AFRL-RI-RS-TR-2010-037
In-House Final Technical Report
February 2010

ARCHITECTURES FOR COGNITIVE SYSTEMS

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2010-037 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
DUANE A. GILMOUR, Chief Computing Architectures Branch	EDWARD J. JONES, Deputy Chief Advanced Computing Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEBRUARY 2010	Final	October 2007 – September 2009

4. TITLE AND SUBTITLE

ARCHITECTURES FOR COGNITIVE SYSTEMS

5a. CONTRACT NUMBER
In House

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Thomas E. Renz

5d. PROJECT NUMBER
459T

5e. TASK NUMBER
AC

5f. WORK UNIT NUMBER
PM

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/RITA
525 Brooks Road
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RITA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2010-037

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2010-0406

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Architectures for Cognitive Systems research project developed a computer core that is optimized to perform massively parallel cognitive computing operations such as are required for performance of cognitive primitive operations. A highly modular many-node chip was designed which addressed power efficiency to the maximum extent possible. Each node contains an Asynchronous Field Programmable Gate Array, AFPGA, on board Static Random Access Memory, SRAM, and an Application Specific Processor core, ASP. The ultimate aim of this architecture was the creation of a dynamically configurable, highly parallel cluster of many modular nodes, to provide power efficient hardware optimization to perform complex cognitive computing operations. This project focused on the design of the core and integration across a four node chip. A follow on project will focus on creating a 3 dimensional stack of chips that is enabled by the low power usage. The chip incorporates structures to enable stacking in a small form factor. A third project will focus on system architecture issues, using many stacks to create a neuromorphic computing platform.
This report describes the completed design trades and architecture for the nodes and chip level integration. At the end of the project, the chip design was nearly ready for fabrication and will be fabricated in the first part of a follow on project.

15. SUBJECT TERMS
Many Node Computing. Cognitive Operations, Autonomic Systems, 3D Computer Architectures, Human Scale Computing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thomas E. Renz
U	U	U	UU	27	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

List of Figures

1.0 Summary

The goal of the Architectures for Cognitive Systems research project was to develop computer hardware that is optimized to perform massively parallel cognitive computing operations such as are required for performance of cognitive primitive operations. A highly modular many-node chip was designed which addressed power efficiency to the maximum extent possible. Each node consists of an Asynchronous Field Programmable Gate Array, AFPGA, on board Static Random Access Memory, SRAM, and an Application Specific Processor core, ASP. The ultimate aim of this architecture was the creation of a dynamically configurable, highly parallel cluster of many modular nodes, to provide power efficient hardware optimization to perform complex cognitive computing operations.

This project focused on the design of the core and integration across a four node chip. A follow on project will focus on creating a 3 dimensional stack of chips that is enabled by the low power usage. The chip incorporates structures to enable stacking in a small form factor. A third project will focus on system architecture issues, using many stacks to create a neuromorphic computing platform capable of 100+ Trillion Floating Point Operations per Second, TFLOPS in the space of a small rack, with power usage of less than 10kW.

This report describes the completed design trades and architecture for the nodes and chip level integration. At the end of the project, the chip design was nearly ready for fabrication and will be fabricated in the first part of the follow on project, which focuses on the multi chip stacking architecture.

2.0 Introduction

A DoD need exists for small, autonomic systems in the battlefield. Autonomy allows the creation of unmanned systems to perform complex, high risk and/or covert operations in the battlefield without the need for constant human operation. Current computing systems are not optimized to perform intelligent operations, such as environmental awareness, learning and autonomic decisions in a size, weight and power form factor that matches platforms envisioned for future use.

This project created a new computer hardware architecture to provide massively parallel computing systems needed for future autonomic operations while dramatically improving computing power per system volume and computing power per energy demand ratios.

The term “cognitive operations” can cover numerous topics from fundamental perception to conscious reflection on the nature of self. In this project, the emphasis was on lower level operations that require massively parallel computing to perform in real time at a resolution rivaling human operations. An example is processing of visual images for object recognition. The project focused on architecture development to enable massively parallel processing and the optimization of algorithms to utilize the new hardware architecture. The approach was to develop an architecture for processing the cognitive primitives that was not subject to limitations to parallelism that restricts Von Neumann type systems.

The Von Neumann computer architecture consists of a sequential instruction based processor plus external memory for storing the program or sequence of instructions, [1]. For 60 years economic, fabrication engineering and algorithm availability issues encouraged computer designs to follow the Von Neumann path by pursuing a single, fast sequential processor. With the creation of each new generation of processor, users have thought up new applications that exceeded the new capabilities, creating the need for further development. The architecture allowed designers to increase processing capacity by adding chip area and energy. The disadvantage of designing for increased sequential speed over energy is that heat increases at a higher rate than the speed of the processor. Eventually, a limit was reached where it wasn’t cost effective to increase the speed of the single processor. The solution to the heat limit was to slow the processor down and use more than one processor in parallel. However, a disadvantage of

using large separate chip processors is that latency from processor to processor and processor to memory is high. The long interconnects further increase energy use.

