# Bridging the FPGA-GPGPU Efficiency Gap

Christopher W. Fletcher[1,2], Ilia Lebedev[2],
Narges B. Asadi[3], Dan R. Burke[2], John Wawrzynek[2]
[1]:M.I.T., [2]:UC Berkeley, [3]: Stanford

U.C. Berkeley Reconfigurable Group,
Stanford Nolan Lab, M.I.T. Hornet Group

Presented during the proceedings
of ISFPGA'2011, February 28th

## Overview

### Preliminaries

- FPGA designs can be customized on a per-application basis, and optimized for multiple execution models.
- GPGPUs specialize in data parallel computation, and perform best on data parallel workloads.

### Research Question

**How does an FPGA compare to a GPGPU when the algorithm in question follows the GPGPU's execution model, and is a good match to the GPGPU's architecture?**

### Context

Our application driver is a data parallel ***Bayesian inference*** algorithm that is actively being used to learn cell signaling pathways in a systems biology setting. **For this research, we have reorganized the Bayesian Inference loop nest to make the algorithm compute-bound on both FPGA and GPGPU.**

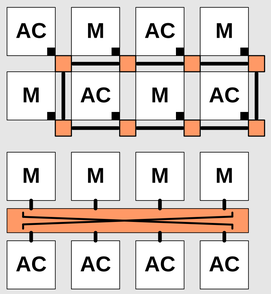
## Motivation

Related work[4] is investigating **microarchitectural template-based** approaches as a way to program FPGAs.

### Idea

Base FPGA designs on microarchitectural patterns (i.e. MIMD, SIMD, etc) at the top level, and customize processing cores (ACs) and interconnects on a per-application basis.

**In this work, we want to find an upper-bound for how well a template-based FPGA design performs, relative to more programmable approaches.**




Example template-based FPGA designs with ring (top) and crossbar (bottom) interconnects (M: memory)

[4]: I. Lebedev, S. Cheng, A. Doupnik, J. Martin, C. Fletcher, D. Burke, M. Lin, J. Wawrzynek. MARC: A Many-Core Approach to Reconfigurable Computing. Proc. of the 6th international conference on reconfigurable computing and FPGAs, 2010.

## Results

### Study

**Determine the throughput of an FPGA core, relative to a GPGPU core.**
Bayesian inference workloads: 32 node (**32n**) and 37 node (**37n**).

| Device name | Silicon {Process (nm), die area ($mm^2$)} | Processing core count | Processing core clock (Mhz) | Performance, as a multiple of the Nvidia GT 330m baseline: 32n | 37n | % total time |
|---|---|---|---|---|---|---|
| Virtex-5 LX155t | {65, 270} | 48 | 250 | 3.52 | 3.58 | 99 |
| Virtex-6 LX240t | {40, 255} | 120 | 300 | 10.0 | 10.9 | 98 |
| Nvidia GT 330m | {40, 100} | 48 | 1265 | 1.00 | 1.00 | 84 |
| Nvidia GTX 480 | {40, 529} | 480 | 1401 | 12.8 | 14.6 | 72 |

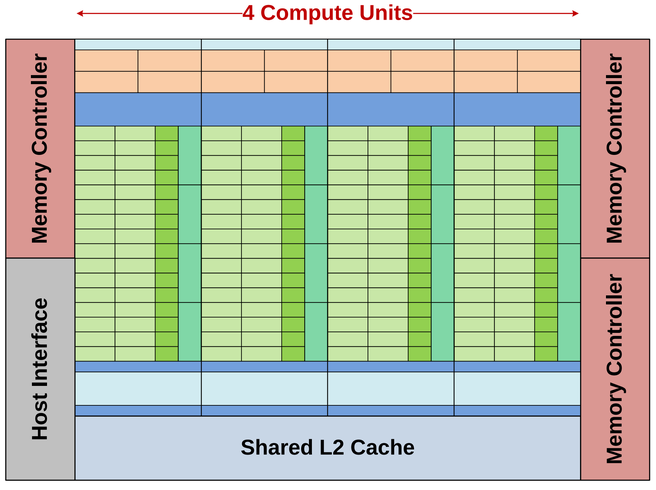
After normalizing for:
(a) core count,
(b) the clock frequency between FPGA devices

