

Embedded Computation of Maximum-Likelihood Phylogeny Inference Using Platform FPGA

Terrence S. T. Mak and K. P. Lam
Department of System Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong
{stmak, kplam}@se.cuhk.edu.hk

Abstract

Our previous work to accelerate phylogeny inference using HW/SW(Hardware/Software) co-design has recently been extended to a more powerful embedded computing platform. In this platform, a microprocessor is immersed into field programmable gate array (FPGA) fabric for realizing an effective environment for HW/SW co-design implementation. Significant improvements in data transmission between hardware and software and higher clock frequency of FPGA have been realized when compared to the JBits interface in the previous design. In addition, the embedded platform provides a greater flexibility in partitioning hardware and software tasks. These new features lead to much faster computation speed for phylogeny inference. In this paper, the architecture for HW/SW co-design in the embedded platform is presented. The FPGA logic design for the tree likelihood evaluation has also been improved to tackle problem of larger scale by adopting the idea of partial likelihood.

1. Introduction

Field Programmable Gate Arrays (FPGA) technology is advancing at tremendous speed and has provided an important implementation platform for high performance computation in bioinformatics. For instance, researchers have built dedicated applications using FPGA for sequence homology search, genetic network search, and protein folding prediction [1, 2].

In [3], an efficient Hardware/Software (HW/SW) co-design scheme using FPGA for the implementation of GAML (Genetic Algorithm for Maximum Likelihood) phylogeny inference was proposed. The approach exploits the flexibility of software (SW) for tree topology searching using a genetic algorithm (GA), and speeds up the tree likelihood computation for fitness evaluation using FPGA hardware (HW). The HW/SW system was implemented using JBits over a parallel port for data transmission between the host PC and FPGA. Despite the significant speedup over software-only approach, the communication overhead for this loosely

coupled HW/SW has been a critical concern for realizing higher computation speed. Consider the BRAM (Block RAM) handshake protocol as specified by high-level Java code:

```
(PC -> FPGA)  jbits.setBram(...)  
(FPGA -> PC)  ramData = jbits.getBram(...)
```

The communication overhead involves the actual data transfer, as well as protocol initiation and termination during each transfer. These low-level processes are not under control of high-level Java code, and their timing cannot be precisely determined.

To circumvent some of the overhead problems, we propose an approach for mapping the HW/SW model on a tightly-coupled embedded computing environment. The following sections describe the embedded computing architecture and its relevance for maximum-likelihood phylogeny inference.

2. Embedded Computing for HW/SW Co-design

Recently, with the availability of Platform FPGA¹, new implementation models and system architectures for embedded computing have emerged. An example is the embedded hardcore of a PowerPC processor within the FPGA fabric. The platform offers great flexibility for user to define task partitioning between the FPGA hardware logic and software running on microprocessor. It also creates a tightly-coupled HW/SW computing environment that significantly reduces communication overhead under the provision of high-speed internal buses.

A central bus infrastructure² with dedicated sub-buses and interconnected bridges is essential for providing a high throughput communication gateway connecting the

¹ The Virtex II-Pro Platform FPGA [4] is used in our study. In this embedded platform, an IBM PowerPC-405 microprocessor is immersed into the FPGA fabric.

² It is referred to as the "CoreConnect" bus infrastructure in the Virtex II-Pro Platform FPGA.

microprocessor and FPGA. More specifically in [5], the PowerPC core accesses high-speed system resources (such as instructions and data) through the *Processor Local Bus (PLB)*; and *On-Chip Peripheral Bus (OPB)* provides the connectivity to the FPGA.

In mapping our HW/SW co-design for phylogeny inference to the embedded platform, the microprocessor core, internal block memory (BRAM), and FPGA computation units form the major components of the HW/SW system. Software code for the Genetic Algorithm (GA) is stored as instructions in the BRAM, which is being accessed by the PowerPC via the PLB bus.

The FPGA computation units provide fine-grained parallel computation for the tree likelihood evaluation. They work as dedicated peripheral controlled by the microprocessor via the OPB bus. DNA sequence data are stored in the internal BRAM accessible only by the FPGA computation units, because the sequence data are required only for computing the likelihood value. This results in greater overall system performance.

