Single Processor Performance

Amdahl's Law

- **Amdahl's law**

$$= \frac{\text{execution time for entire task without enhancement}}{\text{execution time for entire task using enhancement when possible}} ;$$

- **Example : suppose FPSQR is responsible for 20 % of the execution time of a benchmark. One proposal is to add FPSQR (Floating Point Square Root) hardware that will increase this operation by a factor of 10. Other option is to make (with same effort required to built FPSQR hardware) all FP instructions run two times faster. FP instructions take 50% of execution time**
  Answer :
  Speedup_FPSQR = 1/((1-.2) + .2/10) = 1/.802 = 1.25 ;
  Speedup_FP = 1/(.5 + .5/2) = 1/.75 = 1.33 ;

  Improving FP overall is better

CPU Performance Equation

- **Most computers are constructed using a clock running at a constant rate called clock cycles, clock period, cycles, ticks. This is expressed by its length e.g. 2 ns or by its rate 500 Mhz**

- **CPU time for a program**
  = (CPU clock cycles for a program) * (clock cycle time)

$$\text{CPU time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{clock rate}};$$

CPU Performance Equation

IC = Instruction Count for a program ;

Clock cycles per instruction (CPI)

$$= \frac{\text{CPU clock cycles for a program}}{\text{IC}} ;$$

CPU time = IC * CPI * (clock cycle time)

$$= \frac{\text{IC} * \text{CPI}}{\text{clock rate}}$$

$$= \frac{\text{Instruction}}{\text{Program}} * \frac{\text{clock cycles}}{\text{Instruction}} * \frac{\text{seconds}}{\text{clock cycle}} = \text{CPU time} ;$$

CPU Performance Equation

- **Improvement on any one will improve CPU time. Basic technologies involved in changing each characteristic are also interdependent.**

Clock Cycle Time:Hardware technology and organization
CPI : Organization and Instruction Set Architecture
Instruction Count: Instruction Set Architecture and compiler technology

Useful in CPU design is to calculate the number of total CPU clock cycles as :

CPU clock cycles = $\sum_{i=1}^{n} (CPIi) X (ICi)$ **;**

instruction type i

Example: Earliner example expressed in terms of the frequency of the instruction and of the instruction CPI values.

Frequency of FP operations = 25%
Average CPI of FP operations = 4.0
Average CPI of other instructions = 1.33
Frequency of FPSQR = 2%
CPI of FPSQR = 20

Two alternatives: (i) reduce the CPI of FPSQR to 2
(ii) reduce the average CPI of all FP ops to 2

Answer: Only CPI changes, clock rate and IC remain identical. CPI with neither enhancements: $CPI_{orig} = (4*25\%) + (1.33*75\%) = 2.0$

