The Past and Future of FPGA Soft Processors

Jan Gray
Gray Research LLC
jsgray@acm.org

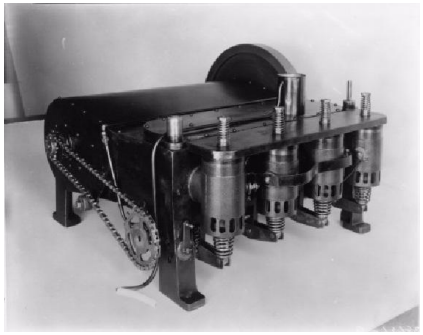
ReConFig 2014 Keynote
9 Dec 2014

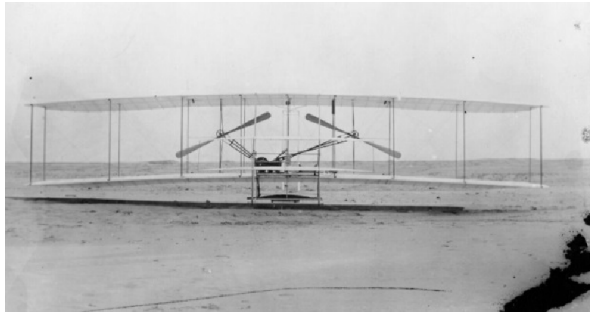
Copyright © 2014, Gray Research LLC. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0) license. http://creativecommons.org/licenses/by/4.0/

In Celebration of Soft Processors

- Looking back
- Interlude: “old school” soft processor, revisited
- Looking ahead

New Engines Bring New Design Eras





1. EARLY DAYS

1985-1990: Prehistory

- XC2000, XC3000: not quite up to the job
  - Early multi-FPGA coprocessors
  - ~8-bit MISCs

1991: XC4000

Xilinx Unveils the XC4000 Family

For many months, our customers have expressed a growing interest in the next generation of Programmable Gate Arrays. Just as microprocessor users have come to expect advance information about the next generation of CPUs, the users of Programmable Gate Arrays are eager to hear about future plans for the next generation of Logic Cell Arrays. Until now such revelations would have been premature. Details were still being finalized, and product availability was too far in the future. Now we are ready to indicate what will become available in mid-1990 as the XC4000 family. We are still necessarily vague about specific details, as we are still working on some of the patents covering key features of the new architecture, but we think that future users should hear in general terms what kind of building blocks will be available from Xilinx. Holding back all information until silicon is available would be a disservice to the community of users who need time to plan the right strategy for their next designs. Early information about the silicon and software applications is also needed to provide third party software support for design entry and design verification.

Technology is evolutionary

The XC4000 will again be based on CMOS SRAM technology. Configuration data will be shifted into the device and will be stored in CMOS latches, similar to the XC2000 and XC3000 parts. But the manufacturing process will use more aggressive sub-micron design rules. This means smaller chips with higher speed, shorter delays, and ultimately lower cost.

Logic density is increased

XC4000 will be a family of devices with logic densities from 2000 to 20,000 gates, more than double the present XC3090's. This complexity range covers 90% of all gate array designs forecast for the mid 1990s.

Speed is enhanced

Improved architectural features will result in a 50% performance improvement over the XC3000 family. Semiconductor processing advances should provide another 30–40% increase in speed. The product of these factors indicates that the XC4000 family will be useful at system clock rates up to twice those possible with the recently introduced 100 MHz version of the XC3000 family.

Versatility is increased

The structure of the XC4000 CLB is similar to the XC3000 CLB. There are two flip-flops, each with a 4-input function generator. But the inputs to each of the function generators in the XC4000 CLB are completely independent. Not only eight, but even nine variables can drive one CLB and can be combined to generate an output or drive a flip-flop. One CLB can thus implement any two independent functions of four variables, and many functions of eight and nine variables are possible, like parity generators and two-bit adders or accumulators. Both flip-flop outputs and combinatorial function outputs are available simultaneously to the general interconnect network.

RAM is available

The XC4000 devices have on-chip high-speed static RAM, up to 2,500 bytes for the largest member of the family, less for the smaller family members. This RAM can be used to store variables, to act as registers, shift registers, FIFO or LIFO buffers, multiple accumulators, look-up tables, etc.

Wide decoding is fast

Decoding of 10- to 32-bit wide address fields was cumbersome and slow in the XC2000 family, but improved with the long lines in the XC3000 family. The XC4000 family goes one step further and provides wide address decoding of up to 50 inputs or internal signals with a speed equivalent to 15 ns PALs.

Counters are compact and fast

The versatility of the CLBs makes it possible to design fast and compact counters. The XC3000 family achieves 25 MHz speed and one bit per CLB for an 8-bit synchronous presettable counter. The XC4000 family beats this with 50 MHz and 2 bits per CLB for the same design. Presettable as well as non-presettable, and up or down as well as up/down counters, 4- to 16-bits long, will be available as part of an extensive macro library.

Arithmetic is simpler and faster

Adders and subtractors using either ripple-carry, carry generate/carry propagate, or conditional sum algorithms are faster and smaller than in the XC3000 family. A 16-bit adder can run at 50 MHz.

Pipelining speeds up the system

The abundance of free flip-flops invites pipelined design techniques for highest throughput speed. Dividing a complex function into several parts and executing them in sequence and in parallel can multiply system performance.

Abundant routing resources

Compared to the XC2000 and XC3000 families, routing resources have been increased dramatically. There are more than twice as many vertical and horizontal long lines, and more local interconnects. Not only are there more routing resources, they are also more accessible to the CLBs, and the access is more regular, less specialized. This makes it easier for the software to complete the routing of complex interconnect patterns. Our software designers worked intimately with our chip architects to specify a structure that is not only powerful and flexible, but also easy for the software to utilize. The goal is to provide automated push-button software that routes almost all designs, even densely populated ones, automatically. But we still give the designer the option to get involved in the partitioning,

XILINX®

XC4000, XC4000A, XC4000H Logic Cell Array Families

Product Description

Features

- Third Generation Field-Programmable Gate Arrays
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate
  - Programmable input pull-up or pull-down resistors
  - 12-mA sink current per output (XC4000 family)
  - 24-mA sink current per output (XC4000A and XC4000H families)
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4000 families of Field-Programmable Gate Arrays (FPGAs) provide the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array.

The XC4000 families provide a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs).

XC4000-family devices have generous routing resources to accommodate the most complex interconnect patterns. XC4000A devices have reduced sets of routing resources, sufficient for their smaller size. XC4000H high I/O devices maintain the same routing resources and CLB structure as the XC4000 family, while nearly doubling the available I/O.

The devices are customized by loading configuration data into the internal memory cells. The FPGA can either actively read its configuration data out of external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA (slave and peripheral modes).

The XC4000 families are supported by powerful and sophisticated software, covering every aspect of design: from schematic entry, to simulation, to automatic block placement and routing of interconnects, and finally the creation of the configuration bit stream.

Since Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening the design and development cycle, but they also offer a cost-effective solution for production rates well beyond 1000 systems per month.

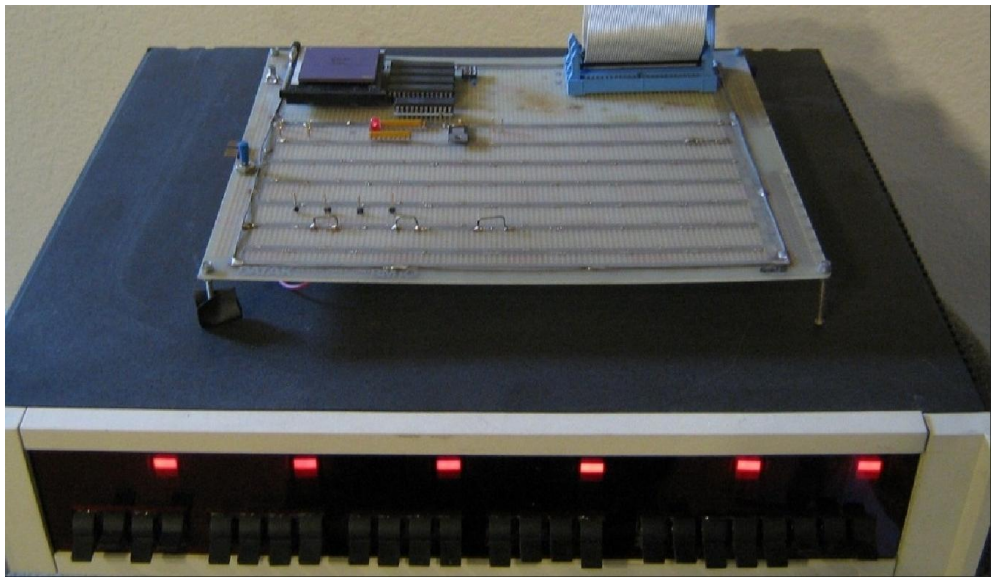
Table 1. The XC4000 Families of Field-Programmable Gate Arrays

Device	XC4002A	4003/3A	4003H	4004A	4005/5A	4005H	4006	4008	4010/10D	4013/13D	4020	4025
Appr. Gate Count	2,000	3,000	3,000	4,000	5,000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLB Matrix	8 x 8	10 x 10	10 x 10	12 x 12	14 x 14	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	28 x 28	32 x 32
Number of CLBs	64	100	100	144	196	196	256	324	400	576	784	1,024
Number of Flip-Flops	256	360	200	480	616	392	768	936	1,120	1,536	2,016	2,560
Max Decode Inputs (per side)	24	30	30	36	42	42	48	54	60	72	84	96
Max RAM Bits	2,048	3,200	3,200	4,608	6,272	6,272	8,192	10,368	12,800*	18,432*	25,088	32,768
Number of IOBs	64	80	160	96	112	192	128	144	160	192	224	256

*XC4010D and XC4013D have no RAM

1991: RISC4005 [P. Freidin]

- The first monolithic general purpose FPGA CPU
- **"FPGA Devices:** 1 Xilinx XC4005 ...
  On-board RAM: 64K Words (16 bit words)
  Notes: A 16 bit RISC processor that requires 75% of an XC4005, 16 general registers, 4 stage pipeline, 20 MHz. Can be integrated with peripherals on 1 FPGA, and ISET can be extended. … Includes a macro assembler, gate level simulator, ANSI C compiler, and a debug monitor."
  [Steve Guccione: List of FPGA-based Computing Machines, http://www.cmpware.com/io.com/guccione/HW_list.html]



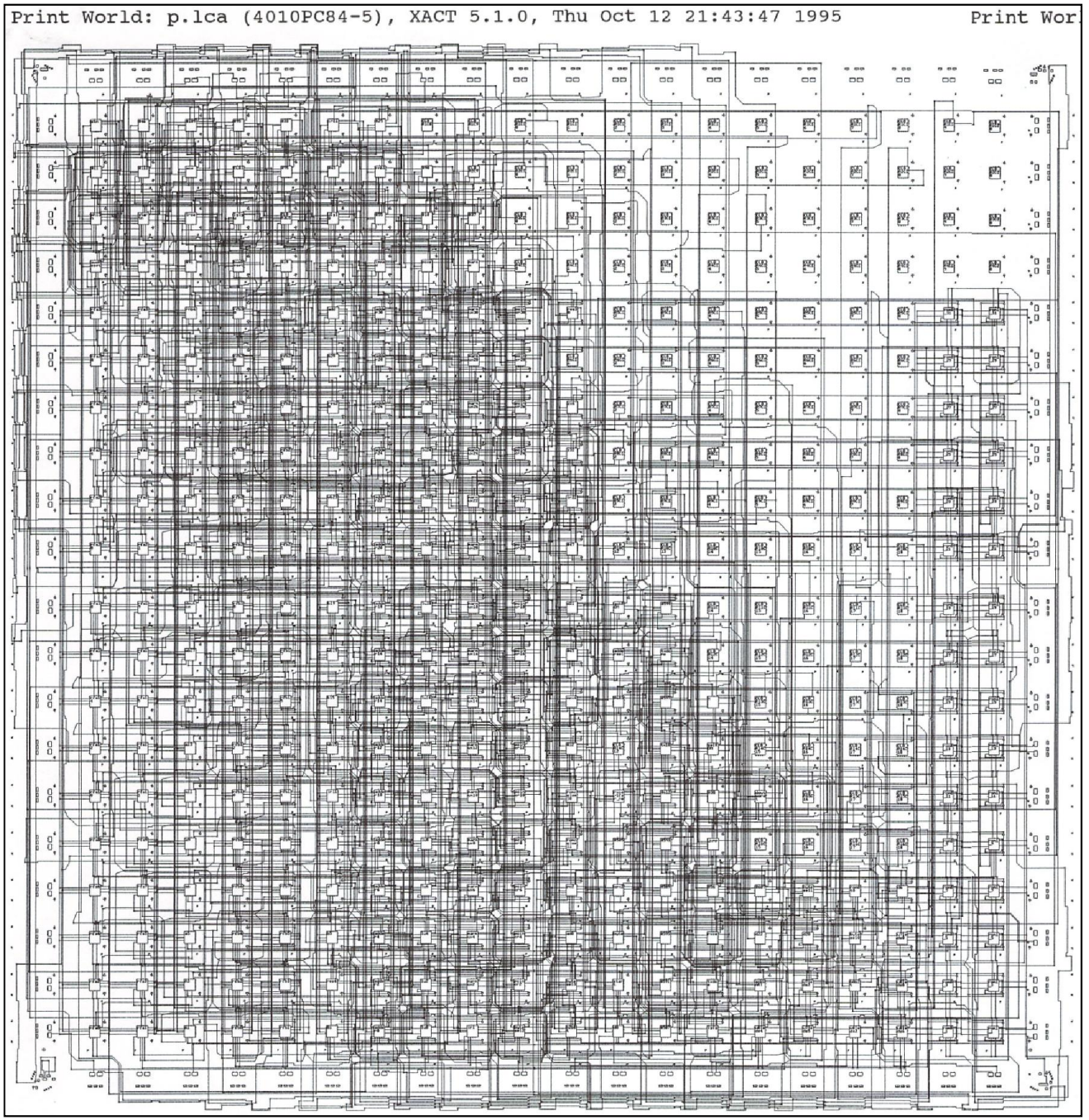
Photos: Philip Freidin

1994-95: Gathering Steam

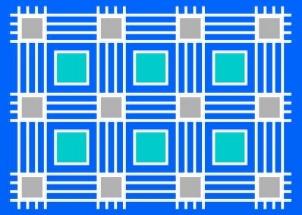
- Communities: FCCM, comp.arch.fpga
  [http://fpga-faq.org/archives/index.html]
- Research, commercial interest – *OneChip, V6502*

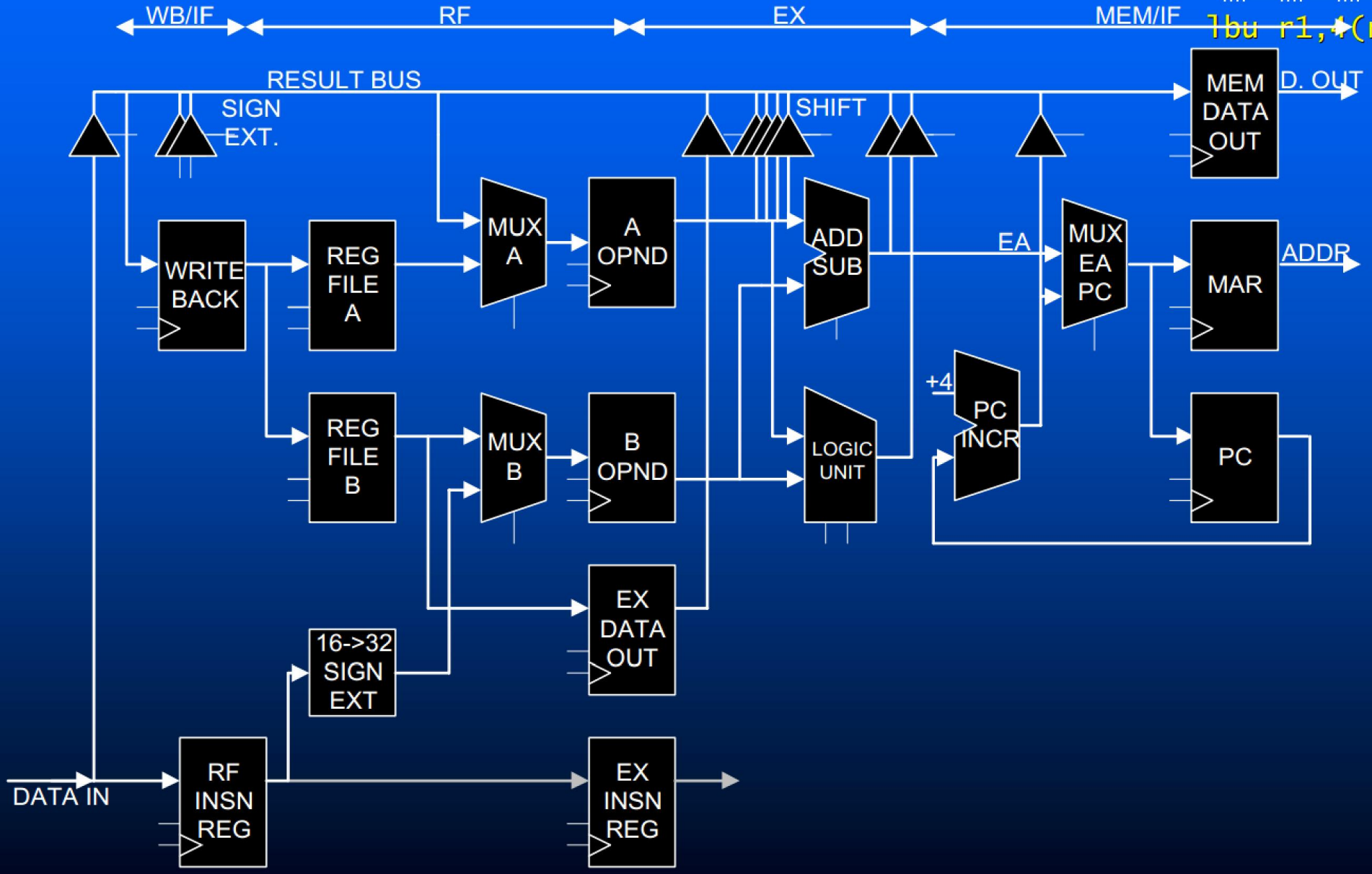
1995: J32

- 32-bit RISC + “SoC”
- Integer only
- 33 MHz ÷ 2ϕ
- 4-stage pipeline
- <60% of XC4010
- C++ → XNF → .bit



J32 Microarchitecture





1995-96: XC4000E and FLEX10K

PRODUCT INFORMATION — COMPONENTS

The High-Performance XC4000E Family

New Family Achieves 50 Percent Performance Increase Over the Popular XC4000 Family

Xilinx has begun shipping the new XC4000E family, the successor to the popular XC4000 FPGA family. The XC4000E FPGA family increases performance over the existing XC4000 family by up to 50 percent, and is ideal for high-density, high-performance applications such as microprocessor bus interfacing, digital signal processing, image processing, and high-speed telecommunications.

This dramatic performance improvement results from an improved design, a new 0.5µ triple-layer-metal process technology, and the new Select-RAM™ on-chip memory feature. The XC4000E family features eight devices ranging from 3,000 to 25,000 gates, including the XC4020E device, the first 20,000 gate device offered by Xilinx. (Higher-density devices will be announced later this year.) Since the XC4000E family is based on the XC4000 family architecture, XC4000E devices are backward compatible (i.e., drop-in, with the same bitstream) with the equivalent XC4000 devices.

Unique capabilities and features of the XC4000E FPGAs include:

- Up to 50 percent faster circuit speed (XC4000E-2 as compared to XC4000-4; *see table on page 24)*
- 60 percent faster carry chain for arithmetic functions
- Lower component prices than the XC4000 family
- Select-RAM memory reduces chip count and design time while increasing RAM performance by up to 2X
- Select-RAM memory is mode-selectable
  - synchronous/asynchronous
  - single-port/dual-port
- 100 percent PCI compliant and suitable

Architectural Features of the XC4000E

The architecture of the XC4000E is a superset of the XC4000; XC4000E devices are bitstream and pin-compatible with the equivalent XC4000 family FPGAs. Thus, current designs using XC4000 FPGAs can take immediate advantage of the XC4000E's increased speed by simply inserting new devices into existing XC4000 sockets.