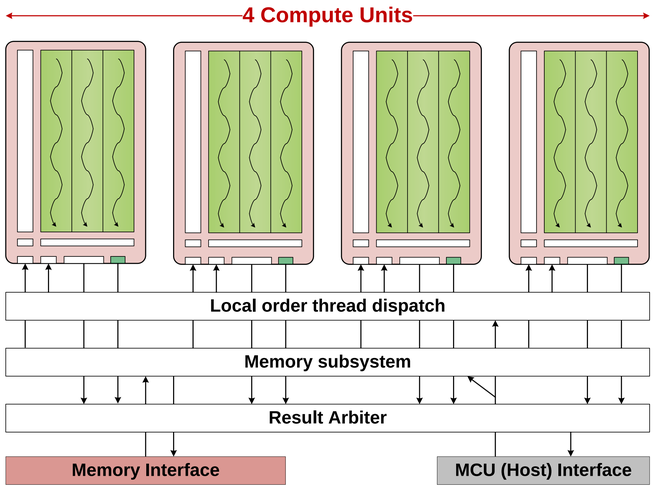
| Normalized Throughput[5] | Nvidia GT 330m | Nvidia GTX 480 |
|---|---|---|
| Virtex-5 LX155t | 4.26–4.30x | 3.00–3.30x |
| Virtex-6 LX240t | 4.00–4.36x | 3.00–3.11x |

[5]:Example: A Virtex-5 core achieves 4.26–4.30x throughput, relative to a GT 330m core.

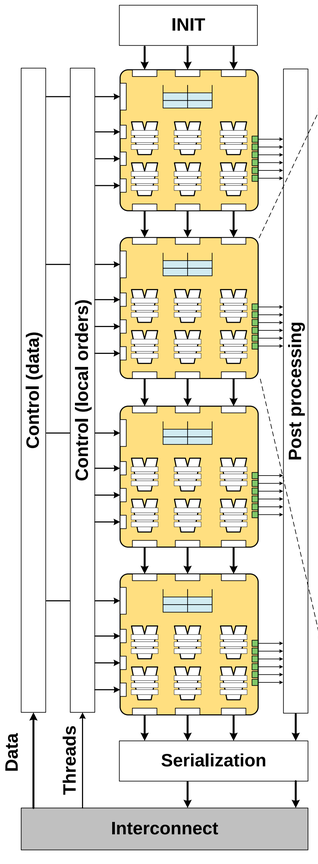
GPGPU

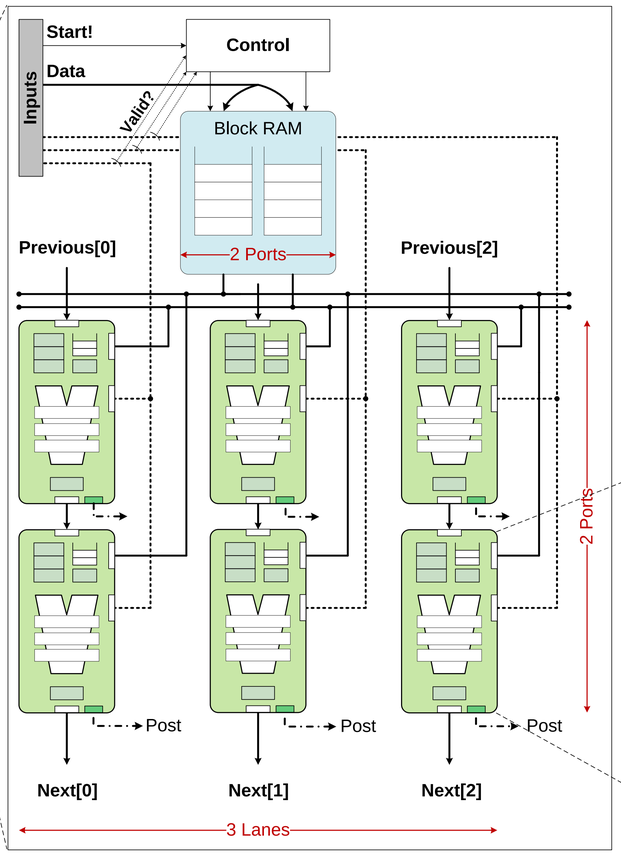
FPGA




## FPGA




## GPGPU[6]

[6]:figures are based on the Nvidia Fermi.