A 180 GFLOP/s, 15 GFLOP/W, 500 million transistor FPGA in 90nm CMOS

Ashok Vittal
Fremont, CA
ashok.vittal@gmail.com

Hare K. Verma
San Jose, CA

Abstract

We present an overview of the Vx200, a 90nm FPGA with 204 integrated floating point units, 408 24kb dual port block memories and 816 32x24 dual port register files. The 340 mm^2 die has 500 million transistors, offers internal memory bandwidth of 1.8 TB/s and external IO bandwidth of 40 GB/s. The device is targeted for compute intensive applications in the high end imaging, test & measurement and high performance computing markets. Benchmark results for filtering, fast Fourier transforms and pixel stream correlation designs are presented showing effective single precision and extended single precision performance of more than 100 GFLOP/s.

Categories and Subject Descriptors

B.7.1 [**Integrated circuits**]: VLSI – *gate arrays, algorithms implemented in hardware.*

General Terms

Algorithms, Performance, Design.

Keywords

FPGA, floating point units, power efficiency, compute intensive applications

1. Introduction

Several compute intensive applications require single chip performance of hundreds of billions of operations per second. Air flow simulation in a jet engine, increased rate stock option portfolio evaluation, real-time image reconstruction in medical imaging & video-rate synthetic aperture radar not only require hundreds of billions of operations per second, but also have dynamic range and precision that necessitate the use of floating point computations. Such applications have been underserved by multi-core processors, general purpose graphics processing units (GPUs) and field-programmable gate arrays (FPGAs). While peak performance of these devices can be attractive, real applications see effective performance a small fraction of peak. Power efficiency is also a key concern in several of these applications because a data center utilizing a cluster of compute nodes may easily have larger power & cooling cost than the amortized cost of the servers themselves.

The fetch-decode-execute-write back model of computation in general purpose processors limits the power efficiency attainable [1]. Large instruction and data caches are necessary to hide memory latency issues, leading to further inefficiencies [2], [3].

FPGAs have traditionally been used for glue logic bus-bridging applications and control can easily be implemented on traditional FPGA fabrics [4], [5]. However, when significant datapath elements are also part of the design and utilizations are high, effective performance suffers. The techniques described in this paper overcome these limitations.

In this paper we propose an architecture which includes floating point units in an FPGA fabric, such that the control portion can be implemented on the fabric and the datapath maps effectively to the signal processing engines with associated memory and dedicated bus-based interconnect. The effective single chip performance achieved is substantially better than other available solutions. Power efficiency is also significantly better than other devices as the algorithm is directly executed in hardware.

Section II provides an overview of our FPGA architecture. Section III describes the various blocks that comprise the device. Section IV outlines the algorithms used in our design software platform and Section V concludes with benchmark results.

2. Architecture overview

This section presents an overview of our device architecture, and explains the various design choices made.

Existing FPGA architectures see a large overhead in terms of critical path delay and die area due to the interconnect. While look up table delays can be less than 100ps, several hundred ps of delay is incurred in the routing multiplexers on the lookup table inputs/outputs & switchboxes. The inclusion of significantly faster local connections in our device [6] enables reduction of interconnect overhead along the critical paths. These local connections are not just nearest-neighbor connections and are designed such that when datapath elements like wide adders, multiplexers, gates, etc are stacked next to each other, fast local connections are guaranteed to be available.

The traditional core cell in existing FPGA architectures have been 4 or 6-input lookup tables with carry-chain support for ripple carry adders and multiplexer chaining. The large fraction (>70%) of area used by routing resources means that more sophisticated core cells would be more efficient from a functionality per die area perspective. Our core cells can be configured to perform 2 independent 4-input functions, 5 4-input

Electronic Design Processes'07, April 12–13, 2007, Monterey, CA, USA.

functions with 4 having common inputs, two 5-input functions with 4 common inputs, one 6-input lookup table, one 8-input symmetric function, one 8-to-1 multiplexer, one 4-bit adder/subtractor/accumulator with registered outputs, 1 8-to-1 priority encoder, 1 4-bit shift register, 4 configurable sequential elements with load, clear & enable with active high/low clock, asynchronous reset and latch/flip-flop configurability. Functions like wide adders require half the number of logic elements in our architecture compared to Xilinx/Altera architectures [7].

