Page 1 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003

Volume I Technical and Management Proposal

Title: Energy-Efficient Butterfly FPGA Hardware and Programming Tools

A proposal submitted to Dr. William Harrod, DARPA/TCTO in response to

DARPA-BAA 10-78:	Omnipresent High Performance Computing (OHPC)
Technical Area:	Energy Efficient Computing
Lead Organization:	University of California, Los Angeles (UCLA) Department of Electrical Engineering Los Angeles, CA 90095-1594
Type of Business:	Other Educational
Team Members:	Dejan Markovic (PI) Venkat Konda (Consultant)

Technical Point of Contact:

Administrative Point of Contact:

Office of Contract and Grant Administration

UCLA Senior Grant Analyst

11000 Kinross Ave, Suite 102

Los Angeles, CA 90095-1406

Email: ocga5@research.ucla.edu

Tel: (310) 794-0155

Fax: (310) 943-1658

Ms. Julia Zhu

Dr. Dejan Markovic, PI UCLA Associate Professor Electrical Engineering Department 56-147D Engineering IV Building 420 Westwood Plaza Los Angeles, CA 90095-1594

Tel: (310) 825-8656 Fax: (310) 206-8495 Email: dejan@ee.ucla.edu

Total funds requested: \$2,374,111

Year 1:	\$789,927
Year 2:	\$792,100
Year 3:	\$792,086

UCLA

Date of proposal: August 4, 2010

Page 2 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003

UNIVERSITY OF CALIFORNIA, LOS ANGELES

 $\mathsf{BERKELEY} \cdot \mathsf{DAVIS} \cdot \mathsf{IRVINE} \cdot \mathsf{LOS} \ \mathsf{ANGELES} \cdot \mathsf{MERCED} \cdot \mathsf{RIVERSIDE} \cdot \mathsf{SAN} \ \mathsf{DIEGO} \cdot \mathsf{SAN} \ \mathsf{FRANCISCO}$



SANTA BARBARA • SANTA CRUZ

OFFICE OF CONTRACT AND GRANT ADMINISTRATION BOX 951406 11000 KINROSS, SUITE 102 LOS ANGELES, CALIFORNIA 90095-1406

> PHONE: (310) 794-0102 FAX: (310) 794-0631

UCLA

www.research.ucla.edu/ocga

August 5, 2010

DARPA/TCTO ATTN: DARPA-BAA-10-78 3701 N. Fairfax Drive Arlington, VA 22203-1714

The Regents of the University of California, Los Angeles, is pleased to submit the following proposal in response to solicitation DARPA-BAA-10-78.

Title: "E	nergy-Efficient Butterfly FPGA Hardware and Programming Tools.
Requested Period of Performance	: September 15, 2010 – September 14, 2013
Amount Requested:	\$2,374,111
Principal Investigator:	Dr. Dejan Markovic Department of Electrical Engineering dejan@ee.ucla.edu 310-825-8656

This application is being submitted in contemplation of an agreement containing mutually agreeable terms and conditions applicable to educational institutions conducting unclassified fundamental research.

Since UCLA is a public/State institution, open dissemination of research results and information, commitment to students, accessibility for research purposes, and legal integrity and consistency are part of the University's Principles/Policy. The University does not discriminate and impose restrictions on any individual as a result of their nationalities.

If an award is made, please be advised that if it is funded by budget category 6.3(Advanced Research) and is considered Non-fundamental research, we will not be able to accept the award due to publication restrictions.

Your favorable consideration of this proposal would be appreciated. Technical questions should be directed to Dr. Markovic. Administrative and contractual questions, should be directed to me at (310) 794-0155 or via email at jzhu@research.ucla.edu.

Sincerely,

Julia Zhu

Julia Zhu Senior Grant Analyst

Table of Contents

Executive Summary	3
Section II – Technical Details	5
2.1. PowerPoint Summary Chart	5
2.2. Innovative Claims for the Proposed Research	6
Problem Description	6
Research Goals	6
Expected Impact	7
2.3. Proposal Roadmap	8
2.4. Technical Approach	10
2.4.1. Network Architecture and Routing Tools	14
2.4.2. Hardware Design	15
2.4.3. Hardware Mapping	19
Demonstrations and Technology Transition	22
2.5. Statement of Work	24
2.6. Intellectual Property	26
2.7. Management Plan	28
2.8. Schedule and Milestones	30
2.8.1. Schedule Graphic	30
2.8.2. Detailed Task Description	31
2.8.3. Project Management and Interaction Plan	33
2.9. Personnel, Qualifications, and Commitments	34
2.10. Organizational Conflict of Interest Affirmations and Disclosure	36
2.11. Human Use	39
2.12. Animal Use	38
2.13. Statement of Unique Capability Provided by Government or Government-Funded Team Member	39
2.14. Government or Government-funded Team Member Eligibility	40
2.15. Facilities	41
References	42
BEEcube Support Letter	43

Executive Summary

UCLA offers to perform research on a revolutionary new FPGA technology consisting of FPGA hardware and supporting mapping tools. We will design, fabricate, and test hierarchical FPGA interconnect network to demonstrate FPGA technology that is 15x more energy-efficient than existing FPGAs. The new interconnect architecture allows for significant reduction in the number of switch points, buffers, and wire length in comparison to standard 2D-mesh architecture used by existing FPGAs. The proposed technology is a radical departure from 2D-mesh design, which for N logic blocks has complexity $O(N^2)$, incomplete and heuristic routing. The proposed technology has only $O(N \cdot \log_2 N)$ complexity, complete and fully deterministic routing. The proposed technology has significant benefits: 15x lower power, 3x lower area, 2x higher performance compared to existing FPGA technology. The new FPGA technology will be used to demonstrate HPC benchmarks with a 15x higher power efficiency for DOD and commercial users. The PI has established interactions with industrial partners that will lead to the transition of ideas into the commercial space.

Section II - Technical Details

2.1. PowerPoint Summary Chart



2.2. Innovative Claims for the Proposed Research

Problem Description

Today's programmable FPGA devices are expensive in size, power, performance, scalability and flexibility. All of this is due to a fundamental problem in 2D-mesh interconnect architecture: it is large in size, has long latency, consumes lots of power, and is not scalable. Interconnect takes more than 75% of the FPGA chip area. Large number of inactive transistors also results in significant leakage power (about 50% of the total FPGA power). Due to inefficient interconnect architecture, there is a 30-50x energy-efficiency gap between FPGA and dedicated chips (Fig. 1).



Figure 1: Energy efficiency for various computing architectures: microprocessors, general purpose DSPs, FPGAs, and dedicated chips. The study is based on chips from the ISSCC conference (normalized to the same technology). FPGAs with DSP cores are 30-50x less energy efficient than dedicated chips.

Research Goals

We will integrate hierarchical interconnect network to demonstrate significant improvements in speed, power, and area as compared to existing FGPAs technology. The hierarchical interconnect architecture requires at least 3x smaller number of active network elements, switch points and drivers. This is illustrated in Fig. 2 for a very simple 2x2 example.



Figure 2: 2D-Mesh and Konda networks for a design consisting of 4 CLB blocks.

For larger number N of configurable logic elements, the benefits of hierarchical network will be even more pronounced (Table 1). Such large cost of the 2D-mesh architecture forces designers to employ heuristics to reduce the number of switch points, which results in insufficient connectivity. The hierarchical network provides complete and deterministic routing.

Number of LUTs	2D-Mesh	Konda butterfly	Savings factor
1 k	1 M	9.97 k	100x
100 k	10 B	1.66 M	6,200x

 Table 1: Number of connections in 2D-Mesh and Konda networks.

Expected Impact

The new FPGA platform will provide significant savings in power compared to today's FPGAs as shown in Fig. 3. Our FPGA technology, which includes hardware and supporting mapping tools, will provide an estimated 15x power reduction as compared to conventional FPGAs.



Figure 3: Power consumption for a range of applications. New FPGA will provide significant power reduction compared to typical Virtex-5 FPGA (normalized to the same technology).

We will provide new FPGA technology consisting of hardware and mapping tools. The hardware and mapping tools will provide significant impacts: 15x lower power, 3x lower area, 2x higher performance compared to existing FPGA technology. The new FPGA technology will be used to demonstrate HPC benchmarks with a 15x higher power efficiency for DOD and commercial users. Equivalently, our FPGA technology can provide >10x higher throughput for the same amount of power (as shown in Fig. 3). This technology will be of use for HPC applications and many other DOD applications which use FPGA technology.

2.3. Proposal Roadmap

Main goals of the proposed research: The main goal of the program is to develop energyefficient programmable hardware and supporting software mapping tools. The hardware is based on hierarchical interconnect architecture that provides significant reduction in interconnect complexity as compared to today's FPGA hardware. With a combination of new interconnect architecture and supporting toolflow, we project over a 15x improvement in energy efficiency while also considerably reducing chip area and improving performance. The proposed work builds on patent-protected network architecture and successful chip demonstrations. The work proposed here focuses on the investigation of needed level of connectivity for large-sale designs, and supporting mapping tools to make the technology accessible to end users.

