**ASSIGNMENT-4**

**CIS-580 INTRO TO COMPUTER ARCHEITECTURE SECTION 50**

**ANIL PAVULURU**

**1.**

**A)**

**Direct Mapped Cache**

A direct mapped cache has one block in each set, so it is organized into S = B sets. To understand the mapping of memory addresses onto cache blocks, imagine main memory as being mapped into b-word blocks, just as the cache is. An address in block 0 of main memory maps to set 0 of the cache. An address in block 1 of main memory maps to set 1 of the cache, and so forth until an address in block B − 1 of main memory maps to block B − 1 of the cache. There are no more blocks of the cache, so the mapping wraps around, such that block B of main memory maps to block 0 of the cache.

In this memory block “i “ is mapped to I modulo 12 of cache. Blocks 0,128,256… are mapped to block 0

Direct Mapping technique is very simple and easy to implement

Search speed is high because there is only one place a block can present

Hit ratio always will not have good results performance is slower with this mapping

OR ELSE SIMPLY

This is the simplest approach of associating main memory blocks with cache memory blocks with a cache block is called direct mapping cache.in this technique, one specific block of main memory can be transferred to a Particular block of a cache which is done by the modulo function.

**B)**

**Set**-**associative cache**

Set-associative cache is a trade-off between direct-mapped cache and fully associative cache. A set-associative cache can be imagined as a (n\*m) matrix. The cache is divided into 'n' sets and each set contains 'm' cache lines. A memory block is first mapped onto a set and then placed into any cache line of the set.

Set [associative caches](https://www.sciencedirect.com/topics/computer-science/associative-cache) generally have lower miss rates than direct mapped caches of the same capacity, because they have fewer conflicts. However, set associative caches are usually slower and somewhat more expensive to build because of the output multiplexer and additional [comparators](https://www.sciencedirect.com/topics/computer-science/comparators).

OR ELSE SIMPLY

When a memory block is first mapped onto a Particular set and then placed into any cache line of the set. The range of caches from direct mapped to fully associative is a continuum of levels of set associativity. Direct mapped in one way set associative and fully associative cache with n blocks is n-wat set associative.

**C)**

**Dynamic branch predictions**

Dynamic branch predictionuses information about taken or not taken branches gathered at run-time to predict the outcome of a branch.

The prediction technique that makes prediction based on previous history situation, by looking at history table is known as Dynamic branch prediction.

The actions during the execution is may change in dynamic branch prediction.

OR ELSE SIMPLY

It uses the information about taken or not taken branches collected at run time to predict outcome of a branch. In this identify individual branches for their pc or dynamic branch history. There are number of dynamic branch predictors in use or being researched nowadays. These include one-level branch predictors, two level branch predictors and a number of components branch predictors.

**D)**

**Static branch prediction**

Static prediction is the simplest branch prediction technique because it does not rely on information about the dynamic history of code executing. Instead, it predicts the outcome of a branch based solely on the branch instruction.

The prediction technique that makes fixed prediction of a branch to either taken or not taken is known as static branch prediction.

The action during the execution is fix in static branch prediction.

OR ELSE SIMPLY

Static branch prediction is a prediction that use information that was gathered before the execution of the program. In this prediction, the compiler determines whether a branch is likely to be taken or likely to be not taken. It does not depend on history. This technique takes decision based on analysis or profile information.

**E)**

**Register Renaming**

It deals with data dependencies between instructions by renaming their register operand. An assembly language programmer or a compiler particularize these operands using architectural registers. Renaming replaced architectural registers names by, in effect, name of value, with a name of a value for each instruction destination operand. This eliminates the name dependencies between instructions. The recognition of true data dependencies between instructions permits a more flexible life cycle for instructions.

**F)**

**Speculative Execution**

It is technique used by modern CPUS to increase speed performance. The CPU may execute some tasks ahead of time, speculating that they will be needed. If the tasks are required, a speed-up is achieved, because the work is already done. In the other words, it is a technique where a computer completes certain tasks before those are known whether these are actually needed, so as to prevent delay that happens due to work after it is known that is needed.