Journal of Systems Architecture 63 (2016) 33–47

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Comparative performance evaluation of latency and link dynamic power consumption modelling algorithms in wormhole switching networks on chip

James Harbin*, Leandro Soares Indrusiak

Real-Time Systems Group, Department of Computer Science, University of York, York, United Kingdom

ARTICLE INFO

Article history:
Received 20 April 2015
Revised 27 August 2015
Accepted 13 January 2016
Available online 22 January 2016

Keywords:
Network on chip
Transaction level modelling
TLM
NoC modelling
Simulation models
Dynamic power consumption

ABSTRACT

The simulation of interconnect architectures can be a time-consuming part of the design flow of on-chip multiprocessors. Accurate simulation of state-of-the art network-on-chip interconnects can take several hours for realistic application examples, and this process must be repeated for each design iteration because the interactions between design choices can greatly affect the overall throughput and latency performance of the system. This paper presents a series of network-on-chip transaction-level model (TLM) algorithms that provide a highly abstracted view of the process of data transmission in priority preemptive and non-preemptive networks-on-chip, which permit a major reduction in simulation event count. These simulation models are tested using two realistic application case studies and with synthetic traffic. Results presented demonstrate that these lightweight TLM simulation models can produce latency figures accurate to within mere flits for the majority of flows, and more than 93% accurate link dynamic power consumption modelling, while simulating 2.5 to 3 orders of magnitude faster when compared to a cycle-accurate model of the same interconnect.

1. Introduction

As the number of cores upon on-chip multiprocessors and system-on-chip (SoC) devices has increased, inter-core communication has become a critical design issue. The design architecture of the NoC (network-on-chip) is a vital factor in performance tuning, given the large influence it has upon communication latency and power consumption. As a result of the highly dynamic nature of application traffic and the potential for interactions between traffic during transmission, most design flows use simulation rather than static analysis to evaluate the power and latency performance delivered by a candidate NoC architecture. NoC interconnect simulation (as distinct from the full system simulation including execution of code upon processing elements) has been identified as an important research issue [1]. The design space of viable NoCs for multicore or SoC problems spans a wide range of candidate architectures and topologies, and is further expanded by the possible variability in application task mapping decisions. Particularly during the early stages of the design process, it is important to accelerate NoC simulation with as little impact upon accuracy as possible, allowing the design space to be explored rapidly. Therefore, methodologies other than cycle-accurate simulation are promising as candidates to rapidly explore the NoC design space.

This paper specifies and evaluates a family of NoC simulation models which are both fast and accurate in comparison to cycle-accurate references. Two NoC architectures are considered which can be accurately described using transaction-level modelling (TLM). The models assume delay-sensitive applications that have certain timing constraints, and therefore the application model includes priorities used in arbitration decisions. Since these application models typically require one flow to be prioritised over another, priority preemptive NoCs following the example of QNoC [2] are the first architecture assumed. However, given that priority preemptive NoC architectures have higher silicon area requirements for implementation, a non-preemptive architecture is also considered and evaluated.

The definition of TLM assumed in this work is that of Cai and Gajski [3], in which components are either transaction initiators, targets or interconnects. The relationship of our models to the TLM definitions specified by Cai and Gajski is considered in Section 5. Compared to cycle-accurate models, the proposed TLM algorithms are simplified to reduce the frequency of simulation events. Events are generated only upon flow admission, flow removal or when

* Corresponding author. Tel.: +44 1904325550; fax: +44 1904 325599.
E-mail addresses: james.harbin@york.ac.uk (J. Harbin), leandro.indrusiak@york.ac.uk (L. Soares Indrusiak).

http://dx.doi.org/10.1016/j.sysarc.2016.01.002
1383-7621/

simulation state must be updated to ensure consistency. Removing the necessity to model low-level details such as the progress of every data flit through arbiters, routers and other simulator-level elements permits the reduction of simulation event count by orders of magnitude.

A fine-grained cycle-accurate model can offer precise simulation of the NoC internals, including the occupation and free status of particular buffers. However, in order to improve execution time performance and reduce simulation algorithm complexity, the buffer occupation of intermediate routers is not considered within the TLM models described in this paper. Even under the design structure of a transaction level model, several design choices are possible regarding the abstraction levels chosen, with resulting implications for timing performance and accuracy. In our earlier work [4–6] the entire route was be treated as a single unified abstraction when making contention decisions. Although this modelling approach is simple and its execution timing performance favourable, heavy contention requires a more fine-grained approach to improve timing accuracy.

The major novelty in this paper is the presentation of the TLM model TLM-NPD, which provides a finer locking granularity at the level of individual links and the ability to model flow behaviour in single-cycle increments in case of contention. The further intent of this paper is to comparatively assess and evaluate the simulation latency accuracy and execution time performance of our family of transaction level models TLM-PRE [4,5], TLM-NP [6] and TLM-NPD, compared to reference cycle-accurate implementations. These evaluations are performed with test cases incorporating two application models and with synthetic traffic.