The Von Neumann architecture is efficient for computation that requires fast serial instructions but is subject to Amdahl's law for parallel operations due to the sequential instruction path. Amdahl's law describes the maximum speed-up to be gained from adding parallel processors to a system which has concurrent instructions spanning the parallel interface [2]. The marginal processing gain, from adding another processor, diminishes with each added processor due to cumulative wait time for concurrent instructions.

Efforts are under way to emulate human-brain scale processing. There are multiple approaches which can be differentiated by the resolution level in the emulation used and the use for the output of the emulation, e.g. [3, 4]. Of contention is whether or not emulation down to the molecular level is required for the computing system to perform and not just simulate or emulate various levels of cognitive functions. What is not in contention is the issue of the energy needed to achieve human scale operations. Given the current rate of progress in energy efficiency, it is estimated that a human scale system using the current processor and large supercomputer architectures will require megawatts to operate [4].

Human scale systems, brains, work around the limits of Von Neumann and Amdahl by using a concurrent, dynamic, massively parallel processing network. In this project, the processor was significantly reduced in size, versus commercial processors. The cognitive operation primitive was set at the functional level. We do not expect to need to emulate molecular biology to achieve performance of perception and semantic operations. As the scale of the total system is increased by clustering nodes, responsibility for cognitive primitives will move up from the traditional "each processor is performing many serial cognitive primitive operations" to a network of nodes level. Each node will be responsible for a single cognitive primitive and be capable of performing the operation very quickly. This network work load architecture will require a very large number of nodes to accommodate a large range of knowledge for cognitive operations. In this manner, the system parallelism is pushed closer to the level at which the cognitive primitive is performed. The network node becomes the functional primitive hardware unit for semantic operations. A semantic network node architecture was impractical in the past because there was more commercial benefit in building one very large, fast processor in a fixed area than to divide the same chip area into many smaller

processors. Unfortunately, large processor systems cost too much in area, power and processing time to create the number of nodes needed to process millions of semantic primitives.

The features that make the architecture developed in this project useful for cognitive operations also make it useful for many other military applications. The architecture makes major progress in the trade space for size, weight, energy demand, cyber security, system reliability, processing speed, modularity, bandwidth internal to a cluster, and flexibility of operation and resource control. The Floating Point Unit, FPU in our ASP was optimized for extremely energy efficient processing of Fast Fourier Transform algorithms. This makes the architecture a very powerful system for Parallel Discrete Event Simulation used in planning tools. The modularity and ability to dynamically reallocate resources provides opportunities for several cyber security hardware features including node level Advanced Encryption Standard, AES encryption.

The project was divided into areas of hardware, software and applications. Concurrent development was performed in each area in relation to each of the other areas. This provided maximum optimization and usage of features. The Hardware Design Results section describes progress in the chip architecture design. The Software Results section describes results in the alteration of the RTEMS operation system for use in this project and the creation of the mesh address architecture for core to core operations. The Applications section describes interactions with projects focused on neuromorphic high performance computing applications. The goal was to obtain hardware and software trades for optimization of algorithm processing and to provide algorithm optimization to make maximum use of hardware and software features. The Future Uses section describes planned follow on projects for this technology. This report describes the work completed to date as of the original project end date of September 30 2009. Work on the chip design and larger system is continuing under the follow on project, Cognitive Cluster on a Chip.

3.0 Methods, Assumptions, Procedures

This project created a new modular, fungible computing node that enables compact massively parallel computing with a memory / processing / communications geometry that is optimized towards the natural geometry of basic autonomic operations. This entailed identification of the specification trade space, component design, system design and operating software development. Power efficiency was aggressively pursued at all levels of the core and cluster to enable high cluster system density.

3.1 Hardware Design

The goal of the hardware design was to maximize power efficiency, core to core connectivity and system modularity while minimizing communication latency. A core concept was chosen that consisted of a small, Application Specific Processor, ASP, a block of core associated Random Access Memory, RAM, and a block of Asynchronous Field Programmable Gate Array, AFPGA. A 128 bit AES hardware block was added to the core design. The fundamental concept was to optimize for the lowest area per core that would support several chosen applications to enable the highest possible core density in a cluster system on a chip. Power efficiency, connectivity and modularity facilitated high density.

A processor was designed for 65 nm fabrication technology with numerous features to enhance performance under size, weight and power restrictions. Design choices were also made to favor modularity, security and dynamic user interaction. For modularity the cores were designed with independent components rather than as monolithic integrated blocks. This allows soft, (post fabrication) and hard, (next version fabrication) changes in individual blocks without requiring redesign of the whole core. New hardware based security features were continuously sought throughout the design process.

Fungibility, the ability to easily interchange units with other, like units, was a continuous design consideration in this project. The difference between fungible and homogeneous is the degree of interchangeability. Parts of a system can be homogeneous but not interchangeable. Parts must be fungible to enable using a neighboring part to re-route around a malfunctioning part. A fungible core aids the creation of clusters that grow or shrink with the demands of the

computation and in the process allows tailoring of power consumption to efficiently fit the computation.

Some blocks such as the floating point multiplier and floating point adder in the ASP were adopted from a previous, 130nm project. Most hardware blocks were newly designed for this project.

