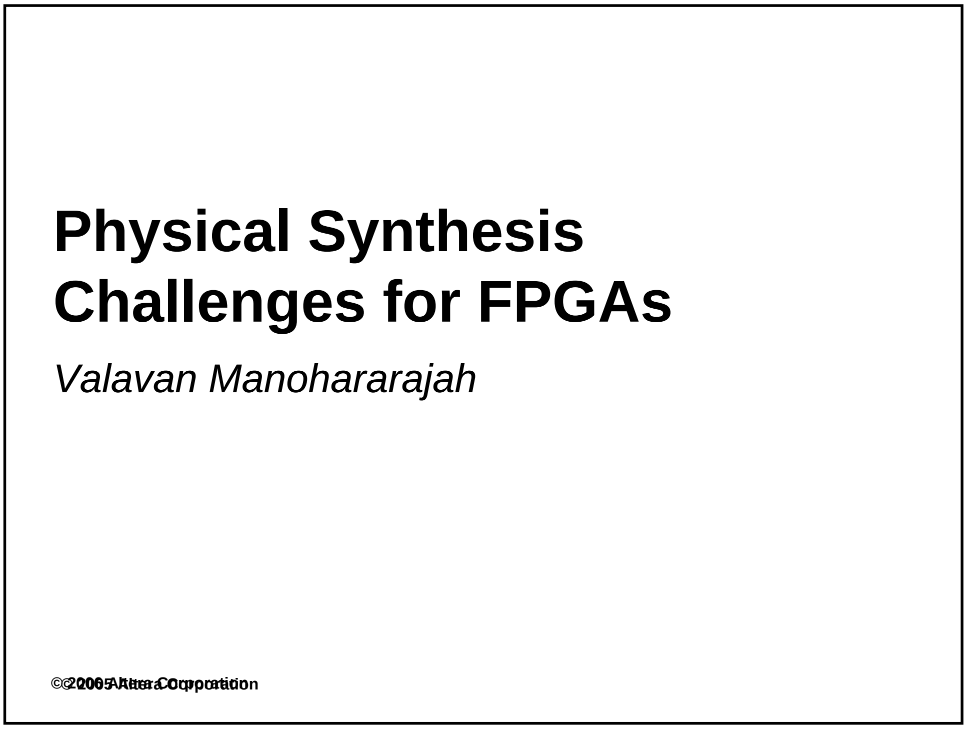
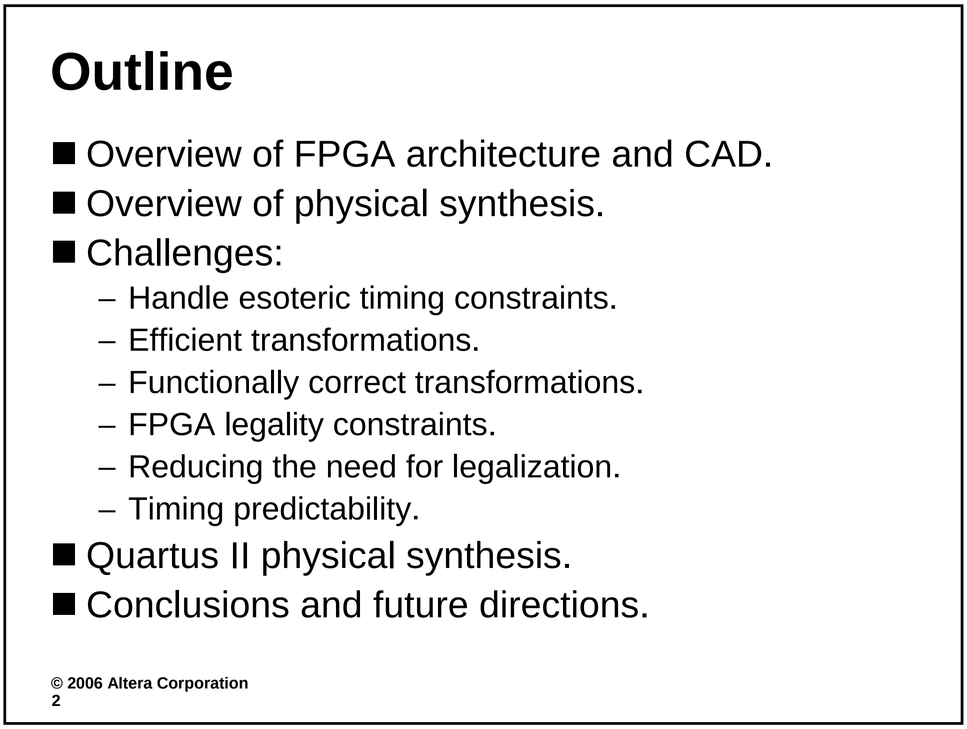


