Critique of Paper Presented on November 10
Venkatraghavan Ramesh
1200072784

Presenter – Fei Hong

The paper discusses a methodology to reduce power spent in one fetch and decode when an instruction that has been decoded is to be used again. This is achieved by means of a Decode Filter Cache used in conjunction with Instruction Filter Cache.

Strengths –
1. The idea of using IFC in conjunction with DFC so that spatial locality can also be exploited is a novel contribution of the paper. This will help avoid the performance degradation from DFC misses, when the next instruction is not decoded yet.
2. The power save during DFC hit is considerable given that one fetch and decode are avoided.
3. The prediction mechanism proposed in the paper seeks to avoid the extra pipeline stage and branch prediction mechanism of the Micro-operation cache. By not using trace caches, it greatly simplifies the hardware requirements for implementing storage of decoded instructions for resource constrained embedded processors.

Weaknesses –
1. The paper is abstract in its explanation of the prediction mechanism, and its implementation. The authors could have included examples, flow-charts to illustrate the manner in which the scheme chooses between fetching directly from I-cache, fetching from IFC and fetching from DFC.
2. The authors are not explicit in their explanation of certain aspects which are left to the assumption of the reader. Some of these are –
   ● The absence of branch prediction will not impact the accuracy of prediction
   ● Switching from IFC to DFC occurs immediately after the first hit on IFC when the decoded instruction is available.
   ● The correspondence between cacheable ratio and acceptable execution frequency
3. It is not explained clearly whether the lines that have an invalid in the valid bit are filled with decoded instructions (which will not be used). There may be cases where a DFC hit may occur but a majority of lines in the sector are invalid, which will cause unnecessary overhead. It has been mentioned though that uncachable and cacheable instruction co-exists in DFC.
4. The paper does not justify its claim regarding performance degradation, that it is less than 1%. The overheads due to implementation of both IFC and DFC, apart from the switching due to prediction mechanism are highly unlikely to result in such minimal degradation.
5. The impact on cycle time due to prediction switching between fetching from I-cache and fetching from IFC or DFC is not discussed.
6. In the simulation, it is not clear how varying the cacheable ratio is varied, in what range and how changing this accounts for the decode width which is machine dependent.
7. The memory overhead for Next Fetch Prediction Table is not mentioned in the experimental setup.