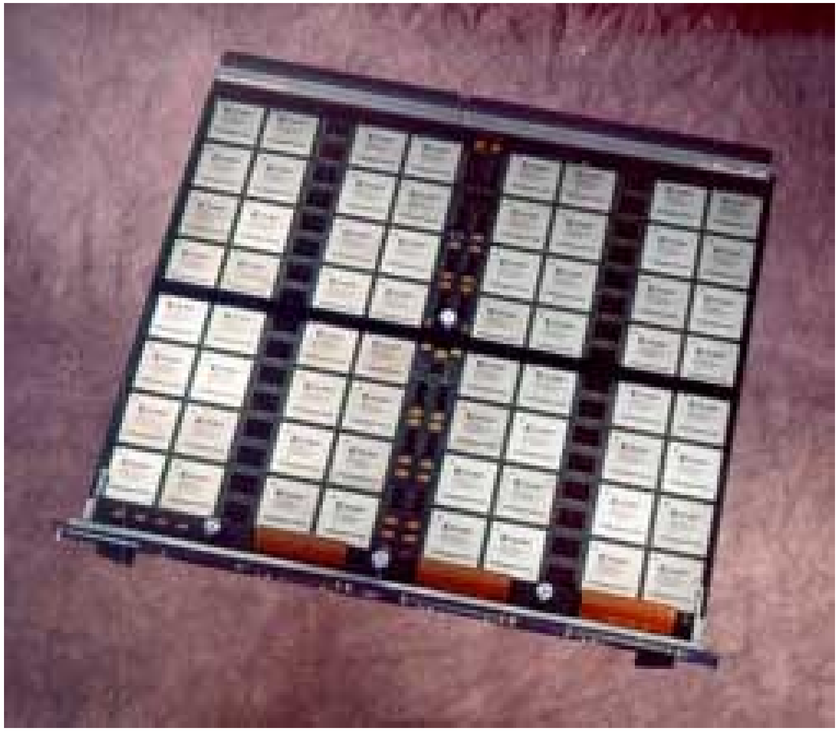
XILINX®

Physical Design For FPGAs

Rajeev Jayaraman
Physical Implementation Tools
Xilinx Inc.

Do you know FPGAs?



- FPGAs are used only in prototyping and emulation systems?
- Can you design anything real in FPGAs?
- FPGAs are too expensive even for moderate volumes right?
- FPGAs are a niche market right?

This isn't your father's FPGA

- FPGAs are being used in mainstream products
  - Networking
  - Telecom
  - DSP
  - Consumer electronics
- More FPGA design starts than ASIC design starts
- 2 FPGA companies in the top 10 chip suppliers

Agenda

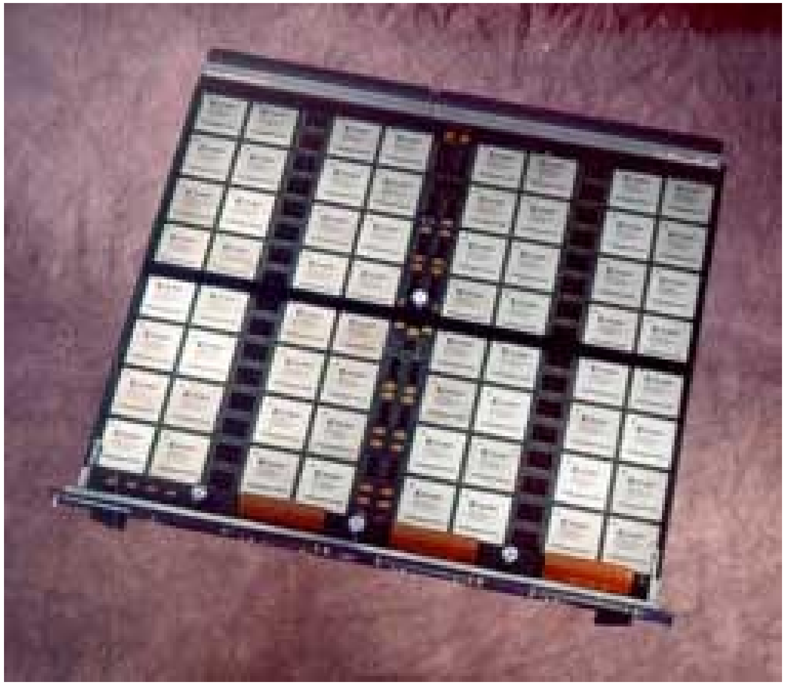
- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

Agenda

- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

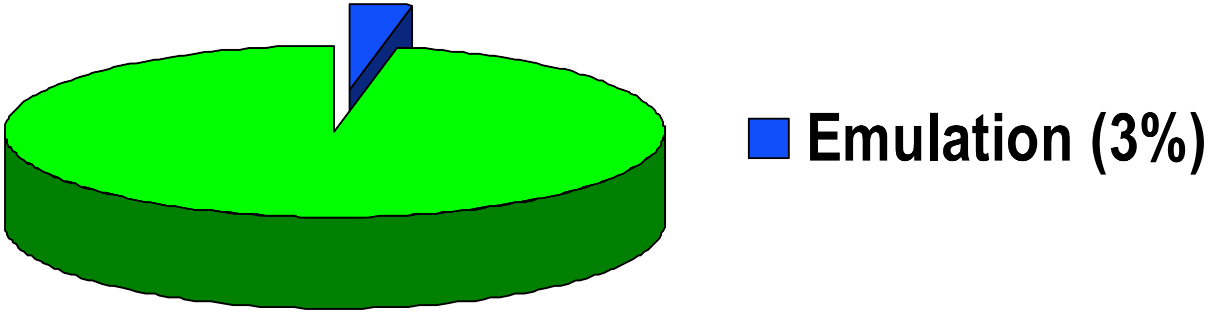
Use in Emulation systems

- Functionally debug complex systems
- Vendor supplied or home built systems



Use in Emulation Systems

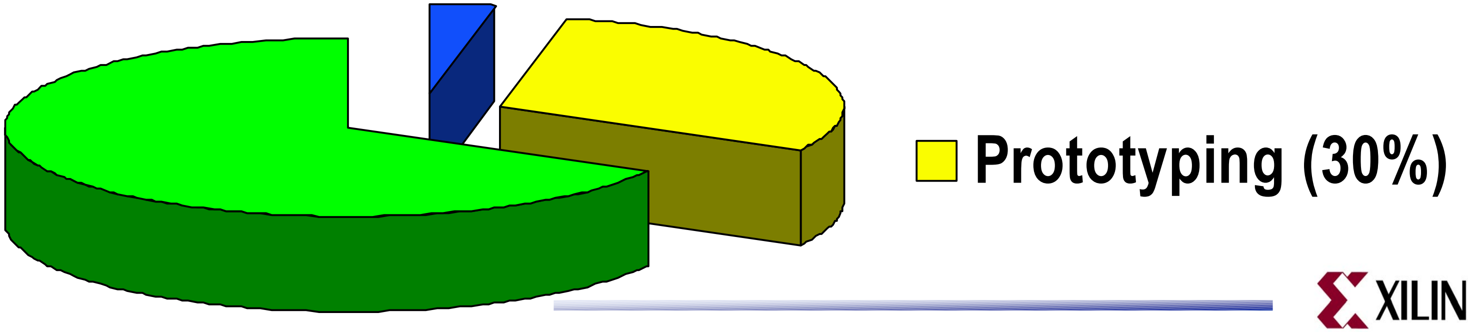
Time-to-market	Fairly high; Fast compile times
Performance	Not stringent
Volume	Very low per application