Physical Synthesis Challenges for FPGAs

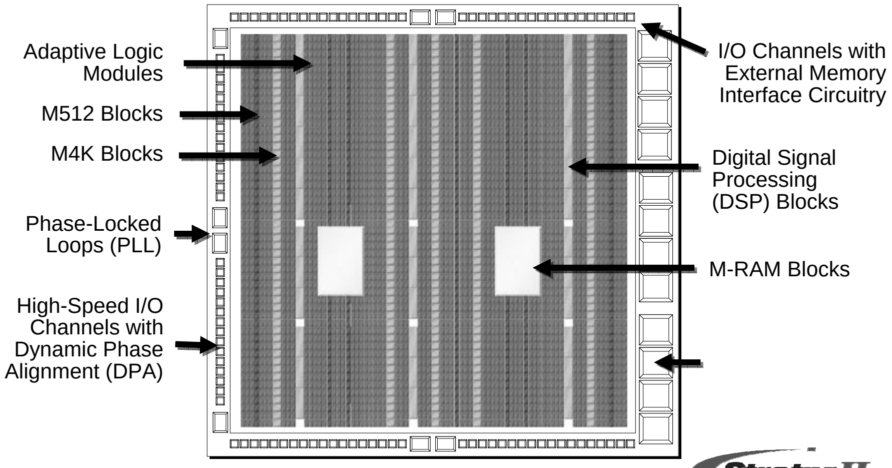
Valavan Manohararajah



Outline

- ■ Overview of FPGA architecture and CAD.
- ■ Overview of physical synthesis.
- ■ Challenges:
  - – Handle esoteric timing constraints.
  - – Efficient transformations.
  - – Functionally correct transformations.
  - – FPGA legality constraints.
  - – Reducing the need for legalization.
  - – Timing predictability.
- ■ Quartus II physical synthesis.
- ■ Conclusions and future directions.

A Modern FPGA



Adaptive Logic Modules

M512 Blocks

M4K Blocks

Phase-Locked Loops (PLL)

High-Speed I/O Channels with Dynamic Phase Alignment (DPA)

I/O Channels with External Memory Interface Circuitry

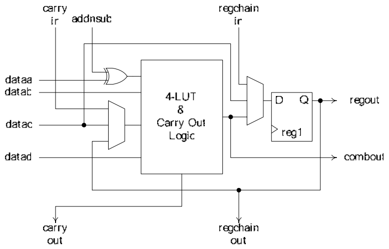
Digital Signal Processing (DSP) Blocks

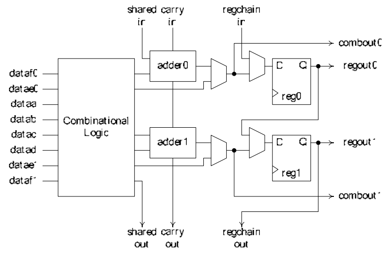
M-RAM Blocks

180K LEs, 384 DSP, 9Mb RAM, >40 clocks, 20 SERDES….

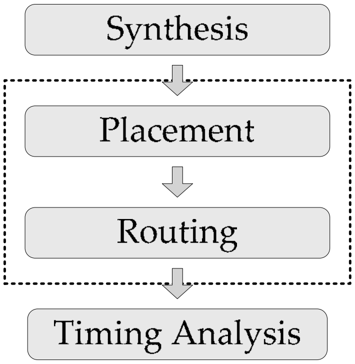
FPGA Logic Elements

- LUTs + FFs
- Arithmetic circuitry.
- Stratix:
  - 4-LUT
- Stratix II:
  - Up to 6-LUTs.
  - Two functions of up to 8 inputs.





The Traditional FPGA CAD flow

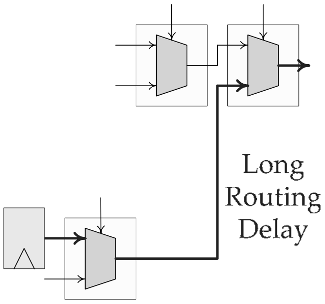


- Synthesis is a disjoint step from P & R.
- Circuit structure is created without knowledge of routing delays.

Synthesis "Mistakes"

- Routing delays make up a large percentage of the total delay on the critical path (> 60%).
- Traditional timing-driven logic synthesis may create sub-optimal circuitry when routing delays are considered.
- Hard to predict where the actual critical path is going to be without P & R info.

Synthesis "Mistakes"



- Consider a possible placement of a balanced multiplexer-tree.
- After P & R, it is evident that a skewed version of the multiplexer tree would be better to compensate for long critical routing delays.

The Physical Synthesis Solution

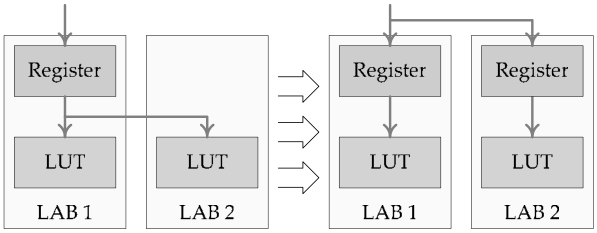
- Use delay information from P & R to guide timing driven optimizations.
- Several possible flows:
  - A big loop that feeds P & R delays back into synthesis (MULTI-PASS).
    - May not converge.
    - Lots of compile time for each run through the loop.
  - Use some form of timing prediction to guide logic synthesis (EARLY).
  - Small incremental changes that are tightly integrated with placement (LATE).

