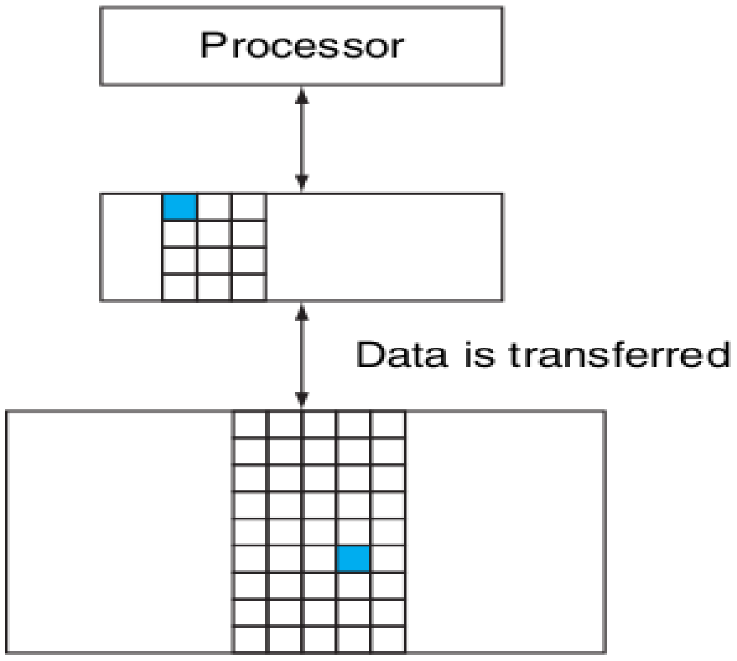
Assignment – 10
COL216 – Computer Architecture
Variable Delays in a Pipelined Processor

This assignment is an extension of the previous assignment. We overcame the data hazards in the previous assignment by forwarding and bypassing. Now in this assignment, we are introduced with the variable delays while excessing memory. A floating point computation may depend on the data operands. So a memory access may exhibit variable delays due to the cache heirarchy.

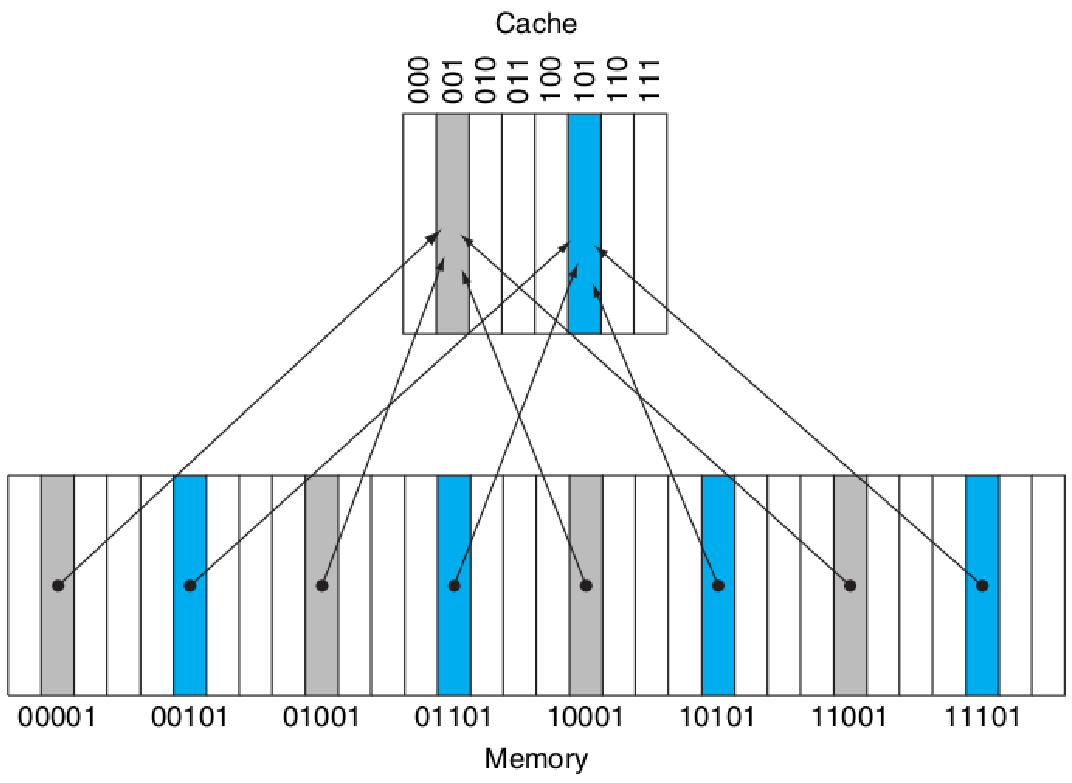
Here is the basic structure of memory heirarchy. Every pair of levels in the memory hierarchy can be though of as having an upper and lower level. With each level, the unit of information that is present or not is called a *block* or a *line*. Usually we transfer an entire block when we copy something between levels.