Use in Prototyping Systems

- Prototype a system
- Maybe deployed in the field in small quantities

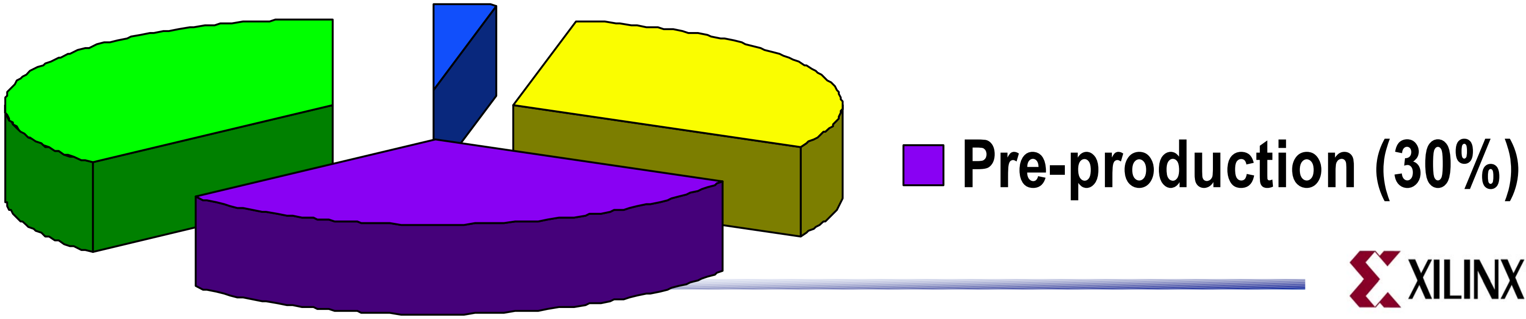
Time-to-market	Fairly high; Fast compile times
Performance	Not stringent
Volume	Low per application



Use in Pre-Production Systems

- FPGAs are central to the system
- Design may migrate to ASICs eventually
  - Most don't because of reprogrammability

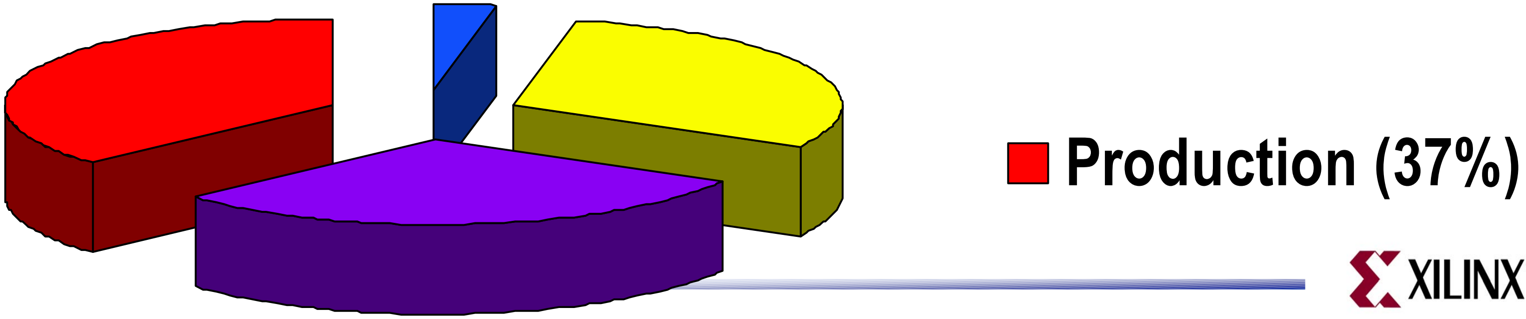
Time-to-market	Fairly high; Fast compile times
Performance	Very critical
Volume	Moderately high per application



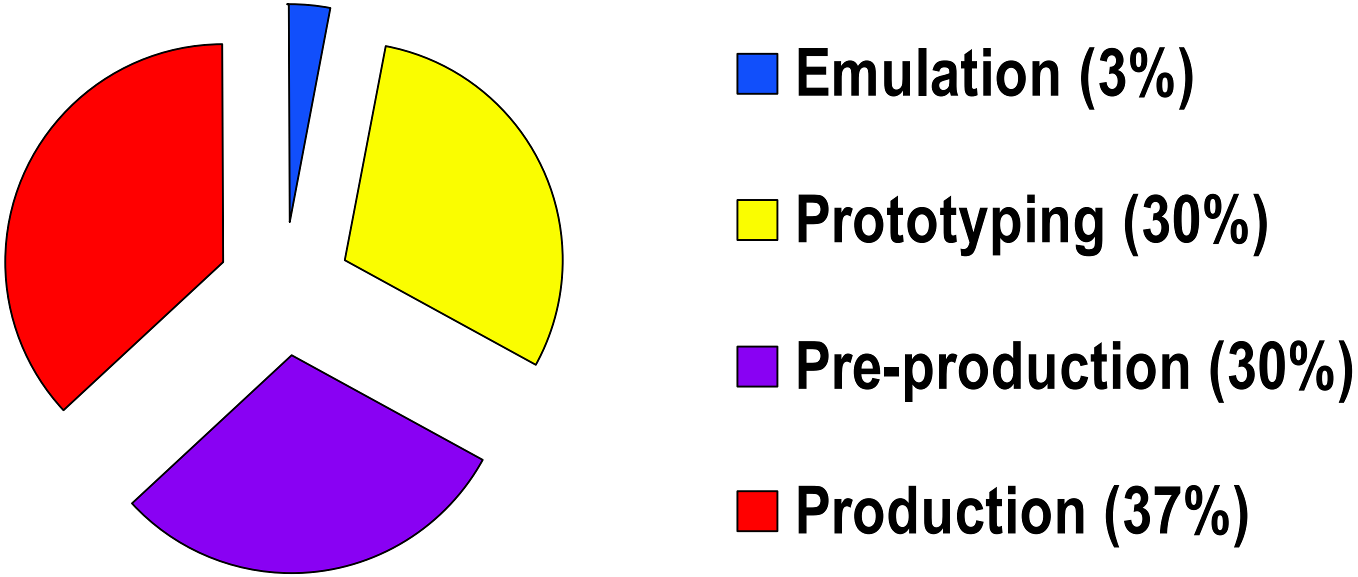
Use in Production systems

- FPGAs are central to the system
- Will not move to ASICs
  - Reasons of volume or reprogrammability

Time-to-market	Fairly high; Fast compile times
Performance	Very critical
Volume	High per application



Use of FPGAs: Summary



Source: Gartner group

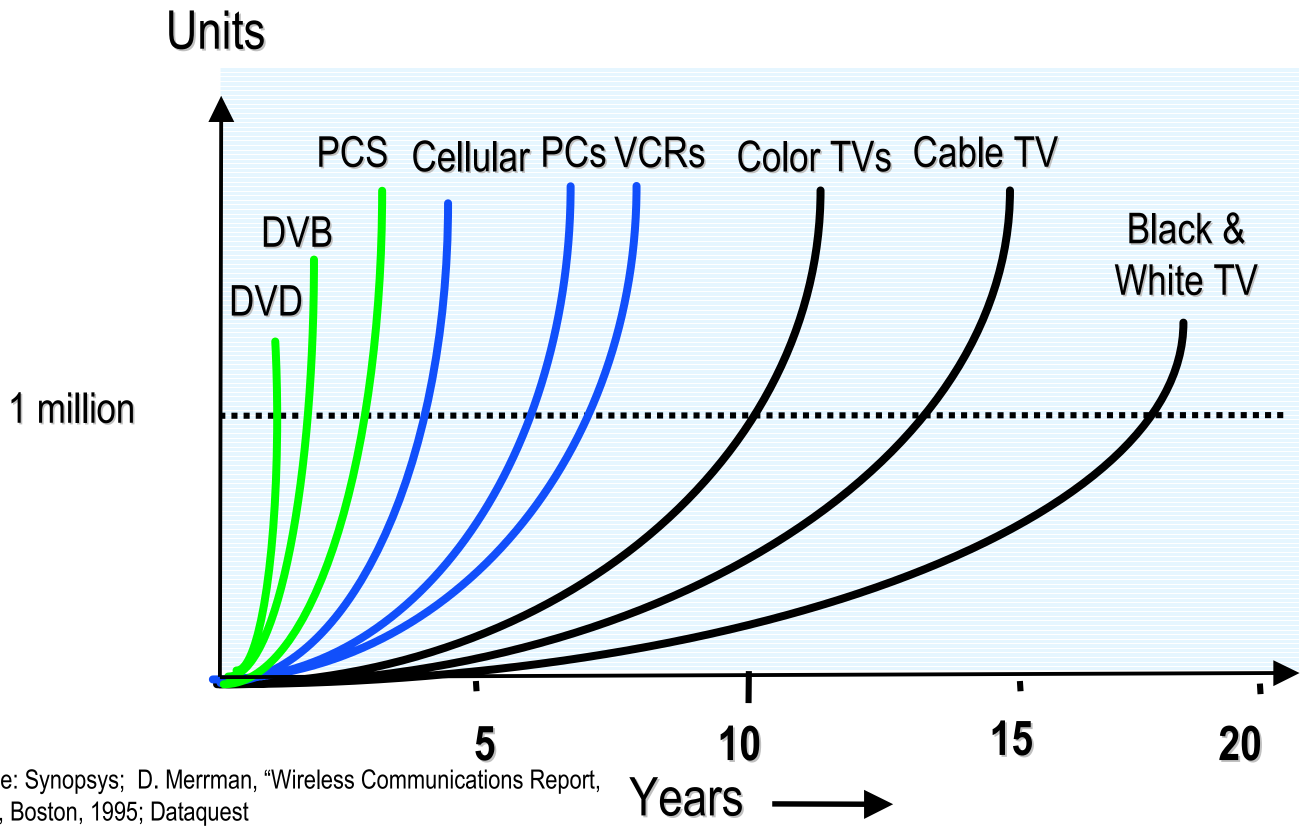
More than 2/3rds of FPGAs have stringent time-to-market requirements and critical performance requirements