Timing Driven Logic Restructuring

- Logic Replication
  - Duplicate registered and combinational logic to reduce routing delays.
- Combinational Re-synthesis
  - Restructure critical regions of logic so that critical signals go through fewer levels of logic.
- Register Retiming
  - Moves registers to shorten the delay of timing-critical paths.

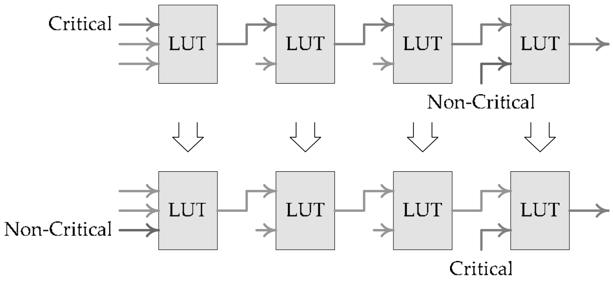
Logic Replication

- Large fanouts may prevent critical signals from being placed in an optimal fashion.
- Logic Replication can help with these problems.



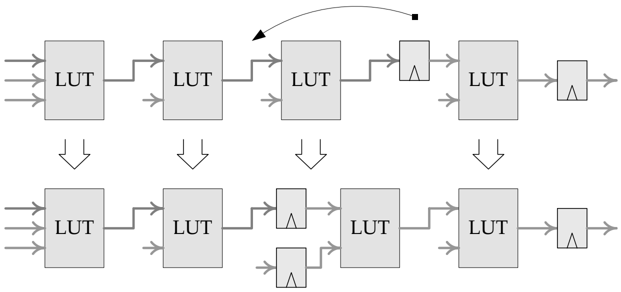
Combinational Re-synthesis

- Identify timing-critical signals and restructures the circuit so that the signal goes through fewer hops.



Register Retiming

- Registers can be moved across logic elements to shorten timing-critical paths.



Challenge #1: Timing Constraints

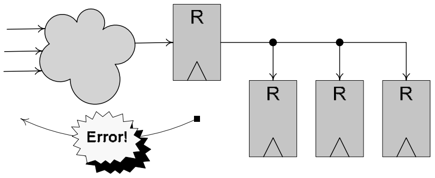
- There are several types of timing constraints that a user may have:
  - Multiple clocks & inverted Clocks
  - IO timing
  - Clock Skew
  - Multi-cycle & cut-path constraints
  - …
- Restructuring operations must keep these in mind and must target the most critical circuitry.

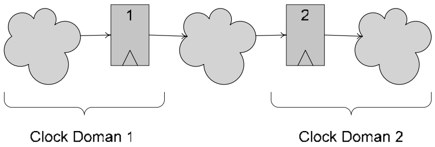
Challenge #2: Efficient Transforms

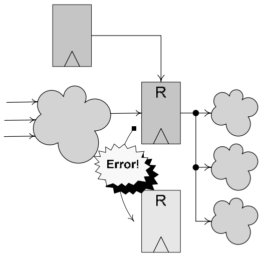
- Optimal retiming is polynomial time solvable:
  - $O(n^2\log(n))$.
  - Leiserson & Saxe (1983).
- Too slow for industrial circuits.
- Some attempts to speed up algorithm, but bound unchanged.
  - N. Maheshwari & S. Sapatnekar (1998).
  - N. Shenoy & R. Rudell (1994).

Challenge #3: Correctness

Have to be very careful when modifying asynchronous paths.







Challenge #4: Legality Constraints

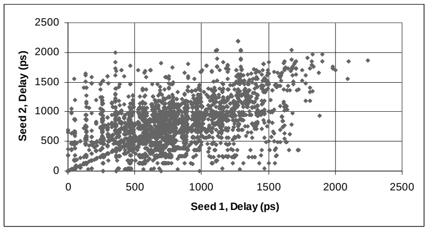
- FPGA architecture imposes several constraints.
  - Finite number of signals and logic elements at a cluster location.
  - To use certain structures, circuit has to be structured in a specific way. For example:
    - Carry chains.
    - Grouping constraints on LUT pairs in an ALM.

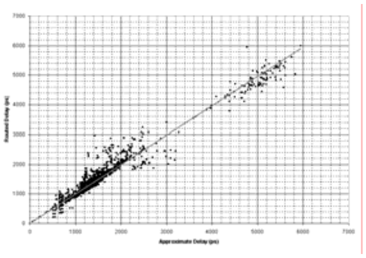
Challenge #5: Cost of Legalization

- Making changes to a circuit after placement will lead to illegalities:
  - – Violate signal constraints.
  - – Overuse logic elements.
- Task of legalization:
  - – Integrate changes into existing circuit.
  - – But don’t modify existing circuit too much.
- Can be a very time consuming step:
  - – 65% of physical synthesis time.

Challenge #6: Timing Predictability

- Very difficult to predict timing before P & R has occurred.
- Predictability is better after placement, but not perfect.



