

CS6811: Research Seminar on Reconfigurable Hardware

Lockwood : Spring 2002

Review of: The SmartDIMM Platform as a Reconfigurable System-On-Chip Prototype

- Paper by:

- Paul Hulina, Lucas Roth, Lee Coraor

- (Penn State U.)

- FPGA 2002 Poster

- On-line as:

- <http://www.cedcc.psu.edu/smartdim/>

- Summarized by: James Moscola



Style of Writing

- Background and Application Survey for SmartDIMM
- What is a SmartDIMM?
 - A reconfigurable platform that attaches to the main memory bus of a desktop PC
- Why use a SmartDIMM?
 - SmartDIMMs offer performance and functionality prototyping of computing-in-memory-architectures (CIMA) that combine microprocessors, reconfigurable FPGA logic, and memory into a single-chip solution.

Introduction

- Contains both FPGA and DRAM components
- Connects an FPGA to the main memory bus of an Intel x86 processor
- Allows communication between CPU and FPGA at over 1 gigabyte per second

Technical Background

- Important Highlights of Project
 - the merging of logic and DRAM
 - the use of reconfigurable logic in conjunction with processors and memory modules
- Must provide method for mapping applications to a complex system such as SmartDIMM
 - partitioning tasks between CPU and reconfigurable logic (FPGA)
 - identify methods for transferring data between CPU and FPGA

Researching the Architecture

- Previous work included an evaluation of different CIMAs.
 - D.Landis, L.Roth, P.Hulina, L.Coraor, S.Deno: Evaluation of Computing in Memory Architectures for Digital Image Processing
 - Processor-In-Memory (PIM)
 - Vector DRAM Processor
 - Multiprocessor-on-a-chip
 - Merged CPU/FPGA/DRAM (CFD)
- CFD was chosen as offering the most promise for future performance gains through analytical performance models

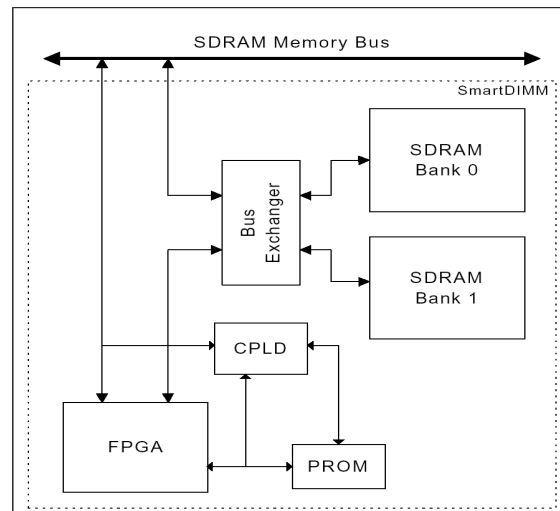
Evaluation Results

Arch. Type	Histo-gram	Filter	Thres hold	Die Area (0.25 u)	Die Area (0.18 u)
RISC CPU	19 ms	67 ms	17 ms	300 mm ²	180 mm ²
PIM	71 ms	0.6 ms	0.1 ms	350 mm ²	140 mm ²
Vector DRAM-333MHz	28 ms	50 ms	0.9 ms	430 mm ²	190 mm ²
Multi-CPU 4 @ 333	8 ms	28 ms	6.5 ms	450 mm ²	200 mm ²
CFD 8000 LE, fast-carry	23 ms	1.5 ms	1.0 ms	430 mm ²	190 mm ²
CFD 16000 LE, fast-carry	12 ms	1.2 ms	1.0 ms	530 mm ²	250 mm ²