Tangible benefits to end users: Over a 15x improvement in energy-efficiency, considerable reduction in chip area (3-4x), and considerable improvement in performance (> 2x) compared to today's FPGA chips. Mapping tools will be developed to automatically map algorithms into hardware and abstract away hardware-specific details from end users.

Critical technical barriers: Hierarchical interconnect networks have been known to the academic and industrial community for a long time, but physical realization of these networks precluded their successful deployment. **The critical difficulty associated with the hierarchical networks is routing congestion during chip synthesis.** Leopard Logic, Inc, is one example of a company that failed to deploy hierarchical interconnect architecture. FPGA startups today, most notably Abound Logic, Tier Logic, Blue Chip Designs, and Achronix, provide customized solutions for increased logic density or speed, but they still don't solve the problem of power inefficiency associated with FPGA chip interconnects.

Main elements of the proposed technical approach: Our approach is based on alternating vertical and horizontal routing. LUTs (or any other processing elements) are partitioned in a 2-D floorplan with switch-boxes placed to allow full routability. An *N*-LUT design requires $log_2(N)$ levels of switch-boxes. Simple example of N = 4 is shown below to illustrate the concept.



Figure 4: Hierarchical Konda interconnect architecture. $O(N \cdot \log_2 N)$ interconnect switches are required for full connectivity. Routing is fully deterministic.

In the case shown in Fig. 4, 2 levels of switch-boxes are required for N = 4 LUTs. LUTs with indices from 0 N/2 - 1 are placed on the left, the remaining LUTs are placed on the right. Switch-boxes are placed next to the LUT columns. Routing between elements with adjacent index is provided as a vertical connection (1st level routing); routing between elements with 2 indices apart is provided with a horizontal connection (2nd level routing). The routing continues in vertical/horizontal fashion for larger *N*.

Basis of confidence: Konda network architecture is a patent-protected technology that is recognized by many semiconductor companies including Cisco, Xilinx, Altera, and LSI Logic. To demonstrate the network in hardware, UCLA team has taped out 3 chips and successfully implemented variants of Konda network and also variants of processor-block features.

Chip 1 (90nm, LUT-slice FPGA, concept demo): A 1024-LUT FPGA was made in 90nm 9SF technology (Dec 2009 run). Our synthesis estimates predict a 250 mW of power and a 600 MHz maximum performance. The chip occupies 2.6 x 2.5 mm² in 90nm. *Status:* lab testing.

Chip 2 (65nm, *LUT and DSP slices, small scale*): A 256-LUT 240-DSP 8-BRAM FPGA was made in 65nm technology (June 2010 run). The chip is aimed to show asymmetric network and heterogeneous computing blocks. The chip occupies 2.1 x 3.1 mm² in 65nm. *Status*: taped out.

Chip 3 (45nm, DSP-slice FPGA, small scale): A 512-DSP slice FPGA is made in IBM 45 nm SOI technology (June 2010 LEAP run). We expect power consumption below 500 mW. This design will be applicable to small-scale applications such as micro UAVs. *Status:* taped out.

Nature and description of end results to be delivered to DARPA: We will provide several deliverables to DARPA and DOD community as listed below.

- Interconnect architecture and routing tools (software).
- Hardware library in 32nm IBM SOI process (compatible with Cadence software).
- Routing software for the new interconnect architecture and hardware library (software).
- Chip demos of varying scale to demonstrate algorithms of interest to DOD (hardware).
- Tool flow for mapping algorithms onto FPGA chips (software).
- Demonstrations of HPC benchmarks using commercial technology.

The first three items in the list are intermediate steps towards the final hardware demonstration that also includes user-friendly mapping tool interface.

Cost and schedule of the proposed effort: \$2,374,111 over 3 years.

2.4. Technical Approach

Problem Description: FPGAs are used in many signal processing and computing applications. DOD mission capability or computing performance can be greatly improved with more energy efficient hardware. FPGA based solutions are very attractive due to their flexibility, similar to that of CPUs. This flexibility comes at a very high energy cost, as shown in Fig. 1.

Looking at the energy efficiency (the amount of energy per unit operation) for a variety of chips from different categories, we observe a 1,000x gap in energy efficiency between microprocessors and dedicated designs. The root cause of this is architectural. Processors have general ALU-type processing unit(s) and large amounts of memory to support time-muliplexing of instructions and data into and out of the ALU(s). Dedicated chips have a variety of processing units, but are very expensive in low-volume and can't be programmed, so they can't be used for HPC applications. General-purpose DSPs are a viable compromise between microprocessors and dedicated designs. Recently, however, FPGA chips have started to gain attention with their increased computing capabilities. Look-up-table (LUT) based chips have energy efficiency similar to that of CPUs and are not very attractive alternatives to CPUs (CPUs are easier to program). Many today's FPGAs have dedicated kernels such as DSP slices, ARM cores, etc. These FPGAs have energy efficiency similar to DSP chips, but they are still 30-50x worse than dedicated chips. The root cause of energy inefficiency in these FPGAs is their interconnect architecture.

Today's FPGAs use 2D-mesh interconnect architecture shown in Fig. 5. Interconnect consists of switch boxes (shown as cross-points), connection points for the buses, and bus drivers (buffers). This architecture is not very scalable: it requires $O(N^2)$ interconnect switches for *N* LUTs. For 1k processing units, this means 1M switches! To overcome this complexity issue, designers employ heuristics to reduce the number of switches. One of the ideas is to reduce connectivity around the edges, as shown in Fig. 5. Another idea is to reduce top-level connectivity in large designs and utilize local connections. These approaches are heuristic and lead to inefficient utilization of hardware resources. Readers may be have experienced that utilizing more than 80% of FPGA resources without sacrificing performance is a big challenge in commercial FPGA systems.



Figure 5: 2D-mesh interconnect architecture. $O(N^2)$ interconnect switches are required for full connectivity. Heuristics are used to reduce the network complexity. These heuristics result in non-deterministic routing.

Even after reductions in network complexity, interconnect still occupies over 75% of area in today's FPGAs. For example, Xilinx Virtex-5 chip has 1.1B transistors; 275M are used for logic, 875M (80%) are used for interconnect. Most of FPGA power is dissipated by the interconnect, as shown in Fig. 6. Further simplifying interconnect (without sacrificing connectivity) would have multiple benefits. First, the interconnect power will decrease. Second, due to reduced interconnect area, overall chip area will also reduce. Third, since the chip area is reduced, the size of wires (and wire capacitance) also reduces. The reduction in wire length and complexity implies further reduction in power. It also implies



improvements in performance. This excess performance can be traded for increased energy efficiency, or simply used to improve computational efficiency. Finally, we benefit from reduced clock power since the clock is now distributed over a smaller area. Therefore, reduction in interconnect complexity is crucially important for improved computing power and performance.

Proposed Network Architecture: In response to the interconnect challenge, we propose to use a proprietary Konda hierarchical interconnect architecture. This interconnect architecture has greatly reduced complexity, $O(N \cdot \log_2 N)$, and it is based on fully deterministic routing. The concept of Konda network is to use simple unidirectional switches and 2x1 multiplexers to hierarchically connect the computing resources (LUTs, DSP slices, ARM IP, etc.).

Eliminating routing congestions and making the 2D circuit layout possible are the key enabling features of the Konda network. An example of N = 8 LUT design with Konda network is shown in Fig. 7. For complete routing $log_28 = 3$ levels of switch matrices are needed. First, vertical tracks connect nearest LUTs, then horizontal tracks are used to connect LUTs at the next level, and finally vertical tracks are used to connect the last level of switches. This structure has full connectivity and completely deterministic 2D routing.



Figure 7: Konda interconnect network architecture and routing tracks for N = 8 LUTs.

The benefits of this network architecture were evaluated using Toronto20 benchmarks. Toronto 20 benchmark suite originated from an FPGA place-and-route challenge that was set up by University of Toronto Researchers [1] to encourage FPGA researchers to benchmark their software design tool chains on large circuits These 20 benchmarks are from real designs and the placed netlists are provided - for a given FPGA logic block consists of a 4-input look-up table (LUT) and a flip flop - to experiment with different routing architectures and routing algorithms. The existing results are experimented with 2D-Mesh network based routing network by providing partial bandwidth i.e., with different switch-box flexibility, connection-box flexibility and a certain number of channels. Konda hierarchical network is also experimented with partial bandwidth provisioning and the results are compared on various dimensions such as 1) number of cross points, 2) route length (delay) 3) performance 4) speed of routing and 5) routability. Konda hierarchical network performed better in several easily-measureable ways and the results are presented in Tables 2 and 3.

