SRT Division Architectures and Implementations

David L. Harris
*Stuart F. Oberman**
Mark A. Horowitz

Computer Systems Laboratory

Stanford University

* now with Advanced Micro Devices

OVERVIEW

- Problem Description
- SRT Division Architectures
- Domino Circuits
- Simulation Results
- Conclusions

PROBLEM DESCRIPTION

- Floating point unit performance very important
  - Scientific applications
  - 3D geometry calculations
  - Performance benchmarks
- Division is third most frequent FP operation
- SRT division produces one quotient digit per step
- How fast can we compute each quotient bit?
  - Algorithms
  - Circuits

SIMPLE DIVISION ALGORITHM

- Consider a simple division algorithm as the baseline
  - Base 2 version of algorithm from grade school
  - D = divisor, P_j = partial remainder after step j, q_j = quotient bit j
  - (1) Compute quotient bit: if ($D <= P_j$) $q_j = 1$ else $q_j = 0$
  - (2) Subtract and shift: $P_{j+1} = (P_j - D \times q_j) \times 2$
  - (3) Repeat 1-3 until done
- This is slow for several reasons:
  - Comparison is slow
  - Each step requires carry-propagate subtraction on a wide word
  - Only one bit is produced per step

IMPROVING DIVISION: SRT ALGORITHM

- **S**(weeney) **R**(obinson) **T**(ocher) division is most common non-restoring division algorithm that improves this algorithm
- Guess at the quotient digit to avoid slow compare
  - Guess using most significant bits of divisor & partial remainder
  - Use redundant quotient digits to correct later if guess was wrong
  - For example, choose q from {-1, 0, 1} for radix-2 divider
- Use redundant partial remainder representation to avoid slow subtract
  - Keep partial remainder in carry-save form
  - Now subtraction is done with a 3:2 CSA rather than a wide CPA
- Maybe use higher radix to obtain more quotient bits per step
  - a radix $r = 2^b$ divider “retires” b bits per iteration
  - r=2 and r=4 are most practical unless prescaling is done
  - radix-4 uses quotient digits {-a, ..., 0, ..., a} with a = 2 or a = 3

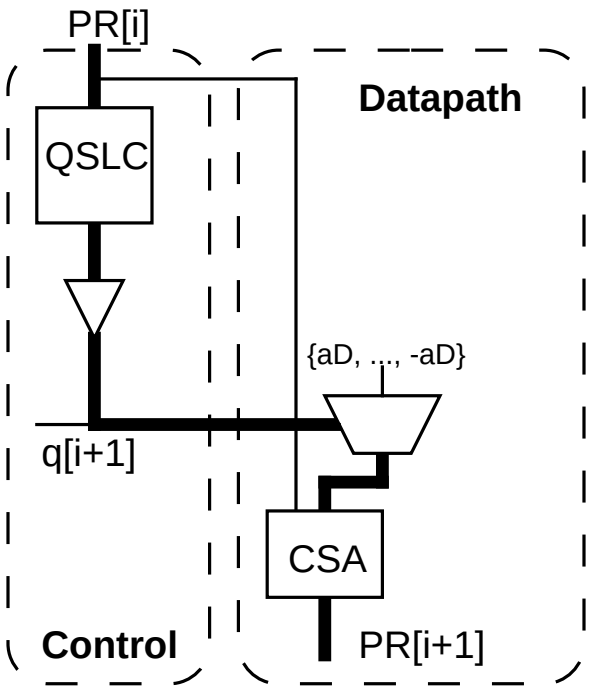
IMPROVING DIVISION: MORE TECHNIQUES

- Overlap s low radix stages to exploit parallelism
  - don't wait for one step to complete before beginning next
  - reduces delay per bit
  - increase in area & complexity

- Prescale the input operands
  - reduces the complexity of quotient selection, allowing r > 4
  - requires extra latency to perform the scaling

In this study, we focus on the architectural effects of different radices and overlapping schemes. We will also explore the benefits of circuit styles.