The XC4000E architecture features an enhanced CLB (configurable logic block) with several new modes for configuring on-chip memory (called Select-RAM). As with the XC4000 architecture, the CLB's look-up tables optionally can be used as

XILINX®

The Altera Advantage News & Views

Newsletter for Altera Customers ◆ Second Quarter ◆ May 1996

Altera Ships 100,000-Gate PLD

Altera is now shipping the EPF10K100 device, which is not only the largest member of the FLEX 10K family, but also the largest device in the programmable logic industry. FLEX 10K devices contain both a logic array and an embedded array that can be used for RAM, ROM, or complex logic functions. With the added capability of the embedded array, the EPF10K100 offers 100,000 gates—a breakthrough for programmable logic.

The EPF10K100 is more than twice as large as any other programmable logic device (PLD) shipping today. Figure 1 compares the speed and density of the FLEX 10K family to the AT&T ORCA and Xilinx XC4000 families. The EPF10K100 contains approximately 10 million transistors; in contrast, the new Pentium Pro (P7) microprocessor from Intel contains 6.5 million transistors.

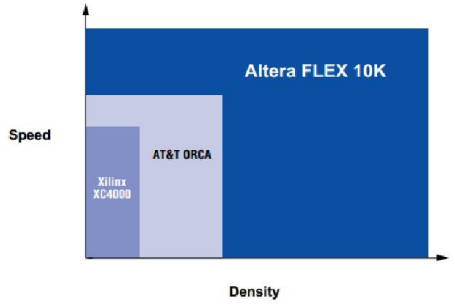
The EPF10K100 is an ideal prototyping and initial production device for ASIC designs. According to market analysts at Dataquest, 80% of all 1996 gate array design starts will require device densities of less than 100,000 gates. As a result, the FLEX 10K family meets the density demands of most gate array designs. In addition to 100,000 gates, the EPF10K100 contains embedded array blocks (EABs) that can integrate specialized arithmetic, digital signal processing (DSP), and large on-chip memory functions. With the EPF10K100, design engineers can now create and prototype 100,000-gate gate array designs with system speeds of over 70 MHz.

FLEX 10K Architecture

The FLEX 10K architecture implements complex functions as efficiently as embedded gate arrays—the fastest-growing segment of the gate array market. Like standard gate arrays, embedded gate arrays implement general logic in a conventional "sea-of-gates" architecture. In addition, embedded gate arrays have dedicated die areas for implementing large, specialized functions. Embedded gate arrays contain functions that are embedded in silicon, which provides reduced die area and increased speed compared to standard gate arrays. However, the embedded functions typically cannot be customized, thus limiting design flexibility.

In contrast, FLEX 10K devices are programmable, providing you with full control over logic while facilitating iterative design changes during debugging. Each FLEX 10K device contains an embedded array and a logic array. The embedded array can implement

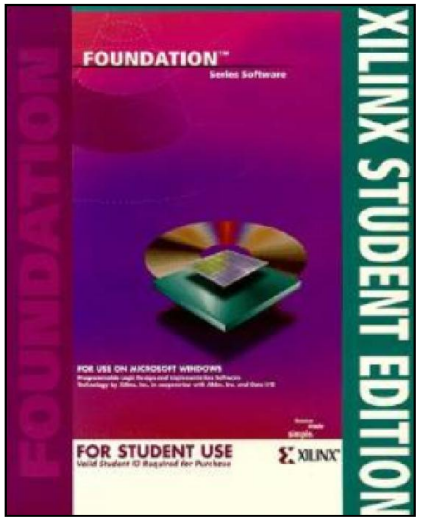
Figure 1. Comparison of Speed & Density

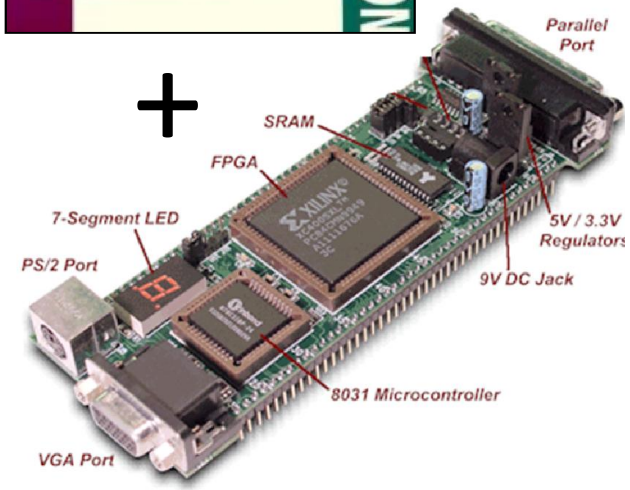


Source: Altera Corporation

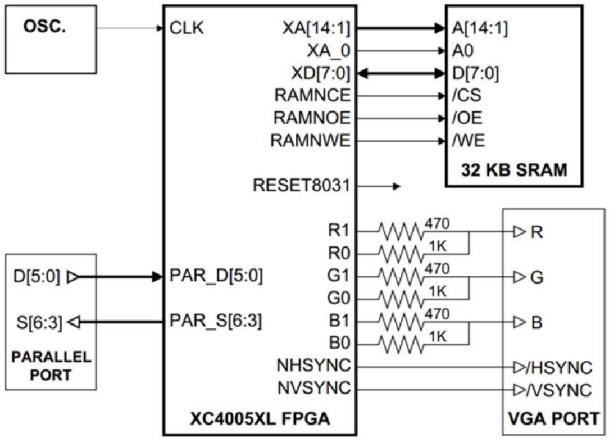
ConF

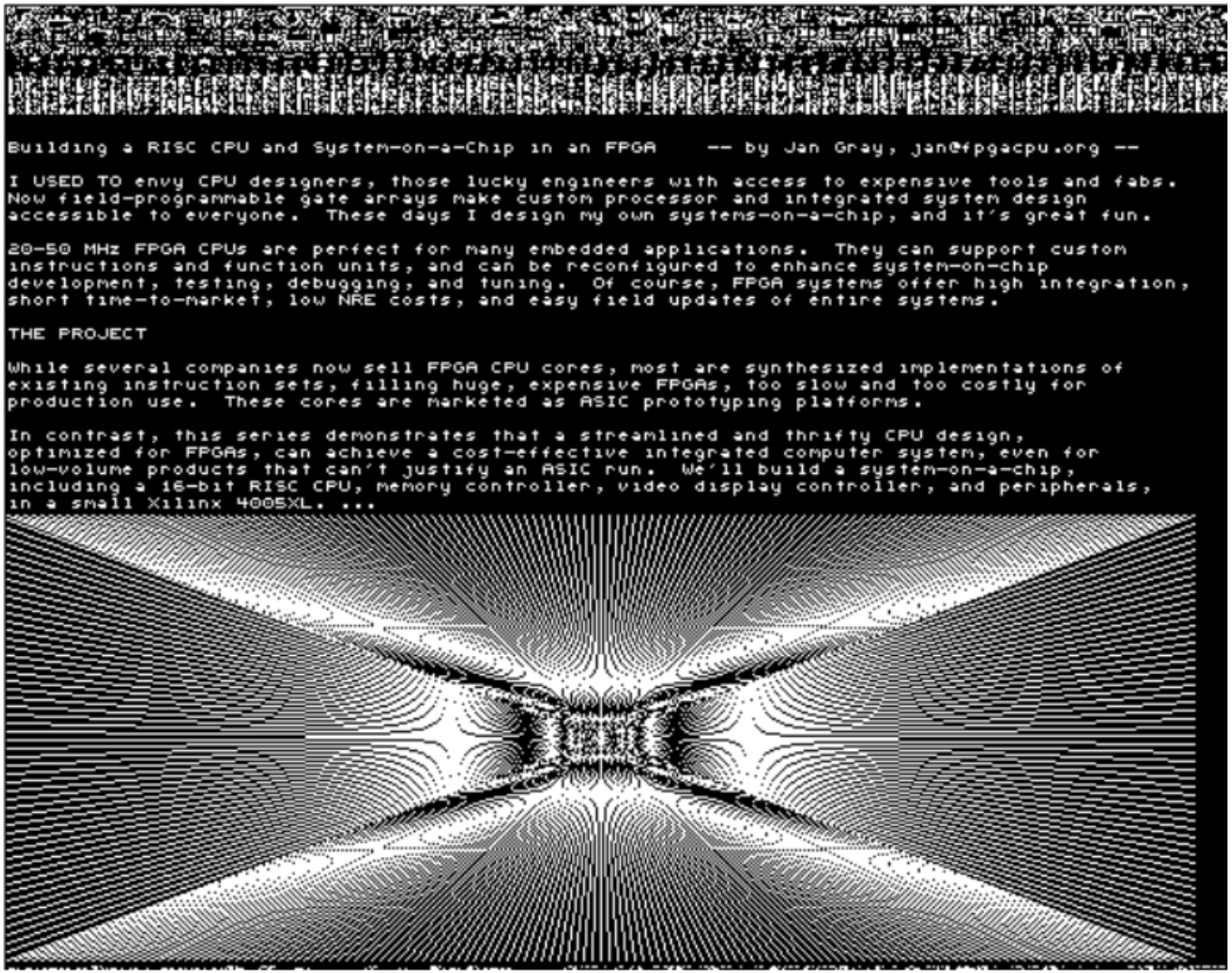
1998: XSOC/xr16 Kit





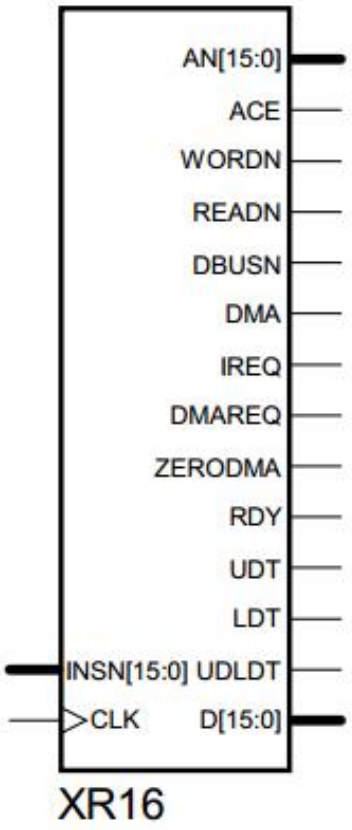
http://www.xess.com/shop/product/xs40-005e/



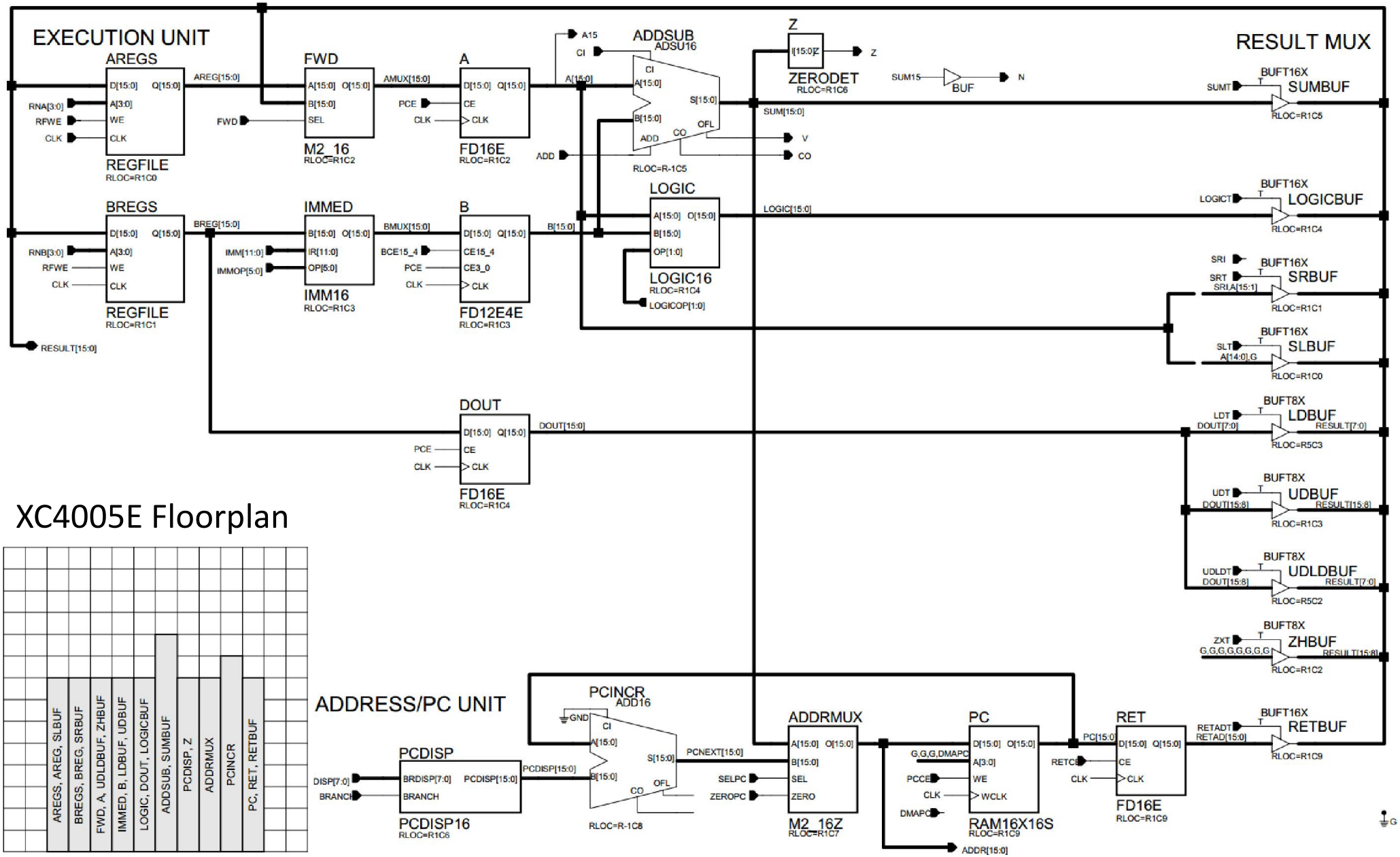


1998: xr16

- 40 MHz 16-bit RISC, DMA
- XC4005E/XL/Spartan-10, 265 LUTs
- LCC C compiler, simulator
- *Building a RISC System in an FPGA,* Circuit Cellar series [http://fpgacpu.org/xsoc/cc.html]
- *FPGA CPU News*, fpga-cpu list



1998: xr16 Datapath



XC4005E Floorplan

1998: Virtex

The New Virtex FPGA Family

VIRTEX™

Much More Than Just a Million Gates...

Now, for the first time, you can create complete, highly complex, high-performance systems in a single programmable device. Using our new Virtex FPGAs and our new high-speed development tools, your creative ideas will reach full production more quickly, more easily, and less expensively than ever before. This is a revolution in logic design.

by Carlis Collins, Managing Editor of Corporate Communications, Xilinx, editor@xilinx.com

These are fast moving times; every day that you waste in unnecessary development, debugging, and manufacturing, is lost revenue and an invitation to your competitors. You have to manage your development time, and wisely leverage all of the available resources, to remain competitive; you have to think in extraordinary ways. That's the motivation behind our new Virtex family.

From the beginning, we coordinated our research and development, in both device and software technology, to create a complete, fully integrated solution for digital logic design. The Virtex solution is extraordinary and it's the wisest use of your time and resources for gaining a competitive advantage.

Virtex is Extraordinary

The Virtex architecture is a new concept in programmable logic technology, one that is both evolutionary and revolutionary. We built on our previous success and added many new features, ones that you requested, that allow you to create a true system on a single chip. These new ASIC-like features, combined with our new fast-compile software and lower pricing, now give you the best solution for all but the highest-volume applications. Designs that once required custom ASICs, can now go straight to production using Virtex FPGAs, saving you a lot of time, trouble, and expense.

There are four key features that make all of this possible:

Virtex Overview

The Virtex architecture includes many more features than can be adequately described in this article, that's why we included an abbreviated Virtex data sheet, starting on page 41. However, here's a review of the Virtex FPGA highlights:

- Nine Virtex devices, 50,000 to 1,000,000 system gates.
- 200MHz on-chip, 160MHz I/O performance.
- Delay-Locked Loop clocking.
- Flexible on-chip RAM.
- Support for 16 I/O standards.
- Core-friendly, fast, predictable routing.
- Very fast device programming.
- Dedicated carry logic for high-speed arithmetic.
- Dedicated multiplier support.
- Cascade chain for wide-input functions.
- Internal 3-state bussing.
- IEEE 1149.1 boundary-scan capability.
- 66-MHz/64-bit PCI compatible.
- Hot-swappable for Compact PCI.
- Die temperature sensing device, on-chip.
- 0.22µ, 5-layer metal process.
- 100% factory tested.
- Prices starting at less than $10 (50K gate XCV50, in high volumes).
- 1M and 300K gate devices available today.

XILINX®

Virtex™ 2.5 V Field Programmable Gate Arrays

DS003-1 (v4.0) March 1, 2013 **Product Specification**

Features

- Fast, high-density Field Programmable Gate Arrays
  - Densities from 50k to 1M system gates
  - System performance up to 200 MHz
  - 66-MHz PCI Compliant
  - Hot-swappable for Compact PCI
- Multi-standard SelectIO™ interfaces
  - 16 high-performance interface standards
  - Connects directly to ZBTRAM devices
- Built-in clock-management circuitry
  - Four dedicated delay-locked loops (DLLs) for advanced clock control
  - Four primary low-skew global clock distribution nets, plus 24 secondary local clock nets
- Hierarchical memory system
  - LUTs configurable as 16-bit RAM, 32-bit RAM, 16-bit dual-ported RAM, or 16-bit Shift Register
  - Configurable synchronous dual-ported 4k-bit RAMs
  - Fast interfaces to external high-performance RAMs
- Flexible architecture that balances speed and density
  - Dedicated carry logic for high-speed arithmetic
  - Dedicated multiplier support
  - Cascade chain for wide-input functions
  - Abundant registers/latches with clock enable, and dual synchronous/asynchronous set and reset
  - Internal 3-state bussing
  - IEEE 1149.1 boundary-scan logic
  - Die-temperature sensor diode
- Supported by FPGA Foundation™ and Alliance Development Systems
  - Complete support for Unified Libraries, Relationally Placed Macros, and Design Manager
  - Wide selection of PC and workstation platforms
- SRAM-based in-system configuration
  - Unlimited re-programmability
  - Four programming modes
- 0.22 µm 5-layer metal process
- 100% factory tested

Description

The Virtex FPGA family delivers high-performance, high-capacity programmable logic solutions. Dramatic increases in silicon efficiency result from optimizing the new architecture for place-and-route efficiency and exploiting an aggressive 5-layer-metal 0.22 µm CMOS process. These advances make Virtex FPGAs powerful and flexible alternatives to mask-programmed gate arrays. The Virtex family comprises the nine members shown in Table 1.

Building on experience gained from previous generations of FPGAs, the Virtex family represents a revolutionary step forward in programmable logic design. Combining a wide variety of programmable system features, a rich hierarchy of fast, flexible interconnect resources, and advanced process technology, the Virtex family delivers a high-speed and high-capacity programmable logic solution that enhances design flexibility while reducing time-to-market.

Table 1: **Virtex Field Programmable Gate Array Family Members**

Device	System Gates	CLB Array	Logic Cells	Maximum Available I/O	Block RAM Bits	Maximum SelectRAM+™ Bits
XCV50	57,906	16x24	1,728	180	32,768	24,576
XCV100	108,904	20x30	2,700	180	40,960	38,400
XCV150	164,674	24x36	3,888	260	49,152	55,296
XCV200	236,666	28x42	5,292	284	57,344	75,264
XCV300	322,970	32x48	6,912	316	65,536	98,304
XCV400	468,252	40x60	10,800	404	81,920	153,600
XCV600	661,111	48x72	15,552	512	98,304	221,184
XCV800	888,439	56x84	21,168	512	114,688	301,056
XCV1000	1,124,022	64x96	27,648	512	131,072	393,216

2000: Nios, *SOPC Builder*

Excalibur Development Kit

with the Nios Embedded Processor

June 2000, ver. 1.0 Data Sheet

The Nios embedded processor is a fully-integrated soft core processor developed to take advantage of Altera's PLD architecture. This section describes the Nios embedded processor and peripherals included in the Excalibur development kit. Figure 2 on page 3 shows a diagram of the Nios processor.

- Nios embedded processor core
  - 16-bit instruction set
  - Data width of 16 or 32 bits
  - One instruction per clock cycle
  - Support for on-chip and off-chip memory
  - Up to 50 million instructions per second (MIPS) performance
  - Windowed register access for fast interrupt handling
  - Register file with up to 512 32-bit registers
  - Shift depth from 8 to 32 bits
  - Register access to peripherals
- Nios microperipherals
  - Universal asynchronous receiver/transmitter (UART)
  - Parallel input/output (PIO)
  - Timer
  - Memory interfaces to SRAM and FLASH
- Low-cost accessibility
  - Free licensing from the Altera web site
  - Royalty-free
- MegaWizard™ Plug-In interface, which creates and configures the Nios embedded processor core
- SOPC Builder software
  - Generates on-chip peripheral bus
  - Configures peripherals and generates C header files

For more information on the Nios embedded processor, see the *Nios Soft Core Embedded Processor Data Sheet*.

