

NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

The Future of Parallel Computing

Building blocks to solve large matrices

James Amadei
Katy Lee
Mentor: Dr. Olaf Storaasli

University of Pittsburgh CASE

NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

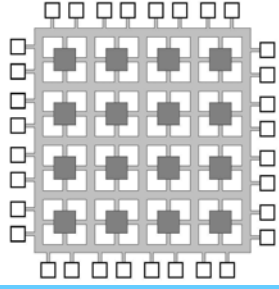
Overview

- Introduction
- VIVA Advantages
- SRAM
- Serial vs. Parallel
- User-Friendly Output
- Traveling Salesman Problem
- Summary
- Coming Attractions

NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

What is an FPGA?

- FPGAs consist of logic blocks with reconfigurable interconnects
- The functionality of the chip is determined by the configuration of interconnects



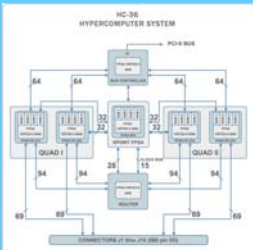
NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

FPGAs and VIVA

- FPGAs perform hundreds of parallel operations/cycle
- VIVA: a graphical programming language
- 7 FPGAs + VIVA = HC-36 (Hypercomputer designed by Starbridge)

NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

Langley's HC-36

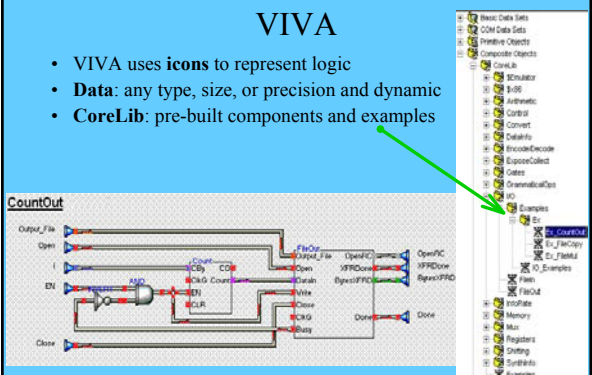


- 4 FPGAs available to users
- 3 FPGAs for internal communication

NASA XILINX VIVA VIVA 2.4 Revision: 010 Date: 01 Apr 04

VIVA

- VIVA uses **icons** to represent logic
- **Data:** any type, size, or precision and dynamic
- **CoreLib:** pre-built components and examples



NASA XILINX VIVA

VIVA System Descriptions (Porting)

- list the resources available in a system
- implement VIVA designs on any system
- develop applications before hardware built
- develop and simulate programs on X86

NASA XILINX VIVA

VIVA Advantages over VHDL

- easy to learn and use
- no syntax to remember
- drag-and-drop interface versus textual interface
- data type polymorphism and recursion
- multiple dimensions

VHDL (1-D)

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY fulladd IS
    PORT (Cin,x,y : IN STD_LOGIC;
          s, Cout : OUT STD_LOGIC);
END fulladd;

ARCHITECTURE LogicFunc OF fulladd IS
BEGIN
    s <= x XOR y XOR Cin;
    Cout <= (x AND y) OR (Cin AND y);
END LogicFunc;
                