				2D-Mesh	Network	Konda H	ierarchical	Network
Toror	nto20 Benc	hmark Info	ormation	Simu	lation		Simulation	
				(Bidirectio	onal wires)	(Unic	lirectional v	vires)
			Number of	Max	Total	Total	Savings	Cross-
Name	Size	LUTs	connections	channel	cross-	cross-	factor	points
			connections	width	points	points	Idettoi	saved
alu4	40	1600	1514	9	177,174	58,737	3.02	118,437
apex2	44	1936	1875	10	237,660	83,180	2.86	154,480
apex4	36	1296	1243	11	175,890	52,482	3.35	123,408
bigkey	54	2916	1694	6	213,876	54,643	3.91	159,233
clma	92	8464	8302	10	1,026,780	359,846	2.85	666,934
des	63	3969	1347	7	338,730	57 <i>,</i> 044	5.94	281,686
diffeq	39	1521	1497	7	131,082	49,275	2.66	81,807
dsip	54	2916	1309	5	178,230	40,972	4,35	137,258
elliptic	61	3721	3604	9	408,510	129,507	3.15	279,003
ex5p	33	1089	1019	11	148,170	44,609	3.32	103,561
ex1010	68	4624	4588	9	506,790	192,391	2.63	314,399
frisk	60	3600	3556	11 483,186		134,686	3.59	348,500
misex3	38	1444	1383	10	177,900	58 <i>,</i> 866	3.02	119,034
pdc	68	4624	4535	15	844,650	239,484	3.52	605,166
s298	44	1936	1929	6	142,596	63,956	2.23	78,640
s38417	81	6561	6349	6	478,260	207,457	2.30	270,802
s38584.1	81	6561	6291	7	557,970	184,030	3.03	373,940
seq	42	1764	1717	10	216,780	73,880	2.93	142,900
spla	61	3721	3644	12	544,680	171,676	3.17	373,004
tseng	33	1089	975	6	80,820	31,599	2.56	49,221

Table 2:	Comparison	of 2D-Mesh	and Konda	interconnect	networks	using	Toronto20	behcmarks.
						<u> </u>		

The benefits of Konda hierarchical network over 2D-Mesh network using Toronto20 Benchmarks are summarized in Table 4. Various configurations of Konda hierarchical network were tested for each benchmark and the results are verified as follows:

• All 20 benchmarks were routed by our algorithms in our network,

- Switches required to route was reduced significantly,
- Fundamental routing algorithms are proven,
- Speed of routing is proven,
- Benchmarks were profiled for Bandwidth requirements.

Table 3: Comparison of 2D-Mesh and Konda interconnect networks using Toronto20 behcmarks. In
addition to considerable savings in the number of cross-points, Konda network uses has far better
percentage utilization (fewer % is better) than the 2D-Mesh network.

Toroi Bench Inforn	nto20 nmark nation	2D-Mesh Simu (Bidirectio	Network lation onal wires)	Konda H (Unic	ierarchical Simulation lirectional v	Network vires)	Other Key the Sim	Results of ulation
Name	Size	Max Ch Width	Total Cross-pts	Total Cross-pts	Savings factor	Cross-pts saved	% Cross- pts used Konda	% Cross- pts used 2D-Mesh
alu4	40	9	177,174	58,737	3.02	118,437	7.9	66
apex2	44	10	237,660	83,180	2.86	154,480	9.3	70
apex4	36	11	175,890	52,482	3.35	123,408	9.4	60
bigkey	54	6	213,876	54,643	3.91	159,233	4.1	51
clma	92	10	1,026,780	359,846	2.85	666,934	8.1	71
des	63	7	338,730	57,044	5.94	281,686	2.9	33
diffeq	39	7	131,082	49,275	2.66	81,807	6.9	75
dsip	54	5	178,230	40,972	4,35	137,258	2.8	46
elliptic	61	9	408,510	129,507	3.15	279,003	7.0	63
ex5p	33	11	148,170	44,609	3.32	103,561	9.5	60
ex1010	68	9	506,790	192,391	2.63	314,399	8.4	75
frisc	60	11	483,186	134,686	3.59	348,500	7.5	56
misex3	38	10	177,900	58,866	3.02	119,034	8.7	66
pdc	68	15	844,650	239,484	3.52	605,166	10.5	56
s298	44	6	142,596	63,956	2.23	78,640	7.1	89
s38417	81	6	478,260	207,457	2.30	270,802	6.0	86
s38584.1	81	7	557,970	184,030	3.03	373,940	5.3	66
seq	42	10	216,780	73,880	2.93	142,900	9.0	68
spla	61	12	544,680	171,676	3.17	373,004	9.3	63
tseng	33	6	80,820	31,599	2.56	49,221	6.7	78

Table 4: Summary of the benefits of Konda hierarchical network. Analytical and empirical results are shown, the numbers are relative to 2D-Mesh network.

Criteria	Analytical	Empirical
Interconnect area	At most 1/3	At most 1/3
Connectivity	2-3x	2-3x
Interconnect Power	1/5 to 1/10	1/5 to 1/10
Interconnect Latency	1/5 to 1/10	1/5 to 1/10
Speed of compilation	Significantly faster	Significantly faster
Scalability across process generations	Close to linear	Close to linear

The conclusions of simulation of Toronto20 benchmarks using Konda hierarchical network matched the benefits derived in empirical analysis. The generic routing tool created for Konda hierarchical network delivers consistent and predictable results. Based on the Toronto20 benchmark results it can be projected that the gap between ASIC's and FPGA's can be closed as shown in Fig. 8, which would significantly improve performance and energy efficiency of HPC hardware. In the proposed work, we will explore further technology improvements.



Figure 8: Konda interconnect network architecture has substantial benefits over today's FPGAs. It is projected to have ASIC-like energy efficiency, power, and performance. Such energy-efficiency levels are more than 100x better than general purpose processors.

2.4.1. Network Architecture and Routing Tools

We will next work on homogeneous and heterogeneous networks featuring arbitrary level of connectivity. The decision about the connectivity level will be aided with feedback from the mapping tools (Task 6) in order to minimize hardware utilization.

Task 1) Routing Architectures for Homogeneous Blocks: Routing tool will be developed for the FPGA with homogeneous blocks. Routing algorithms need to be developed for uni-terminal nets and multi-terminal nets. The hierarchical routing network may be a symmetric network where the number of inputs and the number of outputs are the same. The routing network may also be asymmetric network where the number of inputs and the number of outputs are the same. The routing network may also be asymmetric network where the number of inputs and the number of outputs are not the same. Rearrangeably nonblocking and strictly nonblocking multi-terminal net algorithms will be implemented to demonstrate the routability and the speed of routing. Routing algorithms need to be implemented for configurations of Konda hierarchical network where some of the stages in the network may be partially connected and the other stages are fully connected. The LUT size of the network may be a perfect power of two or non-perfect power of two.

Task 2) Routing Architectures for Heterogeneous Blocks: We will also explore interconnect architectures suitable for heterogeneous blocks. The key architectural challenge is to adapt the Konda hierarchical network for FPGA architecture. A fully connected hierarchical network is an over-kill for FPGA applications. Our goal is to converge on the appropriate design of the routing network in three phases and also adopt it to many different applications end-user applications. Also we need to experiment with many varieties of hierarchical network designs such as Benes

network, butterfly fat-tree network and other optimizations related to properties of FPGA designs. One aspect is to analyze the locality typical in FPGA designs and optimizing or adopting Konda hierarchical network with optimum bandwidth for local connectivity and global connectivity. The typical LUT size that is known to be optimal in a 2D-Mesh routing network may not be optimal for Konda hierarchical network. This is because Konda hierarchical network provides richer connectivity with smaller switch and less number of tracks. Determining the appropriate length of the tracks is another aspect that will be addressed in this task.

2.4.2. Hardware Design

To fully demonstrate the benefits of the proposed interconnect architectures and routing tools, we will implement the network architectures on a series of chips. Hardware design tasks will concentrate on achieving two goals: 1) hardware demonstration of power, area, and performance benefits, 2) development of automated chip routers to facilitate technology transition.

Task 3) **Chip Demonstrations:** Multiple chip demonstrations are planned to further quantify the benefits of the interconnect technology, and to further optimize interconnects based on hardware experiments.

Prior Work: We have designed several chips prior to this program, as summarized in Table 5. This was a self-initiated self-supported work. The results of IBM-90 and TSMC-65 chips will be available in September 2010. The results will be made available to the OHPC community.

Chip ID	Features	Area	Power	Performance	Status
IBM-90	1k LUTs	6.5 mm ²	250 mW	300-600 MHz	Lab testing
TSMC-65	256 LUTs 240 DSPs 1 BRAM	8 mm ²	500 mW	400-700 MHz	Taped out 6/2010
IBM-45-SOI	512 DSPs	4.4 mm ²	500 mW	500-800 MHz	Taped out 6/2010

Table 5: Summary of FPGA chips built prior to the OHPC program.

The chips summarized in Table 5 are an initial demonstration of hardware feasibility of the interconnect network. The chips also demonstrate the integration of heterogeneous blocks (LUT, DSP, BRAM) for small-scale examples. Before describing the features of proposed chips, below is the description of design methodology used in prior work.

Figure 9 illustrates hierarchical design approach that starts with switch-matrix design. The switch-matrix blocks are custom-designed to allow tiling and hierarchical expansion. Design techniques used here will become cornerstones for the automation (Task 4).



Figure 9: Herearchical design procedure starting with switch-matrix design, integration of a slice, a hierarchical macro, and chip-level integration.

