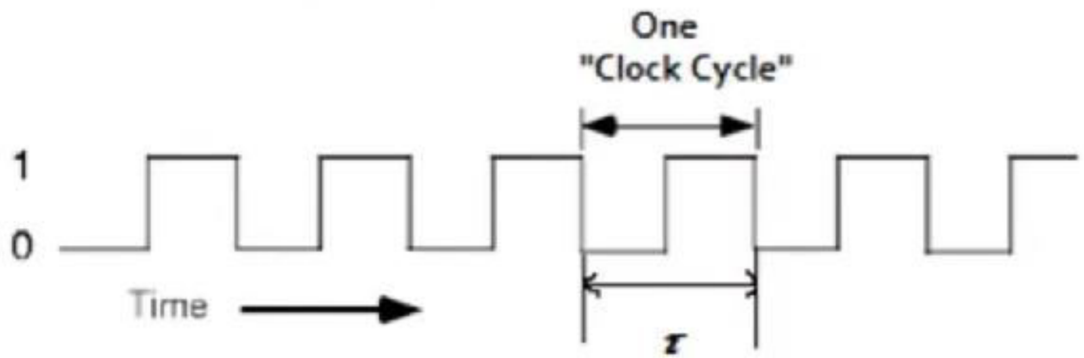
SYSTEM ATTRIBUTES TO PERFORMANCE: CPU PERFORMANCE EVALUATION

System attributes to performance

- **Turn around Time** : It is *simplest measure of program performance*. It is the time which includes disk and memory access, input and output activities, compilation time, OS overhead and CPU time.
- Other important attributes to calculate CPU Performance are:
  1. **Clock rate and CPI**
  2. **Execution Time**
  3. **MIPS Rate**
  4. **Throughput Rate (Performance)**

1. Clock rate and CPI

- ***CPU Clock:*** Computers are constructed in such a way that events in hardware are synchronized using a **clock**



- ***Cycle Time/ clock time/clock period (τ):***

  CPU is driven by a **clock** of constant **cycle time τ (in ns)**

- ***Clock rate/ clock frequency (f):*** is defined by the **inverse of cycle time**

$$\boxed{\text{Clock rate } f = 1/\tau}$$ **in Hz (= cycle/second)**

Also, Cycle time $\tau = 1/f$

Example

1. $f = 1\ \text{GHz} = 10^9\ \text{Hz}$

   $\tau = \dfrac{1}{f} = \dfrac{1}{10^9}$ SPC $= 1\text{ns}$

2. $f = 500\ \text{MHz}$

   $= 500 * 10^6\ \text{Hz}$

   $\tau = \dfrac{1}{f} = \dfrac{1}{500 * 10^6\ \text{Hz}}$ SPC $= 2\text{ns}$

Example

1. $f = 1\ \text{GHz} = 10^9\ \text{Hz}$

   $\tau = \dfrac{1}{f} = \dfrac{1}{10^9}$ SPC = 1ns

2. $f = 500\ \text{MHz}$

   $= 500 * 10^6\ \text{Hz}$

   $\tau = \dfrac{1}{f} = \dfrac{1}{500 * 10^6\ \text{Hz}}$ SPC = 2ns

Clock rate and CPI …

- ***Instruction count (Ic)***: *Instruction count (Ic) is the* ***size of a program****, which is the no. of machine instructions to be executed in a program. Sometimes called instruction path length*
- *A machine instruction is a sequence of micro-operation. A micro-operation is an elementary hardware operation that can be carried out in one clock cycle.*
- *Thus a single machine instruction may take one or more CPU cycles to complete.*
  - *We can characterize an instruction by* **Cycles Per Instruction (CPI)**
- ***CPI*** : Different machine instructions may require different number of clock cycles to execute. Therefore ***CPI (Cycles Per Instruction)*** becomes an important parameter.
  - ***CPI measures the time needed to execute each instruction i.e. number of clock cycles per instruction for a program***

$$CPI = \frac{\text{CPU Clock cycles for a program}}{\text{Instruction Count}} = \frac{C}{Ic}$$

Clock rate and CPI …

- ***Average (or Effective) CPI of a program:***
  - Average (or Effective) CPI of all instructions executed in the program on a given processor
  - Different instructions can have different CPIs

2. Execution Time (CPU Time)

- *Execution Time (CPU Time): It is the time needed to execute the program*

  Thus the *Execution Time / CPU time (T in seconds / program) is* given by:

$$\boxed{T = Ic * CPI * \tau} \quad \text{----(1)}$$

Where Ic = *Total no. of instructions executed*

CPI = *average number of Cycles per instruction (CPI) for a program*

τ = *Clock cycle time (Clock period)*

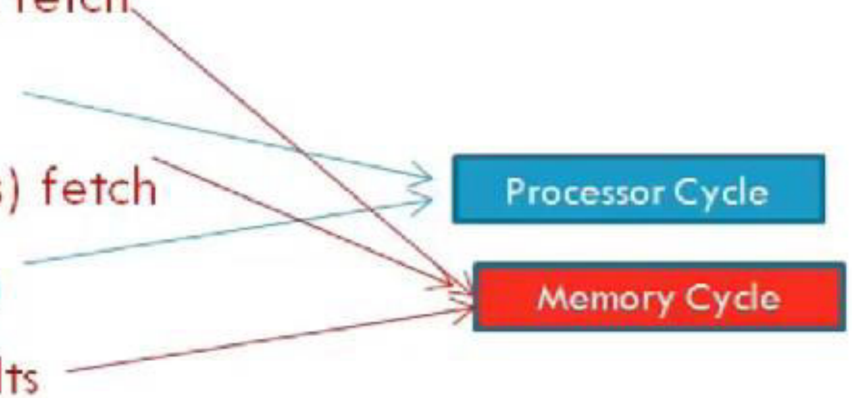
T = *Execution Time per program in seconds*

- *The **units** for CPU Execution time are:*

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

Execution Time (CPU Time)...

