Mohamed Abdelfattah <mohsaied.88@gmail.com>

Your FPGA 2015 Submission (Number 70)

1 message

dchen@illinois.edu <dchen@illinois.edu> Thu, Nov 13, 2014 at 1:01 AM
To: mohamed@eecg.utoronto.ca

Dear Mohamed Abdelfattah:

On behalf of the FPGA 2015 Program Committee, I am delighted to inform you that the following submission has been accepted to appear at the conference as a full paper:

Take the Highway: Design for Embedded NoCs on FPGAs

We received a total of 102 submissions. After a rigorous reviewing process, we accepted 21 long papers and 15 short papers, which include Designers' Track papers.

The Program Committee worked very hard to thoroughly review all the submitted papers. Please repay their efforts, by following their suggestions when you revise your paper.

Here are the formatting and preparation instructions for your paper to be included in the proceedings:

http://www.sheridanprinting.com/typedept/fpga2015.htm

Kindly review the templates and formatting guidelines. Please familiarize yourself with the formatting and review in general for the helpful formatting hints included.

The submission deadline for your paper is December 17th, 2014. No extensions will be given. Kindly attend to your submission early if you have travel planned near the submission deadline.

You would expect two emails:

1. One email from ACM Rightsreview (rightsreview@acm.org) with a link to the electronic ACM copyright-permission form(s) to be completed. Upon completing the electronic form, you will receive a confirmation email. Contained within the confirmation email will be the ACM copyright-permission block text, conference data, and DOI string/url specific for your submission and mandatory to appear on the first page. See formatting help in item #1 on how to include this text and minimize the space impact on the first page.

2. One email from Sheridan Communications (acm@sheridanprinting.com). This email will contain your assigned submission id#, link to the formatting instructions (same as the link above), and a unique-link to submit your final publication ready version on or before the submission deadline.

The reviews and comments are attached below. Again, try to follow their advice when you revise your paper.

Congratulations on your fine work. If you have any additional questions, please feel free to get in touch.

Best Regards,

Deming Chen
Program Chair

FPGA 2015

==========

FPGA 2015 Reviews for Submission #70

==========

Title: Take the Highway: Design for Embedded NoCs on FPGAs

Authors: Mohamed Abdelfattah, Andrew Bitar and Vaughn Betz

==========
REVIEWER #1
==========

Reviewer's Scores

Technical Contribution and Quality (1-6): 6
Originality: 5
Readability: 6
Relevance: 6
Confidence: 5
Overall: 10

Comments

This paper describes the results of experiments with NoC in FPGA.

This paper is strong overall, with a few limitations.
Section 2, define VC before use.

In order to get the full bandwidth, the NoC must be shared, which implies congestion will be an issue. Contention among network accesses limits such network bandwidth to something below the maximum. An application that tested this or an experimental setup that allowed experimentation with different data sources and sinks would give valuable data.

==========
REVIEWER #2
==========

Reviewer's Scores

Technical Contribution and Quality (1-6): 5
Originality: 4
Readability: 6
Relevance: 6
Confidence: 3
Overall: 9

Comments

Good case study on several designs demonstrating the benefits of embedded NoCs in FPGAs. Well written and interesting.

One concern is that one of the designs (JPEG, streaming and latency-sensitive) would have trouble with any sort of collisions in the NoC. In other words, it appears the authors have assumed a Quality of Service of 100% to make the streaming happen seamlessly. Given that, it's not clear what benefit there is to the NoC over, say, directly connecting up the system with sufficient pipelining.

I would have liked to see more discussion around how the NoC is designed specifically for the FPGA. How does the fact that it's embedded in an FPGA (as opposed to an ASIC or other fixed system) impact your NoC decisions? That's not clear.

Also, there was no mention of the tracks required to build the NoC. Fewer long lines may be required, thanks to the presence of the NoC, but the NoC itself requires another set of long wires. Was the regular interconnect stripped down to create tracks for the NoC? Was the interconnect tile able size reduced? If so, what was the impact on area and performance?

==========
REVIEWER #3
==========

Reviewer's Scores

Technical Contribution and Quality (1-6): 6
Originality: 5
Readability: 6
Relevance: 6
Confidence: 5
Overall: 10

Comments

Building on prior work that has investigated the introduction of embedded networks on chip (NoCs) into an FPGA's routing fabric, this paper explores the architectural implications of the interface between the programmable fabric and the embedded NoC. The paper describes FabricPort, one such interface, and investigates the requirements of both latency-sensitive and -insensitive applications to the upgraded architecture, along with multicase routing.