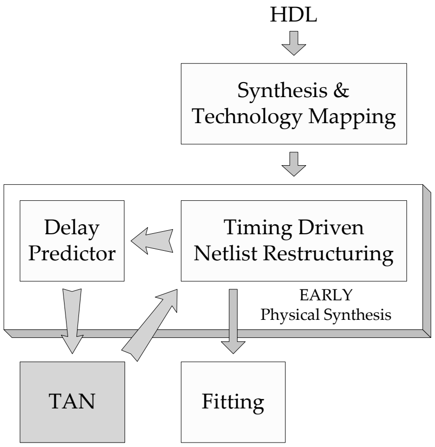


Quartus II Physical Synthesis

- Uses both EARLY and LATE Physical Synthesis techniques (Two-Pass).
- EARLY Physical Synthesis
  - – Occurs after synthesis, but before placement.
  - – Lots of freedom to restructure the circuit.
  - – Hard to predict delays.
- LATE Physical Synthesis
  - – Occurs after placement, but before routing.
  - – Less freedom to restructure the circuit.
  - – Much easier to predict delays.

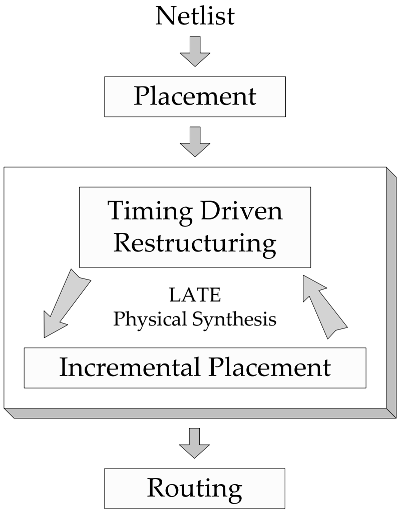
EARLY Physical Synthesis

- Early physical synthesis consists of a sequence of netlist optimization routines that are guided by a delay prediction engine.


HDL
Synthesis & Technology Mapping
Delay Predictor
Timing Driven Netlist Restructuring
EARLY Physical Synthesis
TAN
Fitting

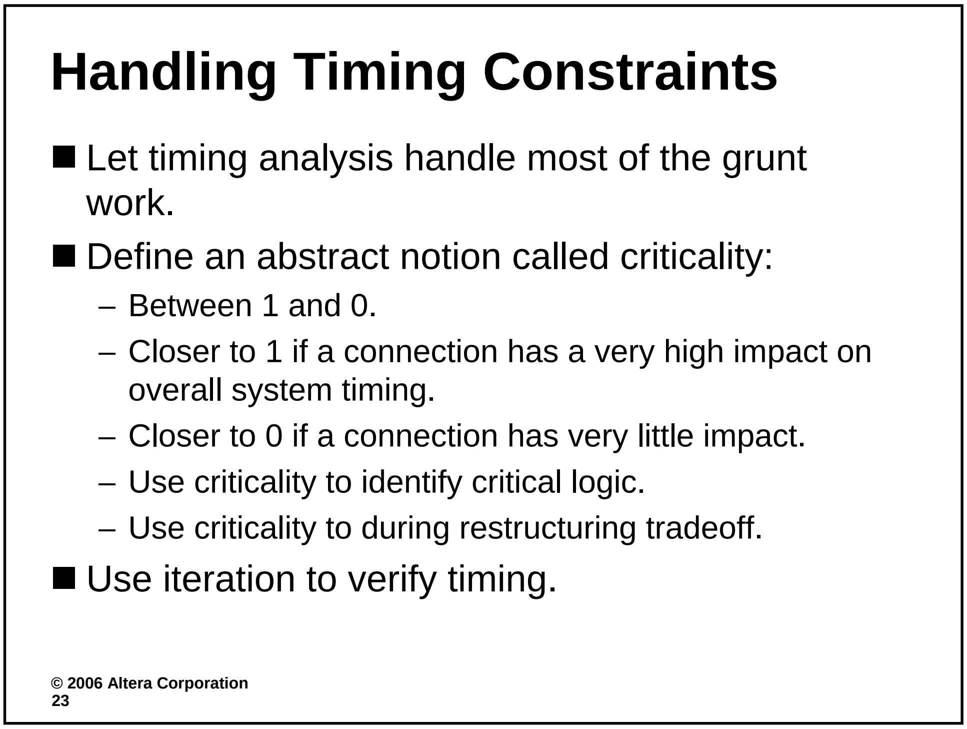
LATE Physical Synthesis

- Runs between P & R.
- Static timing analysis is used to determine the paths that are critical.
- Incremental netlist modifications are made to improve the structure of critical regions of logic.
- Logic elements may be modified and/or added.
- Inc. placement is used to integrate new logic.



