Architectural Enhancements in Stratix V^{TM}

David Lewis, David Cashman, Mark Chan, Jeffrey Chromczak, Gary Lai, Andy Lee, Tim Vanderhoek, Haiming Yu

Altera Corp

Overview

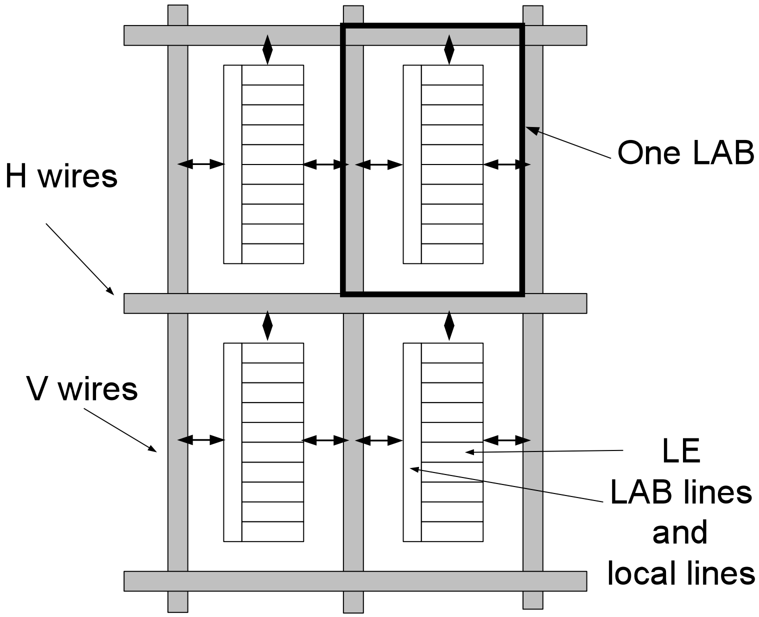
- Background on Stratix architecture and terms
- Flip flop enhancements
- Memory architecture
- Routing modifications
- Fast adder
- Summary

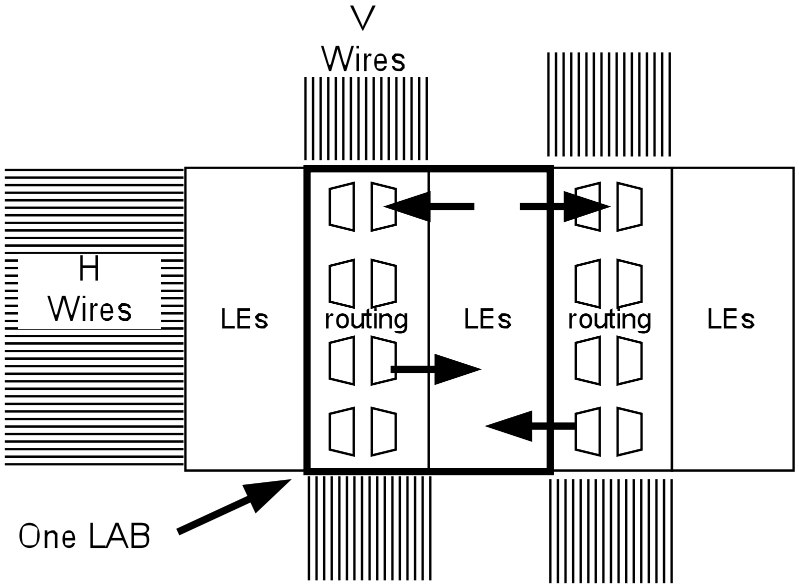
Stratix Overview

- Logic Array Block (LAB) means the logic elements (LEs) and the routing
- LEs in Stratix II and later are adaptive logic modules (ALMs) containing LUTs and FFs
- ALM contains a fracturable 6 LUT that can also implement two 5 LUTs or four 4 LUTs feeding adders
- Shared inputs limit the ALM to 8 input signals
  - 5 LUTs must share 2 inputs

LAB: Logic and Routing

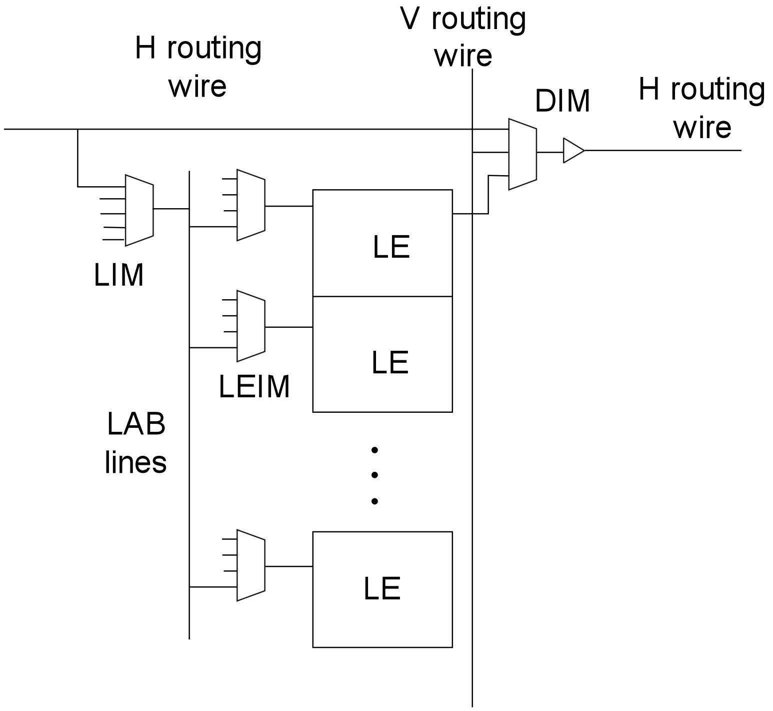
- LAB has inputs and outputs to 3 channels: 2V and one H
- Works well with routing wires over routing mux pool, and LE has access to 2 mux pools