3.2 Software Design

Control software design in the project was begun with an open version of Real Time Executive for Multiprocessor Systems, RTEMS. This choice enabled the Air Force to retain the complete source code and knowledge of everything in the operating system, OS. Software needs for the project included a processor and node level operating system and software to enable the control functions envisioned for the AFPGA. The control software will direct resources for heterogeneous parallelization of computing tasks while effecting dynamic power efficiency measures. Dynamic resource allocation will enable future system architecture configurations such as the ability to turn processors on and off as needed and allocate memory.

Hardware and operating system features were added to the OS to allow user changes to the microcode. Modules were added to provide functionality for the AFPGA, security features and memory access. Concurrent design with hardware was used to maximize functionality and efficiency.

3.3 Applications

The cognitive operations models used in this project were chosen for their representation of the state of the art, user applicability, model accessibility and representation of a range of model geometries and HW/SW requirements. Concurrent development was used to increase efficiency. For example, Fast Fourier Transform, FFT uses the common Vector-Matrix Multiply operation. On the hardware side, registers were added to the Floating Point Unit and configured to maximize throughput. On the software side, new microinstructions were created to take advantage of the new register configuration and reduce the number of read / write calls. On the application side, the algorithm was rewritten to make the most efficient use of the new microinstructions. The ability to perform concurrent development was made possible by the built in modularity and flexible microcode OS.

3.4 Participants

The project was led by the Air Force Research Laboratory Advanced Computing Division. AFRL researchers provided the Application Specific Processor, ASP design and project integration. The Asynchronous Field Programmable Gate Array design was provided by Cornell University. Cornell also provided some of the design flow facilities. ITT industries provided the operating system, design integration expertise and a cognitive computing model. Oklahoma State contributed cell and processor design, design tool flow and block integration. Binghamton University contributed hardware analysis for timing and heat, and a cognitive model. There was input on applications and large scale hardware systems from several other basic research projects.

3.5 Schedule

The project was originally scheduled for October 1 2007 to September 30 2009. Delays caused collaborators contracts to not start until June of 2008. The original schedule called for the design to be ready for the February 2009 foundry run, but the 9 month delay in the start caused the design completion to slip past the deadline. The next foundry run with the same technology was set for February 2010, after the scheduled end of the project. As of the writing of this report, the design work is on schedule to make the February 2010 design run. The funding was put in place for the 2010 fabrication run and the fabrication and testing work is expected to be completed under the follow on Cognitive Cluster on a Chip set of projects.

4.0 Results

The design of the chip is nearly complete as of the scheduled end of this project. The design work will continue under the Cognitive Cluster on a Chip project and will go to tape-out for fabrication in February 2010. This section describes progress to date in the areas of hardware architecture and chip design and operating system architecture.

4.1 Hardware

The goal of the hardware design was to maximize power efficiency, core to core connectivity and system modularity while minimizing communication latency. At each decision point in the design, new features were weighed against the cost in core area, the number of instructions required to run the feature and the security vulnerabilities added or removed by the feature. Modularity and efficient core to core connectivity was considered a significant power demand issue. Power efficiency, connectivity and modularity facilitate fabrication of high density clusters.

A core architecture was chosen that consisted of a small, Application Specific Processor, ASP, a block of core associated Random Access Memory, RAM, and a block of Asynchronous Field Programmable Gate Array, AFPGA. Figure 1 shows the major components of the chip design. Four cores are tiled onto a 7mm by 7mm chip. Block sizes, shapes and locations may change as the design goes to final place and route. The fabrication technology chosen for use in this project was the IBM 65 nm Trusted Foundry process. Fabrication will be accomplished through a multi-project wafer run at the National Security Agency, NSA Trusted Access Program Office, TAPO.

The Static Random Access Memory, SRAM was designed using the Virage Logic cell library for 65nm. A ½ MB block was fit into the memory area budget of 4mm^2. The address line is 512 bits wide and the bus is running at the processor speed of 500MHz. Each core has a 32KB bank of non-cacheable memory space termed the Input / Output Register set. These registers provide control and status information pertaining to constituent members of the element that include the configuration of the core, state machines for input and output data flow control, designation of the local Media Access Control, MAC address, mesh routing table and mesh control registers.

The floating point unit, FPU was designed for efficient operation and to accommodate microcode that also promotes efficiency. There are separate single and double precision adders and multipliers with the ability to execute 2 double or 4 single precision instructions per clock. The data path accommodates a 192 bit microinstruction. Power simulation estimates the processor achieving the project goal of 50 GFLOPS/W on a FPU intensive operation. The microcode store was designed to enable fast hardware morphing of the opcode. This nearly eliminates any time penalty for using core level opcode obfuscation. Provision was made for secure control of the opcode morphing key through the provision and proper configuration of IEEE 1149.1 JTAG interface, (Joint Test Action Group).