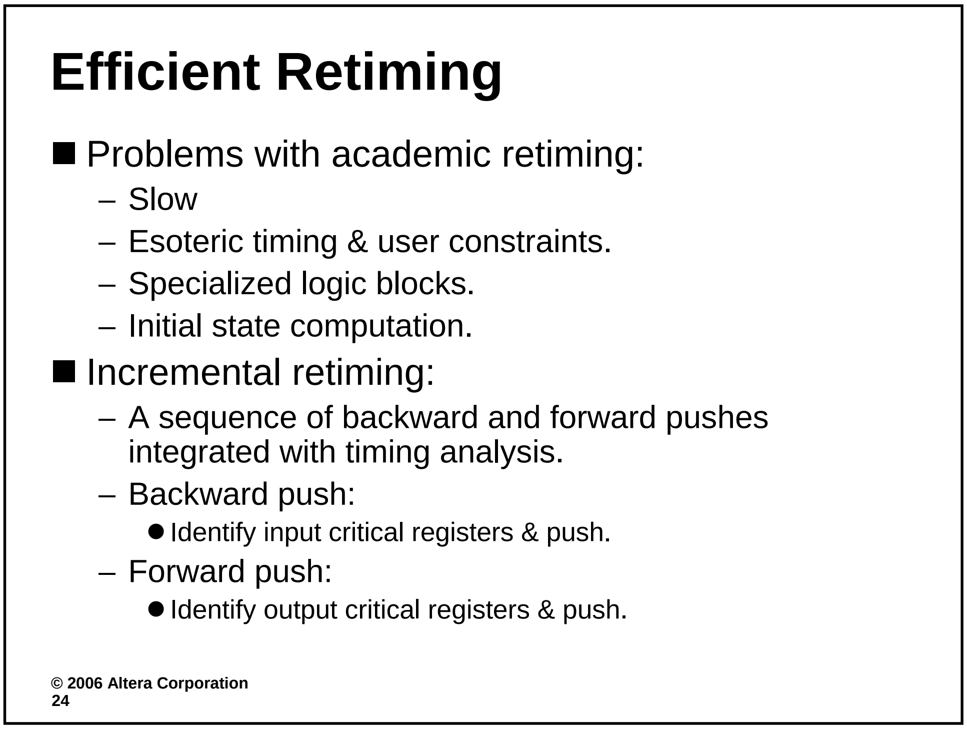
Benefits of the Two Phase Approach

- Reduces the need for legalization
- Can attempt more extensive early restructuring.
- Gives placement a better starting point.



Handling Timing Constraints

- Let timing analysis handle most of the grunt work.
- Define an abstract notion called criticality:
  - Between 1 and 0.
  - Closer to 1 if a connection has a very high impact on overall system timing.
  - Closer to 0 if a connection has very little impact.
  - Use criticality to identify critical logic.
  - Use criticality to during restructuring tradeoff.
- Use iteration to verify timing.



Efficient Retiming

- Problems with academic retiming:
  - Slow
  - Esoteric timing & user constraints.
  - Specialized logic blocks.
  - Initial state computation.
- Incremental retiming:
  - A sequence of backward and forward pushes integrated with timing analysis.
  - Backward push:
    - Identify input critical registers & push.
  - Forward push:
    - Identify output critical registers & push.

