Applying Dataflow Analysis to Dimension Buffers for Guaranteed Performance in Networks on Chip

Andreas Hansson\(^1\), Maarten Wiggers\(^2\), Arno Moonen\(^1\), Kees Goossens\(^3,4\) and Marco Bekooij\(^4\)
\(^1\)Eindhoven University of Technology, Eindhoven, The Netherlands
\(^2\)Twente University of Technology, Enschede, The Netherlands
\(^3\)Delft University of Technology, Delft, The Netherlands
\(^4\)Research, NXP Semiconductors, Eindhoven, The Netherlands
m.a.hansson@tue.nl

Abstract

A Network on Chip (NoC) with end-to-end flow control is modelled by a cyclo-static dataflow graph. Using the proposed model together with state-of-the-art dataflow analysis algorithms, we size the buffers in the network interfaces. We show, for a range of NoC designs, that buffer sizes are determined with a run time comparable to existing analytical methods, and results comparable to exhaustive simulation.

1 Introduction

A growing number of applications, often with real-time requirements, are integrated on the same System on Chip (SoC), in the form of hardware and software Intellectual Property (IP). Applications are split into tasks, and it is the onus of the interconnect to facilitate the real-time requirements of the inter- and intra-task communication.

Networks on Chip (NoC) offer latency and throughput guarantees [6,8]. The guarantees depend on the arbitration in the routers and Network Interfaces (NI), but also on the NI buffers. These decoupling buffers absorb differences in speed and burstiness between the IP and the NoC [5], and thereby hide network internals, such as packetisation, arbitration, and end-to-end flow control [2,9]. If the NI buffers are not sufficiently large, the guarantees are violated. The size must, however, be minimised, as the buffers are a major contributor to NoC power and silicon area [3].

Existing approaches to dimension NI buffers [3,5] are based on linear bounds [5], resulting in a low run time but large buffers, or exhaustive simulation [3], with smaller buffers but a run time of several days for larger SoC designs. In this work, we model the NoC and the IP using a dataflow graph. In contrast to [3,5], that are based on network calculus, dataflow analysis cannot only dimension the buffers given the temporal requirements, but also determine the temporal behaviour of the SoC for given buffer sizes, e.g. to analyse if new applications fit on an existing NoC.

As the main contributions of this paper, we: 1) show how to construct a dataflow graph for a NoC communication channel, 2) use this model with state-of-the-art dataflow analysis techniques [10] to dimension the NI buffers. The run time is comparable to existing analytical methods, and the results are comparable to exhaustive simulation.

Section 2 describes the proposed channel model. In Section 3, we apply dataflow analysis [4,10] to determine conservative bounds on the NI buffer sizes. Finally, conclusions are drawn in Section 4. More details are found in [7].

2 Channel model

We use Cyclo-Static Dataflow (CSDF) [1] models to compute buffer sizes. A CSDF graph is a directed graph, consisting of actors connected by edges. An actor has distinct phases of execution, and synchronises by communicating tokens over edges. An actor is enabled to fire when tokens are available on all its input edges and transitions from phase to phase in a cyclic fashion.

The proposed channel model is shown in Figure 1. In the figure, \(n \times 1\) denotes a vector of ones of length \(n\), and the italic symbol \(i\) denotes a vector of ones of appropriate length. The Response Times (RT) [1] of the individual actors appear above and below the actors. Similar to [3,5], the model is based on the notion of a producer and consumer, connected by a forward channel that carries data and a reverse channel that carries end-to-end flow-control credits. The buffers of the channel are represented by \(\beta_p\) and \(\beta_c\).

Our method allows any CSDF model of the IP, but to enable a comparison with existing models, the IP behaviour is described by a period of \(p_p\) and \(p_c\) cycles, and a burst size of \(b_p\) and \(b_c\) words, for producer and consumer, respectively. The model reflects that only one word can be produced per cycle, thereby reducing the resulting buffer sizes.

In this work, we model the Æthereal NoC [6] that uses time-division multiplexing (TDM) to provide latency and throughput guarantees. The model has five parameters, the period of the TDM wheel \(p_n\), and four parameters related
to the allocated resources: the forward and reverse path, $\phi_f$ and $\phi_r$, plus the time-slot allocation in the two directions, denoted $t_f$ and $t_r$. The throughput and latency of the NI is determined by $t_f$ and $t_r$. The functions $\theta_{d}, \rho_{d}^{-1}, \theta_{h}$ and $\rho_{h}^{-1}$ conservatively bound the latency and rate for data and credits respectively. The router network is modelled as a latency only, given by the path and the function $\theta_{p}$.

The aforementioned bounding functions are determined by the NoC architecture, and include e.g. packetisation overhead, pipelining delay and arbitration. While the parameters and functions used in Figure 1 are specific for the Æthereal NoC, the model is applicable as long as the arbiters that are applied in the NIs and routers can be characterised as latency-rate servers [11], e.g. [2] and [8].

### 4 Conclusions

The latency and throughput guarantees of Networks on Chip (NoC) depends on appropriately sized decoupling buffers in the network interfaces, situated between the Intellectual Property (IP) modules and the router network. Existing buffer-sizing methods are based on network calculus and rely on coarse linear bounds or exhaustive simulation, resulting in either large buffers or impractically long run times. In this work, we propose to capture the behaviour of the NoC and the IPs using a dataflow model. The presented model is an important step in enabling the use of dataflow analysis for NoC resource allocation. The proposed method is evaluated by comparing with existing buffer-sizing approaches on a range of SoC designs. Buffer sizes are determined with a run time comparable to existing analytical methods, and results comparable to exhaustive simulation. For larger SoC designs, where the simulation-based approach is not practical, our approach finishes in seconds.

### References