FPGA Technology Snapshot : Current Devices and Design Tools

Helena Krupnova, Gabriele Saucier

CSI/INPG, Grenoble, France

bogushev@imag.fr

Outline

- Motivations
- Latest FPGA devices overview
  - Xilinx, Altera, Actel, Lucent FPGAs
  - quantitative and qualitative comparisons
- FPGA design tools
  - flows
  - hot issues
  - strategy comparisons and analysis
- Conclusions

INPG / Design&Reuse Trainings on SoC Design Using Design And Prototyping Platforms

- 8 Trainings since March 1998
  - Grenoble (France), Noida (India), Santa Clara (USA)
- About 250 participants in total
- Topics:
  - latest FPGA devices & design tools
  - overview of the FPGA-based emulation and prototyping boards
  - IP and macro blocks available for FPGA implementations
  - multi-FPGA partitioning
  - case studies
  - invited lectures (BARCO Silex, Intel, BULL, STMicroelectronics)
  - vendor presentations
- Future seminars:
  - August 2000, Taiwan; October 2000, UK

Latest FPGA Devices

- Increased capacity
  - several Mlns PLD gates
- Hierarchical structures
  - logic, routing hierarchies
- Embedded memories
  - RAM (single/dual port), ROM, CAM, FIFO support
  - implementation of general logic in Embedded Array Blocks
- Enhanced embedded arithmetic resources
  - carry, cascade chains, dedicated adder, multiplier, counter support
- Support of different I/O standards (LVDS)
- Clock management circuits (DLL, PLL)

FPGA Characteristics

Device Family	Largest Device	Capacity, Basic Cells	Capacity, Eq Gates	MAX User IO Pin Number	Technology	Process
Xilinx 4000	XC40250XV	8,464 CLBs	250,000	448	SRAM	0.25µ
Xilinx Virtex, Virtex-E	XCV3200E	73,008 LCs	4,074,387	804	SRAM	0.18µ
Xilinx Virtex-E Extended Memory	XCV812E	21,168 LCs	254,016 (logic gates)	556	SRAM	0.18µ
Altera FLEX 8000	81500	1,296 LEs	16,000	208	SRAM	0.5µ
Altera FLEX 10K	EPF10K250	12,160 LEs	250,000	470	SRAM	0.25µ
Altera Apex20K, Apex20KE	EP20K1500E	51,840 LEs	1,500,000	808	SRAM	0.18µ
Altera ACEX 1K	EP1K100	4,992 LEs	100,000	333	SRAM	0.18µ
Actel PROASIC	A500K510	51,200 tiles	410,000	623	Flash	0.25µ
Lucent ORCA	OR3L225B	11,552 LUTs	340,000	612	SRAM	0.25µ