Altera® users now have a single system-on-a-programmable-chip (SOPC) solution. The Excalibur™ development kit, featuring the Nios™ soft core embedded processor, contains all the tools that hardware and software engineers need to create high-performance systems in Altera programmable logic devices (PLDs). This kit provides a PLD-optimized processor that designers can implement immediately, improving design workflow and speeding time-to-market.

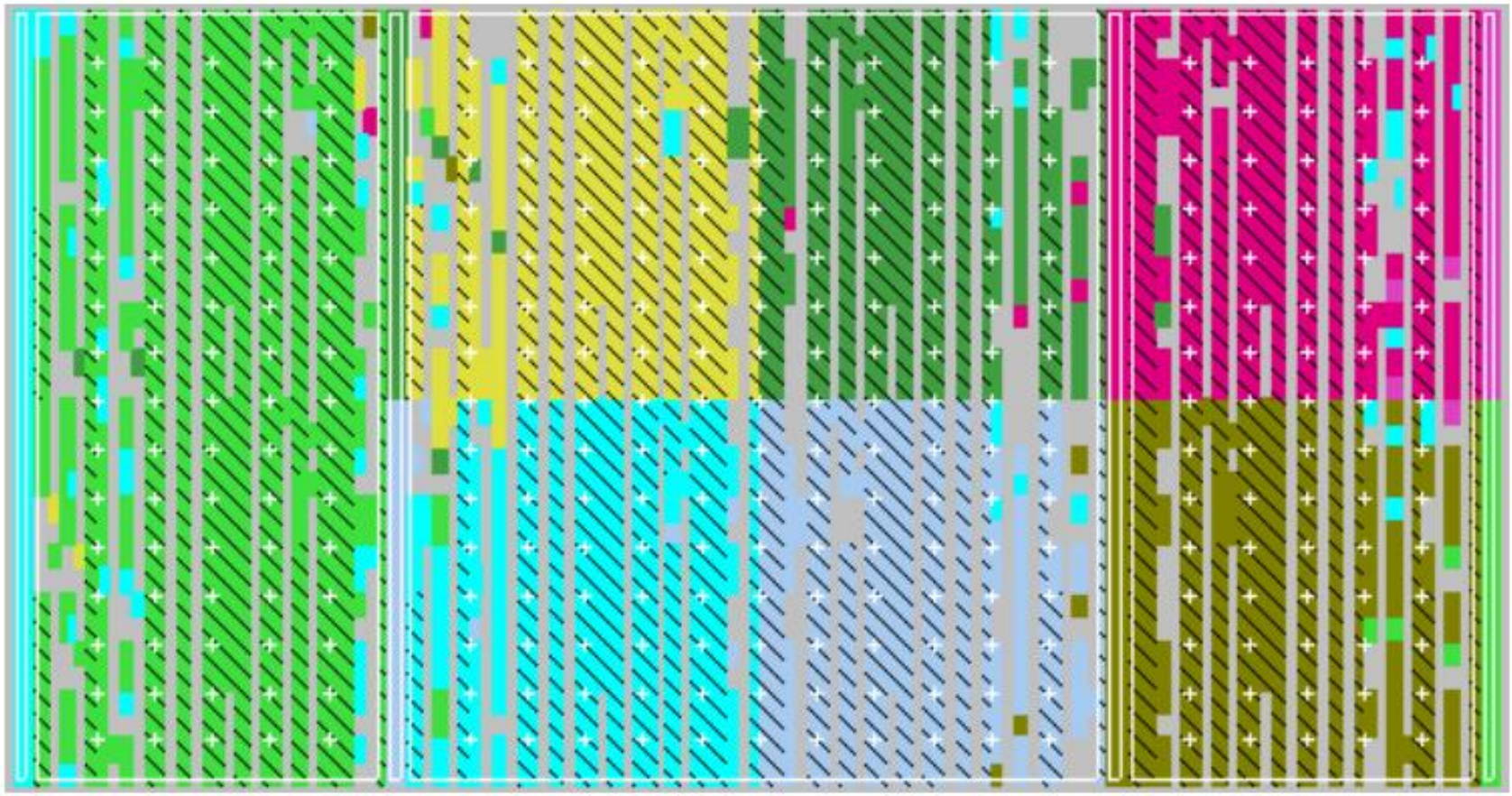
As shown in Figure 1, the Excalibur development kit contains:

Figure 1. Excalibur Development Kit, with the Nios Embedded Processor

- Nios configurable RISC processor core and peripherals
- GNUPro® compiler and debugger from Cygnus®, a Red Hat® company
- The Quartus™ development software
- ByteBlasterMV™ download cable
- Development board populated with the APEX™ EP20K200E device
- SOPC reference design

2000: FPGA Chip Multiprocessors

- 3rd gen 16/32-bit RISC PE: 200/330 LUTs **+ 1 BRAM**
- 8 cores fit in an XCV50E, 60 in an XCV600E



2001-02: Virtex-II/Pro, MicroBlaze, EDK

XILINX®

Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Introduction and Overview

DS083 (v5.0) June 21, 2011 — **Product Specification**

Summary of Virtex-II Pro™ / Virtex-II Pro X Features

- High-Performance Platform FPGA Solution, Including
  - Up to twenty RocketIO™ or RocketIO X embedded Multi-Gigabit Transceivers (MGTs)
  - Up to two IBM PowerPC™ RISC processor blocks
- Based on Virtex-II™ Platform FPGA Technology
  - Flexible logic resources
  - SRAM-based in-system configuration
  - Active Interconnect technology
  - SelectRAM™+ memory hierarchy
  - Dedicated 18-bit x 18-bit multiplier blocks
  - High-performance clock management circuitry
  - SelectI/O™-Ultra technology
  - XCITE Digitally Controlled Impedance (DCI) I/O

Virtex-II Pro / Virtex-II Pro X family members and resources are shown in Table 1.

Table 1: **Virtex-II Pro / Virtex-II Pro X FPGA Family Members**

Device[1]	RocketIO Transceiver Blocks	PowerPC Processor Blocks	Logic Cells[2]	CLB (1 = 4 slices = max 128 bits)		18 X 18 Bit Multiplier Blocks	Block SelectRAM+		DCMs	Maximum User I/O Pads
				Slices	Max Distr RAM (Kb)		18 Kb Blocks	Max Block RAM (Kb)		
XC2VP2	4	0	3,168	1,408	44	12	12	216	4	204
XC2VP4	4	1	6,768	3,008	94	28	28	504	4	348
XC2VP7	8	1	11,088	4,928	154	44	44	792	4	396
XC2VP20	8	2	20,880	9,280	290	88	88	1,584	8	564
XC2VPX20	8[4]	1	22,032	9,792	306	88	88	1,584	8	552
XC2VP30	8	2	30,816	13,696	428	136	136	2,448	8	644
XC2VP40	0[3], 8, or 12	2	43,632	19,392	606	192	192	3,456	8	804
XC2VP50	0[3] or 16	2	53,136	23,616	738	232	232	4,176	8	852
XC2VP70	16 or 20	2	74,448	33,088	1,034	328	328	5,904	8	996
XC2VPX70	20[4]	2	74,448	33,088	1,034	308	308	5,544	8	992
XC2VP100	0[3] or 20	2	99,216	44,096	1,378	444	444	7,992	12	1,164

Notes:
1. -7 speed grade devices are not available in Industrial grade.
2. Logic Cell ≈ (1) 4-input LUT + (1)FF + Carry Logic
3. These devices can be ordered in a configuration without RocketIO transceivers. See Table 3 for package configurations.
4. Virtex-II Pro X devices equipped with RocketIO X transceiver cores.

RocketIO X Transceiver Features (XC2VPX20 and XC2VPX70 Only)

- Variable-Speed Full-Duplex Transceiver (XC2VPX20) Allowing 2.488 Gb/s to 6.25 Gb/s Baud Transfer Rates.
  - Includes specific baud rates used by various standards, as listed in Table 4, Module 2.
- Fixed-Speed Full-Duplex Tranceiver (XC2VPX70) Operating at 4.25 Gb/s Baud Transfer Rate.
- Eight or Twenty Transceiver Modules on an FPGA, Depending upon Device
- Monolithic Clock Synthesis and Clock Recovery
  - Eliminates the need for external components
- Automatic Lock-to-Reference Function
- Programmable Serial Output Differential Swing
  - 200 mV to 1600 mV, peak-peak
  - Allows compatibility with other serial system voltage levels
- Programmable Pre-emphasis Levels 0 to 500%
- Telecom/Datacom Support Modes
  - "x8" and "x10" clocking/data paths
  - 64B/66B clocking support

XILINX® logiCORE™

MicroBlaze™ RISC 32-Bit Soft Processor

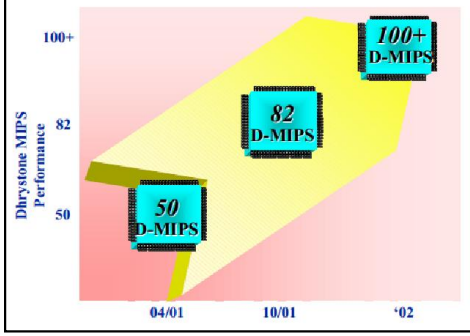
August 21, 2002 — **Product Brief**

Features

- Supports Virtex, Virtex-E, Virtex-II Pro, Spartan-II, and Spartan-IIE devices
- Performance: 102 Dhrystone MIPS (D-MIPS) on Virtex-II Pro device at 150 MHz
- Minimum logic requirements: 900 logic cells
- 32-bit pipelined RISC architecture
- 32 x 32-bit general purpose registers
- Implementation in Virtex-II and later devices support hardware multiply
- Supports Local Memory Bus (LMB) for fast access of on-chip BRAMs
- Supports IBM CoreConnect On-chip Peripheral Bus (OPB) for accessing peripherals
- Processor peripherals compatible with PowerPC on Virtex-II Pro
- Complete hardware and software development tool and debug solution

Performance

The MicroBlaze processor is one part of an expanding array of processor functions that work together seamlessly to create the highest possible performance on a single FPGA. Beyond providing complete design solutions today, the MicroBlaze development platform will continue to support the designer in the future.



LogiCORE™ Facts	
Core Specifics	
Special Features	Up to102 D-MIPS at 150 MHz
Provided With Core	
Documentation	Embedded Software Tools Handbook and Embedded Processor IP Handbook
Design Files	Simulation Model Generator and Xilinx Generic Netlist Format (ngo netlist)
Design Tool Support	
Xilinx Implementation Tools	Xilinx ISE 5.1
Verification Tools	MicroBlaze GNU Debugger and Xilinx Microprocessor Debug (XMD) Tools
Support	
Please contact the Xilinx Hotline for technical support. Xilinx provides technical support for this LogiCORE Product when used as described in Product Documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices not listed above, or if customized beyond that allowed in the product documentation, or if any changes are made in sections of design marked as "DO NOT MODIFY".	

Table 1: MicroBlaze Speed, Performance, and Logic Cells in Xilinx Devices

Device Family	Speed	Performance	Logic Cells
Virtex-II Pro (-6)	150 MHz	102 D-MIPS	900
Virtex-II (-5)	125 MHz	82 D-MIPS	900
Virtex-E (-7)	75 MHz	49 D-MIPS	1050
Spartan-II (-6)	65 MHz	43 D-MIPS	1050
Spartan-IIE (-7)	75 MHz	49 D-MIPS	1050

2002: The End of the Beginning

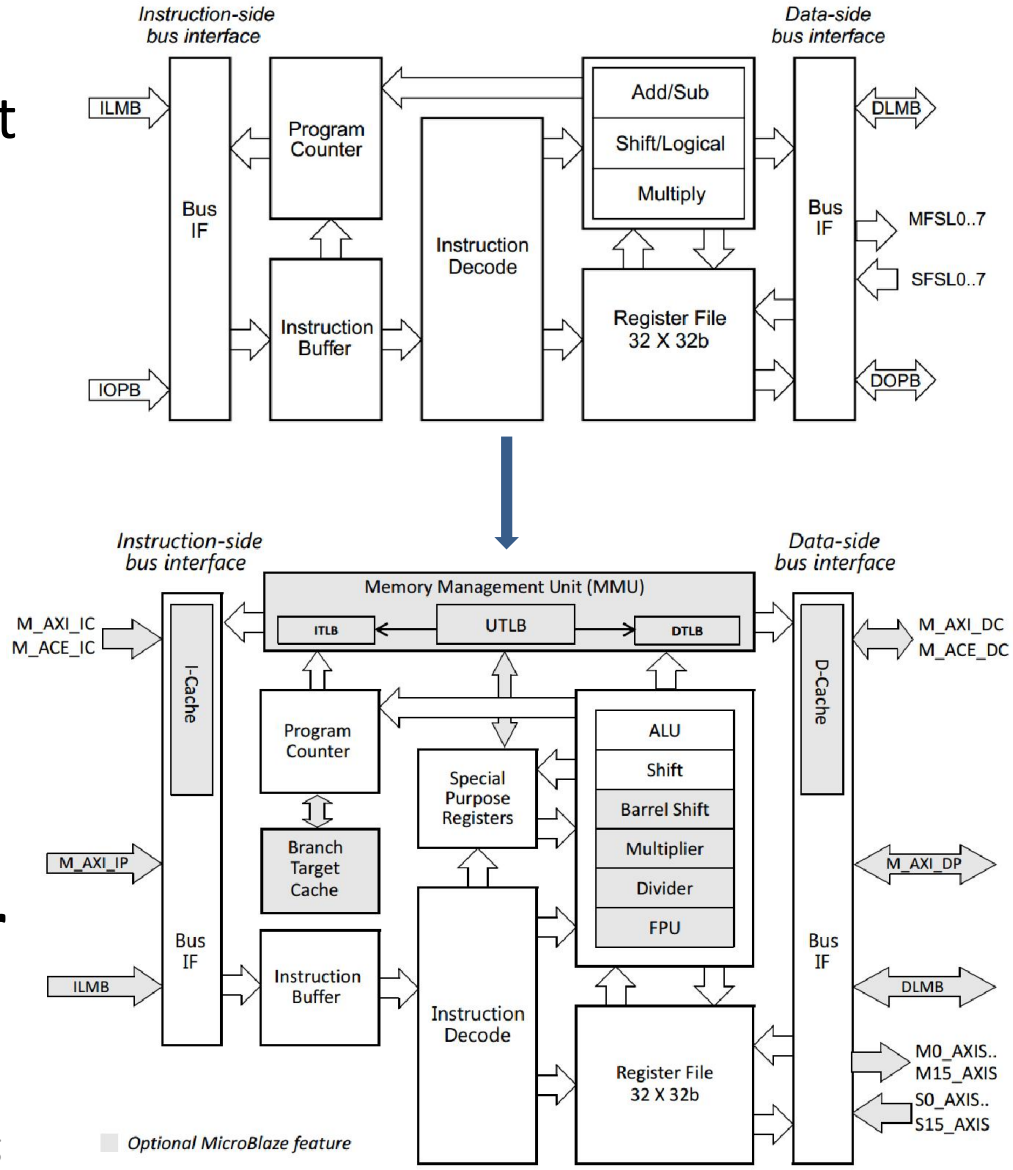
- Diverse 3rd party soft processors
  - Little MCUs – *KCPSM →PicoBlaze*
  - Commercial RISCs – *ARC, LEON SPARC*
  - Legacy ISAs – *6502, Z80, 68000*
  - Hobbyist / open source – *OpenRISC*
  - Language specific cores – *Java, Forth, Erlang*
  - Teaching – *Chalmers, Cornell, Georgia Tech, Hiroshima, Mich. State, NM Tech, Riverside, Tokai, UCSC, Valladolid, Virginia Tech, WUStL*
- **Nios, MicroBlaze: comprehensive SoC platforms**

2001-2014: MicroBlaze Evolution/Configurability

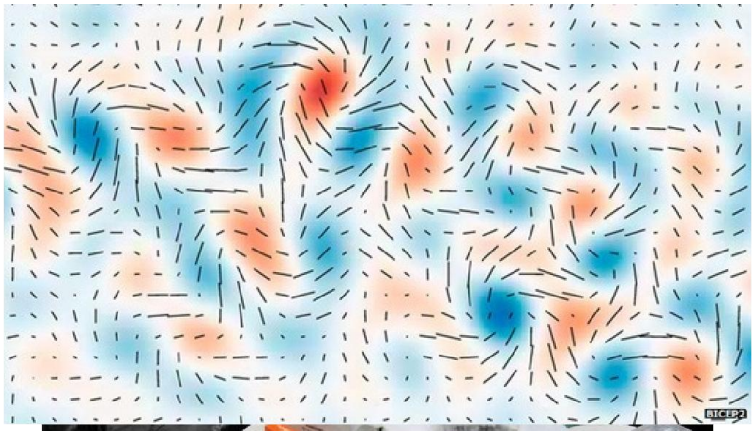
Version:

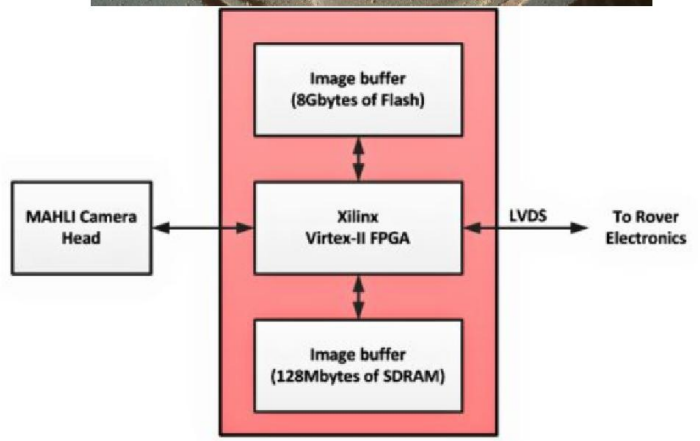
1. 3-stage, mul, bshift , CoreConnect
2. div, FSL, I$, D$, 150 MHz
   = 100 DMIPS
3. cache links
4. FPU, debug trace
5. 5-stage pipeline, **= 240 DMIPS**
6. 3/5-stage
7. MMU, exceptions, Linux
8. AXI4, fault tolerance, 330 MHz,
   = 400 DMIPS

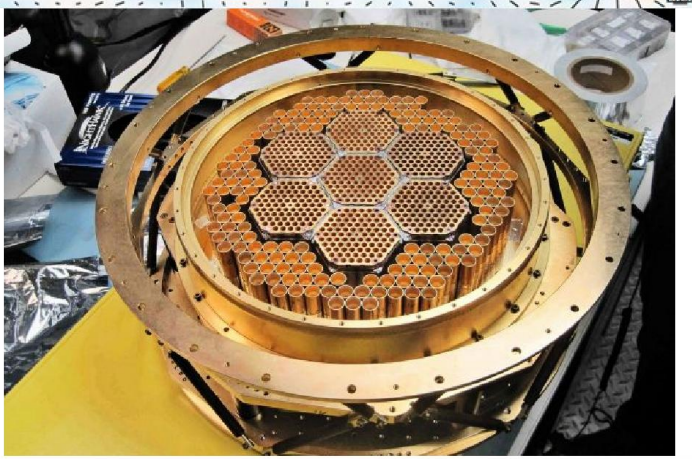
~4.4× faster / 12 years = **+13%/year**

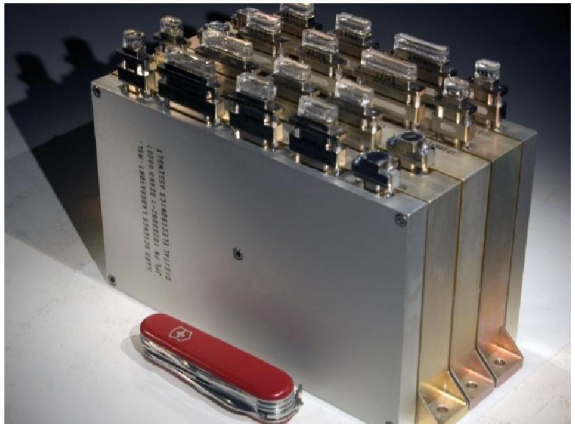


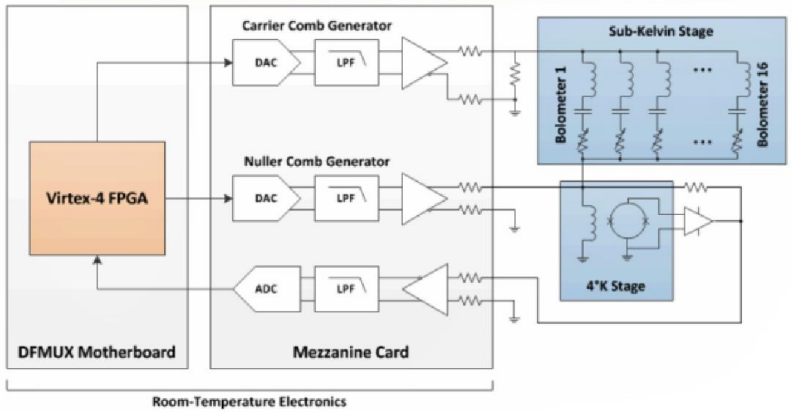
Remarkable (But Typical) Applications











http://forums.xilinx.com/t5/Xcell-Daily-Blog/Mars-Curiosity-Rover-s-MAHLI-images-a-dusty-penny-on-Mars-with/ba-p/369275

http://www-robotics.jpl.nasa.gov/publications/Reg_Willson/Edgett_etal_MAHLI_7Jul2012published.pdf

http://forums.xilinx.com/t5/Xcell-Daily-Blog/The-Search-for-Gravity-Waves-and-Dark-Energy-Gets-Help-from/ba-p/491048

The Utility of Soft Processors

- Run existing software, in the FPGA
  - Replace external MCU
  - Run RTOS / Linux / drivers / networking / web server
- Control plane
  - Replace complex state machines
  - Hardware boot, diagnostics, telemetry
- Accelerators
  - Customize with app-specific instruction sets
  - *Tightly* couple software to accelerators
- Computer architecture research
  - *RAMP*, CHERI, FlexPRET*

2. DOING IT “OLD SCHOOL”: AN AUSTERE APPROACH TO SOFT PROCESSOR DESIGN

Design Study: How Many Austere 32-bit RISCs Fit in a Modern FPGA?