Page 16 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003

The IBM-90 chip illustrates LUT-only design. Consistent with predictions from Tables 2 and 3, Konda network achieves at least 3x lower interconnect area. Conventional chips have over 75% of interconnect area. In our chip, we have 50% logic (LUTs) and 50% switches (wires), which confirms the 3x interconnect area reduction estimate. Even with nearly-full connectivity, our chip has only 50% of area of the interconnect (3x less than commercial).



Figure 10: Detail of switch-matrix block (left), Energy-delay optimization of LUT macros (right).

Figure 10 shows the detail of the switch-matrix block. Local I/O connections allow tiling of layout macros, while pins in the middle are being used for hierarchical routing. Plot on the right shows energy-delay optimization after gate sizing and supply voltage tuning. We designed for 0.9 V supply (corresponding to the nominal/slow corner). The design is based on the sensitivity optimization methodology [6] that balances impact of all tuning variables. At a solution point, sensitivities to all variables are equal. In the energy-delay space, this means that the energy-delay lines obtained by tuning individual variables around a design point should be tangent. This is shown in the final design, where the V_{DD} and sizing (W) lines have similar slopes at $V_{DD} = 0.9$ V. With these optimizations being made at the circuit level, we ensure that power efficiency considerations are propagated from system level down to the technology level. Our 1k FPGA is estimated to consume total of 250 mW of power when fully utilized. The energy efficiency per CLB slice is 0.96 pJ at 0.9V. We also performed deep pipelining to maximize performance. Synthesis estimates show a 600 MHz performance. This performance is significantly better than 450 MHz achieved by Xilinx Virtex-5 parts (Virtex-5 is built in a faster 65 nm technology).



Figure 11: Switch-point in 2D-Mesh network highlighting bidirectional nets (left), Switch-matrix in Konda network (right) highlighting unidirectional nets.

A very important feature of Konda interconnect architecture is that it uses only unidirectional wires, as shown in Fig. 11. The 2D-Mesh architecture uses bidirectional nets as shown on the left. Going from bidirectional to unidirectional nets results in a lower switching net capacitance. Implementation of the Konda switch-matrix is done with simple 2x1 muxes.

After demonstrating the feasibility of Konda network architecture on chip, the next bit challenge is to support the integration of heterogeneous blocks (LUTs, DSPs, BRAMs, etc.). Also, one should explore irregular switch-matrix architecture to reduce top-level wiring. An irregular switch-matrix design shown in Fig. 12 is implemented on the TSMC-65 chip (Table 5).



Figure 12: Irregular switch-matrix architecture to reduce top-level wiring (left), Chip demonstration of LUT, DSP, and BRAM modules using the irregular SM architecture (right).

The idea implemented in the network architecture from Fig. 12 is to use full connectivity only near the center of the chip. In our previous FPGA designs, long wires are routed across the entire chip to connect the bordering switch matrices at the topmost level. The drawback of these long wires is the requirement of many buffer insertions, consuming excessive power and routing area. The irregular switch matrix architecture reduces the number of top-level buffers by 95% since the bordering switch matrices now route through the center switch matrices to connect to the other side.

Proposed Work: The chips summarized in Table 5 are just an initial effort towards optimized FPGA implementation. This proposal will focus on hardware designs with reduced-complexity irregular network architectures developed in Tasks 1 and 2. Together with mapping tools from Tasks 5 and 6, we will be able to minimize level of connectivity required for chip-level routing. This will be demonstrated in actual hardware designs as explained below.

We plan to make use of the DARPA LEAP program for chip fabrication. We will demonstrate medium-scale designs for which regularity of layout cells, in addition to architecture regularity, will play a key role in improving tolerance to manufacturing variations. We will thus explore designs with regular layout geometries in IBM's 32nm SOI process. The evaluation of circuit metrics as compared to standard-cell based CMOS design will result in a library of FPGA building blocks which will be used for chip demonstrations. In addition to the IBM library cores and routing tools (Task 4), we will also consider TSMC libraries due to their general use and to provide additional options for technology transition.

In CMOS run A (see Sec 2.8.1), we will test the use of library IP for the integration of mediumscale FPGA processor (5K LUTs). Since 32nm SOI process is new and not yet fully tested by designers, we will work with homogeneous LUT-only design to minimize the risk of potential manufacturing and design issues. This chip is a 10x larger than the IBM-90 and will allow us to confidently explore mapping algorithms for this level of design complexity.

In CMOS run B (see Sec 2.8.1), we will design a chip with 15K DSP slices. The chip will be compliant with FMC expansion modules (160 I/O pins). This design will be able to support DSP-centric applications such as signal and information processing. The chip will be tested using BEE4 module (coming out in Fall 2010) from BEEcube. Such setup will allow us to do side-by-side comparison with Virtex-6 Xilinx FPGAs. We will work with BEEcube on HPC application benchmarking and will also welcome inputs from the DOD community.

In CMOS run C (see Sec. 2.8.1), we will demonstrate LUT/DSP/BRAM based design with over 15K LUTs, over 15K DSP slices, and adequate BRAM memory. The chip will be also compliant with FMC for testing with the BEE4 module. Chips B and C will make use of irregular network architecture and optimized connectivity as described in Task 4. The chips from CMOS run C will be used for inter-module communication with multiple BEE4 boards to show expandability to large HPC benchmarks.

Task 4) Automated Chip Routing Tools: To facilitate the integration of medium- and largescale FPGA chips, and to enhance our technology transition capabilities, we will work on automated chip routing tools. The tools are intended to automate custom design strategies developed in our prior work. We will also automate design techniques further developed under Task 3, particularly CMOS runs A and B (see the scheduling chart in Sec. 2.8.1).

Advanced routing tools will need a library of switch-matrix blocks with varying degree of connectivity. Figure 13 shows example of full- and half-connectivity cells as well as full-to-half connectivity interface cells. The use of these simple cells, and others, will enable us to support network routers (developed in Tasks 1 and 2) with arbitrary level of connectivity.



Figure 13: Switch-matrix (SM) blocks include full-connectivity (left), full-to-half-connectivity (middle) and half-connectivity (right) features.

An example of cell design for future automated routing is shown in Fig. 14. In the chip shown in Fig. 12, DSP slices have to be designed with fixed width due to size constraints from configuration SRAM blocks. In our architecture, we use wordline (WL) to drive SRAM modules on both sides (left and right of the WL circuits). WL routing is done in metal 3 (M3) as shown in Fig. 14. We must allow M3 tracks for neighboring SM. The routing channel for 7 bits of SRAM



Figure 14: Switch matrix design showing detail of SRAM routing tracks. Upper SM (right) is a mirrored-version of the lower SM (left). Alternating 4/3 bits per row and custom muxes are used to facilitate 7-bit routing of SRAM configuration bits.

is made using alternating 4/3-bit horizontal tracks. We also make use of custom muxes to reduce redundant input inverters (details not shown on the figure). These concepts will be automated.

Automation of other routing tasks, in addition to the one illustrated in Fig. 14, will be performed. The outcome of Task 4 will be the router that can take arbitrary number of LUT, DSP, and BRAM cells and, for a given level of connectivity, provide routed chip that implements optimal network architecture developed in Task 2. This kind of routing capability will allow us to customize chip features and rapidly construct energy-efficient FPGAs for HPC applications (analogous to different Xilinx chip families, for example). The automated routing will also provide chip design community a tool for the utilization of our library macros. We will maintain library of macros in IBM 32nm SOI technology (for DOD community) and also TSMC 32/28nm technology (for DOD and commercial use).

2.4.3. Hardware Mapping

For an FPGA to be effectively used by its consumers, an automated mapping tool must be provided as well. Mapping tool for commercial FPGAs are provided to convert user-provided Verilog or VHDL into a gate-level design, and automatically place-and-route these gates onto the FPGA. As a result, the user has complete automation from Verilog/VHDL to a functional FPGA. The mapping and place-and-route software is a crucial component of this project, and major efforts ought to be allocated to provide an efficient tool.

Gate-level Synthesis: The first step of the mapping process is to create a mappable design from the Verilog or VHDL input. The process is called logic synthesis, an intricate procedure requiring complex algorithms.

To optimize our resource usage, we plan to adopt commercial synthesis tools such as Synopsys® Design Compiler or Cadence® RTL Compiler. Both these tools operates on a standard cell library that contains information regarding the timing and functional characteristics of each logic gate. The tool then converts the input design into a netlist consisting of gates from the standard cell library.

The main task here is to create a standard cell library mappable to our custom FPGAs. If the FPGA is constructed from 4-input LUTs, then the standard cell library ought to include different combinations of 2, 3, and 4-input logic gates. The FPGA mapper can then determine the appropriate values to program to each LUT based on the logic gate.

Netlist Optimization: Although the synthesized netlist is fully functional, it may not be optimal for our FPGA applications. The logic netlist should therefore be optimized by the mapper for area reduction and speed improvements. This is the second step of the mapping process.

