Advanced Circuit Techniques for High-Performance Microprocessor and Low-Power DSPs

Sanu Mathew, Ram Krishnamurthy,
Mark Anders, Steven Hsu and Shekhar Borkar

Circuit Research, Intel Labs
Intel Corporation, Hillsboro, OR

Intel Labs

Outline

- Challenges & Circuit Solutions:
  - High-performance power-efficient execution core
    - 6.5GHz single-rail domino 32-bit Han-Carlson ALU
    - 4GHz semi-dynamic 32-bit sparse-tree AGU
  - Leakage-tolerant register files
    - Conditional/burn-in keeper
    - Pseudo-static bitlines
  - Low-power datapaths for DSP applications
    - 1GHz 16-bit static multiplier

2

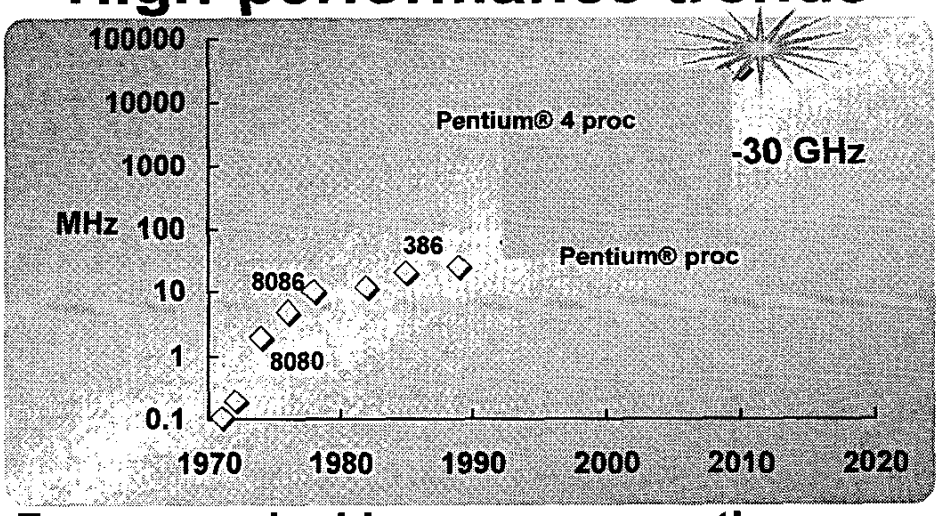
0-7803-8713-9/04/$20.00 ©2004 IEEE

Outline

- Challenges & Circuit Solutions:
  - High-performance power-efficient execution core
    - 6.5GHz single-rail domino 32-bit Han-Carlson ALU
    - 4GHz semi-dynamic 32-bit sparse-tree AGU
  - Leakage-tolerant register files
    - Conditional/burn-in keeper
    - Pseudo-static bitlines
  - Low-power datapaths for DSP applications
    - 1GHz 16-bit static multiplier

3

High-performance trends

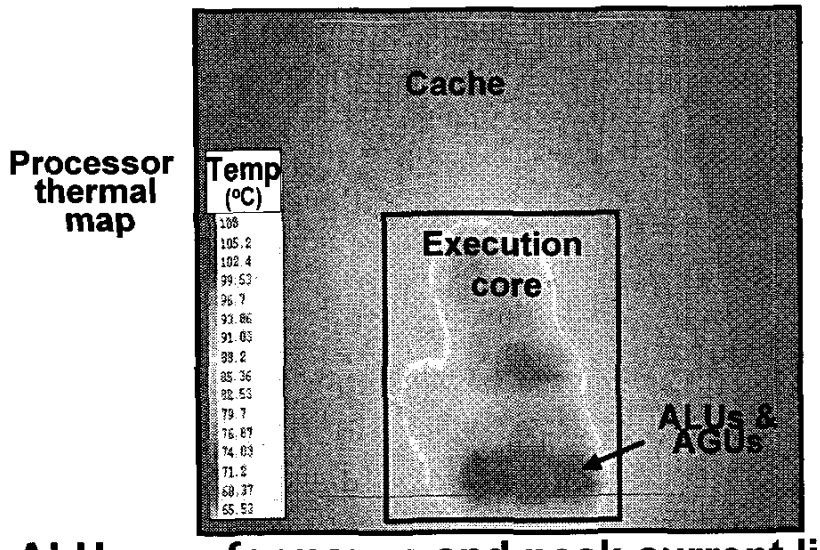


- Frequency doubles every generation
- Performance-critical units
  - ALUs & AGUs
  - Register files, L0 Caches

} Single-cycle throughput & latency

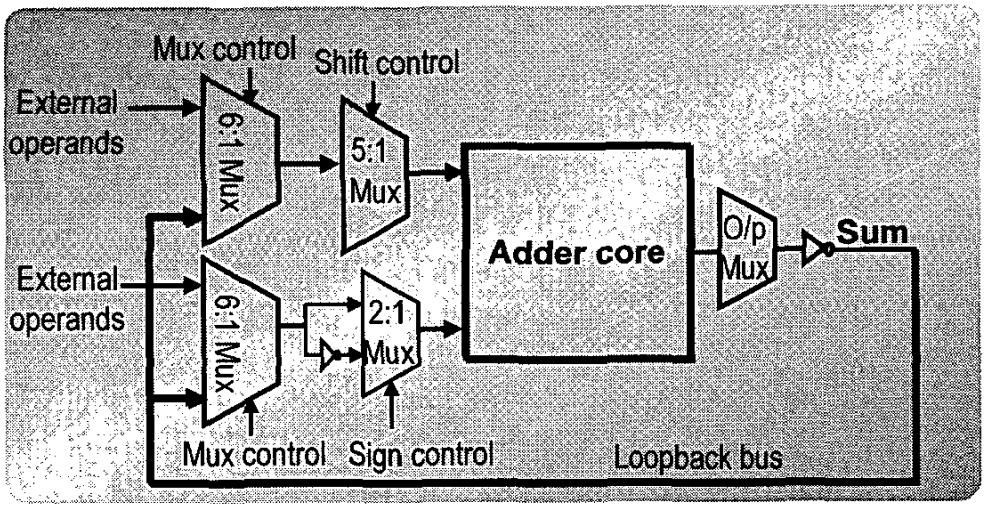
4

Motivation



- ALUs: performance and peak-current limiters
- High activity ⇒ thermal hotspot
- Goal: high-performance energy-efficient design

32-bit ALU architecture



Multiple ALUs clustered together in the execution core ⇨High power density

A 6.5GHz, 130nm Single-ended Dynamic ALU

[M. Anders et al, ISSCC 2002]

Intel Labs

$$Sum_i = A_i \oplus B_i \oplus Carry_{i-1}$$

$$Carry_i = A_i \cdot B_i + (A_i + B_i) Carry_{i-1}$$

Intel Labs

Partial Sum

$$Sum_i = \overbrace{A_i \oplus B_i}^{\text{Partial Sum}} \oplus Carry_{i-1}$$

$$Carry_i = A_i \cdot B_i + (A_i + B_i) Carry_{i-1}$$

Intel Labs

Partial Sum

$$Sum_i = \overbrace{A_i \oplus B_i}^{\text{Partial Sum}} \oplus Carry_{i-1}$$

$$Carry_i = \underbrace{A_i \cdot B_i}_{\text{Generate}} + \underbrace{(A_i + B_i)}_{\text{Propagate}} Carry_{i-1}$$

Intel Labs

Partial Sum

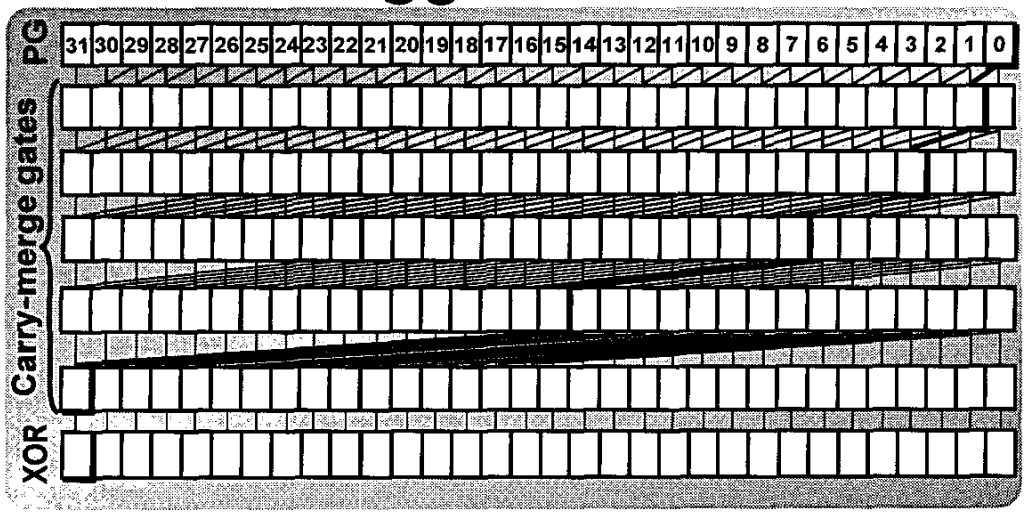
$$Sum_i = \overbrace{A_i \oplus B_i} \oplus Carry_{i-1}$$

$$Carry_i = \underbrace{A_i \cdot B_i}_{\text{Generate}} + \underbrace{(A_i + B_i)}_{\text{Propagate}} Carry_{i-1}$$

$$Carry_i = G_i + P_i \cdot Carry_{i-1}$$

Intel Labs

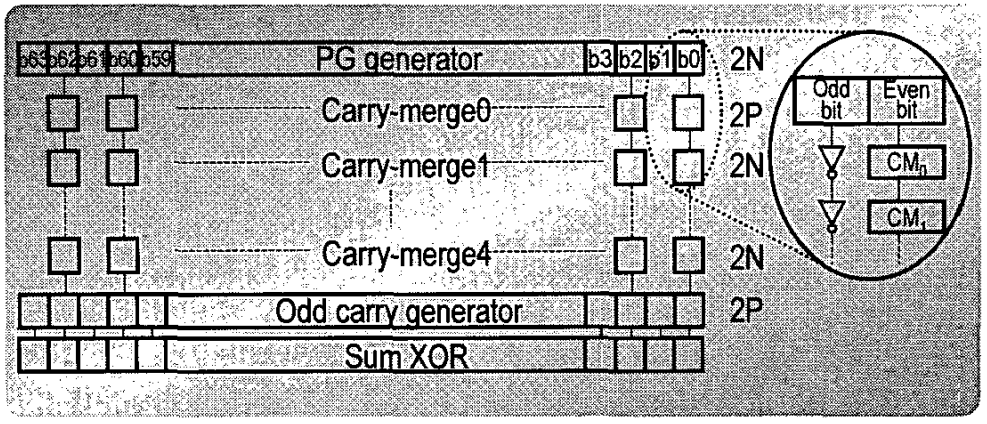
32-bit Kogge-Stone Adder



- Critical path = PG+5+XOR = 7 gate stages
- Generate,Propagate fanout of 2,3 } Energy inefficient
- Maximum interconnect spans 16b }

12

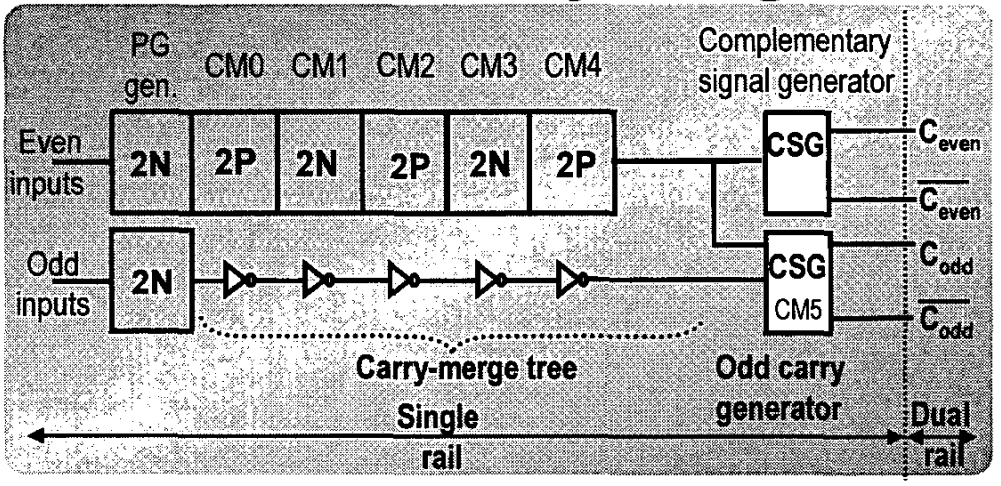
32-bit Han-Carlson adder core



- Carry-merge done on even bitslices
- 50% fewer carry-merge gates vs Kogge-Stone
- Extra logic stage generates odd carries