Agenda

- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

XILINX®

Changing Industry Dynamics

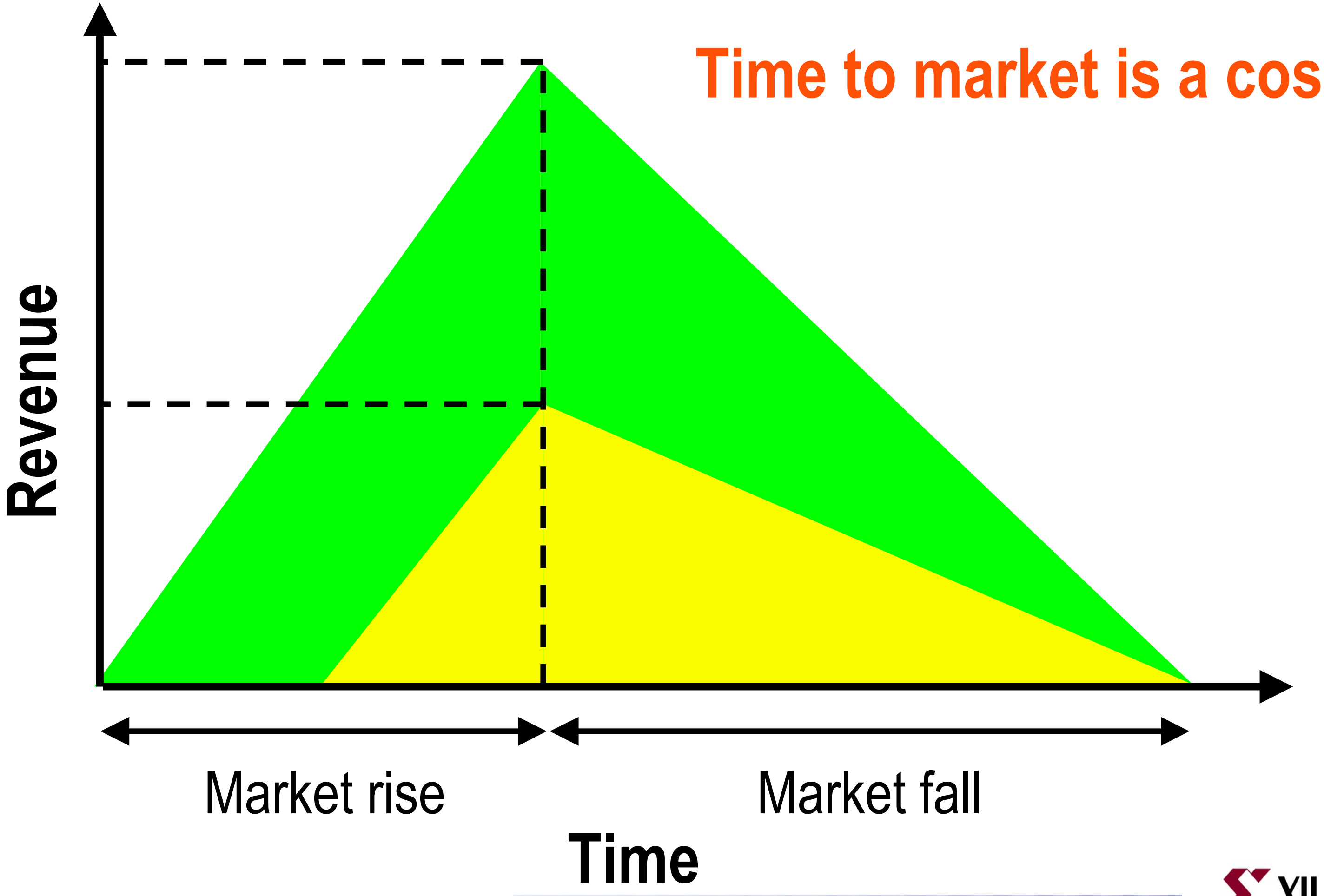


Source: Synopsys; D. Merrman, "Wireless Communications Report, " BIS, Boston, 1995; Dataquest

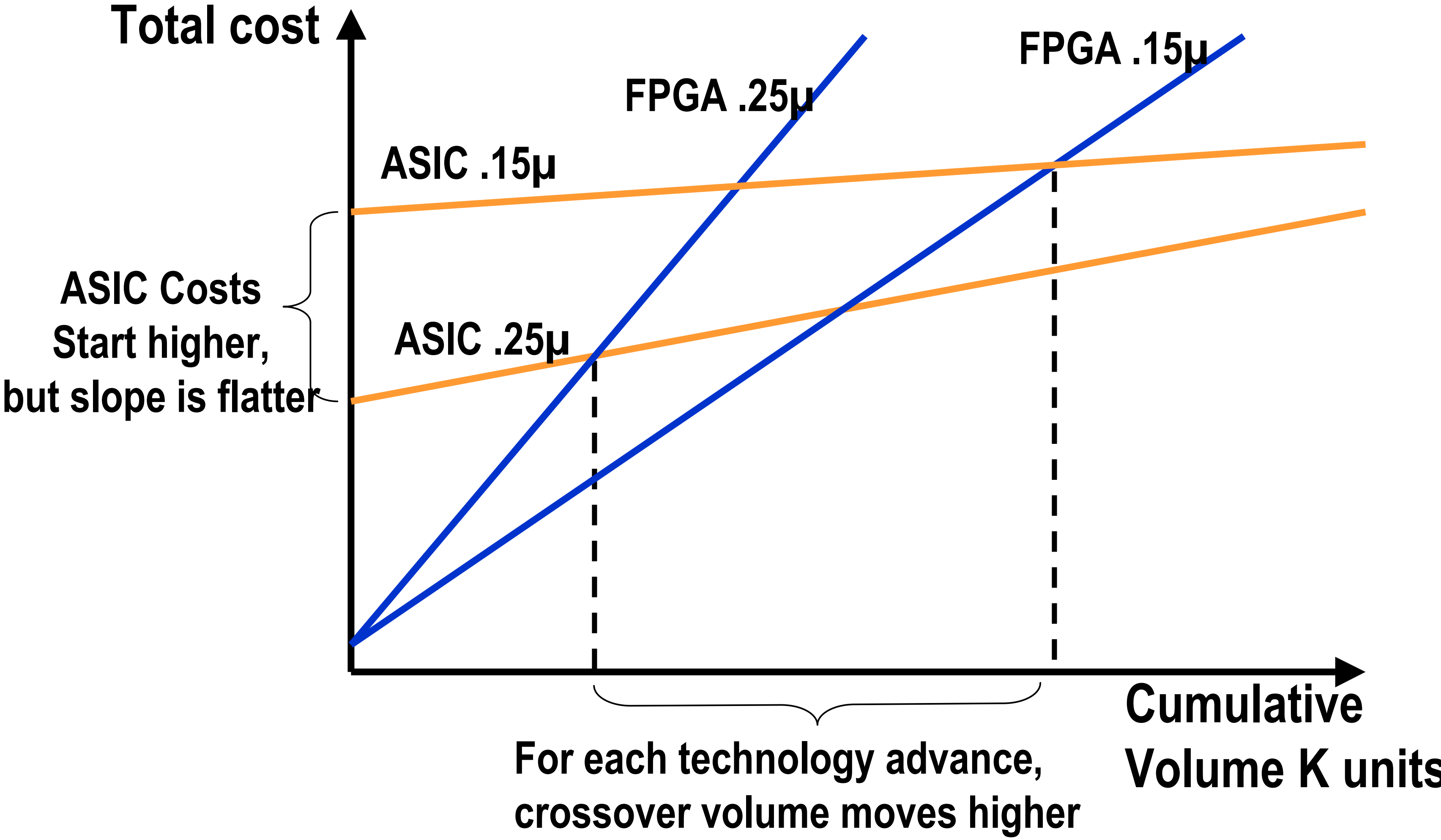
New products are taking less time to go in to volume.
New products also stay in volume for shorter periods.

