Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

ICEENG

6th International Conference
on Electrical Engineering
ICEENG 2008

Design space exploration of PISA architecture for ONU auto-discovery process

By

Alie El-Din Mady*   Andrea Tonini**   Davide Finardi***

Abstract:

The goal of the paper is to optimize the PISA architecture for the ONU Auto-discovery process. This Auto-discovery process has been written in C language following the IEEE 802.3ah MPCP standard. Using *SimpleScalar* [3] simulation tool, the architecture profile is evaluated in order to decide the range of the design exploration. Then, using *Wattch* [1] and *CACTI* [2] simulation tools the CPI, average power consumed and cache area are calculated for each design point, the cost function is defined and evaluated for each design point using greedy strategy. The Auto-discovery process has been written in VHDL and using *Synopys power compiler* [4] the power consumption has been calculated and then we compared between the VHDL implementation and the PISA architecture from the power consumption point of view.

Keywords:

Design space exploration, Ethernet Passive Optical Network, Multi-Point Control Protocol, ONU Auto-discovery process.

_____

   * ALaRI, Universita della Svizzera italiana, Lugano, Switzerland
  ** EMC SA, Bellinzona, Switzerland
*** Etnoteam Spa, Milan, Italy

1. Introduction:

The *Ethernet Passive Optical Network* (EPON) is an emerging access network technology that provides a low-cost method of deploying optical access lines between a carrier's *Central Office* (CO) and a customer site. EPON implements the concept of a *Full Services Access Network* (FSAN) that delivers converged data, video, and voice over a single optical access system.

An important functional requirement for the EPON technology is the Auto-discovery process. When the network is powered up for the first time or when a new *Optical Network Unit* (ONU) is added to the network, the Auto-discovery process informs the *Optical Line Terminal* (OLT) about the existence of the ONU and its capability. Auto-discovery is based on the information that the OLT can assign resources to the ONUs and coordinate the communication over the shared optical medium [5].

One of the optimization techniques for the Embedded Systems is design space exploration. With this technique the design is considered as a black box, and using the *Portable Instruction Set Architecture* (PISA) a wide set of design parameters can be tuned in order to extensively explore the design space. For an application we can consider: the number of cache sets, cache size, cache associatively and replacement technique for the cache, the number of arithmetic logic units (ALU), the number of floating point arithmetic logic units (FPALU), the number of multipliers (MUL), the number of floating point multipliers (FPMUL) and the branch prediction[3].

2. MPCP Auto-discovery Process:

In this section the Auto-discovery process of *Multi-Point Control Protocol* (MPCP) is briefly recalled (see Figure 1) [5]:

1. OLT broadcasts the GATE message by using the discovery gate generation process to the ONUs; using the gate reception process the ONU can detect the GATE discovery message.
2. The un-initialized ONU sends the REGISTER_REQ message to the OLT within its grant window.
3. The OLT generates REGISTER message using register generation process, this message includes the *Logical Link ID* (LLID) uniquely assigned to identify the ONU.
4. DBA registers the LLID of the registered ONU; OLT sends a point-to-point normal GATE massage carrying transmission grant times that are stored in the ONU to synchronize upload transmission opportunities.
5. The ONU finally sends a REGISTER_ACK message to the OLT to complete the registration process.
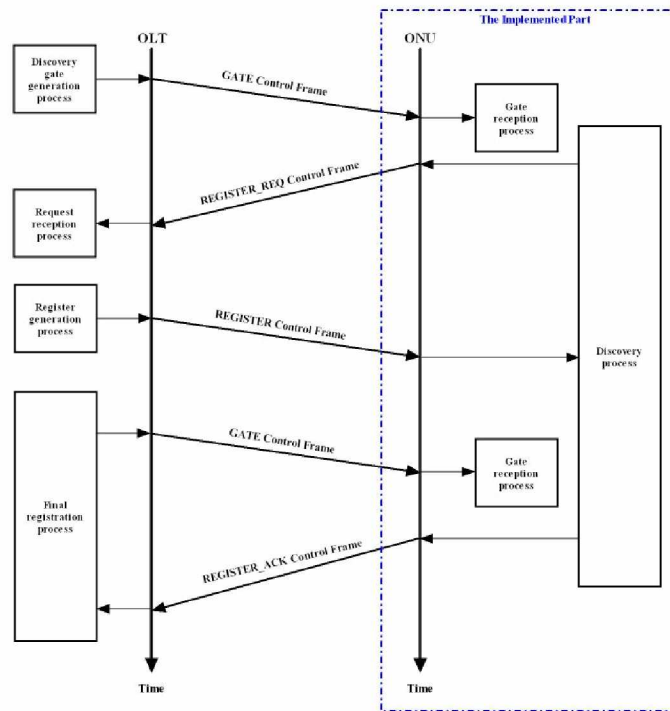
Table (1): *Auto-discovery Profiling*

3. MPCP Auto-discovery Profiling:

Using *Sim-Profile* with the "-all" option gives the full profiling details as mentioned in Table 1 and Figure 2 [6]:

Figure (1): *Auto-discovery Messages Exchange*

Instruction	Absolute occurrence	Percentage%
load	15368	18.15
store	11644	13.75
uncond branch	4227	4.99
cond branch	12840	15.16
int computation	40596	47.93
fp computation	0	0.00
trap	15	0.02

Figure (2): Auto-discovery Profiling

It is obvious that floating point operations and trap are not much of interest to Auto-discovery application. So we would allocate the minimum hardware to them, but the main interest will be to allocate resources to:
1. Integer ALU operations (47.93%)
2. Load/Store operations (31.9 %)
3. Branching (20.15%)

4. Cost Function:

Area of the cache, CPI and average power per cycle are considered as three dimension of the cost function:

$$Cost\_Function = CPI \times Average\ Power\ per\ Cycle \times CacheArea \qquad (1)$$

The area of the caches was calculated using the *CACTI* simulation tool ". */cacti <csize> <cacheline> <assoc> <tech> <NBanks>*" and the CPI and the average power per cycle were calculated using the *Wattch* simulation tool. The *Wattch* simulation tool provides four power metrics based on the conditional clocking style:
1. No conditional clocking
2. Simple conditional clocking
3. Aggressive ideal clocking

4. Aggressive, non-ideal clocking ("cc3", some power still consumed when disabled)

The aggressive, non-ideal clocking was considered as it is the most realistic and optimized.

5. Exhaustive Search [3]:

In the exhaustive search the design points were explored depending on the design profile. As we mentioned in the profiling that the most interesting parameters in this application are the cache parameters and the number of the ALUs.

The selected range of cache structure

[*<no of sets>:<blocksize>:<associativity>:<Replacement Policy>*]:
1. Instruction cache level1(il1): {32,64,128}:{16,32}:{1,2,4}:l
2. Data cache level1(dl1): {32,64,128}:{16,32}:{1,2,4}:l
3. Unified cache level2(ul2): {64,128}:{32,64}:{1,2,4}:l

The selected range of ALU structure [*-res:ialu <no of ALUs>*]: {2,3}

The cost function was calculated for each design point (see Figure 3) and hence the optimal design point was evaluated with the lowest cost function as shown in Table 2.
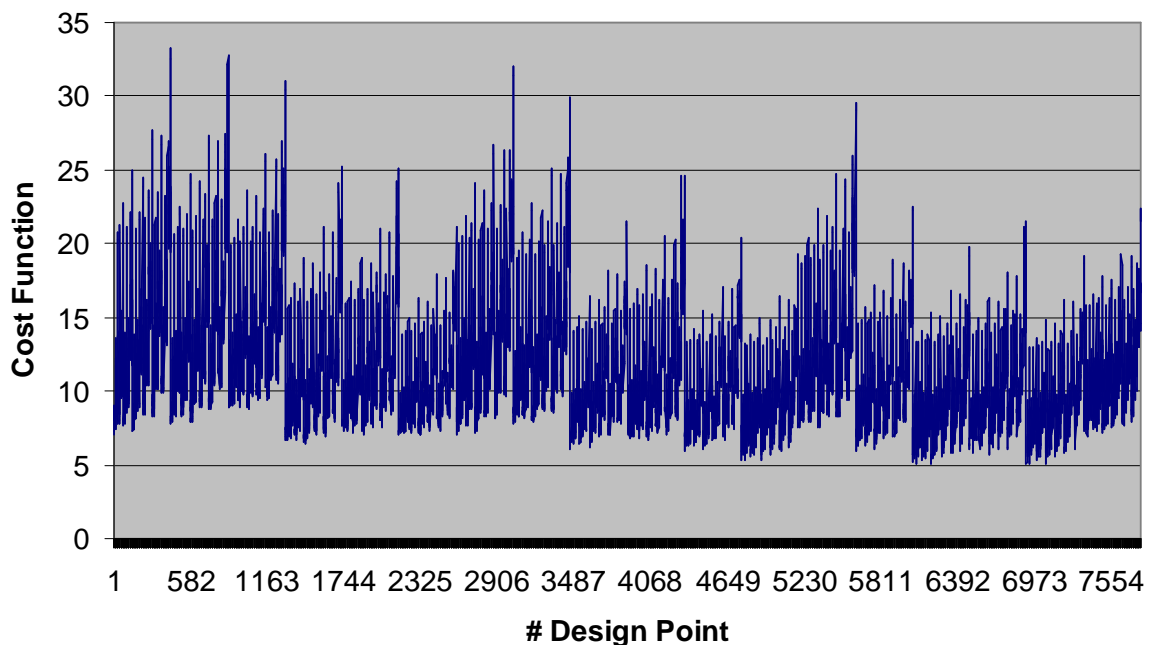

Figure (3): Cost Function for the Design Points of Exhaustive Search

Table (2): *The Optimal Design Point of Exhaustive Search*

Optimal Design Point Configuration	
ALU	3
IL1	128:16:4:l
DL1	32:16:2:l
UL2	64:32:1:l
Optimal Design Point Cost	
Cost Function	5.048099
Average Power per Cycle	11.6434
CPI	1.0559
Cache Area	0.410606

6. Reduction to a Single Objective [3]:

This search approach concentrates only on one (or several) directions of the search space and a constraint optimization. In the pervious section only the cache size and the number of ALUs were considered while the branching and the cache replacement policy were ignored.