13

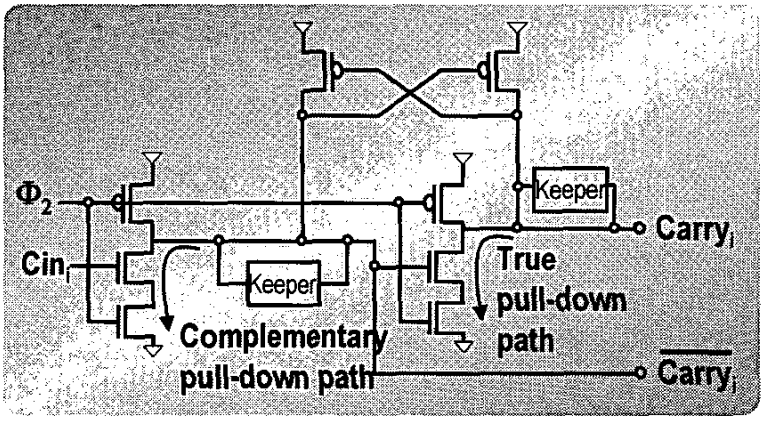
Han Carlson carry-merge tree



- Single rail adder core
- CSG circuit generates dual-rail carry

14

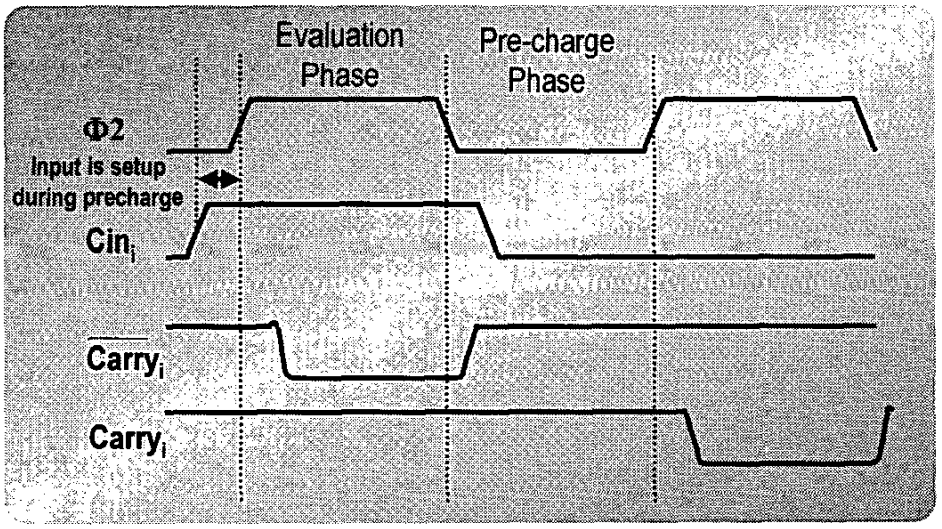
Complementary signal gen.



- Domino-compatible Carry/$\overline{\text{Carry}}$
- Permits a single-rail carry-merge tree design
- Not time-borrowable – Penalty absorbed by placing gate at Φ_2 boundary

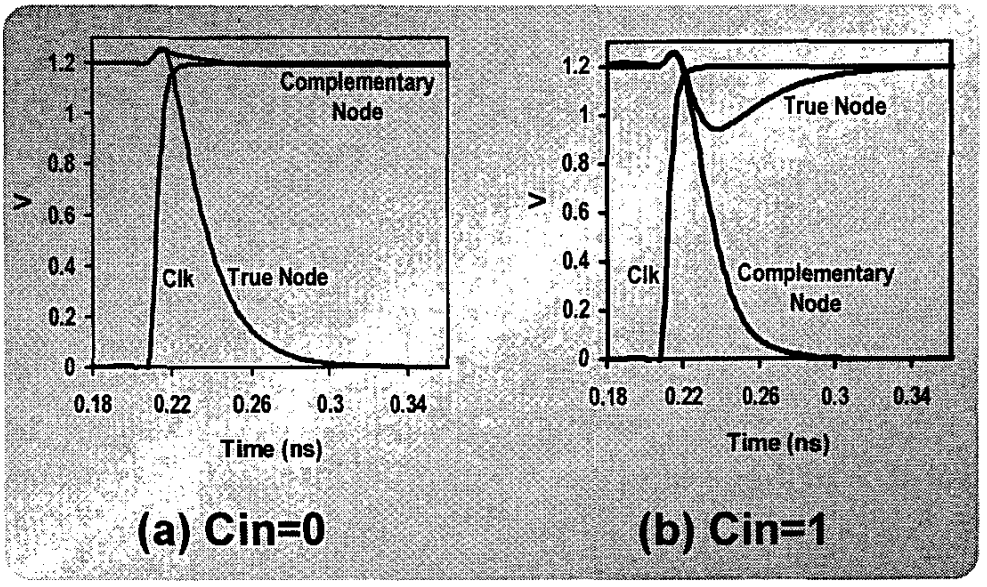
15

CSG: Timing Diagram



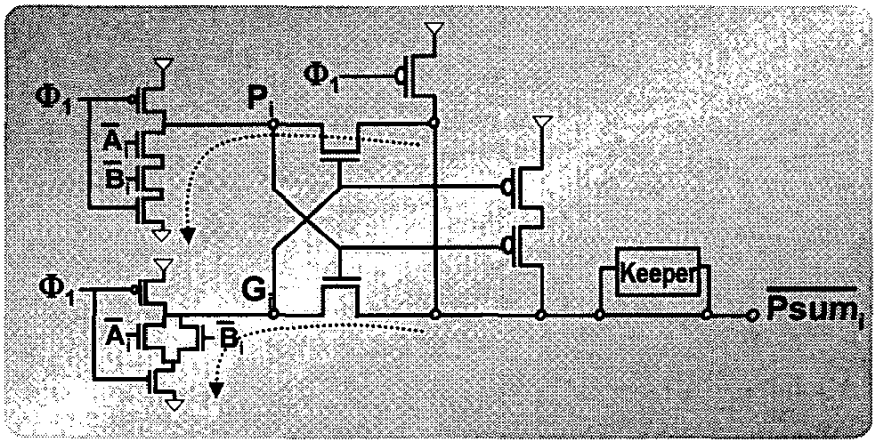
16

CSG: Simulation waveforms



17

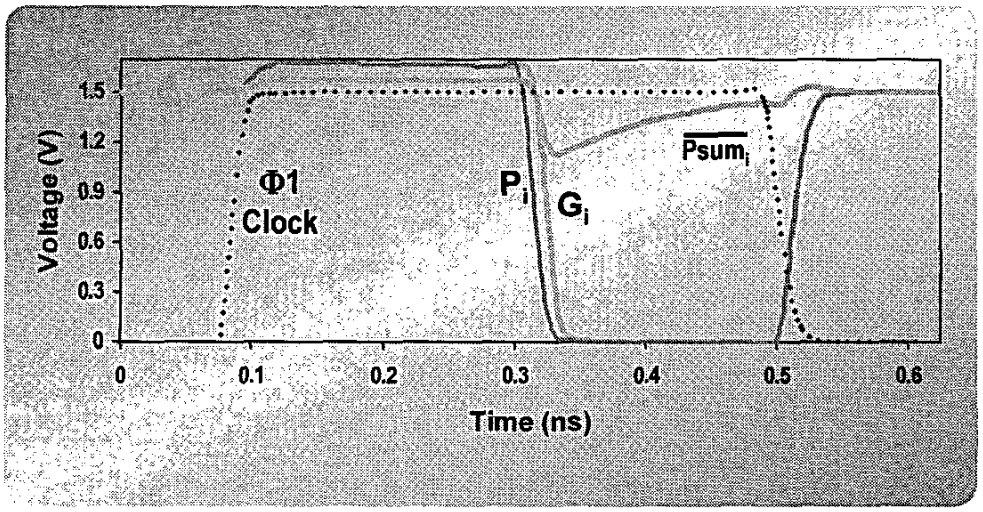
Partial sum generator



- Generates domino-compatible partial sum
- Placing the gate at Φ_1 boundary mitigates output noise-glitches