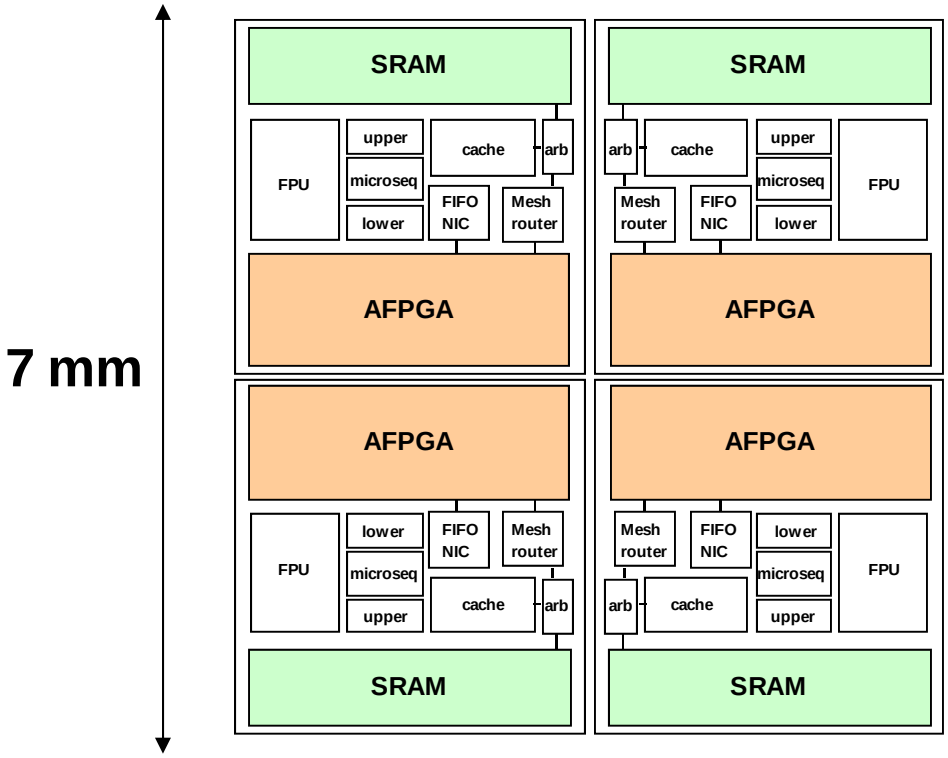

Figure 1. Chip Level Block Layout

The processor cache is 128 lines by 512 bits wide. An additional bank of RAM was added to facilitate FFT calculations. Speed testing has the processor running at the project goal of 500MHz.

A Mesh routing system was designed to enable shared resources across a large cluster of cores. Figure 2 shows the major components of the Mesh design. Between each processor and its cache, there is an Input / Output, I/O arbiter and bank of registers that control routing of

messages. Routing can go to the processor's memory, or through mesh switching to other processor's memory. Off chip SRAM is also accessed by this path. Blocks of AFPGA all communicate through the Mesh. Chip level interfaces include 10 Gigabit Media Independent Interface, XGMII Ethernet, SpaceWire Lite, RAW and secure access JTAG.

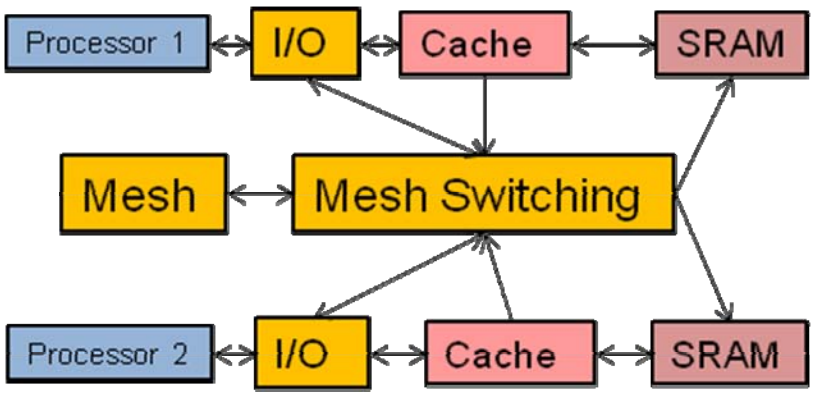

Figure 2. Mesh Hardware Blocks Between Two Cores

The AFPGA block is organized into 256 reconfigurable tiles with 2Kb of SRAM per tile. In the 4mm^2 budgeted for the AFPGA block there can be 1024 four bit look up tables running at close to 1GHz. The AFPGA is internally synchronous and has two user controlled asynchronous interfaces with the processor. The direct interface provides for 64 bit words and an address space of 14 bits. A Mesh interface provides an 8 bit interface with the processor. There are flyover links between AFPGAs with fixed connectivity's and 8 bit off chip asynchronous I/O. The AFPGA internal clock was demonstrated by simulation to be near the project goal of 1 GHz.

Each core controls a loosely dedicated section of FPGA fabric. The FPGAs are arranged in pairs, with routing channels connecting two adjacent FPGAs. This allows a microprocessor to use more FPGA fabric than is available in its dedicated section. Because programming access is exclusive, in order for processor A to borrow FPGA fabric from processor B, A requests that B configures its FPGA appropriately and connects the inter-FPGA routing channels.

