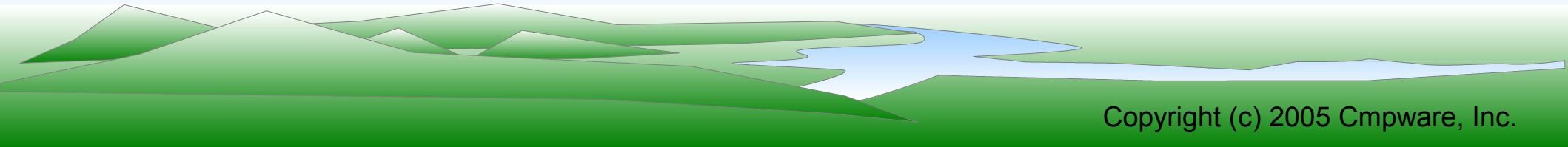
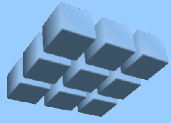


Reconfigurable Computing

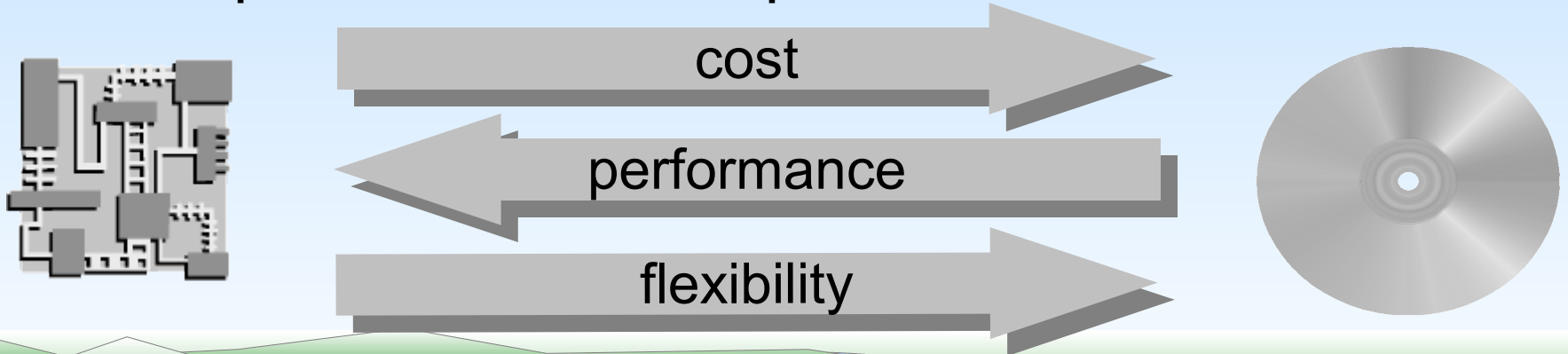
Steven A. Guccione
Cmpware, Inc.

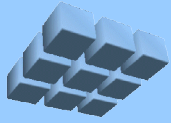




Introduction

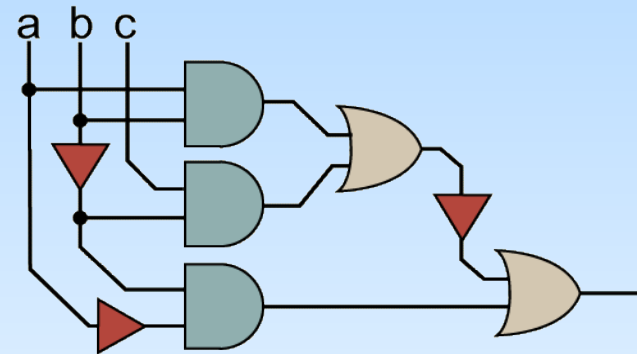
- Two traditional implementations for system design: hardware and software
- **Hardware:** custom circuitry for a specific task
 - High Performance, expensive and inflexible
- **Software:** programmable microprocessor
 - Low performance, inexpensive and flexible