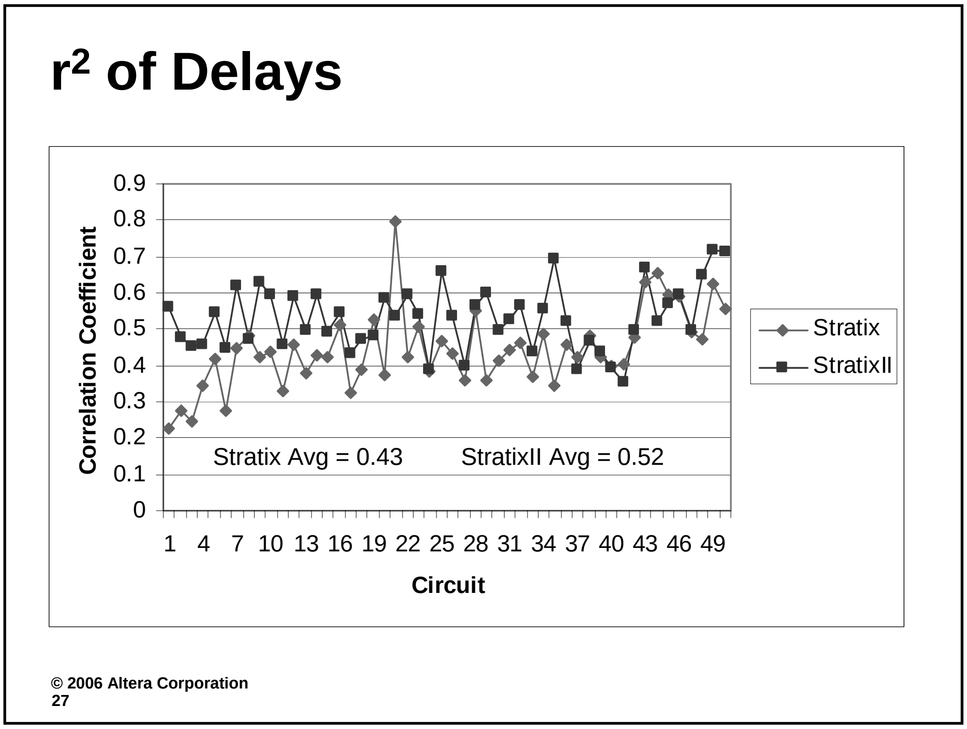
Comparing Optimal vs Incremental

Circuit	# 4-LUTs	Incremental(ns)	Optimal(ns)	Original(ns)	Gain(Opt)	Gain(Inc)
alu2	197	24	23	43	87.0%	79.2%
mult32a	116	12	12	96	700.0%	700.0%
s9234.1	461	28	28	32	14.3%	14.3%
mm9a	142	68	68	71	4.4%	4.4%
s838	167	32	32	35	9.4%	9.4%
oc_cordic	1513	4	4	7	75.0%	75.0%
elliptic	3602	32	32	72	125.0%	125.0%
s38584.1	6281	36	36	39	8.3%	8.3%
frisc	3539	33	32	92	187.5%	178.8%
hc11	3860	80	80	120	50.0%	50.0%
fip_des	15388	32	31	36	16.1%	12.5%

- MCNC and OpenCore designs.
  - Circuits are simple with no special FPGA features.
- Very close or the same in most cases!

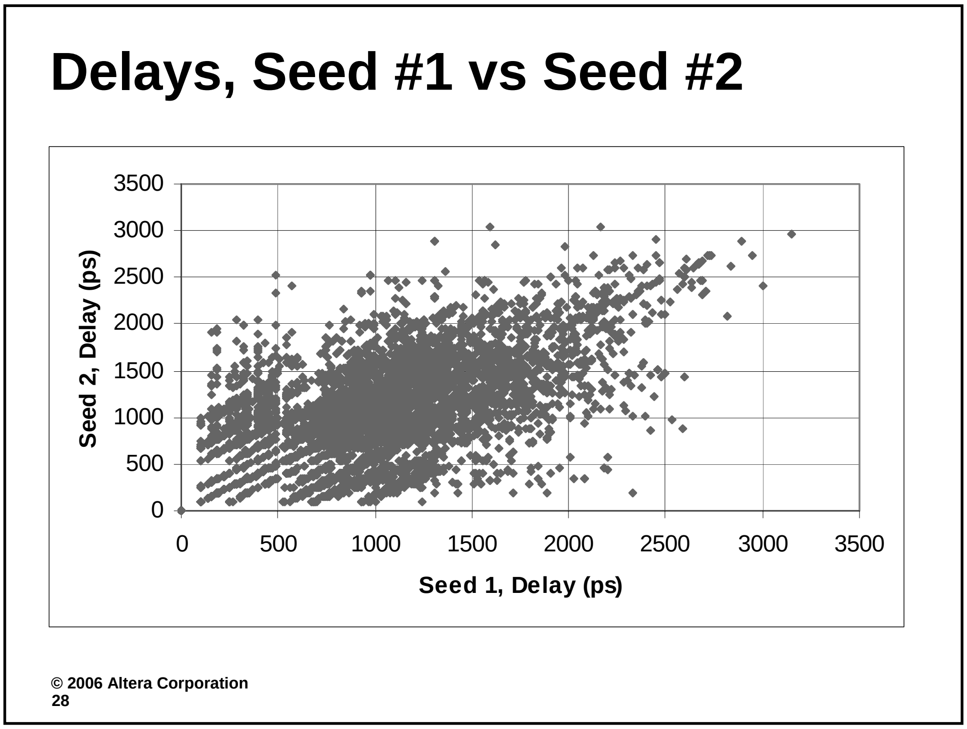
Timing Predictability

- Behavior of the QuartusII placer can be changed by choosing a different seed.
- Determine an upper bound on the predictability:
  - Run placer with two different seeds.
  - First seed is to be a predictor for the second.
- Measure predictability using correlation coefficient (r^2).
- Focus on the critical connections:
  - Fitter pays more attention to the critical connections.
  - Observe connections with a criticality of 0.5 or higher.