In modern FPGAs, the majority of the area and delay are attributed to interconnects between logic gates. Therefore it is beneficial to maximize the logic function of each LUT instead of spreading the same function over many LUTs. To achieve such optimality, logic gates with less than 4 inputs are searched for logic recombination with its neighboring gates:



Figure 15: Illustration of gate-level synthesis.

Shown are two 2-input gates drive another 2-input gate. These 3 gates can be combined to a 4-input gate to fit into a single LUT instead of spreading over 3 LUTs, wasting both logic and routing resources. More strictly speaking, any sequential logic gates with a total unique input of 4 or less can be combined into a single 4-input LUT.

In our FPGAs, a 4-input LUT can be reprogrammed as two 3-input LUTs, where 2 of the inputs are shared. Two 3-input gates can potentially share a single LUT as long as the number of unique inputs is 4 or less. Two 2-input gates can always share a LUT as well. The mapper tool can exploit such feature during the placement process, as mentioned later.

Task-5) Place & Route Algorithms and Tools: Once the netlist has been optimized, place and route can begin. Each logic gate ought to be placed before it can be routed. Finding a suitable placement for each logic gate can greatly affect the performance of the final design, and poses a significant challenge to the tool.

The placement process is divided into coarse placement, which determines the gate partition, and fine placement, which determines the exact gate location in each partition. Gate partition takes place first, and the goal is to partition the gates into quadrants to minimize cross-partition

interconnects. This process can be modeled by an optimization problem, where the cost is total number of wires crossing horizontal, vertical, and diagonal boundaries. A penalty cost can be added to ensure even distribution of gates across partitions.

In most convex optimization problems, only the local minima can be determined based on the initial condition. Since an optimal initial condition cannot be determined, and the cost function may contain numerous local minima, a simulated-annealing based algorithm is used for gate partition. Based on the size of the design, numerous locally-optimal solutions can be found, and the lowest-cost solution is chosen.



Figure 16: Mapper for hierarchical-interconnect FPGAs.

Once the gate partition is determined, first level of fine placement can proceed. The longest interconnects are those requiring diagonal connections across the area. These gates are placed near the center of the area, where the diagonal connections are the shortest. Gates with vertical and horizontal connections are then placed near their respective edges to minimize their wire connections.

Hardware *Routing:* Since the routing resources in FPGA are limited, routing should occur at the same time as placement. This prevents a gate from being placed at a nonroutable location. When a gate is being considered for a location, all of its inputs and outputs should be located. The input and output gates that are already placed must be able to route to this location, else a new location must be determined. In cases where more than one possible routes exists, each routing candidate should be evaluated for interconnect length, and the shortest interconnect is chosen.



Figure 17: Recursive execution of sub-level PnR.

Page 22 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003

Automated Place and Route: The place-and-route tool can utilize recursive placement. Since all the partition-crossing gates are placed, all the non-placed gates in each partition are local to that partition, meaning they do not connect to any gates outside the partition. For each quadrant, the aforementioned place and route occurs again to place gates within the subquadrants, until all the quadrants (and subquadrants) are processed.

In some cases, a gate may not be placed inside the chosen partition, then other partitions at the level are considered for placement. If none of the partitions can accept the gate to be placed, a higher hierarchy is searched for placement.

Task 6) Tools for Hardware Mapping: The final step of the place-and-route process is the output the design. This step creates the exact bit sequence required to program the scan chain on the FPGA, which configures all the LUTs and interconnects to create the programmed function.

For each of our FPGA, we have knowledge of the exact bit location for each of the LUT configurations, as well as the switch-matrix configurations. For the logic block shown below, four LUTs are programmed as one configurable logic block. The scan-chain (SI) first controls internal configurations (such as 4-input/3-input mode, carry-chain propagation, register output, etc), and then controls each of the four LUT values. The corresponding configurations from the place-and-route output are then mapped to these bits on the scan-chain. The switch matrix bistream is determined in the similar fashion since all the interconnect configurations are already known from place-and-route.

Task 7) Demonstrations and Technology Transition: We will use chips from CMOS runs B and C (see Sec 2.8.1) to demonstrate the benefits of our hardware as compared to Xilinx Virtex-6 chips by using the BEE4 module from BEEcube.

Collaboration with BEEcube: We will work with BEEcube (support letter attached at the end of Volume I) to do technology demonstration and initial transition to HPC applications. We will make use of future BEE4 platform consisting of 4 Virtex-6 FPGA chips (LX240 family) and featuring FMC interface (160 pins/chip). A custom PCB board with our FPGA will comply to the FMC interface specifications. BEE4 FPGA chips will be used to execute computations on our custom chips, as shown in Fig. 19. This setup will also allow side-by-side benchmarking of Virtex-6 FPGAs and our chips.



Figure 18: Configuration bits for a Slice-L (4 LUTs) block.



Figure 19: Technology demonstration and initial transition using BEEcube HPC technology.

Page 23 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003

To further demonstrate inter-module communication and scalability to larger systems, we will connect 4 BEE4/Chip modules, as shown in Fig. 19. We will use BEEcube HPC benchmarks for initial evaluations. We welcome inputs from DOD community and other teams in the OHPC program about example algorithms.

In addition to BEEcube benchmarks, we will explore parallel execution of neural spike sorting algorithms from UCLA Medical School, Department of Neurology. We have data from our scientific collaborators (Prof. Rick Staba, and Prof. Chris Giza) who monitor neural action potentials in human patients with acute epilepsy. Data recording (64 channels, 10 bits, 28 kS/s) over 3 weeks aggregates 2 TB of data per patient and presents an extreme signal processing challenge. We have tried to process one hour of data (60 GB) using a CPU cluster with 40 3 GHz processors. Although we sample at 30 kHz, the total run-time was 68 minutes due to sequential nature of CPU processing. When the same algorithm was mapped to an FPGA running with a 300 MHz clock, we estimated processing time to be 0.4 seconds. Since we can execute complete algorithm iteration in one clock cycle, we get a 10,000x improvement in execution (300 MHz / 30 kHz). The bottleneck is how fast we can feed the data into the FPGA, not the speed of the FPGA itself. We would like to collaborate with other teams in the OHCP program who are exploring storage bandwidth issues.

Future Implications: Upon successful demonstration with BEEcube, the technology can be further scaled up to a rack system as shown in Fig. 20. A number of FPGA nodes will be used to execute common operations. Optionally, one could have a small number of general-purpose CPU blades for non-standard operations. This configuration will require software development to abstract away hardware details from the user and can be a topic addressed by other OHPC teams. Finally, Konda network can also be applied for efficient rack-to-rack connectivity to produce ultralow-power and ultra-highperformance HPC systems for extreme computations.



Figure 20: Future possibilities with our technology: FPGAs can be used in rack-scale systems (left), .Konda network technology can also provide superior rack-to-rack connectivity for server farms (right).

2.5. Statement of Work (SOW)

We propose to develop energy-efficient programmable hardware and supporting mapping tools. The hardware is based on proprietary Konda interconnect architecture that provides significant reduction in interconnect complexity as compared to existing FPGAs. The new architecture and supporting tools are projected to provide over a 15x improvement in energy efficiency while also considerably reducing chip area and improving performance.

Task 1: Architectures for Homogeneous Blocks (Lead: Konda) \$200K, Q1-Q3

Objective: Define interconnect architecture and routing tools for designs with a given number of homogeneous blocks such as LUT or DSP slices.

Approach: Rearrangeably nonblocking and strictly nonblocking multi-terminal net algorithms will be implemented to demonstrate the routability and the speed of routing. Routing algorithms need to be implemented for configurations of Konda hierarchical network where some of the stages in the network may be partially connected and the other stages are fully connected. The LUT size of the network may be a perfect power of two or non-perfect power of two.

Exit criteria: When the chosen interconnect architecture demonstrates the optimal tradeoff between interconnect size and performance (as quantified by Toronto20 benchmarks).

Deliverables: Architectural diagram of the interconnect structure (block-level schematic).

Task 2: Architectures for Heterogeneous Blocks (Lead: Konda) \$200K, Q3-Q6

Objective: Define interconnect architecture and routing tools for designs with a given number of heterogeneous blocks such as a combination of LUT, DSP slices, memory, and IP elements.

Approach: We will make use of the infrastructure developed for Task 1 and customize interconnects local to each type of heterogeneous block.

Exit criteria: Interconnect architecture with optimal size-performance tradeoff as quantified by Toronto20 benchmarks.

Deliverables: Architectural diagram of the interconnect structure (block-level schematic).

Task 3: Chip Demonstrations (Lead: *Markovic*) \$900K, Q1-Q11

Objective: To demonstrate the feasibility of the interconnect architecture on different design complexity and compare performance and power with commercial FPGAs.

Approach: We will design and verify three FPGA chips with increasing level of complexity (5K, 15K, 45K LUTs). The chips will be constructed from custom-designed macros, including processing, memory, and interconnect blocks.

Exit criteria: Hardware demonstration of superior performance (>2x), energy (15x), and area metrics (>2x) as compared to commercial FPGAs.

Deliverables: Library of FPGA macros, chip demonstration results.

Task 4: Automated Chip Routing Tools (Lead: Markovic) \$100K Q4-Q5

Objective: To automatically place-and-route an FPGA according to hardware requirements.