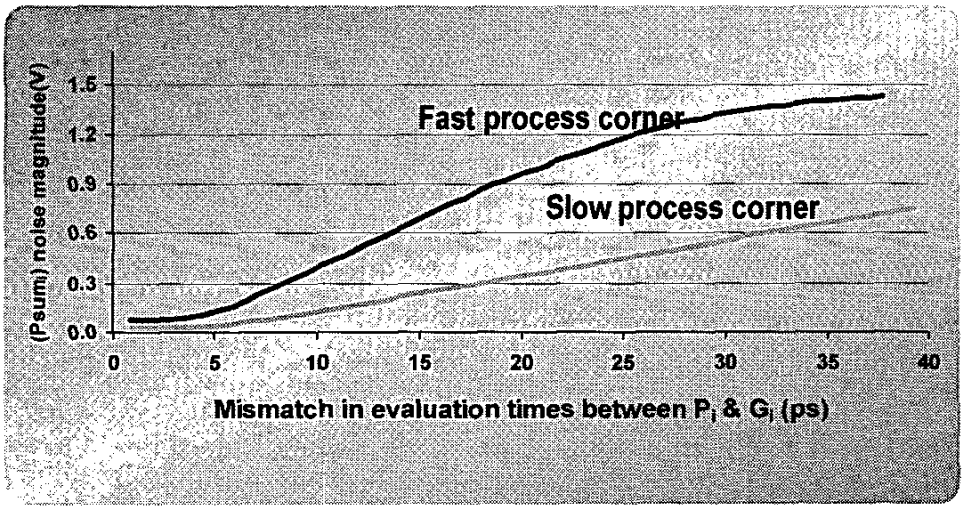
18

Dynamic XNOR: Simulation Waveforms



19

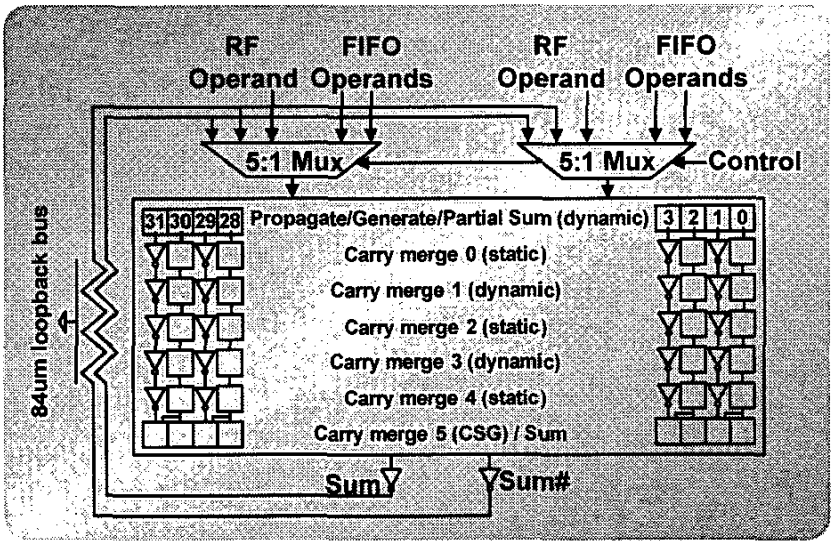
Dyn. XNOR: Noise Sensitivity



- Mismatch in input evaluation times can cause output noise

20

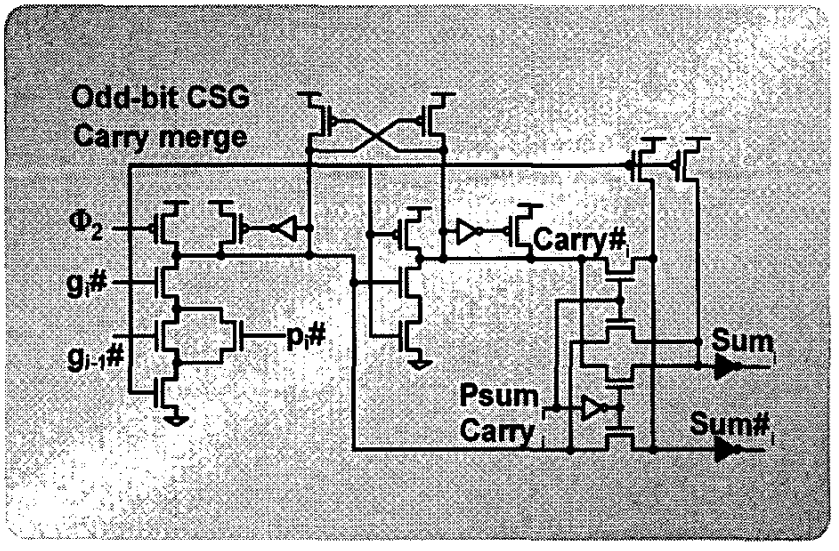
Han-Carlson ALU Organization



- Single-rail dynamic 9-stage low-V_t design

21

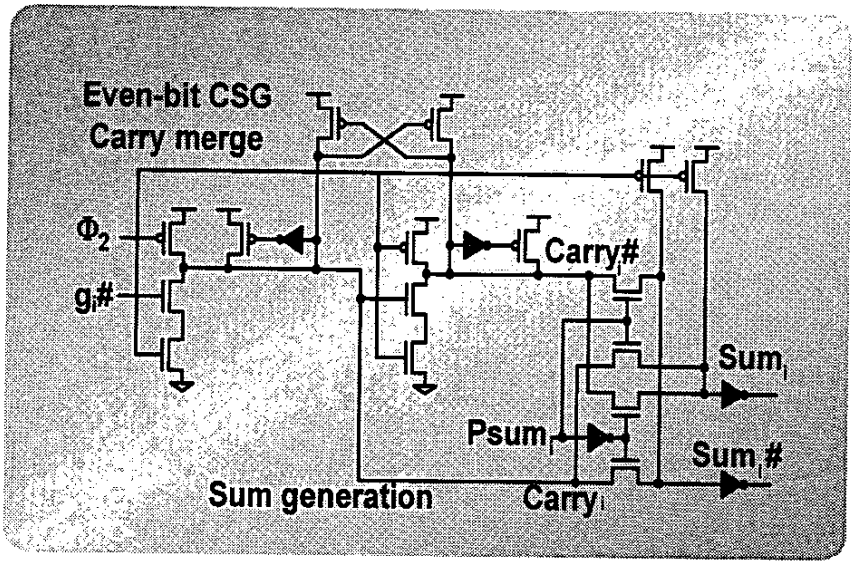
Odd-bits CSG Sum Generation



- Final carry-merge CSG(dual-rail carry output)
  → pass-transistor sum XOR

22

Even-bits CSG Sum Generation

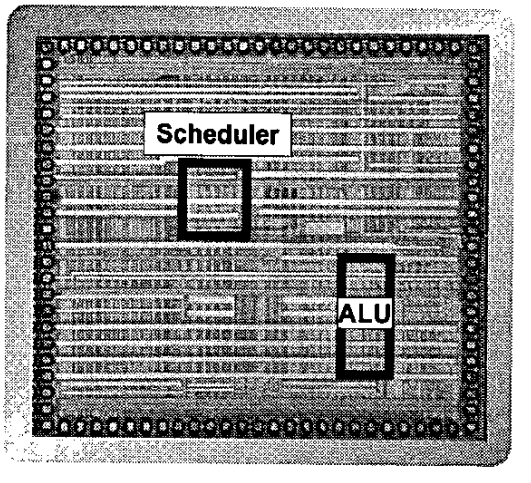


- Domino-compatible sum
- Dual-rail sum from single-ended g inputs

23

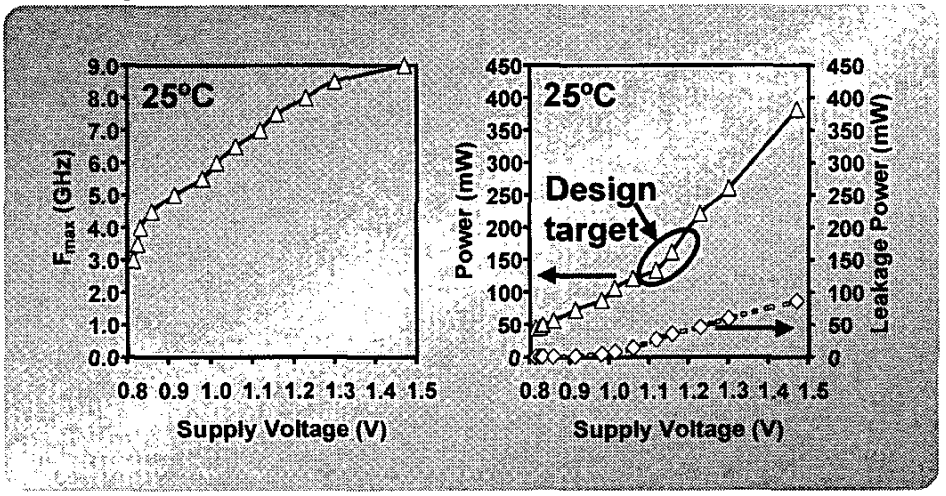
Die Micro-photograph

- 130nm 6-metal dual-Vt CMOS
- Scheduler:
  - 210µm x 210µm
- ALU:
  - 84µm x 336µm



24

Delay and Power Measurements



- 6.5GHz at 1.1V, 25°C
- Power: 120mW total, 15mW leakage
- Scalable to 10GHz at 1.7V, 25°C

25

Improvements Over Dual-rail Domino

Area	50%
Performance (Delay)	10%
Active Leakage	40%
Robustness	equal

- Leakage reduced by eliminating dual-rail logic
- Robustness not compromised
- CSG improves both area and performance

26

A 4GHz 130nm Address Generation Unit with 32-bit Sparse-tree Adder Core

[S. Mathew et al, VLSI Symp. 2002]

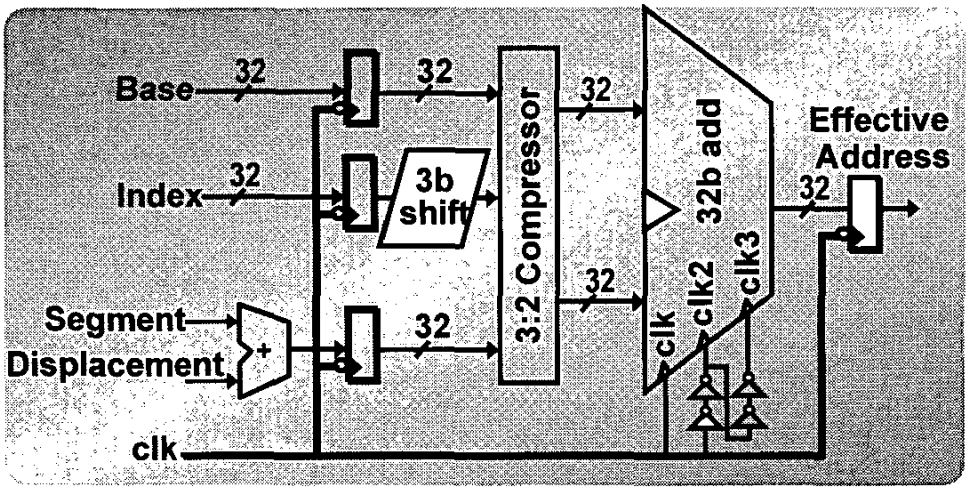
Intel Labs

Outline

- Address Generation Unit (AGU) organization
- Sparse-tree adder core
- Dual-V_t semi-dynamic design
- Sub-130nm scaling trends
- Summary

28

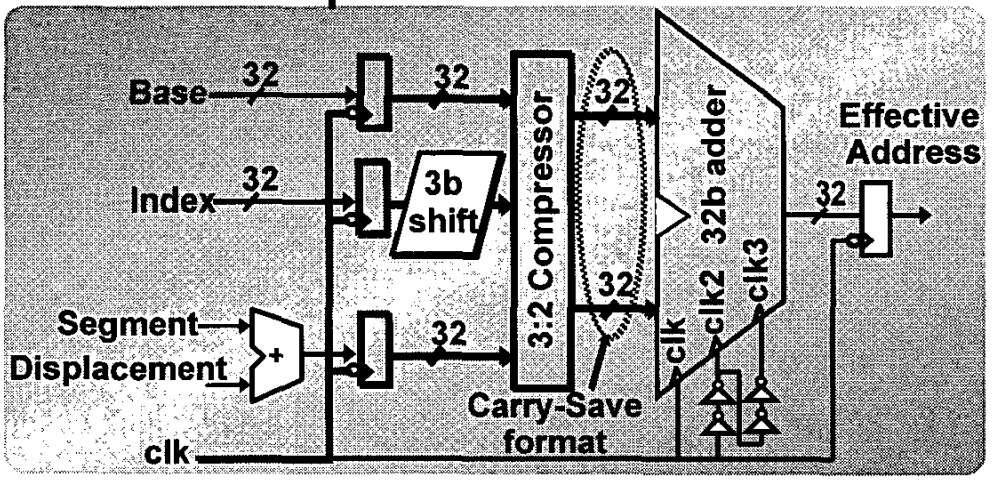
AGU Architecture



- Single-cycle latency and throughput
- Effective Address = Base + Index*Scale + (Segment +Displacement)
- 2-phase address computation

29

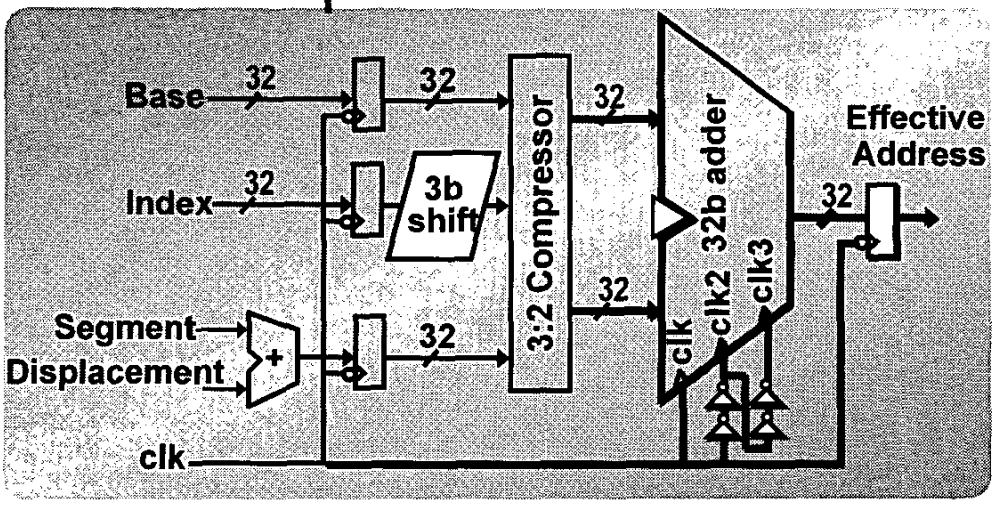
AGU Operation: Phase 1



- Index pre-scaled via 3-bit barrel shifter
- 3:2 compressor renders partial address:
  - Carry-save format
- Adder in pre-charge state

30

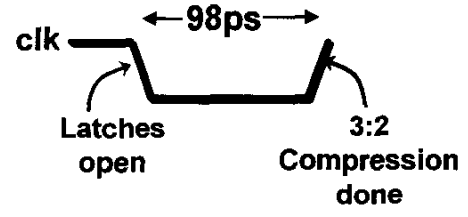
AGU Operation: Phase 2



- **Carry-save to binary format conversion:**
  - **2's complement parallel 32-bit adder**