XILINX®

Time to market is critical



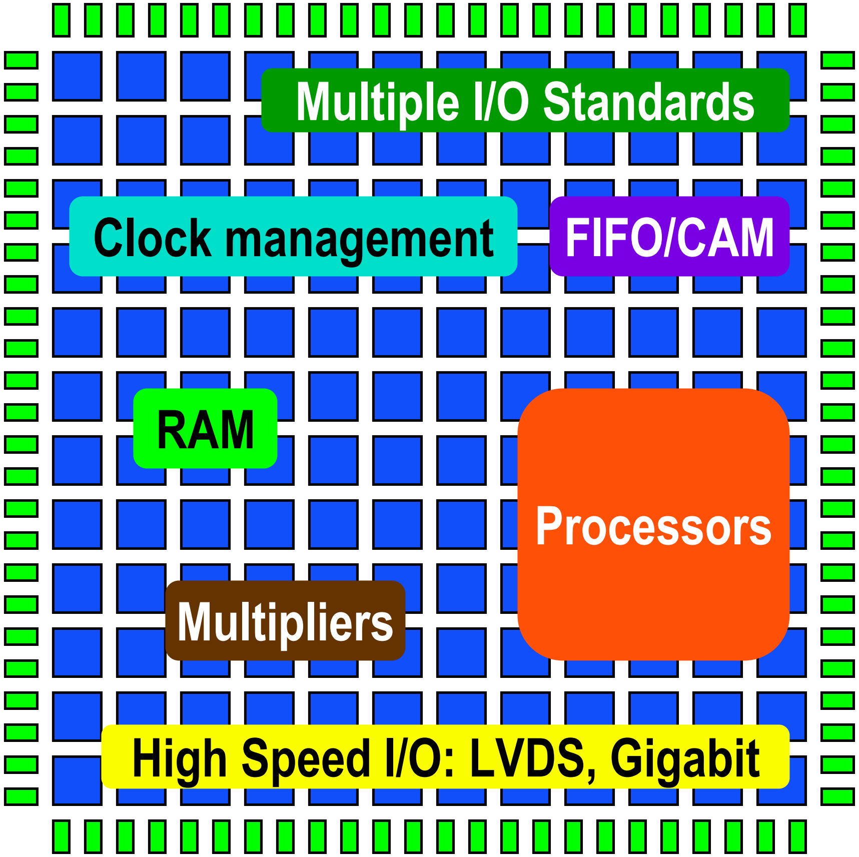
FPGA Unit Cost



ASIC Costs
Start higher,
but slope is flatter

For each technology advance,
crossover volume moves higher

FPGA device density and features



5 Years ago FPGAs were only gates and routing
~25000 gates

Today, there are several system-level features.
~10,000,000 gates

FPGAs have exploded in size and features

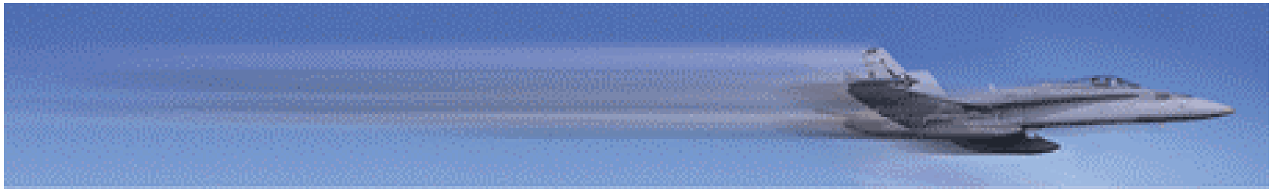
XILINX®

FPGA software

- Wide variety of tools available
  - From EDA vendors: Synthesis and verification
  - From FPGA vendors: Physical design tools and bitstream generation
- Ease of use
  - HDL to Bits
- Fast compile times
  - Million gates in less than an hour

Primary Goals of FPGA Software

- Time to market
  - Extremely fast compile times (HDL to bits)
  - Ease of use (no time to learn non intuitive flows)
- Design performance
  - Squeeze most performance out of FPGA

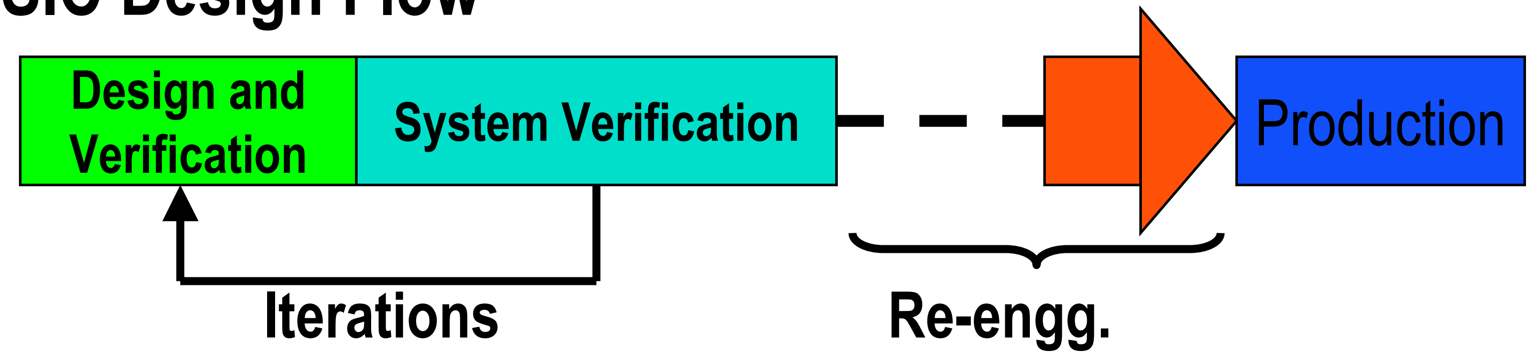


Agenda

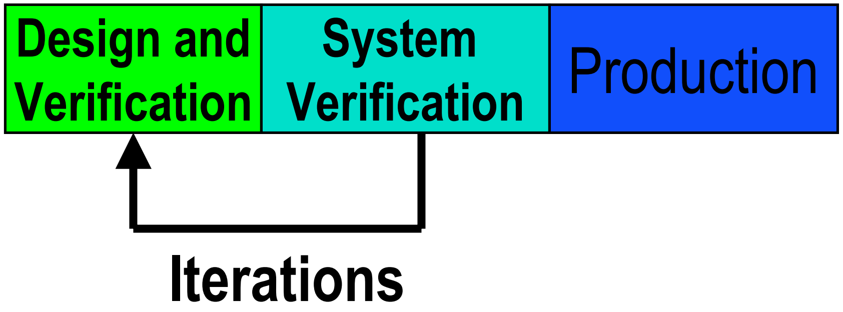
- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