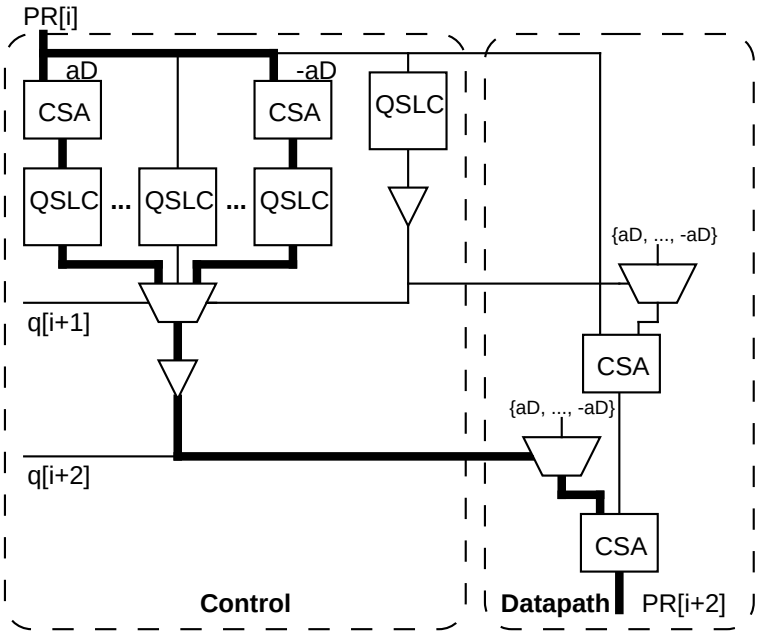
ARCHITECTURES



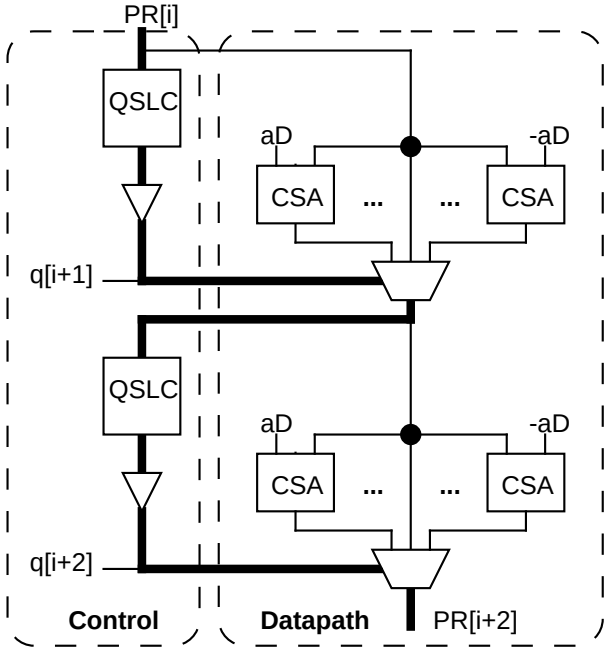
Non-overlapped divider

OVERLAPPED ARCHITECTURES

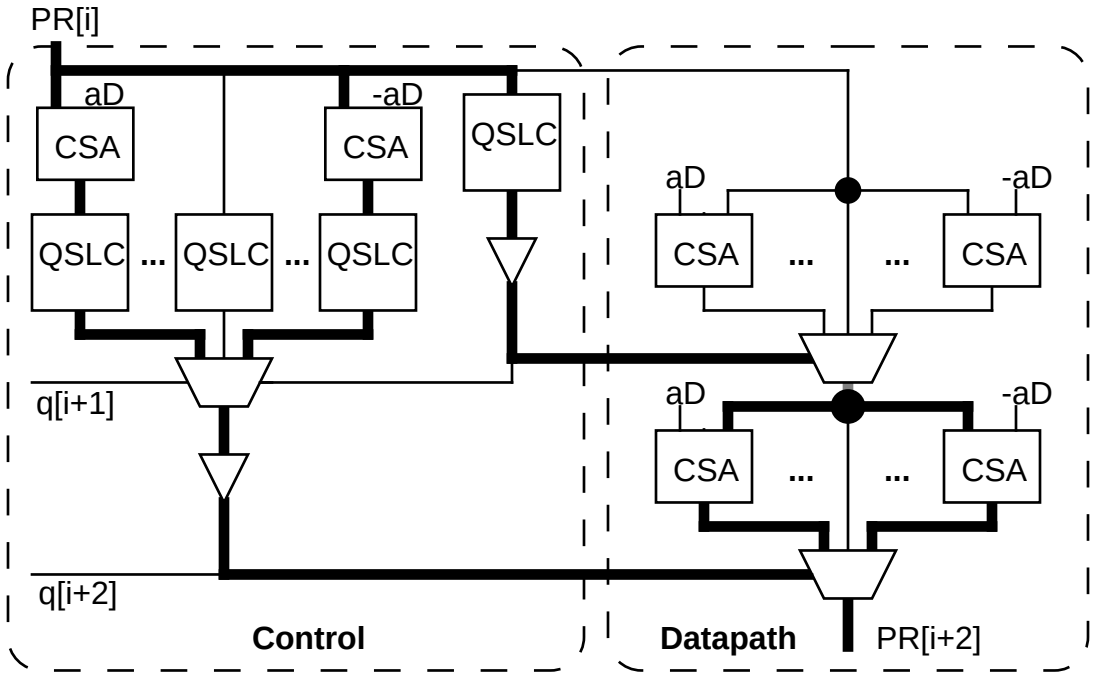
Overlapped quotient selection



Overlapped remainder formation

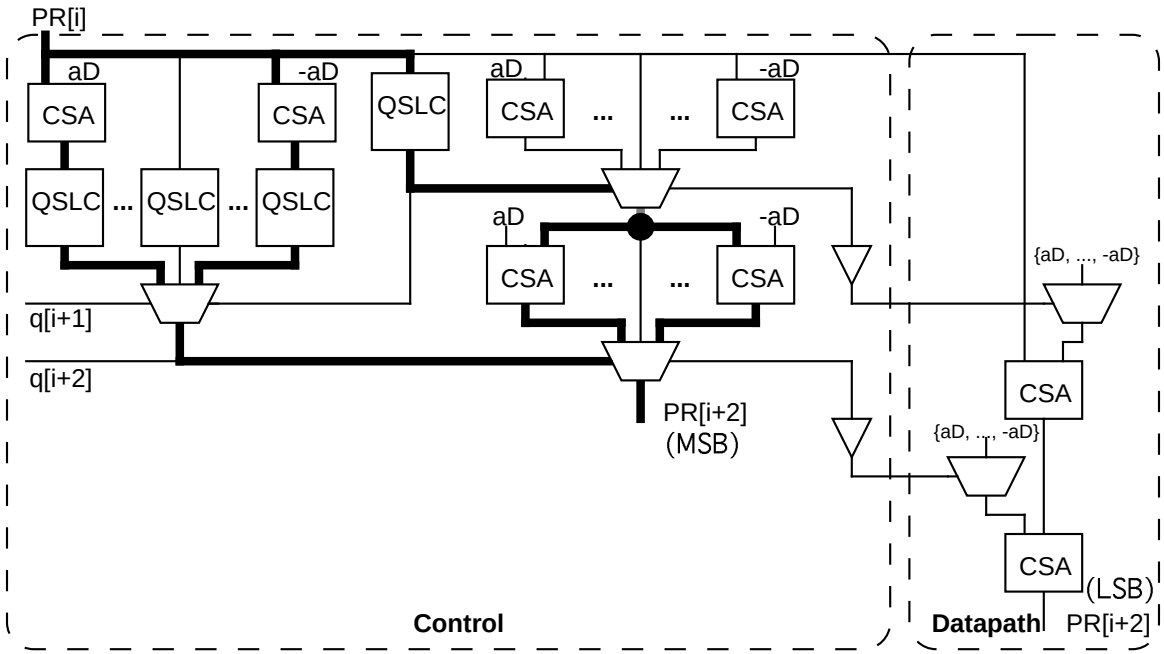


OVERLAPPED ARCHITECTURES



Overlapped remainder and quotient selection

OVERLAPPED ARCHITECTURES



Hybrid overlap

CIRCUITS

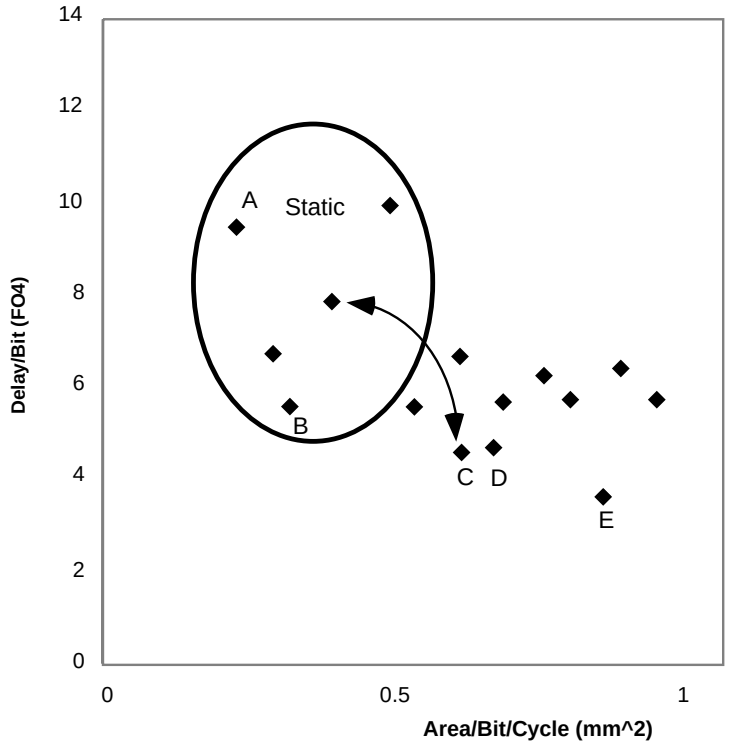
- Two fundamental circuit styles: static and dynamic
- Popular family of dynamic circuits is domino
  - Faster than static due to lower input cap & switching threshold
  - Requires monotonic inputs
  - Implementation of inverting gates requires dual-rail signals
- Textbook domino has overhead from latches, clock skew, & dead time