- The execution of instruction (i.e. Instruction cycle) requires some steps:
  - instruction fetch
  - **decode**
  - operand(s) fetch
  - **execution**
  - store results



- In instruction cycle, only decode and execution phases are carried out in the CPU (Processor cycle). The remaining three operations may require memory access (memory cycle).
- Therefore, ***CPI = Instruction cycle = processor cycles + memory cycles.***

  Usually, memory cycle is k *times* the processor cycle τ

  CPI = Instruction cycle = p + m * k

Execution Time (CPU Time)....

Therefore, CPU time T is given by:

$$\boxed{T = I_c * (p + m * k) * \tau} \qquad \text{----(2)}$$

Where

I_c = instruction count

p = number of processor cycles

m = number of memory references

k = ratio between memory and processor cycle

τ = processor cycle time

T = CPU Time (Execution Time)

System Attributes

- **The 5 performance factors (I_c, p, m, k, τ) are influenced by 4 system attributes:**

System Attributes	Ic	p	m	k	τ
Instruction set architecture.	X	X			
Compiler technology.	X	X	X		
CPU implementation & control		X			X
Cache & memory hierarchy				X	X

1. **Instruction-set architecture (ISA)** affects Ic (program length), p (processor cycle)
2. **Compiler technology** *affects* Ic, p *and* m (memory reference count)
3. **CPU implementation and control** *determines the* total processor time $= p * \tau$
4. **Cache and memory hierarchy** *affects* memory access time$= k * \tau$

3. MIPS Rate

- Let C be the ***Total number of clock cycles*** *needed to execute a program.*

  Therefore, ***CPU time (T) = Total number of clock cycles * clock cycles***

$$T = C * \tau = C / f$$

Furthermore, as we know $CPI = C / Ic$ So, $\boxed{C = CPI * Ic}$

and $T = CPI * Ic * \tau$

Therefore,

$$\boxed{T = \frac{Ic * CPI}{f}} \quad \text{----(3)}$$

- **The processor speed is often measured in terms of MIPS (Millions Instructions Per Second).** It is simply called **MIPS Rate** of a given processor.

$$\boxed{MIPS\ Rate = \frac{Ic}{T * 10^6} = \frac{f}{CPI * 10^6} = \frac{f * Ic}{C * 10^6}} \quad \text{----(4)}$$

Using equation (4)

$$\boxed{T = \frac{Ic}{MIPS * 10^6}}$$

4. Throughput rate (Performance)

- **Throughput** (**Ws**) **or Performance**: The number of programs executed per unit time is called system throughput (in programs /second)

$$W_s = \frac{1}{T} = \frac{1}{\text{Execution time}}$$

$$\boxed{W_s = \frac{f}{I_c * CPI} = \frac{MIPS * 10^6}{I_c}}$$

Instruction types and CPI

- Consider a program executing on a processor with **n** types or classes of instructions (like load, store, ALU, branch, etc.)

We know *CPI = CPU Clock cycles for a program (C) / Instruction Count (Ic)*

I_{ci} = number of instructions of type *i* executed

CPI_i = cycles per instruction for type *i*

So,

For CPU design:

$$\boxed{CPU\ clock\ cycles = CPI * I_c = \sum_{i=1}^{n} (CPI_i * I_{ci})}$$

The overall CPI is given by:

$$\boxed{Average\ (or\ Effective)\ CPI \quad \sum_{i=1}^{n} (CPI_i * I_{ci}) = \frac{\sum_{i=1}^{n} (CPI_i * I_{ci})}{I_c} = \sum_{i=1}^{n} \left(CPI_i * \frac{I_{ci}}{I_c}\right)}$$

Example 1

- For the multi-cycle MIPS Processor, there are five types of instructions:

  Load (5 cycles)

  Store (4 cycles)

  R-type (4 cycles)

  Branch (3 cycles)

  Jump (3 cycles)

If a program has:

  50% load instructions

  25% store instructions

  15% R-type instructions

  8% branch instructions

  2% jump instructions

$$CPI = \frac{\sum_{i=1}^{n} (CPI_i * I_{ci})}{I_c}$$

then, the CPI (or Average CPI) is:

$$\mathrm{CPI} = \frac{5 \times 50 + 4 \times 25 + 4 \times 15 + 3 \times 8 + 3 \times 2}{100} = 4.4$$

Example 2

- An instruction set has three instruction classes A, B, C:

Instruction class (Eg: ALU, Branch, etc)	CPI
A	1
B	2
C	3

- Two code sequences have the following instruction counts (Ic):

Code Sequence	Instruction counts *(Ic)* for instruction class A	B	C
1	2	1	2
2	4	1	1

- ***CPU cycles for sequence 1*** = $CPI * Ic$ = 2 x 1 + 1 x 2 + 2 x 3 = 10 cycles

  CPI for sequence 1 = *clock cycles* / Ic = 10 /5 = 2

- ***CPU cycles for sequence 2*** = $CPI * Ic$ = 4 x 1 + 1 x 2 + 1 x 3 = 9 cycles

  CPI for sequence 2 = *clock cycles* / Ic = 9 / 6 = 1.5