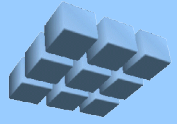




Programmable Logic

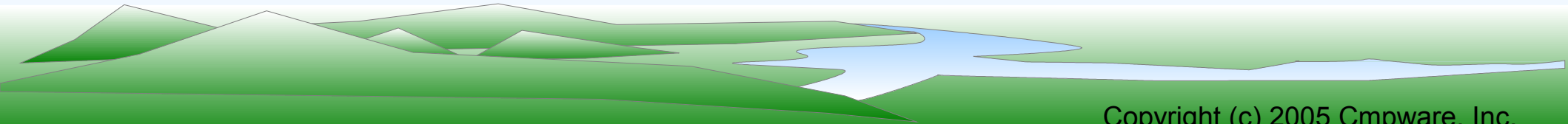
- Pioneered by MMI in mid-1970s
- Devices that could be user-customized
- Made custom hardware more flexible and less expensive
- Early technology: PALs
 - Programmable Array Logic
 - One-time programmable AND-OR array
 - Limited size (dozens of gates)
 - Very useful for hardware interfacing

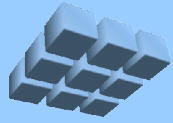




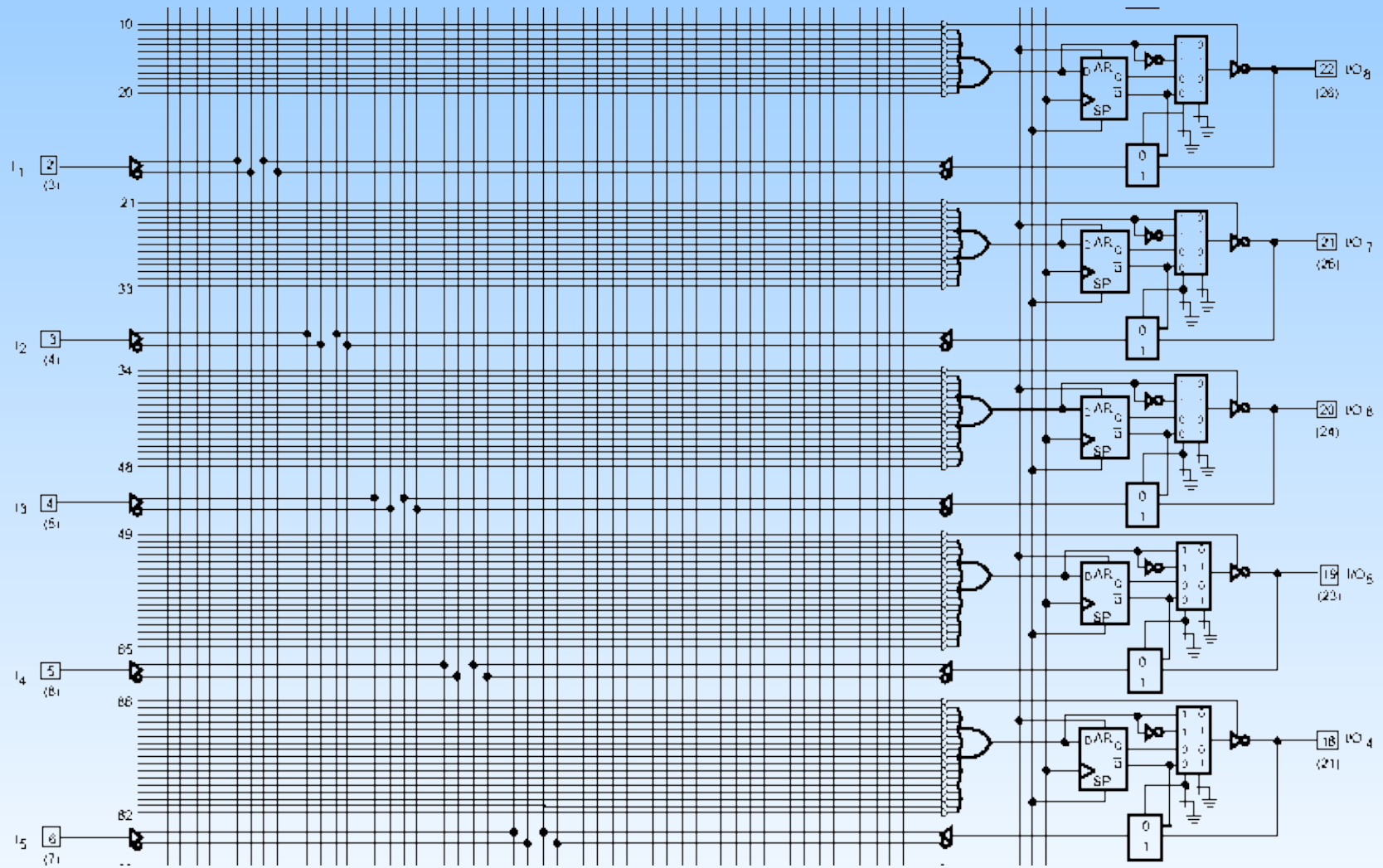
Programmable Array Logic (PAL)