FPGA Characteristics

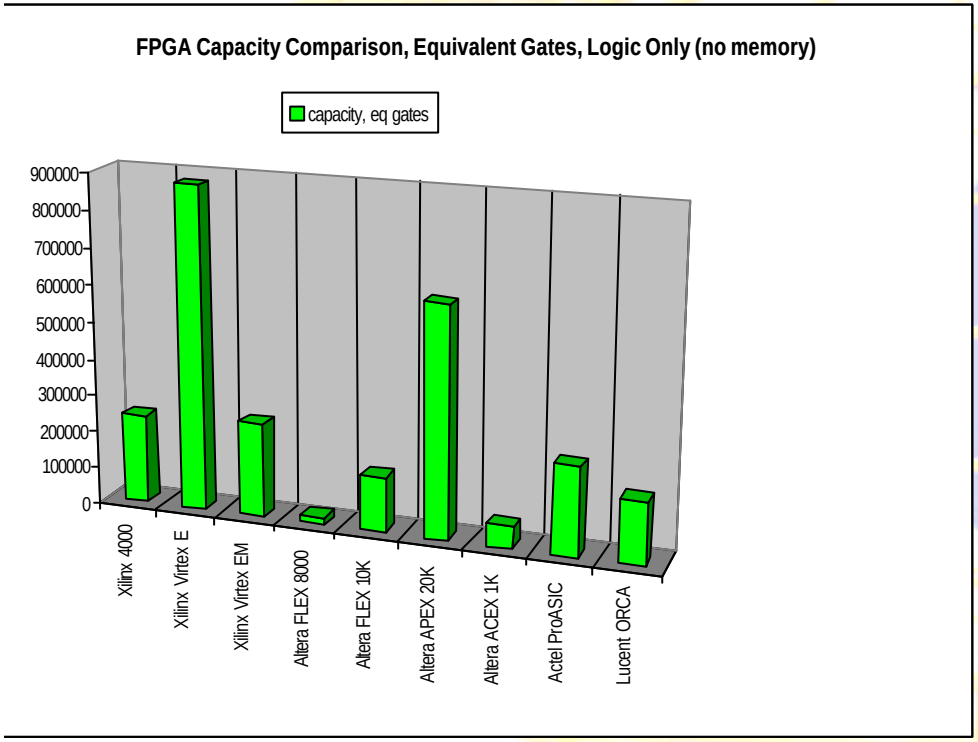
Device Family	Logic Primitive	Basic Cell (CLB)	Architecture	Interconnect	Carry Chains	Memories	Clock Management
Xilinx 4000	LUT4	2 4-input LUTs, 1 3-input LUT, 2 FF per CLB (F, G, H-function generators)	CLBs organized in rows and columns	CLB routing : vertical and horizontal channels (single-length, double-length, quad, octal lines and long lines), Programmable Switch Matrices	YES	270K RAM bits of Distributed Select RAM	N/A
Xilinx Virtex, Virtex-E	LUT4	2 LUTs per SLICE, 2 SLICES per CLB, 4 FF per CLB	CLBs organized in rows and columns	FastConnect – intra-CLB, local routing, general purpose routing (horizontal and vertical channels)	YES	851K RAM bits of BlockRAM + 1M of Distributed Select RAM,	DLL
Xilinx Virtex-E Extended Memory	LUT4	2 LUTs per SLICE, 2 SLICES per CLB, 4 FF per CLB	CLBs organized in rows and columns	FastConnect – intra-CLB, local routing, general purpose routing (horizontal and vertical channels)	YES	1,146K Block RAM bits + 301K Distributed Select RAM	DLL
Altera FLEX 8000	LUT4	1 LUT, 1 FF per Logic Element (LE)	8 LEs per LAB, lABs organized in rows and columns	Local Interconnect within LAB, Row&Column Interconnect	YES	N/A	N/A
Altera FLEX 10K	LUT4	1 LUT, 1 FF per Logic Element (LE)	8 LEs per LAB, LABs organized in rows ans columns	Local Interconnect within LAB, Row&Column Interconnect	YES	41K RAM bits of Embedded Array Blocks (EABs)	PLL
Altera Apex20K Apex20K E	LUT4	1 LUT, 1 FF per Logic Element (LE)	MultiCore : LUT, Product Term, Memory ; 10 Les per LAB, 16 LABs per MegaLAB, MegaLABs organized in rows and columns	Local Interconnect within LAB, MegaLAB Interconnect, Row&Column Interconnect	YES	ESB capable to implement RAM, ROM, CAM, ProductTerm, 442K RAM	PLL
Altera ACEX 1K	LUT4	1 LUT, 1 FF per Logic Element (LE)	8 LEs per LAB, lABs organized in rows and columns	Local Interconnect within LAB, Row&Column Interconnect Full-length, half length row channels	YES	EABs (49K bits)	PLL
Actel PROASIC	3input-1output tile	Tile can be configured either as a comb. Cell or as FF	Tiles are organized in blocks, blocks of tiles are organized in rows and columns	Local network, long line network, bus network, global network	NO	Max of 60 Embedded RAM Blocks (256x9)	N/A
Lucent ORCA	LUT4	PLC consists of a PFU and a SLIC ; PFU contains 8 LUT4 and 9 FFs	Array of PLCs, the whole array is split in four quadrants	Segmented hierarchical routing : Interquad routing, inter PLC routing	YES	185K : implemented as a special operating mode of the PLCs	PCM