r^2 of Delays

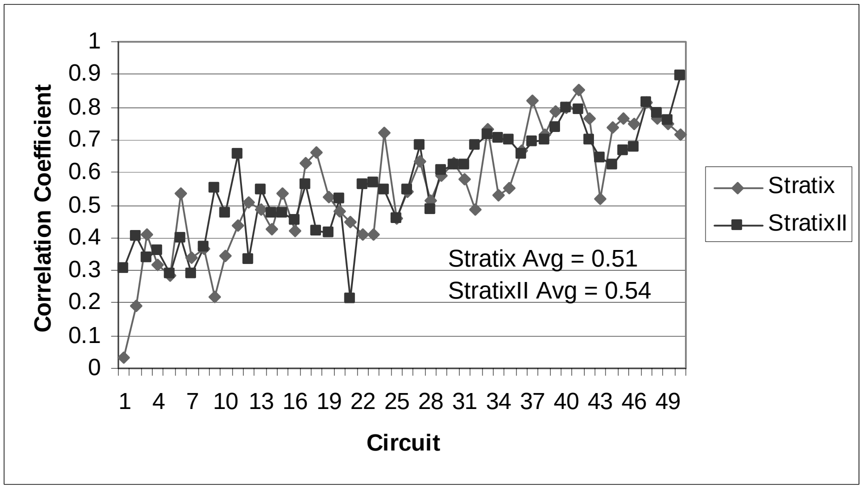
© 2006 Altera Corporation
27



Delays, Seed #1 vs Seed #2

© 2006 Altera Corporation
28

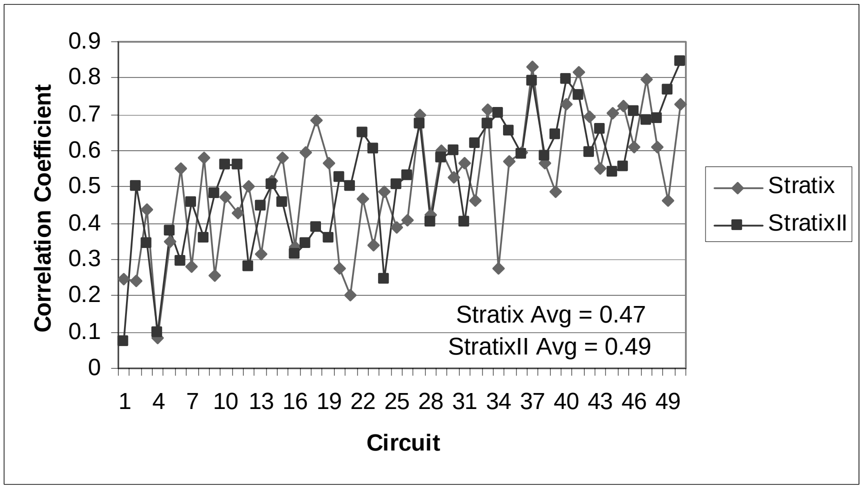
r^2 of Criticalities



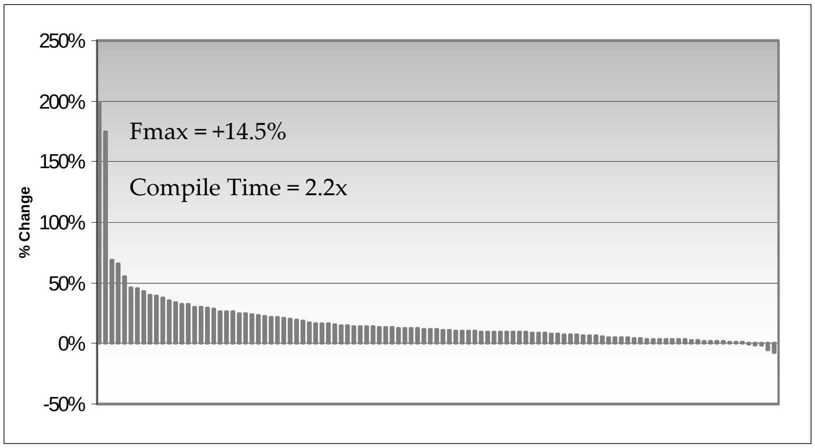
A Simple Early Timing Model

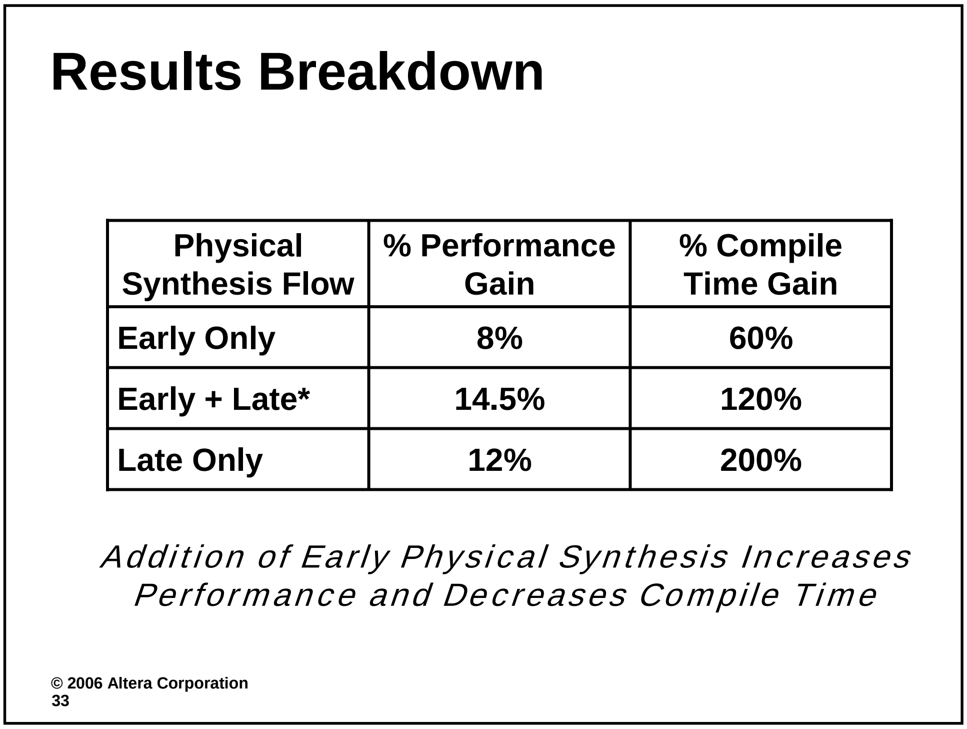
- For each connection type gather observed delays over several circuits.
- Connection type identified by:
  - – Driving node type.
  - – Driving port type.
  - – Driven node type.
  - – Driven port type.
- Compute a weighted delay for each connection type, placing more emphasis on on faster observed delays.

