Formic: Cost-efficient and Scalable Prototyping of Manycore Architectures

Spyros Lyberis, George Kalokerinos, Michalis Lygerakis, Vassilis Papaefstathiou,
Dimitris Tsaliagkos, Manolis Katevenis, Dionisios Pnevmatikatos and Dimitris Nikolopoulos
Institute of Computer Science, Foundation for Research and Technology (FORTH-ICS)
Heraklion, Crete, Greece
{*lyberis,george,ligeraki,papaef,tsaliag,kateveni,pnevmati,dsn*}*@ics.forth.gr*

***Abstract*—Modeling emerging multicore architectures is challenging and imposes a tradeoff between simulation speed and accuracy. An effective practice that balances both targets well is to map the target architecture on FPGA platforms. We find that accurate prototyping of hundreds of cores on existing FPGA boards faces at least one of the following problems: *(i)* limited fast memory resources (SRAM) to model caches, *(ii)* insufficient inter-board connectivity for scaling the design or *(iii)* the board is too expensive. We address these shortcomings by designing a new FPGA board for multicore architecture prototyping, which explicitly targets scalability and cost-efficiency. *Formic* has a 35% bigger FPGA, three times more SRAM, four times more links and costs at most half as much when compared to the popular Xilinx XUPV5 prototyping platform. We build and test a 64-board system by developing a 512-core, MicroBlaze-based, non-coherent hardware prototype with DMA capabilities, with full network-on-chip in a 3D-mesh topology. We believe that Formic offers significant advantages over existing academic and commercial platforms that can facilitate hardware prototyping for future manycore architectures.**

***Keywords*-programmable circuits; prototypes; multicore processing; design methodology.**

I. Introduction

There are two main approaches on multicore systems prototyping: simulating them in software vs. prototyping them in hardware, using FPGAs. Building multicore systems using FPGAs is considerably harder and slower, but running programs on the final system is fast. Moreover, the modeling process provides better insight on the impact of architectural changes and helps to avoid pitfalls of unrealistic software simulation parameters.

The dominant hardware paradigm for modern multicores is the cache-coherent shared memory machine. However, as we enter the manycore era we face more and more challenges regarding both the complexity of hardware cache coherency protocols and the efficiency of shared-memory programming models. One approach that mitigates the problem is to move to non-coherent architectures that rely on distributed memory and explicit communication [1].

To model such architectures on FPGAs, we find that the existing prototyping boards [2] are limited in at least one of the three following aspects: *(i)* they do not feature enough SRAM memory, which is needed to model cache behavior, *(ii)* they do not have enough off-board links to allow connecting many boards together to build scalable systems, or *(iii)* their cost is prohibitive to obtain in large quantities. To address the previous points, we design a new prototype board which is specifically targeted to model architectures of high core counts.

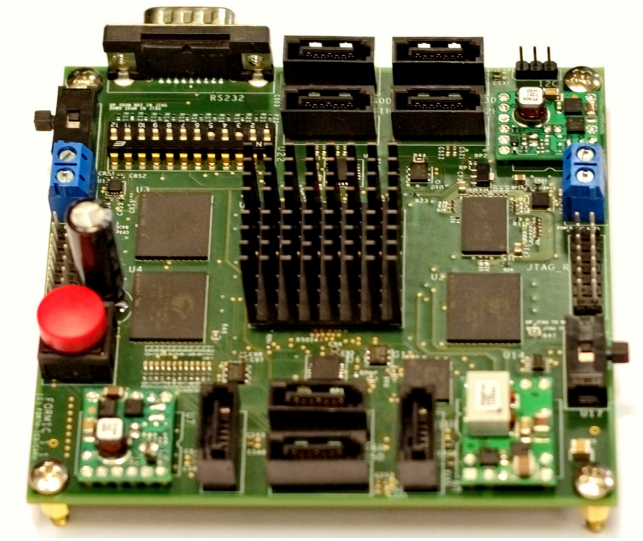

Figure 1. The Formic board

The main contributions of this work are:

- The design of Formic, a novel FPGA prototyping board
- The development of a non-coherent, scalable, hardware architecture for Formic and its proof-of-concept design using 64 boards of 512 total cores

II. The Formic Printed Circuit Board

We introduce *Formic,* a novel hardware prototype board designed specifically to be a cost-efficient building block for scalable systems. It is minimal in concept, small, has both SRAM and DRAM memories, features convenient SATA connectors and is optimized to be a part of a larger system.