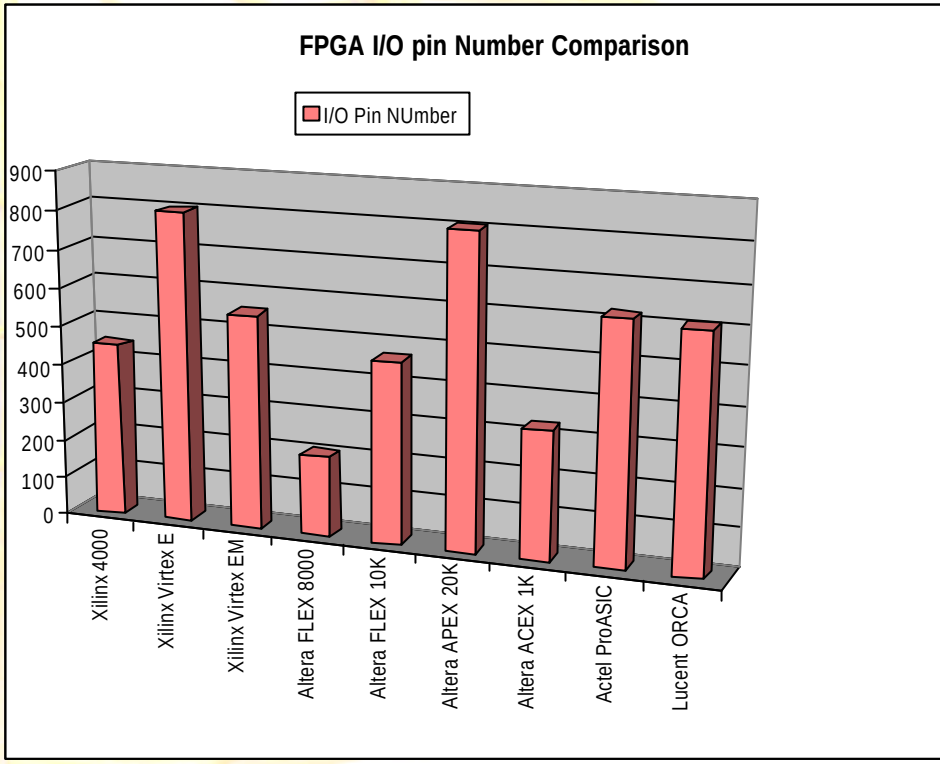
FPGA Density Cross Reference

Device Family	Largest Device	Logic Block	# Logic Cells (1) per Logic Block	Capacity, Logic Cells	# Memory bit	Capacity, Typical Gates (Logic and RAM)	Capacity, Logic Gates (no memory)
Xilinx 4000	XC40250XV	CLB	2.375	8,464 CLBs	270K(2)	440,000 (7)	241,000
Xilinx VirtexE	XCV3200	CLB	4.5	73,008 LCs 16,224 CLBs	851K+1M(2)	4,074,387 (3)	876,096
Xilinx Virtex E Extended Memory	XCV812E	CLB	4.5	21,168 LCs 4,704 CLBs	1,146K+301K(2)	-	254,016
Altera FLEX 8000	81500	LE	1	1,296 LEs	N/A	16,000 (4)	16,000
Altera FLEX 10K	EPF10K250	LE	1	12,160 LEs	41K	250,000 (4)	149,000
Altera Apex20K	EP20K1500E	LE	1	51,840 LEs	442K	1,500,000 (4)	622,000
Altera ACEX 1K	EP1K100	LE	1	4,992 LEs	49K	100,000	60,000
Actel PROASIC	A500K510	tile	0. 4(?)	51,200 tiles	138K	410,000	240,000 (?)
Lucent ORCA	OR3L225B	LUT	1	11,552	185K(5)	340,000 (6)	166,000

