RAMA: A Reconfigurable Datapath System-on-Chip Architecture

Glen Haas glen@itc-usa.com *George Landers* george@itc-usa.com
Levon Petrosian levonpg@itc-usa.com *Neal Stollon* neal@itc-usa.com
Les Veal les_veal@itc-usa.com
Infinite Technology Corporation
Richardson, Texas

Abstract: RAMA (Reconfigurable Array MultiMedia Architecture) is a proof of concept implementation of a reconfigurable multi-datapath architecture developed to focus on DSP applications in video and multimedia. RAMA's capabilities, based on RAD (Reconfigurable Arithmetic Datapath) architecture and related technical innovations are presented. These include the use of reconfigurable busses to accelerate data movement between processing blocks and external logic buffering for concurrent datapath and logical co-processing. Hardware issues of multi-datapath DSP architectures based on RAD and approaches taken during RAMA development are presented.

Introduction:

As increasingly high complexity and multi-staged DSP functions become needed in a range of applications, processing architectures that support MIMD (Multiple-Instruction Multiple Data) operations become more competitive and viable approaches to providing DSP capabilities. This paper presents RAMA (Reconfigurable Array MultiMedia Architecture), a MIMD DSP Array based System-on-Chip (SoC) architecture. RAMA is based on a proprietary high performance reconfigurable datapath technology (capable of internal data transfer up to 64 Gbits/sec) developed by Infinite Technology Corporation to provide distributed datapath performance (up to 2 billion MAC and ALU operations/second) over a range of data-stream algorithms.

The DSP core element of this architecture is a MIMD RADarray™ processing engine, based on Infinite Technology Reconfigurable Arithmetic Datapath (RAD) core signal processor architecture [1]. RAMA proof of concept will integrate multiple instances of a baseline RADcore™ architecture with a RISC processor and memory resources to demonstrate extremely high-speed arithmetic performance in DSP applications.

RADcore architectures are an innovative approach to implementing low latency DSP functions. By replacing register based operations with a reconfigurable bus structure that interconnects execution unit blocks (both datapath elements such as ALUs and MACs and memory resources), RADcore facilitates streamlined dataflow operations over a range of DSP applications. For signal processing datapath algorithms such as imaging, optimizing the pipelined operations allows fewer data dependencies and higher throughput by allowing the datapath to be reconfigured for efficient implementation of different imaging algorithms.

Technically unique features integrated into RAMA RADcore and RADarray architectures support accelerated DSP performance over different classes of algorithms, and in particular to streaming video algorithms. At 250+ MHz core operation, each RADcore coprocessor instance is concurrently capable of 500 million multiply accumulate operations and 500 million arithmetic or logical operations per second (based on RADcores with dual MAC/dual ALU resources) to support video applications. RAMA's MIMD RADarray hierarchical architecture is configured with 4 RADcore co-processors, each of which can operate independently, or in conjunction with other RADcore coprocessors and RISC processor. RAMA integrates a distributed Memory Array and Bus arbitration scheme to optimize the RADarray performance and data throughput.

In presenting the RAMA architecture, key issues of interfacing of a MIMD datapath array based architecture to other SoC sub-systems are addressed. External logic buffering, a RAMA innovation, is presented as a method of tightly integrating DSP operations with external general purpose (FPGA/ASIC) logic operations in a co-processing mode of operation. This provides performance advantages over traditional DSP architectures on a range of system level applications that require both high performance datapath and logic processing.

Reconfigurable Datapath Architectures

A key differentiation of RAMA from other MIMD system level architectures is the extensive use of reconfigurable busses for intra-chip data transfer at several hierarchical levels of the design. These levels include RADcore to memory transfers (over the memory bus), RADcore to RADcore data transfer (over the RADbus™), and execution unit to execution unit communication within a RADcore (over the Reconfigurable Channel Bus).

The performance advantage of reconfigurable busses is seen in Figure 1A and 1B. Figure 1A illustrates a multi-staged algorithm made up of a series of DSP functions. Each function in the algorithm is implemented using a single RADcore in the array. Since each functional operation is addressed by a separate dedicated core, a simplified dataflow architecture with high performance pipelining is achievable. The power of reconfiguration at this inter-core level is that both the dataflow between cores and functionality of the cores themselves is dynamically configurable under software control.