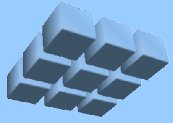
- Produces user-defined digital circuits
- Programmed with boolean logic equations
- Useful for interfaces, state machines
- Lowers cost of designing hardware
 - New ICs can cost millions in initial investment
 - Manufacturing new ICs requires high volumes
- Adds flexibility; mistakes can be fixed





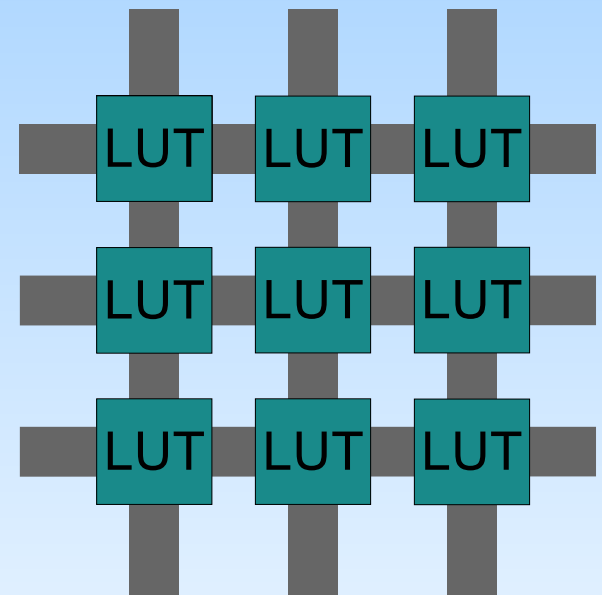
Programmable Array Logic

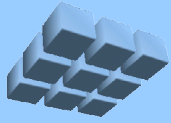




Field Programmable Gate Array

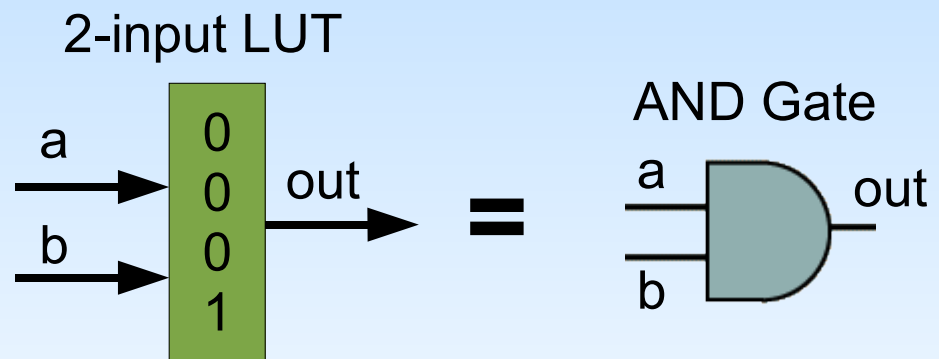
- Pioneered by Xilinx in the mid-1980s
- A reprogrammable logic array
- RAM-based programming
- Look-Up Table (LUT) logic
- Switch-based interconnect
- Initially hundreds of gates
- Often used to replace PALs



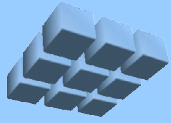


Look-Up Tables (LUTs)

- Small Random Access Memory (RAM)
- Usually 3 or 4 inputs (8 or 16 bits)
- Can implement arbitrary logic functions
- Not very efficient (Approx. 100x penalty*)
- Reprogrammable

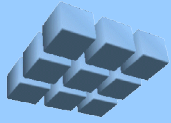


* 3000 transistors per LUT (incl. routing config bits); LUT == 5 gates



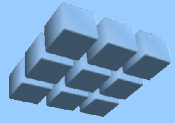
Logic Emulation

- Use large numbers of FPGA to speed up digital circuit simulation (“emulation”)
- Useful for verifying correctness of custom circuits before (expensive) fabrication
- Sold by several smaller companies (*PiE*, *QuickTurn*, *Inca*) in the mid-1980s
- *Teramac*: Full custom emulation machine from HP. Also used for computation.



