructions per second.

1.b) $\boldsymbol{CPU_{time}}$ = Number_of_instructions /$\boldsymbol{MIPS*10^6}$

Another form of the equation to calculate the CPU time is:

CPUtime = Number_of_instructions * CPI * t_{CPU}

Computer R1:

CPUtime = Number_of_instructions /$\boldsymbol{MIPS * 10^6}$ = $12000 / 88.1 * 10^6 = 136.2 * 10^{-6}$ = = $136.2\ \mu s$

Computer R2: ***CPU***time = Number_of_instructions $\boldsymbol{MIPS * 10^6}$ = $12000 / 29.28 * 10^6$ = $410 * 10{-}6 = 410\ \mu s$

2. CPU performance$_B$/CPU performance$_A$ = CPU time$_A$/CPU time$_B$

- 6 = 3/CPU time$_B$
  User CPU Time = .5 seconds
  Since the I/O time is unaffected by the performance increase, it still takes 1 second to do I/O. Therefore it takes 1 + .5 = 1.5 seconds to run Program A on the faster CPU
  Wallclock Time = 1.5 seconds
  System Performance in MFLOPS = Number of Floating Point Operations * 10^6/Wallclock Time
  Old System Performance (10) = #FLOP * $10^6/4$
- #FLOP = 40 * 10^6
  New System Performance = 40 * $10^6/1.5$
  MFLOP/sec = 26.667

3. We have the instruction count: 10^9 instructions. The clock time can be computed quickly from the clock rate to be 0.5×10^{-9} seconds. So we only need to to compute clocks per instruction as an effective value:

Value	Frequency	Product
3	0.5	1.5
4	0.3	1.2
5	0.2	1.0

CPI = 3.7

Then we have,

Execution Time = $1.0*10^9*3.7*0.5*10^{-9}$ = 1.85 sec

4)

If we say that there are 100 instructions, then:

30 of them will be loads and stores.
50 of them will be arithmetic instructions.
20 of them will be all others.
(30 * 6) + (50 * 4) + (20 * 3) = 440 cycles/100 instructions
Therefore, there are 4.4 Cycles per instruction.

5)

Given that 80% of 10^9 instructions require single cycle i.e. no conditional branching & for 20% an extra cycle required.

Time taken by 1 cycle = 10^{-9} sec

Total time=$10^{-9}(80/100 * 10^9 + 20/100 *2 *10^9)$

= $10^{-9} *10^9(4/5 + 2/5)=6/5=1.2$ seconds