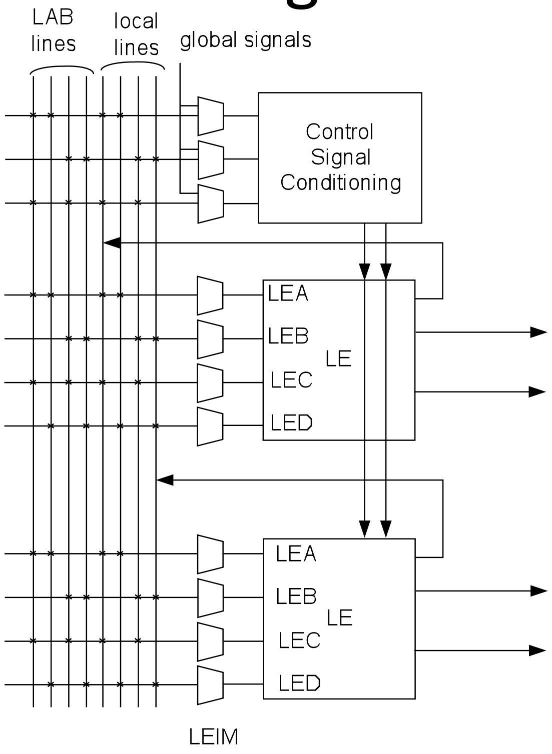




Routing Details

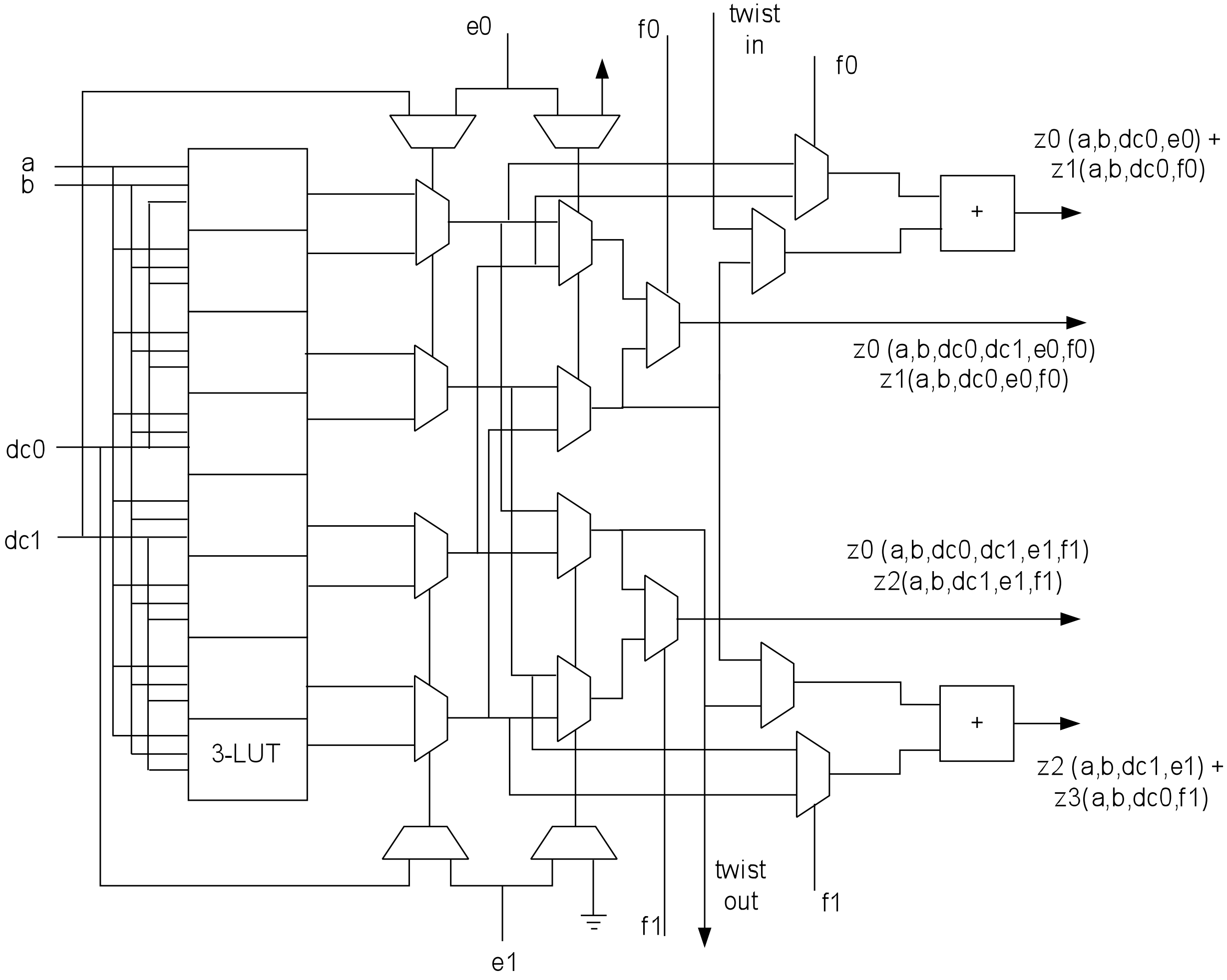
- Mux based routing
  - DIM (driver input mux) drives routing wires
  - LIM (LAB input mux) drives LAB lines; LEIM drives LE
- Partial population of internal LAB routing