The paper is structured as follows. Section 2 surveys the literature on TLM for NoCs, comparing and contrasting the approaches presented with the present work. Section 3 motivates the work by describing the difficulties in accurate latency prediction, particularly in non-preemptive NoCs. Section 4 describes the NoC scenarios, specifying the synthetic and application traffic models used in the evaluation results. Section 5 specifies in detail the family of TLM models evaluated in the work, typically via pseudocode implementations. Section 6 evaluates the accuracy and execution time performance of the implementation under a variety of traffic models, and provides a discussion of the comparative merits of the various models in view of the results. Finally, Section 7 details potential extensions to the current work, and Section 8 concludes the paper.

2. Literature review

The goal of transaction level modelling is to improve simulation speed by the abstraction away of low level events such as individual flit transmissions, in favour of boundary events. TLM is frequently associated with SystemC [7] although the methodology is suitably generic to be applied to other languages and simulation frameworks. The TLM 2.0 [8] framework models a VLSI system such as a NoC or SoC as groups of transaction initiators or targets (communicating nodes) and interconnects which transfer transactions from initiators to targets. In [9], SystemC TLM models are used for NoC simulation by treating the transmission across multiple arbiters as a single transaction. However, since blocking delays upon the path are only estimated statistically, accuracy may be compromised in complex application models.

Schirner and Dömer [10] investigate the tradeoff between TLM accuracy and simulation speed, finding that TLM may potentially be four orders of magnitude faster than cycle-accurate models. However, the work introduces simplifications that reduce accuracy, with a potential average inaccuracy of 35% reported in timing user transactions for their most abstract TLM model. TLM models have been applied to the individual processing elements, and can retain accuracy if using a granularity larger than individual instructions (assisted by an earlier cycle-accurate static analysis phase). The approach presented in this paper is distinct from this earlier work, as our work involves the application of TLM to the NoC and not code execution on the PEs.

Bus TLM modelling is considered in Result-Oriented Modelling (ROM) [11] which optimistically predicts transaction delays, and retroactively corrects in the case of contention. ROM can provide error-free timing prediction, however, frequent cascading corrections produce a reduction in simulation event speed and require an increase in modelling complexity. Considering TLM models for on-chip interconnects, timing points can be identified from the protocol specifications of the bus [12]. Simulation speed improvements of up to two orders of magnitude can be obtained while retaining accuracy in comparison to a cycle-accurate model. This approach relies on accurate identification of timing points from the bus protocol specification, which is difficult as interconnects become more complex. In contrast, our approach discovers preemption points for simulation dynamically during simulation, from the arrival of contending traffic from the application model.

In [13], a speed-up of 50 times for the TLM models predicting NoC interconnect latency compared to the cycle-accurate reference is shown, with accuracy of 99.9%. This is obtained by using local time references for individual tasks communicating over the NoC, and only synchronising when tasks are common initiators or targets of a single transaction. Modifications to the simulation kernel to use lightweight schedulers [14] with a common time reference were shown to produce a 38% speed up. However, this approach requires simulation kernel modifications, which our approach does not require. In another approach [15], simulation parallelisation has been explored to take advantage of multiple CPU cores on the host simulation machine. By effectively dividing the independent tasks, a speedup almost linearly proportional to the number of cores can be demonstrated.

Existing work has covered simulation of wormhole NoCs [16], which reduce the total number of simulation events by simulating only packet headers and trailers. Our current approach requires the simulation of the progression of all flits since it is necessary to register their power consumption impact. Previous works by the current authors introduced a family of fast TLM algorithms which are further clarified in Section 5 and evaluated with new results. A TLM algorithm for priority preemptive NoCs [4] is referred to as TLM-PRE within this paper. This model was studied and evaluated for its power consumption accuracy in preemptive NoCs [5]. A fast non-preemptive TLM model was applied to application task mapping in [6]. This non-preemptive model is referred to as TLM-NP in this paper. A more advanced non-preemptive NoC model was presented in [17], although the model presented in this current paper as TLM-NPD incorporates significant alteration to its internal model of how flows advance through the network, in order to improve latency prediction performance.

3. Problem description

It is likely that cycle-accurate simulation of NoC interconnect data transmission will prove prohibitive for future realistic application cases, particularly when evaluating a wide design space. In our earlier work on assessing and improving NoC simulation algorithm execution speed, cycle-accurate simulation of 2 s of execution of a target application required approximately 10 min [5]. Since the cycle-accurate framework operates at flit granularity, simulator events are required every time a flit advances through an architectural entity such as an arbiter or buffer. This not only scales in proportion to the amount of data transmitted, but leads to wasteful overheads in simulator state management and event