Core reconfiguration to support diverse functions is seen in Figure 1B. Each core contains a suite of multi-function programmable Execution Units (EXUs) that are interconnected over a Reconfigurable Channel Bus (RCB). Setting the interconnection defines EXU to EXU dataflow operation. Changing the interconnection of the EXUs creates new dataflow to implement substantially different functions. The EXUs in RAMA RADcores are datapath centric (ALUs, MACs, multi-port memories), however Reconfigurable Cores for different applications may have significantly different sets of execution units.

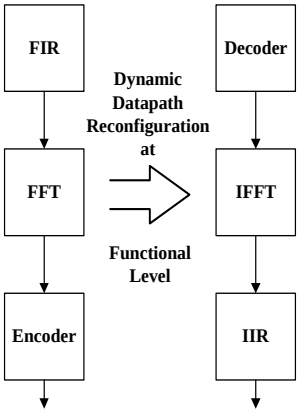

Figure 1A: Inter-Core Level Reconfiguration

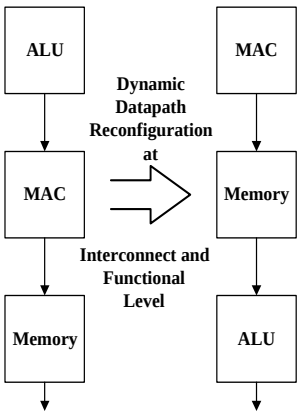

Figure 1B: EXU Unit Level Reconfiguration

Extensive use of reconfigurable datapath bussing in RAMA provides a significant amount of flexibility in both the chip and core level. Each core's instruction sequences operate independently with MIMD control and coordination provided by a RISC host processor.

RAMA Architecture

RAMA's chip level architecture decomposes into three major subsystems, which are hierarchically integrated to form a loosely coupled MIMD structure. The major subsystems in RAMA (as seen in Figure 2) are:

1. The host RISC processor and its supporting memory and control interfaces
2. RADarray, a RADcore datapath processor array and full speed data bus interface (RADbus). RADarrays, configured from multiple RADcore engines, provide dedicated co-processing to a RISC host processor; with each RADcore operating independently or in concert with other RADcores over a dedicated Databus (RADbus) and memory bus (MemBus).
3. Memory subsystem consisting of on-chip memory, external memory interfaces, memory arbiter, and the memory bus fabric. On-chip memory bus arbitration is integrated for control of the inter-core memory bus (MemBus), which provides concurrent Read and Write busses with address and data channels running through dedicated read and write bus fabrics.

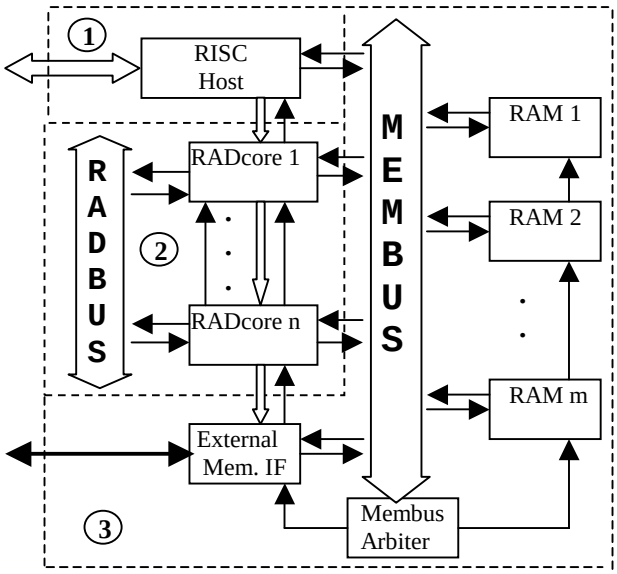

Figure 2. RAMA Block Diagram showing:

1. **RISC processor subsystem**
2. **RADarray subsystem**
3. **Memory subsystem**

RADarray Subsystem

The key innovation in RAMA MIMD operations is the RADcore Datapath processor. Key innovations of RADcore are its Reconfigurable Bus Encapsulation architecture for datapath interconnect efficiency, and the integration of a controller/sequencer and Distributed Instruction Word (DIW) Memory in each core to support autonomous datapath processing [2].