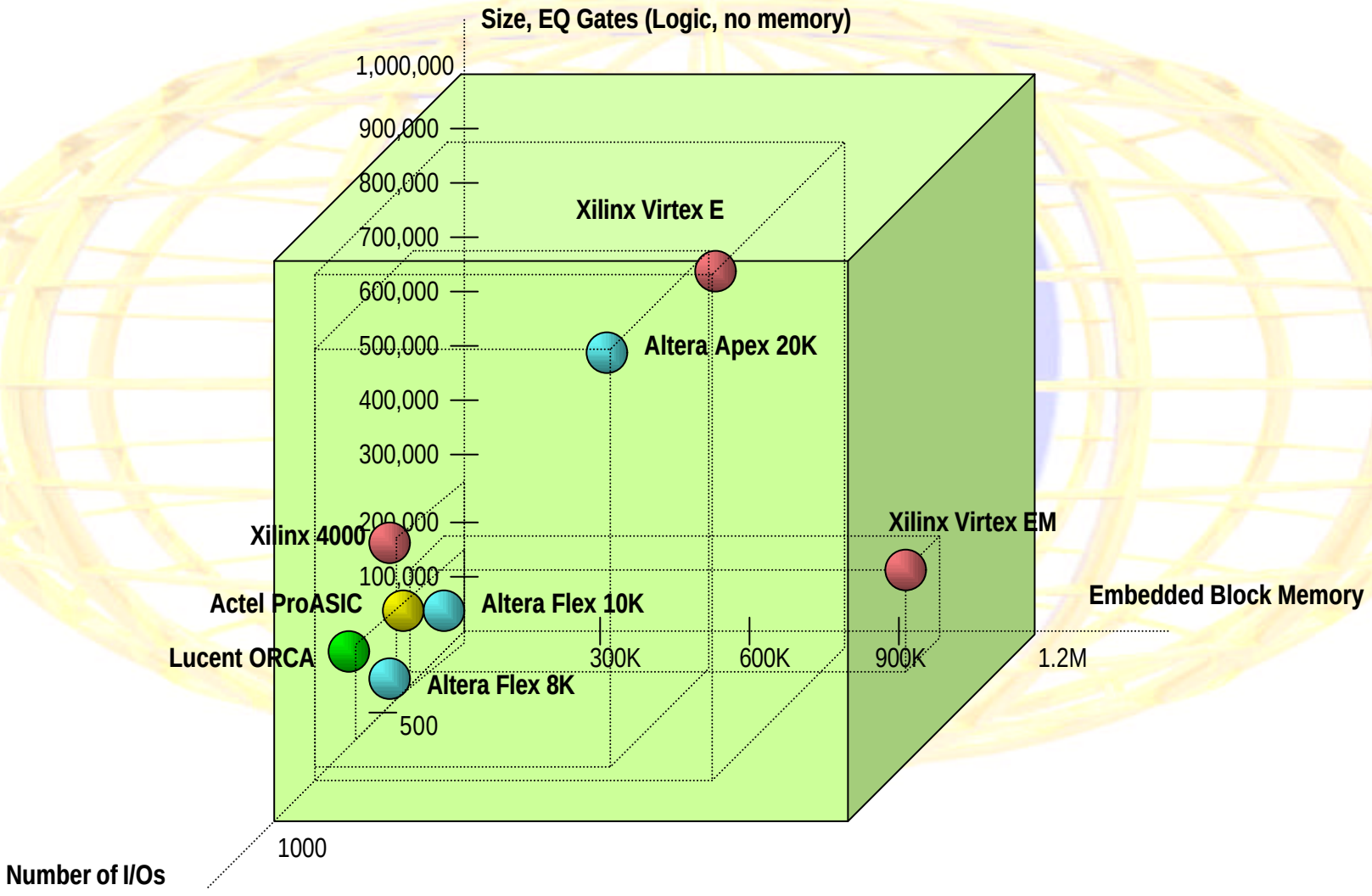
(1) A Logic Cell (by Xilinx) consists of a 4-input Look-Up Table (LUT) and a flip-flop (FF)
(2) Distributed Select RAM - uses Logic Cell resources
(3) Xilinx XC4000: about 28.5 gates/CLB (from XAPP059);
(4) Altera : about 12 gates/LE
(5) RAM mode uses logic cell resources
(6) 108 gates per PFU/SLIC; 12 gates per LUT/FF pair; the estimation assumes 30% of logic used as memory