The Formic board (figure 1) consists of a Xilinx Spartan-6 LX150T FPGA, three Cypress 9-Mbit 166-MHz ZBT SRAMs and a single Micron 1-Gbit 400-MHz DDR2 SDRAM chip. Each SRAM offers a raw bandwidth of 5.3 Gbps and the DRAM has a peak bandwidth of 12.8 Gbps. We selected the specific Spartan-6 device as an optimal

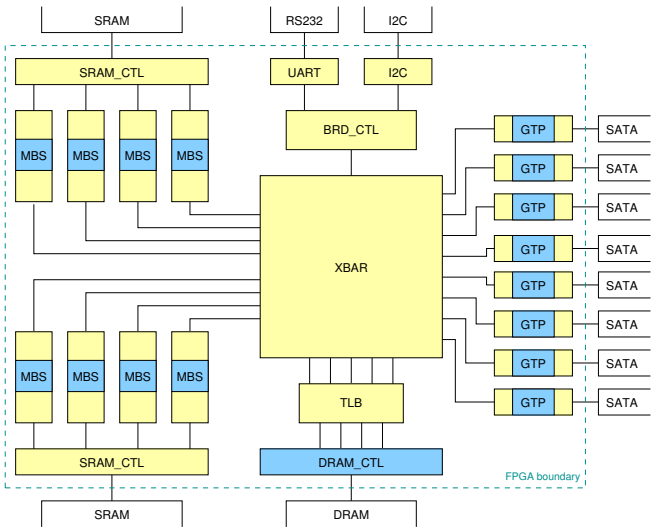

Figure 2. Block diagram of a single FPGA in the non-coherent multicore hardware prototype design. Dark parts indicate usage of Xilinx IP blocks.

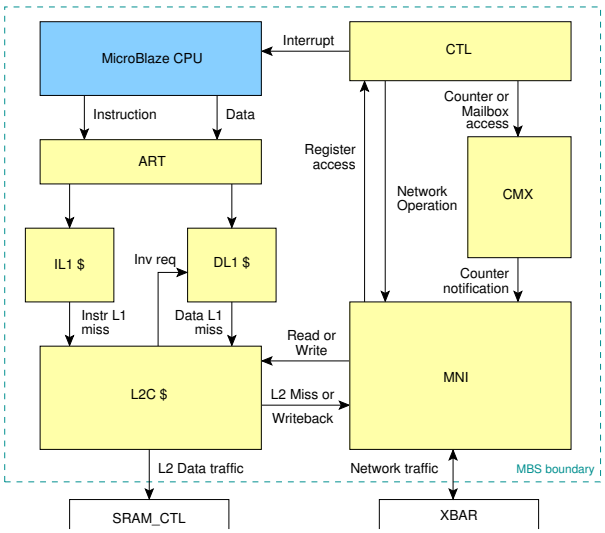

Figure 3. Block diagram of a MicroBlaze Slice (MBS). Arrows indicate which side initiates the related bus transaction.

tradeoff among its high capacity (92K 6-input LUTs, 184K flip-flops, 4.8 MBit BRAMs), its high number of high-speed GTP serial links (8 x 3.0 Gbps) and its low cost (≈$250). This represents a 35% increase in LUTs and 87% savings in cost compared to the XUPV5 Virtex-5 LX110T-1 device (≈$2,000). We offset the intrinsically lower performance of the Spartan parts by selecting the fastest Spartan-6 speed grade (-4), to approach as much as possible the slow Virtex-5 speed grade (-1) used in the XUPV5 platform.

The Formic board is specifically designed to be a building block for large systems. To this end, it has a small form factor (10 x 10 cm). The eight FPGA GTP links are accessible through standard SATA connectors in two groups of four (top and bottom). Half of each group are in "Host" and half in "Device" connection modes, so that plain (instead of crossover) SATA cables can interconnect the boards of a system. All needed voltages are generated on board from a 12V unregulated input. At the left and right PCB sides we place mirrored power supply and buffered JTAG chain connectors, so that boards can be connected in chains – this is controlled using slide switches. We include a configuration PROM for the FPGA, so that large systems can boot fast. Twelve DIP switches are used to identify each board, allowing for systems with up to 4096 boards. Large passive coolers are used for the FPGAs, so that bulk fans can cool multiple boards and minimize the audible noise.