Each RADcore contains a core controller/sequencer block, a DIW Instruction Memory and a set of execution Units (EXUs), including an IO EXU, and a number of ALU, MAC and Memory EXUs. All are encapsulated and interconnected is through a Reconfigurable Channel Bus and supporting Flag and Initialization Busses (Figure 3.). Details of some of the critical areas of the RADcore design follow. Datapath resources in each RAMA RADcore include 2 ALUs, 2 MACs, 3 local 4-port memories and an external logic interface. RADcores for other SoC architectures may be configured with other types and arrangements of EXUs as best fit a given application area.

Controller/Sequencer

RADcore operations are controlled by distributed instruction sequences to each execution unit. Instruction flow is controlled by the core controller/ sequencer. The core controller/sequencer also provides an interface between the on chip host (RISC) processor and the rest of the RADcore. Primary controller functions include:

- Synchronization of data transfers via the I/O EXU for RADcore processing operations. During initialization, the controller initializes and coordinates loading of

Instruction data (via the IO) into the DIW memory.

- Receiving host instructions, interrupt handling based on EXU status, and coordinating host data transfers.
- DIW memory program control, by incrementing through a range of instruction words. Controller Instructions allow a program counter to loop on a DIW set or to branch to non-sequential DIWs. The controller supports up to four loop counters to control 4 nested loops in a DIW program.

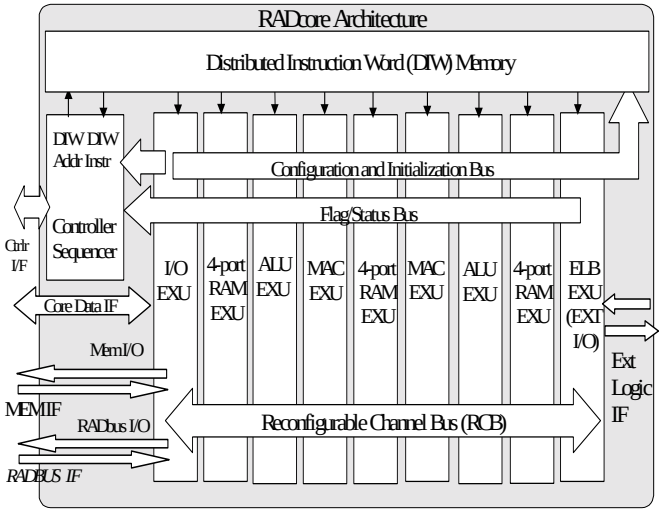

Figure 3: RADcore Architecture

RADcore Bus Architecture

There are four primary RADcore internal busses. The Reconfigurable Channel Bus (RCB), Flag/Status Bus, Initialization Bus, and DIW Instruction Bus provide localized (intra-core) data transfer and control to EXUs.

- Reconfigurable Channel Bus: The RAMA RCB is configured with 15 data busses per RADcore. Each EXU input can access (read) data from any one of the 15 data bus channels of the RCB under DIW bus selection control. An EXU output is allocated to one dedicated bus of the 15 that are available.
- Flag Bus: Each EXU has programmable/dedicated status flags that indicate conditions (memory stall, overflow, etc.) or completion of EXU operations. Flags are evaluated by the controller for conditional update of instruction sequencing (branching) or generation of interrupts to the host processor.
- Initialization Bus: The Initialization Bus contains data and address busses and control signals that load the DIW and allow initialization data to be loaded in each EXU (and RADcore local memories).
- DIW Instruction Bus: An Instruction Bus made of a series of concatenated instruction word wide busses that distribute instructions from the DIW to all RADcore (controller and EXU) blocks (as shown in Figure 4).

Instruction Word Control

The DIW allows a RADcore to maintain a coherent pipelined structure during processing operations via instruction word control. DIW based execution unit control offers precision in control of resources and independent EXU operations while maintaining flexible software control at a systems level.

