113 Architecture 2020 Exam Sample Solution

Disclaimer: This is not an official answer key, so please correct any mistake if you find one. There are more than one possible solution for some questions, so keep in mind about that!

1. **a.** $\text{MIPS} = \frac{\#\text{ of instructions}}{\text{CPI}\times\#\text{ of instructions}\times\text{cycle time}} \times \frac{1}{1{,}000{,}000} = \frac{1}{\text{CPI}\times\text{cycle time}\times 1{,}000{,}000}$

   b. We first calculate the ratio of cycle count for P1 and P2.

$$\begin{aligned}\text{cycle count} &= \text{CPI} \times \#\text{ of instructions} \\ &= \frac{\#\text{ of instructions}}{\text{MIPS} \times \text{cycle time} \times 1{,}000{,}000} && (\text{From 1a}) \\ &= \frac{\#\text{ of instructions} \times \text{clock rate}}{\text{MIPS} \times 1{,}000{,}000} && (\text{cycle time} = \frac{1}{\text{clock rate}})\end{aligned}$$

$$\begin{aligned}\frac{\text{cycle count P1}}{\text{cycle count P2}} &= \frac{\frac{\#\text{ of instructions P1}\times\text{clock rate P1}}{\text{MIPS P1}\times 1{,}000{,}000}}{\frac{\#\text{ of instructions P2}\times\text{clock rate P2}}{\text{MIPS P2}\times 1{,}000{,}000}} \\ &= \frac{r \times \#\text{ of instructions P2} \times p \times \text{clock rate P2} \times \text{MIPS P2}}{\#\text{ of instructions P2} \times \text{clock rate P2} \times q \times \text{MIPS P2}} \\ &= \frac{rp}{q}\end{aligned}$$

   Then we calculate the ratio of execution time:

$$\begin{aligned}\frac{\text{exec. time P1}}{\text{exec. time P2}} &= \frac{\text{cycle count P1} \times \text{cycle time P1}}{\text{cycle count P2} \times \text{cycle time P2}} \\ &= \frac{rp}{q} \times \frac{\text{clock rate P2}}{\text{clock rate P1}} \\ &= \frac{rp}{q} \times \frac{1}{p} \\ &= \frac{r}{q}\end{aligned}$$

   If $r > q$, then P2 is faster by $\frac{r}{q}$. If $r < q$, then P1 is faster by $\frac{q}{r}$.

   If $r = q$, then both P1 and P2 are about the same speed.

c. Not necessarily. Different instructions have different cycle counts. In the calculation for MIPS, CPI stands for **average** cycle per instruction. If floating point operations have a higher cycle count than this average, the execution time for the floating point operations will be longer. P4 might be slow in general cases, but good at floating-point operations. In this scenario, P3 might have a lower MFLOP rating than P4 despite its higher MIPS rating.

2. **a. i)** Below is a possible solution to the assembly code:

```
long foo(long x, long n, long limit) {
  long mask;
  long result = 0;
  for (mask = 1; mask < limit; mask = mask << n) {
    result = result | (x & mask);
  }
  return result;
}
```

ii) Below is the sample solution

Expression	Type	Operant Value	Assembly Code
Q[3 * 2 - 4]	int	3	movl %rdx(-4, 3, 2), %eax
Q[i * 3 - 2]	int	Mem[0x124]	movl %rdx(-2, %rcx, 3), %eax
Q + 6 + i	int*	0x12C	leaq %rdx(6, %rcx), %rax
&Q[i + 2]	int*	0x11C*(thanks to Fankai Yan)*	leaq 0x2(%rdx, %rcx), %rax

b. i) There are $64 \times 1024 \div 16 = 4096$ cache lines in the cahce. The main memory address will be $\log_2 (2^5 \times 2^{10} \times 2^{10}) = 25$ bits.

For fully associative cache: offset will have $\log_2 16 = 4$ bits, no field for set index since cache is fully associative(one big set), and tag will have $25 - 4 = 21$ bits.

For directly mapped cache: offset will still have 4 bits, set index will have $\log_2 4096 = 12$ bits, and tag will have $25 - 4 - 12 = 9$ bits.

For 8-way associative cache: offset will still have 4 bits, set index will have $\log_2 \frac{4096}{8} = 9$ bits, and tag will have $25 - 4 - 9 = 12$ bits.

ii) The corresponding binary for those addresses are (omit leading 0s):

Address	Binary Representation
0	...00 0000
12	...00 1100
16	...01 0000
20	...01 0100
32	...10 0000

The first four bits are always offset. When **0** is sent to the cache, it will be a cold miss since this is the first query to 0 and the cache is initially empty. Then **12** is sent to the cache, which is a hit(the set index/tag are the same as address **0**). When **16** is sent to the cache, this is again a miss, and **20** will be a hit. **32** will bit another miss, thereby the miss rate is $\frac{3}{5} = 60\%$