By carefully assigning the FPGA I/O pins to the outer rings, we use only ten PCB layers. The minimum trace width used is 5 mils (0.127 mm) and the smallest drilled holes are 0.3 mm. There are seven generated power supplies on board: 1.23V for the FPGA core, 1.8V for the DRAM, 0.9V for its address pins termination, 2.5V for the SRAMs, two separate 1.20V for the GTP links (top/bottom FPGA banks) and 3.3V for the RS-232 and some regulator bias pins. Two high-quality, differential, 150 MHz oscillators clock the top and bottom GTP banks; these enable a 3.0 Gbps link operation. The FPGA receives a third differential clock input of 200 MHz, which feeds the internal PLLs to generate all needed frequencies for the logic. The board also features twelve LEDs, an RS-232 port, a generic two-pin connector for slow, tri-stated management buses such as I^2C as well as a big, red, comforting reset button.

III. The Scalable Hardware Architecture

The first hardware design project which uses the Formic board is a prototype of a non cache-coherent manycore architecture, based on ideas of the SARC project [3], which was fully implemented in software simulation and partially implemented on a XUPV5 hardware platform [4]. Each board fits in its FPGA eight CPUs, their private L1 and L2 caches, eight GTP links and a full network-on-chip centered around a 22-port crossbar. A variable number of boards can be interconnected in a 3D-mesh using the GTP links, growing the system as required. This hardware architecture will be used in the ENCORE project [5], as the basis of a manycore, task-based runtime system.

Figure 2 shows the block diagram of a single FPGA. There are eight *MicroBlaze Slice (MBS)* blocks, each of them featuring a Xilinx MicroBlaze CPU, its cache hierarchy and the related network-on-chip interface. Four MBS blocks share an SRAM for their L2 data storage, so two SRAM controllers are provided to handle the partitioning and the interface to the SRAM chips. Eight GTP link controllers connect the network-on-chip to the other boards. A board controller handles board-related interfaces, such as the RS-232 port, the I^2C port, the global timer and the TLB

mechanism. Our hardware uses global virtual addresses, so just before the DRAM controller a board-level TLB translates virtual addresses to physical. All these parts are interconnected by a crossbar.

We use the 32-bit Xilinx MicroBlaze RISC CPUs, whose area-optimized, 3-stage pipeline version — including a single-precision FPU, but stripped of its native caches and MMU unit — drops to only 2,100 LUTs. Figure 3 shows the internal of an MBS block, where the MicroBlaze core is located. The CPU has two 32-bit interfaces, one for the instruction and one for the data side. Both are fed into an *Address Region Table (ART)* block, which is programmed by software to specify five regions on the 32-bit global virtual address space and to perform permission checking.

Instruction accesses are cached by a 4-KB, two-way set associative L1 cache *(IL1),* and data accesses are cached by an 8-KB, two-way set associative, write-through, write-no-allocate L1 cache *(DL1).* Because of the write-through behavior, the L2 cache is always up-to-date with (but not necessarily inclusive of) the DL1 contents. DL1 also has an invalidation interface in order to keep up with L2 data changes that may occur from incoming DMAs. L1 misses end up in the *L2 cache (L2C),* a 256-KB, eight-way set associative, write-back, private cache.

The *MBS Network Interface (MNI)* block handles the communication with the network-on-chip. L2C can initiate *Misses,* for fetching cache lines from the local DRAM, and *Writebacks* for cache lines that it evicts to the local DRAM. Our system supports DMAs from/to explicitly named MBS blocks, through a DMA engine inside MNI which can initiate *Reads* and *Writes* from/to the local L2C.

CPU peripheral registers accesses are handled by the *Control (CTL)* block, which keeps 38 memory-mapped registers and controls appropriately all MBS blocks. CTL also features an interrupt controller, which implements 8 maskable and 5 non-maskable interrupts, a private timer, 13 performance counters and a delinquent load/store tracing interface.

The *Counter & Mailbox (CMX)* block keeps 128 counters that can be used to track the progress of ongoing DMA operations [3]. The counters can be polled by the CPU, send an interrupt and/or send notification packets to other counters when the programmed number of acknowledgment packets has been received. CMX also implements a 4-KB incoming mailbox, which can be written from the network and read by the CPU, and a single-word mailslot which is used for remote register reads.