Virtex-7 690T:
(433K 6-LUTs, 1470 BRAMs)
÷ (330 LUTs, 1 BRAM) = ?

- What about…?
  - 6-LUTs? DSPs?
  - NoC? Memory model?
- Let's see

XILINX®

7 Series FPGAs Overview

DS180 (v1.16) October 8, 2014 **Product Specification**

General Description

Xilinx® 7 series FPGAs comprise three new FPGA families that address the complete range of system requirements, ranging from low cost, small form factor, cost-sensitive, high-volume applications to ultra high-end connectivity bandwidth, logic capacity, and signal processing capability for the most demanding high-performance applications. The 7 series FPGAs include:

- Artix®-7 Family: Optimized for lowest cost and power with small form-factor packaging for the highest volume applications.
- Kintex®-7 Family: Optimized for best price-performance with a 2X improvement compared to previous generation, enabling a new class of FPGAs.
- Virtex®-7 Family: Optimized for highest system performance and capacity with a 2X improvement in system performance. Highest capability devices enabled by stacked silicon interconnect (SSI) technology.

Built on a state-of-the-art, high-performance, low-power (HPL), 28 nm, high-k metal gate (HKMG) process technology, 7 series FPGAs enable an unparalleled increase in system performance with 2.9 Tb/s of I/O bandwidth, 2 million logic cell capacity, and 5.3 TMAC/s DSP, while consuming 50% less power than previous generation devices to offer a fully programmable alternative to ASSPs and ASICs.

Summary of 7 Series FPGA Features

- Advanced high-performance FPGA logic based on real 6-input look-up table (LUT) technology configurable as distributed memory.
- 36 Kb dual-port block RAM with built-in FIFO logic for on-chip data buffering.
- High-performance SelectIO™ technology with support for DDR3 interfaces up to 1,866 Mb/s.
- High-speed serial connectivity with built-in multi-gigabit transceivers from 600 Mb/s to maximum rates of 6.6 Gb/s up to 28.05 Gb/s, offering a special low-power mode, optimized for chip-to-chip interfaces.
- A user configurable analog interface (XADC), incorporating dual 12-bit 1MSPS analog-to-digital converters with on-chip thermal and supply sensors.
- DSP slices with 25 x 18 multiplier, 48-bit accumulator, and pre-adder for high-performance filtering, including optimized symmetric
- Powerful clock management tiles (CMT), combining phase-locked loop (PLL) and mixed-mode clock manager (MMCM) blocks for high precision and low jitter.
- Integrated block for PCI Express® (PCIe), for up to x8 Gen3 Endpoint and Root Port designs.
- Wide variety of configuration options, including support for commodity memories, 256-bit AES encryption with HMAC/SHA-256 authentication, and built-in SEU detection and correction.
- Low-cost, wire-bond, lidless flip-chip, and high signal integrity flip-chip packaging offering easy migration between family members in the same package. All packages available in Pb-free and selected packages in Pb option.
- Designed for high performance and lowest power with 28 nm, HKMG, HPL process, 1.0V core voltage process technology and 0.9V core voltage option for even lower power.

Virtex-7 FPGA Feature Summary

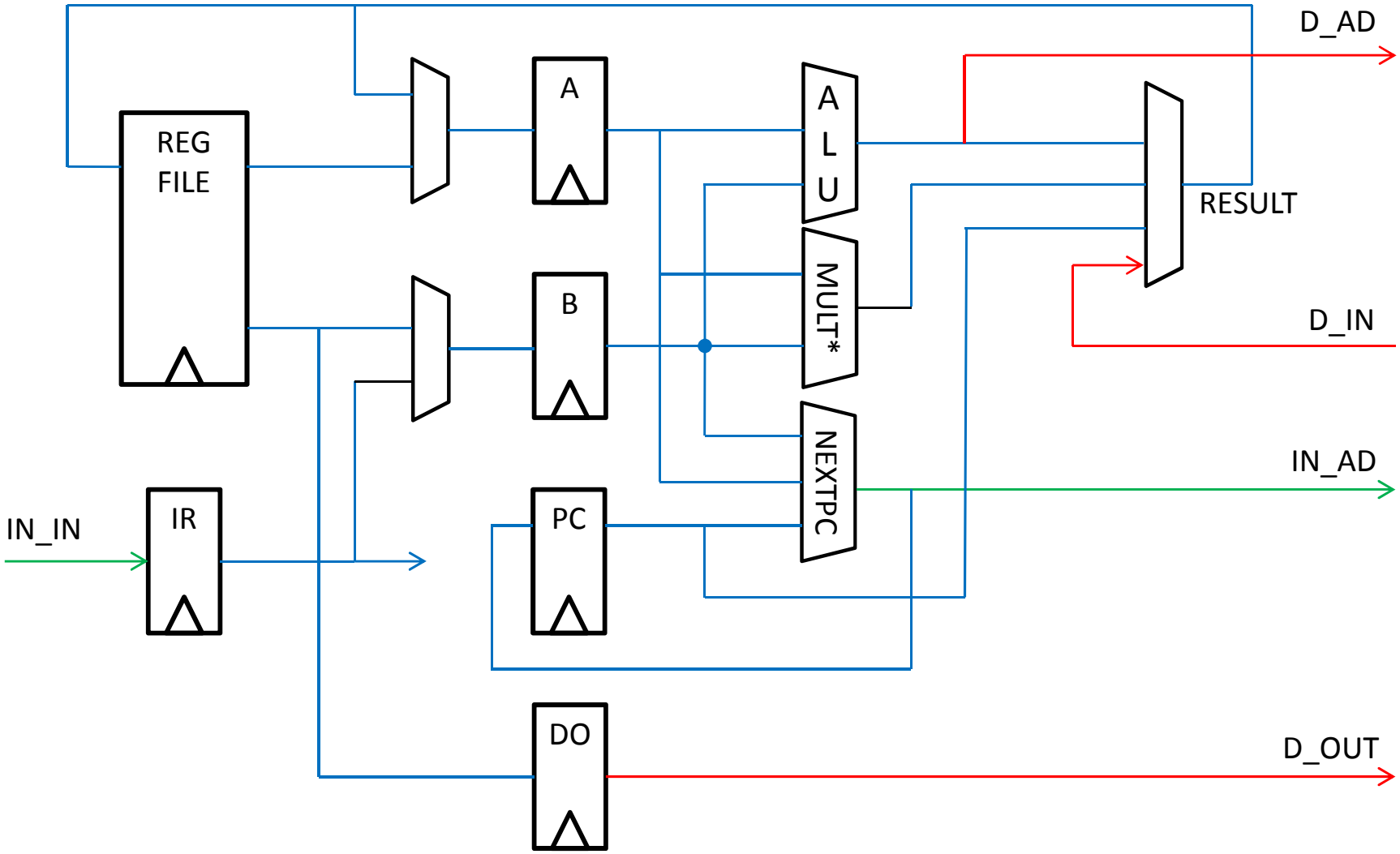
Table 6: **Virtex-7 FPGA Feature Summary**

Device[1]	Logic Cells	Configurable Logic Blocks (CLBs)		DSP Slices[3]	Block RAM Blocks[4]			CMTs[5]	PCIe[6]	GTX	GTH	GTZ	XADC Blocks	Total I/O Banks[7]	Max User I/O[8]	SLRs[9]
		Slices[2]	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)									
XC7V585T	582,720	91,050	6,938	1,260	1,590	795	28,620	18	3	36	0	0	1	17	850	N/A
XC7V2000T	1,954,560	305,400	21,550	2,160	2,584	1,292	46,512	24	4	36	0	0	1	24	1,200	4
XC7VX330T	326,400	51,000	4,388	1,120	1,500	750	27,000	14	2	0	28	0	1	14	700	N/A
XC7VX415T	412,160	64,400	6,525	2,160	1,760	880	31,680	12	2	0	48	0	1	12	600	N/A
XC7VX485T	485,760	75,900	8,175	2,800	2,060	1,030	37,080	14	4	56	0	0	1	14	700	N/A
XC7VX550T	554,240	86,600	8,725	2,880	2,360	1,180	42,480	20	2	0	80	0	1	16	600	N/A
XC7VX690T	693,120	108,300	10,888	3,600	2,940	1,470	52,920	20	3	0	80	0	1	20	1,000	N/A
XC7VX980T	979,200	153,000	13,838	3,600	3,000	1,500	54,000	18	3	0	72	0	1	18	900	N/A
XC7VX1140T	1,139,200	178,000	17,700	3,360	3,760	1,880	67,680	24	4	0	96	0	1	22	1,100	4
XC7VH580T	580,480	90,700	8,850	1,680	1,880	940	33,840	12	2	0	48	8	1	12	600	2
XC7VH870T	876,160	136,900	13,275	2,520	2,820	1,410	50,760	18	3	0	72	16	1	6	300	3

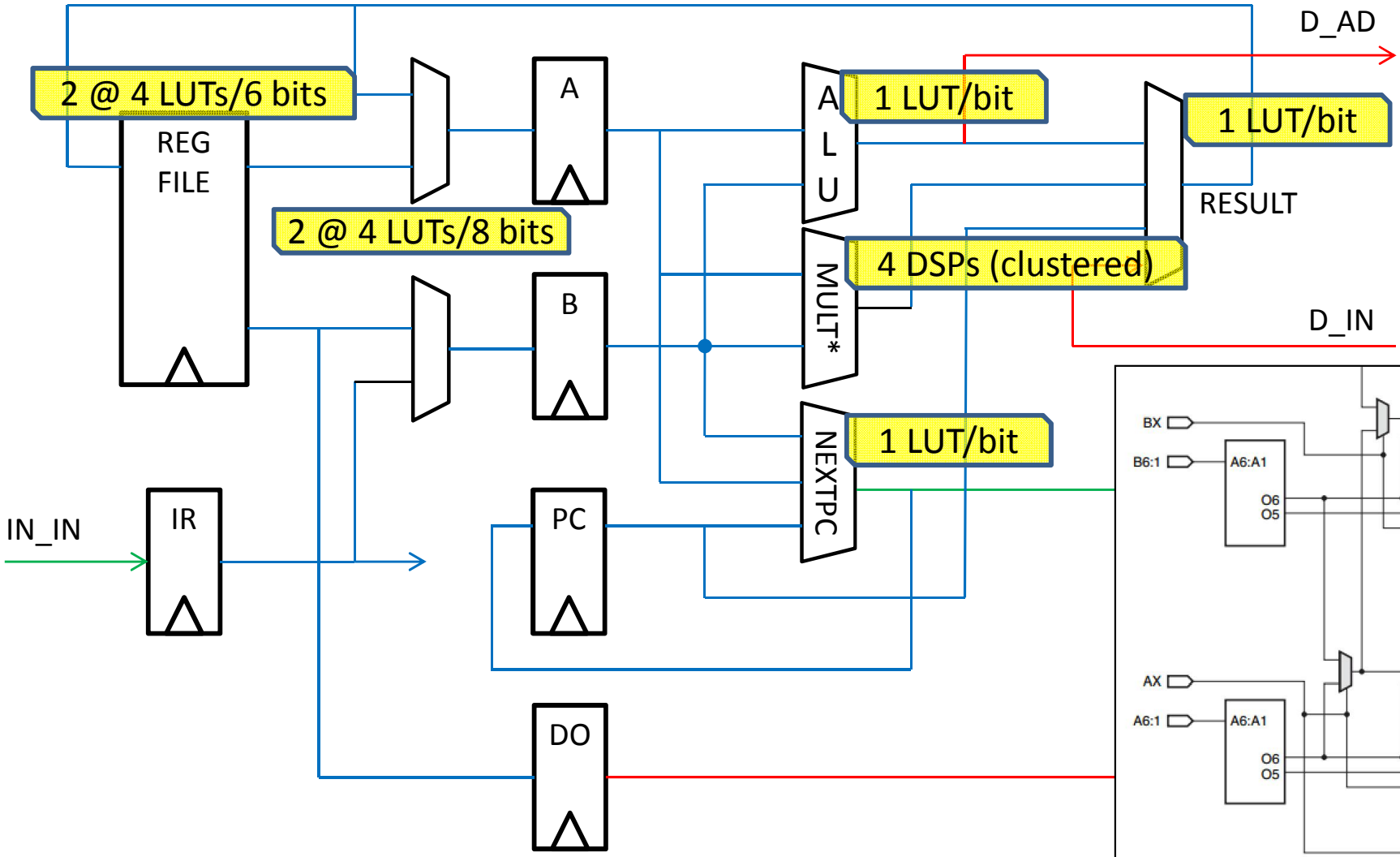
Planning an Austere RISC PE – *Psilos*

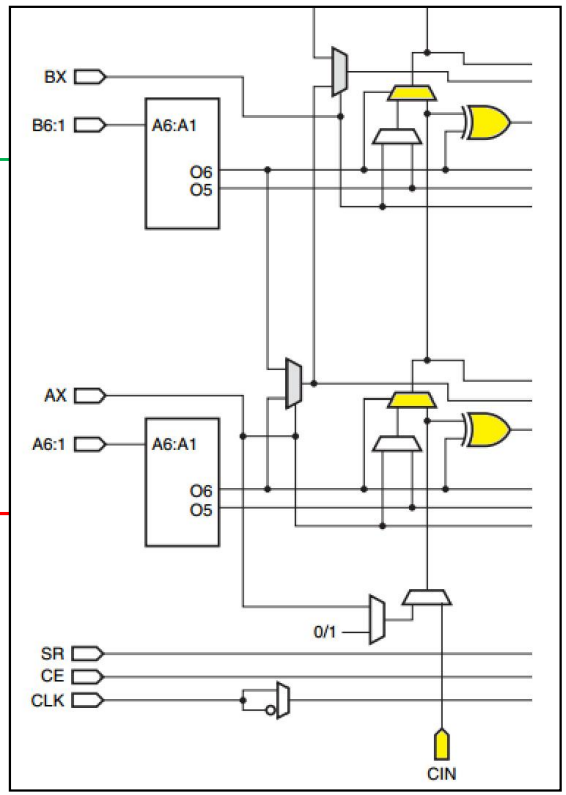
- Assumptions
  - Xilinx 6-LUTs
  - CMP *on-die* shared memory
  - Run small kernels (no I$)
  - MicroBlaze integer subset
- Essential: I-fetch, reg file, ALU, PC/branches, lw/sw
- Configurable: lb/sb, lh/sh
- Cluster-shared: mul ~~bshift~~
  - $A<<k = A \times 2^k$
    $A>>k = mulh(A \times 2^{32-k})$
- Pipeline design
  - Critical path: operand regs → ALU → mux → mux → operand regs
  - BRAM for instructions, data
  - IF:DC:EX~~:MEM~~
- 2-core clusters
  - Share 4 KB instruction BRAM
  - Share 4 KB local data BRAM
  - Share 32×32=64 mul
  - 2 BRAM → 10R×16C slices
  - 1 PE ≤ ½×10×16×4×¾ = 240 LUTs

