Delivering Acceleration: The Potential for Increased HPC Application Performance Using Reconfigurable Logic

David Caliga
SRC Computers, Inc
4240 N. Nevada Ave
Colorado Springs, CO 80907
719-262-0213

david.caliga@srccomp.com

David Peter Barker
SUPERsmith
PO Box 2226
Salinas, CA 93902-2226
831-442-8343

dbarker@supersmith.com

ABSTRACT

SRC Computers, Inc. has integrated adaptive computing into its SRC-6 high-end server, incorporating reconfigurable processors as peers to the microprocessors. Performance improvements resulting from reconfigurable computing can provide orders of magnitude speedups for a wide variety of algorithms. Reconfigurable logic in Field Programmable Gate Arrays (FPGAs) has shown great advantage to date in special purpose applications and specialty hardware. SRC Computers is working to bring this technology into the general purpose HPC world via an advanced system interconnect and enhanced compiler technology.

Categories and Subject Descriptors

Compiler technology

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Reconfigurable computing, FPGA

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SC2001 November 2001, Denver (c) 2001 ACM 1-58113-293-X/01/0011 $5.00

Proceedings of the ACM/IEEE SC2001 Conference (SC'01)
1-58113-293-X/01 $20.00 © 2001 ACM

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

1. INTRODUCTION

The SRC-6 system is a unique architecture capable of supporting a combination of up to 512 Intel® microprocessors and 256 Multi-Adaptive Processors (MAP™) on a common shared memory. A patented SRC crossbar switch design supports the connection of these processors with up to 256 memory ports with each port containing 16 banks of SDRAM (see Figure 1). The architectural design of the memory sub-system provides flat access latency from any microprocessor or reconfigurable processor in the SRC-6 system. This combination will offer in excess of 3 TFlops of theoretical peak [256 Pentium 4 processors and 128 MAPs on 32b floating-point data]. SRC Computers is pushing forward to harness this into sustained performance.

This paper explains how SRC Computers, Inc. has made advances in the reconfigurable computing field by incorporating FPGA technology at many levels in the SRC-6 system architecture.

Also described is SRC's patented FPGA-based MAP that user applications can utilize to deliver algorithm-specific computational acceleration.

2. RECONFIGURABLE COMPUTING DEFINED

In simplest terms, reconfigurable computing, based on FPGA technology, could be defined as the capability of reprogramming hardware to execute logic that is designed and optimized for a specific user's algorithms. Associated compiling technology can provide a transparent method of integrating the computational capability of FPGA technology and microprocessors into a single application executable code. The use of such integrated compiling technologies enables reconfigurable architectures to extend beyond the FPGAs and the "glueware" that attaches them to the host computer.

Automatic compilation of applications onto reconfigurable architectures generates the logic for both the specific hardware configuration and also the execution management of the FPGA resources. The compilation environment must also be extensible to a wide range of user applications on a single system.

SRC Computers has taken into account all of these factors in developing its unique reconfigurable computing capability for the SRC-6 system.