Signal processing engines which integrate computational elements and storage elements are seamlessly integrated into the FPGA fabric. The computational unit can perform an IEEE 754-compliant extended single precision or single precision multiply-add, a 36-bit multiply, a 32-bit multiply-accumulate, 2 18-bit multiplies, 2 16-bit multiply-accumulates, 4 9-bit multiplies or 4 8-bit multiply-accumulates with rounding and saturation. Associate with each compute unit are two 24 kb block memories and 4 32x24 dual port register files configurable in terms of depth and width. A 2-dimensional bus-based interconnect between the signal processing engines ensures that key kernels of typical algorithms, like filtering or radix-2 butterflies of fast Fourier transforms can attain predictable 450 MHz performance.

3. Device overview

This section describes the implementations of the various blocks on the Vx200: the FPGA fabric, the signal processing engines, the configurable IOs and clock network. The section concludes with an overview of the verification flow used. The 18mm x 19mm die shot is shown in Figure 1 below.

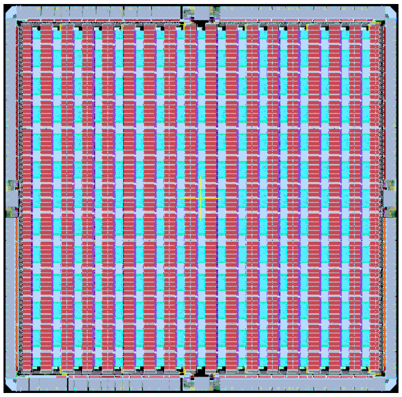

Figure 1. Vx200 die shot

The die is packaged in a 42.5mm x 42.5mm 1680-pin flip-chip ball grid array with 1mm pitch.

3.1 Fabric implementation

The logic and routing blocks that comprise the fabric were implemented using 9 layers of metal in a 90nm CMOS process. The configuration memory uses logic design rules. The block area is 24k square microns, achieving 10.2 32-bit GOPS/mm^2.

3.2 Signal processing engines

The layout of the logic and routing tile that is repeated in a 2-dimensional fashion over the die is shown in Figure 2. The fabric logic and routing blocks are on the left and right. The memories are at the top and bottom and the computational unit is at the center. The bus-based interconnect is to the left of the memories and compute units.

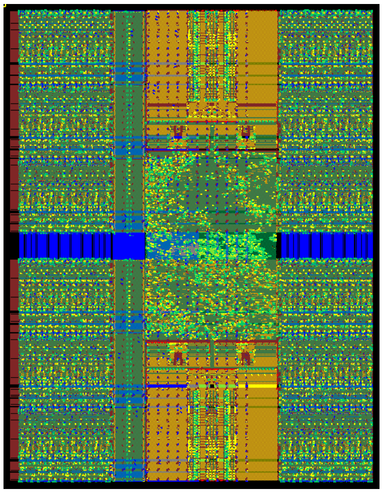

Figure 2. Repeated logic and routing tile

3.3 IO banks

The device offers 830 user IOs grouped into multiple banks – one bank meant for configuration/JTAG, 8 banks for clock inputs, 4 single ended banks and 4 single or differential banks. The IOs can be configured as several single ended standards including HSTL, SSTL, PCI, PCI-X, LVTTL and LVCMOS with programmable drive strengths. Four banks include IOs configurable as LVDS up to 1.2 Gbps. Four x64 DDR2 interfaces at 800Mbps can be implemented using the single ended banks – this is enabled by 112 DLLs and the dedicated IO clock tree. The IO clock tree provides configurable clocking to support x8, x16, x32 and x64 DDR2 interfaces, with the DDR flops and per-bit delay included in the pads.

3.4 Clock network

The clock network was designed to ensure skew of less than 250ps between any two flip-flops on this large die. The chip has 24 PLLs each of which are wide range with input frequencies ranging from 7MHz to 800 MHz and output frequencies ranging from 10 MHz to 1 GHz. The clock jitter is less than 2.5% of the clock cycle and correct VCO operation was observed up to 3.9 GHz. Sixteen global clock H-trees are distributed, feeding flexible local & regional clock networks.