RISC PE Datapath

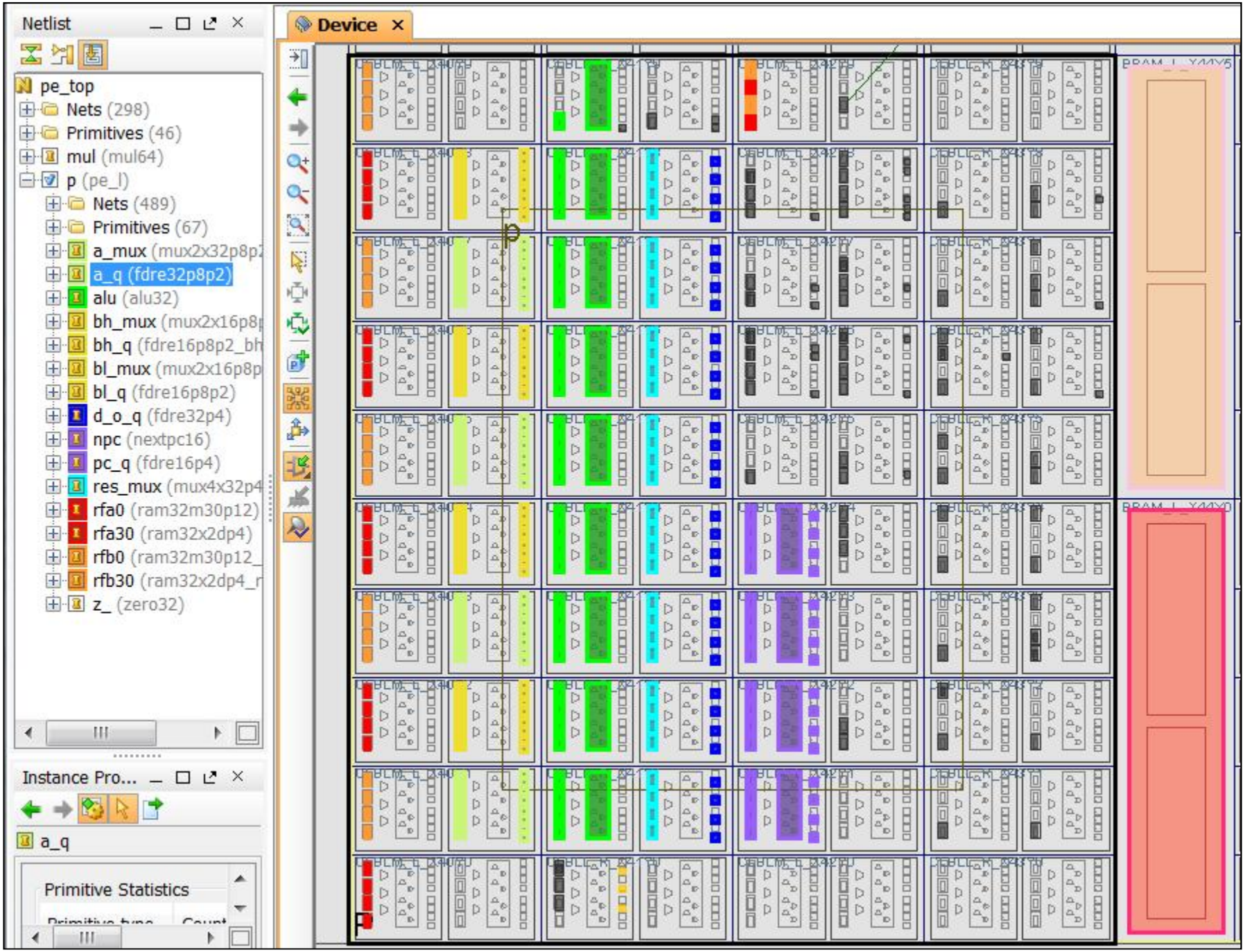


RISC PE Technology Mapping

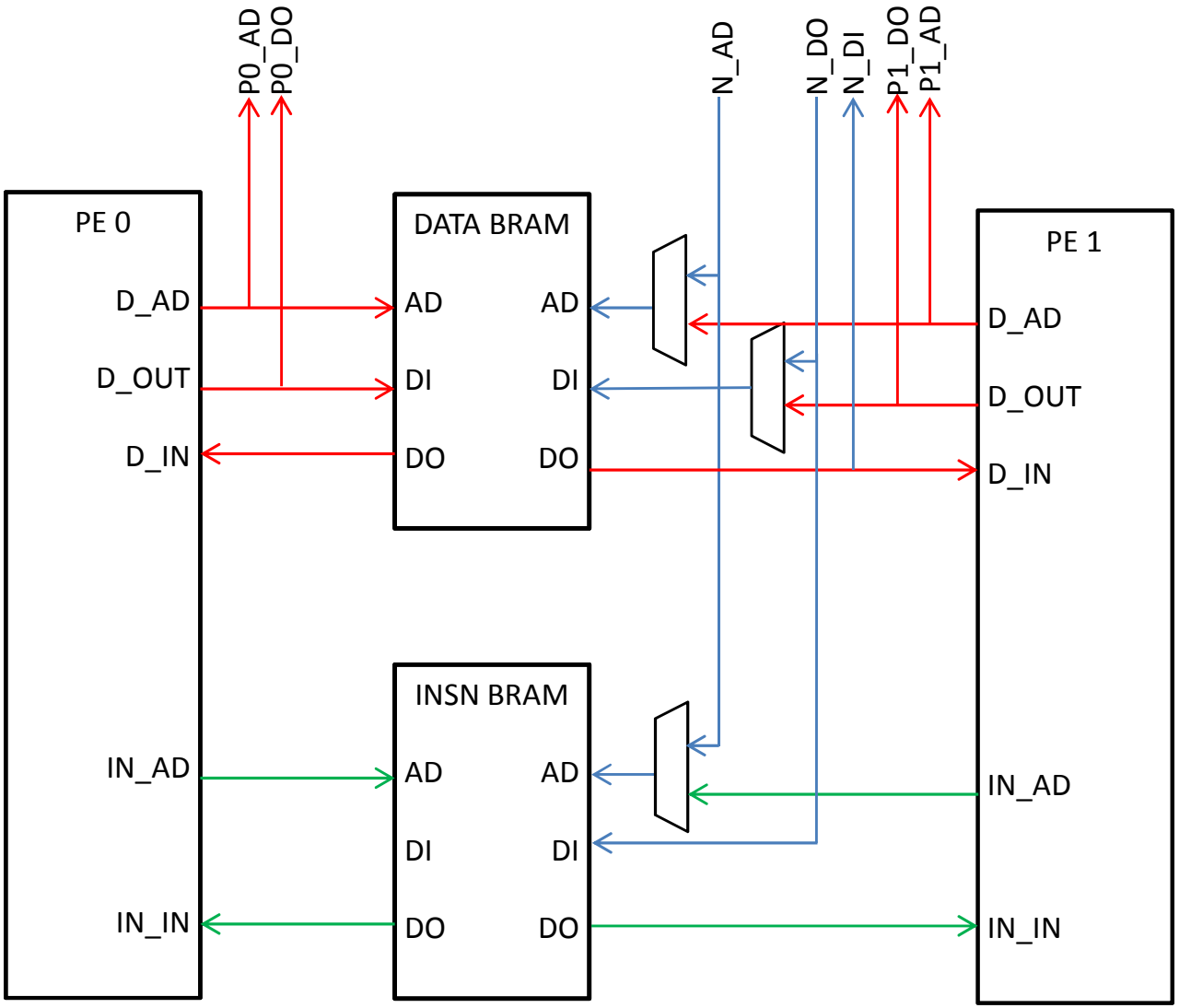




Floorplan: <200 6-LUTs, 300 MHz



2-PE Cluster



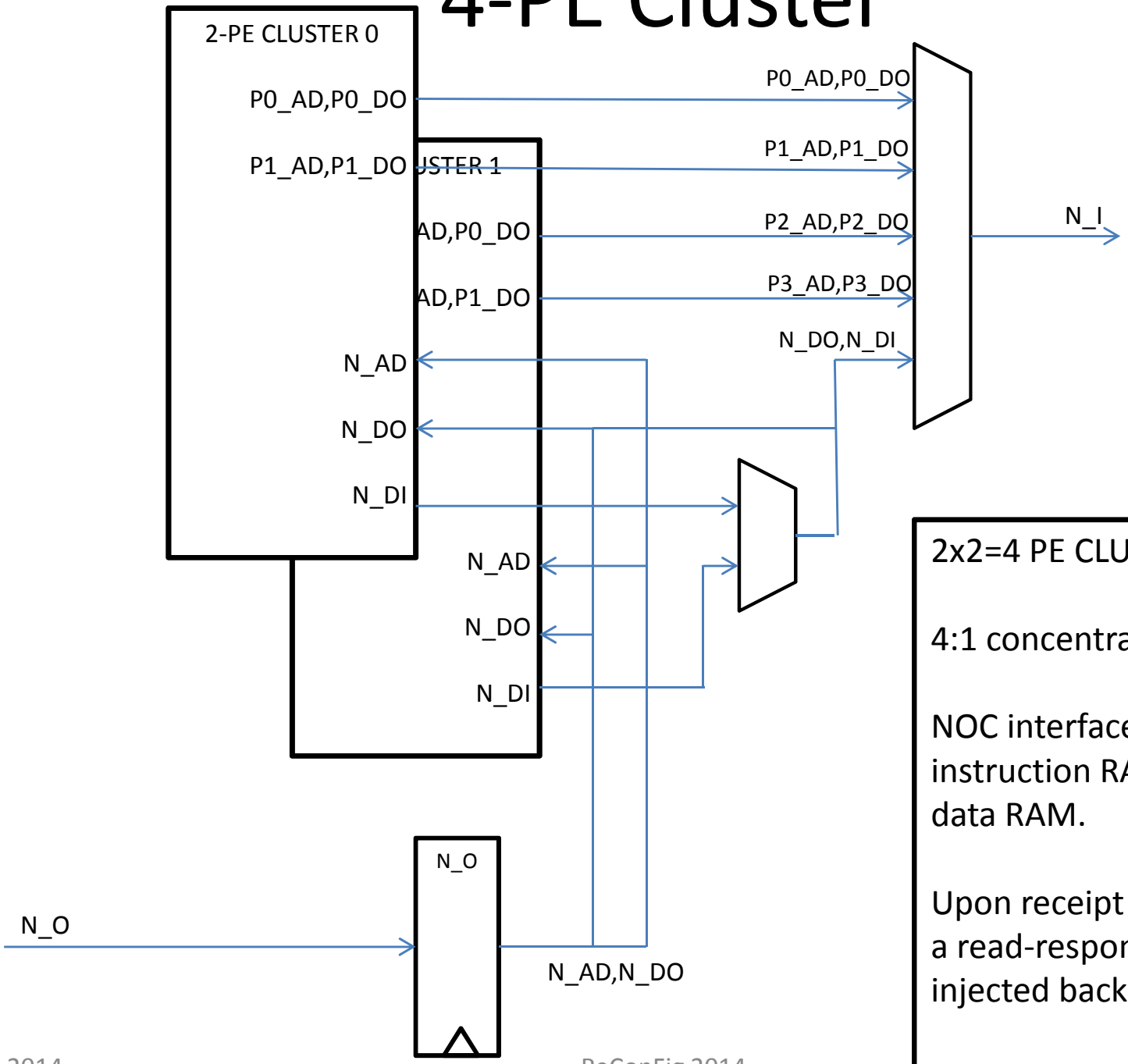
PEs share 4KB instruction RAM and 4 KB data RAM.

NOC interface can write instruction RAM and read/write data RAM.

PE 0 enjoys dedicated ports.
PE 1 shares with NOC.
Priority: NOC, PE 1.

Not shown: PEs 0 and 1 share the non-pipelined 32x32=64 multiplier.

4-PE Cluster



2x2=4 PE CLUSTER

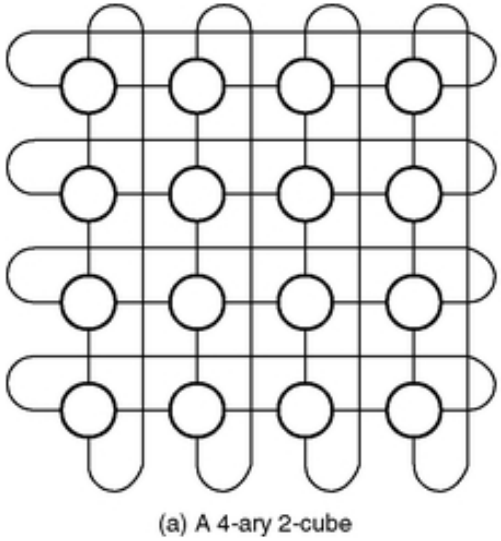
4:1 concentration of NOC accesses.

NOC interface can write instruction RAM and read/write data RAM.

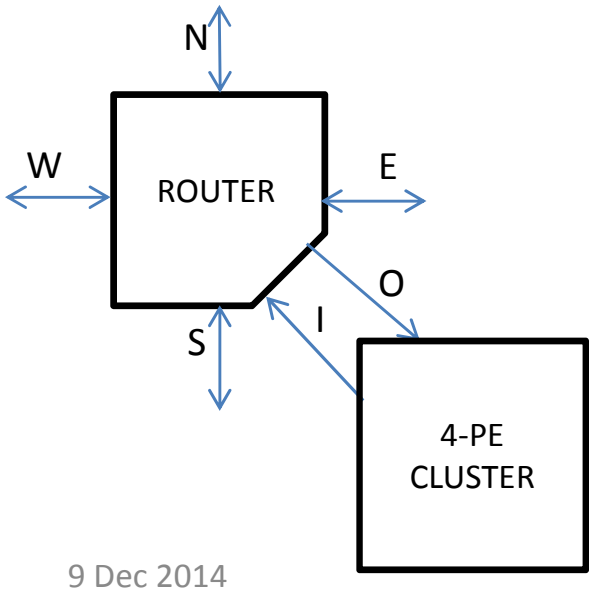
Upon receipt of a read-request, a read-response (e.g. write) is injected back into NOC.

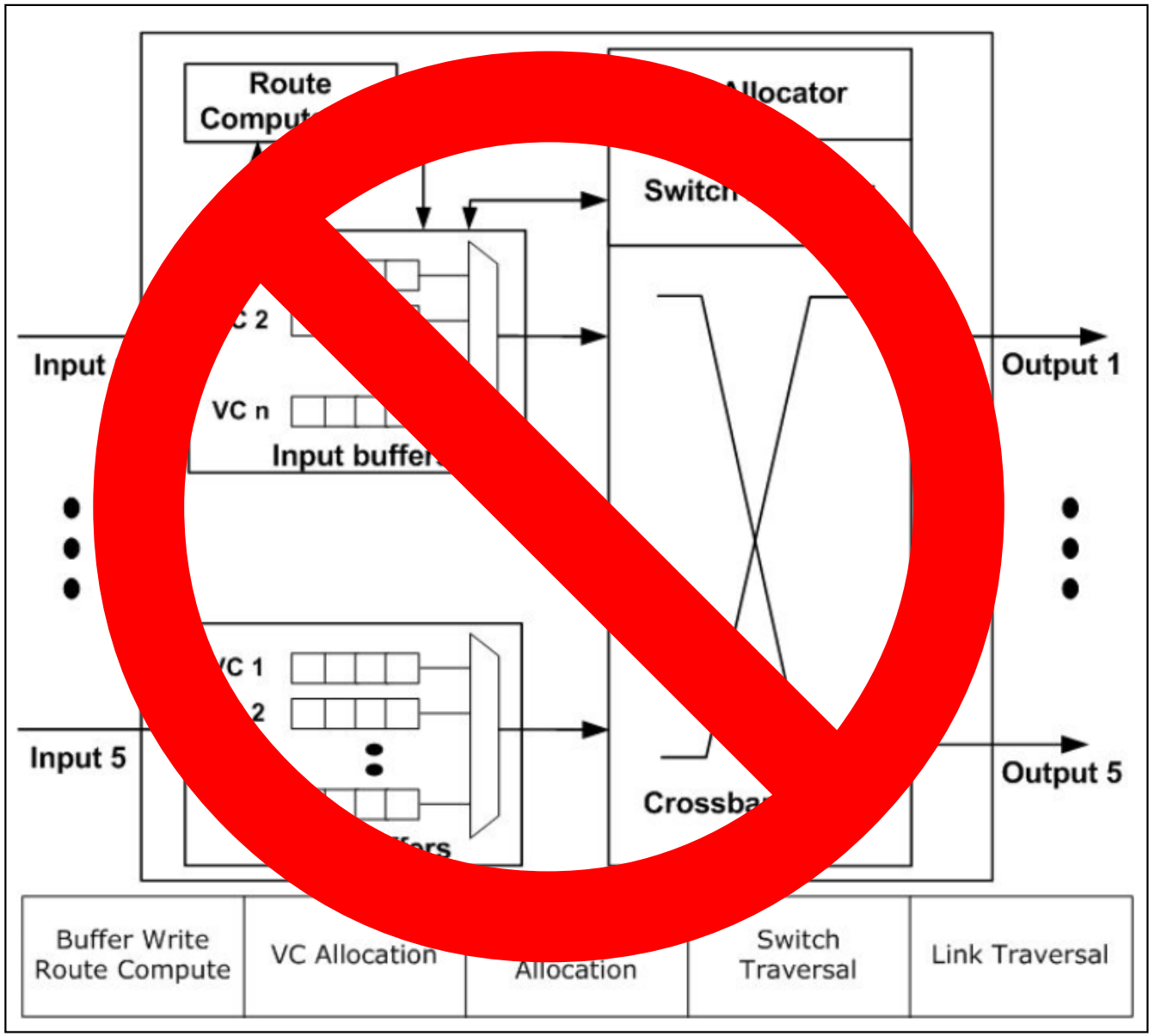
(Not shown: read-response FIFO.)

A 25R×10C 2-D Torus NoC Using 250 5-port Routers (!)

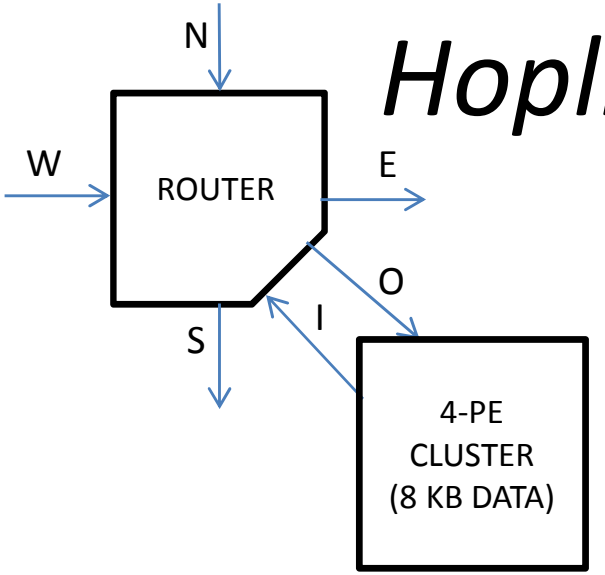


(a) A 4-ary 2-cube





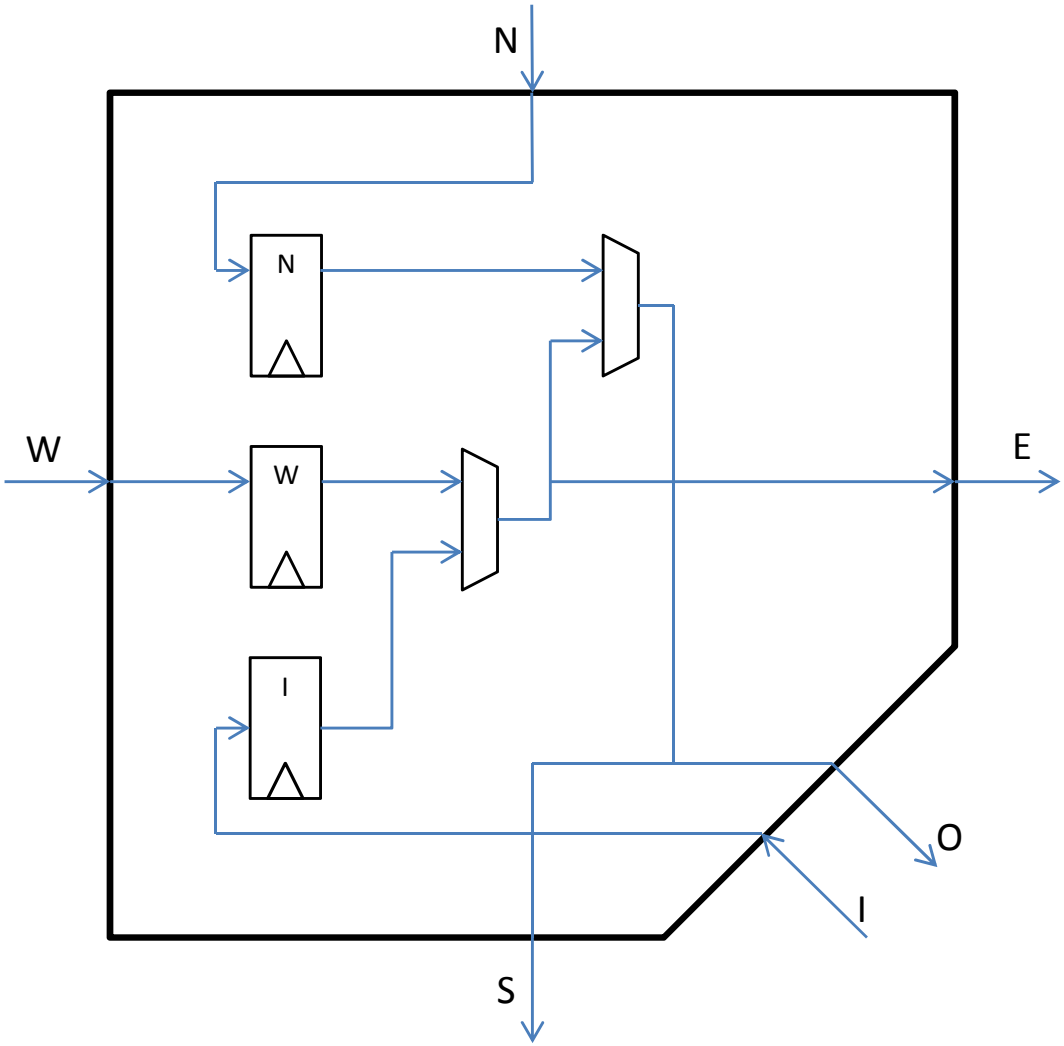
Hoplite Austere X:Y Router



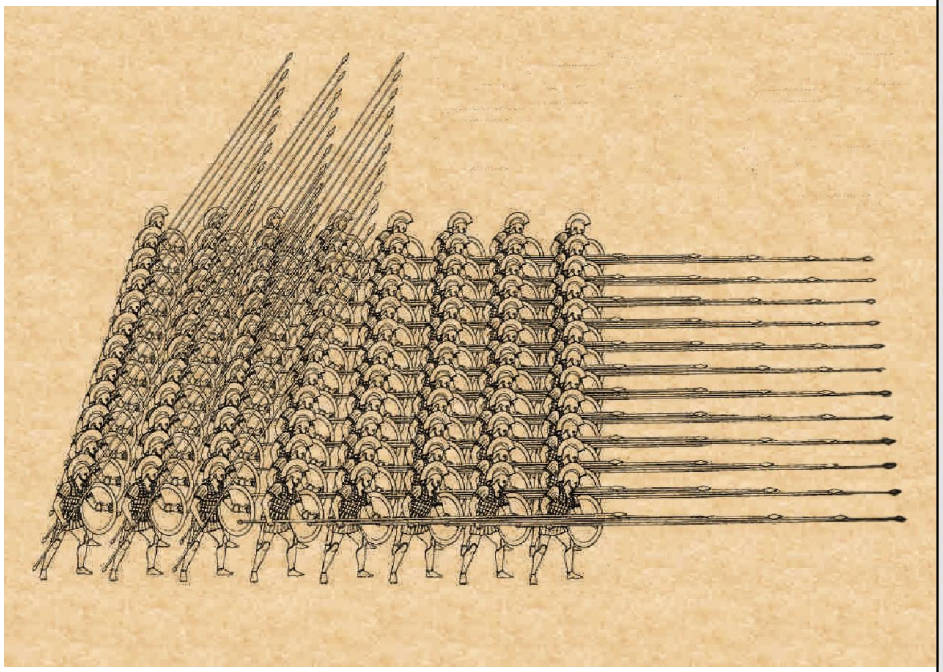
HOPLITE: AUSTERE X:Y ROUTER
(**Unidirectional** 2D Torus)

Wide. Fast. Simple. Tiny.
Bufferless, but no dropped traffic.
Dimension Ordered Routing: X>Y.
Priority: N, W, I.

