Development of Abstract Model for Shared Cache Memory

Student's Name	**R. Ajay**	**RTES (FT-2014)**
Academic Supervisors	P. Padma Priya Dharishini	
Industrial Supervisor(s)		

R. Ajay
ajaycoco@gmail.com
Ph. No: 0 96865 59933

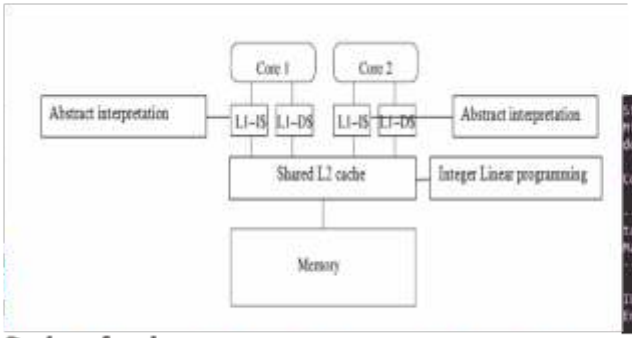
Keywords: Shared Cache Memory, Abstract Interpretation, Integer Linear Programming

Abstract:

Cache memory reduces the access time of memory by increasing the speed of execution. Cache memory is smaller memory that stores the memory blocks that are used frequently. When there is a reference to a memory block it checks first in the smaller memory that is the cache memory, if the memory block is found then it is a hit known as cache hit or it is a miss known as cache miss. When there is a miss, the block is loaded to the cache memory from the main memory by replacing some other block. The execution time resulting from the cache miss is called cache miss penalty.

An abstract model for shared L2 cache is developed where L1 cache is implemented using Abstract Interpretation and shared L2 cache is implemented using Integer linear programming algorithm. Maximum execution time is calculated based on the implemented cache model for L1 cache and shared L2 cache and the desired properties of the program is analyzed based on the pre-computed cache contents.

The developed cache scheme is simulated and analyzed using the chronos tool. The obtained results are analyzed using both chronos simulator and Simple-scalar simulator. Maximum execution time in clock cycles for all the benchmark programs are calculated by running them on both chronos simulator and multi-core simulator and it proves that maximum execution time in clock cycles for all the benchmark programs run on chronos simulator is greater than the benchmark programs that are run on multi-core simulator. Hence the results are precise because it considers all the cache contents



Design of cache memory

Chronos output

Conclusion: The maximum execution time in clock cycles for all the benchmark programs run on chronos simulator is greater than the benchmark programs that are run on multi-core simulator. Hence the results are precise by considering all cache contents.