CPU Basic Concepts and Performance Metrics

Objectives

1. To understand the concepts of throughput, CPI, CPU time, clock rate, MIPS and FLOPs
2. To solve CPU performance related exercises

Tasks

1. Given that the opcode of an instruction set has the width of 8 bits:
   - What is the full instruction set size? **Answer: 2^8=256**
   - What would the opcodes of the last 2 instructions be in HEX? **Answer: $FF_{16}=1111\ 1111_2$, $FE_{16}=1111\ 1110_2$**

2. Which plane has better performance?

Plane	London to Moscow	Passengers
Airplane 1	6 hours	100
Airplane 2	3 hours	20

- **Response time**: The time between the start and completion of a task. It includes time spent executing on the CPU, accessing disk and memory, waiting for I/O and other processes, and operating system overhead.
- **Throughput**: The total amount of work done in a given time.
- **CPU execution time**: Total time a CPU spends computing on a given task (excludes time for I/O or running other programs).

Airplane 2 is two times faster in terms of flying time, but slower in terms of throughput as throughput1=16.6 passengers/hour and troughput2=6.6 passengers/hour

3. Basic concepts :
   - A given program will require
     - some number of instructions (machine instructions)
     - some number of clock cycles
     - some number of seconds
   - The **clock rate** (cycles per second) is the inverse of the **clock cycle time** (seconds per cycle), for example, if a computer has a clock cycle time of 5 ns, the clock rate is (*1 / 5 x 10^{-9} sec)=200MHz*
   - **CPI** (cycles per instruction). The CPI is the average number of cycles per instruction
   - **CPU time** is the time to execute a given program
   - **Different instructions take different number of CPU cycles**, e.g., division takes more cycles than addition, floating point instructions take more cycles than fixed point, accessing memory takes more than accessing registers etc.
   - **CPU clock cycles** is the number of CPU clock cycles
   - Given the above concepts :
     - clock rate=1/clock cycle time (1)
     - CPU time = CPU clock cycles x clock cycle time (2)

- CPU time = CPU clock cycles / clock rate , *because of (1) and (2)* (3)
- CPU clock cycles = (instructions/program) x (clock cycles/instruction)=
  = Instruction count x CPI (4)
- CPU time = Instruction count x CPI x clock cycle time, *because of (2) and (4)* (5)
- CPU time = Instruction count x CPI / clock rate, *because of (3) and (4)* (6)
- CPU time=(instructions/program) x (clock cycles / instruction) x (seconds/clock cycle) , *because of (4) and (2)* (7)

4. Consider that the CPU clock rate is 1 MHz and the Program takes 45 million cycles to execute. What's the CPU time? **Answer: 45,000,000 * (1 / 1,000,000) sec = $45*10^6 * (1/10^6)$ sec = $45*10^6 * 10^{-6}$ sec = $45 *10^{6-6}$ sec = $45 *10^0$ sec = 45*1 sec = 45 seconds**

5. Why in 32-bit CPUs we can use only up to 4GBytes of RAM memory?

   Answer: In 32-bit CPUs the address bus is 32bit wide. This means that there are 32 digits to address all words in main memory and thus the memory consists of 2^{32} words/bytes, i.e., 4Gbytes.

6. If main memory is of 32Mbyte and every word is of 4 bytes, how many bits do we need to address any single word in memory?

   ***Answer**: The memory address space is 32 MB, which means $32 * 2^{20}=2^5*2^{20}=2^{25}$. However, each word is four (2^2) bytes, which means that we have $2^{25}/2^2 = 2^{23}$ words. Note that (Mem.size=number.words x word.size). This means that we need $\log_2 2^{23} = 23*\log_2 2 = 23*1 =$ 23 bits, to address each word.*

7. ***(Optional – not assessed)*** A program has 100 instructions from which 25 instructions are loads (each take 3 cycles), 50 instructions are add (each takes 1 cycle) and 25 instructions are branch (each takes 2 cycles). What is the CPI for this benchmark? **Answer: CPI = 3*(25/100) + 1*(50/100) + 2*(25/100) = ((0.25 * 3) + (0.50 * 1) + (0.25 * 2)) = 1.75 cycles per instruction**

8. ***(Optional – not assessed)*** Assume a program of 1.000.000 instructions and two implementations of the same instruction set architecture (ISA). CPU.A has a clock cycle time of 10 ns. and a CPI of 2.0, while CPU.B has a clock cycle time of 20 ns. and a CPI of 1.2. Which CPU is faster for this program?

   Answer:

   CPU time = Instruction count x CPI x clock cycle time. Thus,
   CPU.A time = $10^6 * 2.0 * 10 * 10^{-9} = 2 * 10^{6+1-9}$ seconds = $2 * 10^{-2}$ sec = 2/100 sec = 0.02 sec
   CPU.B time = $10^6 * 1.2 * 20 * 10^{-9} = 1.2 * 2*10 * 10^6 * 10^{-9}$ seconds = $1.2 * 2 * 10^{7-9}$ seconds = $2.4 *10^{-2}$ sec = 2.4/100 = 0.024 sec
   CPUA is faster 0.024/0.020=1.2 times

9. ***(Optional – not assessed)*** Performance Metrics
   - **MIPS** : millions of instructions per second
   - **FLOPS** : floating point operations per second

Consider a CPU of 500MHz and three different classes of instructions: Class A, Class B, and Class C, which require one, two, and three cycles, respectively. The first code uses 5 billions Class A instructions, 1 billion Class B instructions, and 1 billion Class C instructions. The second compiler's code uses 10 billions Class A instructions, 1 billion Class B instructions, and 1 billion Class C instructions. Which sequence will be faster according to MIPS? Which sequence will be faster according to execution time?

Answer:

CPU Clock cycles1= $(5 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 10 \times 10^9$

CPU Clock cycles2= $(10 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 15 \times 10^9$

CPU time1= $10 \times 10^9 / 500 \times 10^6 = 20$ *seconds (CPU time = CPU clock cycles / clock rate)*

CPU time2= $15 \times 10^9 / 500 \times 10^6 = 30$ *seconds*

MIPS = instruction count / (execution time x 10^6*)*

MIPS1= $(5 + 1 + 1) \times 10^9 / 20 \times 10^6 = 350$

MIPS2= $(10 + 1 + 1) \times 10^9 / 30 \times 10^6 = 400$

Algebra basics

$a^x * a^y = a^{x+y}$

$a^x / a^y = a^{x-y}$

$1/a^x = a^{-x}$

$\log_b a^x = x * \log_b a$

$\log_2 2 = 1$

$16000 = 1.6 * 10000 = 1.6 * 10^4$

Further Reading

Chapter 2 in 'Computer Organization and architecture' available at
http://home.ustc.edu.cn/~leedsong/reference_books_tools/Computer%20Organization%20and%20Architecture%2010th%20-%20William%20Stallings.pdf