EE 382M
VLSI–II: Advanced Circuit Design
Superpipelined FPU Circuits

Bob Montoye, IBM
Ramyanshu Datta, TI
Outline

• Why superpipelining is important.
• Circuit enablers for superpipelining.
  – Domino superpipelining
  – OPL superpipelining
• Conclusions and challenges
Barriers

• What is the function of a latch?
  – Test/scan (required occasionally (MORE IN IBM))
  – Pipelining

• A barrier is a more generalized form of a latch that synchronizes data, but does not necessarily hold state.

• This talk will focus on using barriers in a more general way than fixed location latches.
Pipelining from the circuit designers viewpoint

- Unpipelined
  - Instruction in
  - Computation delay (d)
  - Instruction out
  - An instruction completes every d time units

- Pipelined
  - Instruction in
  - inst 1 inst 2 inst 3 inst 4 inst 5
  - Instruction out
  - An instruction completes every d/5 time units + latch overhead.

- Deeper pipeline
  - Instruction in
  - Instruction out
  - Latch overhead becomes an increasingly large percentage of area and delay.
Why Superpipelining

• Increase frequency faster than process scaling.
  – Each pipeline stage must do less work.
• But, quickly reach diminishing returns due to latch overhead.
• Solution: Get rid of the latches!
• This is a wave pipeline.
• Data flows in "waves" through the logic with no barriers to synchronize it.
• This is difficult since a fast moving wave can overtake a slow moving wave and wipe out a computation.
Why Superpipelining

- Purpose of the latches is to synchronize data.
- Many circuit structures can be used to do this.
- These structures are much "lighter" than latches since their only function is synchronization.
- Domino gate in precharge is the simplest one.
Domino Barriers

A domino gate functions as a natural barrier.

- When in precharge a domino gate cannot propagate data or change the state of following gates.
- Evaluation phases overlap to propagate data.
A Superpipeline with Domino Barriers
Domino SWAP test chip

- 24 x 24 multiplier without final add
  - Domino circuits with 7 clock phases
  - New operand added every 3 1/2 domino circuit delays (200ps)
  - 9 stages of domino - driver, AND, 6 x CSA
  - 40K devices
  - Cross section simulated at 5FO=4

- Power was limiting factor
  - Clock drivers were a majority of device area
Even Lighter Barriers: OPL

- Output Prediction Logic (OPL) is a very fast logic style developed at University of Washington by Sheng Sun, Carl Sechen and others.

- OPL circuits were initially developed to minimize latency, but their structure includes a natural barrier.
Why Static CMOS Is Slow

- All gates are inherently inverting
- On any circuit path, in the worst case:
  - Every output must fully transition from 1 to 0, or 0 to 1
  - If a path is not inverting, it is not critical.
Output Prediction Logic

- Goal: reduce the worst case
- Assume all outputs on a critical path will be 1
- You will be correct EXACTLY half the time
  - Every other gate on the path will not have to make transition
- Critical path delay will be reduced by at least 50%
Output Prediction Logic

• Problem:
  – 1 at every output (and therefore input) is not a stable state for an inverting gate
• Solution:
  – Disable each gate (1 at inputs and a 1 output is no longer a contradiction)
  – Disable each gate until its inputs are ready for evaluation
  – Predicted output value is therefore maintained
OPL Dynamic NOR3
OPL: Clocking

The University of Texas at Austin
EE 382M Class Notes
Foil # 15
Optimal OPL Clocking

- Consider a gate whose (controlling) input goes low: output should remain 1

a. Early Clock  
b. Optimal Clock  
c. Late Clock
OPL Pipeline

clk1

clk2

clk3

clk4

clk5

New data

eval
hold
pre
Summary

• Circuit technique: OPL
• Pipeline: matched fast stages, levelization
• Sizing: precharge/keeper
• Clocking:
  – 5 clock phases required, separation=25ps
  – ARL VCO can meet this requirement
  – This experiment used powerspice generated clocks
• 8 Ghz pipeline rate
Conclusions and challenges