The processor interacts with the FPGA in three ways:

- **Configuration** - Resetting the FPGA at start up and making changes in AFPGA
- **Programming** - Logic functions with standard bits

- **Privileged bits** - JTAG accessed for write protection control and normal operation (sending and receiving data to/from the FPGA).

There are three modes of interaction timing:

- **Timing-driven -** Data is sent to FPGA, result of computation is retrieved after some number of cycles. Given the long latency of other methods, this is expected to be the most popular method of operation.
- **Polling -** Data is sent to FPGA, FPGA signals to the microprocessor when data is ready for retrieval; core is awaiting the results in a microcode loop and proceeds upon arrival.
- **Interrupt -** Same as polling, except core has proceeded with other work and takes an interrupt (IRQ1 = output_data_ready) when the result is delivered.

The 128 bit Advanced Encryption Standard block designed by ACICS.ws was obtained from OpenCore [5]. The block was modified for the fabrication technology used in this project and found to require less than 0.1 mm^2 in area. The AES is addressed through JTAG with the ability to maintain 4 keys simultaneously. Three of the keys are used for data operations and one is reserved for local core use only. The National Security Agency, NSA has approved the use of accredited 128 bit AES based encryption for protection of information up to the SECRET level. For TOP SECRET and above, 192 bit or 256 bit AES is required [6].

Information Assurance was built into all hardware and software blocks at all stages of the design. A nontrivial technique was used to maintain trusted access throughout the project to all designs. The personnel working on the project were approved before access was given and secure areas were set up at collaborator's facilities to maintain security. Fabrication was arranged through the Trusted Foundry operated by NSA.

4.2 Operating System and Software

The operating system designed for this project began with Real Time Executive for Multiprocessor Systems, RTEMS which is a simple, open source operating system for parallel and embedded processor systems. The choice of RTEMS allowed the retention of the source code so that changes can be made to the operating system and the exact source code is known at

all times. The microcode physical store was designed to provide for insertion of new microinstructions, at the design stage and later, as needed by users. New instructions were created and emulated in about two weeks each to improve a Fast Fourier Transform algorithm, speed up hardware processing of morphing opcodes and implement AES encryption.

Microinstructions were written to securely control the JTAG interface. Microinstructions written for the AFPGA interface allow the user to tailor the rate of communication between the processor and the AFPGA. A user with an application that makes many data passes could set the interface to access at a set interval. A user with few data passes could save wait time by setting the interface call rate to only open the channel at an irregular demand signal.

The ability to tailor the operating system at the microinstruction level was used to quickly deliver a 10x improvement in the GFLOPs/Watt ratio for the FFT demonstration algorithm by creating and implementing pre-fetch and evict instructions. The addition of "squash" instructions for efficiently diagonalizing a matrix provided an 8x performance improvement for the Brain State in a Box demonstration project. Processor utilization efficiency was modeled for the improved BSB model run on a Cell Broadband Engine system and 70% of peak processor performance was reached. On the Architectures for Cognitive Systems, ACS processor, with registers designed to better utilize the new microinstructions, 81% of peak was reached in emulation.

An address architecture was created for the Mesh routing system. The address architecture allows every node in the stack to see and access every other node's memory and I/O registers with a local address. Figure 3 shows the architecture for a 16 node, 32 bit version of the address space. The 16 node's SRAM memories occupy the lowest addresses with ½ MB of allocated space per node. The upper addresses are reserved for the I/O registers of the same 16 nodes. The remaining addresses were reserved for off chip SDRAM. A block of off chip SDRAM will be needed to hold the OS for the smaller versions of the system. A goal for future systems is to scale to the point where the OS can be distributed among a small number of dedicated nodes. The address space will be expanded to at least 64 bits in the future to provide local addressing for large clusters of nodes on stacks and clusters of stacks beyond that.