ALM Logic

- Yikes
- There are actually reasons for all of these features



Flip-Flop Count and Design

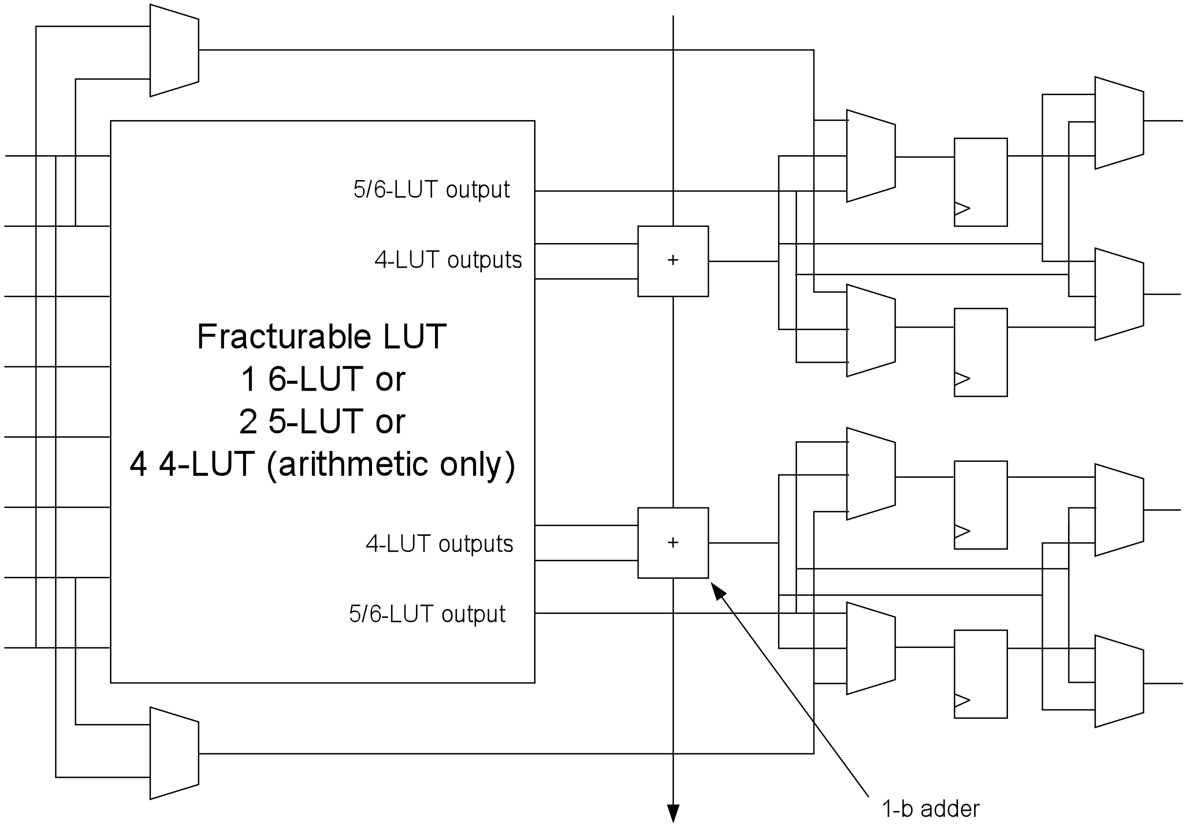
- FF count relative to LUTs is trending higher as designers introduce more pipelining
- Pipelining also increases importance of fast FFs as they contribute fixed overhead to each path
- Stratix V introduces 4 FF per ALM and time borrowing while preserving conventional FF model

Extra FFs

- Although trend towards FF density increasing, typical designs have FF:LUT not much >1 and only some designs are highly FF intensive
- Add FFs to improve density of FF intensive designs with minimal cost impact for designs that don’t use them
- Avoid adding more intra-LAB routing by sharing input pins with LUT