• Systems are increasing pipeline rates faster than technology scaling
  – The cost of a pipeline barrier is becoming more important
  – We have explored two light barrier schemes, both of which have a rolling barrier (which travels along the pipeline)

• Scheme 1 involved footed domino as a way to hold state
  – Strength – traditional domino analysis will suffice
  – Weakness – requires dual rail logic and its power/area implications

• Scheme 2 involved OPL precharge as a rolling barrier
  – Strength – dual rail not required – more compact and lower power
  – Weakness – worse noise and timing margin
Limited Switching Dynamic Logic
Motivation

• Demand for performance is growing
  • Scientific computations, gaming
• Frequency gains due to scaling has hit a wall
  • Limitations due to power
  • Limitations due to variability
• Deeper pipelines yield lower gain
  • Latch overhead
  • Large number of cycles
• Fast circuit families like domino
  • High switching power
  • Difficulty in implementing inverting logic
**LSDL**

- **Limited Switching Dynamic Logic**
  - Merged logic-latch approach
  - Dynamic front end for speed
  - Latch at dynamic node to reduce switching

---

EE 382M Class Notes

Foil # 22

The University of Texas at Austin
**LSDL**

- Two levels of amplification to maintain smaller device sizes in the computational tree
  - Lower area
  - Lower supply voltage
  - Higher computational stacks

Get 1 good thing

• Merge 2 bad things
LSDL

- Computations can be carried out inside the latch
  - Lower latch overhead
- Static inverters can be inserted easily
  - No dual rail needed
Comparisons to LSDL

<table>
<thead>
<tr>
<th></th>
<th>Static</th>
<th>Domino</th>
<th>LSDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switching factor</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Invert</td>
<td>easy</td>
<td>dual rail</td>
<td>easy</td>
</tr>
<tr>
<td>Computation P-Fets</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Output</td>
<td>stacks</td>
<td>inverter</td>
<td>inverter</td>
</tr>
<tr>
<td>Clocks</td>
<td>none</td>
<td>standard</td>
<td>pulse</td>
</tr>
</tbody>
</table>

EE 382M Class Notes

The University of Texas at Austin
LSDL 4-to-2 Carry Save Adder (CSA)

Datta et. al. ISCAS 2004
LSDL 4-to-2 Carry Save Adder (CSA)

- Circuit Equations
  - Sum4 = a xor b xor c xor d
  - co= (a xor b xor c).d
  - Fast_Carry=a_p(b_p+c_p) + b_p+c_p
  - Sum=(Sum4) xor (fast_carry)
  - Car=co + (sum4 . fast_carry)
Comparison with Static CMOS

- **Cycle time = 60%**
- **Switching power = 31.5%**
- **Leakage = 37.7%**
- **Area = 65.5%**
LSDL Pipeline

- LSDL latches connect in split latch configuration to form a pipeline.
  - Most computation done in the latches.
- Static logic can be inserted between LSDL latches
  - Allows for inversions, which eliminates need for dual rail signaling
LSDL Clocking

Clkg

Delay1

Buffer

C1

Delay2

Delay1

Delay1

Delay2
LSDL Benefits

• Smaller design:
  • Almost all computation in n-fet trees.
  • 2 stages of gain allows computation devices to be small.
  • N-fet area twice p-fet area in multiplier unit.

• Low power design:
  • Smaller computation devices => shorter wires => smaller drivers.
  • Dynamic circuit power consumption controlled by latching outputs.
LSDL Design Constraints

- Most efficient computation done in an LSDL latch
  - Suited to deep pipelines / high frequency

- LSDL is edge triggered:
  - No transparency => no time borrowing
  - Clock skew and jitter must be tightly controlled
A Double Precision Floating Point Multiply

Robert Montoye, Wendy Belluomini, Hung Ngo, Chandler McDowell, Jun Sawada, Tuyet Nguyen
IBM Austin Research Laboratory

Brian Veraa, James Wagoner, Mike Lee
IBM Enterprise Server Group
Multiply/Add Fused Overview