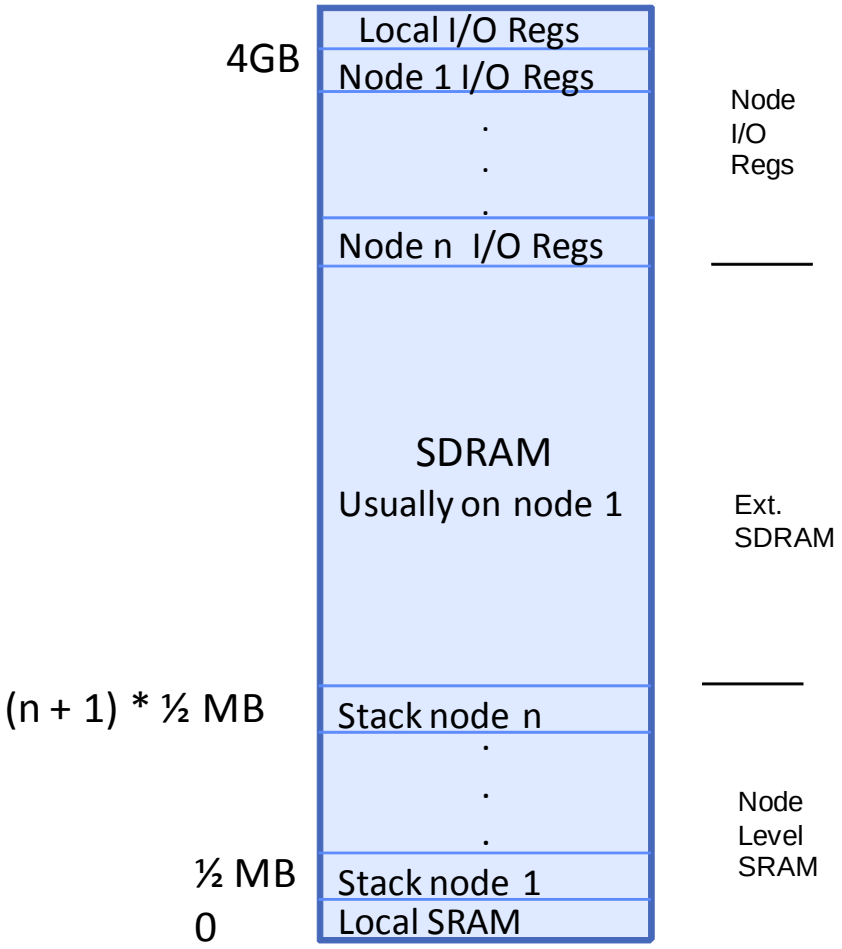

Figure 3. Address Architecture for the Mesh

4.3 Cognitive Models

Cognitive model tasks funded by this project include work performed by the Binghamton group with their cognitive operations model and work on a neuromorphic based model of visual processing performed by ITT. The cognitive models were chosen not so much to make major strides in cognitive sciences, but rather to provide 360^{o} feedback for concurrent optimization in cognitive operations of the total hardware, operating system and application models. Trades were made in each of the three areas, HW, OS and Applications to optimize for implementation of the other two. Further trades were made to each area to take even more advantage of the new optimizations. This process was continuous throughout the project.

The Binghamton BSB project created and implemented a new design for a 256 state 128 neuron Brain State in a Box model. The model was modularized to be able to scale with the availability of local memory. Load time was optimized by taking advantage of parallel access to local memory. An instantiation of the model was performed in a FPGA as shown in Figure 4. Further information can be found in the Binghamton project final report [7].

