

Assignment 10: Variable Delays in a Pipelined Processor

ENTRY NO. 2018CS10354_2018CS10348

In this assignment we have created a function which takes two arguments, the one which is the int value and the other one which is also the value. One is for probability value and the other one is for N value.

It returns 0 or N on the basis of the condition that it checks before it returns these.

Inside the function I have used a rand() function which generates a random number with equal distribution, so we are taking modulo with 100 so it will generate a number between 0 to 99 with equal distribution, now I am checking if this number lies between $0 - 100 * x$

Ex- if x is 0.2 then it checks if the number is in range 0-20 if it lies then return 1 else return N.

If it returns 1 then we continue the code without doing any changes but when it returns N we are stalling the process. In the next cycle, we only call the MEM() once and we simply increment the cycle to N-2.

Rest of the program is the same as mentioned in the report of the assignment 8 and assignment 9.

The reason for which we are doing this is to actually know that not every time we actually get hit (get the data value from memory in the one cycle) but what happens in reality is that there would be a chance of getting misses also in that case we take extra cycle to get that data, till we stall the rest of the instruction.

When we get the data we resume the process. So to analyse that we are doing part.