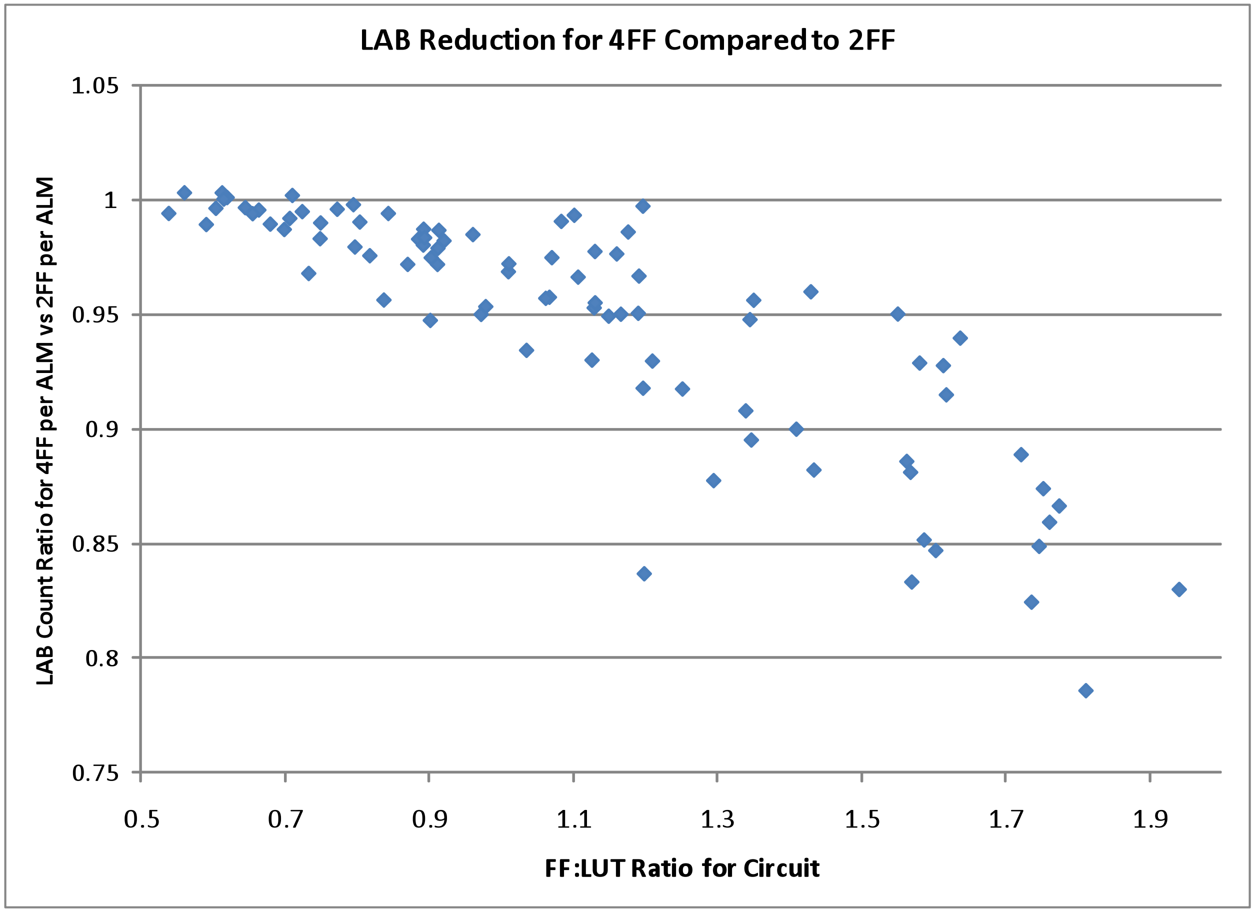
Extra FF

- 4 FF but sharing inputs to ALM and clocking between pairs



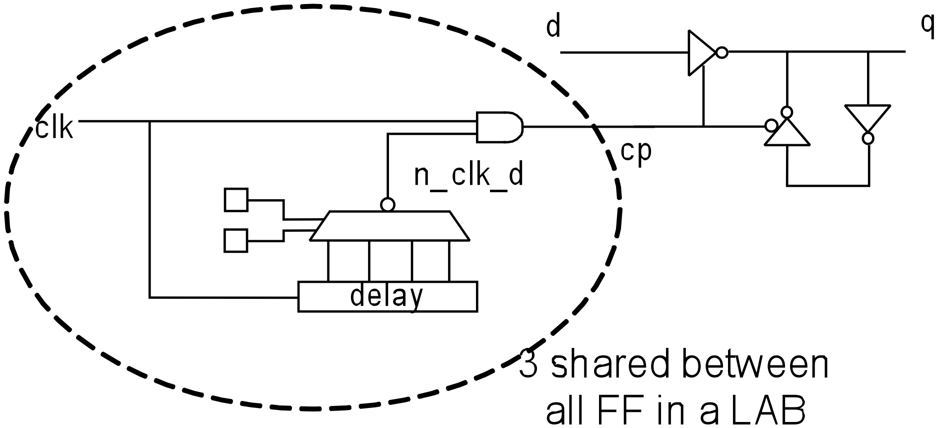
Extra FF Benefits

- 7% fewer LABs on average; 10% for FF:LUT > 1

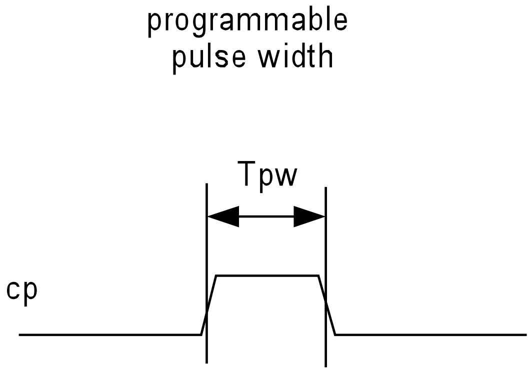


Improving FF Cost and Performance

- Pulse latches are fast and cheap
  - Eliminate one stage of master-slave FF
  - Eliminate delay of one stage of MS FF
  - Enable time borrowing
- 3 pulse generators are shared between all FF in a LAB

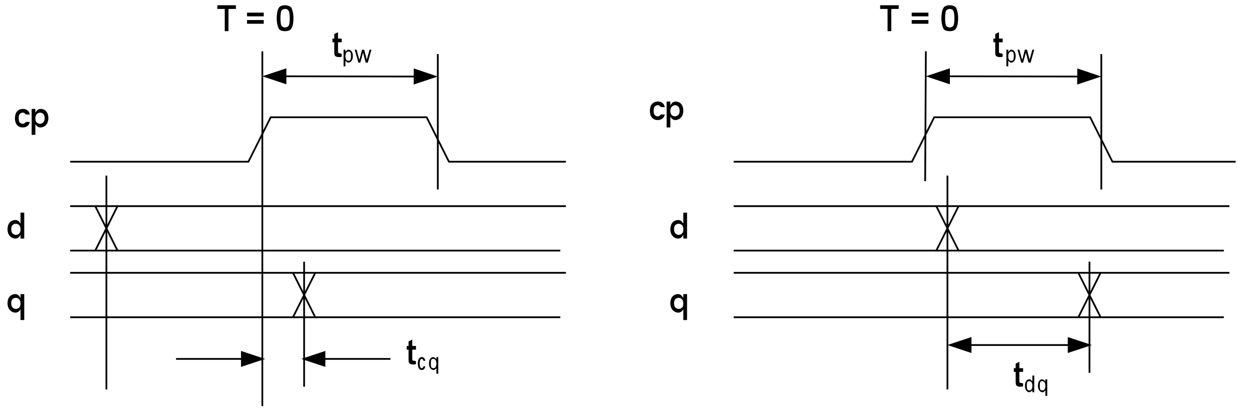


3 shared between all FF in a LAB



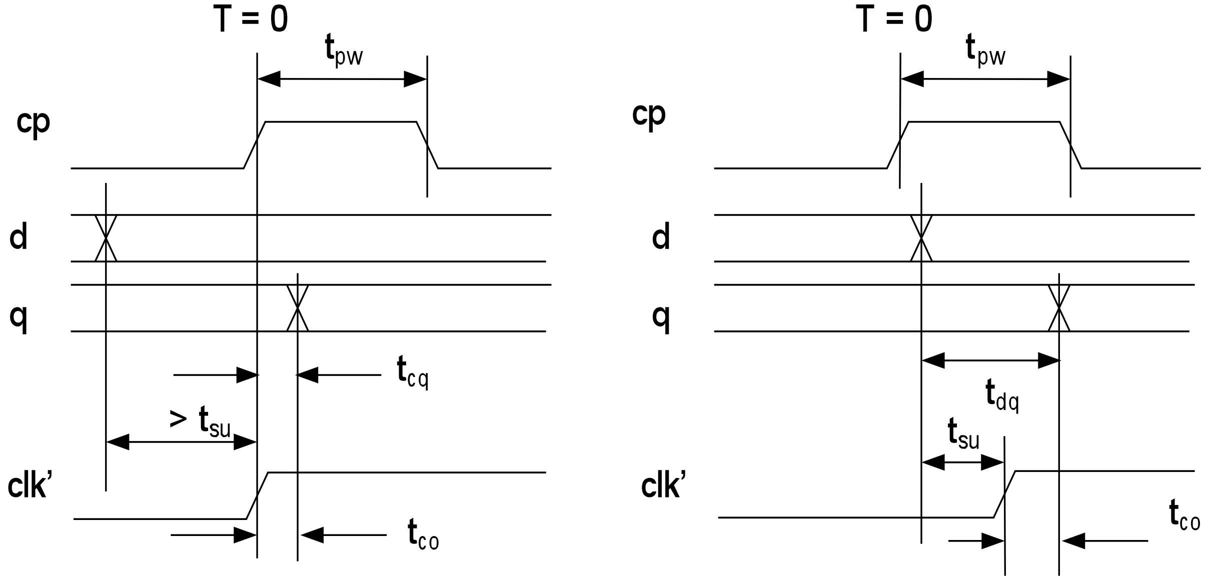
Time Borrowing Flip Flop

- Consider level sensitive latch driven by pulse generator
  - If data arrives early with respect to clock pulse, then q launches at clock edge + Tcq
  - If data arrives late with respect to clock pulse, then q launches at data arrival + Tdq