FPGA vs ASIC design cycle

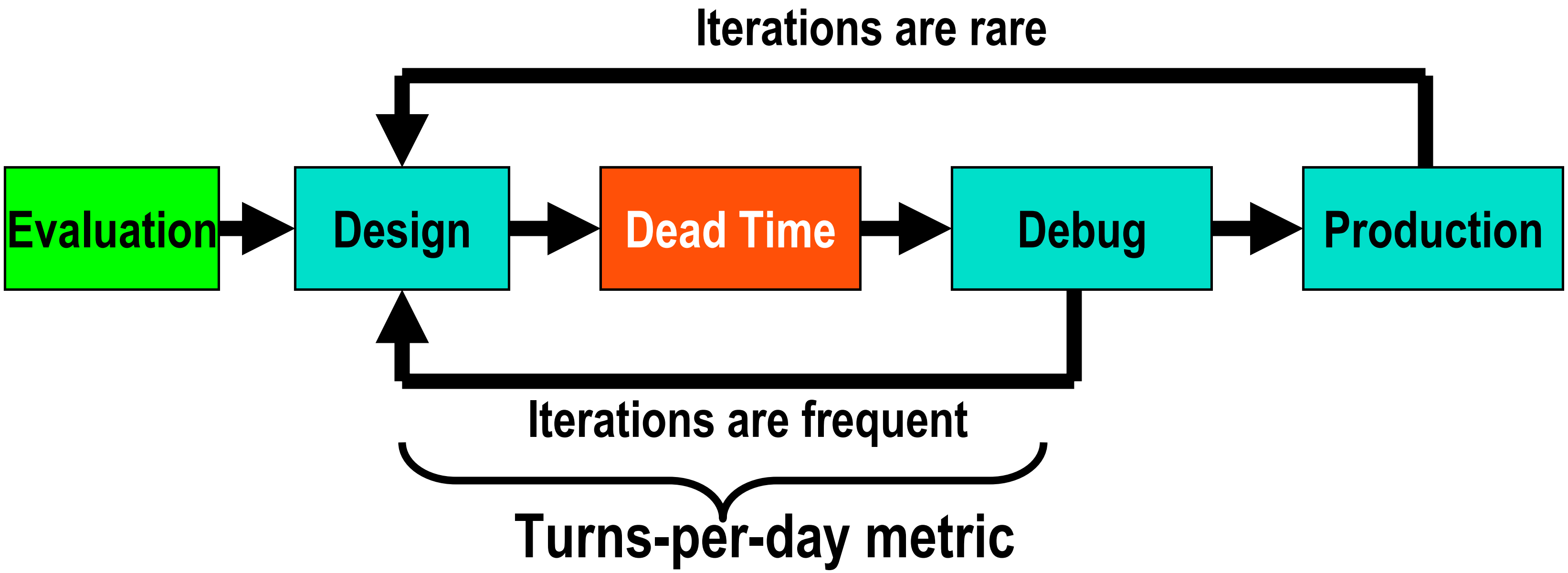
ASIC Design Flow



FPGA Design Flow



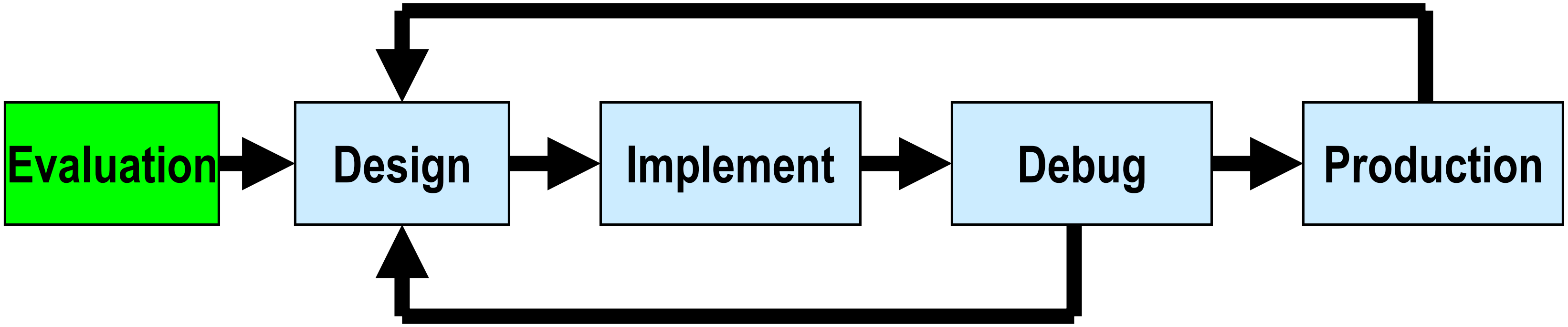
The FPGA Design Cycle



Dead time is really important because designers would rather be doing logic design or debugging

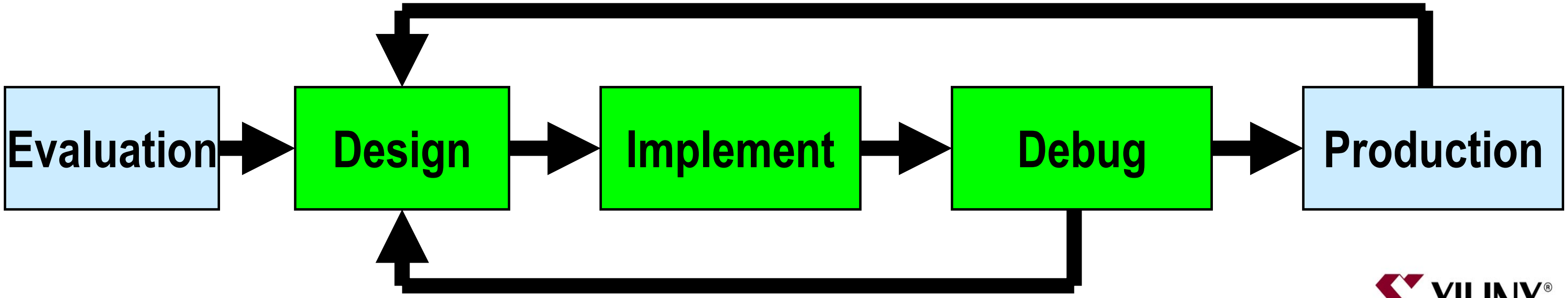
Software Requirements: Evaluation

- Software must be very fast
- Software must provide reasonably good estimates
  - Final design performance
  - FPGA device size for implementation (Cost)
  - Implementation time



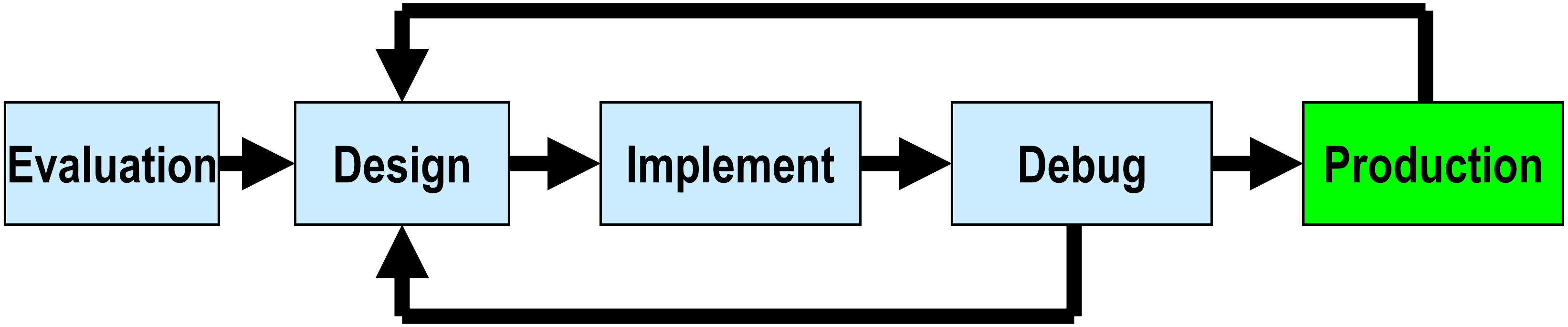
Software Requirements: Design/Debug

- Software must be fast
  - Fast implementation => Less dead time
- Must give reasonably good performance
  - Most compilations trade-off performance for faster runtimes
  - Final compilation is for best possible performance at the expense of runtime



Software Requirements: Production

- Late design changes and bugs are being fixed
  - Software must produce best possible performance
  - Cannot degrade performance or area
  - Runtime is not an issue



Deep Sub-Micron Effects

- FPGAs are process drivers
  - Latest process technology.
  - Process leaders
- Signal integrity
  - FPGAs designed with enough margin
  - Users don’t have to design around DSM effects
  - FPGA software does not have factor this in yet!