At the heart of the network-on-chip is a 22-port crossbar switch. It uses combined input & output queueing, supported by custom elastic buffers which can hold six packets per queue per VC. All network parts use credit-based flow control for completely lossless transmissions. The network has three separate VCs and uses dimension-order routing to avoid protocol deadlocks. The GTP links are connected to the network-on-chip by blocks which operate on the crossbar elastic buffers. They insert, check and remove CRC headers on the fly to guard for any physical layer errors. They also handle board-to-board flow control by transmitting and receiving credit packets.

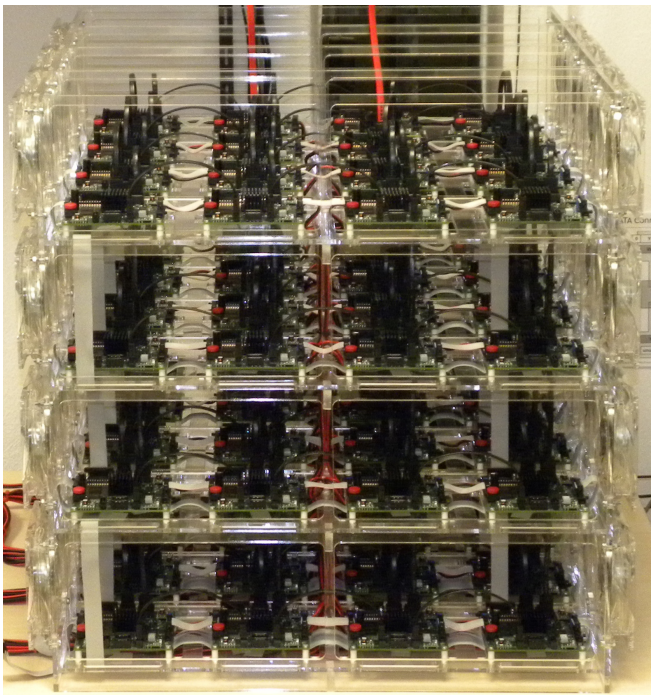

Figure 4. The 4x4x4 cube of 64 Formic boards. Each octo-core board connects with X, Y and Z-axis GTP links to its neighbors.

Packets going to the local DRAM pass through a five-port TLB, which translates the virtual addresses to physical and accesses the DRAM. We use five crossbar ports to correctly match the four Xilinx DRAM controller ports. The TLB can also introduce a minimum, timestamp-based programmable delay to slow down memory accesses in order to model large main memory delays.

We use a variety of edge-aligned clocks to model efficiently the high-bandwidth parts using fast but narrow datapaths; *e.g.* the crossbar is clocked at 160 MHz but has only 16-bit datapaths. This allows for a very efficient FPGA design, but limits the CPU performance – we clock the CPUs at only 10 MHz to maintain realistic cache and communication latencies. We consider this trade-off acceptable, as modeling very high core count on FPGAs using this technique are still orders of magnitude faster than simulating them in software.

IV. The 512-core Proof-of-Concept Design

To evaluate the Formic board and build a proof-of-concept design, we manufactured 68 Formic boards in total. The board-level correctness evaluation was done using a self-testing hardware design in the FPGA, which exercises simultaneously and continuously the three SRAMs, the DRAM

Table I
COMPARING FORMIC TO OTHER HARDWARE PROTOTYPING PLATFORMS

Platform	Dimensions	FPGAs	Layers	Components	LUTs	BRAM	SRAM	DRAM	Board links	Price
BEE2	13.8" x 17.3"	5 × Virtex-II	22	4000	372K (4-inp)	3 MB	–	20 GB	180 Gbps	$10K
BEE3	12" x 16.5"	4 × Virtex-5	18	2500	389K (6-inp)	3.7 MB	–	64 GB	352 Gbps	$18K
XUPV5	8.3" x 5.5"	1 × Virtex-5	14	913	69K (6-inp)	0.7 MB	1 MB	256 MB	7.2 Gbps	$2K
Formic	**4" x 4"**	**1 × Spartan-6**	**10**	**336**	**92K (6-inp)**	**0.6 MB**	**3 MB**	**128 MB**	**19.2 Gbps**	**< $1K**

Table II
LATENCY OF OPERATIONS IN CPU CLOCK CYCLES

Task	CPU clock cycles
L1 hit	1
L2 hit	4
L2 miss	22 + programmable delay
DMA engine initiation	12 (message) - 24 (full)
Network packet latency	3 - 4 (on-board), 5 - 6 (per-board hop)

and all eight GTP links at full speed. The board design was proven to be fully functional from the first run. Formic consumes 0.18 A at 12 V (2.16 W) when the FPGA is deprogrammed, 0.72 A (8.64 W) during the self-test and 0.56 A (6.72 W) when running our prototype design.