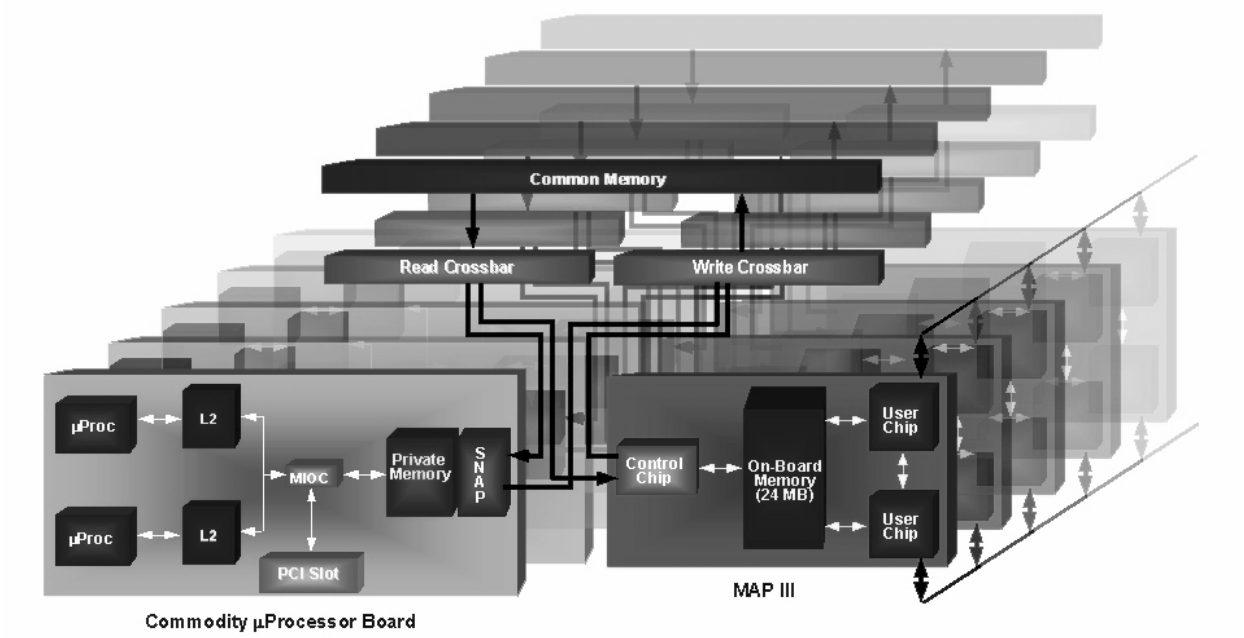


Figure 1. Overview of SRC-6 System Architecture

Proceedings of the ACM/IEEE SC2001 Conference (SC'01)
1-58113-293-X/01 $20.00 © 2001 **ACM**

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

3. MULTI-ADAPTIVE PROCESSOR (MAP)

The heart of the SRC-6 system is its Multi-Adaptive Processing feature provided by MAP units (see Figure 2). These units utilize hardware-implemented functions, which can greatly accelerate application algorithms over compiler implemented instruction sets for microprocessors.

Major architectural characteristics of MAP include:

- Each MAP executes independently of the general-purpose processors, including loading and storing needed data, after being provided with a list of commands to execute.
- The control of the MAP is done by the application via a Command List (COMLIST). The COMLIST contains a list of controlling instructions for the Control Chip. Examples of functions performed by these instructions are Direct Memory Access (DMA) reads and writes and the execution synchronization with the User Array.
- MAP units have access to Common Memory (CM).
- Common Memory addresses specified by commands or generated by user applications are virtual addresses. Addresses are translated, virtual to physical, by Translation Look-aside Buffer (TLB) entries with the DMA logic of each MAP.
- The User Array portion of a MAP unit is configured for algorithmic requirements. This logic can read and write on-board memory through multiple ports and can interact with the control logic and DMA Engine.
- By means of chain ports, MAP units can communicate between themselves without using any memory bandwidth. Thus a particular MAP can send partial results to another unit, or similarly, can receive such partial results from another MAP.
- Full system interrupt and semaphore capability is available between MAP units or the microprocessors.

Multi-processor applications can easily utilize the MAP by identifying the relevant portions of the parallel computational algorithms. The application can utilize N microprocessors and M MAPs.

4. RECONFIGURATION LOGIC PERFORMANCE POTENTIAL

The attraction of using FPGAs in SRC's MAP is the ability to generate algorithm specific logic, which has the potential of orders of magnitude speedups for computationally intensive algorithms. There have been many demonstrations showing this magnitude of speedup in algorithms for genetic sequencing, encryption/de-encryption, string searches and integer forms of image and signal processing.

The performance improvement of these algorithms can come from any or all of the following:

- Memory bandwidth improvements
  - Six ports to memory
- Data flow parallelism
  - Bit-sized data allows for multiple parallel processing streams per 64 bits of data read from memory or algorithms that may need multiple 32 or 64-bit input values
- Computational block level re-scheduling
  - Re-schedule independent computations to be concurrent in time
- Instruction Set Architecture (ISA) effectiveness
  - Create operations that are "right-sized" in bits relative to the type of data, i.e. 6-bit or 256-bit integer operations

The following examples show potential performance improvement contributors of an FPGA over that of a microprocessor.

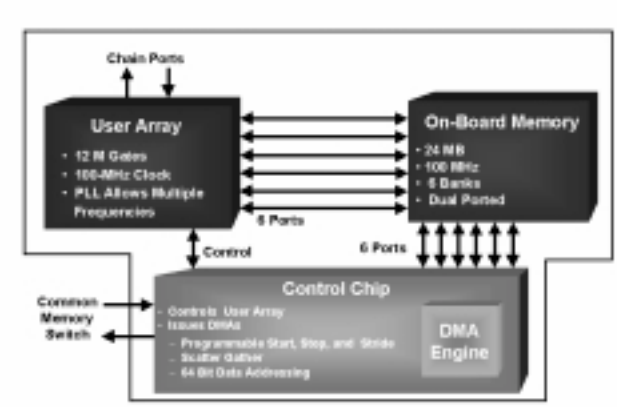


Figure 2. MAP Block Diagram

1-58113-293-X/01 $20.00 © 2001 ACM

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

Example 1:

This example shows a performance comparison between an FPGA and a 900-MHz microprocessor.