The University of Texas at Austin

EE 382M Class Notes
Multiplier and MAF

• A multiplier contains:
  • A Booth decoder
  • A Booth mux
  • Several levels of compressors (3:2 & 4:2)

• A MAF unit
  • Performs operation \((A \times B) + C\) using a single instruction
  • Requires a pre-normalization (shifting) step for aligning \(C\) with the product of \((A \times B)\)
  • Addition followed by post-normalization and rounding

• This paper is only about the multiplier
Chip Overview

• Goal: Reduce cost of high-frequency chips
  – Cost is driven by power and area

• Initial target unit: 53x54 multiply
  – Largest component of FP dataflow
  – Modified Wallace tree

• Circuit family: Limited Switch Dynamic Logic
  – Merged Logic/Latch approach

• Custom design and placement, wired with autorouter.

• Circuit family and physical organization critical to results.
53x54 Multiplier Architecture
Multiplier Floorplanning

• Three drivers of area reduction:

• Layout of Booth Mux: 3:2 multiplier stage:
  – This is the part of the pipeline with the most terms
  – 60% of total area.

• Rectangular floorplan:
  – Fits better in overall chip floorplan

• Use of autorouter to allow freedom of cell placement.
Booth Mux/3:2 Layout

- 12 Booth muxes
- 4 3:2’s
- Data bits: M1
- Select bits: M3
- Local interconnect and some M2 to make connections
- 96 data inputs compress to 8 outputs.
Rectangular Floorplan

27 Booth Encoded Terms  →  Wallace Tree Reduction

EE 382M Class Notes  Foil # 40  The University of Texas at Austin
Die Photo

Booth Mux/3:2

Booth Encode

Booth Mux/3:2

3:2

4:2

3:2

4:2

Data drivers

3:2

4:2

3:2

EE 382M Class Notes

Foil # 41

The University of Texas at Austin
Autorouting

- Cells were placed based on the custom floorplan.

- Autorouting was used to wire all point to point nets.
  - Clock nets were hand placed.
  - Data and booth select signals were hand routed.
  - Critical nets within Manhattan distance of cells.

- Tested multiple placements quickly.
  - Over 100 possible placements were tried.

- Removed rigid constrains on bit positions needed.
Test Method

Legend:
- LSDL Circuits
- Static Synthesized Circuits

Input FIFO
- LFSR
- Flip-flops
- Advance State

Multiplier
- Counters
- Advance State
- To LCBs, counters, etc

Output FIFO
- MISR
- Flip-flops

Clock control
Test Board
Measured Results

• Chip fabricated in IBM CMOS9S (130nm SOI)
  – 8 metal layers.
  – M1-M4 used for multiplier wiring, M1-M3 completely used, M4 30% used

• The multiplier is 495\(\mu\)m x 315\(\mu\)m, 0.15mm\(^2\)

• Scaled for technology, this is a 50% reduction in area over the best previously published implementation [Itoh, JSSCC 2001]
  – The LSDL multiplier is pipelined while the one in [Itoh] is not.

• Frequencies up to 2.2GHz at 1.2V and 25\(^\circ\)C

• Power: 522mW at 80% SF and 261mW at 0% SF
  – Between 119pJ and 237pJ per multiply
Light Emissions

80% SF

0% SF
Conclusions

• Dynamic design can result in lower area and power.
  • Moves most computation into small n-fet trees.
  • Inverters always used as drivers.
  • Smaller devices => shorter wires => smaller drivers.
  • Dynamic power is controlled by latching outputs.

• Creative floorplanning and careful layout have a dramatic impact on design size and quality.
References

• **OPL**
  


• **MAF**
  
References

- **LSDL**
  - Datta, Ramyanshu; Montoye, Robert; Nowka, Kevin; Sawada, Jun; Abraham, Jacob A.; *Design of Shifting and Permutation Units using LSDL Circuit Family*, Asilomar Conference on Signals, Systems and Computers, pp. 1692-1696, 2006