Table I compares Formic to the Xilinx XUPV5 board [6] and the Berkeley Emulation Engine boards [2]. BEE2 and BEE3 use multiple FPGAs per board, but do not offer any SRAM. XUPV5 uses a single FPGA and a single SRAM module. Formic offers better SRAM and GTP links ratio per FPGA LUT count than all three boards. Compared to XUPV5, Formic has a 35% bigger FPGA, 3 times more SRAM, half as much DRAM and 4 times more GTP links. Moreover, it is smaller, has 4 less PCB layers, one third the total components and features an FPGA 8 times cheaper in list price. For our test run of 68 boards, including the prototyping costs and the price of the Spartan-6 devices that Xilinx donated to us, we estimate the total cost to be well under $1,000 per board. Instead, XUPV5 has a list price of $2,000, although researchers can obtain it at a discounted rate for $750 [6]. A production run of Formic boards will lead to an even more pronounced price advantage.

We used the Formic boards to create a 3D-mesh of 64 boards (figure 4), which implements a 512-core proof-of-concept prototype. Table II shows the latency in CPU clock cycles for certain tasks. The L2 miss is quite fast at 22 cycles, so a timestamp-based programmable delay is used to model realistic main memory delays. To initiate a full DMA the software needs 24 clock cycles. A more compact "Message" operation can be initiated in only 12 cycles to send a single 32-bit word to the destination. A minimum-sized packet (36 bytes) needs 3-4 cycles to traverse the internal network; board-to-board traversals add 5-6 cycles per hop. These delays are in line with state-of-the art 2D-mesh multicore architectures [1].

The Formic board design is fully described in Verilog and uses roughly 66,000 lines of code. It is accompanied by a full-system simulation environment with automatic non-regression testing. The full octo-core design uses 75% of the LX150T FPGA. We use a script-based, floorplanned, hierarchical flow in Xilinx EDK 12.4 tools.

V. CONCLUSION

We introduce *Formic,* a cost-efficient building block designed for scalable multi-board prototypes. Formic fills an important gap of both academic and commercial platforms, which either are too expensive or do not feature adequate SRAM and board-to-board connections. We design a scalable hardware architecture and implement a 64-board, 512-core proof-of-concept prototype. We argue that although hardware prototyping is harder than software simulation, the effort is well worth the added insight, the modeling accuracy and the execution speed. The Formic 512-core prototype design is available at http://formic-board.com.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union 7^{th} Framework Programme [FP7/2007-2013], under the ENCORE (grant agreement n^o 248647) and TEXT (n^o 261580) Projects. We would like to thank Xilinx for the donation of 64 Spartan-6 FPGA devices.

REFERENCES

[1] J. Howard, S. Dighe, Y. Hoskote, S. R. Vangal, and D. Finan, "A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS," in *ISSCC*. IEEE, 2010, pp. 108–109.

[2] J. D. Davis, C. P. Thacker, and C. Chang, "BEE3: Revitalizing computer architecture research," Microsoft Research, Tech. Rep. MSR-TR-2009-45, April 2009.

[3] M. Katevenis, V. Papaefstathiou, S. G. Kavadias, D. N. Pnevmatikatos, F. Silla, and D. S. Nikolopoulos, "Explicit communication and synchronization in SARC," *IEEE Micro*, vol. 30, no. 5, pp. 30–41, 2010.

[4] S. G. Kavadias, M. Katevenis, M. Zampetakis, and D. S. Nikolopoulos, "On-chip communication and synchronization mechanisms with cache-integrated network interfaces," in *Conf. Computing Frontiers*. ACM, 2010, pp. 217–226.

[5] "ENCORE EU FP7 programme," www.encore-project.eu.

[6] "Xilinx university program XUPV5-LX110T development system," www.xilinx.com/univ/xupv5-lx110t.htm.