No segmentation/reassembly.
No flits. No VCs. Simple flow control.
Grossly unfair, but deadlock-free.

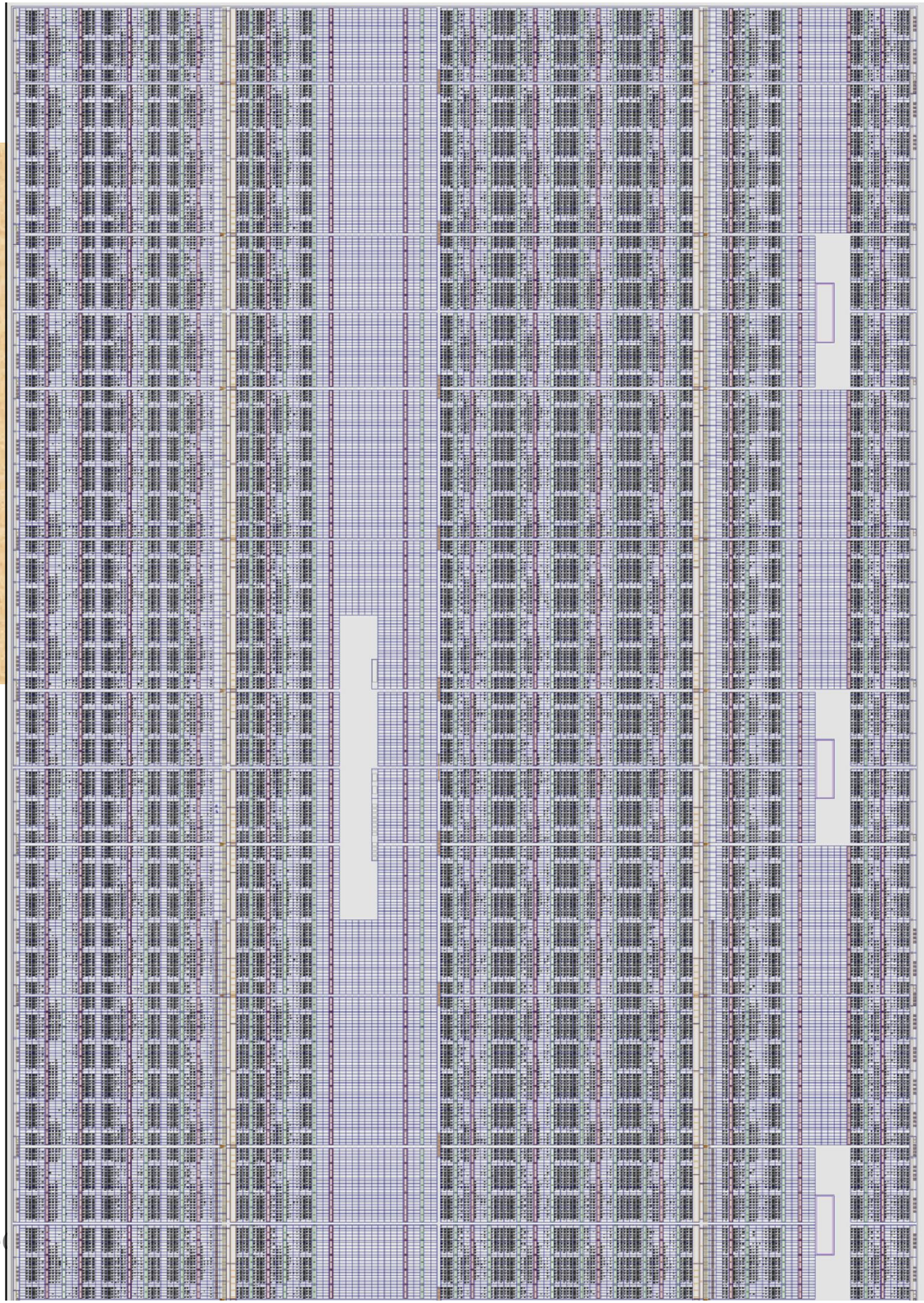


Phalanx



http://www.freewebs.com/apworldpolitics/phalanx%20facts%20picture.jpg

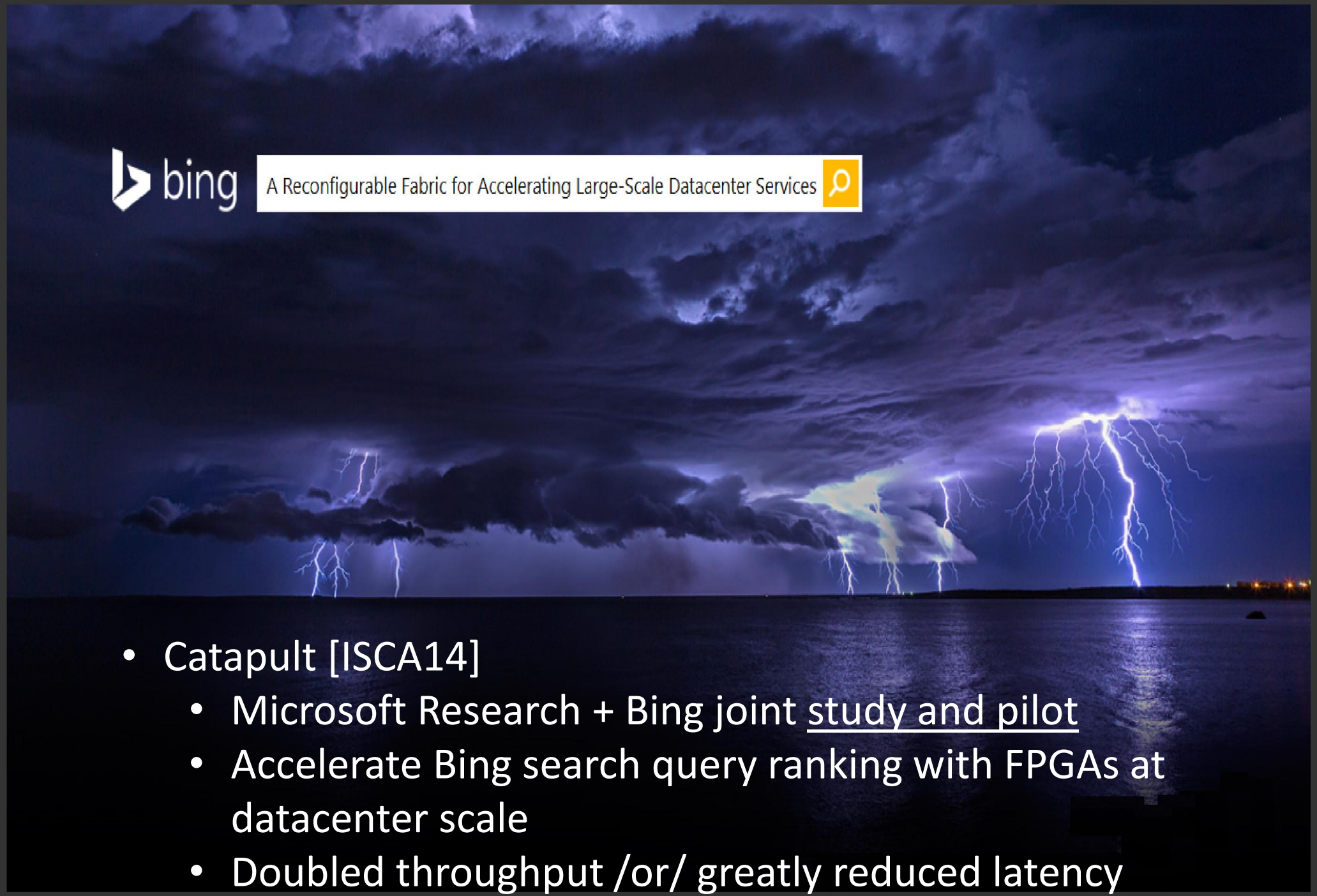
- 25R×10C×4-PEs + routers
  - *1000 PEs* / Virtex-7-690T
- OK, but …?
  - Latency?
  - External memory system?
  - Workloads?



3. THE FUTURE OF SOFT PROCESSORS

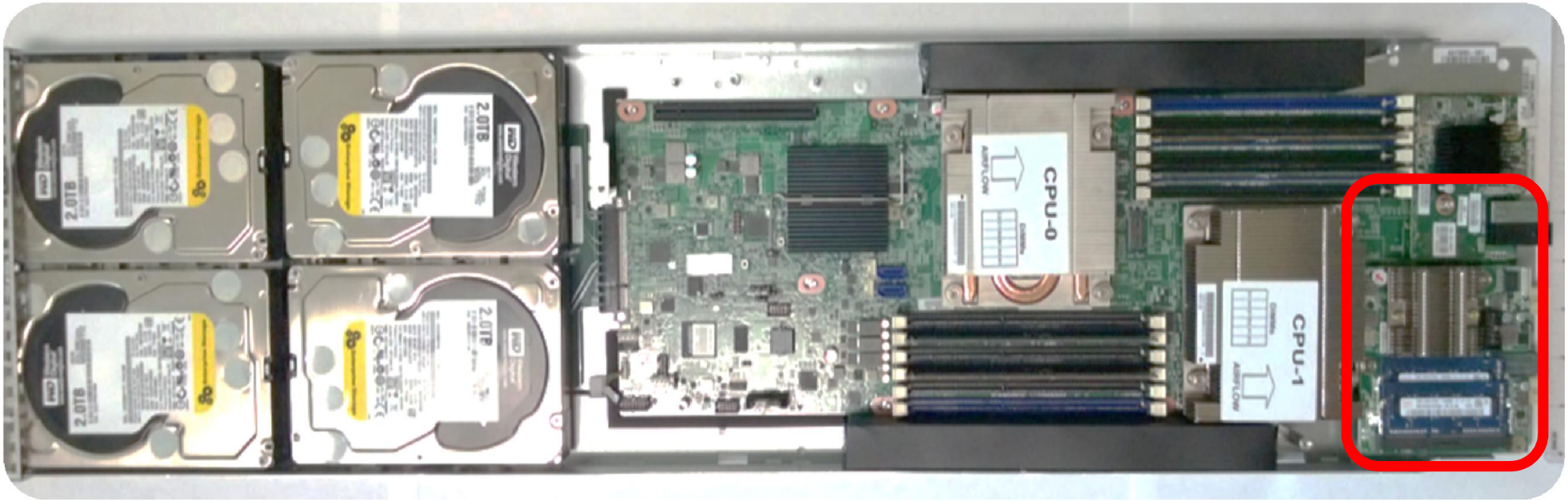
FPGAs: What's Next?

- Today
  - 20-28 nm low-power
  - 600 KLUTs, 1000s DSPs
  - 2.5D packaging
    - 1.2 MLUTs
    - Hetero: 28 Gbps serdes die
  - + 2×ARM A9s+SoC
    - ARM tools and IP
- *Soon? (speculative)*
  - 10-14 nm *lower-power* but ~~Dennard~~
  - 2+ MLUTs, 1000s DSP-FPUs, TFLOPS!
  - 2.5D packaging
    - 4+ MLUTs
    - + DRAM, 50+ Gbps serdes
  - "Datacenter edition"?
    - 4-8×64-bit ARM+SoC + DRAM + ?
  - Opportunities and challenges!



- Catapult [ISCA14]
  - Microsoft Research + Bing joint study and pilot
  - Accelerate Bing search query ranking with FPGAs at datacenter scale
  - Doubled throughput /or/ greatly reduced latency

Server Node += FPGA



Air flow

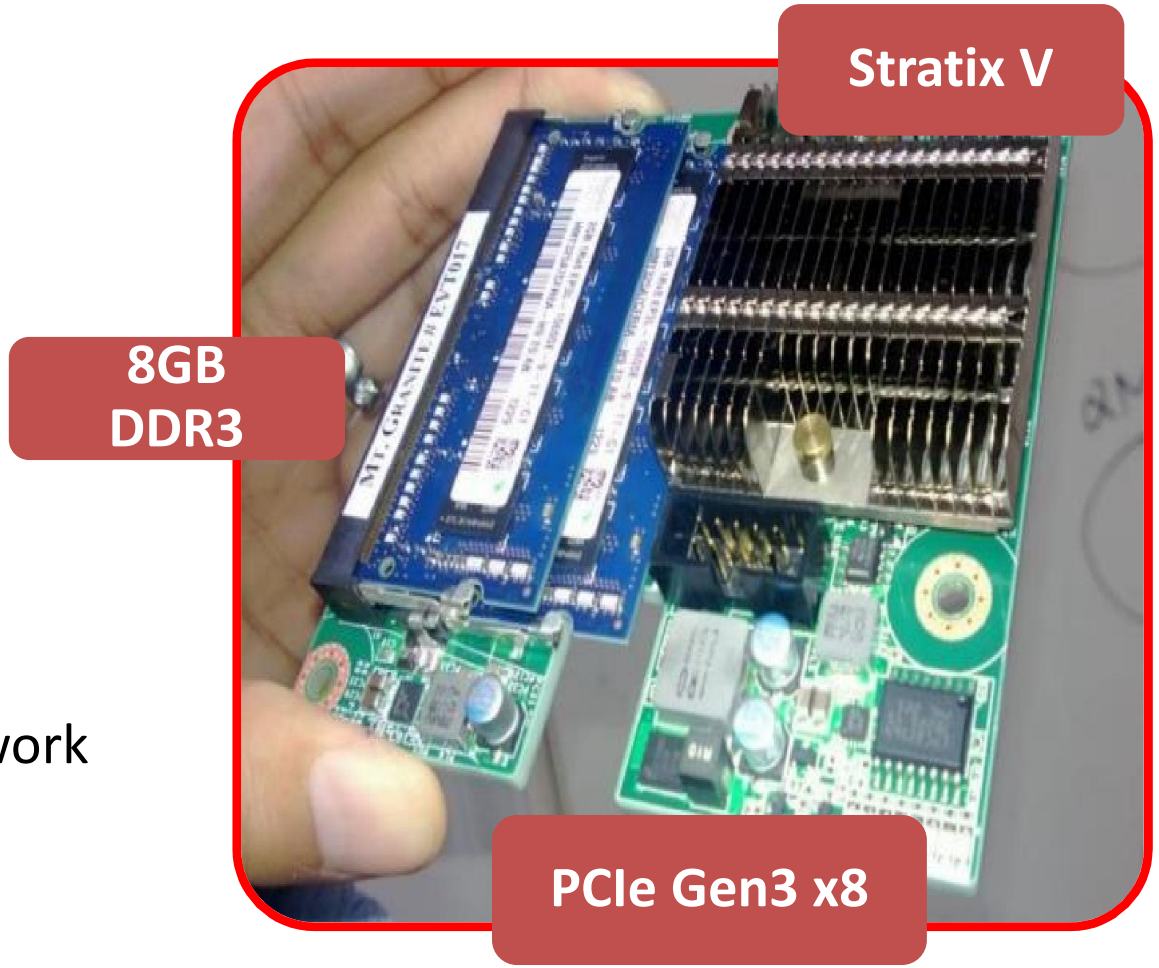
Two 8-core Xeon 2.1 GHz CPUs
64 GB DRAM
4 HDDs, 2 SSDs
10 Gb Ethernet

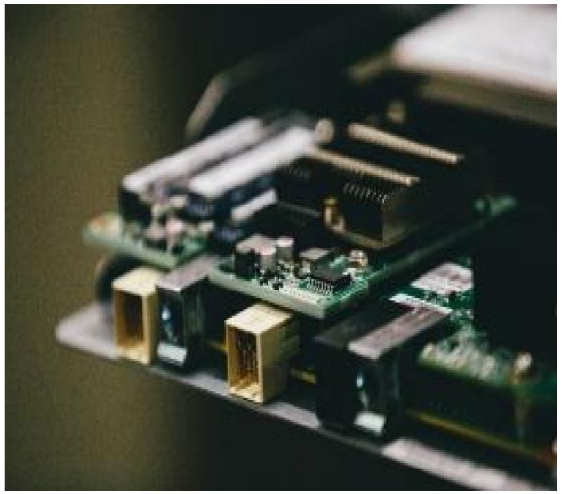
Catapult FPGA Accelerator Card

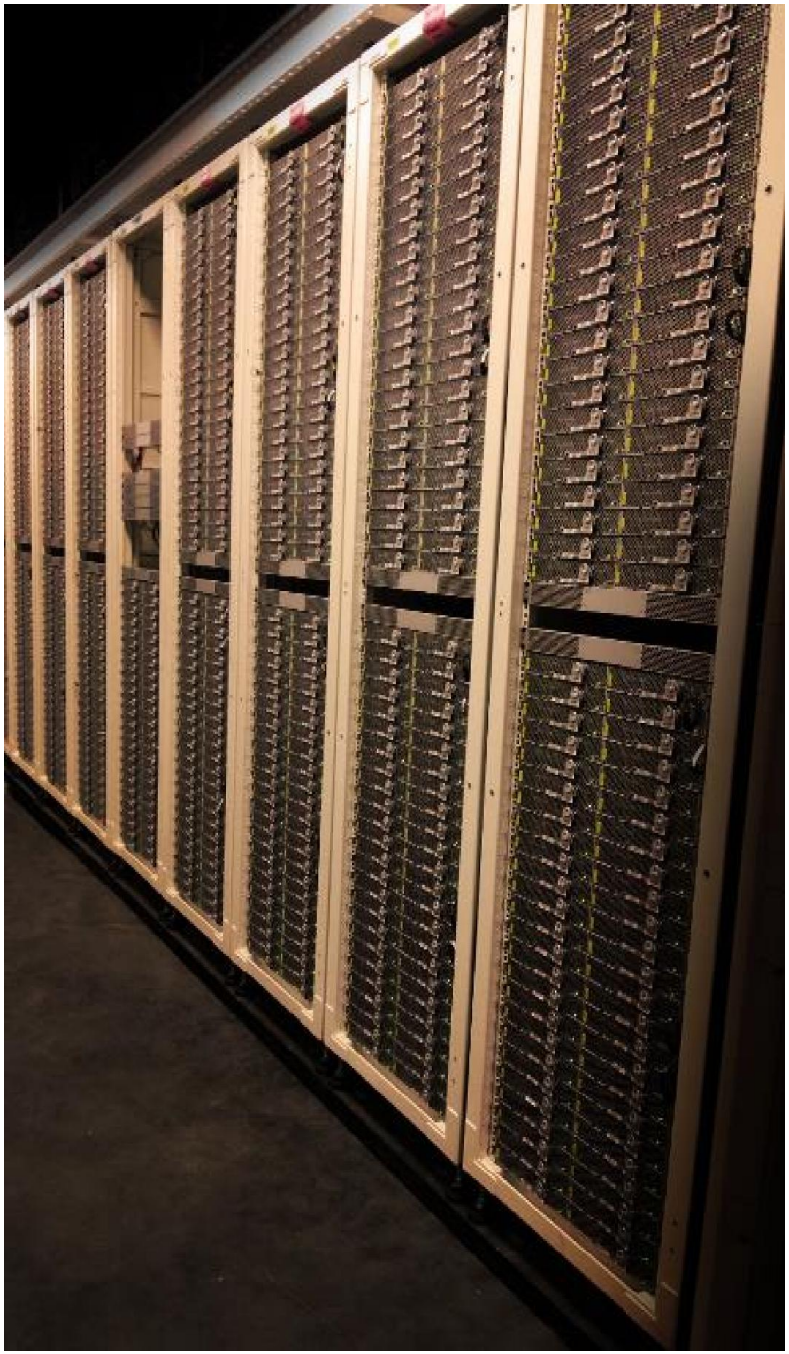
Altera Stratix V D5:
172,600 ALMs,
2,014 M20Ks,
1,590 DSPs

