Address Translation & IRAM Cache

Omri Dassa, Ori Yeffet
Advisor: Mr. Nadav Rotter

The Rachel and Selim Benin School of Computer Science and Engineering
The Hebrew University of Jerusalem
Intel Corp.

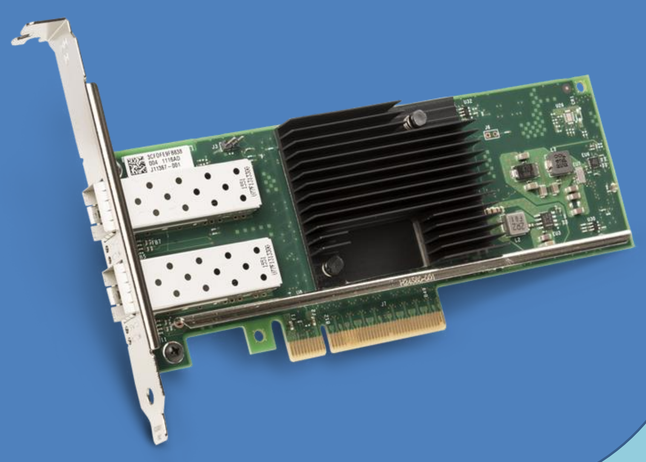
Introduction

Cache-based hardware solution is presented to the Instruction Memory (IRAM) space issue of Intel's NIC (Network Interface C. The solution supports pages protection, so user can block eviction of certain pages.

The method make use of three main hardware mechanisms: (1) Latch-based Mapping table of the IRAM page addresses to the Flash Memory page addresses. (2) Flop-based cyclic 2 way link-list implementation of the cache LRU eviction policy. (3) Combinatorial mechanism, Miss Handler, routes requests according to the address space of the request.

Another hardware solution, called Address Translation, is presented to the use of Address Virtualization. Enabling several virtual sources of several OSs to share the same shared hardware, supporting large amount of virtual users of different types and demands.

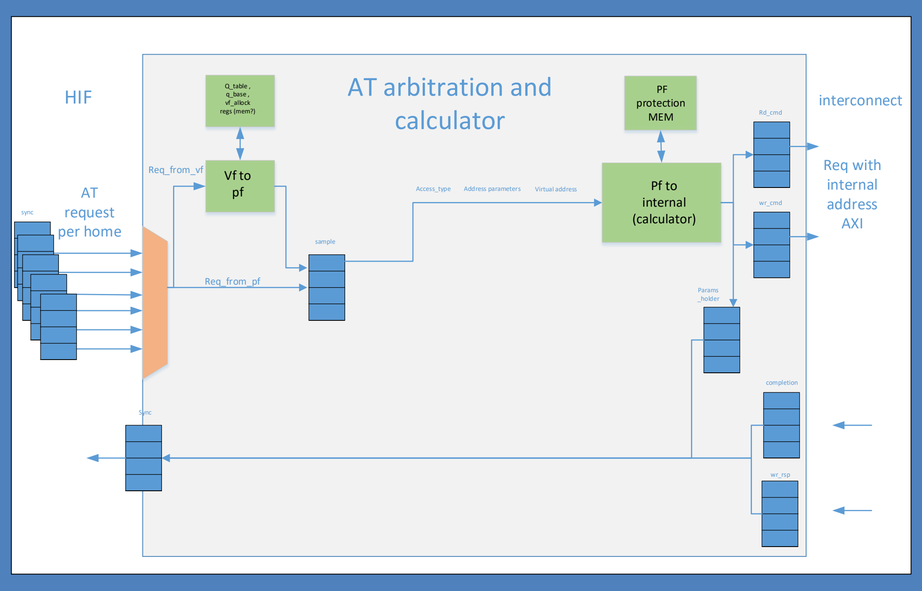
The method uses two main hardware mechanisms:
(1) Pipe Line architecture achieves top performance and divides the translation into stages. (2) Parallel logic in order to calculate many translations at the same time and match the correct result.



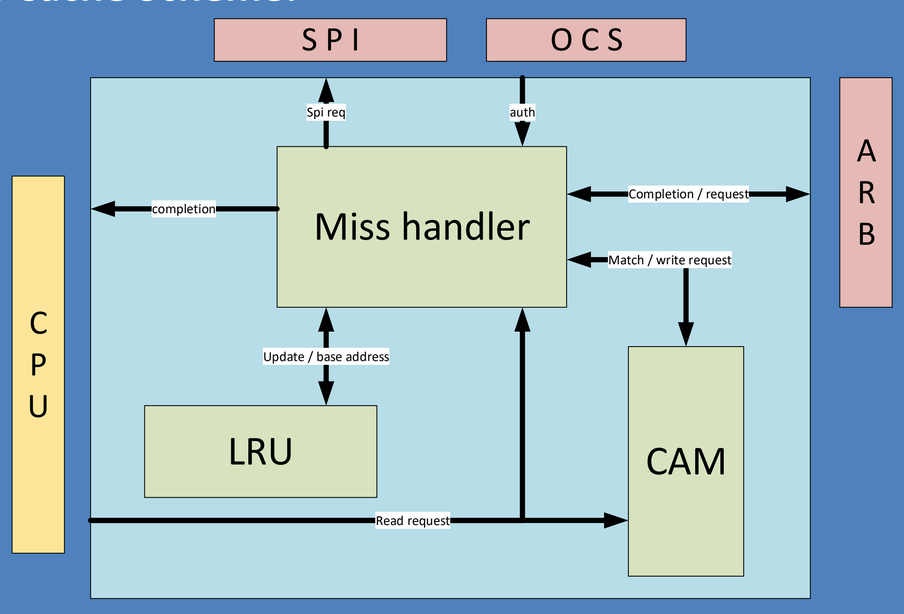
Methods

We designed the following block schemes in order to address the solution:

Address Translation Scheme:



IRAM Cache Scheme:

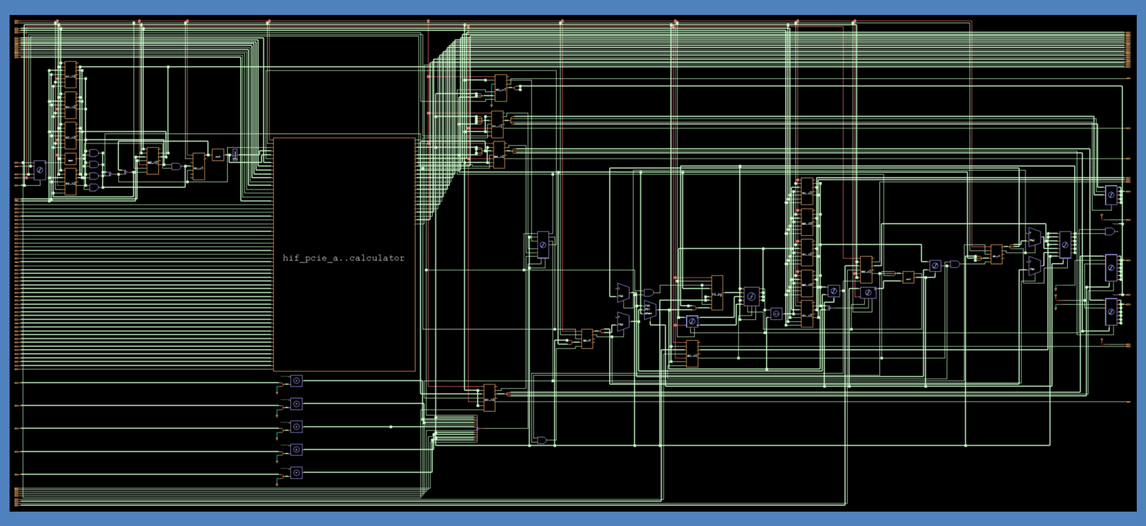


Results

The project was written in Verilog (hardware description language). And was synthesized to the following schemes:

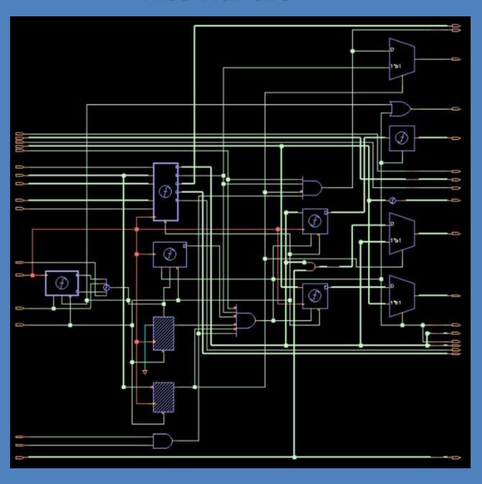
Address Translation actual area times 1,000:

Address Translation Scheme:

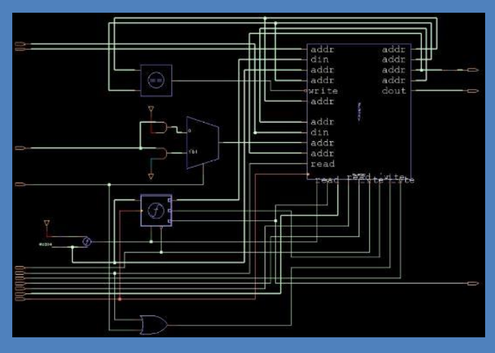


IRAM Cache Scheme:

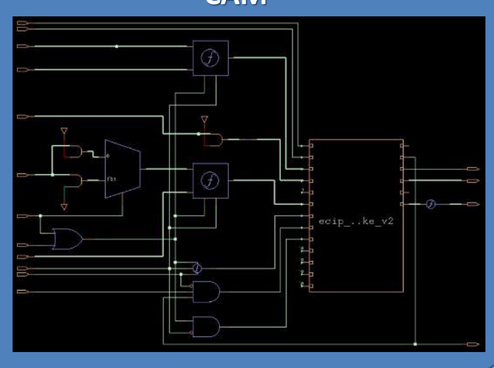
Miss Handler



LRU



CAM



IRAM Cache actual area times 1,000:

Conclusions

Both of the components that we designed are most valuable for the next generation of NICs manufactured by Intel Corporation in the sense of their functionality and enablement. the above mechanisms allow NICs clients to move to multi-core architecture, higher flexibility with product updates and so enabling more management features. Supporting all these was done without harming the latency and with a clock cycle faster in about 30%.

Acknowledgments

We would like to thank the DCG organization at Intel corporation, and our Design and Verification team led by Amir Paran for all the technical guidance. Mostly we would like to thank our wives Meital and Tali for supporting us during the long hours we invested in this project.