Deep Sub-Micron Effects

- Routing delay always dominates logic delay for FPGAs
  - Not process related
  - Routing delay = Several Programmable Switches (pass gates) + Several metal segments
- Address it with FPGA architecture
  - Key factor in determining architecture quality metrics
  - At least make it predictable

Software complexity

- ASIC designers are logic designers
  - Risk-averse and methodical
  - Spend lot of time verifying
  - Don't want to spend time in physical design
  - Separate engineers for physical design
- FPGA designers
  - Would rather debug on the bench
  - Realize must spend time in physical design
  - Expect physical design to be "hands-off"

Software should be simple and require minimal support

XILINX®

FPGA device quantization

- FPGAs available only in certain sizes
  - 10 devices from 3,000 LUTs to 122,000 LUTs for Virtex-II
- Marginal cost of using an extra LUT or routing resource is zero
  - Marginal cost jumps if the design does not fit the device
- Area minimization may not always be a factor
  - Use to FPGA advantage
  - Logic replication may be free while it costs in ASICs

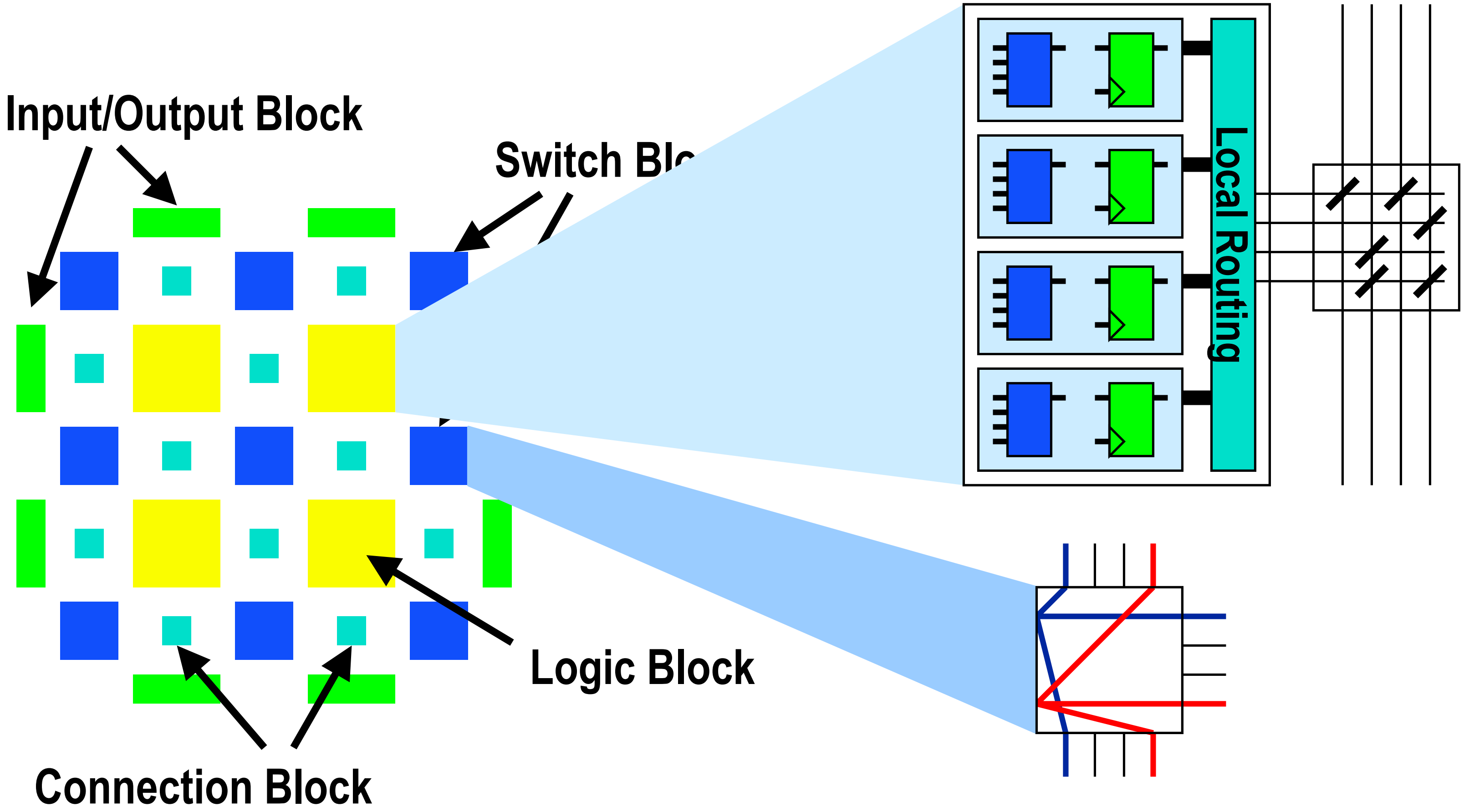
New architecture development

- FPGA Architectures primarily evaluated by CAD tools
- A feature that cannot be supported by CAD tool is very often not added to an architecture
- FPGA software is available at least 6 months before FPGA silicon