PCIe Gen 3 x8
8GB DDR3-1333
Powered by PCIe slot

FPGA – FPGA: torus network

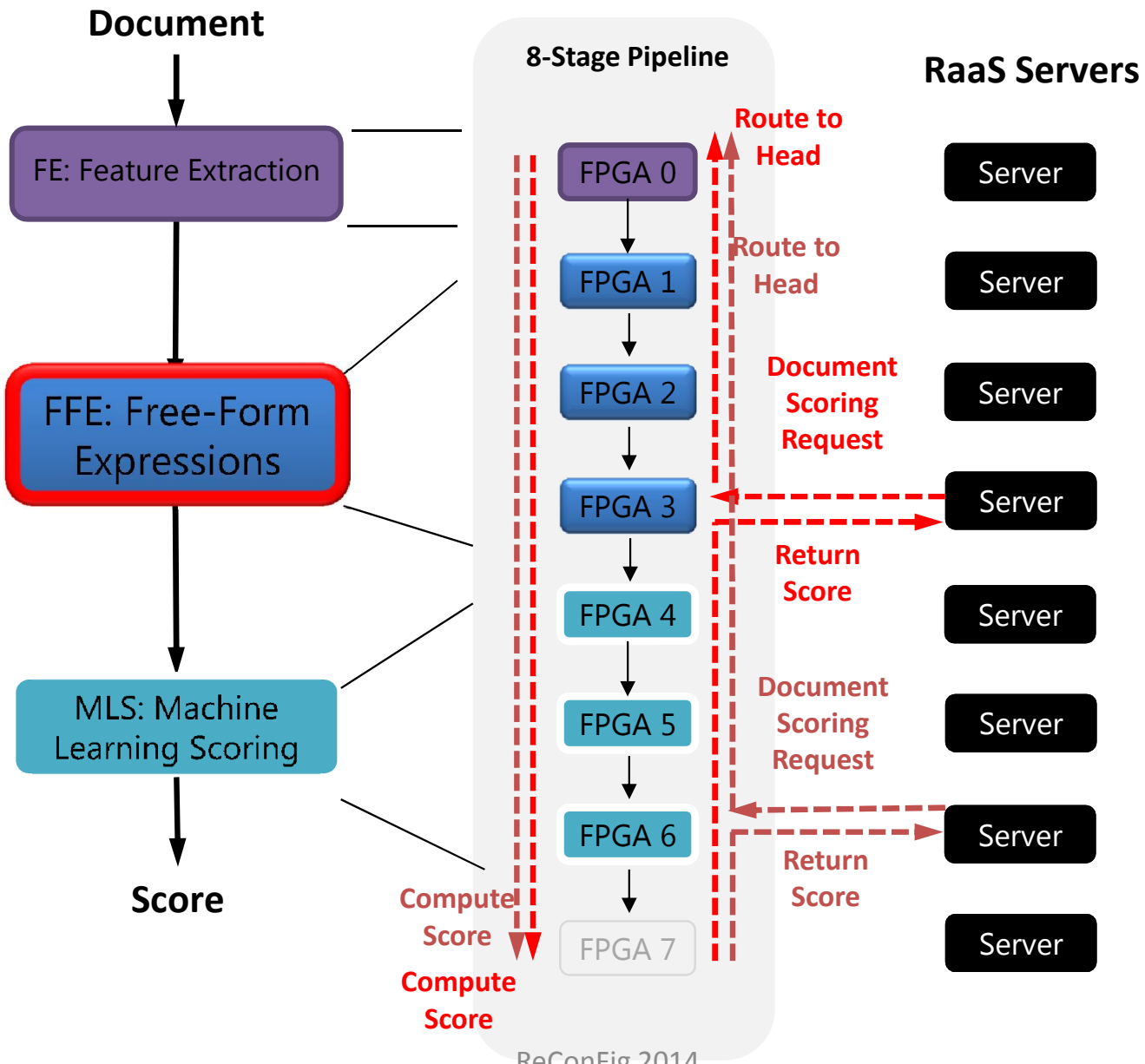




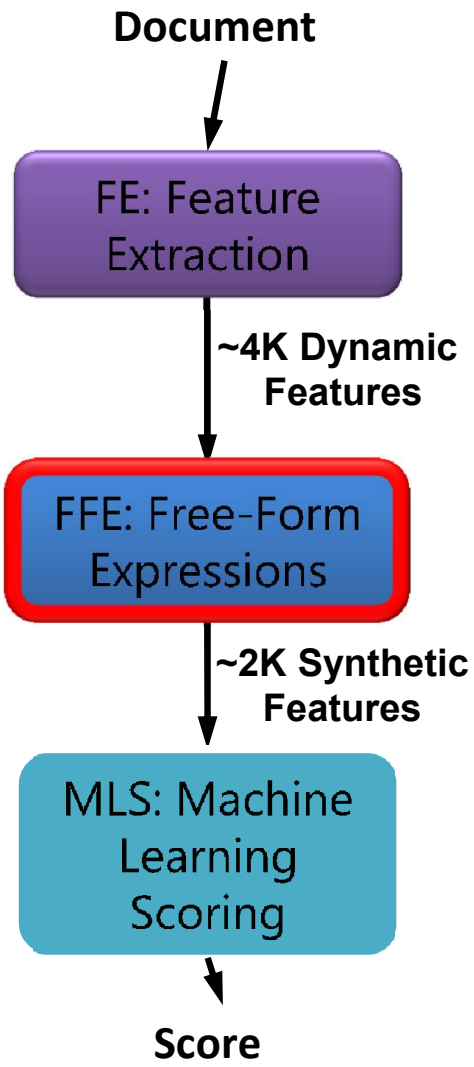


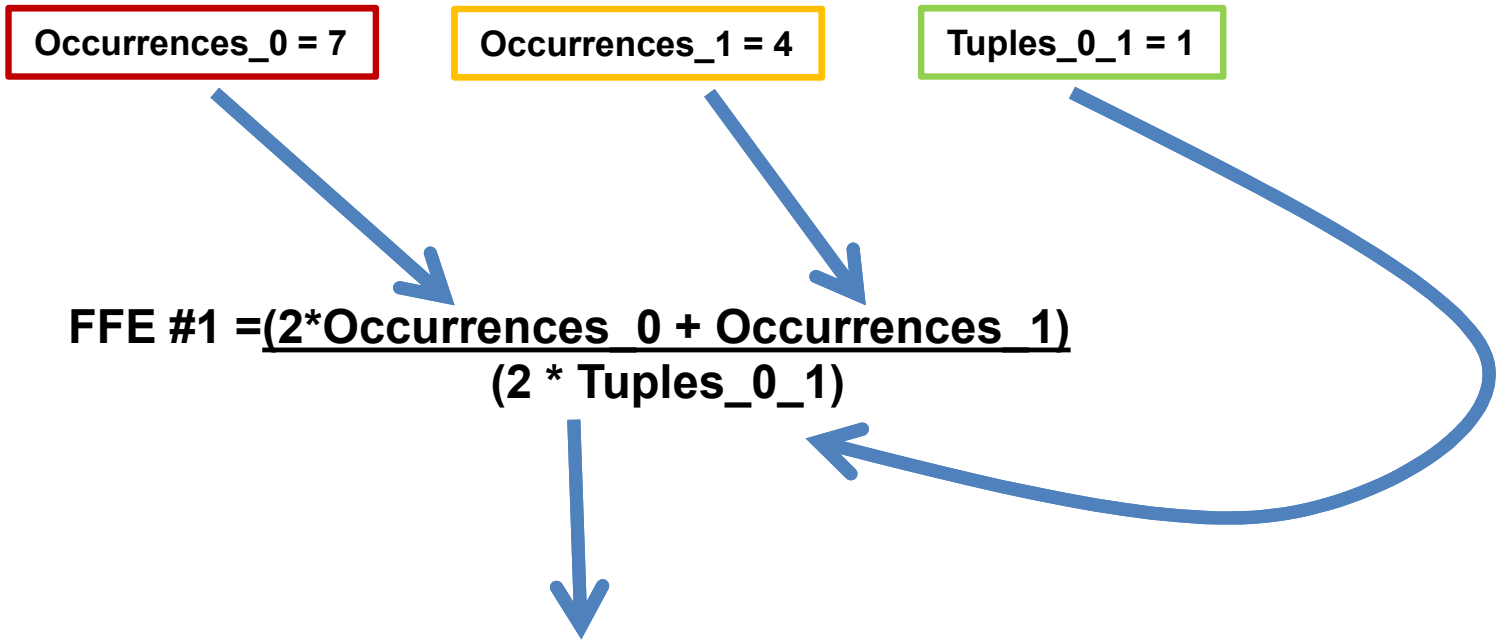
1,632 Server Pilot Deployed in a Production Datacenter

FPGA Accelerator for Search Ranking



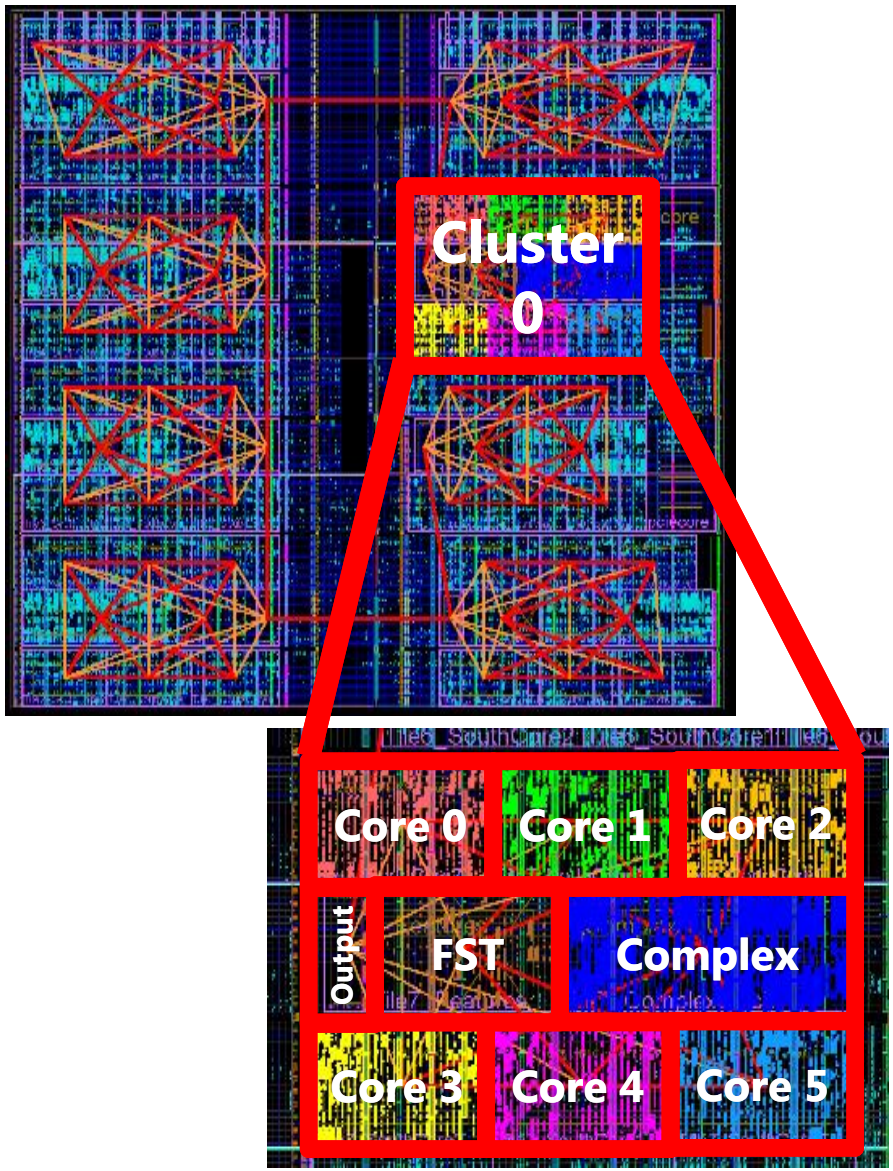
FFE: Free Form Expressions





FFE Soft Processor Arrays

- Many FFE "programs" → soft processor approach
- 4 hardware threads/core
- 6 cores/cluster
  - –Shared input vector
  - –Shared pipelined FPUs
- 10 clusters/FPGA
  - –60 cores, 240 threads
- C++ compiler



The Design Productivity Challenge

- Catapult Bing ranking: **~20 KLOCs** C++ → FPGAs

- RTL design productivity << software development
  - Smaller talent pool, weak tools, bespoke, fragile
- Missing essentials
  - Abstraction builders: languages, types, libraries, services, OS
  - Reuse, composability, portability, longevity

- Often the workload *is* software already, changes often
  - Expensive to port *and maintain*

Making HW Dev More Like SW

- Vivado HLS → one core
- Altera and Xilinx OpenCL tools!
  - Compile kernels to parallel FPGA datapaths + FSMs
  - Manage kernel invocations, memory traffic
  - Debug on host, cycle accurate model, FPGA
- Shortcomings
  - Hours per design spin
  - Specific parallel programming models only
  - 0% of software is OpenCL today

New Applications of Soft Processors?

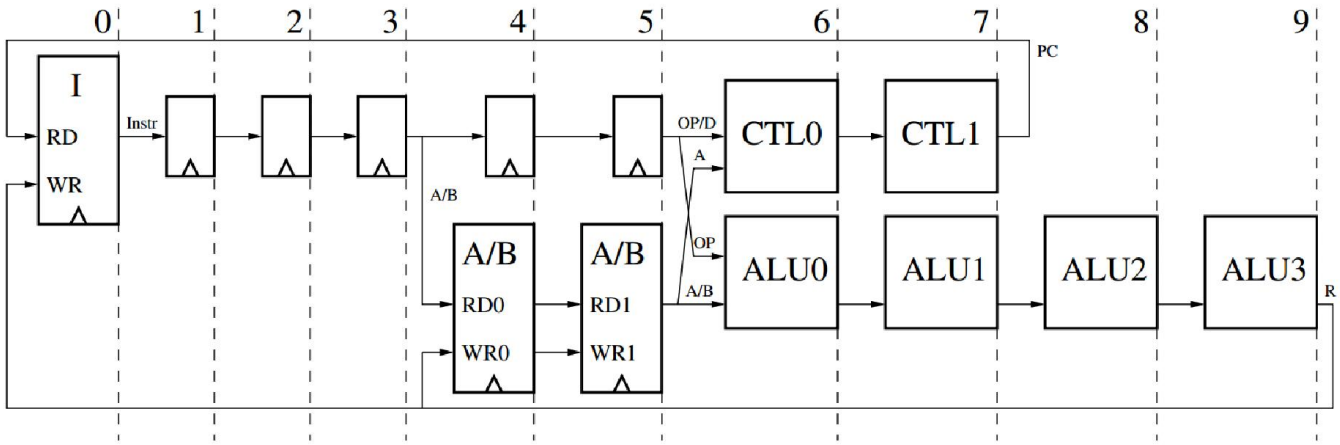
- Soft processor **array** overlays as OpenCL targets
- Recompile *parallel* host workloads to FPGA, unchanged, tightly coupled to accelerators

Some Parallel Programming Models and Soft Processor Array Classes

- Process networks and messages
  - CPUs+channels – *[Ambric]*
- Linear algebra
  - vector DSP/FPU – *VectorBlox MXP*
- Latency tolerant, task/data parallel
  - MIMD/SIMT– ***[MTA-GPU]***
- "Amdahl's Law" bottlenecks
  - Out-of-order superscalars **– *MSR EDGE research***

MIMD: Fine-Grained Multithreaded PEs

- Latency tolerant, high throughput, **task** parallel
- Many hardware thread contexts
  - LUT RAM PC register file, BRAM general register file
  - Fast, simple, deep, hazard free pipeline
  - HW thread scheduling
  - *FlexPRET*
  - *Octavo* [FPGA'12] Fmax = 550 MHz



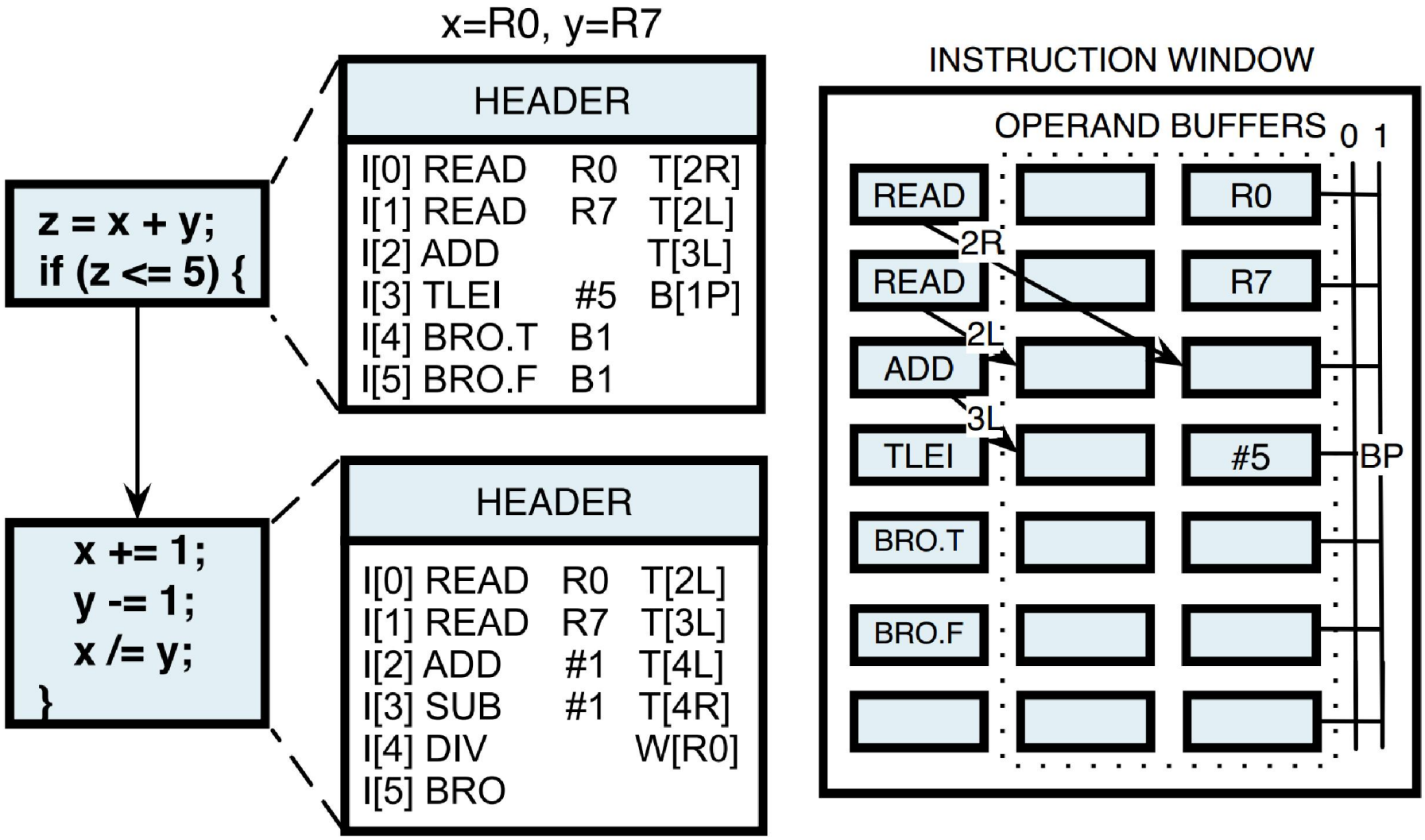
MTA-GPU: A MIMD/SIMT Hybrid

- GPU SIMT: **data** parallel
- MIMD ∩ SIMT = ?
  - Both: many register files, deep hazard-free pipelines
  - MIMD: *Tera MTA* – switch threads each cycle
  - SIMT: *GPU* – spatial issue instruction over warp of PEs
  - Let's do a time warp instead
- Hybrid: MTA + SIMT { time warps + predication }
- Multi-paradigm – a nice OpenCL target

Amdahl's Law?
Single Threaded Performance

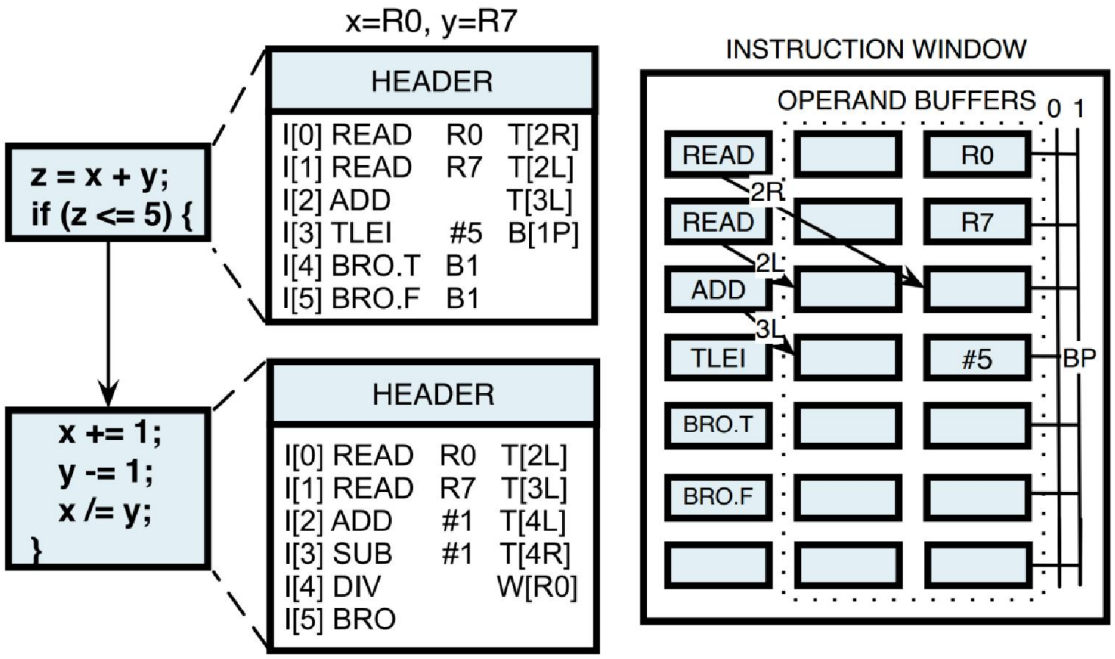
- In-order scalar RISCs *still* dominate
- Higher ILP via out-of-order superscalar?
  - Complexity, renaming, scheduling, recovery → many ported RAMs, CAMs → huge → slow
- Idea: *EDGE* (explicit data graph execution) ISA
  - ISA + compiler banish renaming, CAMs, complexity
  - Better? → MSR **research project**