Approach: Library of FPGA macros for the technology of interest, automatic Verilog generation, supporting scripts for chip synthesis.

Exit criteria: Design and layout of an FPGA using the automated toolflow.

Deliverables: Scripts for synthesis and tools for automatic Verilog generation (that use the library of FPGA macros delivered in Task 3).

Task 5: Place and Route Algorithms and Tools (Lead: Markovic) \$300K, Q1-Q6

Objective: Determine the optimal physical location and routing for each LUT / macro to maximize resource utilization and chip performance.

Approach: Partition LUT / macro blocks into sets with coarse and fine levels of granularity. Place and route gates for minimum interconnect delay. The algorithm repeats hierarchically until all levels of interconnect are placed and routed.

Exit criteria: Successful place-and-route of Chip A (5K LUTs) to its full interconnect utilization. *Deliverables:* Software demonstration of an automated place-and-route flow.

Task 6: Tools for Hardware Mapping (Lead: Markovic) \$100K, Q3-Q4

Objective: To map the place-and-routed design to a bitstream format for FPGA programming.

Approach: Creates the exact bit sequence based on the scan chain configuration implemented on chip. This configures all the LUTs and interconnects to execute the programmed function.

Exit criteria: Successful programming of scan chain with demonstrated functionality.

Deliverables: An automated tool for bitstream programming.

Task 7: Demonstrations and Technology Transition (Lead: *Markovic*) \$500K, Q7-Q12

Objective: Demonstrate the benefits of our technology for HPC applications.

Approach: Use chips from CMOS runs B and C with the BEE4 platform (consisting of four Virtex-6 FPGA chips) to perform HPC benchmarking. The BEE4 FPGA chips will be used to execute computations on our custom chips and perform side-by-side comparison of Virtex-6 FPGAs against our chips.

Exit criteria: Functional HPC platform based on BEE4 and custom FPGA chips.

Deliverables: Results from HPC benchmarking, hardware demonstration of performance and energy efficiency on BEE4-based platform.

Task / Cost	Year 1	Year 2	Year 3	Total
Task 1: Homogeneous Architecture	\$200K	-	-	\$200K
Task 2: Heterogeneous Architecture	-	\$200K	-	\$200K
Task 3: Chip Demonstrations	\$350K	\$350K	\$200K	\$900K
Task 4: Automated Chip Routing	\$50K	\$50K	-	\$100K
Task 5: Place-and-Route Algorithms and Tools	\$200K	\$100K	-	\$300K
Task 6: Tools for Hardware Mapping	\$100K	-	-	\$100K
Task 7: Demonstrations and Technology Transition	-	\$100K	\$400K	\$500K

Tuble 0. Tuble Cost and Deficulte

2.6. Intellectual Property

Venkat Konda has filed 10 patent applications and assigned them to Konda Technologies Inc. More patent applications are in the pipeline. The following is the complete list of patent applications:

- [1] Large Scale Crosspoint Reduction with Nonblocking Unicast & Multicast in Arbitrarily Large Multi-stage Networks
 - US Provisional Patent Application Number: 60/905,526
 - Date Filed: March 6, 2007
- [2] Fully Connected Generalized Multi-stage Networks
 - US Provisional Patent Application Number: 60/940, 383
 - Date Filed: May 25, 2007
 - [2a] Fully Connected Generalized Multi-stage Networks
 - PCT Patent Application Serial Number: PCT/US08/56064
 - Date Filed: March 6, 2008
 - [2b] Fully Connected Generalized Multi-stage Networks
 - US Patent Application Serial Number: 12/530,207
 - Date Filed: September 6, 2009
- [3] Fully Connected Generalized Butterfly Fat Tree Networks
 - US Provisional Patent Application Number: 60/940, 387
 - Date Filed: May 25, 2007
- [4] Fully Connected Generalized Multi-Link Butterfly Fat Tree Networks
 - US Provisional Patent Application Number: 60/940, 390
 - Date Filed: May 25, 2007
 - [4a] Fully Connected Generalized Butterfly Fat Tree Networks
 - PCT Patent Application Number: PCT/US08/64603
 - Date Filed: May 22, 2008
 - [4b] Fully Connected Generalized Butterfly Fat Tree Networks
 - US Patent Application Number: 12/601,273
 - Date Filed: November 22, 2009
- [5] Fully Connected Generalized Rearrangeably Nonblocking Multi-Link Multi-stage Networks
 - US Provisional Patent Application Number: 60/940, 389
 - Date Filed: May 25, 2007
- [6] Fully Connected Generalized Strictly Nonblocking Multi-Link Multi-stage Networks
 - US Provisional Patent Application Number: 60/940, 392
 - Date Filed: May 25, 2007

- [7] Fully Connected Generalized Folded Multi-stage Networks
 - US Provisional Patent Application Number: 60/940, 391
 - Date Filed: May 25, 2007
 - [7a] Fully Connected Generalized Multi-link Multi-stage Networks
 - PCT Patent Application Serial Number: PCT/US08/64604
 - Date Filed: May 22, 2008
 - [7b] Fully Connected Generalized Multi-link Multi-stage Networks
 - US Patent Application Serial Number: 12/601,274
 - Date Filed: November 22, 2009
- [8] VLSI Layouts of Fully Connected Generalized Networks
 - US Provisional Patent Application Number: 60/940, 394
 - Date Filed: May 25, 2007
 - [8a] VLSI Layouts of Fully Connected Generalized Networks
 - PCT Patent Application Number: PCT/US08/64605
 - Date Filed: May 22, 2008
 - [8b] VLSI Layouts of Fully Connected Generalized Networks
 - US Patent Application Number: 12/601,275
 - Date Filed: November 22, 2009
- [9] VLSI Layouts of Fully Connected Generalized Networks with Locality Exploitation
 - US Provisional Patent Application Number: 61/252, 603
 - Date Filed: October 16, 2009
- [10] VLSI Layouts of Fully Connected Generalized and Pyramid Networks
 - US Provisional Patent Application Number: 61/252, 609
 - Date Filed: October 16, 2009

2.7. Management Plan

Program management plan is shown in the figure below.



UCLA and KondaTech will closely work together to provide demonstrations and technology transfer to the DoD community. Below is a description of various tasks and their interaction.

Routing Architectures: KondaTech will be responsible for the development of interconnect architectures and supporting routing tools. This includes both homogeneous-block (Task 1) and heterogeneous-blocks (Task 2) architectures. UCLA will provide information about building blocks (e.g., LUT, DSP, memory) for Task 2. The developed architectures and routing tools will be benchmarked using standard Toronto20 FPGA benchmarks.

UCLA will be responsible for hardware design and hardware mapping efforts.

Hardware Design: Chip demonstrations (Task 3) will be initially made using theoretical routing architecture concepts developed at KondaTech. In this phase, we will investigate types of chip routing procedures that need to be automated during chip design. Custom routing of low-level interconnects will be enabled by the regularity of the interconnect architecture. KondaTech will provide input into automated chip routers (Task 4) to ensure that routing tools are properly transferred to chip design. We will then make use of the chip routers for the final chip design.

Hardware Mapping: UCLA will lead the hardware mapping effort, which goes in conjunction with chip design. Here, we will focus on developing algorithms (Task 5) that would optimally map algorithms to the newly developed interconnect architectures. We also have the capability to do wordlength optimization and high-level architecture transformations prior to mapping. The algorithm mapping will attempt to minimize resource utilization and power, and also maximize performance. The details of the mapping algorithms will be abstracted away from the user by developing mapping tools (Task 7) to provide automated algorithm mapping onto hardware.

Demonstrations and Technology Transition: We will demonstrate the execution of complete algorithms in order to validate power and performance gains of our technology. We will actively solicit and welcome inputs from the DOD community about the algorithmic examples that would best serve the demonstration of hardware and mapping tools.

Demonstration Platform: We will work with **BEEcube** to execute initial technology transition plan. BEEcube is now a well recognized provider of high-performance processing technology. The technology is based on FPGA hardware, a library of IP cores, and software development tools for high-performance computing and other applications. We will use BEEcube technology for our chip demonstrations. In particular, we will make use of the FMC expansion modules on their hardware platforms to control our chips. This includes programming of the chips and program execution. The 160-pin connection slots based on VITA Standard 57.1 provide BEE-to-chip interface. With this setup, we will be able to boost the performance of the existing hardware and quantify the benefits of our technology in actual computing environment. We will use four hardware units from BEEcube and setup inter-module communication in order to demonstrate scalability of our design (as described in Section 2.4).

Team size and composition:

UCLA team:

Dejan Markovic, PI

Four full-time graduate students to work on hardware design (Tasks 3 and 4), hardware mapping (Tasks 5 and 6), and technology demonstrations. They will also collaborate with Dr. Konda on Task 2. Each task will require the effort of two full-time students.

KondaTech: Venkat Konda, Consultant

UCLA and KondaTech will work closely to maximize the impact of our technology.

2.8. Schedule and Milestones

Project schedule, task description, and management plan are described in this section.