These variable delays slow down the processor by consuming extra cycles in excessing memory. If the desired data is available in cache memory, we consider this a HIT. If not, then we need to excess the main D-memory as we consider this situation as MISS.

Excess of main memory is much slower than cache. The probability of a HIT increases with the total number of blocks in cache memory. But size of cache memory decreases the speed of excessing cache memory also.

A direct-mapped cache with eight entries showing the addresses of memory words between 0 and 31 that map to the same cache locations, is showen below.



Lets x be the probability of a HIT. i.e. the operation completes in one cycle with a probability of x.

(1-x) be the probability of a MISS. i.e. the operation requires N cycles to be completed.

IMPLEMENTATION :

If the current instruction uses memory excess, then it may a HIT or MISS. For the simulation, we simply generate a random number between 0 to 99 and compare this random number to the given probability.*100.

If random number is greater then probability*100
then it a MISS, so we will add (N-1) cycles to total cycles, and continue the program as previous.

Else
it is a HIT, as obvious it will take one cycle to complete the operation and the further process is same.

If we operate the same program again and again for various values of x and N, we observe the folloing points.

Given that total number of instructions that excess Data-memory : 21.

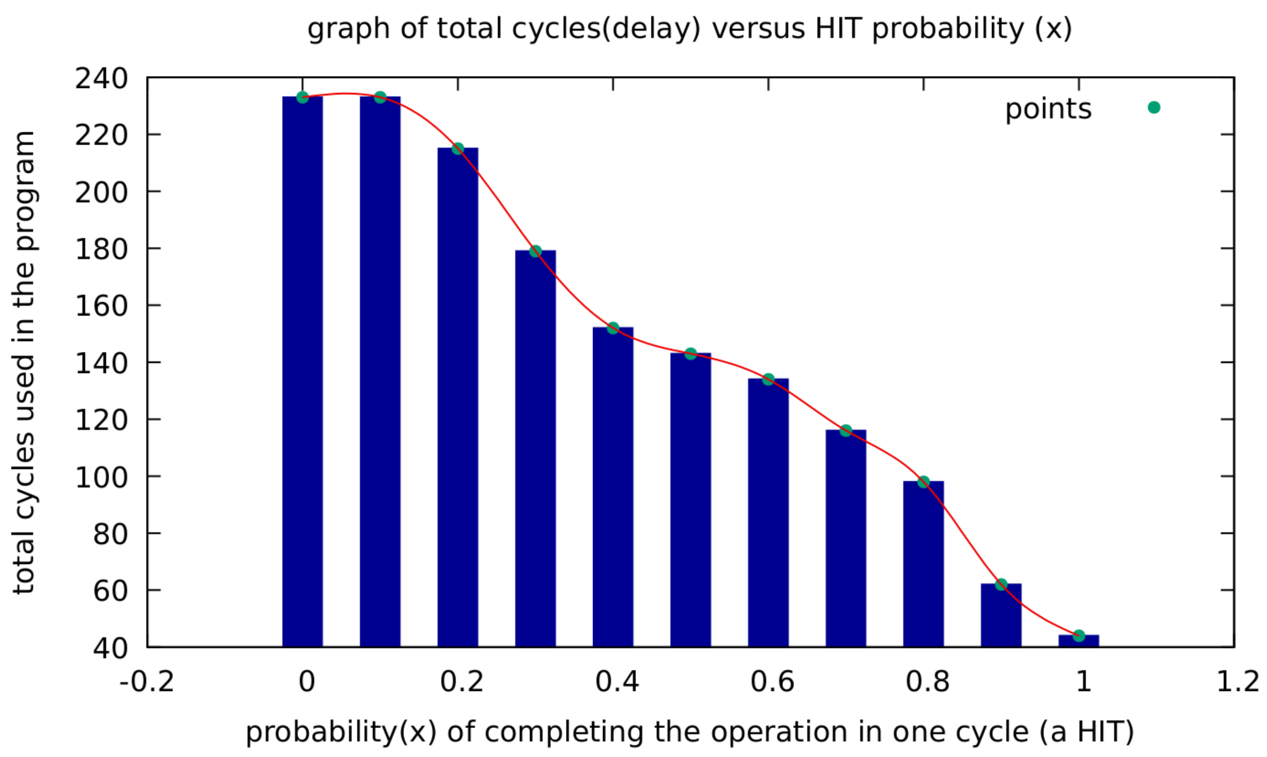
take N = 10

Use commands :

make

make run

Then we got the following plot for the range of x in [0, 1]