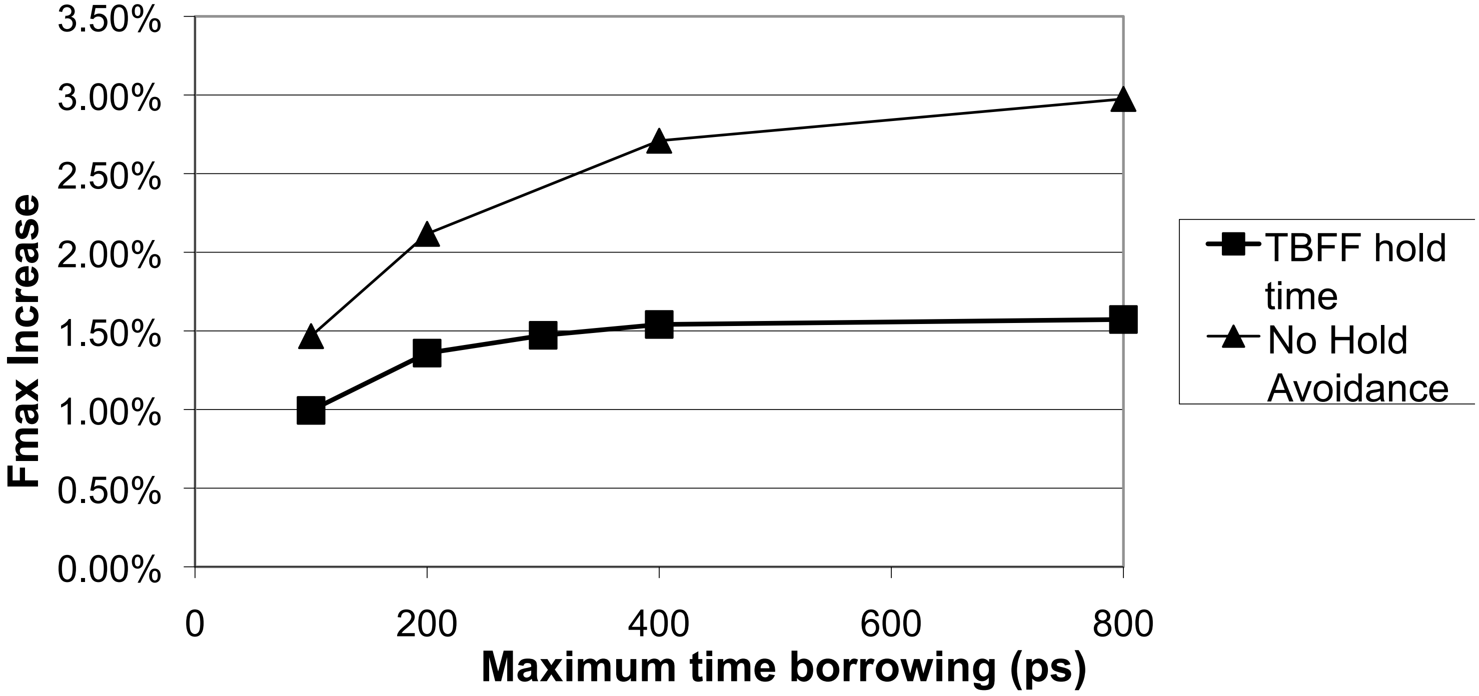
Time Borrowing Modeled as Programmable Clock Skew

- Pulse latch can be modeled as a programmable clock skew
- Notionally can select any clock skew with infinite precision



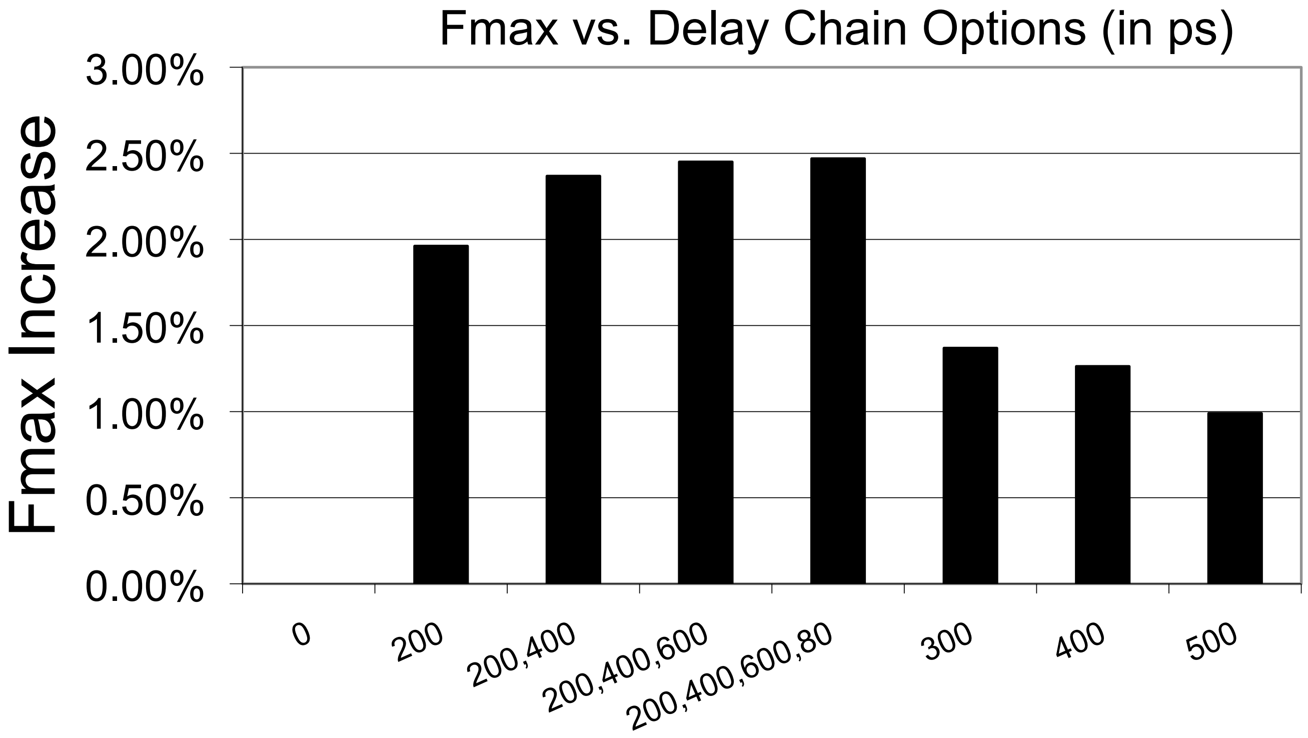
Measured Performance Benefit

- First experiment assumes one tap per 100ps and measures performance vs. number of taps
- Hold time requirements eat into gain



Discrete Delay Taps

- Evaluate a number of discrete combinations of delay tap settings
- Longer taps less gain due to hold time


Fmax vs. Delay Chain Options (in ps)

TBFF vs. Clock Skewing Edge-Triggered FF

+ Intrinsic speed and lower area of pulse latch

+ TBFF allows effective arbitrary clock skewing vs. fixed steps; possible to share same pulse for two different notional skews