Simple Timing Model Correlation



Current Results

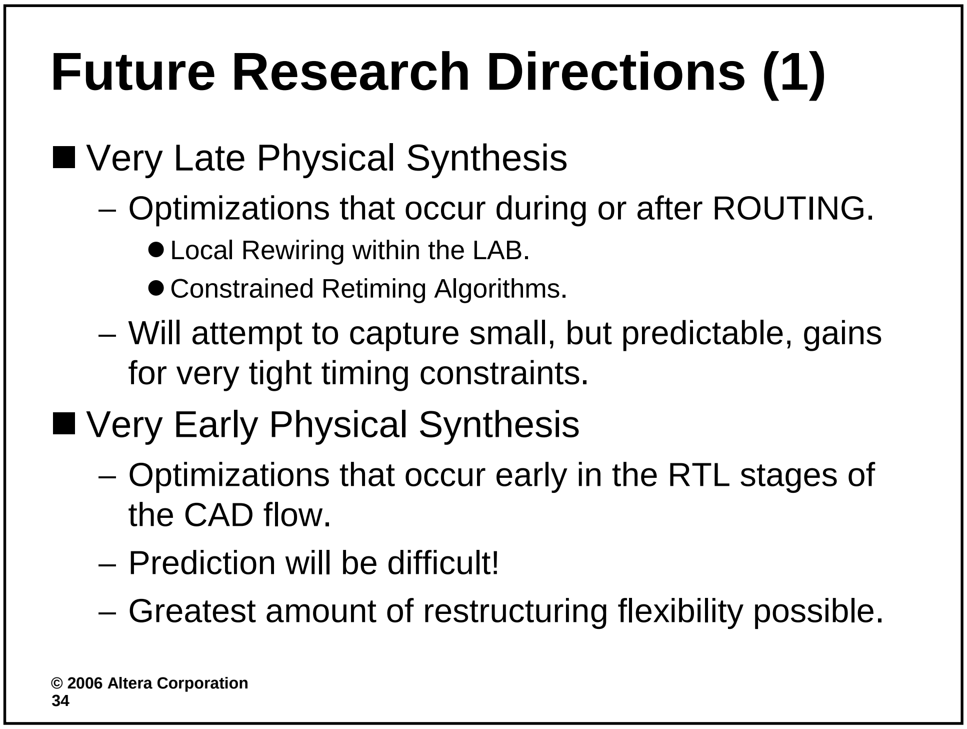




Results Breakdown

Physical Synthesis Flow	% Performance Gain	% Compile Time Gain
Early Only	8%	60%
Early + Late*	14.5%	120%
Late Only	12%	200%

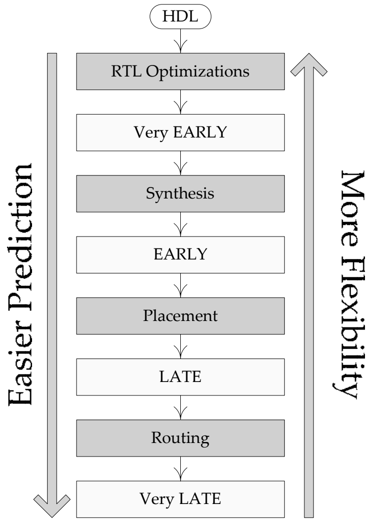
Addition of Early Physical Synthesis Increases Performance and Decreases Compile Time



Future Research Directions (1)

- Very Late Physical Synthesis
  - – Optimizations that occur during or after ROUTING.
    - Local Rewiring within the LAB.
    - Constrained Retiming Algorithms.
  - – Will attempt to capture small, but predictable, gains for very tight timing constraints.
- Very Early Physical Synthesis
  - – Optimizations that occur early in the RTL stages of the CAD flow.
  - – Prediction will be difficult!
  - – Greatest amount of restructuring flexibility possible.

Future Research Directions (2)



HDL

RTL Optimizations

Very EARLY

Synthesis

EARLY

Placement

LATE

Routing

Very LATE

Easier Prediction

More Flexibility

- Key problem:
  - The points in the CAD flow with the greatest restructuring flexibility offer the least amounts of predictability!