FPGA Capacity and I/O Pin Comparisons

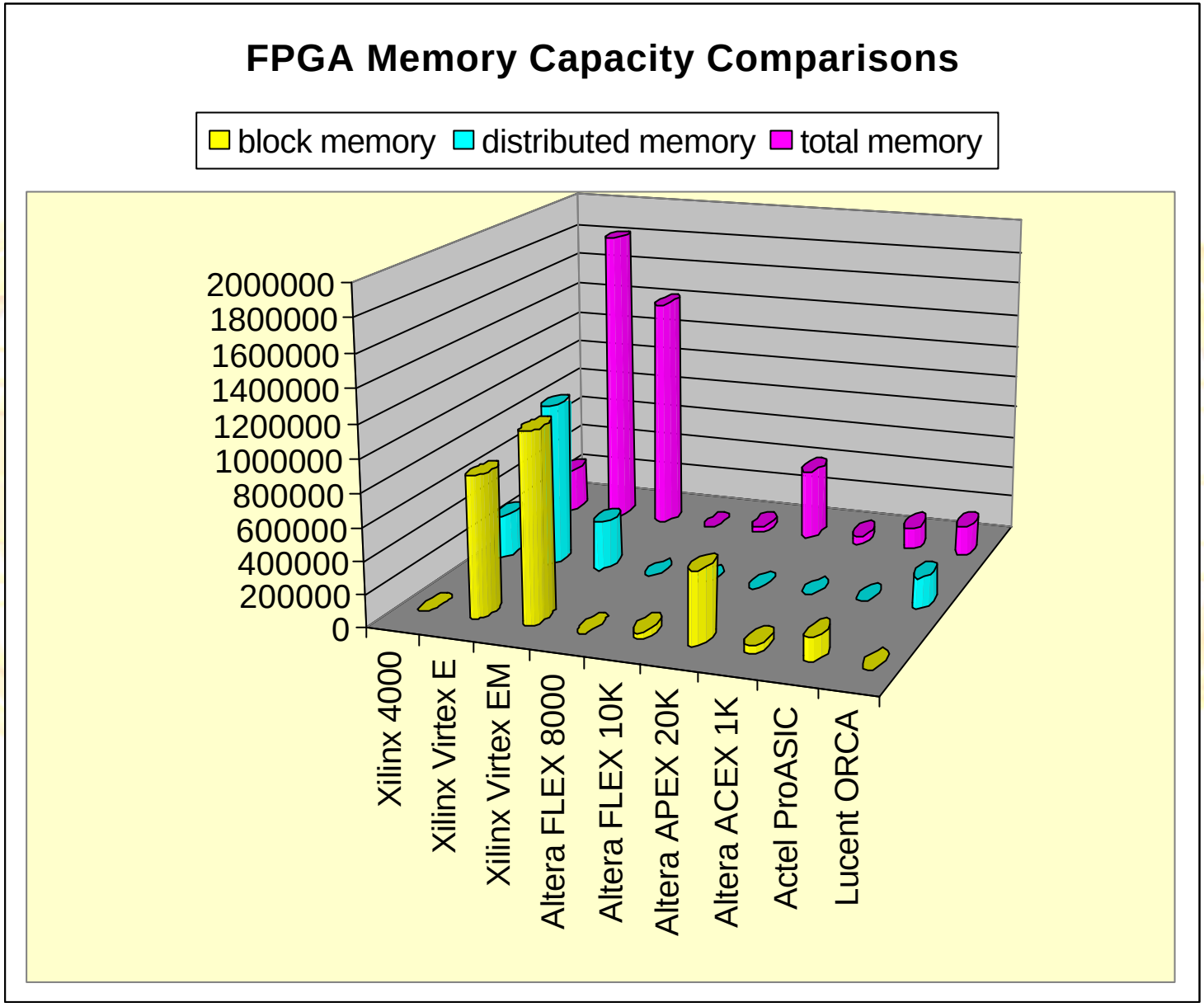




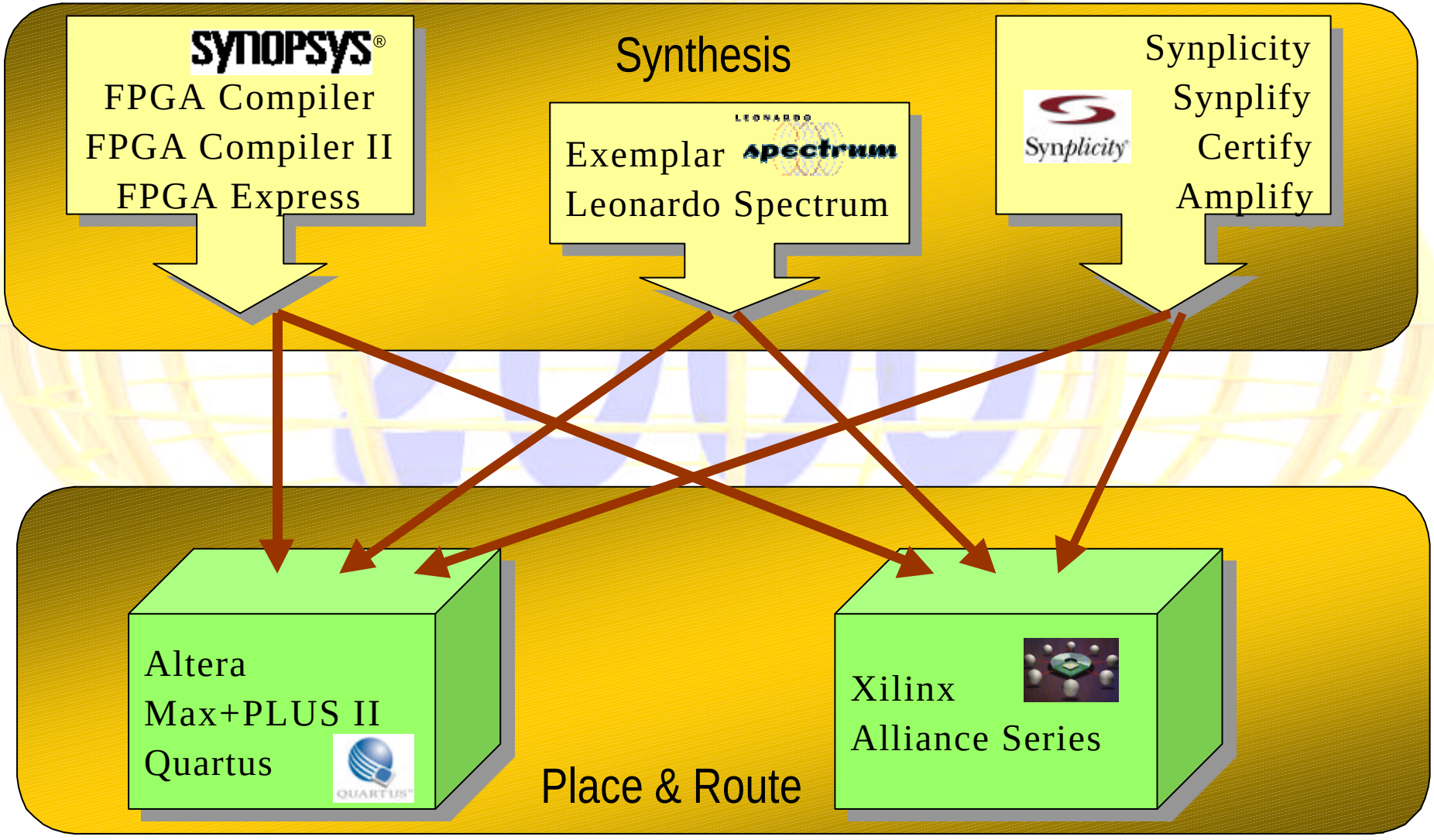
FPGA Quantitative Classification



Memory Capacity Comparisons



FPGA Design Tools



FPGA Design Flow Specific Issues

- Partitioning for synthesis
- Floorplanning
- Macro block processing
- Memory inference
- Timing constraints
- FPGA advanced synthesis features
  - pipelining, retiming, register duplication, operator sharing
- HDL coding styles
  - CASE vs. IF-THEN-ELSE
- Debugging facilities
  - embedded logic analyzer functions
- Design issues
  - latches/ gated clocks/ using DLLs, PLLs/ tri-state busses/ buffering/ FSM encoding styles