+ TBFF absorbs jitter and uncertainty

+ Local variation in TBFF pulse generator reduces size of borrowing window and increases hold time but doesn't add to critical path; variability in clock skewing adds uncertainty to total critical path across sequential paths

Memory Architecture

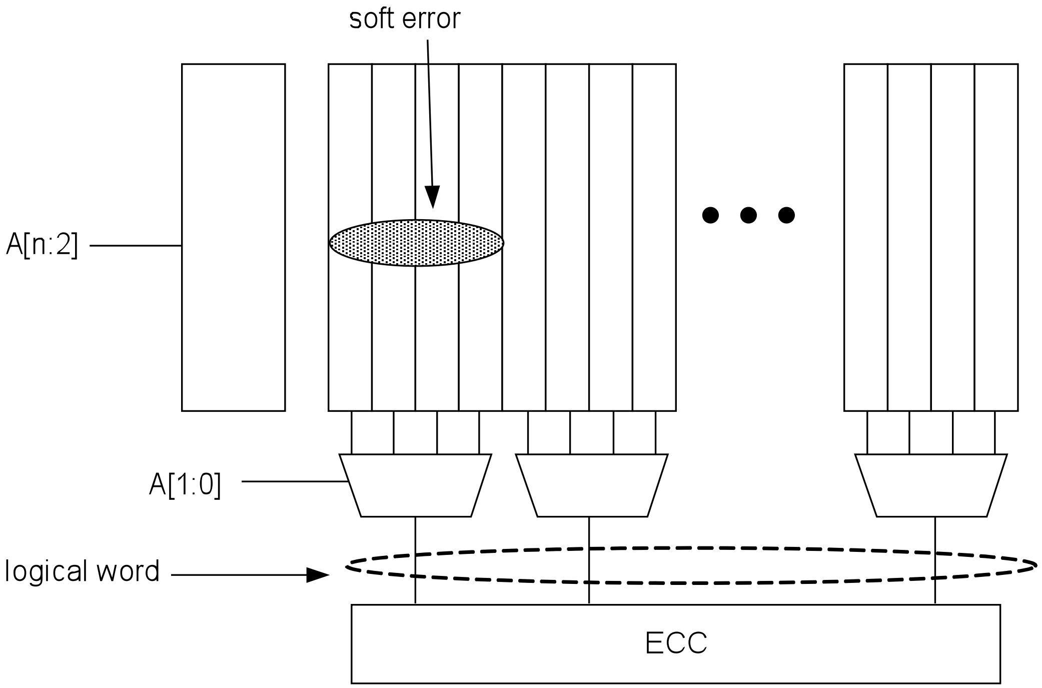
- Increasing amount of embedded memory in FPGAs
- Stratix V contains up to 50Mb of memory
- Soft errors in memory now significant contributor to overall soft error rate for the part
- Shrinking memory cell size means that significant fraction of the soft errors strike multiple bits
- S V includes hard error correcting logic in each embedded memory block with capability to correct all errors of up to 8 bits and detect errors of up to 12 bits

Stratix V ECC Methods

- Previous memory blocks used 36b words
- SECDED ECC on 32b data demands at least 39b
  - This is a rather weird number
- Increasing word size to 40 gives room for more powerful ECC that can correct 2 adjacent bits or detect 3 adjacent bits
- Column muxing spreads out bits in logical word so that adjacent logical bits are 4 bits apart in the array

Stratix V ECC

- 40b word with 4 way column muxing
- ECC corrects 2b logically adjacent errors and detects 3b logically adjacent errors
- Array is immune from errors up to 8b and detects up to 12b → essentially 0 soft error rate

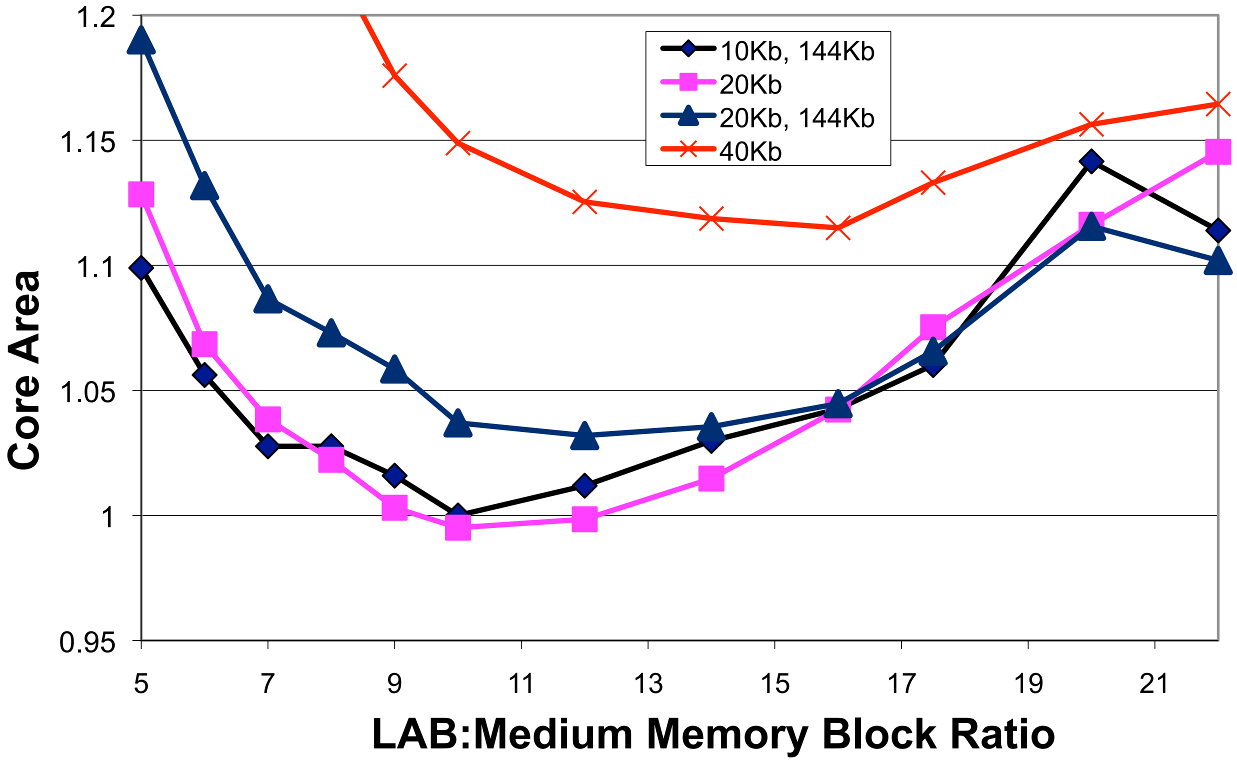


Memory Size

- Increasing demand for memory also drives increase towards larger block size
- Relative area efficiency advantage of large block (144Kb) diminishing compared to medium (10,20,40Kb)
- Compare Stratix IV 10Kb + 144Kb to 20Kb, 20Kb + 144kb, and 40Kb blocks
- Vary memory to LAB ratio
- Fit all designs in smallest core that can hold that design with that memory:LAB ratio
- Measure core area for a set of designs