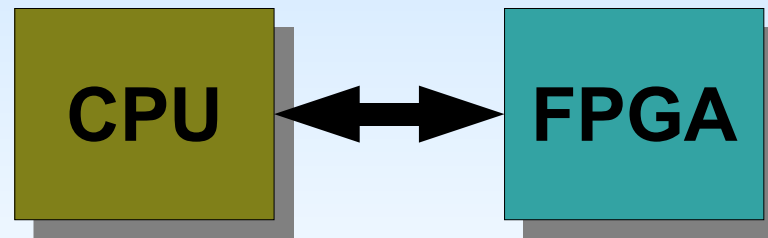
Reconfigurable Computing

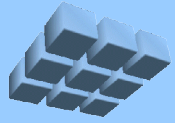
- **Definition:** Using Reconfigurable Logic (FPGAs) to perform calculations
- **Goal:** provide hardware speeds with software programmability
- Much research and commercial activity throughout the 1990s
- FPGA densities grew from hundreds of gates to millions of gates (approx. 65% per year)
- Significant computation possible



Reconfigurable Computing Technology

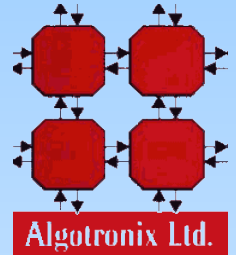
- Hardware implementations orders of magnitude faster than software
- The RC approach:
 - Identify computationally intensive “kernels”
 - Implement kernels in RC (FPGA) co-processor
 - All other software executes in CPU

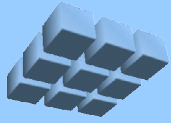




Early RC Systems: Late 1980s

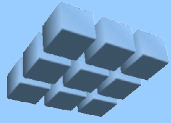
- **Splash**: Large military system from Supercomputer Research Center (SRC).
- **PAM**: Research system by DEC (later Compaq, then HP). 4 FPGAs per board.
- **Virtual Computer Corporation**: Large FPGA board from small start-up company.
- **Algotronix**: Small Scottish company with proprietary FPGAs. Acquired by Xilinx.
- **GigaOps**: Image processing applications.





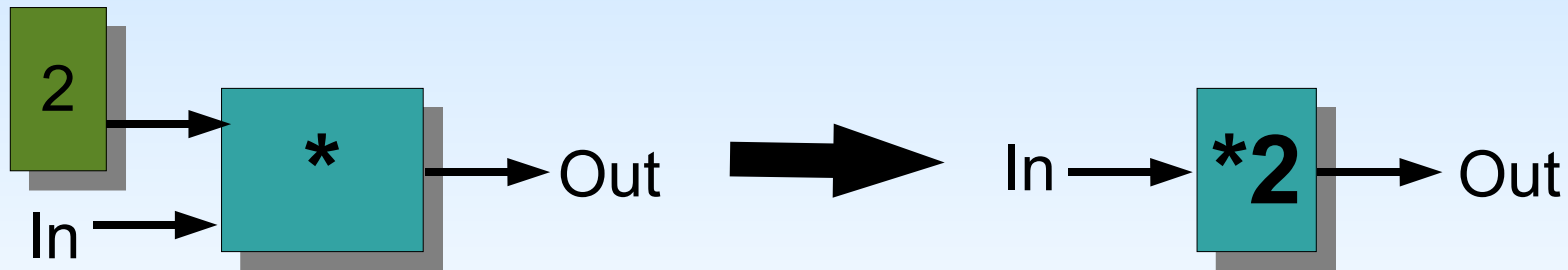
Tool Support

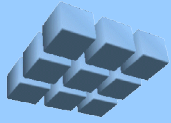
- Many early “C” compiler projects
 - Prism II (Athanas at Brown U.)
 - Data Parallel C (Gokhale at SRC)
 - Handel-C (Page at Oxford)
 - PamDC (Shand at DEC)
- Handel-C commercialized by Celoxica
- Manual HW / SW partitioning and implementation common