3.5 Verification flow

Each of the device primitives in the architecture has a schematic used for layout purposes and a Verilog equivalent read in by the design software platform for configuration purposes.

Transistor-level formal verification was used to ensure that the software view of the device was consistent with the implemented primitives. We were able to run full chip Verilog simulation using VCS on the LVS netlist to verify that bit-streams were downloaded correctly, and the device functioned as expected on over a hundred full chip designs. Formal verification was also used extensively on hundreds of designs to verify that synthesis & mapping did not introduce any logic bugs.

4. Design software platform

This section describes the design software platform that customers use to program our device. The algorithms used are touched upon.

We developed an RTL to bit-stream compiler. The single executable with a unified data model accepts Verilog or VHDL and SDC timing constraints. It performs synthesis, mapping, placement, routing, timing analysis and generates the bit-stream that is used to program the device. The timer handles false paths, multi-cycle paths and min/max delays. The innovations on the placement front are described elsewhere [8]. The router is based on A* path search and uses a cost-based approach to resolve overlaps. The device primitives are described using Verilog and .lib – over 70 architectures were tried, changing the core cell, interconnect fabric and ensuring that the hundreds of designs, including dozens of potential customer designs would see push-button results that were better in terms of quality of results, and at least 5X better in terms of run time compared to existing FPGA implementation flows.

5. Benchmark results

We ran several customer designs and measured performance. The results are in table below. Our 90nm device offers at least 3 times the performance of comparable area 65nm FPGAs. The power dissipation for the 180 GFLOP/s result is 12 W at the typical process corner, 85 degrees C, 1.2V, yielding power efficiency of 15 GFLOP/W. The 256x256 2D FFT with floating point data including 2 row engines and 2 column engines does not fit on any 65nm FPGA as it requires more logic for the floating point units than available.

Table 1. Performance comparisons (GFLOP/s)

Design	Vx200	90nm FPGA	65nm FPGA
Polynomial evaluation	180	25	48
FIR filter	180	25	52
IIR filter	154	26	52
256x256 2DFFT	137	-	-

Silicon is back from TSMC and functionality has been validated on over a hundred tests with automated test equipment at the test house and in our lab. Fabric performance at 750 MHz has been observed on simple designs.

6. Acknowledgments

This paper represents the efforts of the Velogix engineering team: Ravi Sunkavalli, Conrad Kong, Manoj Gunwani, Elliott Delaye, Rahul Nimaiyar, Bo Hu, Ravi Kurlagunda, Yean-Yow Hwang, Izak Kense, Cliff Yang, Chandra Mulpuri, Sourav Maji, Nilanjan Chatterjee, Jia Liu, Kan Liu, Boaz Apotheker, Tzun-Wei Lee, Alex Shur, Sudip Nag, Ravi Gutala and Chandra Kashyap.

7. References

[1] Vangal, S. et al. *An 80-tile 1.28 TFLOPS Network-on-Chip in 65nm CMOS*. Proceedings of the International Symposium on Solid State Circuits, February 2007, pp. 98-99.

[2] T.-Y. Yeh. *Low-power high-performance architecture of the PWRficient processor family*. Hot Chips, August 2006.

[3] J.A. Kahle et al. *Introduction to the Cell multiprocessor*, IBM Journal of research and Development, 2005

[4] S. Douglass, K. Vissers and P. Alfke, *Virtex 5, the next generation 65nm FPGA*, Hot Chips, August 2006

[5] Lewis, D. et al. *The Stratix II logic and routing architecture*, Proceedings of the International Symposium on FPGAs, 2005.

[6] A. Vittal and H.K. Verma *Programmable integrated circuit architecture.* U.S. Patent 6980029, December 2005.

[7] H.K. Verma and A. Vittal *Programmable functional generator and method operating as combinational, sequenctial and routing cells*, U.S. Patent 6980025, December 2005.

[8] Hu, B. *Timing-driven placement for heterogeneous field programmable gate arrays*, Proceedings of the International Conference on Computer-Aided Design, November 2006