Memory Block Size Experiment

- Single 20Kb block achieves area parity with existing architecture and gives less design effort, more regular placement of embedded blocks

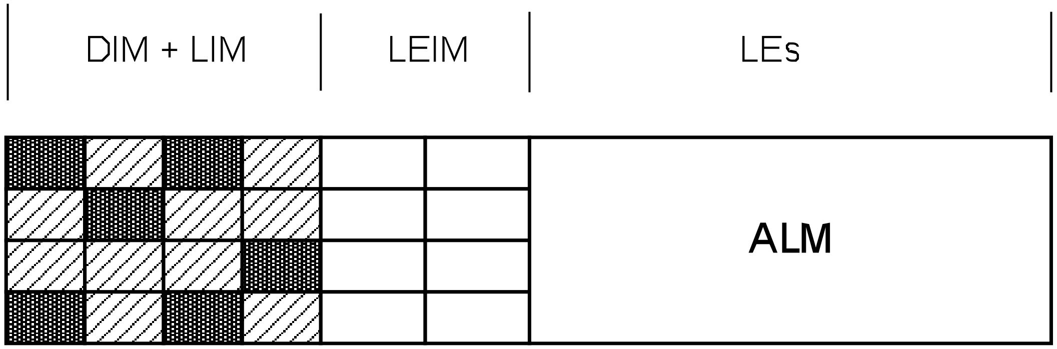


LUTRAM Improvements

- LUTRAM introduced in Stratix III with minimal cost implications by minimizing hardware to support it
- Added wiring from ALM inputs to write address decoder and write pulse generator
- Led to understanding of performance limiting factors and where to add hardware
- Introduction of four FFs per ALM also enables registering both read and write data
- Introduced hardened write address register in Stratix V
- Write address, write data, and read data now can all be in same LAB as the RAM

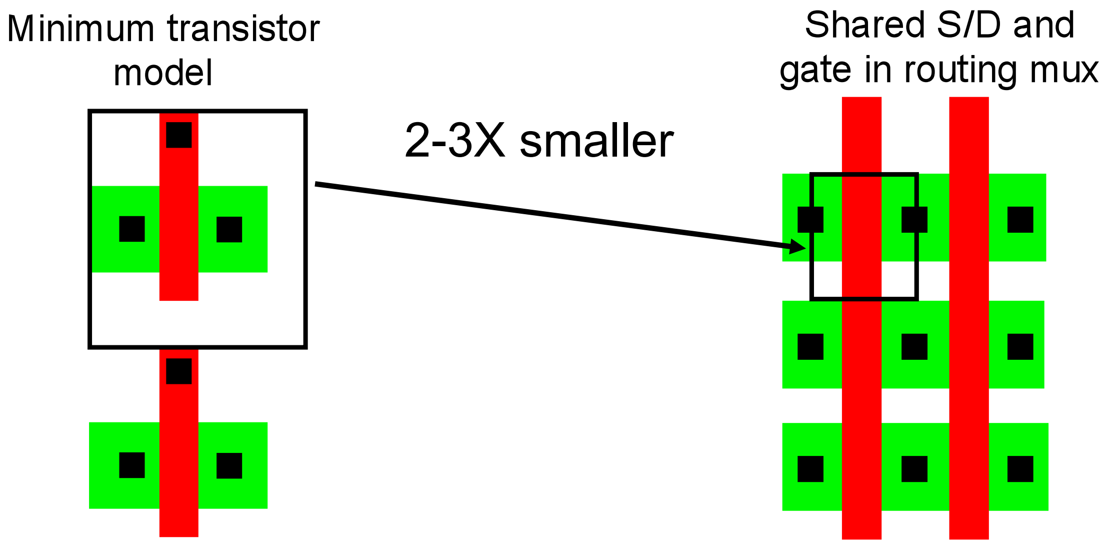
Routing Enhancements

- Modular layout leads to high efficiency for routing muxes
- Complementary aspect to efficiency is constraints on numerology
- Minor changes in routability accommodated by adjusting wire topology and metallization



Routing Optimization

- Routing is about 50% of LAB area (15% of die area)