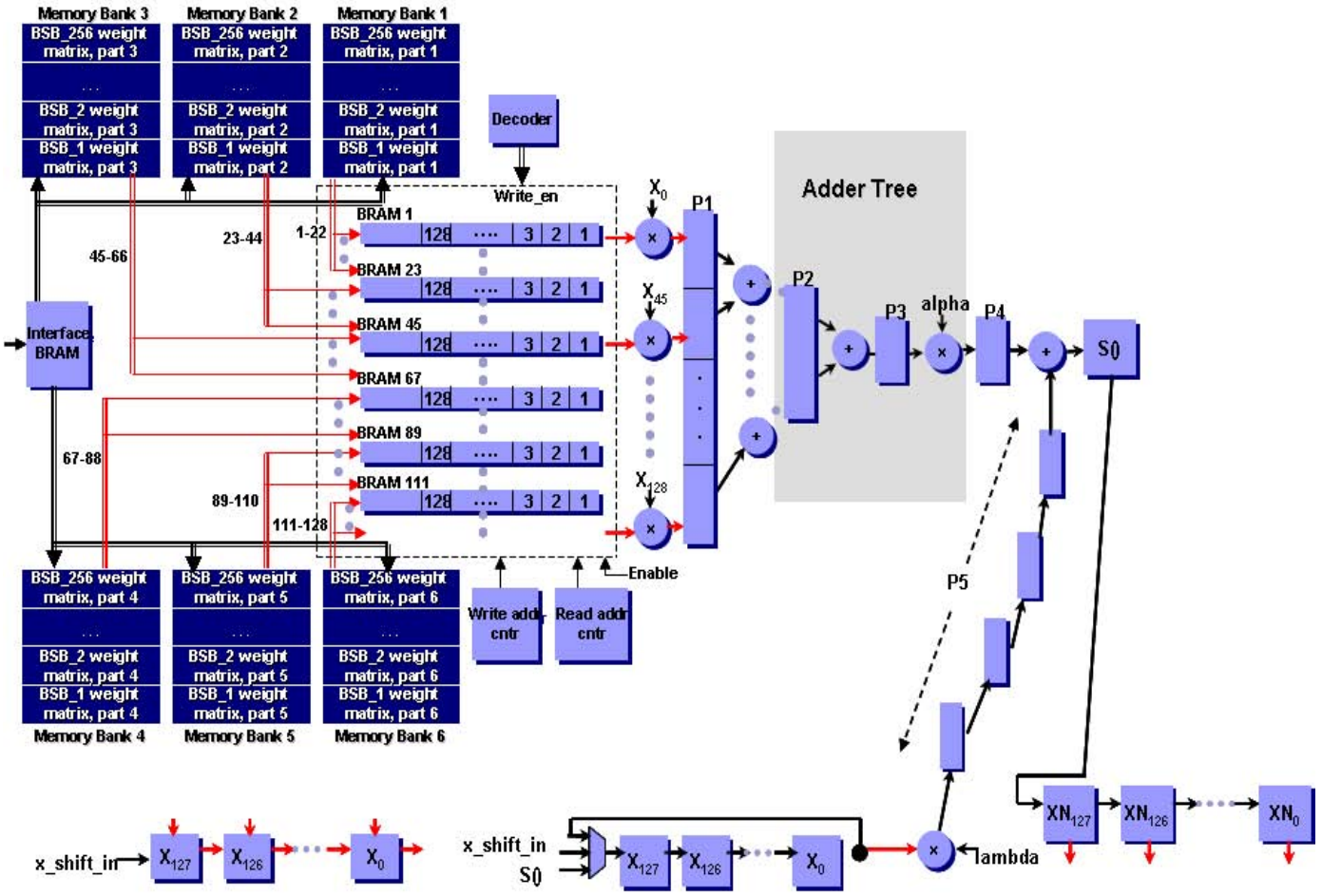

Figure 4. Hardware Design for 256 128-neuron BSB models

The ITT cognitive task focused on large scale cognitive applications. Algorithms were investigated for resource requirements and prospects for optimization. The algorithms investigated include:

- Bayesian tree networks: hypothesized as a mechanism of visual perception [8,9,10]. The algorithm is described in detail in [11].
- BSB implementation as a mechanism of visual and auditory perception [12].

- Spiky neural model: the basis of hypothesis regarding the dynamical mechanisms of cognition [13].
- Simple cell model: Anatomical mechanism [14]. The geometry of receptive field patterns is fundamental to visual perception.
- Confabulation: An algorithm which hypothesizes how networks of neurons respond to sequences of stimuli [15].

Each of the algorithms and their conceptual models were examined for the cognitive primitive performed and the processing power and resource requirements needed to run them. The neuromorphic model of the visual cortex and the language confabulation model were chosen for further development. Resource requirements from those models were added to the ACS design trades and work was performed to improve the models for massively parallel systems. Both models were optimized to run on the Cell Broadband Engine architecture. Emulation was performed with a focus on areas to further optimize models and the system running the models. Further details are available in [16].

5.0 Conclusions

The goal of the Architectures for Cognitive Systems research project was to develop computer hardware that is optimized to perform massively parallel cognitive computing operations such as are required for performance of cognitive primitive operations. A highly modular many-node chip was designed which addressed power efficiency to the maximum extent possible. Each node consists of an Asynchronous Field Programmable Gate Array, (AFPGA), on board Static Random Access Memory, SRAM, and an Application Specific Processor core, (ASP). The ultimate aim of this architecture is the creation of a dynamically configurable, highly parallel cluster of many modular nodes, to provide power efficient hardware optimization to perform complex cognitive computing operations.

This report described the completed design trades and architecture for the nodes and chip level integration. Simulations of the current design show the chips reaching project goals for performance. A four core chip was described with ½ MB of SRAM, a 128 bit AES, a 1000 look up table block of AFPGA and a mesh router associated with each core. The processor speed was demonstrated in simulation at the project goal of 500MHZ. The AFPGA was demonstrated in simulation at near 1 GHz.

The project was originally scheduled for October 1 2007 to September 30 2009. Delays caused collaborators contracts to not start until June of 2008. As of the writing of this report, the design work is on schedule to make the February 2010 foundry run. The funding was put in place for the 2010 fabrication run and the fabrication and testing work is expected to be completed under the follow on Cognitive Cluster on a Chip set of projects.

6.0 Future Work

This project focused on the design and fabrication of a modular core and local mesh. Transitions for chips with this architecture could include very small systems that have very limited power available but require much more computing power than is currently available at that energy level.

Figure 5 shows the plan for future integration. Thinned 4 to 16 node chips would be stacked with vertical vias for layer to layer connections. The stacks containing 100s of cores would be connected by the stack level mesh for complete sharing of resources. A goal of the stack project is less than 1 watt per tier in the stack. That would enable stacks of up to 50 tiers depending on the extent of available stacking technology. Transition from the stack level of integration would go to systems that require the TFLOPS of processing in systems with only a few cubic centimeters space and less than 100W available power. That includes autonomous missiles, UAVs and small sensor platforms.

Another long term development track involves mounting many stacks on a board system to approach the computational scale of the complete neocortex. It is expected that the 3D integration into stacks would be a three year project and full scale system development would entail another fabrication run at 32nm which would be large enough to make thousands of chips. Figure 5 shows the planned development from nodes to stacks to mega-node systems.