I like the topic, and I find the paper to be well written and technically sound. I tend to be a very cranky reviewer, so the absence of negatives is essentially a very big positive, even if the tone of my review doesn't sound particularly enthusiastic.

I'd like to know a little bit more about the RTL2Booksim platform that was used for experimental evaluation here, but I realize that there isn't room for it. Maybe a separate paper?

The places where I would focus on improving the paper are as follows:

(1) The paper doesn't completely describe the issue of stalls at the boundary port. The paper more or less makes claims that they won't be a problem, but doesn't give the level of detail that I would want to see; additionally, it is unclear if there were any stress-tests to evaluate the accuracy of hte claims.

(2) I would like to see a little bit more discussion of how multiple packets are merged into one flit.
—

Minor issues:

Figure 4 is somewhat awkward, as subfigure (a) flows from left to right, while subfigure (b) flows from right to left.

Figure 4 appears out of order in the paper. There is a nice LaTeX package that can automatically fix this issue: fixltx2e. My Ph.D. students swear by it. (I swear by MS Word — and do so in shame).

==========
REVIEWER #4
==========

Reviewer's Scores

Technical Contribution and Quality (1-6): 6
Originality: 5
Readability: 5
Relevance: 6
Confidence: 5
Overall: 10

Comments

This is an excellent paper, very detailed and complete.

The paper describes implementation of NOC on an FPGA as an addition to the traditional FPGA routing. There is a ton of detail - how to connect the NOC into the FPGA, the sizes of the datapaths and analysis of them, the control logic for the NOC, and discussion of packet ordering dependencies.

The authors show two case studies: first is a set of JPEG compression (not as interesting) and second is an Ethernet switch able to achieve over 800 MHz, which is very interesting.

One thing that would have been nice to see is the cost-benefit on the inclusion -- what happens to the design that doesn't manage to use the NoC? The 150 wires
added to make the NOC need to be subtracted from the general-purpose routing. An experiment that shows a design with the greater FPGA size vs. the FPGA+NOC with the smaller FPGA but NOC would be a pretty interesting add.

Overall very nice paper

==========
REVIEWER #5
==========
Reviewer's Scores

Technical Contribution and Quality (1-6): 5
Originality: 5
Readability: 6
Relevance: 6
Confidence: 6
Overall: 9

This paper describes the design of a NOC specifically targeted to FPGAs. While based on a standard efficient NOC, it incorporates features that are required for FPGA applications.

I found the paper very readable, with good background and a good, detailed presentation of the design. Particular attention is paid to the requirements of FPGA designs, in particular, for latency constrained streams, which dominate current designs. A strong case is made for latency-insensitive designs, especially for constructing systems from large components, which can take better advantage of a switched interconnect.

I liked the attention to detail with the FabricPort and the use of virtual channels to allow multiplexing of streams to allow (a limited amount of) deadlock avoidance. What do you do if the fabric presents invalid flits, for example by leaving out Tail bits?

I was not quite sure of your credit-based flow control. I assume that flow control is performed between neighboring routers on a hop-by-hop basis. But I was wondering why it needs 10 clock cycles when the roundtrip must be much less than that. I assume this means that every output port has a buffer at least 10 deep per VC.

The paper starts with the premise that a single link should be able to handle the bandwidth of the highest bandwidth interface, in this case DDR3 1067 that works out to 150 bits at 1.2 GHz, for 22.5GB/sec. While one can make a case for this, it is difficult to see how these links can be utilized by most designs. One can argue that it is a good way to distribute data from fixed interfaces throughout the fabric, but most applications rely on lots of relatively thin pipes 32/64 bits, not really fat pipes (600 bits in this case).

The paper discusses the idea of a Permapath that allows latency sensitive streams. This would seem to severely constrain the utilization of the network given that the network must use oblivious dimension-order routing. Can you comment on this?

The JPEG study was interesting as it pointed out the difference between short and long distance communication. Clearly the NOC works very well when there is high-bandwidth, long-distance communication. One might say that this relaxes the constraints on system floorplanning and placement. However, I think your example is extreme, which is valid for looking at the worst case, but which may mislead in terms of actual benefit.

The Ethernet switch is probably the best example of a design that can leverage this NOC. The problem with this example is that the transceivers are said to each be connected to a Router, but assuming the transceivers are placed at the left and right edges, half of the connections will be very long, exactly what the NOC is trying to solve. Returning to the decision of how to provision the NOC, one would conclude that there needs to be an NOC router near every bank of transceivers (x4 10G) plus a router near every memory controller. Can you comment on this problem where hardening logic means that resources are fixed and cannot be redeployed based on the application?