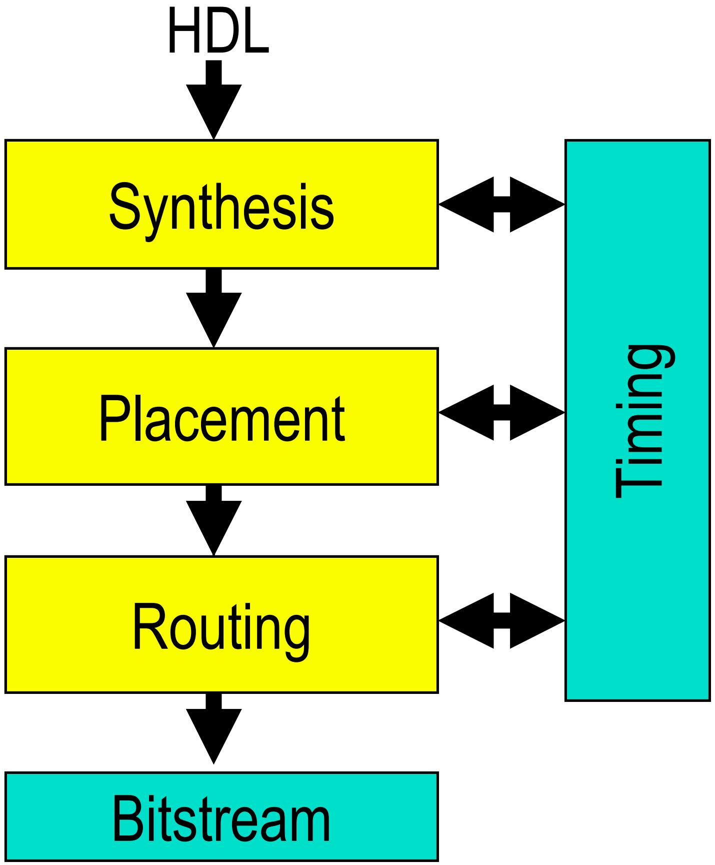
Agenda

- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

Traditional FPGA Architecture



FPGA Design Implementation Flow



Synthesis:
Input: HDL, target FPGA arch.
Output: Logic elements (LUTs, FFs), I/O

Placement:
Input: Logic elements, FPGA device
Output: Placed logic elements

Routing:
Input: Placed logic elements
Output: Switch programming

Agenda

- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

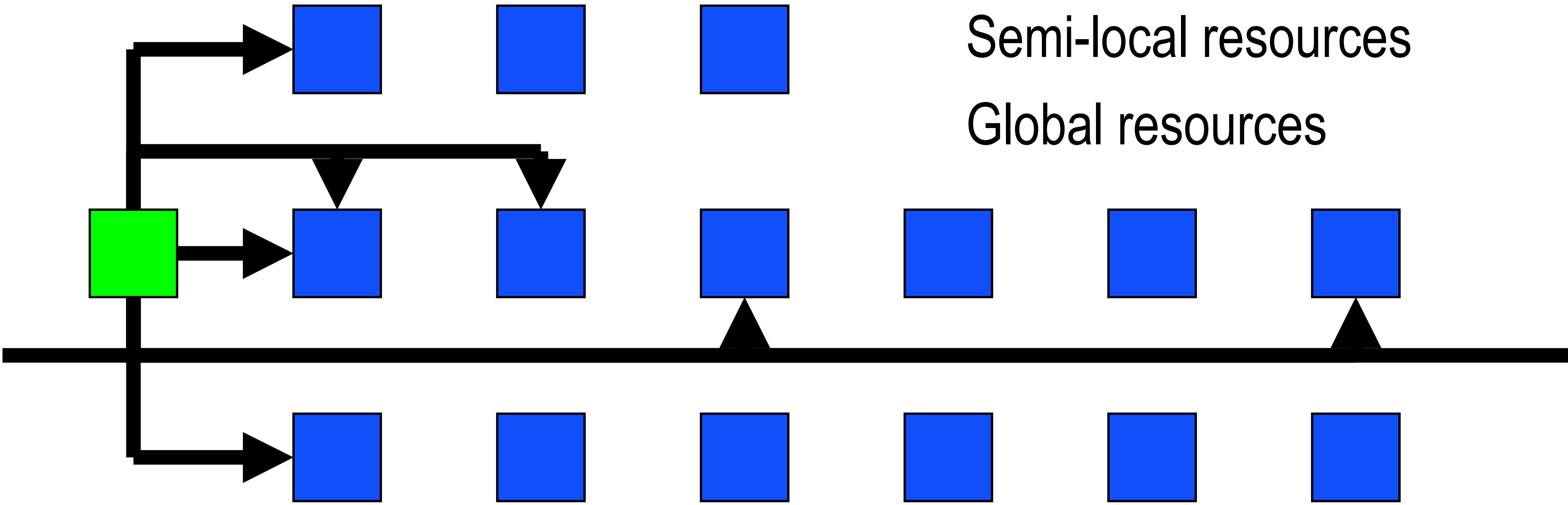
Placement

- Placement problem is very similar to ASICs
  - Lot fewer movable objects
  - 10M FPGA ~ 300,000 movable elements
- Standard metrics and algorithms
  - Simpler metrics work best
  - Bounding box, cut numbers, simple congestion metrics
- Estimating delays during placement
  - Easier than ASICs
  - Finite set of routing resources

Delay Estimation In Placement

- ASICs
  - RC tree analysis for proposed route
  - Computationally expensive
  - Very little pre-computation
- FPGAs
  - Fixed set of most likely routes
  - Pre-computed delays for routes
  - Architecture makes delay predictable

Delay Estimation for FPGAs



Local resources

Semi-local resources

Global resources

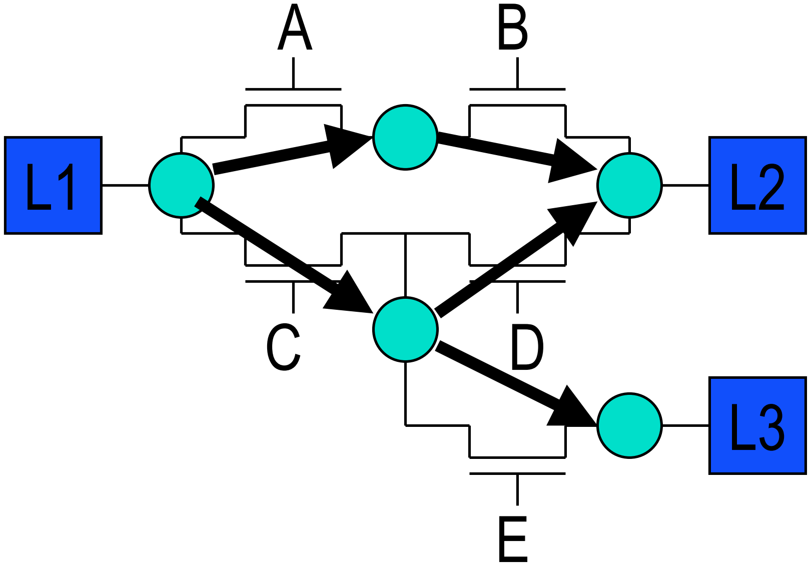
Delay Estimation in FPGA Placement

- Architecture dictates routing connections for optimal results
- Critical signals must use architectural recipe
- Non-critical signals may use other routes

Agenda

- FPGAs
  - How are they used?
  - Why are they used?
- ASICs and FPGAs
  - What is different?
  - What implications are there for Physical Design
- FPGA Physical Design
  - FPGA architecture
  - Placement
  - Routing
- Conclusions

Routing Model for FPGAs

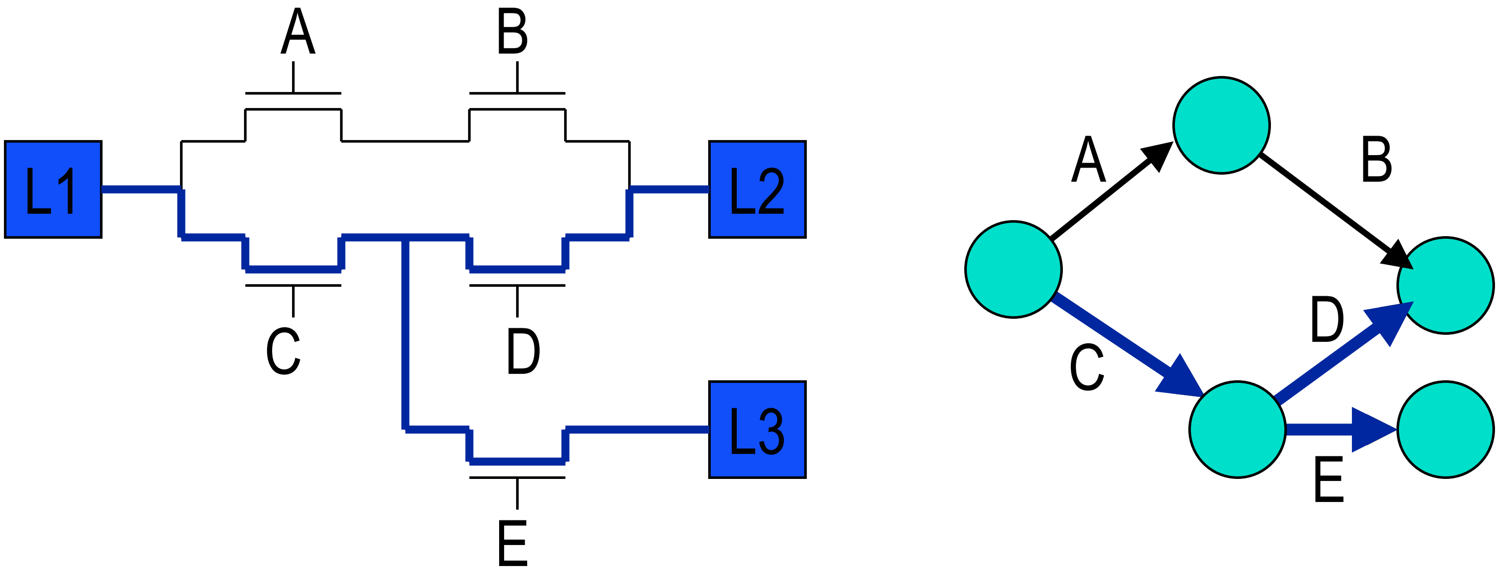


Architecture represented as a routing connectivity graph.