Starting from the optimal point of exhaustive search, the cache replacement policy and the branch prediction are considered in this section (see Figure 4).

Regarding the replacement policy [<*Replacement Policy*>] there are three possibilities:
1. 'l'-LRU
2. 'f'-FIFO
3. 'r'-random

The branch prediction [–*bpred <Prediction Technique>*] offers there different possibilities:
1. Bimodal(default)
2. 2-level
3. Perfect

Hence the final optimal design point has evaluated using the reduction to a single objective approach with the lowest cost function as shown in Table 3.
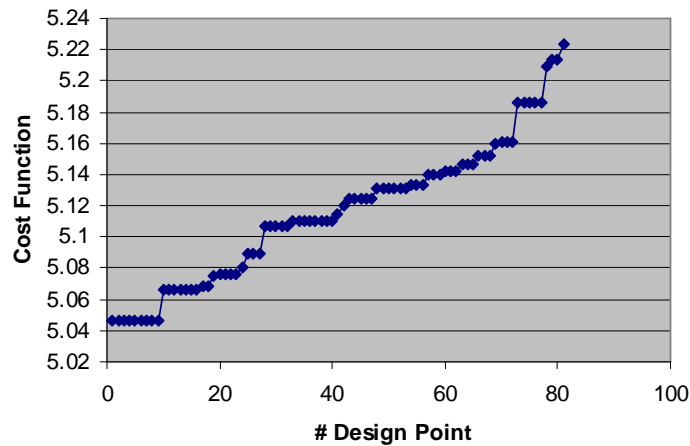
Figure (4): *Cost Function for the Design Points of Reduction to a Single Objective*

Table (3): *The Optimal Design Point*

Optimal Design Point Configuration	
ALU	3
IL1	128:16:4:l
DL1	32:16:2:l
UL2	64:32:1:l
Bpred	Perfect
Optimal Design Point Cost	
Cost Function	5.04645
Average Power per Cycle	11.6418
CPI	1.0557
Cache Area	0.410606

7. VHDL Implementation:

The Auto-discovery process has been written in VHDL and using *Synopsys Design compiler* the design has been synthesized. Total cell area and data arrival time have been calculated using *Synopsys Design compiler* and total power consumption has been calculated using *Synopsys Power compiler* for 1 Gbps and 10 Gbps bit rate (see Table 4).

Synthesis Results for 1 Gbps	
Total Power	134.8521 µW
Total Cell Area	38832
Data Required Time	3.10 ns
Data Arrival Time	3.92 ns
Synthesis Results for 10 Gbps	
Total Power	1.3360 mW
Total Cell Area	38784
Data Required Time	0.50 ns
Data Arrival Time	3.92 ns

Table (4): *Synthesis Results*

8. Result Comparison:

Since the Auto-discovery process efficiency is constant, estimated around 0.03% of the overall processing; hence, the Auto-discovery process design is independent on the design technique chosen [7].

It is obvious that the power consumption of the VHDL implementation (134.8521 µW for 1 Gbps and 1.3360 mW for 10 Gbps) is very low compared with the total power consumed by the PISA architecture (estimated total power consumption 61.817mW) and hence the dedicated design is better from the power consumption point of view.

9. Conclusions:

This paper has introduced the design space exploration in order to optimize the PISA architecture of the ONU Auto-discovery process. Exhaustive search has been primarily used to get the optimal design point and then using the reduction cost with a reduction to a single objective strategy the final optimal design point was calculated.

The optimal point has three ALUs improving CPI performance (1.0557) at the cost of higher average power (11.6418) and the branch prediction technique has improved the cost function from (5.048099 with bimodal) to (5.04645 with perfect).

We recommended in this paper using of the dedicated design (VHDL implementation) than using the PISA for the Auto-discovery process since the power consumed is much lower (61.817mW for PISA and 1.3360 mW for VHDL) and the process efficiency is constant.

References:

[1] D. Brooks, V. Tiwari and M. Martonosi, *Wattch: A Framework for Architectural-Level Power Analysis and Optimizations*, Computer Architecture, 2000 proceedings of the 27th International Symposium on, 2000.

[2] P. Shivakumar and Norman P. Jouppi, *CACTI 3.0: An Integrated Cache Timing, Power, and Area Model*, CACTI manual, 2002.

[3] S. Kunzli, L. Thiele and E. Zitzler, *Modular design space exploration framework for Embedded systems*, Computers and Digital Techniques, IEE Proceedings, Mar 2005.

[4] Synopsys, Inc, *Power Compiler: Automatic Power Management within Galaxy™ Design Platform*, Synopsys Datasheet, 2007.

[5] G. Kramer, *Ethernet Passive Optical Networks*, McGraw-Hill Professional, ISBN: 0071445625, Mar 2005.

[6] Douglas C. Burger and Todd M. Austin, *The SimpleScalar Tool Set, Version 2.0*, UW Madison Computer Sciences Technical Report #1342, June 1997.

[7] G. Kramer, *How efficient is EPON?*, Teknovus, Inc.