Run-Time Reconfiguration

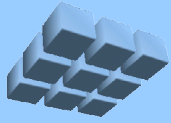
- HW tools design fixed circuits
- FPGAs circuits can be changed dynamically
- Example: a constant multiplier
 - Saves area, power and is faster
 - Difficult to implement from a library (N^2)
 - FPGAs can dynamically customize circuitry





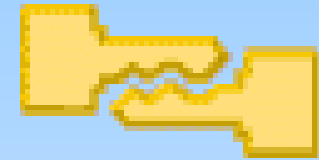
JBits

- FPGA API developed by Xilinx researchers
- Gave bit-level access to FPGA configuration
- Enabled dynamic logic and routing
- Bypassed traditional “batch mode” circuit design tools (schematics, HDLs)
- Gave access to closed FPGA architectures
- Some JBits apps faster than custom circuits
 - Circuit customization
 - Hardware re-use



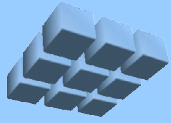
JBits Applications

- DES Encryption:
 - Customized circuit to encryption key
 - Resulting circuit faster than custom circuits
- Gene matching:
 - Customized circuit to data to be matched
 - Faster than a 10,000 processor server farm



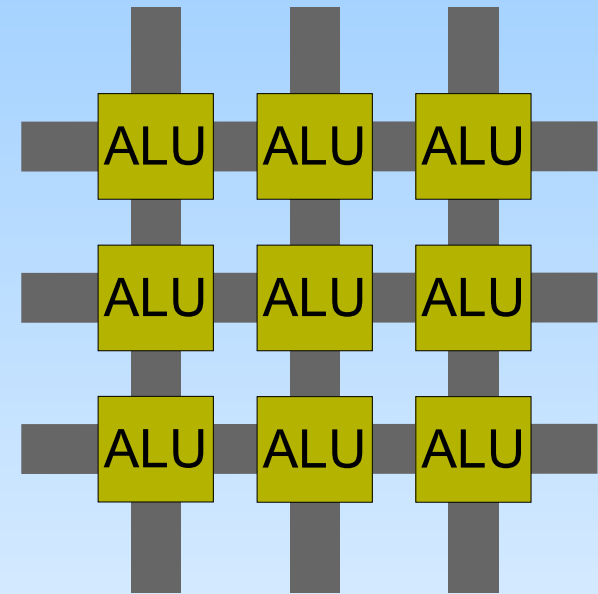
Cameron Patterson. “**DES Encryption in Virtex FPGAs Using JBits**”. In *IEEE Workshop on FPGAs for Custom Computing Machines*, pages 113-121, Los Alamitos, CA, April 2000. IEEE Computer Society Press.

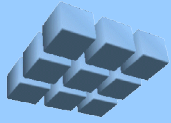
Steven A. Guccione, Eric Keller, “**Gene Matching using JBits**”, *Proc. 12th Int. Workshop on Field-Programmable Logic and Applications (FPL 2002)*, Springer, LNCS 2438, 2002, pp. 1168-1171.



Generation 2: ALU Arrays

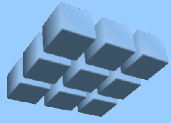
- Replace LUTs with ALUs
- More efficient logic
- Simpler mapping (DSP)
- FPGA-like routing
- Commercial ALU Arrays:
 - Chameleon (*CS2000* family)
 - Elixent (*C-Fabrix*, *RAP Array*)
 - PACT (*XPP Array*)