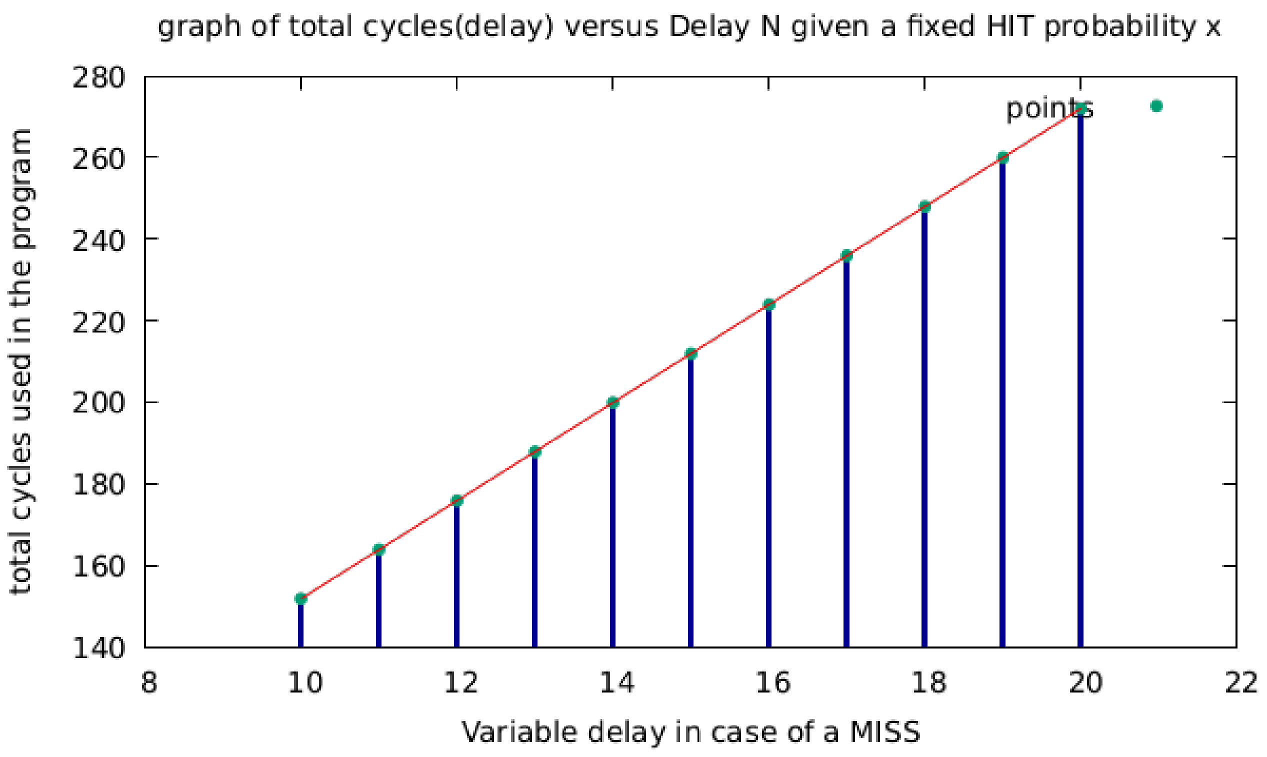
It shows that total number of executed cycles decreases as the probabilty of a HIT increases.

Now lets keep x fixed

take x = 0.4 (HIT probability)
vary N from N = 10 to 20.
Gievn total number of instructions that excess Data-memory : 21

Use commands :
make
make run2

Then we got the following plot :



This graph shows that the number of executed cycles linearly increases with N as we fixed the probability of a HIT, so the number of instructions that are MISS is constant. And extra cycles will be (N-1)*(total MISS instructions).

Assignment By
Nikita Bhamu 2018CS50413
Manoj Kumar 2018CS50411