```

Adder

VIVA (3-D)

NASA XILINX VIVA

Storing Data on the FPGA

- large matrices require quick data access (host ⇔ FPGA is typically slow)
- 3 Step Transfer in VIVA
 - transfer data from host to FPGA
 - calculation on FPGA
 - write results to host

NASA XILINX VIVA

FPGA RAM

- FPGAs have 288 KB block RAM each
- RAM configurable for any data type (fixed, float ...)
- flexible RAM via CoreLib (4x12, 1x16, 4x16 ...)
- data type and RAM selected by user in VIVA

NASA XILINX VIVA

DDR SRAM

- Double Data Rate Static Random Access Memory
- DDR doubles bandwidth - transfers data 2X/cycle
- SRAM better than DRAM (refresh unnecessary)

NASA XILINX VIVA

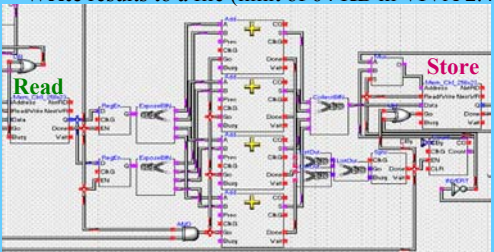
DDR SRAM

- large amounts of data stored near FPGAs
- 2* memory channels for parallel I/O
- 512 MB* accessible, as of VIVA 2.4

*VIVA 2.5 will expand

SRAM Test Successful

- **Read** 256 MB file into channel 0
- **Add** and **Store** results in channel 1
- **Write** results to a file (limit of 64 KB in VIVA 2.4)



The diagram shows a complex circuit with multiple channels. Channel 0 is labeled 'Read' and Channel 1 is labeled 'Store'. The circuit includes various logic blocks, registers, and data paths, with a central processing unit and memory blocks.

Coding Strategies

Serial

- noun (series): a number of things or events of the same class coming **one after another** in spatial or temporal succession

Parallel

- noun: an arrangement or state that permits **several operations** or tasks to be performed **simultaneously** rather than consecutively

definition by Webster

Greatest Common Divisor (GCD)

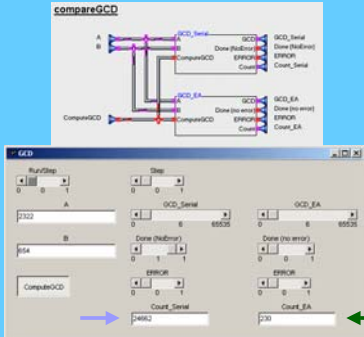
- For Loop
 - Direct
 - Sequential
- Euclid's Algorithm (EA)
 1. If $b|a$ then $\text{gcd}(a, b) = b$.
 2. If $a = bt + r$, for integers t and r , then $\text{gcd}(a, b) = \text{gcd}(b, r)$.

Example using EA
Let $a = 2322$, $b = 654$.

$2322 = 654 \cdot 3 + 360$	$\text{gcd}(2322, 654) = \text{gcd}(654, 360)$
$654 = 360 \cdot 1 + 294$	$\text{gcd}(654, 360) = \text{gcd}(360, 294)$
$360 = 294 \cdot 1 + 66$	$\text{gcd}(360, 294) = \text{gcd}(294, 66)$
$294 = 66 \cdot 4 + 30$	$\text{gcd}(294, 66) = \text{gcd}(66, 30)$
$66 = 30 \cdot 2 + 6$	$\text{gcd}(66, 30) = \text{gcd}(30, 6)$
$30 = 6 \cdot 5$	$\text{gcd}(30, 6) = 6$

Therefore, $\text{gcd}(2322, 654) = 6$.

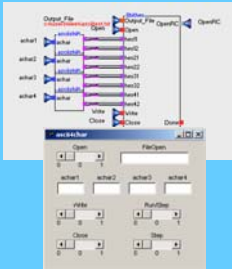
GCD



The schematic shows a 'compareGCD' block with inputs A and B, and outputs for GCD_Serial, GCD_EA, and Count. The simulation window shows the results for A=2322 and B=654. The 'Count_Serial' is 2322 and 'Count_EA' is 6. A green arrow points to the 'Count_EA' value with the text '100X faster'.

New User-Friendly Output Capability

- additional capability: convert hex to ASCII
- supports ASCII output
 - letters
 - numbers
 - special characters
 - string of numbers



The schematic shows an 'ASCII' block with inputs for 'Output_File' and 'ASCII'. The simulation window shows the output file 'ASCII.txt' with the contents 'a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9'.

Traveling Salesman Problem (TSP)

- Given a specified number of "cities" along with the cost of travel between each pair of them, find the cheapest way of visiting all the cities and returning to the first city visited
- Chose for test purposes an asymmetric TSP with 65 cities (TSP 65)*

*University of Heidelberg, <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>

NASA XILINX VIVA Vixia 2.4 Revision: 010 Date: 01 Apr 04

Parallelism

- Genetic Algorithm
 - for optimization
 - combinatorial problem
 - inherently parallel
 - operate on entire populations of candidate solutions
 - stochastic
- VIVA and FPGA
 - easy to use
 - inherently parallel
 - hardware layout
 - combinational logic
 - efficient use of resources

NASA XILINX VIVA Vixia 2.4 Revision: 010 Date: 01 Apr 04

Genetic Algorithm Implementation

Top Level View

Run Time Environment

Input parameters

Outputs

NASA XILINX VIVA Vixia 2.4 Revision: 010 Date: 01 Apr 04

Applying Temporal Parallelism

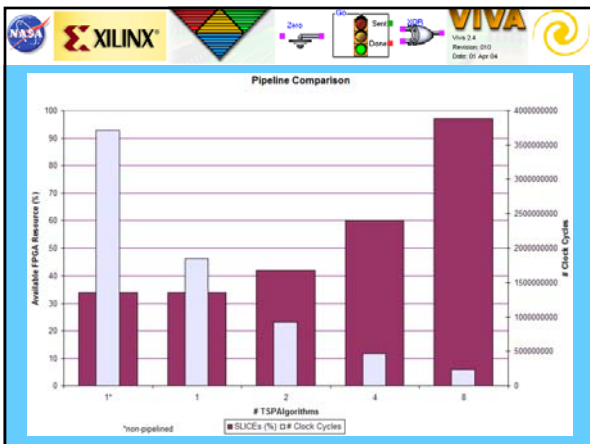
TSPAlgorithm non-pipelined

TSPAlgorithm pipelined

NASA XILINX VIVA Vixia 2.4 Revision: 010 Date: 01 Apr 04

Applying Spatial Parallelism

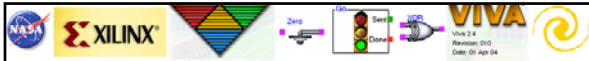
TSPMColFun



NASA XILINX VIVA Vixia 2.4 Revision: 010 Date: 01 Apr 04

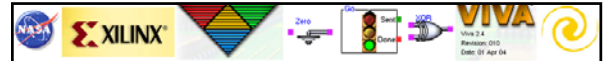
Accomplishments

- robust design (2 different systems)
 - one with DMA engine
 - one without
- different versions of VIVA
- portability across FPGAs
- MSFC TSP Algorithm runs on LaRC HC-36




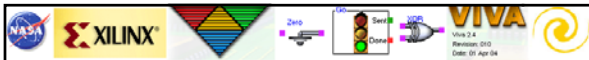
Combined Summary

- transfer, store, & access data to/from SRAM
- developed user-friendly output
- CPUs are general purpose
 - Inefficient use of on-chip real estate
- FPGAs can be application specific
 - Allows on-the-fly reconfiguration



Coming Attractions

- VIVA 2.5 expected to
 - double number of memory channels
 - lift 64 KB file write limitation
 - support negative numbers in FileOut
- write string of characters to output file
- cooperative development of parallel solver (Langley and Starbridge)



THANK YOU