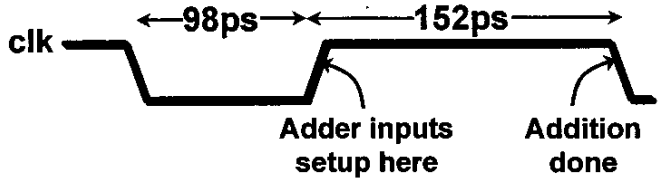
31

Timing Diagram



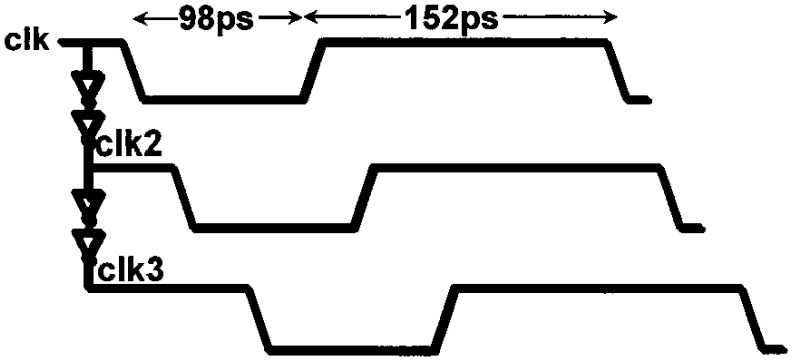
32

Timing Diagram



33

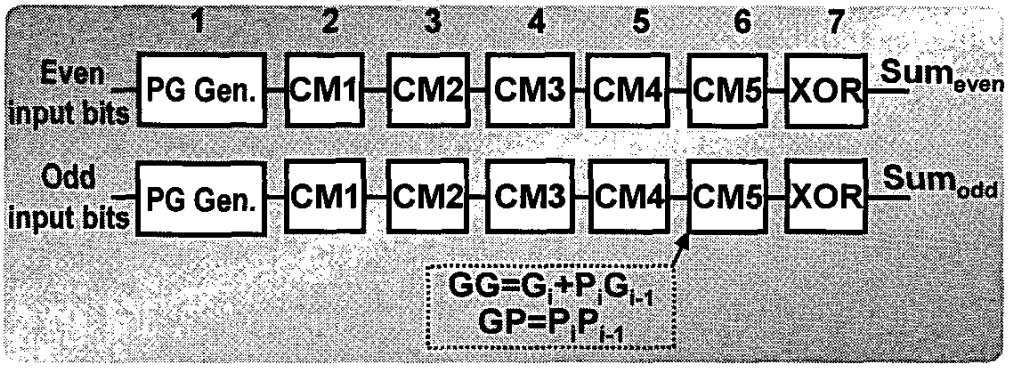
Timing Diagram



- **Seamless time-borrowable clock boundaries**
- **152ps (6.6GHz) 32-bit adder core required**

34

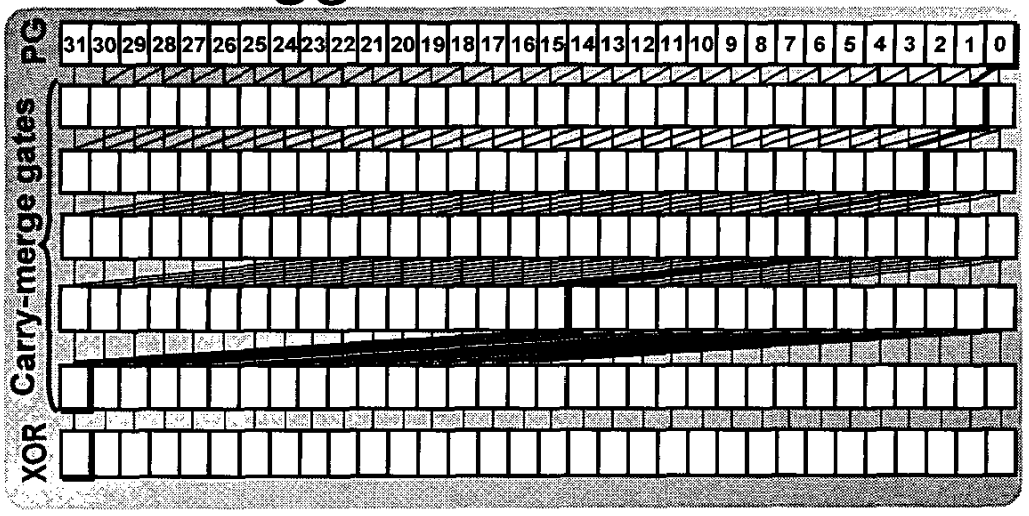
High-performance Adders: Kogge Stone



- Generate all 32 carries:
  - Full-blown binary tree ⇒ energy-inefficient
- # Carry-merge stages = $\log_2(32)$ ⇒ 5 stages

35

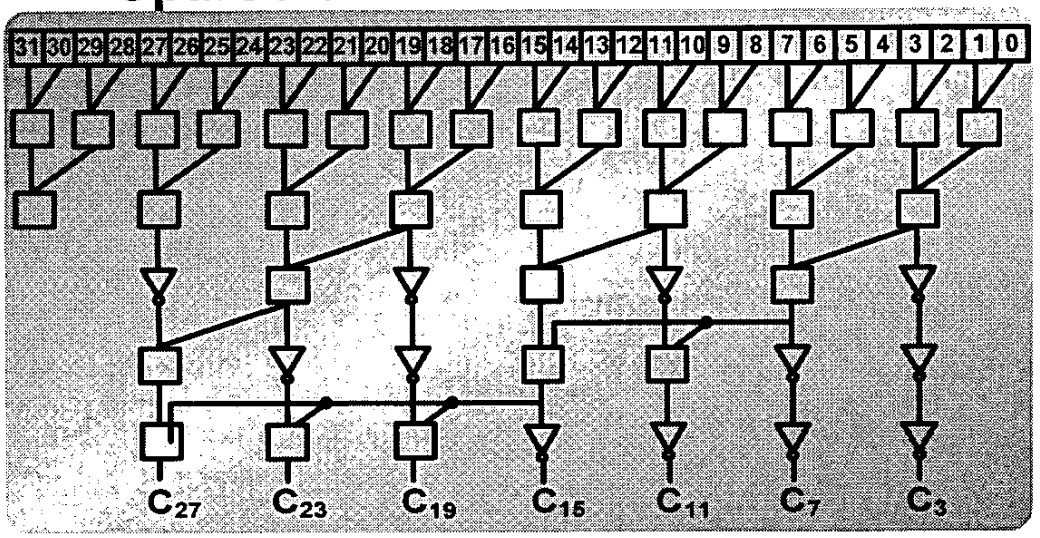
Kogge-Stone Adder



- Critical path = PG+5+XOR = 7 gate stages
- Generate,Propagate fanout of 2,3 } Energy inefficient
- Maximum interconnect spans 16b }

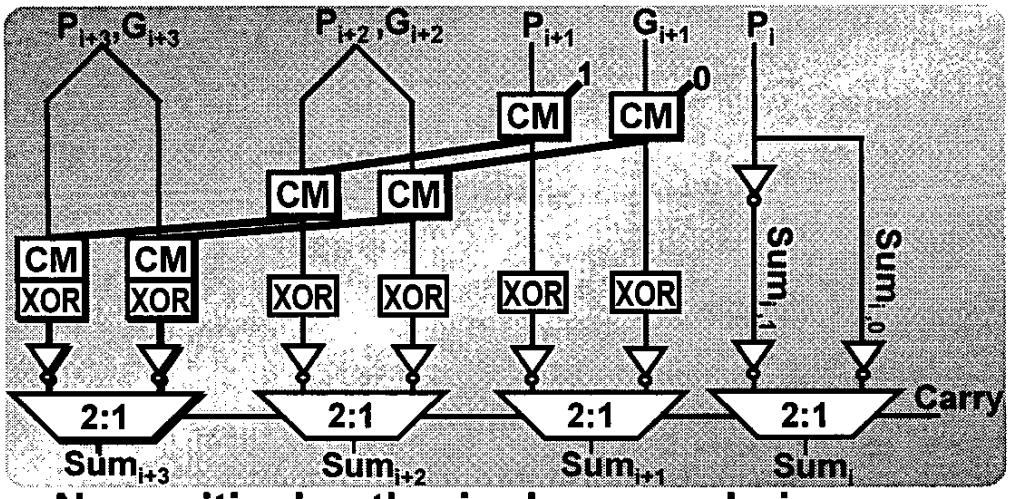
36

Sparse-tree Adder Architecture



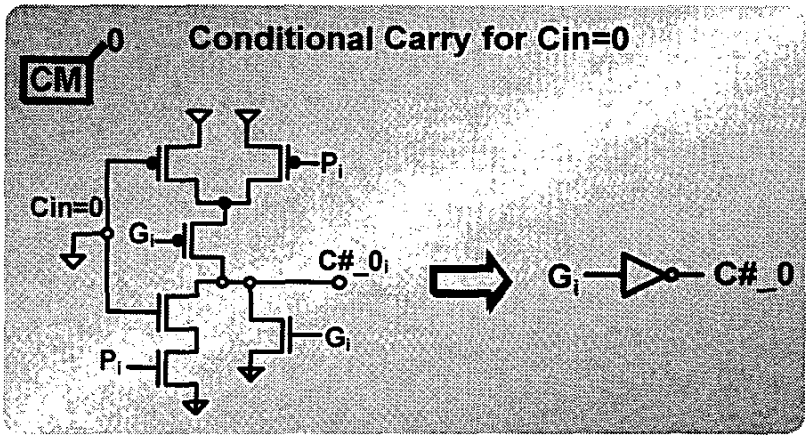
- Generate every 4th carry in parallel
- Side-path: 4-bit conditional sum generator
- 73% fewer carry-merge gates⇒energy-efficient

Non-critical Sum Generator



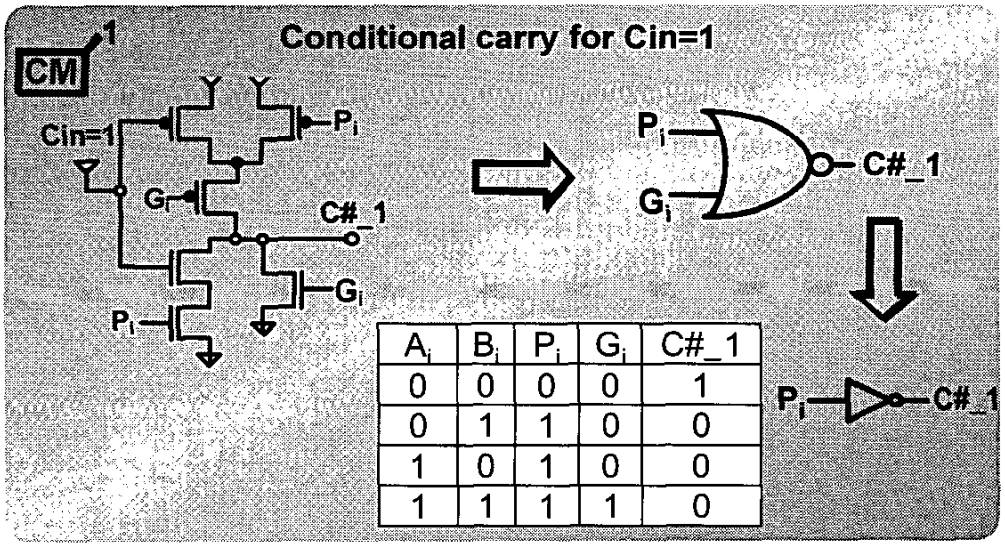
- Non-critical path: ripple carry chain
- Reduced area, energy consumption, leakage
- Generate conditional sums for each bit
- Sparse-tree carry selects appropriate sum