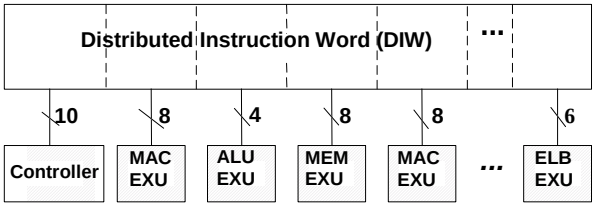

Figure 4: DIW Control Implementation

A notable advantage of DIW over other methods of distributed control and interfacing is the ability to pass instructions and parameters to part of a core on a real time basis. Figure 4 illustrates the DIW interface with EXU blocks. Note that different EXU resources may have different instruction sizes based on the number of modes of operation DIW control of hardware functionality can be applied to optimize application specific characteristics of a design. DIW controlled gating of clock distribution to EXUs supports power reduction by clock disabling during times when the EXU is not being utilized.

Execution Unit Architecture

In addition to the controller/sequencer, each RAMA RADcore contains execution units that perform datapath operations. Datapath EXUs provisioned in each RADcore in the RAMA RADarray are:

- Data IO: The data IO EXU interfaces RADcore to RADbus and memory buses. The IO provides 8 channels of data transfer, configured for:
- 2 Read and 2 Write bus channels to RADbus, which support 16 bit 266 MHz data transfers to other RADcores in the RADarray
- 2 Read and 2 Write bus channels to Memory bus, along with request/grant signals to the memory arbiter, and dedicated programmable address generators for each port. Each MemBus IO supports 32 bit 133 MHz data transfers between RADcores and memory blocks.
- ALU: ALU EXUs support 16 bit signed/unsigned arithmetic and shift operations, and bit level masked logical and shuffle-exchange/swap operations.
- MAC: MAC EXUs support iterative 16 bit operand multiply-accumulate operations with multi-operand input registers and 48 bit internal resolution [3].
- Memory: 4 port MEM EXUs support (2 Read/2 Write) memory operations within the RADcore. Each port has a dedicated programmable address generator.
- External Logic Buffer: ELB EXU supports RADcore to off chip interfacing for external co-processing. The

ELB supports simultaneous instruction word control export, core data export, and external data import. The ELB provides elastic storage of imported and exported data in order to facilitate asynchronous transfers between RADcore and external logic clock speeds (Figure 5). The ability to interface external logic to RADcore operations is a RAMA innovation to support high performance co-processing [4].

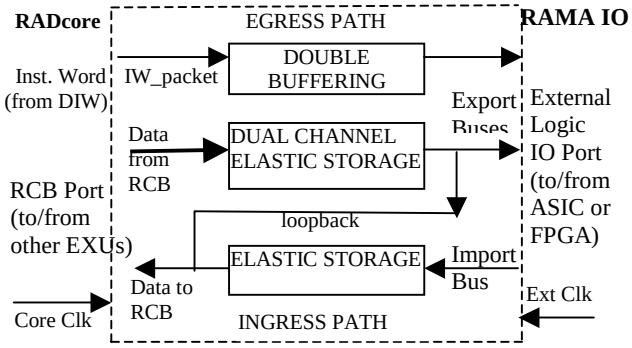

Figure 5: External Logic Buffer EXU

RISC Processor Subsystem

RAMA integrates a 32-bit configurable RISC core from ARC Cores Inc. as its on-chip host processor. The configured ARC core is a general purpose programmable RISC processor and master for RADarray and memory subsystems operation. The ARC core has separate data, instruction, and control busses with RAMA specific interfaces (Host Mem IF, Ctrl IF) to encapsulate the ARC core and provide interface compatibility to the rest of RAMA (Figure 6). The RAMA Ctrl Bus interfaces to the Controller/Sequencer of each RADcore, providing Host/Datapath co-processing communication.

