“Implementation and Comparison of Two Cache Prefetching Schemes”

Team Members:

S.No.	Name	UFID
1	Avinash Ayalasomayajula	0699-6946
2	Neel Kanjaria	0669-6781
3	Yashwanth Katta	3451-7972

Team Name: Team Alan Turing

`

Motivation:

The present processors(superscalar) are aiming at improving performance by having CPI < 1. But, for that, more than one functional units are required and having so we have to make sure that this inclusion gives us the effective CPI < 1 without degrading the performance. One way to achieve this is to reduce the number of cache misses because when there is a cache miss, there is some delay in the execution, hence increasing the CPI. So reducing the probability of cache misses is one of the ways to optimize the performance. This can be done using **“Prefetching”.**

Proposal and Reason:

Although there are both instruction and data prefetching which can be implemented by using hardware or software, we have taken an interest in using the hardware implementation of instruction prefetching. Data prefetching has been implemented efficiently and is highly demanded. But as the need for better performance increases, instruction prefetching is being studied extensively. This field is a new approach and since there are very few promising schemes to implement, there are a still more theories that can be studied and implemented. There are a few techniques that have been studied, but **“Hybrid Prefetching”** and **“Wrong Path Prefetching”** piqued our interest.

1. **Hybrid Prefetching:** A hybrid of the next-line and target line prefetching schemes. It has two prefetch components - next sequential cache line and the other is to prefetch the target of the current cache line if it is non-sequential cache line(like branches) based on its previous behavior. This can be done through a target prefetch table **.**
2. **Wrong Path Prefetching:** This technique tracks the wrong path of execution in hopes that the mis predicted instructions brought into the buffer during speculative execution will turn out to be prefetched for later correctly predicted instructions. It has two components - next sequential line and target line of the current cache line. This target line is prefetched once the branch instruction is detected in the decode stage irrespective of the branch being taken or not.

The performance of these techniques is going to be compared based on cache pollution and miss penalty. Both techniques are a betterment over their predecessors. These techniques would be implemented using simplescalar.