Conductor segments are nodes

Programmable point are arcs

Routing Model for FPGAs

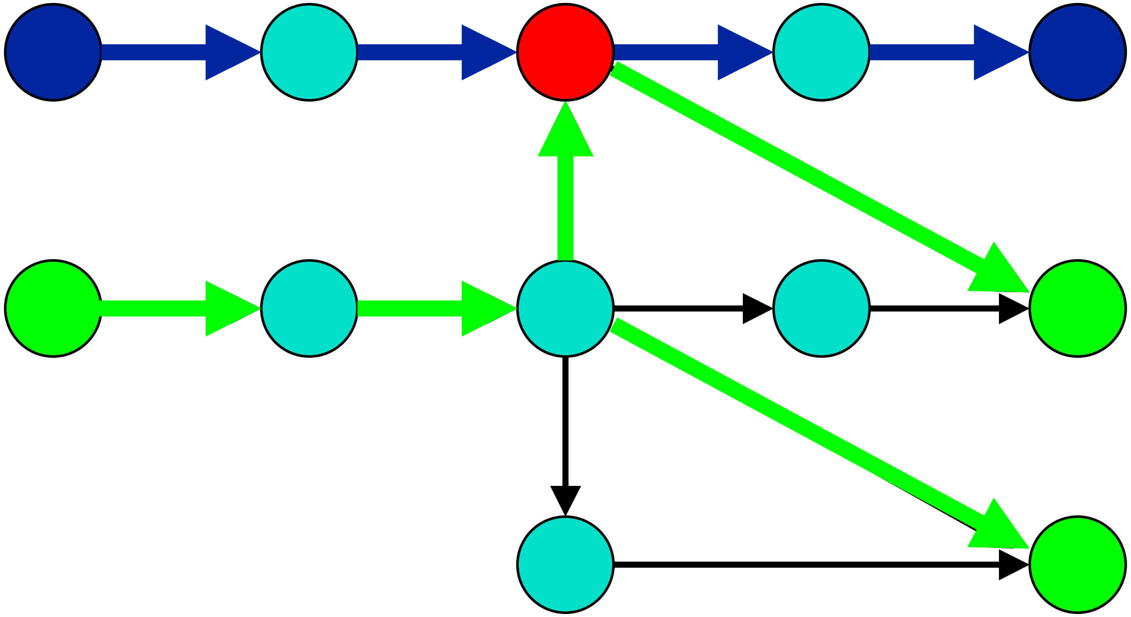


Rectilinear grid-based graph routing algorithms must be modified

FPGA Routing Algorithms

- No distinct global and detailed routing phase
  - Regions cannot be separated and routed independently
  - Channel routing algorithms don’t work
- Global resource assignment phase
  - Use architectural insights
- Maze routing can be used
  - Side effect of contention is a serious problem

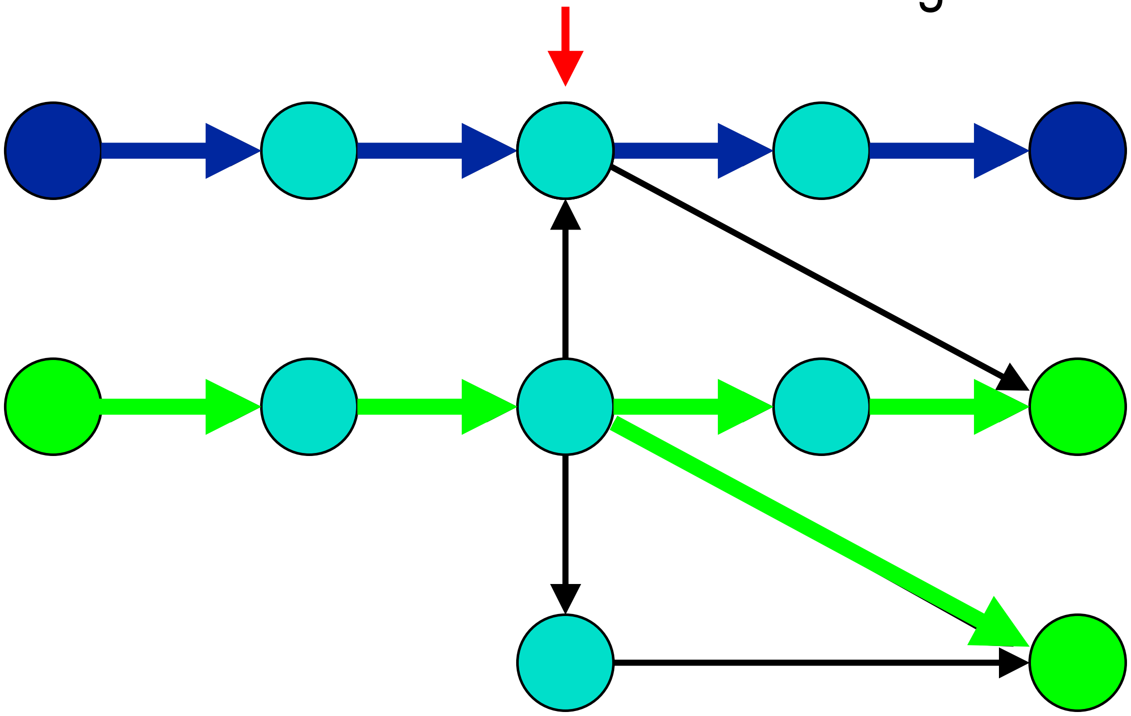
PathFinder algorithm



Routing is unsuccessful due to resource contention!

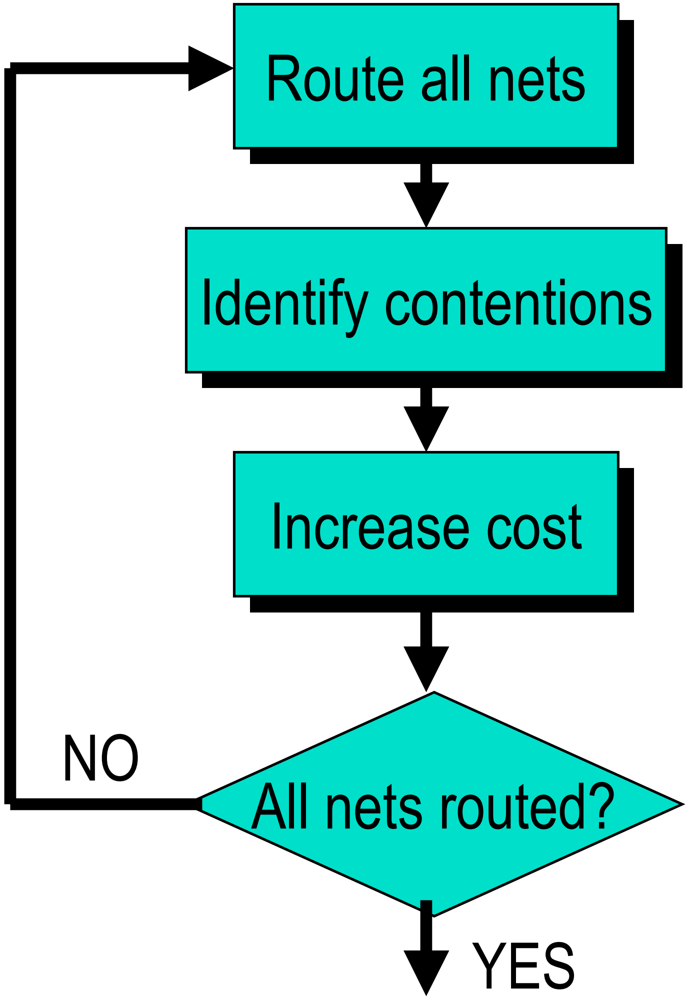
PathFinder Algorithm

Increase the weight of this resource



Routing is now successful!

PathFinder Algorithm



- Route the nets individually
- Identify the contentions and increase the cost monotonically
- Repeat until all nets are routed or too many iterations have been done.

PathFinder Algorithm

- The individual nets are routed using:
  - Some variation of maze
  - Table lookup
- Each individual net can be routed very fast
  - Absence of obstacles
- Accounts for resource contention better than standard maze routing

Other FPGA Routing Approaches

- Basic idea of embedding a routing on a connectivity graph
- Interesting approach to formulate problem as a Boolean Satisfiability Problem
- Solution to the SAT problem gives a feasible routing

Physical Synthesis for FPGAs

- Wire-load models are still the preferred method
  - Used to be very inaccurate
  - Architectural improvements have reduced error
- A recent approach to FPGA physical synthesis
  - Run through place/route once
  - Re-synthesize based on previous results
  - Good experimental results
  - Oscillations could be a problem

Conclusions and Future Work

- FPGAs are in the mainstream
- Physical design is similar in FPGAs and ASICs
  - Differences make for interesting twist to existing algorithms
- FPGA-based algorithms research
  - Currently, we just borrow from ASICs
  - Use architecture for different formulation
- Speed, Speed and more Speed