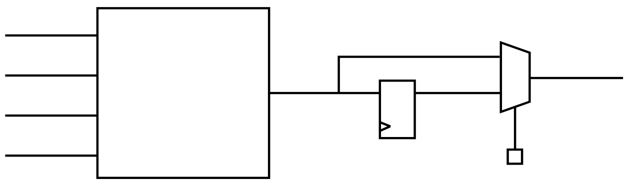


Minimum transistor model

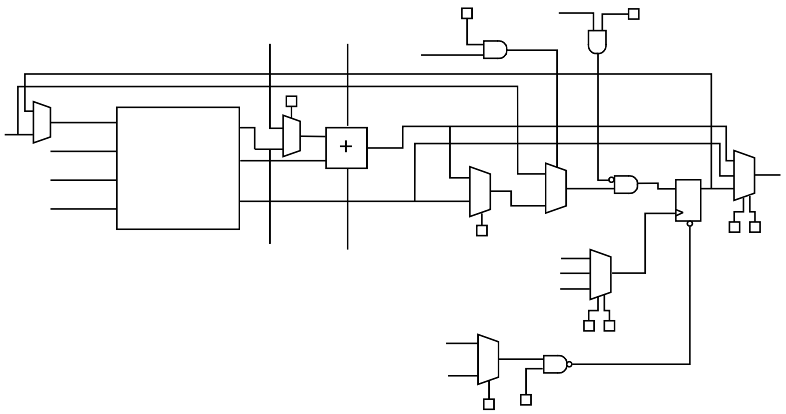
2-3X smaller

Shared S/D and gate in routing mux

Academic LE



Simplified Stratix LE



Routing Enhancements

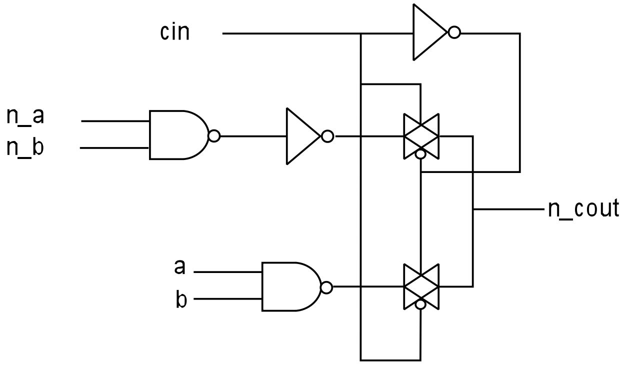
- Change H wire lengths to mix of H3 and H6 for ~1% improved performance and routability
- Take advantage of increased number of metal thicknesses available
  - 50% of V4 wires are on mid-thickness metal → faster
  - 80% of critical routes on V wires get the fast wires
  - ~8% fmax increase compared to all thin metal
- Long wires increased to H24 and V16 to be closer to optimal long distance routing delay
- Fast stitching to keep long distance routes fast

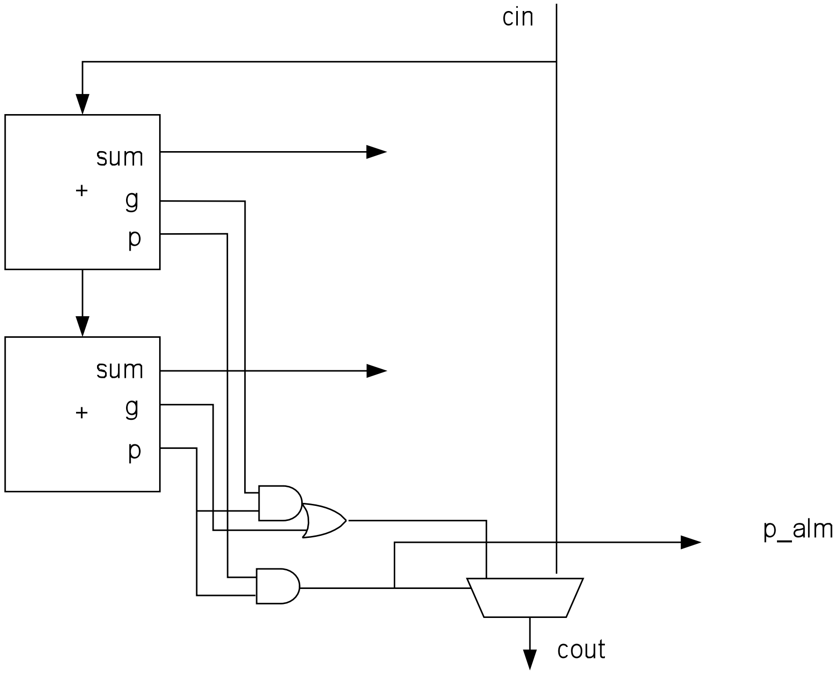
Adder Enhancements

- Single bit ripple carry for many generations
- Stratix included carry select across groups of 5 bits
- Ultimately judged not worth area and removed in Stratix II
- Ripple carry until Stratix IV

Two Bit Skip

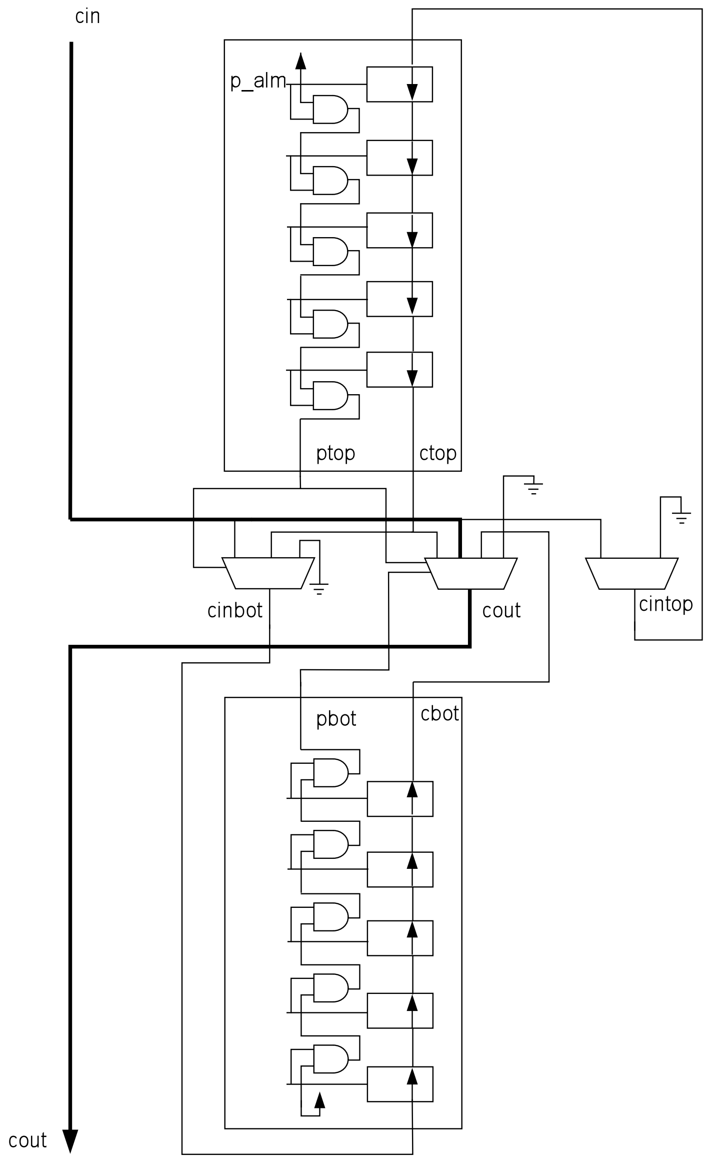
- Stratix IV introduced two-bit skip
- Less area than single bit skip because only skip mux transistors need to be large
- About twice as fast per bit





Stratix V Two Level Skip

- Two levels of carry skip in Stratix V
- Two-bit skip across ALM
- 20-bit skip across entire LAB
- Needs only one AND gate per ALM + skip logic in control signal region
- Long carry chains ~5X faster per bit



Summary

- Introduction of 4 FF per ALM with shared signals to keep low cost, and time borrowing to improve performance
- Unified memory architecture with single memory type and improvements to LUTRAM
- More use of heterogeneity in routing line length and metallization
- Two level carry skip with a single mux delay per 20 bits