  - Skew-tolerant Domino uses multiple overlapping clock phases to eliminate the latches and overhead
  - roughly 25% faster than textbook domino in aggressive systems

SIMULATION RESULTS

- Assigned a divider design project in an advanced VLSI design class at Stanford University
- 12 teams explored a variety of radix-2 and radix-4 double precision SRT dividers
- Variety of skill levels and area / performance tradeoffs
- Circuits used skew-tolerant domino circuits

- Delay results are from HSPICE simulation
  - 1.0 μm HP-CMOS26B process design rules and models
  - Normalized to fanout-of-4 inverter delay in that process
- Area estimates from total transistors and device sizes
- Compare to data on static dividers from Ercegovac and Lang

RESULTS



Design	Arch	r	a	s	Delay /bit (FO4)	Area/ bit/ cycle
A	Non-over-lapped	4	2	1	9.5	0.23
B	Hybrid overlap	4	2	2	5.7	0.33
C	Hybrid overlap	2	1	2	4.6	0.63
D	Overlap QS	2	1	3	4.7	0.68
E	Hybrid overlap	4	3	2	3.7[a]	0.86

a. + initial latency computing D*3 divisor multiple

CONCLUSIONS

- Performance of domino circuits tightly clustered
  - many architectures yield comparably good performance
  - choice of radix-2 or radix-4 stages makes little difference
- Recommended architectures:
  - hybrid overlapped radix-2 (s=2)
  - quotient selection overlapped radix-2 (s=3)
  - hybrid overlapped maximally-redundant radix-4 (s=2)
- Delay / bit of recommended architectures is 4-5 FO4
- Circuit style has larger impact on performance than architecture
  - dual-rail domino offers 1.5 - 1.7x speedup over static
  - dual-rail domino designs typically larger than static designs and have twice as many wires