Figure 1 shows our architecture for the interface logics in coordinating the different types of data flow between the FPGA and OPB. The interface logics are needed to define the specific bus width for different buses. Data bus is the one with the highest bandwidth, and is directly connected to the internal BRAM through the memory mapping logics. This bus is mainly used for the likelihood value and tree topology data that require a wide bus. The control bus and acknowledgement bus are directly connected to the on-chip registers instead of the BRAM. In this way, the microprocessor can control the FPGA logics more efficiently. A typical C program can define the handshake protocol with low-level bus control, as follows:

```
(PowerPC -> FPGA)
    XGpio_DiscreteWrite(&addrbus, 1)
    XGpio_DiscreteWrite(&databus_out, ...)
(FPGA -> PowerPC)
    likelihood = XGpio_DiscreteRead(&databus_in)
```

In contrast to the JBits protocol discussed previously, the direct internal bus access to memory and register reduces much of the communication overhead in data transfer for embedded computation. For instance, using a FPGA clock frequency of 100 MHz on a 32-bit wide OPB can attain a data transfer rate of 400 Mbytes/s.

The embedded platform provides an effective way of communication between the FPGA hardware and microprocessor software. The memory mapping logic can readily convert data types into FPGA fixed-point format. With the defined bus architecture, the microprocessor thus efficiently control the FPGA computation units by reading and writing values via the appropriate internal buses

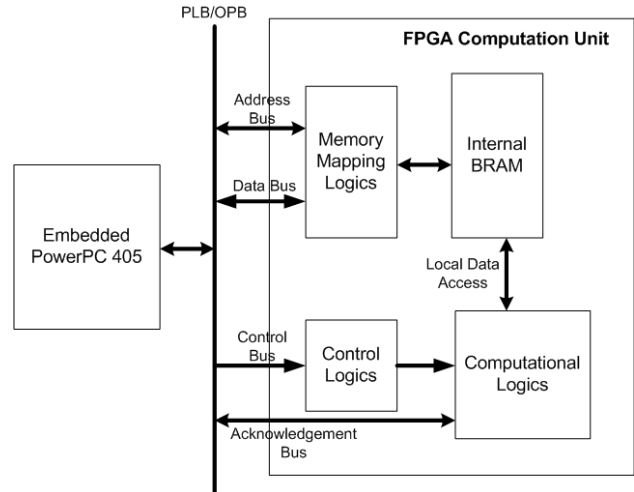


Fig. 1 Interface logic architecture

3. Maximum Likelihood Evaluation and Task Partitioning

A major objective of HW/SW co-design is to obtain higher system performance through task partitioning and algorithmic implementation in hardware and software. As with the previous approach [3], embedded computation of the GAML is partitioned into two main tasks: software implementation for the genetic algorithm (GA) in the microprocessor (PowerPC); and hardware realization for the likelihood evaluation in FPGA.

PowerPC Software Implementation

In our current design, we consider an *unrooted n-taxa*³ phylogenetic tree as a modified binary tree data structure. All internal nodes have two children, while the root has three. Using binary tree operations, the software searches the tree topology based on the TBR (Tree Bisection and Recombination) and recombination, which perform GA mutation and crossover, respectively [6]. Through these operations the tree search space can be effectively explored.

A post-order tree traversal is also implemented in software. It generates a tree topology matrix for the subsequent FPGA computation of partial likelihood evaluation in parallel. The tree topology matrix has five attributes: the first is the node index uniquely assigned to each DNA sequence (second attribute); the next two are the children node indexes; and the leaf attribute. This matrix is readily written to the internal BRAM, while the branch lengths are stored in another array associated

³ An unrooted *n-taxa* tree refers to a tree with *n* leaf nodes (or taxa) and *n-2* internal nodes. Internal nodes and leaf nodes are of degree 3 and 1, respectively. The degree of a node is the number of branches (or links) directly connected to the node.

with the tree topology matrix. The internal BRAM thus serves an essential storage function in the handshake protocol between the software and FPGA hardware. The software program outputs the tree topologies to the FPGA computational units, while the hardware returns the likelihood values of each tree to the software.

FPGA Hardware Implementation

The tree likelihood evaluation is computational intensive, as numerous multiplications⁴ of conditional probabilities are needed. The FPGA hardware can provide a fine-grained (in terms of multipliers, adders, and simple logic) parallelization for the required computation. However, a brute-force implementation (as in our previous work [3]) is rather resource demanding to meet the *exponential* increase in computation (with respect to n) for large scale problem.