Optimized First-level Carry-merge



- Carry-merge stage reduces to inverter
- Conditional carry_0 = G_i#

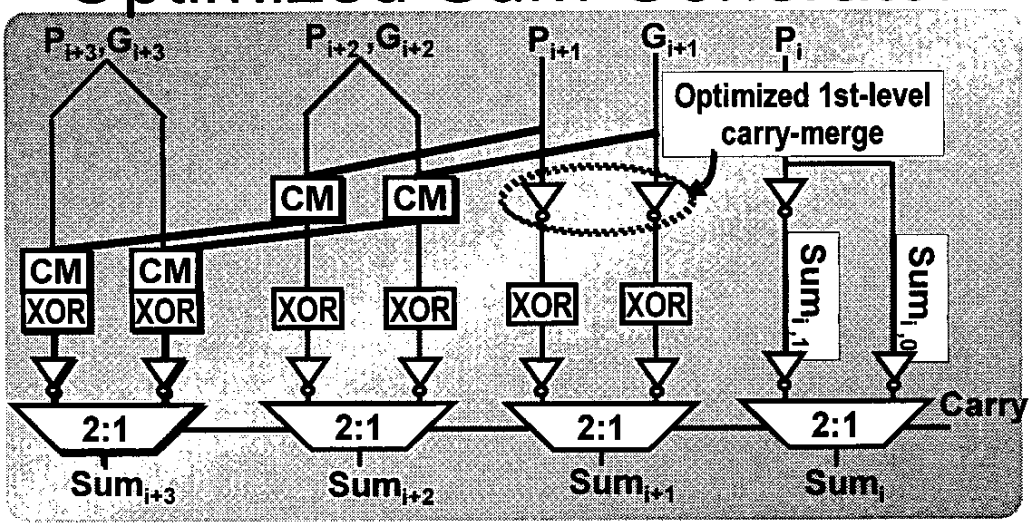
Optimized First-level Carry-merge



A_i	B_i	P_i	G_i	C#_1
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	0

- P_i & G_i correlated
- Conditional carry_1 = P_i#

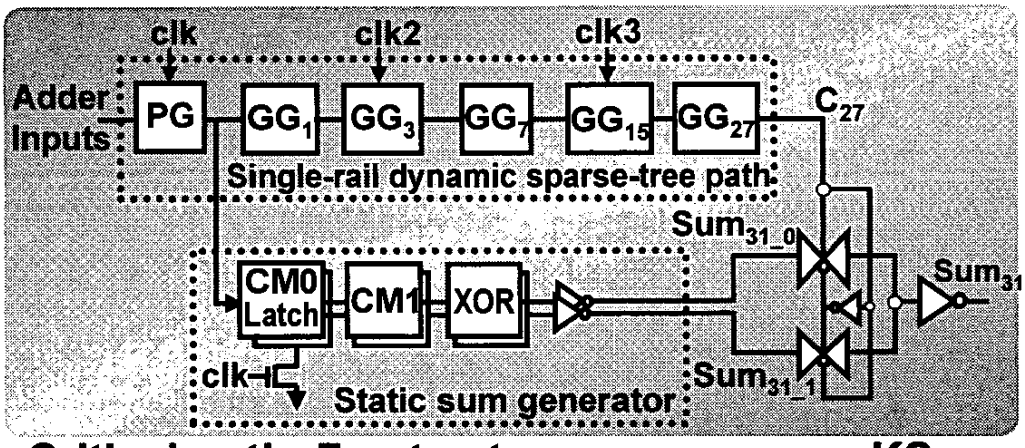
Optimized Sum Generator



- Optimized non-critical path: 4 stages

41

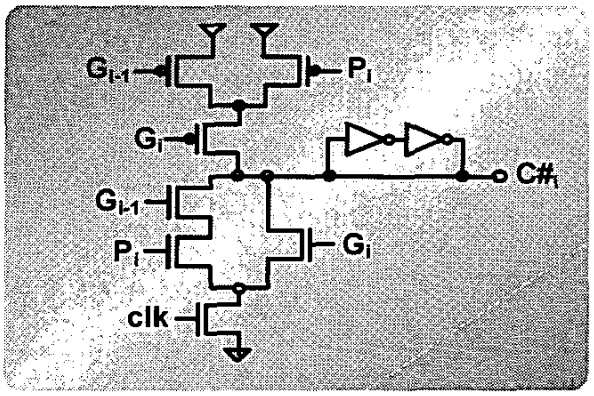
Adder Core Critical Path



- Critical path: 7 gate stages ⇒ same as KS
- Sparse-tree: single-rail dynamic
- Exploit non-criticality of sum generator
- Convert to static logic⇒Semi-dynamic design

42

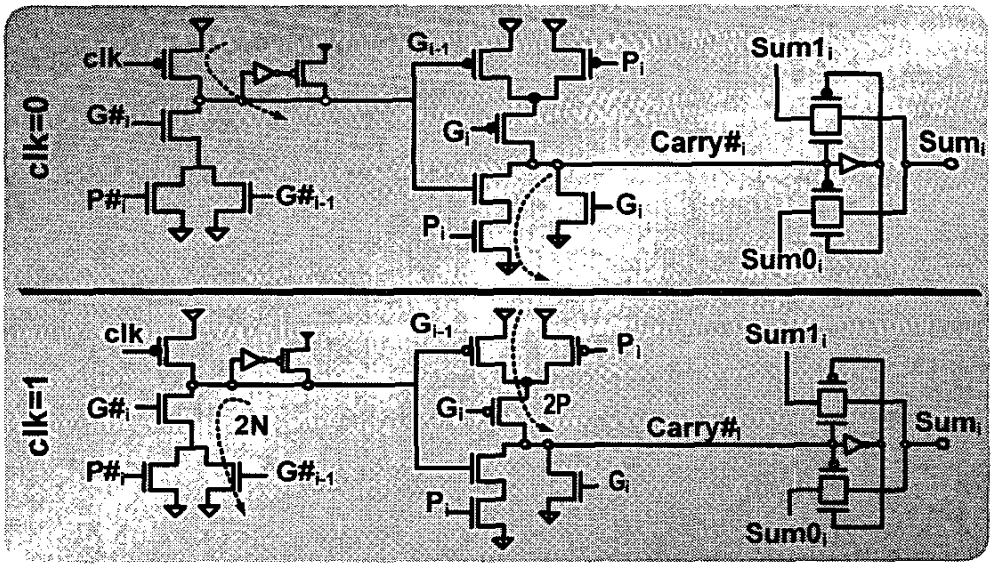
1st-level Carry-merge: Static Latch



- Holds state in pre-charge phase
- Prevents pre-charging of static stages

43

Domino-Static Interface



- Sum=Sum0 during pre-charge
- Mux output resolves during evaluation

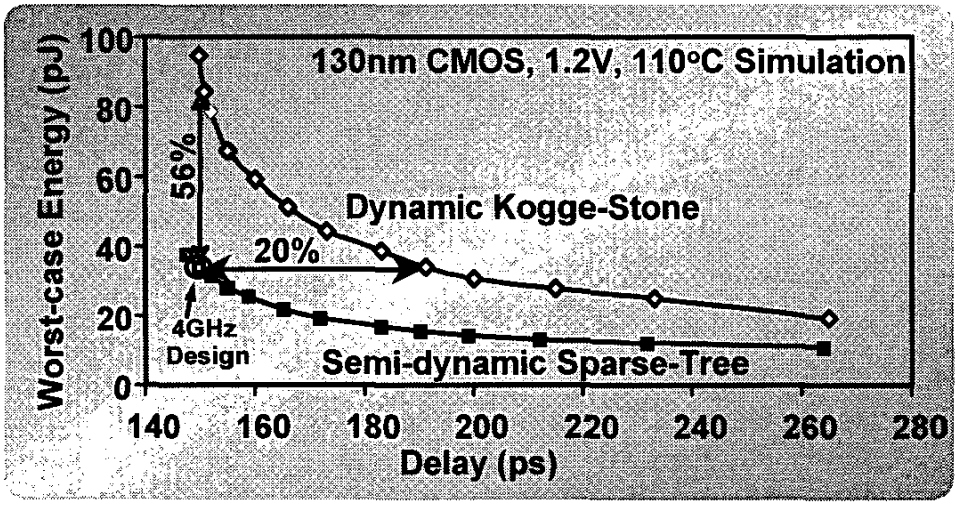
44

Sparse-tree Architecture

- Performance impact: (20% speedup)
  - 33-50% reduced G/P fanouts
  - 80% reduced wiring complexity
  - 30% reduction in maximum interconnect
- Power impact: (56% reduction)
  - 73% fewer carry-merge gates
  - 50% reduction in average transistor size

45

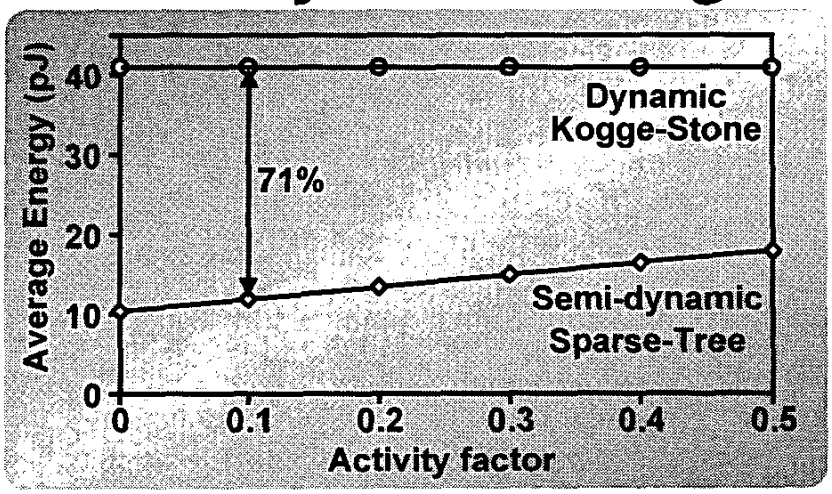
Energy-delay Space



- 20% speedup over Kogge-Stone
- 56% worst-case energy reduction
  - Scales with activity factor

46

Semi-dynamic Design



- Static sum generators : low switching activity
- 71% lower average energy at 10% activity

47

Dual-V_t Allocation

130nm CMOS, 1.2V, 110°C Simulation

	Low-V_t	Dual-V_t
Delay	152ps	152ps
Switching Energy	36pJ	34pJ (-6%)
Leakage Energy	0.9pJ	0.4pJ (-56%)

- Exploit non-criticality of sidepaths
  - Use high-V_t devices
- 0% performance penalty
- 56% reduction in active leakage energy

48

Scaling Performance

	130nm	100nm
Delay	152ps	102ps (-33%)
Switching Energy	36pJ	18pJ (-50%)
Leakage Energy	0.9pJ	0.7pJ (-23%)

- **Average transistor size = 3.5μm**
  - **Reduces impact of increasing leakage**
- **80% reduction in wiring complexity**
  - **Reduces impact of wire resistance**
- **33% delay scaling, 50% energy reduction**

49

Summary

- **4GHz AGU in 1.2V, 130nm technology**
- Sparse-tree adder architecture described
- 20% speedup and 56% energy reduction
- Semi-dynamic design:
  - **Energy scales with switching activity**
- Dual-V_t non-critical paths:
  - **Low active leakage energy**
- **6.5GHz ALU and scheduler at 1.1V, 25°C**
  - –Scalable to 10GHz at 1.7V, 25°C

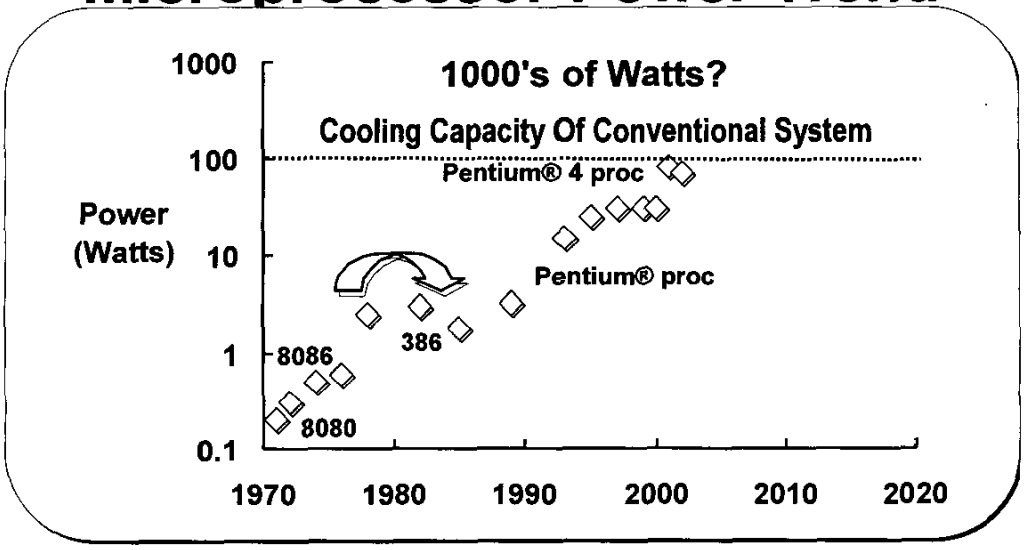
50

Outline

- Challenges & Circuit Solutions:
  - High-performance power-efficient execution core
    - 6.5GHz single-rail domino 32-bit Han-Carlson ALU
    - 4GHz semi-dynamic 32-bit sparse-tree AGU
  - Leakage-tolerant register files
    - Conditional/burn-in keeper
    - Pseudo-static bitlines
  - Low-power datapaths for DSP applications
    - 1GHz 16-bit static multiplier

51

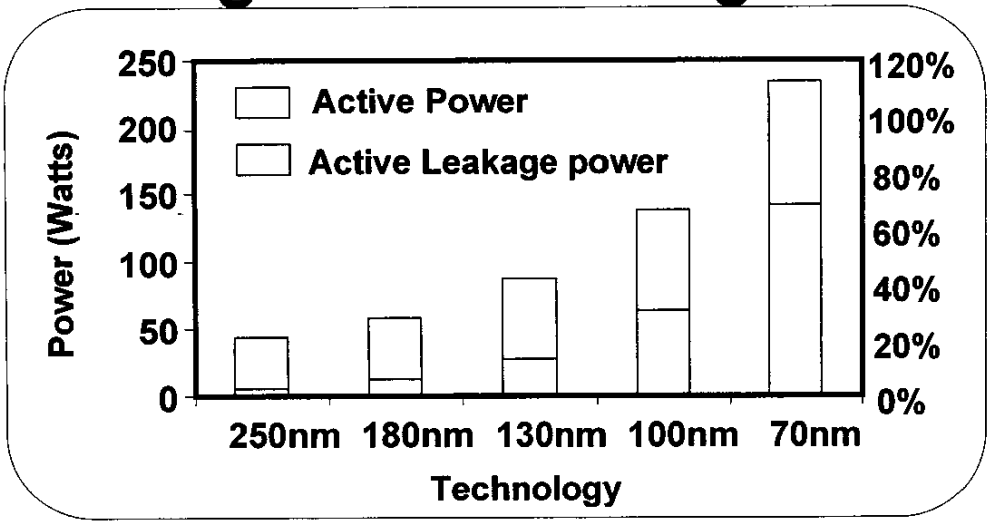
Microprocessor Power Trend



- C scales by 30% per generation...
- ...but Vcc scales by 10-15% only
- Must maintain or reduce power in future