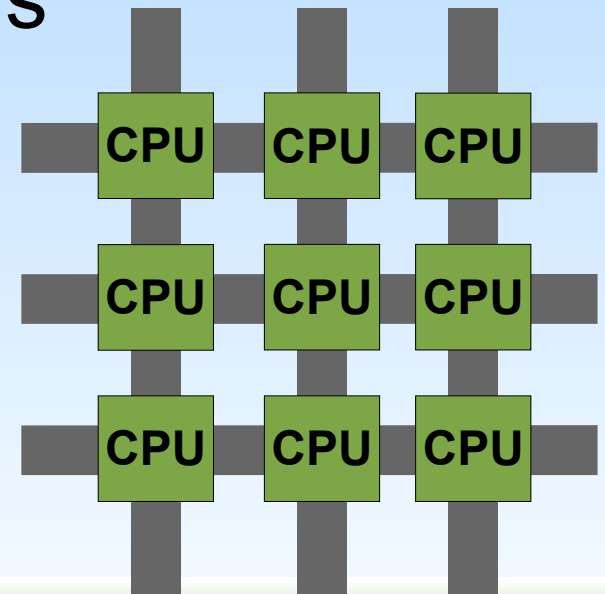
CPUs and FPGAs

- Hard-wired CPUs in FPGAs
 - Xilinx Virtex II Pro
 - Altera Excalibur
 - QuickLogic QuickMIPS
- Soft-wired CPUs becoming popular in FPGAs
 - 32-bit RISC Soft CPUs == ~1,000 LUTs
 - Hundreds of soft CPUs possible in an FPGA
 - A highly re-programmable system



Generation 3: CPU Arrays

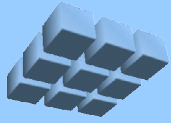
- Next step in reconfigurable architectures
 - LUT --> ALU --> CPU cells
- FPGA cell size similar to 32-bit RISC CPU
 - 10,000s of processors on a chip possible
- Many multicore custom designs
- Commercial CPU Arrays:
 - QuickSilver (*ACM*)
 - PicoChip (*PC102*)
 - ClearSpeed (*CSX600*)





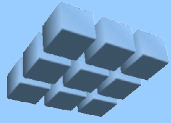
Reconfigurable Computing Issues

- HW + SW co-design:
 - Often HW tools used for HW, SW tools for SW
 - HW / SW interfacing often difficult
 - Requires highly skilled engineers
- Performance:
 - Co-processor communication issues
 - Size of reconfigurable hardware
 - *Amdhal's Law*: large amounts of parallelism required for large speed-ups (90% parallel == 10x speed-up, max).



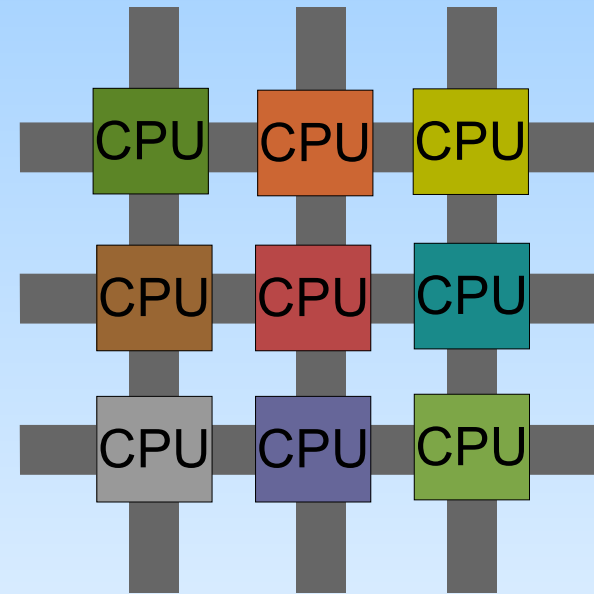
Conclusions

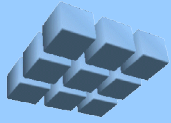
- Reconfigurable Computing has extended system performance
- Problems still exist, primarily with tools
- FPGA companies supporting computation
 - Custom hardware support for DSP (multipliers)
 - Toolkits for Matlab interfacing
 - High Level Language projects like *JBits* and *Lava*
- New Engineering System Level (ESL) tools
- New architectures on the horizon



Future Directions

- Larger FPGAs
 - Higher performance circuits
 - Support for floating point
- New architectures
 - Heterogeneous arrays
 - Processor arrays
- Software tools
 - Celoxica (*Handel-C*), accelChip (*Matlab*), Xilinx (*Forge*), Accelerated Technologies (*ImpulseC*), ...





Resources

- **Conferences:**

- *Field Programmable Custom Computing Machines (FCCM)*
- *Field Programmable Logic (FPL)*
- *Field Programmable Technology (FPT)*

- **FPGA Companies:**

- *Xilinx*
- *Altera*

- **University Research:**

- *Virginia Tech Configurable Computing Lab*
- *BYU Configurable Computing Lab*
- *Oxford Hardware Compilation Lab*