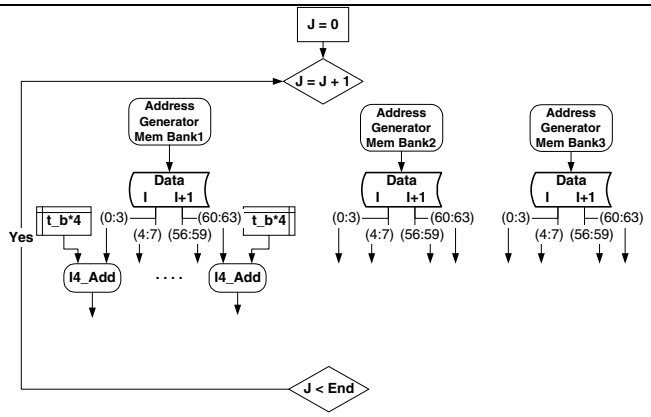
Perform an integer add operation on each 4 bits of a data stream. The MAP reads in 64 bits of data from each memory bank. The algorithm will use 3 memory banks to read data from the input stream and the logic will create 16 parallel computation streams.

Feature	Speedup over Microprocessor	Derivation
Clock rate	100/900 or 1/9	100 for the FPGA and 900 for the microprocessor
Memory bandwidth	3	3 memory banks to read data from input stream
Data flow parallelism	16	16 parallel computation streams
Block level re-scheduling	1	None
ISA effectiveness	10	Number of instructions on the microprocessor required to perform the equivalent integer operation on the 4-bit data value
Total	**53.3**	**1/9 *3 * 16* 1 * 10 = 53.3**

Algorithm Pseudo Code

```
Loop over 4-bit values in 32-bit data value
(isrc) and add a 4-bit value to input
value. Store resulting 4-bit value (ires)

     len = 4     //4-bit value
     ipos = 29   //start at position 29

    Do 100 j = 1, 16
      ibgn = (j-1)* 4 + 1

      mvbits (isrc,ibgn,len,itemp,ipos)

      ires  = idest + t_b4(j)
           //4-bit value stored in ires

      mvbits (ires,ipos,len,idest,ibgn)
100 Continue
```

Segment of Logic Flow



DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

Example 2:

The following example shows a comparison between an FPGA and the Alpha EV6.

Perform a 32-bit floating-point convolution filter on a set of vector data. The convolution filter will be 64 points. The filter will be stored in a set of registers in the FPGA chip. Two values of the input data will be read every clock. The output computation rate will generate two output values every clock.

Operation	FPGA	DEC Alpha EV6
Read input data values	2 values every clock	1 value every clock
Operations every clock	128 Multiply-Adds	1 Mult and 1 Add

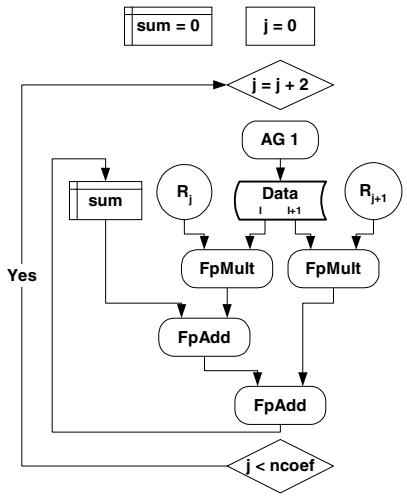
Feature	Speedup over Alpha EV6	Derivation
Clock rate	100/800 or 1/8	100 for FPGA and 800 for Alpha EV6
Memory bandwidth	2	2 values read every clock
Data flow parallelism	1	N/A
Block level re-scheduling	64	Convolution filter is 64 points
ISA effectiveness	3/2	Number of instructions on the FPGA relative to the microprocessor per clock
Total Speedup	**24**	**1/8 * 2 * 1 * 64 * 3/2 = 24**

A segment of the filter code is shown here:

Filter Pseudo Code

```
C Loop over output points

   Do 100 n = 1, nout
     sum = 0.

C Loop over filter coefficients
     Do 200 j = 1, ncoef
       sum = data_in(n+j-1) * filter(j)
100   continue

     data_out(n) = sum

100 continue
```

Logic Flow



DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.