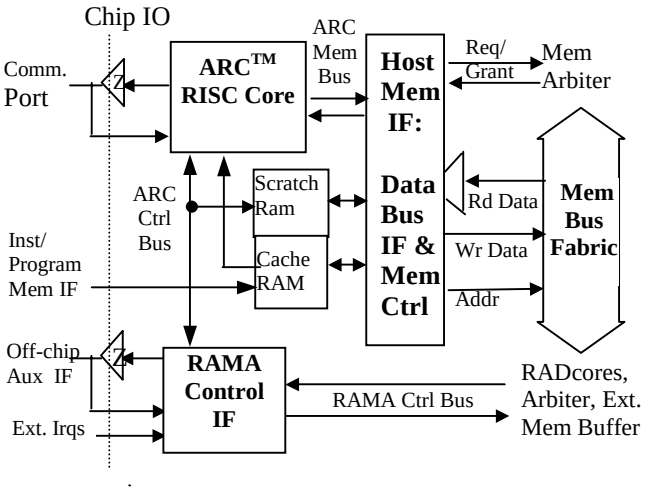

Figure 6: RAMA RISC Processor Subsystem

Memory Subsystem

The RAMA Memory subsystem supports data memory transfer between RADarray, RISC processor and internal and external memory ports. In order to support MIMD operations, RAMA has a 3 level memory architecture that provides local dedicated RAM resources to each processing core and distributed RAM resources that are available to each core on an as needed basis. All internal memories are multi-ported to allow simultaneous read/write access. Dedicated memory resources include a scratch ram in the (ARC) RISC core and memory EXUs (3 per RADcore). Five blocks of dual-ported RAM are shared (arbitrated) between RISC and RADcores. An External Memory Buffer provides external data access for RISC and RADcore as well as DMA access to internal memory.

The Memory Bus Fabric is implemented as independent Read and Write bus channels in order to implement maximum internal memory utilization and bandwidth, supporting concurrent 21.2 Gbit/s Read and Write transfer rates between processor and memory cores (Figure 7).

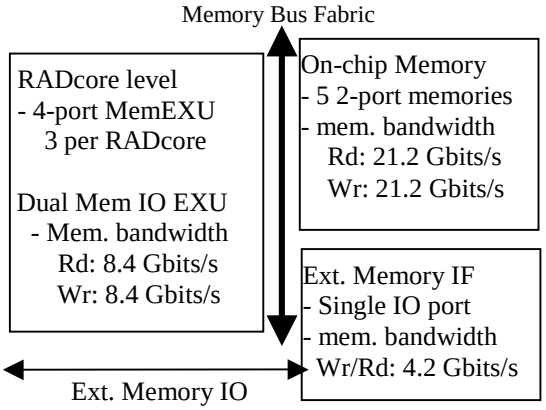

Figure 7. RAMA Memory bandwidth

The memory subsystem contains two programmable cores to facilitate multi-core operations. A Memory Arbiter controls memory access between RADcore memory IO, RISC processor memory IO, on-chip memory, and external memory interfaces by granting ownership of a memory port (Read or Write) based on programmable priority access requests from RISC or RADcore processor or external memory DMA.

Arbitration allows RADcore write and read IOs to independently request and access ownership of on-chip memory write and read ports. (Figure 8) The Arbiter insures that only one processor IO can own a given memory port at a time using priority masks for dynamic memory allocation of multiple cores requiring access to a given memory. Each memory port has its own priority mask, so arbitration is determined independently for each memory port [5]. Arbitration is autonomous (processor support is not required) under normal operating conditions.

The External Memory Buffer (EMB) interfaces RAMA blocks connected to the memory bus (RISC,

RADcores, and on-chip memory) to external (off chip) memory. The EMB supports two modes of operation, depending on whether it is buffering a processor core accessing external memory or controlling DMA for internal and external memory transfers. The EMB has interfaces to the memory arbiter and has integrated address generation logic, allowing it to autonomously control external memory accesses.