52

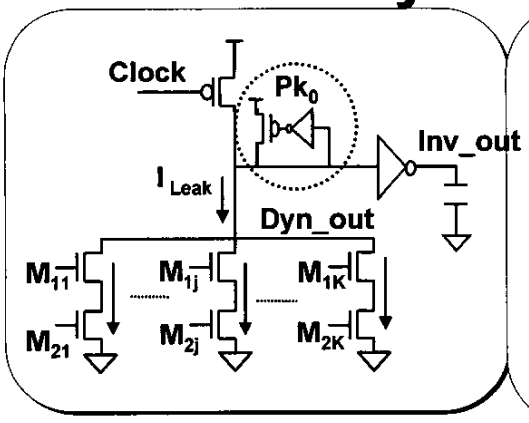
Leakage vs. Switching Power

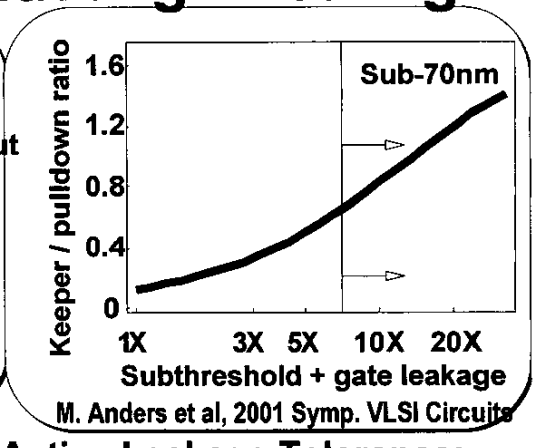


- I_{off} increase 3-5X per generation
- Active leakage power > 50% of total power
- Aggressive active leakage control required

53

Functionality with High Leakage



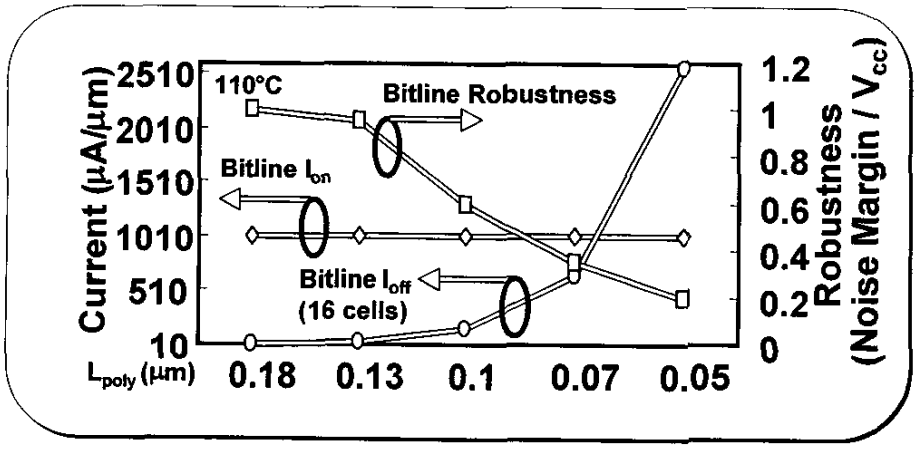


Sub-70nm Dynamic Circuit Active Leakage Tolerance:

- Cache, RF, Arrays, Bitlines most affected
- Keeper sizes > 50% of pulldown strength
- High contention ⇒ degraded performance
- Slow keeper shutoff ⇒ high short-circuit power

54

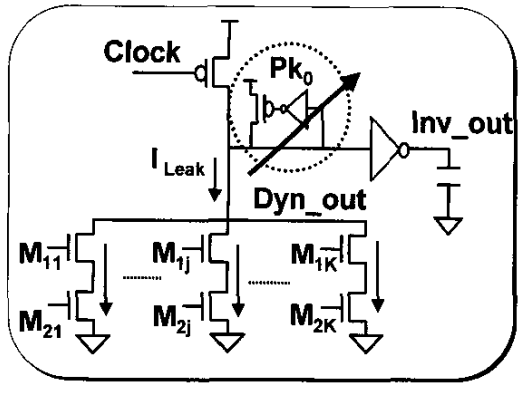
Bitline Leakage Tolerance

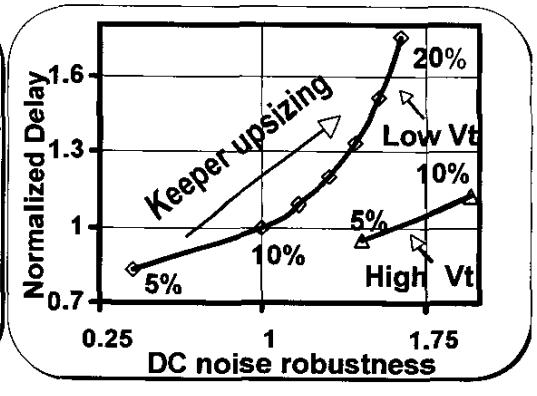


- Bitline I_{on}/I_{off}: 60% ↓ per generation
- Leakage tolerant bitline techniques required

55

Improving Dynamic Leakage Tolerance: Keeper Upsizing



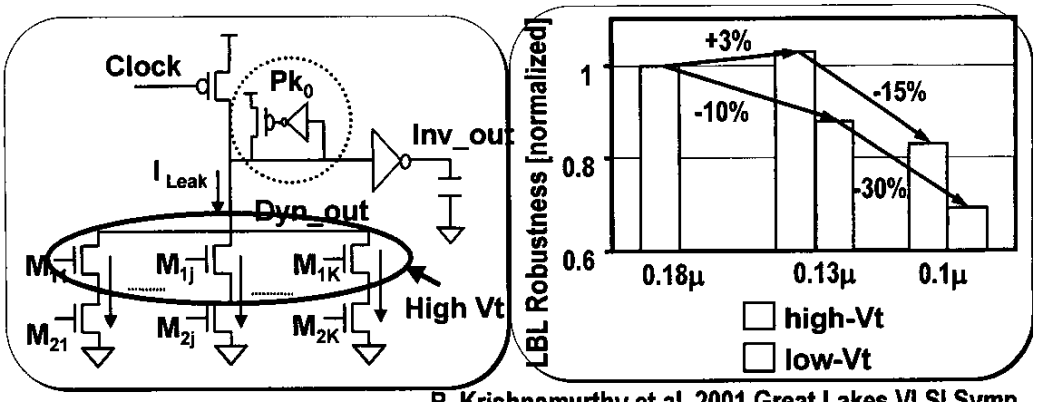


A. Alvandpour et al, 2001 Symp. VLSI Circuits

- Robustness = DC Noise Margin / Vcc
- Traditional noise engineering ⇒ diminishing ROI

56

Dual Vt Scaling Trends

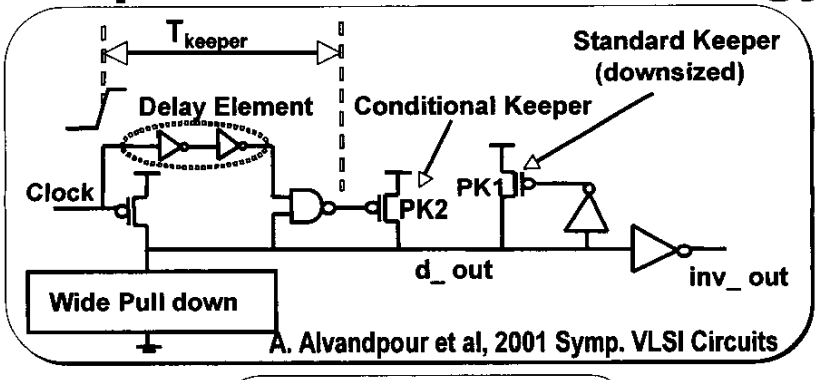


R. Krishnamurthy et al, 2001 Great Lakes VLSI Symp.

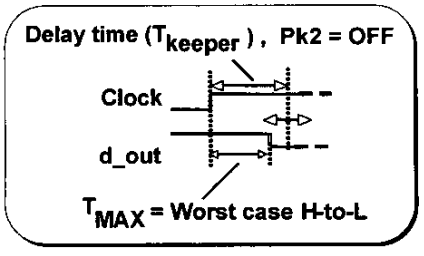
- Replace NMOS pulldowns with high-V_t
- Good one-time solution for 130nm node
- 15-30% degradation for both high- and low-V_t in sub-130nm
- Dual-Vt bitlines don't scale well beyond 130nm

57

Leakage-tolerant Conditional Keeper Domino Technology

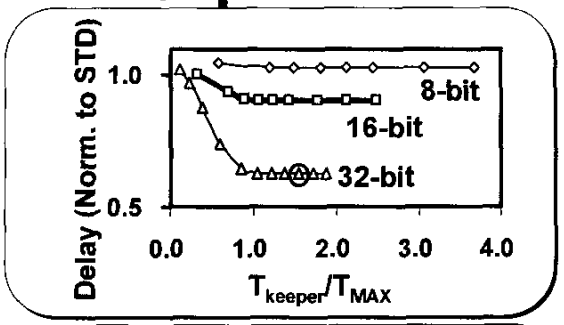


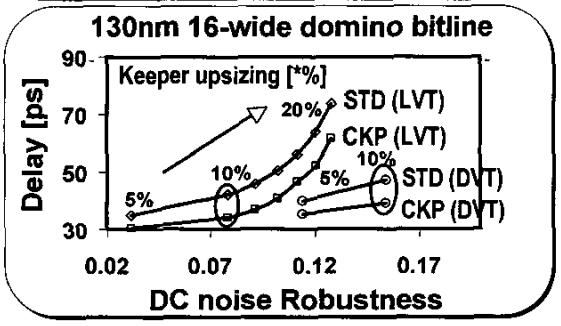
A. Alvandpour et al, 2001 Symp. VLSI Circuits



58

Leakage-tolerant Conditional Keeper Domino Technology

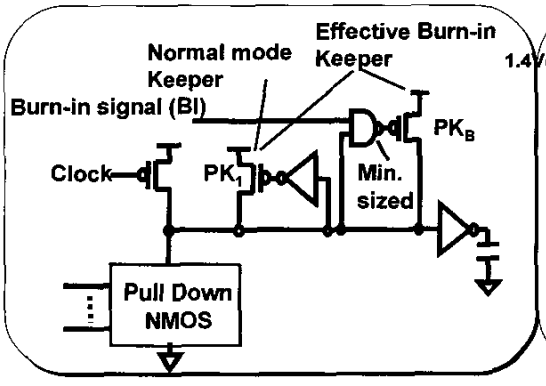


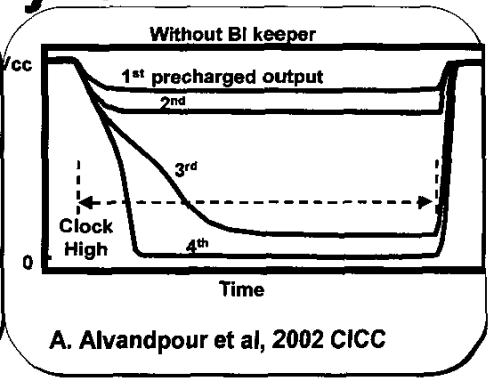


- Motivation:
- Weak keeper (low contention) during evaluation window
- Strong keeper activated only if dynamic node "high"
- 20% delay reduction at same robustness
- High-performance "dual-V_t enabler"