2.8.1. Schedule Graphic



Key deliverables from the program will include (due dates are referenced to the program start):

- (1) Layout library of FPGA building blocks in IBM's 32nm SOI technology (available through the DARPA LEAP program). *Due: after 4 months.*
- (2) Routing architectures for homogeneous blocks (e.g. LUTs) with varying degree of connectivity. Deliverable: software benchmarking results. *Due: after 8 months.*
- (3) Routing architectures for heterogeneous blocks (LUTs, DSP slices, BRAMs, other IP) with varying degree of connectivity. Deliverable: software benchmarking results. *Due: after 17 months.*
- (4) Test results from CMOS run A to demonstrate small-scale integration (< 5k LUTs). Deliverable: hardware measurements. *Due: after 15 months*.
- (5) Test results from CMOS run B to demonstrate medium-scale integration (< 10k LUTs). Deliverable: hardware measurements. *Due: after 26 months*.
- (6) Test results from CMOS run C to demonstrate large-scale integration (< 20k LUTs). Deliverable: hardware measurements. *Due: after 32 months*.
- (7) Tools for generating a bitstream for a mapped design. *Due: after 12 months.*
- (8) Tools for performing place-and-route of the optimized netlist for FPGA programming. *Due: after 15 months.*
- (9) Tools for integrating place-and-route tool with existing testing solutions. *Due: after 24 months.*
- (10) Hardware and tool flow demonstrations for relevant DOD algorithms; commercialization of technology. *Due: after 36 months.*

2.8.2. Detailed Task Description

Part I: Network Architectures and Routing Tools

We will next work on homogeneous and heterogeneous networks featuring arbitrary level of connectivity. The decision about the connectivity level will be aided with feedback from the mapping tools (Task 6) in order to minimize hardware utilization.

Task 1: Routing Architectures for Homogeneous Blocks

Routing tool will be developed for the FPGA with homogeneous blocks. Routing algorithms need to be developed for uni-terminal nets and multi-terminal nets. The hierarchical routing network may be a symmetric network where the number of inputs and the number of outputs are the same. The routing network may also be asymmetric network where the number of inputs and the number of outputs are not the same. Rearrangeably nonblocking and strictly nonblocking multi-terminal net algorithms will be implemented to demonstrate the routability and the speed of routing. Routing algorithms need to be implemented for configurations of Konda hierarchical network where some of the stages in the network may be partially connected and the other stages are fully connected. The LUT size of the network may be a perfect power of two or non-perfect power of two.

Task 2: Routing Architectures for Heterogeneous Blocks

The key architectural challenge is to adapt the Konda hierarchical network for FPGA architecture. A fully connected hierarchical network is an over-kill for FPGA applications. Our goal is to converge on the appropriate design of the routing network in three phases and also adopt it to many different applications end-user applications. We will experiment with many varieties of hierarchical network designs. One aspect is to analyze the locality typical in FPGA designs and optimizing or adopting Konda hierarchical network with optimum bandwidth for local connectivity and global connectivity. The typical LUT size that is known to be optimal in a 2D-Mesh routing network may not be optimal for Konda hierarchical network. This is because Konda hierarchical network provides richer connectivity with smaller switch and less number of tracks. Determining the appropriate length of the tracks is another aspect that will be addressed.

Part II: Hardware Design

To fully demonstrate the benefits of the proposed interconnect architectures and routing tools, we will implement the network architectures on a series of chips. Hardware design tasks will concentrate on achieving two goals: 1) hardware demonstration of power, area, and performance benefits, 2) development of automated chip routers to facilitate technology transition.

Task 3: Chip Demonstrations

In CMOS run A, we will test the use of library IP for the integration of medium-scale FPGA processor (5K LUTs). This chip is 10x larger than the IBM-90 and will allow us to confidently explore mapping algorithms for this level of design complexity.

In CMOS run B (see Sec 2.8.1), we will design a chip with 15K DSP slices. The chip will be tested using BEE4 module (coming out in Fall 2010) from BEEcube. Such setup will allow us to do side-by-side comparison with Virtex-6 Xilinx FPGAs. We will work with BEEcube on HPC application benchmarking and will also welcome inputs from the DOD community.

In CMOS run C (see Sec. 2.8.1), we will demonstrate LUT/DSP/BRAM based design with over 15K LUTs, over 15K DSP slices, and adequate BRAM memory. The chips from CMOS run C will be also used for inter-module communication with multiple BEE4 boards to show expandability to large HPC benchmarks.

Task 4: Automated Chip Routing Tools

To facilitate the integration of medium- and large-scale FPGA chips, and to enhance our technology transition capabilities, we will work on automated chip routing tools. The tools are intended to automate custom design strategies developed in our prior work. We will also automate design techniques further developed under Task 3, particularly CMOS runs A and B.

Part III: Hardware Mapping

For an FPGA to be effectively used by its consumers, an automated mapping tool must be provided as well. The mapping and place-and-route software is a crucial component of this project, and major efforts ought to be allocated to provide an efficient tool.

Task 5: Place and Route Algorithms and Tools

Finding a suitable placement for each logic gate can greatly affect the performance of the final design, and poses a significant challenge to the tool. The placement process is divided into coarse placement, which determines the gate partition, and fine placement, which determines the exact gate location in each partition. Gate partition takes place first, and the goal is to partition the gates into quadrants to minimize cross-partition interconnects. This process can be modeled by an optimization problem.

Since the routing resources in FPGA are limited, routing should occur at the same time as placement. In cases where more than one possible routes exists, each routing candidate should be evaluated for interconnect length, and the shortest interconnect is chosen. We will hierarchically extend the partition and place-and-route within an automated flow.

Task 6: Tools for Hardware Mapping

The final step of the place-and-route process is the output the design. This step creates the exact bit sequence required to program the scan chain on the FPGA, which configures all the LUTs and interconnects to execute the programmed function. We will make the mapping tool compliant with existing FPGA design environments such as Xilinx XSG/EDK toolset.

Task 7: Demonstrations and Technology Transfer

We will use chips from CMOS runs B and C to demonstrate the benefits of our hardware as compared to Xilinx Virtex-6 chips by using the BEE4 module from BEEcube as a test platform for HPC benchmarking. A custom PCB board with our FPGA will comply to the FMC interface specifications. BEE4 FPGA chips will be used to execute computations on our custom chips. This setup will also allow side-by-side benchmarking of Virtex-6 FPGAs and our chips. To further demonstrate inter-module communication and scalability to larger systems, we will connect 4 BEE4/Chip modules in an evaluation platform.

2.8.3. Project Management and Interaction Plan

UCLA and KondaTech will closely interact during the program. Our interaction plan consists of several means of communication:

- We will conduct weekly teleconference meetings using desktop sharing software
- Due to geographic proximity, Dr. Konda will visit UCLA once a month to meet with UCLA team and conduct detailed discussions about the project
- The PI will additionally interact with Dr. Konda regarding project management

Project management is illustrated in Section 2.7 and also summarized in Table 7.

Person / Task	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
D. Markovic		+	+	+	+	+	+
V. Konda	+	+		+		+	+

KondaTech (V. Konda) will be responsible for the development of interconnect architectures (Tasks 1 and 2). UCLA will provide information about building blocks (e.g., LUT, DSP, memory) for Task 2 and assist in verification of interconnect architectures.

UCLA (D. Markovic) will be responsible for hardware design and hardware mapping efforts (Tasks 3 to 6). KondaTech will assist in automating chip routers (Task 4) to ensure integration of the interconnect architectures from Tasks 1 and 2. The expertise of KondaTech will also be used to transition mapping software to commercial tool environments.

UCLA and KondaTech will work together on HPC demonstrations described in Section 2.4.

2.9. Personnel, Qualifications, and Commitments

Dejan Markovic is an Assistant Professor of Electrical Engineering at the University of California, Los Angeles. He completed the Ph.D. degree in 2006 at the University of California, Berkeley. His current research is focused on integrated circuits for emerging radio and healthcare systems, design with post-CMOS devices, optimization methods and CAD flows.

Prof. Markovic's research accomplishments include sensitivity-based circuit optimization [6] and DSP architecture optimization [11] for reduced power and area. As a demonstration of these concepts, his group has designed a number of complex digital chips for parallel data processing. Representative chips shown in Fig. 21 show performance range by 5 orders of magnitude (kHz to multi-GHz) and power density range of 3 orders of magnitude (μ W/mm² to mW/mm²).



Figure 21: Sample chips designed by PI's group showing broad range in performance and ultra-low power density.

The PI's research in low power has been recognized by multiple awards:

- **2007 David J. Sakrison Memorial Prize** (Best Ph.D. Dissertation at UC Berkeley), awarded to the PI for his contributions to low-power digital circuit design.
- **2008 Outstanding MS Thesis Award**, UCLA EE Dept, received by R. Nanda, for her M.S. Thesis titled: "DSP Architecture Optimization in Matlab/Simulink Environment," June 2008.
- **2010 Outstanding MS Thesis Award**, UCLA EE Dept, received by V. Karkare, for his M.S. Thesis titled: "A 130 uW, 64-Channel, Spike-Sorting DSP Chip," Dec. 2009.
- **2010 DAC/ISSCC Student Design Contest Winner**, awarded to Chia-Hsiang Yang for his paper titled "A 2.89mW 50GOPS 16x16 16-Core MIMO Sphere Decoder."

