Assignment 2

Joy Awino ICS B 151542

1. Determine the effective CPI, million instructions per second (MIPS) rate, and execution time (CPU) for each machine and comment on the results

Effective CPI = (Instruction count * Cycles per instruction)/Instruction count

Machine A

{(10 * 1) + (6 * 3) + (2 * 4) + (4 * 3) } * 10^6 / (10+6+2+4) * 10^6 = 2.1818

Machine B

{(12 * 1) + (10 * 2) + (2 * 4) + (4 * 3) } * 10^6 / (12+10+2+4) * 10^6 = 1.857

Million Instructions per Second = clock rate * 10^6/ effective CPI * 10^6

Machine A

200 * 10^6/2.1818 * 10^6 = 91.97

Machine B

200 * 10^6/ 1.857 * 10^6 = 107.70

Execution time = Instruction count * CPI / Clock rate

Machine A

{(10+6+2+4) * 10^6} * 2.1818/ 200 * 10^6 = 0.24s

Machine B

{(12+10+2+4) * 10^6} * 1.857 / 200 * 10^6 = 0.26s

Comments

Even though machine B has a higher MIPS than machine A, it needs longer CPU time to execute the similar set of instructions.

2. Consider the difference between processors in terms of pipelining i.e. how pipelining is handled from processor 4004 to Pentium series to Celeron, Duo Core, and Intel Core, Core i series.

Intel 4004 was the first microprocessor and did not employ pipelining as each instruction was executed sequentially.

Pentium series introduced deeper instruction pipelines where it had a 5 stage pipeline which allowed concurrent execution of instructions.

Celeron processors were more budget-oriented and have a similar pipelining structure to the pentium series except with fewer features and reduced cache sizes to lower costs.

Duo core processors featured a dual core design which enabled simultaneous execution of multiple instructions on separate cores.

Intel core processors increased pipeline depth which further improved performance by reducing pipeline stalls.

Core I series had a more advanced microarchitecture since it introduced features like hyper-threading which allows each physical core to handle multiple software threads simultaneaously.

3. A nonpipeline system takes 100ns to process a task. The same task can be processed in a 5-stage pipeline with a clock cycle of 20ns. Determine the speedup ratio of the pipeline for 100 tasks. What is the theoretical speedup that could be achieved with the pipeline system over a nonpipelined system?

Total time to process 100 tasks

Pipeline time

$T_k = \{k + (n-1)\}\ \tau$

$T_k = \{5 +(100-1)\}20 = 2080ns$

Sequential time

$T_1 = nk\tau$

$T_1 = 100 * 5 * 20 = 10000ns$

Speedup ratio

$S_k = T_1/T_k$

$S_k = 10000/2080 = 4.8077$

The theoretical speedup that could be achieved is determined by the number of stages which is 5.

4. Discuss the concept of RAID as applied in storage system. Your discussion should highlight the general idea and any 4 levels used in the implementation of RAID.

RAID stands for Redundant Array of Independent Disks. In RAID data is stored across many disks with extra disks added to the array thereby introducing redundancy.

Raid level 0 enhances performance by dividing data into blocks across the entire array and storing them across multiple drives simultaneously. There is no redundancy.

RAID level 1 provides 100% redundancy and good performance as an exact copy of the data is maintained on two drives.

RAID level 2 consists of a set of data drives and a set of hamming code drives. Hamming code drives can correct single bit errors and detect double bit errors.

RAID level 6 stripes data at the block level and uses dual distributed parity. It is write-sensitive but highly fault tolerant.

5. Explain how caches are used to exploit the locality of reference for a performance benefit. Explain for both spatial locality and temporal locality.

Caches exploit spatial locality by storing not just the requested data, but also additional data located nearby in memory. This is achieved when the cache retrieves not only the requested data but also a block of contiguous data.

Caches exploit temporal locality by retaining recently accessed data in the cache. This is achieved when the cache stores recently accessed data so that when a program reaccesses the same memory location the data is quickly retrieved from the cache.