59

Burn-in Tolerant Dynamic Circuits



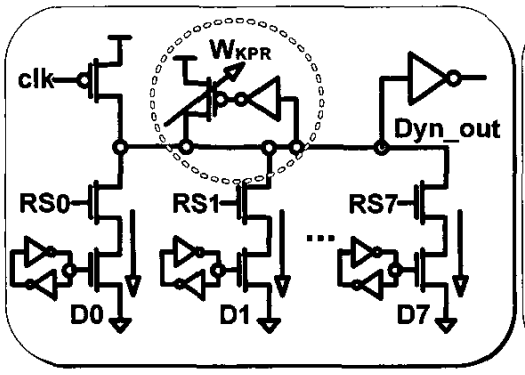


A. Alvandpour et al, 2002 CICC

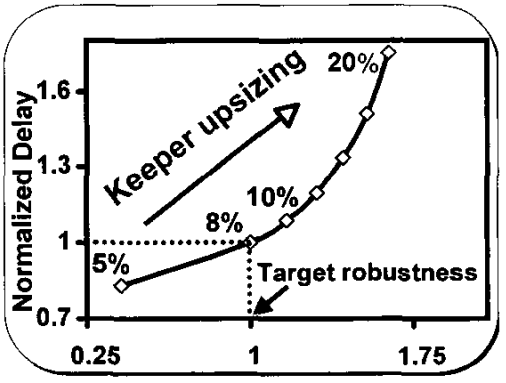
- Leakage sensitive circuits not functional at burn-in
  - Elevated supply and temperature
- Larger keepers increase delay at "normal" condition
- Conditional keeper enables functional burn-in testing
- 2X lower noise during burn-in
- 50% better delay than upsizing BKM keeper

60

Leakage Variations Impact



Dynamic 8-way Bitline

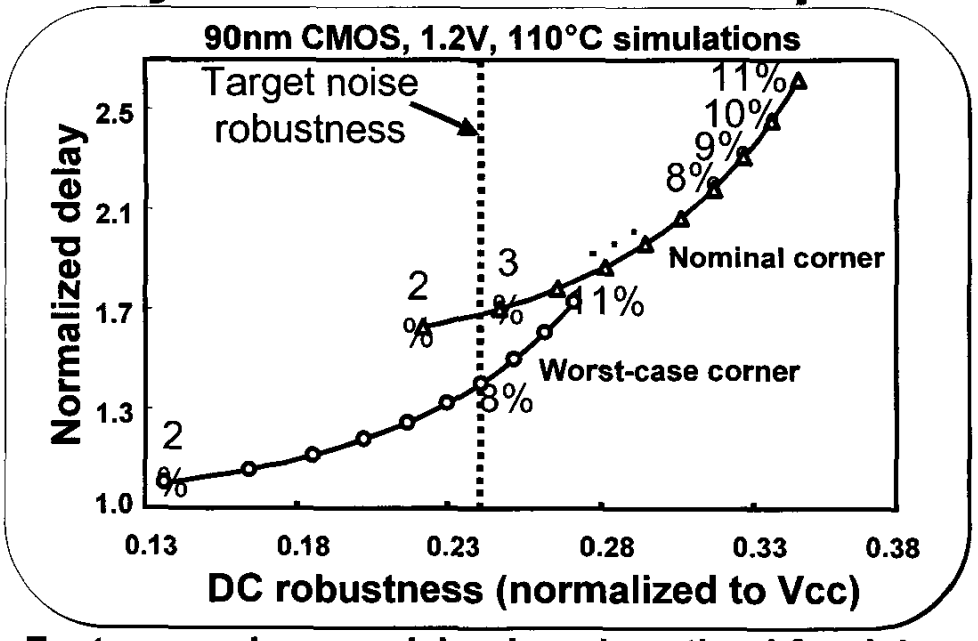


DC noise robustness

- Dynamic circuit NMOS pulldown leakage variation:
  - Keeper size determined for target robustness at worst-case leakage corner
  - Excess leakage dies: fail to meet target robustness
  - Lower leakage dies: over-designed for robustness

61

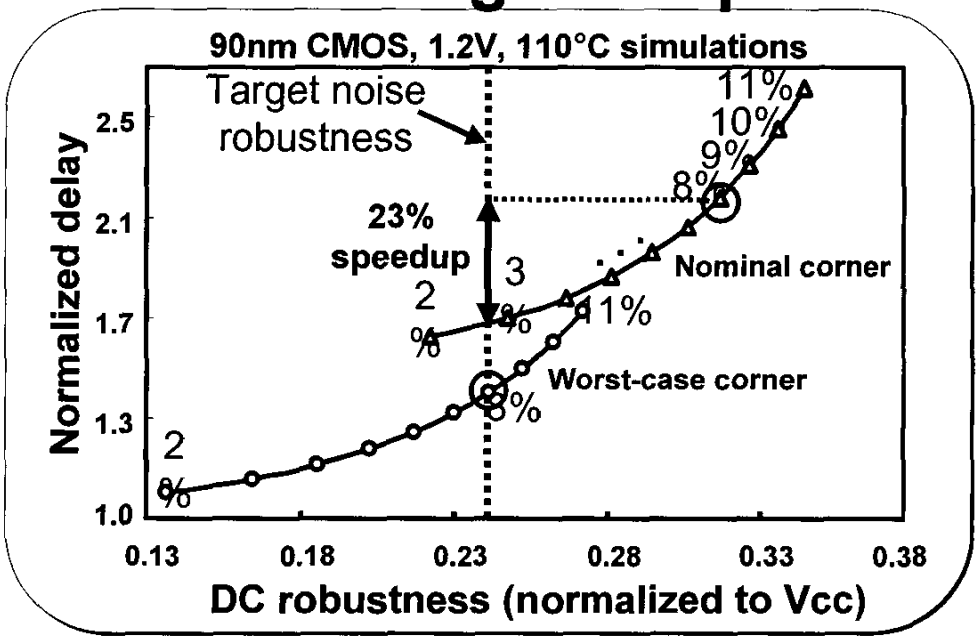
Delay and Robustness Spread



- Fast corner keeper sizing is sub-optimal for delay

62

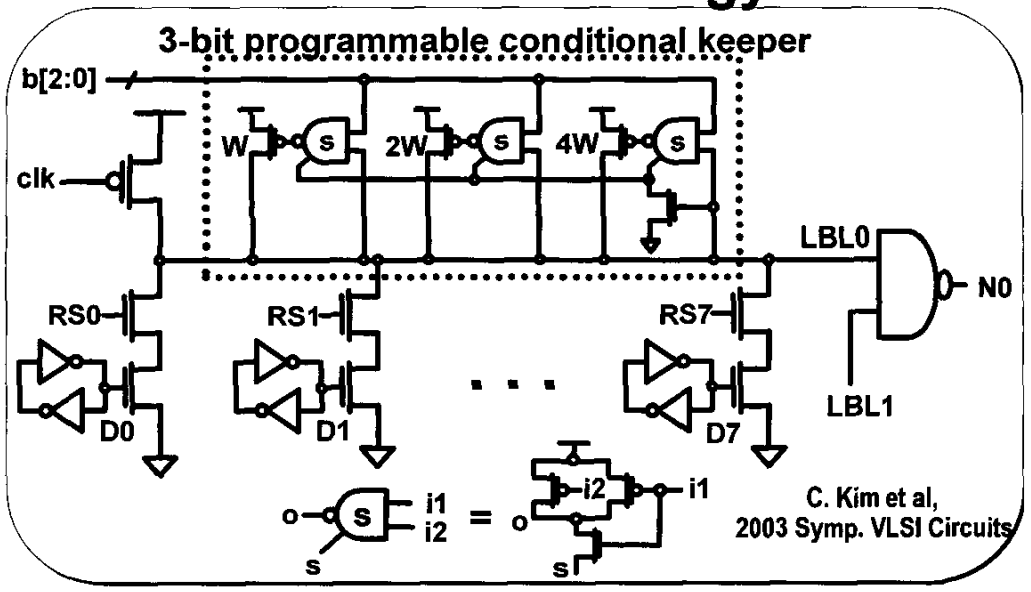
Variable Strength Keeper Size



- Goal: downsize keeper on nominal leakage dies

63

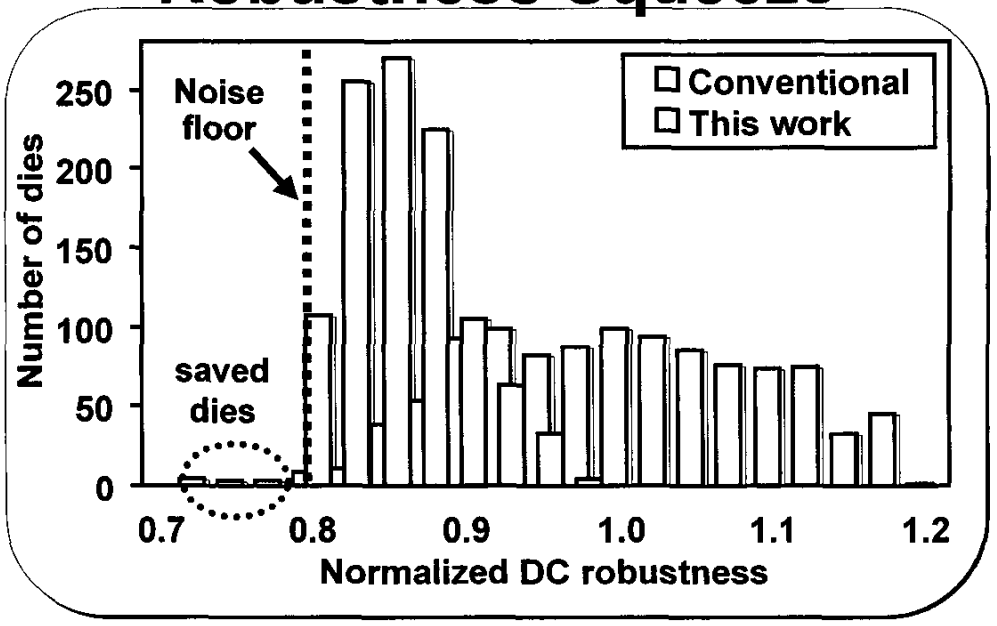
Process Compensating Dynamic Circuit Technology