$$CPI_{newFPSQR} = CPI_{orig} - 2\%(CPI_{origFPSQR} - CPI_{enhancedFPSQR})$$

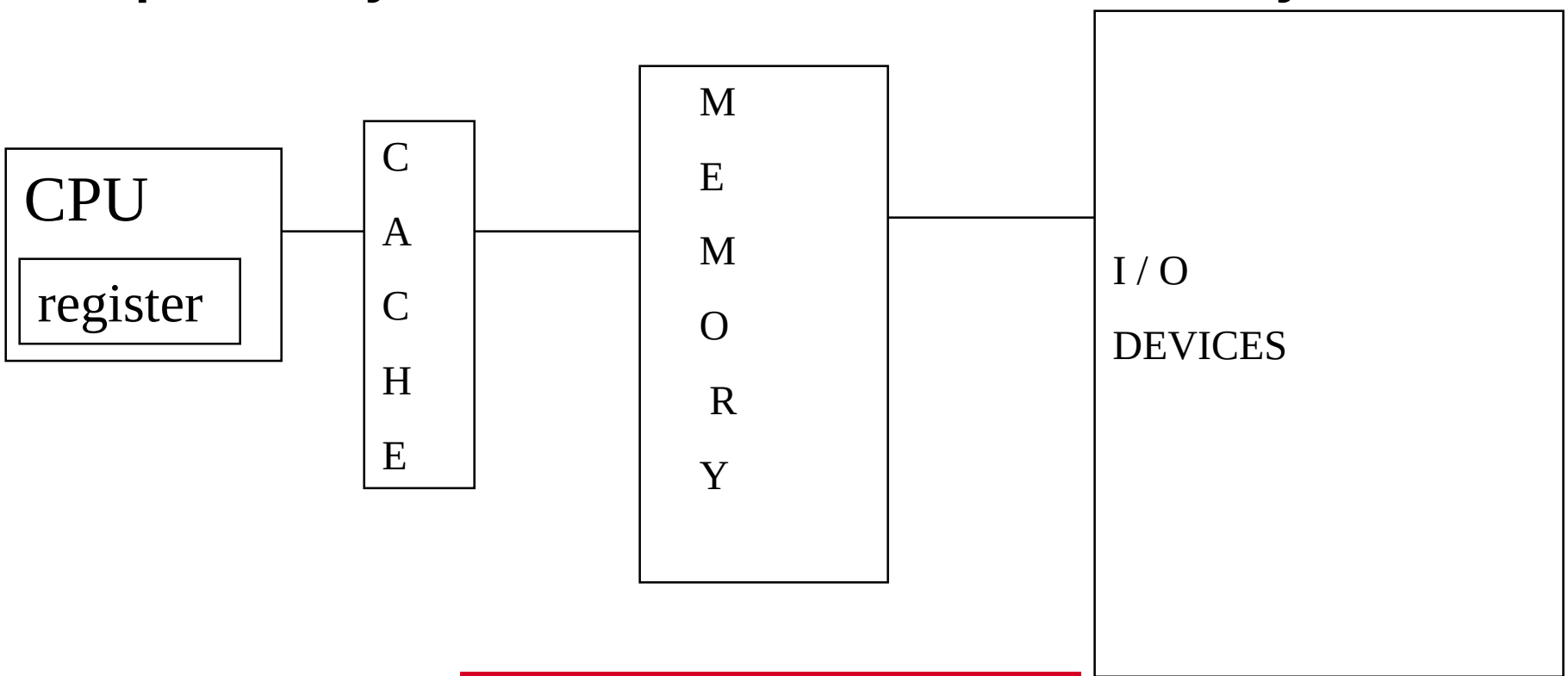
$$= 2.0 - 2\%(20 - 2) = 2.0 - 0.36 = 1.64;$$

$$CPI_{newFP} = CPI_{orig} - 25\%(4 - 2) = 2.0 - 0.5 = 1.5 \ (= 2*25\% + 1.33*75\%);$$

$$\text{Speedup} = \frac{\text{CPU time orig}}{\text{CPU time enhanced}} = \frac{(IC)*(\text{clock cycle})*(CPI_{orig})}{(IC)*(\text{clock cycle})*(CPI_{enhanced})} = \frac{2.0}{1.5} = 1.33;$$

Concept of Memory

- **Increase memory bandwidth (BW) and decrease the time to access memory. Both crucial to system performance.**
- **Principle of locality says that the data most recently used is very likely to be accessed again the near future. Try to keep recently accessed items in fastest memory.**



Concept of Memory (~1995 data)

Registers (compiler)	Cache (hardware)	Main memory (OS)	Disk storage (OS/User)
< 1 Kb	< 4 Mb	< 4 Gbytes	> Gbytes ~ terabytes
CMOS	Onchip or offchip CMOS, SRAM	CMOS DRAM	Magnetic disk
2 – 5ns	3 – 10 ns	80 – 400 ns	5,000,000 ns
4000-32000 Mb/sec	800 – 5000 Mb/sec	400 – 2000 Mb/sec	4 – 32 Mb/sec

Caches

- A cache is a small, fast memory located close to the CPU that holds the most recently accessed code or data.
- When the CPU finds a requested data item in the cache, it is called a cache hit; when the CPU does not find a data item it needs in the cache, cache miss occurs.
- CPU stall happens when there is a cache miss
- A fixed size block of data, called a cache line, or block, containing the requested word is retrieved from the main memory and placed into the cache.
- Time required for cache miss depends on both the latency and the BW of the memory to cache.

Virtual Memory

- If a computer has virtual memory then some objects may reside in the disk. The address space in memory is usually broken into fixed size blocks, called pages. At any time a page resides either in main memory or disk.
- When CPU references an item that is not in the cache or main memory a page fault occurs and the entire page is moved from the disk to main memory.
- Page fault takes too long and hence they are handled in the software and the CPU is not stalled. The CPU switches to some other task while the disk access occurs.
- Cache and main memory has same relationship as main memory and disk.