Using Felsenstein's idea [9] on node pruning and the related recursive formulation of partial likelihood [10], our current FPGA design (Figure 2) has significant improvements over the brute-force approach by substantially reducing the required number of multiplications⁵. In following the computational order specified in the post-order tree traversal, partial likelihood of all the tree nodes can be evaluated iteratively. This results in a more efficient likelihood computation scheme for large scale problems.

4. Results and Discussion

We have implemented the GAML based on the proposed embedded platform for the HW/SW co-design model. In our preliminary testing, embedded computation was found to offer significant improvements over the previous PC-FPGA design [3]. For an unrooted 4-taxa case of computing the phylogeny of DNA sequences with 500 nucleotide sites, the embedded platform completed 100 GA iterations in 1.87s, while each likelihood evaluation took 5.77ms. This compares favorably with the previous results [3] for the rooted 4-taxa case (10 minutes for 100 GA iterations and 0.21s for likelihood evaluation). Apparently, the reduced communication overhead and the parallel partial likelihood computation scheme are two major factors contributing to this significant improvement. Embedded computation also offers much flexibility for studying various design alternatives of task partitioning in

⁴ For n DNA sequences each with l nucleotide sites, the likelihood computation for the n -taxa unrooted tree requires $4^{n-2}(2n-4)l$ multiplications.

⁵ The required number of multiplications is $(an+b)l$, where a, b are constants determined from the specific tree topology and the parallelization scheme chosen.

HW/SW co-design. For example, instead of implementing Jukes-Cantor (JC) model [8] in FPGA, complicated nucleotide substitution models can be implemented using efficient software for matrix spectral decomposition. On the other hand, post-order tree traversal may be more readily implemented in FPGA hardware instead of software.

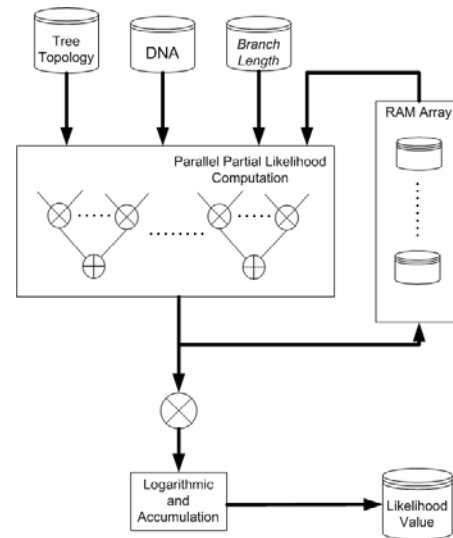


Fig. 2 Hardware implementation for tree likelihood evaluation

References

- [1] Yamaguchi Y., Miyajima Y., Maruyama T., Konagaya A., "High Speed Homology Search using Run-time Reconfiguration", *12th Intl Conf on Field Programmable Logic and Applications*, France, 2002
- [2] TimeLogics, in web <http://www.timelogic.com/>
- [3] Mak T. and K. P. Lam, "High Speed GAML-based Phylogenetic Tree Reconstruction Using HW/SW Codesign", *Proc. of IEEE Computer Society Bioinformatics Conference (CSB'03)*, pp. 470-473, 2003
- [4] "Virtex-II Pro™ Platform FPGA Handbook", *Xilinx*, 2002
- [5] IBM CoreConnect Bus Architecture, in web http://www-3.ibm.com/chips/techlib/techlib.nsf/products/CoreConnect_Bus_Architecture
- [6] Lewis P., "A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data", *Molecular Biology Evolution*, 15(3):277-283, 1998
- [7] Mak T. and K. P. Lam, "FPGA-based Computation for Maximum Likelihood Phylogenetic Tree Evaluation", in *Field-Programmable Logic and Applications Conference*, 2004
- [8] Jukes, T. H. and Cantor, C. H. "Evolution of protein molecules", In Munro, H. M. (Ed.) *Mammalian Protein Metabolism*. Academic Press, New York, pp. 21-123. 1969
- [9] Felsenstein, J., "Evolutionary trees from DNA sequences: a maximum likelihood approach", *J. Mol. Evol.*, 17:368-376, 1981
- [10] Adachi, J. et al., "MOLPHY version 2.3, program for molecular phylogenetics based on maximum likelihood", Tech. Report, The Institute of Statistical Mathematics, Tokyo, Japan, 1996.