Architecture: Hardware Design

- Components of the SmartDIMM device
 - User Application FPGA
 - Xilinx Virtex XCV300 (programmable in under 5ms)
 - configurable statically via onboard PROM or dynamically via memory bus
 - SmartDIMM design allows expansion to XCV1000E
 - Dual 32MB banks of SDRAM
 - accessible by both the CPU and the FPGA
 - CPU must always have seamless access to memory
 - CPU can access one bank of memory while FPGA accesses the other bank of memory.
 - » increases memory utilization
 - Complex Programmable Logic Device (CPLD)
 - Altera MAX7128AE
 - directly accessible by CPU over memory bus
 - controls bank-switching
 - handles configuration of user application FPGA
 - contains performance counters and debugging registers accessible by the CPU

Architecture: Hardware Design (... continued)



Architecture: Application Design Framework

- Common functions are built into a library available to user applications
 - Verilog Source Code Library
 - SDRAM interface
 - utilizes full bandwidth of PC100 memory bus
 - can only access non-CPU-accessible bank
 - data must be transferred from conventional SDRAM to SDRAM on SmartDIMM before FPGA can access it
 - CPU interface & CPU/FPGA synchronization
 - set of five registers accessible by CPU and FPGA
 - » four registers provide information on availability and location of data in the SDRAM
 - » fifth register acts as a mutex on four data registers
 - » All communication with CPU takes place through CPU polling these registers

Architecture: Software Interface

- Currently supported by Linux
 - kernels 2.2.x and 2.4.x
- Two components to software driver
 - kernel-level driver
 - allocates the resources on the SmartDIMM device
 - maps them into the PC memory space
 - maps to 64MB address block
 - operations are performed by accessing this memory block
 - user-level library
 - provides basic interface for programming and interacting with SmartDIMM device

Image Processing

- requires manipulation of single-bit or small-bit-width data
- contains a great deal of parallelism
- well suited for FPGA design

Image Processing: Performance Evaluation

- implemented variety of image processing kernels in VHDL
- compared to conventional software
- three hardware configurations
 - conventional CPU
 - Intel Celeron 500MHz
 - Xilinx Virtex XCV300
 - utilized full 240-pin bandwidth
 - projected SmartDIMM performance
 - achieved 70 to 80% of available performance from FPGA

Image Processing: Performance Evaluation

Operation (time per pixel in image)	500MHz CPU	XCV300 Theoretical Maximum	SmartDIMM (estimated)
Threshold	19.65ns	1.84ns	2.62ns
3x3 Filter	37.24ns	1.89ns	2.62ns
Edge Enhance	26.54ns	2.42ns	3.02ns

Java Acceleration

- provide extension to idea of dynamically translating Java byte codes into native code at run time
- significant leap in Java execution speed is possible
 - map performance-critical sections of Java application directly to FPGA hardware
 - i.e. DES encryption algorithm has estimated 100x speedup when compared to native code execution

Java Acceleration: Key Issues

- Partition Code between CPU and FPGA
 - profiling technique to identify critical sections of an application
 - algorithms to identify parallelism that can be exploited
- Translating byte-code to FPGA bit file
 - dynamic synthesis of hardware from byte-code
 - use pre-compiled cores for common methods
 - avoids hardware compilation cost
 - application specific cores are faster than synthesized cores
- Mechanism for passing parameters to and from sections of the application code mapped onto the FPGA

Network Interface

- Focus on interface from computer system to network
 - decrease latencies
 - improve bandwidth
- Problems with current NIC design
 - slow PCI bus
 - move to memory bus to avoid overhead of copying data and slow bus speeds
 - memory distinction causes performance drain
 - shared memory for exchanging information between host PC and NIC
 - typically have a single processor
 - FPGA offers methods for high parallelism at low cost

Encryption

- DES, RSA, IDEA, etc. provide secure communication across the internet
 - computationally intensive
 - implementation in Virtex FPGA can achieve throughput of 1.2 gigabytes per second
 - typical PC can only achieve 10MB per second

Conclusion

- Created a useful, inexpensive tool for performance evaluation and rapid prototyping
- Better than FPGA-based PCI expansion cards
 - improves bandwidth with higher bus speeds
 - improves latency between FPGA and CPU and memory