PI's group has unparalleled infrastructure for the design and optimization of digital chips, based on a number of tools developed and maintained over the past decade. PI's key publications in the area of low-power design are provided in [2-16].

Venkat Konda is an inventor, experienced entrepreneur and the CEO of Konda Technologies which he founded in 2007 based on a breakthrough layout using only horizontal and vertical tracks for Benes/BFT hierarchical networks, seminal rearrangeably and strictly non-blocking multicast routing algorithms with an architecture optimum with switch cost, power and performance. Venkat is currently in the process of commercializing the IP in FPGA interconnects, System-on-Chip interconnects and warehouse-scale datacenter switches. Prior to it, Venkat invented seminal algorithms for rearrangeably and strictly non-blocking multicast routing algorithms for Clos Networks and founded a startup Teak Networks, to commercialize into packet switch fabrics which are also applicable to design cheaper optical cross connects. Venkat received PhD degree in Computer Science & Engineering from the University of Lousiville, KY in 1992, and M.S in Electrical Engineering from the Indian Institute of Technology, Kharagpur in 1988.

Key Individual	Project	Pending/Current	2010	2011	2012
Dejan Markovic	HEALICs	Current (Co-PI)	176 hours	176 hours	176 hours
	STT-RAM	Current (Co-PI)	176 hours	176 hours	176 hours
	NEMS	Current (Co-PI)	176 hours	176 hours	176 hours
	NSF CAREER	Current (PI)	88 hours	88 hours	88 hours
	C2S2	Current (PI)	88 hours	88 hours	88 hours
	NVL	Pending (Co-PI)	176 hours	176 hours	176 hours
	FPGA	Proposed	176 hours	176 hours	176 hours
Venkat Konda	FPGA	Proposed	1000	1000	1000
			hours	hours	hours

Table of individual time commitments is provided below:

Note: Dejan Markovic is a Co-PI on three DARPA projects, two of which can directly benefit to the OHPC program. Namely, STT-RAM can be used for the realization of FPGA memory blocks. Also, zero-leakage NEM relay switches can be used for effective realization of FPGA switch-matrix blocks. PI's commitment to the proposed work is already evidenced by self-initiated and self-supported work described in Sec 2.4.1. The OHPC program will, therefore, not add a significant workload to the PI.

2.10. Organizational Conflict of Interest Affirmations and Disclosure

2.11. Human Use

2.12. Animal Use

2.13. Statement of Unique Capability Provided by Government or Government-funded Team Member

Not applicable.

2.14. Government or Government-funded Team Member Eligibility

2.15. Facilities

The description of UCLA and KondaTech facilities is provided below.

Dejan Markovic, PI

Office: UCLA Engineering IV Building, Room 56-147E (approx. 140 sq. ft).

Graduate student offices: UCLA Engineering department allocates the required office space for graduate students from a common pool. The PI has 14 graduate students.

Computing resources: The PI and his students have access to a 10-node high-performance linuxbased compute cluster, 4 windows-based remote desktop servers. Hardware resources are complemented with software tools for chip design (Cadence, Synopsys, Mentor), algorithm design (Matlab, Simulink), and FPGA prototyping tools (Xilinx, Synplicity).

Laboratory space: The PI has access to several labs equipped with chip instrumentation for the testing of digital circuits. He is primarily using Integrated Circuits and Systems Lab (ICSL) at UCLA EE department. The equipment includes signal generators, spectrum analyzers, logic analyzers, high-speed oscilloscopes, and a high-speed probe station.

Venkat Konda, Consultant

Office: Konda Technologies, 6278, Grand Oak Way, San Jose, CA (approx. 100 sq. ft).

Computing resources: Dr. Konda has a server with two quad-core AMD Athlon 64x2 processors and fedora Linux operating system, two windows-based desktop/laptop machines.

No Government Furnished Property is required for conduct of the proposed research.

Referenes

- [1] [Online], available: http://www.eecg.toronto.edu/~vaughn/challenge/challenge.html
- [2] D. Marković, C. C. Wang, L. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-Power Design in Near-Threshold Region," Proceedings of the IEEE, vol. 98, no. 2, pp. 237-252, Feb. 2010.
- [3] C.-H. Yang and D. Marković, "A Flexible DSP Architecture for MIMO Sphere Decoding," IEEE Trans. Circuits & Systems I, vol. 56, no. 10, pp. 2301-2314, Oct. 2009.
- [4] V. Wang, K. Agarwal, S.R. Nassif, K.J. Nowka, and D. Marković, "A Simplified Design Model for Random Process Variability," IEEE Trans. Semiconductor Manufacturing, vol. 22, no. 1, pp. 12-21, Feb. 2009.
- [5] D. Marković, B. Nikolić, and R.W. Brodersen, "Power and Area Minimization for Multidimensional Signal Processing," IEEE Journal of Solid-State Circuits, vol. 42, no. 4, pp. 922-934, Apr. 2007.
- [6] D. Marković, V. Stojanović, B. Nikolić, M.A. Horowitz, and R.W. Brodersen, "Methods for true energy-performance optimization," IEEE Journal of Solid-State Circuits, vol. 39, no. 8, pp. 1282-93, Aug. 2004.
- [7] F. Chen, et al., "Demonstration of Integrated Micro-Electro-Mechanical Switch Circuits for VLSI Applications," in Proc. IEEE Int. Solid-State Conference (ISSCC'10), Feb. 2010, pp. 26-27.
- [8] V. Karkare, S. Gibson, and D. Marković, "A 130-uW, 64-Channel Spike-Sorting DSP Chip," in Proc. IEEE Asian Solid-State Circuits Conference (A-SSCC'09), Nov. 2009, pp. 289-292.
- [9] C.-H. Yang and D. Marković, "A 2.89mW 50GOPS 16×16 16-Core MIMO Sphere Decoder in 90nm CMOS," in Proc. IEEE European Solid-State Circuits Conference (ESSCIRC'09), Sep. 2009, pp. 344-348.
- [10] F. Chen, H. Kam, D. Markovic, T.-J. King Liu, V. Stojanovic, and E. Alon, "Integrated Circuit Design with NEM Relays," in Proc. Int. Conf. on Computer Aided Design (ICCAD'08), Nov. 2008, pp. 750-757.
- [11] R. Nanda, C.-H. Yang, and D. Markovic, "DSP Architecture Optimization in Matlab/Simulink Environment," in Proc. Int. Symp. on VLSI Circuits (VLSI'08), June 2008, pp. 192-193.
- [12] D. Markovic, C. Cheng, B. Richards, H. So, B. Nikolic, and R.W. Brodersen, "ASIC Design and Verification in an FPGA Environment," in Proc. Custom Integrated Circuits Conference (CICC), Sep. 2007.
- [13] D. Markovic, B. Richards, and R.W. Brodersen, "Technology Driven DSP Architecture Optimization within a High-Level Block Diagram Based Design Flow," in Proc. 40th Asilomar Conf. on Signals, Systems and Computers, Nov. 2006. (Invited)
- [14] D. Markovic, B. Nikolic, and R.W. Brodersen, "A 70 GOPS Multi-Carrier MIMO Chip in 3.5mm2 and 34mW," in Proc. IEEE Int. Symposium on VLSI Circuits (VLSI), June 2006, pp. 196-197.
- [15] R. W. Brodersen, M.A. Horowitz, D. Markovic, B. Nikolic, and V. Stojanovic, "Methods for true power minimization," in Proc. Int'l Conf. on Computer Aided Design (ICCAD), Nov. 2002, pp. 35-42. (Invited)
- [16] V. Stojanovic, D. Markovic, B. Nikolic, M.A. Horowitz, and R.W. Brodersen, "Energy-delay tradeoffs in combinational logic using gate sizing and supply voltage optimization," in Proc. 28th European Solid-State Circuits Conference (ESSCIRC), Sept. 2002, pp. 211-14.

Page 43 of 43 IPR2020-00262

VENKAT KONDA EXHIBIT 2003



39465 Paseo Padre Parkway Suite 3700 Fremont, CA 94538 510.252.1136 (P) 888.700.8917 (F)

August 3, 2010

Dr. William Harrod DARPA TCTO

Dear Dr. Harrod:

I have reviewed the proposal entitled "Energy-Efficient Butterfly FPGA Hardware and Programming Tools," to be submitted by Dejan Markovic and his colleagues to DARPA TCTO.

I find this work of significant value to the development of our future HPC products and am interested to have my engineers interact with UCLA team during the course of this project to ensure that the final outcome can be smoothly transferred to a product.

I look forward to working with Dejan Markovic and his colleagues on this research.

Sincerely,

rd 2

Dr. Chen Chang Founder, CEO BEEcube, Inc. 39465 Paseo Padre Parkway Suite 3700 Fremont, CA 94538 Phone: (510) 252-1136