Partitioning for Synthesis

- Tradeoff between the faster CPU times and better results
- RMM guideline :
  - putting the registers at the front or the back of the hierarchy blocks
  - grouping related combinatorial logic into the same module
  - separating timing-critical blocks from area-critical blocks
  - limit clock to 1 per block and synthesize the blocks separately
- Preserving the hierarchy blocks of about 50K gates and flattening the sub-hierarchy

	Exemplar Leonardo	Synplicity Synplify	Synopsys Design Compiler, FPGA Compiler, FPGA Express
Partitioning for synthesis	• Manual hierarchy manipulation (group/ ungroup) • Auto hierarchy control (defining treshold) • The hierarchy is preserved by default	• Automatic hierarchy handling (dissolving, optimizing and rebuilding) • syn_hier attribute	• Manual partitioning by group/ungroup command

Floorplanning

- « Divide and conquer » approach to the place and route problem
  - creating efficient local placement solutions (RLOC constraints in Xilinx)
  - evenly distributing logic among the chip, avoiding routing congestion (LOC constraints in Xilinx)
  - putting logic belonging to the critical block close together (clique constraints in Altera)
- Requires the expert knowledge of the targeted architecture and the design tool
- Gate level - FPGA vendor floorplan tools
- RT-level floorplanner - Amplify tool from Synplicity

Macro Block Processing

- Inference, Instantiation
  - HDL code portability vs. efficiency
- Macro blocks are :
  - specifically optimized for speed/area
  - have predefined placement solutions

	Exemplar Leonardo	Synplicity Synplify	Synopsys Design Compiler, FPGA Compiler, FPGA Express	Xilinx	Altera
Macro block proces-sing	• Modgen macro generator • Inference • Instantiation	• Inference • Instantiation • Context-driven module generation	• Inference (DesignWare) • Instantiation	• CoreGEN, LogiCORE, LogiBLOX macro generation tools • Macros should be instantiated in HDL code	• LPM functions • MegaWizard Altera IP generator

Memory processing

- Implementation possibilities:
  - embedded block memory
  - special LUT configuration mode (distributed memory)
  - using general logic resources
- Automatic inference for limited cases
- Memory wrapping and manual handcrafting

	Exemplar Leonardo	Synplicity Synplify	Synopsys DC, FC, FPGA Express	Xilinx	Altera
Memory implementation	Automatic RAM Inference from HDL code	• Automatic RAM inference from HDL code • Only synchronous RAM is inferred, asynchronous RAM not supported • syn_ramstyle attribute (to select between Xilinx RAM styles)	no RAM inference	• instantiating Xilinx RAM primitives in VHDL code • generating memories with MemGen, LogiBLOX programs	• EAB memory functions are accessible via LPM • *genmem* utility • Altera megafunctions (for FIFO, CAM, etc.)

Timing Optimization

- Timing Constraints
  - clock constraints
  - input arrival, outut required time constraints
  - multicycle path constraints
  - false path constraints
  - multiple clock constraints
- Timing Optimization
  - pipelining, retiming
  - constraints on the fanout (inserting buffers for cutting long-delay nets)
  - resource sharing
  - register duplication
  - assigning critical logic functions to EABs
- Overconstraining can lead to worse results
  - result is not a linear function from constraint values

Conclusions

- Device vendors
  - extended libraries of IPs and macro blocks
  - FPGA test boards for the latest devices
  - embedded logic analyzers
  - internet-oriented software
  - experts programs
- Tool vendors
  - moving to the higher level of abstration
    - RTL floorplanning
  - better integration between synthesis and layout
    - second-pass synthesis with place and route SDF files
    - floorplanning-based synthesis
  - incremental synthesis / place and route, ECO support
  - optimizing runtimes