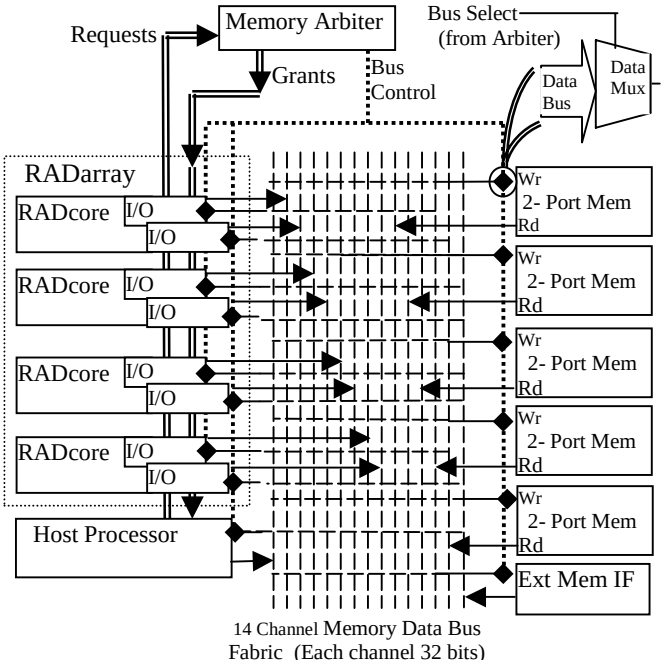

Figure 8. RAMA Processor to Memory Bus Fabric with Arbitration Driven Connectivity

Summary

A Reconfigurable Array MultiMedia Architecture (RAMA) capable of MIMD Datapath functions has been introduced. RAMA contains several innovations at both the core and subsystem level, notably the hierarchical integration of reconfigurable bussing, that may be leveraged for increased performance in diverse DSP applications. At time of paper submission, RAMA is undergoing final architectural and physical verification and is anticipated that silicon will be available by end of year 2000. More information on RAMA can be found at: http://www.infinite-tech.com.

References

[1] *N. Stollon, G. Landers, T. Lahutsky* "A High Performance Reconfigurable Signal Processor with Distributed IW Architecture", *Proceedings of DesignCon99/IP World Forum February 1999*

[2] *N. Stollon,* "A Reconfigurable Bus IP for Modular Function Integration", *Proceedings of IP99/Systems on a Chip Conference April 1999*

[3] *N. Stollon, C. Connell, G. Landers* "Tradeoffs in Multiply Accumulate Architectures for Efficient Implementation of DSP Algorithms", *Proceedings of 1999 Int. Conf. on Signal Processing Applications & Technology (ICSPAT), October 1999*

[4] *N. Stollon, R. L. Veal*. " A Reconfigurable Datapath and Programmable Logic Based System-on-a-Chip Architecture", *Proceedings of DesignCon2000/IP World Forum February 2000*

[5] *N. Stollon, G. Landers, R. L. Veal.* "Configurable Bus Arbitration for Communication Between Multiple Concurrent IP Blocks", *Proceedings of IP2000/Systems on a Chip Conference April 2000*

Author Biographies:

Glen Haas is the Vice President of Engineering at Infinite Technology Corporation. He has 32 years experience in Logic, Linear and Telecommunications semiconductor products in bipolar, NMOS, CMOS, BiCMOS and Gallium Arsenide technologies. At Texas Instruments, Mr. Haas was a Senior Member of the Technical Staff and Product Line Manager for Linear Telecom Products. Mr. Haas has a BSEE from the University of Cincinnati.

George Landers is Vice President of RAD Architecture and Development at Infinite Technology Corporation. He has 35 years experience in the semiconductor industry in product development and technical marketing in areas of memory, reconfigurable logic and signal processing. At AMD, Mr. Landers was Manager of PLD Strategic Marketing and is a co-inventor of AMD's MACH product line. Mr. Landers has a BSEE from University of California at Los Angeles.

Levon Petrosian is a Member of the Technical Staff at Infinite Technology Corporation. He has 7 years experience in reconfigurable IC and PLD design and programming, and in design of cache and memory management systems and 12 years experience in software design. Mr. Petrosian has a M.A. in Theoretical Physics from Yerevan State University and has published in diverse areas of physics.

Neal Stollon is Director of System-on-a-Chip development and is the Senior Member of the Technical Staff at Infinite Technology Corporation. Dr. Stollon has previously worked and published on diverse areas of IC design, automation, and program management at Texas Instruments and DSC Communications. Dr. Stollon has a Ph.D. in EE from Southern Methodist University and is a licensed P.E. (Tx).

Robert L. Veal is Vice President of Marketing and Sales, and leads the business development activities of Infinite Technology Corporation. Mr. Veal is a 26-year veteran of the electronics industry, working in areas of sales, marketing, and product and general management at Texas Instruments and Compass. Mr. Veal has a M.A. in Physics from Harvard University.