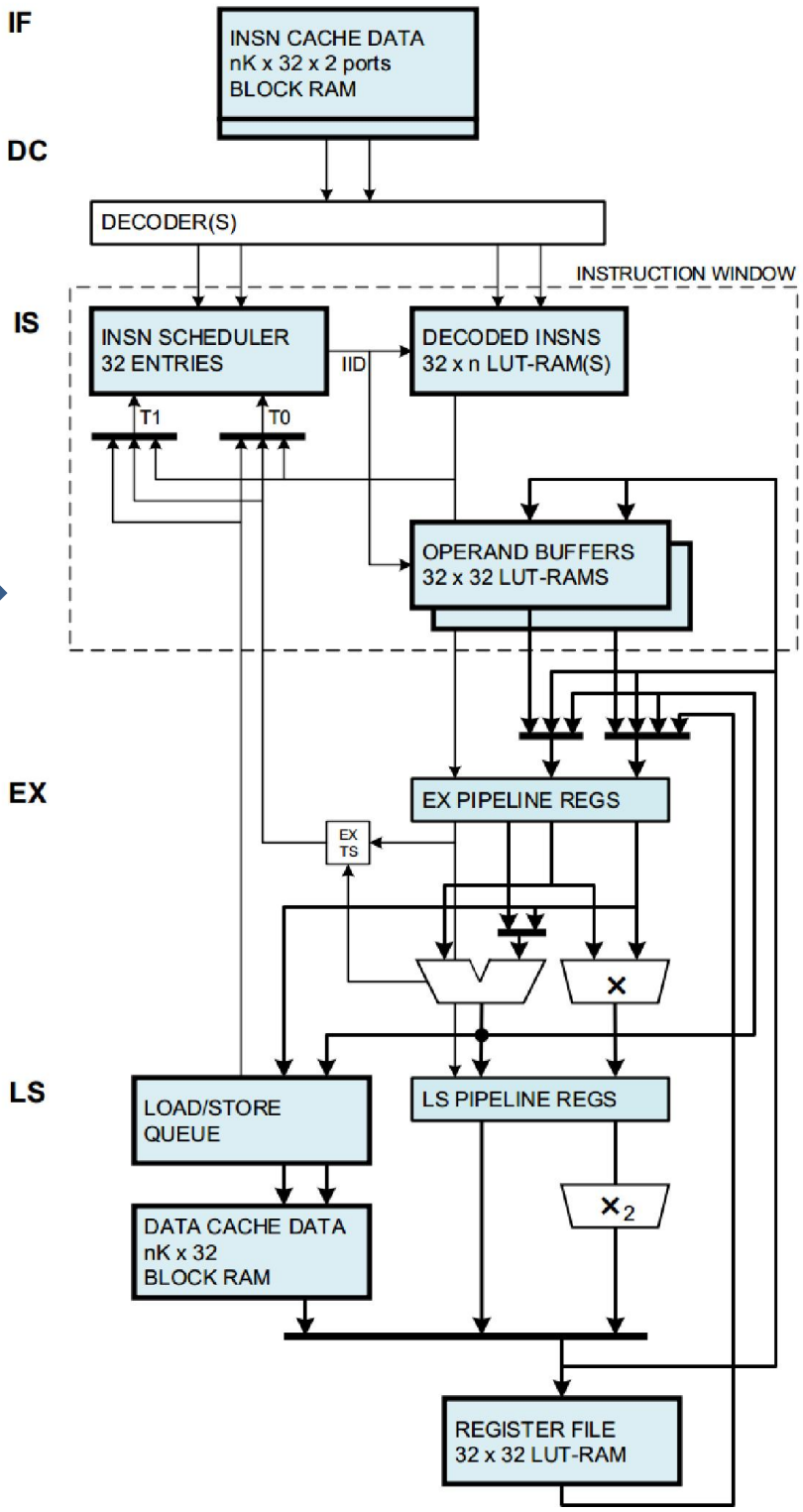
EDGE Block Structured ISA



PR=PREDICATE BID=BROADCAST ID XOP=EXTENDED OPCODE

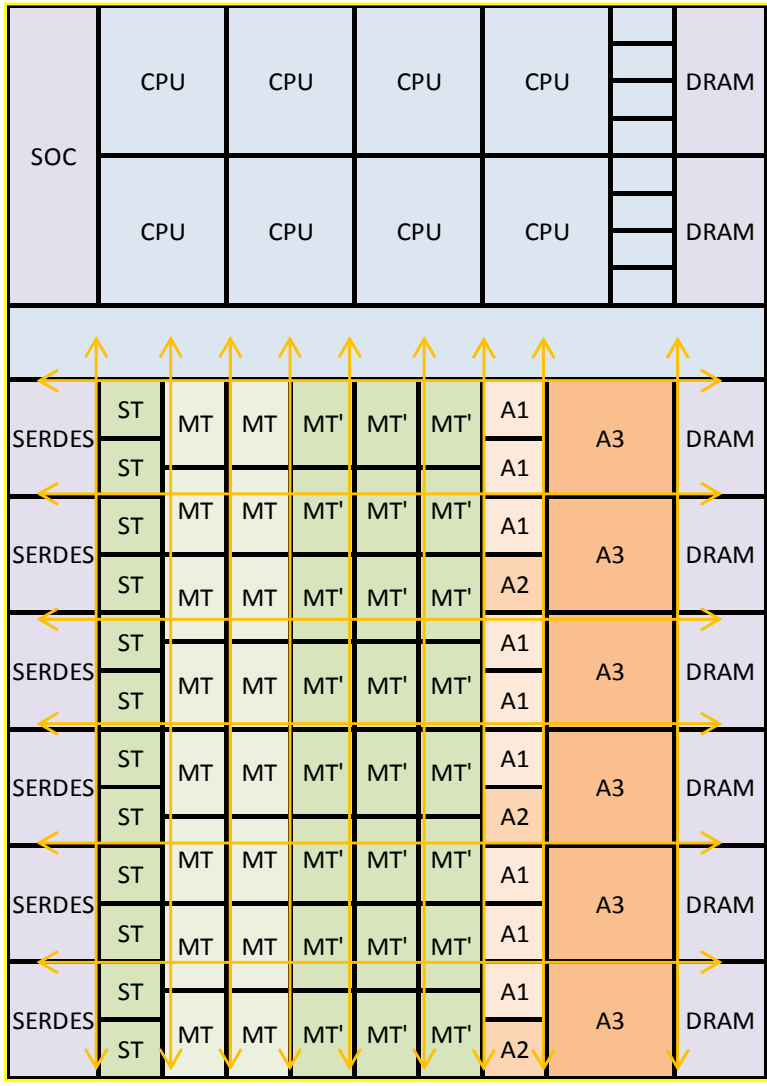
A Simple EDGE Microarchitecture for FPGA CPUs





Putting it All Together: Composable Heterogeneity

- **Parallel software** models, languages, libs
- Make it easy to compose software, and hard and soft cores into a "MPPAA"
- Crucial MPPAA infrastructure:
  - **Common NOC + host interfaces**
  - Routers, CPUs, accelerators, caches …
- RC as iterative performance engineering
  - CPUs → OpenCL, soft CPUs → specialized CPUs + memories → accelerators
- More 10s recompiles; fewer 1h PARs; zero months interfacing to host



In Summary

- FPGAs: explosion in resources but slow progress on Fmax → parallel computing
- "Software mostly" methodologies require new soft processors, processor arrays
- Also: ports of parallel programming models, tools, composition interfaces, frameworks

- *"It's a great time to be us."*

Thank You!

- Questions?

BACKUP

1998: “The Myriad Uses of Block RAM”

- **register files**; many-hundred-bit-word register file; vector register file; windowed register file, …
- **multiple register file contexts**, including user/kernel/interrupt handler shadow contexts, or multiple threads' contexts;
- **operand/data stacks**, control (incl. return address) stacks; unified locals+operands stack; storing one or more activation records; systems which automatically store and reload same, burst or trickle-back;
- **m-read n-write multiported** register files or stacks, via the embedded RAM's inherent multiport access, or time-multiplexed access, or replicated copies, or supporting multiple concurrent writes using replication + 'which replica valid'
- multiplier, divider, and/or trigonometic **lookup tables**, or partial lookup tables, coefficient lookup tables, interpolant estimate tables;
- **control stores**: wide encoded state machines, microcode, nanocode (multiple-level structures), possibly writeable;
- **branch prediction:** branch target address caches, branch target instruction caches, return address caches, branch history tables;
- instruction and/or data **cache data**, direct mapped or n-way set associative; **cache tags** (for on- or off-chip cache data); MOESI style cache coherence bits; snoop tags; … ; victim buffers; write buffers and write-accumulation structures;
- **segmentation registers, translation lookaside buffers**, mapping address to real address, present, valid, and/or dirty bits …;
- **per-task state tables**, including priorities, task state, next-task info, attributes, and masks; fast dedicated thread local storage;
- **debug support tables** including breakpoint code address registers, breakpoint data address / value registers, nonsequential IP history, branch taken history, memory access history;
- **on-chip RAM or scratchpad RAM**; multiple banks of same supporting multiported access, or interleaving; optionally preinitialized; on-chip ROM; use of these for interpreter or emulator code or data,
- **I/O buffers/FIFOs/queues** in general, linear, circular, or linked list; DMA staging buffers
- **garbage collection support**: read, write barriers via page table attribute bits or region table address checks; card marking bit arrays
- **on-chip network message/cell buffers**; queues; virtual channel message/cell buffers; node/address-to-info maps; use of same for message passing or shared memory multiprocessors and packet/cell switched network interconnect fabrics;
- **graphics**: display list, command queue, vertex lists; transformation matrices; compositing and accumulation buffers; texture cache

[http://www.fpgacpu.org/usenet/bb.html]

MicroBlaze: So Many Features

- Fault Tolerance, including Error Correction Codes (ECC) and Lockstep support
  - LMB BRAM memory
  - Parity protection on internal BRAMs and caches
- Floating Point Unit (FPU)
  - IEEE 754 compatible
  - Single precision
- Memory Management Unit (MMU)
  - Full MMU with VM supported by Linux
  - MPU mode region protection for secure RTOS
- Instruction and Data Caches
  - Size configurable: 2kB - 64kB (BRAM based)
  - Microcache: 64B – 1024B (LUTRAM)
  - Direct mapped write-through or write-back
  - Victim cache size configurable: 2, 4 or 8 lines
  - Instruction Stream Buffers
- Branch prediction logic
- Branch target cache
- Execution Hardware Acceleration
  - Barrel Shifter (1 cycle operation)
  - Integer Divide (32 cycle operation)
  - Multiply (1 cycle operation)
  - Instruction Set Extensions
  - Pattern Compare Instructions
- Machine Status Register Set and Clear
- Atomic Access
- Endian Conversion Support
- Hardware Exception Support
  - Unaligned access
  - Illegal instruction
  - Data bus error
  - Instruction bus error
  - Divide Exception
  - Floating Point Exception
  - FSL Exception
  - MMU Exception
- Low Latency Interrupt Mode
- Debug Logic
  - JTAG control via a debug support core
  - Up to 8 hardware break points
  - Up to 8 hardware watch points

Less is More / Simpler is Better

- Simpler is smaller
  - Many more cores/device, or less expensive device
- Smaller is faster
  - Fewer, shorter wires are faster wires
  - Easier to tune; less to tune; more spins/day
- Smaller is more energy frugal
- Simpler is easier to understand, maintain, test

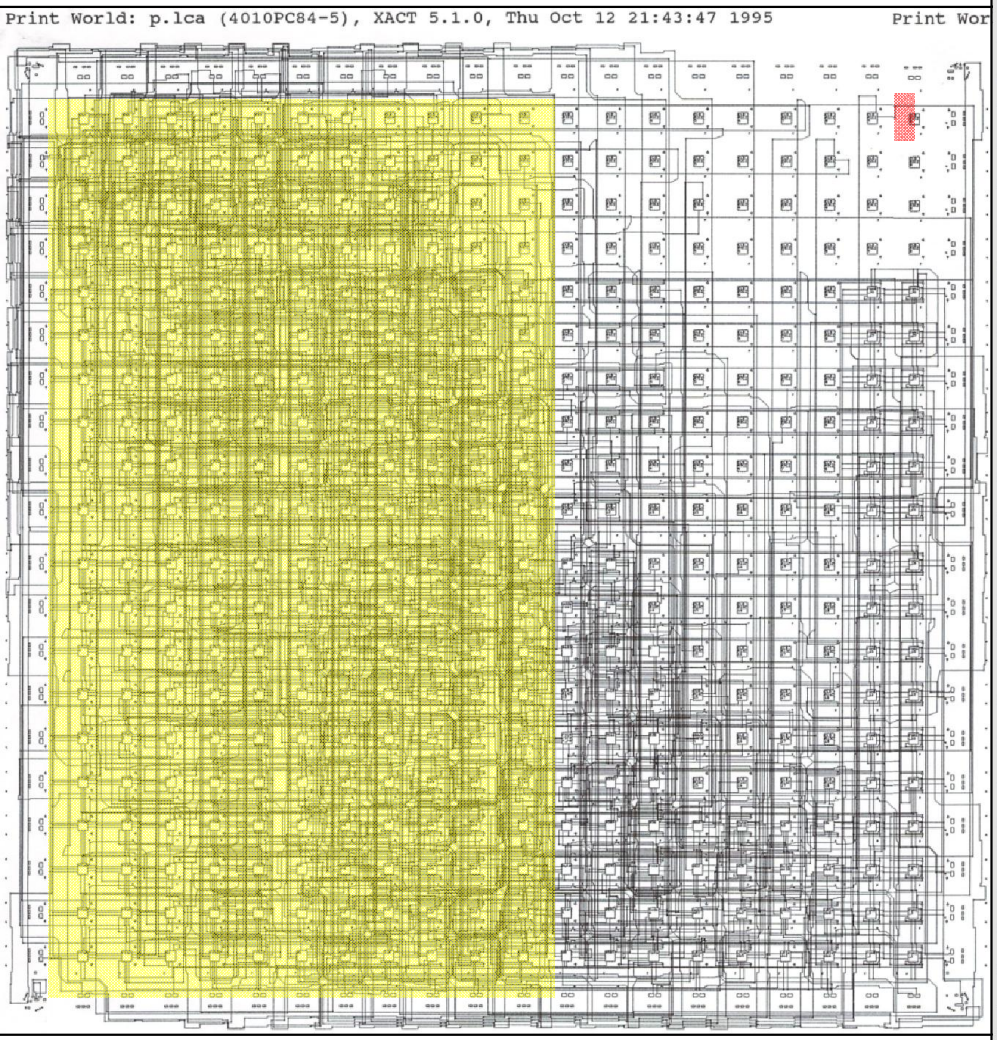
“Jan’s Razor”

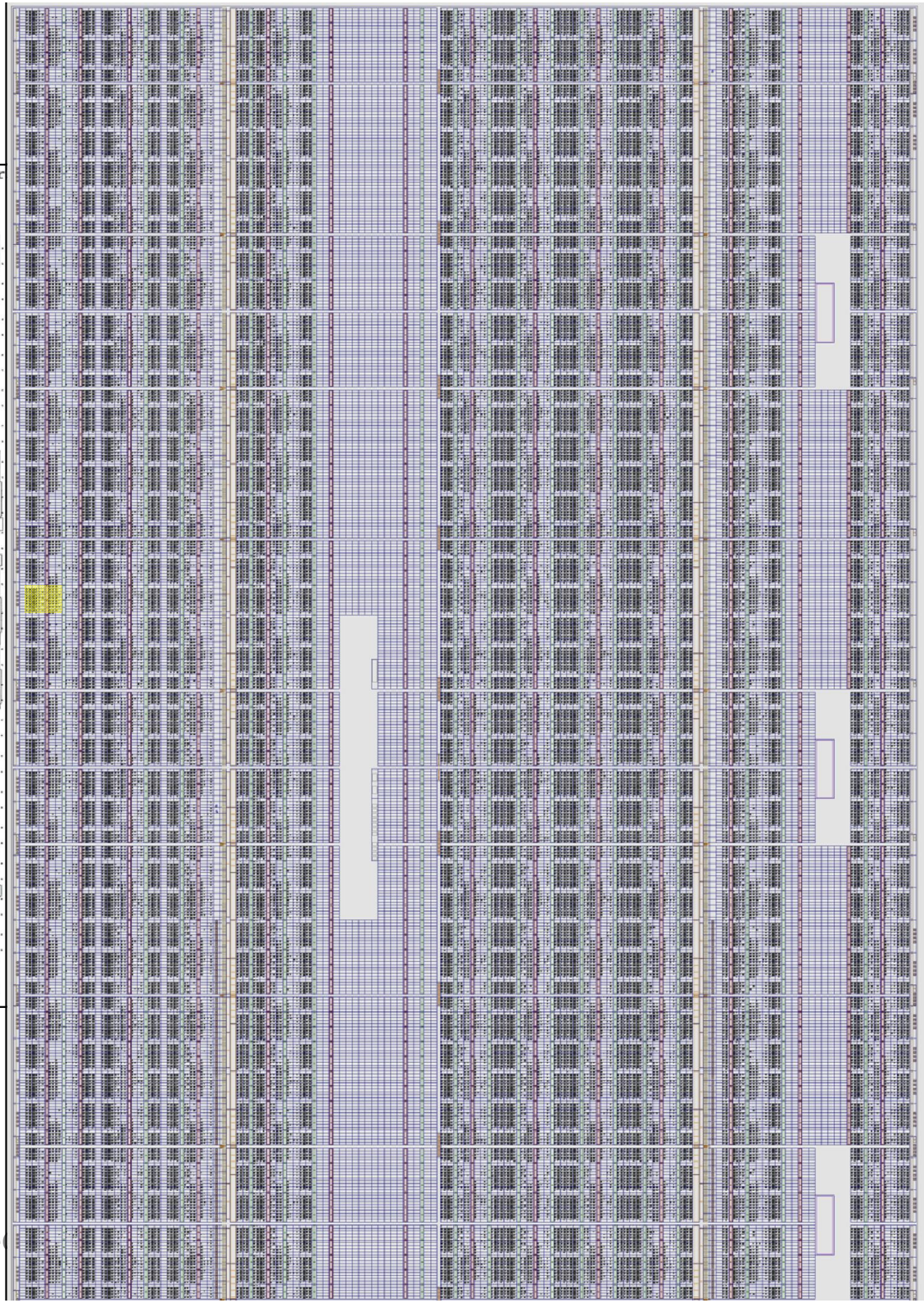
*“In a **chip multiprocessor** design, cut resources from each CPU, to maximize CPUs per die.”*

[http://www.fpgacpu.org/log/mar02.html#020305]

- Smaller processors → more processors
- Share less-used resources in cluster of cores

1995 → 2014!





- 500× LUTs*, 1000× PEs
  - **+40-45%/year**

Computing Goes Specialized, Integrated, Heterogeneous

- *Autumn of Moore's Law, Winter of Dennard Scaling*
- Continued performance scaling = ↓↓↓energy/result
- Cut waste:
  - Complex → essential
  - CPUs → specialized accelerators, RAMs, interconnects
  - Rethink abstraction layers *SW*↔*SW*↔*HW*↔*HW*
  - Provision right

- Datacenter architecture?

The Law of Energy Efficiency Inevitability

A more energy efficient approach will replace a less efficient one. Eventually.

MIMD vs. SIMT

- MIMD: Tera MTA
  - thd = next_thd; pc = pcs[thd]; ir = imem[pc];
    a = regs[thd][ir.ra]; b = regs[thd][ir.rb];
    regs[thd][ir.rd] = execute(thd, ir, a , b);
- SIMT: Classic GPU
  - thd = next_thd; pc = pcs[thd]; ir = imem[pc];
    for (i in WARP_SIZE) {
      a = regs[i][ir.ra]; b = regs[i][ir.rb];
      regs[i][ir.rd] = execute(i, ir, a , b);
    }