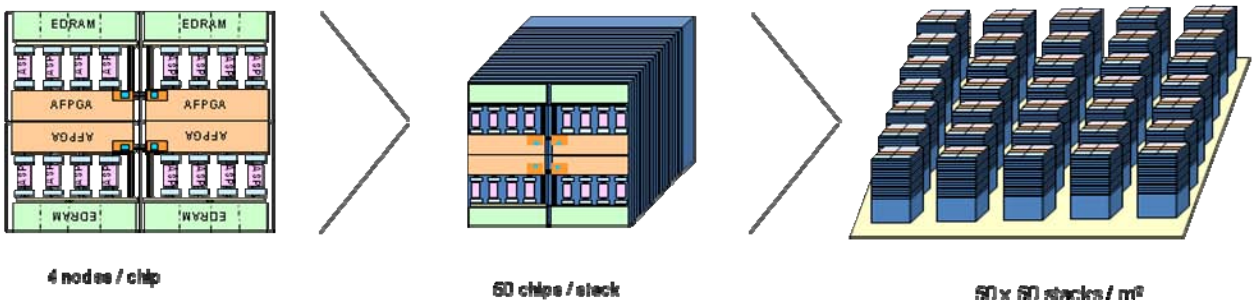

Figure 5. Multi-Scale Modular System Integration Scheme

The use of state of the art commercial fabrication makes fabrication funding the limit on the potential scale of the final system. With reasonable expectations for the processing power per energy and peak watts per chip it is possible scale to 10 PetaFlops in a million node system that consumes less than 100kW.

It is envisioned that the ultimate version of the operating system will allow executive level control of simultaneous formation of clusters of processors within stacks and clusters of stacks in multi-stack systems in order to accomplish multiple large parallel tasks. The executive function would dynamically determine the cluster geometry required to perform a task, parse out resources and adjust for minimum energy use. The complete system would be able to perform multiple cognitive operations, system control and communication tasks simultaneously in response to internal and external demand. These are the control requirements for a cognitive platform capable of achieving an assigned mission under internal, autonomic control.

7.0 References

[1] Von Neumann, J., "First Draft of a Report on the EDVAC" June 1945, Univ. of Penn., Draft report made by Von Neumann on work by J. Eckert, J. Mauchly and others on development of the concept of stored programs for computers. It led to the modern general platform design.

[2] Amdahl, G. M., "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities", Proc. AFIPS Conf., V. 30, pp. 483-485, 1967

[3] http://bluebrain.epfl.ch/

[4] http://www-03.ibm.com/press/us/en/pressrelease/28842.wss#release

[5] ASICS.ws IP design, www.ASICS.ws, www.OpenCore.org

[6] CNSS Policy #15, National Security Agency, www.nsstissc.gov

[7]Wu, Q., "Large Scale Hybrid Computing Architectures for Neocortical Models", AFRL-RI-RS-TR-2008-294, Binghamton University, PO Box 6000, Dept. of Electrical Engineering, Binghamton NY 13902, 2008

[8] Lee, T., Mumford, D., "Hierarchical Bayesian Inference in the Visual Cortex", Journal of the Optical Society of America, V. 2, N. 7, pp. 1434–1448, July 2003

[9] George, D., Hawkins, J., "A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex", Proceedings of the International Joint Conference on Neural Networks. IEEE, 2005

[10] Dean, T., "Hierarchical Expectation Refinement for Learning Generative Perception Models", Technical Report, Brown University, Providence, Rhode Island, Aug 2005

[11] Pearl, J., "Probabilistic Reasoning in Intelligent Systems," Morgan Kaufman Publishers, San Francisco, California, 1988

[12] Anderson, J., "The BSB model: A Simple Nonlinear Autoassociative Neural Network, Associative Neural Memories", Oxford University Press, Inc., New York, NY, 1993

[13] Izhikevich, E., "Polychronization: Computation with Spikes", Neural Computation, V. 18, pp.245-282, 2006

[14] Hubel D., Wiesel T., "Receptive Fields and Functional Architecture of Monkey Striate Cortex"., J. Physiol., V. 195, pp. 215–243, 1968

[15] Hecht-Nielsen R., "Mechanism of Cognition", Biomimetics: Biologically Inspired

Technologies, Bar-Cohen, Y. [Ed.], CRC Press, Boca Raton, FL, 2006
[16] Fitzgerald, D., et.al., “Advanced Computing Architecture Technology and Applications”, AFRL Final Technical Report, ITT Corporation Advanced Engineering and Sciences, 474 Phoenix Drive, Rome, NY, 13441, 2010

8.0 Acronyms

ACA	Advanced Computing Architectures
ACS	Architectures for Cognitive Systems
AES	Advances Encryption Standard
AFIT	Air Force Institute of Technology
AFPGA	Asynchronous Field Programmable Gate Array
AFRL	Air Force Research Laboratory
AI	Artificial Intelligence
ASP	Application Specific Processor
BSB	Brain State in a Box
Core	Generally considered a Processor and its associated local Memory
DMPI	Dynamic Message Passing Interface
EDRAM	Embedded Dynamic Random Access Memory
FFT	Fast Fourier Transform
FPASP	Floating Point Application Specific Processor
FPU	Floating Point Unit
GFLOPS	Giga (Billion) Floating Point Operations per Second
HW	Hardware
IP	Internet Protocol
JBI	Joint Battlespace Infosphere
JTAG	Joint Test Action Group
MAC	Media Access Control
MB	Mega Bytes
MHz	Million Cycles per second
Node	The functional unit in this project: one aFPGA and connected cores
OS	Operating System
SRAM	Static Random Access Memory
RTEMS	Real Time Executive for Multiprocessor Systems
Stack	The set of stacked, thinned chips

SW	Software
TFLOPS	Trillion Floating Point Operations per Second
TPM	Trusted Platform Module
XGMII	10 Gigabit Media Independent Interface