- Shared-NAND: 2 less NMOS devices, dense layout

64

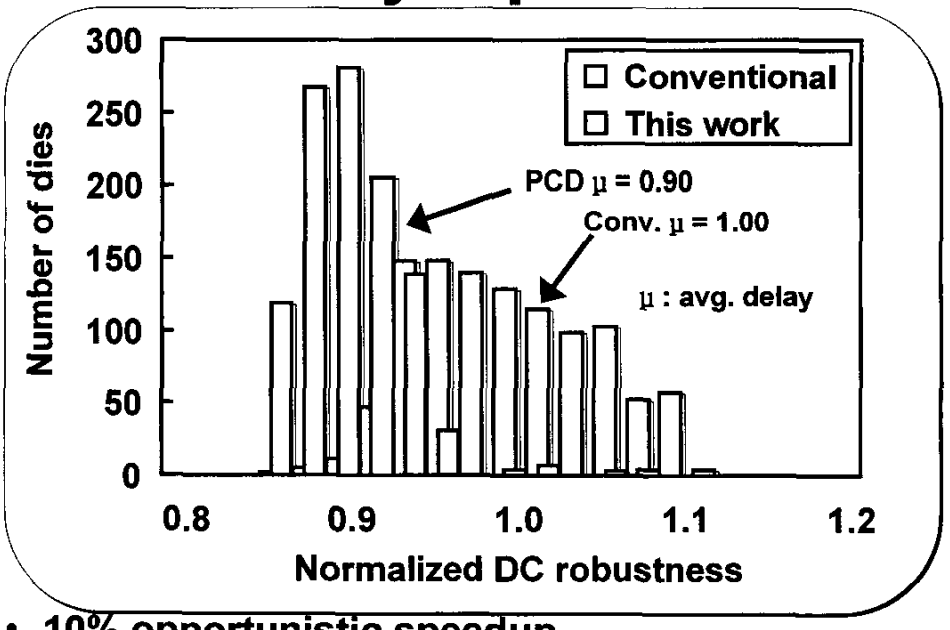
Robustness Squeeze



- 5X reduction in robustness failing dies

65

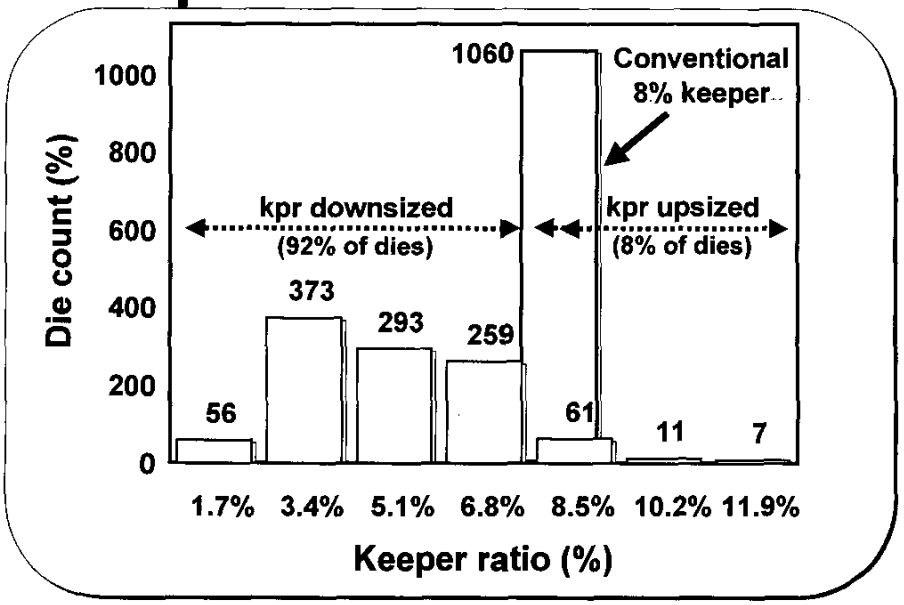
Delay Squeeze



- 10% opportunistic speedup

66

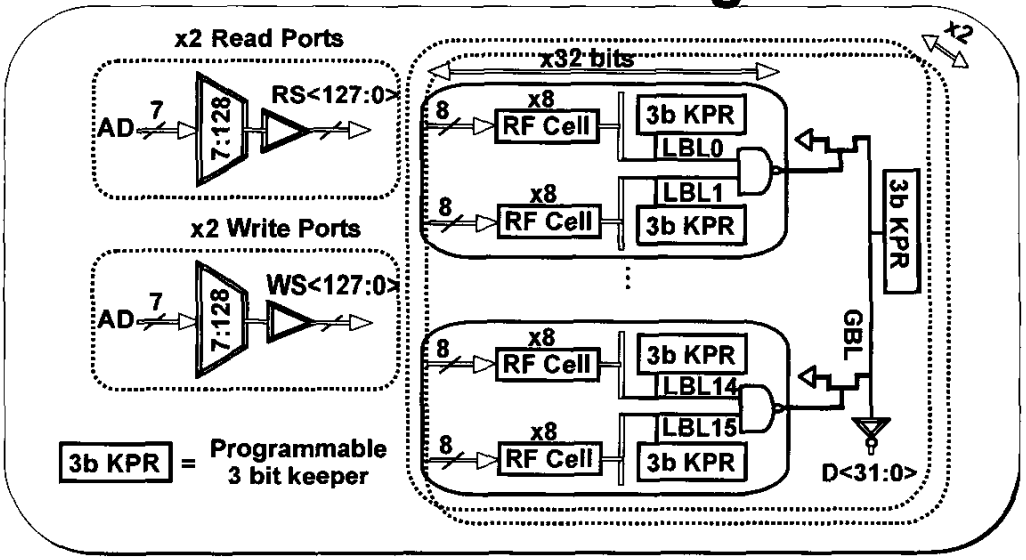
Keeper Ratio Distribution



- Keeper downsized in 92% of dies

67

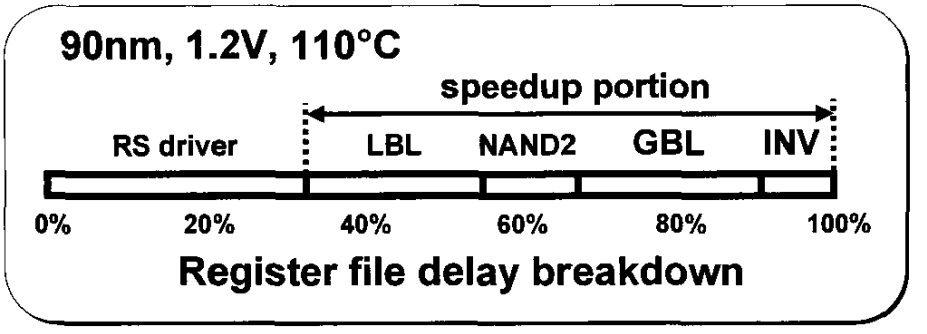
128x32b 2R2W PCD Register File



- Single-ended read, 8 bitcells/LBL, 8-way GBL
- Keeper folded into existing layout templates

68

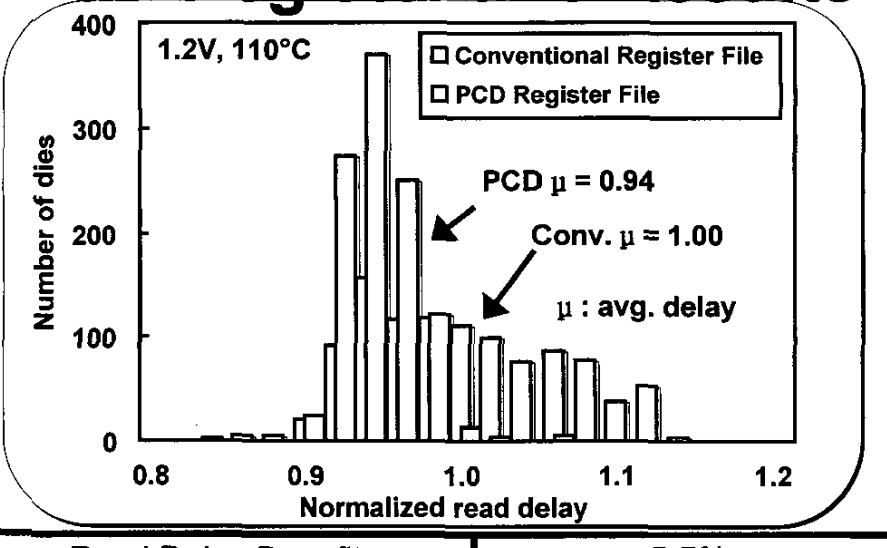
PCD Register File Delay, Energy



- **Speeds up 67% of RF critical path delay**
- **2% worst-case total energy overhead**

69

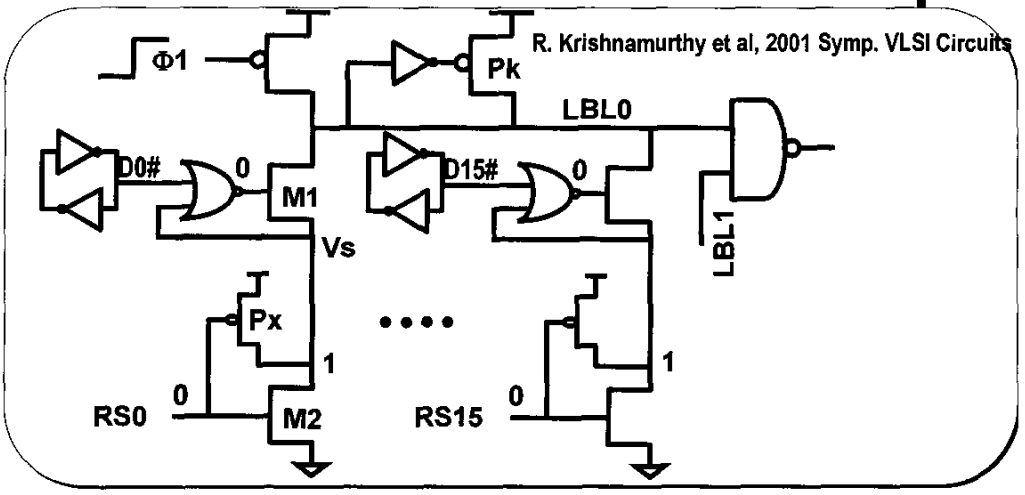
PCD Register File Results



Read Delay Benefit	5.5%
Robustness Failing Dies	0.2% (5X ▼)
Read Delay Variation: σ/μ	6.1%→2.3% (2.7X ▼)

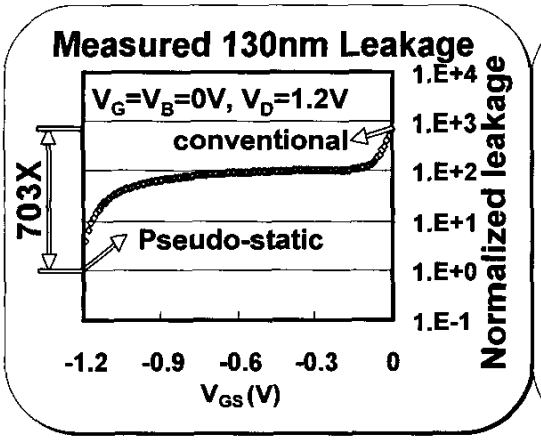
70

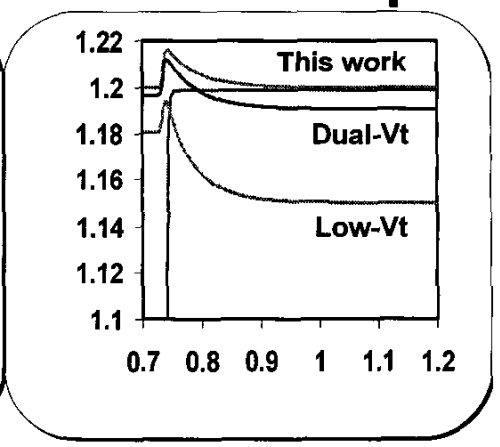
Pseudo-static Bitline Technique



- Goal: $V_{GS} = -V_{cc}$ and $V_{DS} = 0$ on deselected bitline's access transistors
- No oxide stress or additional bias voltages

Pseudo-static Bitline Technique





- 130nm measurement: 703X reduction in bitline leakage (~4 process generations)
- Scalable replacement for dual-Vt bitlines

6GHz 130nm Pseudo-static RF

130nm, 1.2V, 110C

LBL Scheme	Read Delay	DC robustness (DC noise margin/Vcc)	Energy/transition (normalized)
Low-Vt	158ps	0.072	1.0
Dual-Vt	178ps	0.157	0.95
This work	165ps	0.214	1.02

- **256-entryx32-bit 4-read, 4-write ported register file**
- **Single-cycle latency & throughput: performance critical**
- **6GHz operation (8% read delay improvement) with simultaneous 36% robustness benefit over dual-V_t**
- **Scalable to sub-130nm technologies**

73

Outline

- **Challenges & Circuit Solutions:**
  - **High-performance power-efficient execution core**
    - **6.5GHz single-rail domino 32-bit Han-Carlson ALU**
    - **4GHz semi-dynamic 32-bit sparse-tree AGU**
  - **Leakage-tolerant register files**
    - **Conditional/burn-in keeper**
    - **Pseudo-static bitlines**
  - **Low-power datapaths for DSP applications**
    - **1GHz 16-bit static multiplier**

74

A 90nm 1GHz 22mW 16x16-bit 2's Complement Multiplier for Wireless Baseband

Zeydel et al. VLSI Symp. 2003

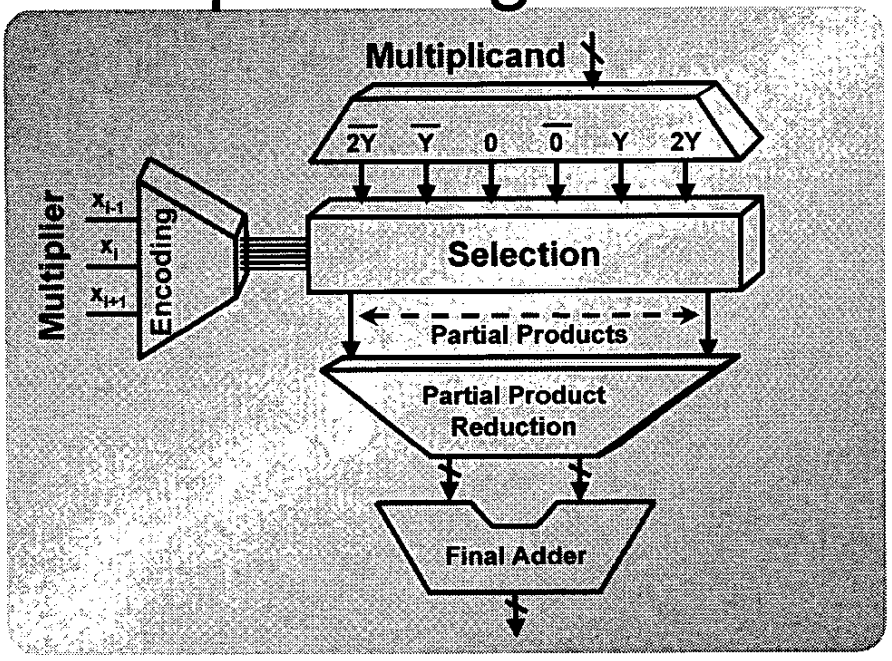
Intel Labs

Outline

- **Multiplier Block Diagram**
- **Booth Encoding and Select**
- **Optimized Partial Product Reduction**
- **Signal Arrival Optimized Final Adder**
- **90nm Energy-Delay Results**
- **Conclusions**

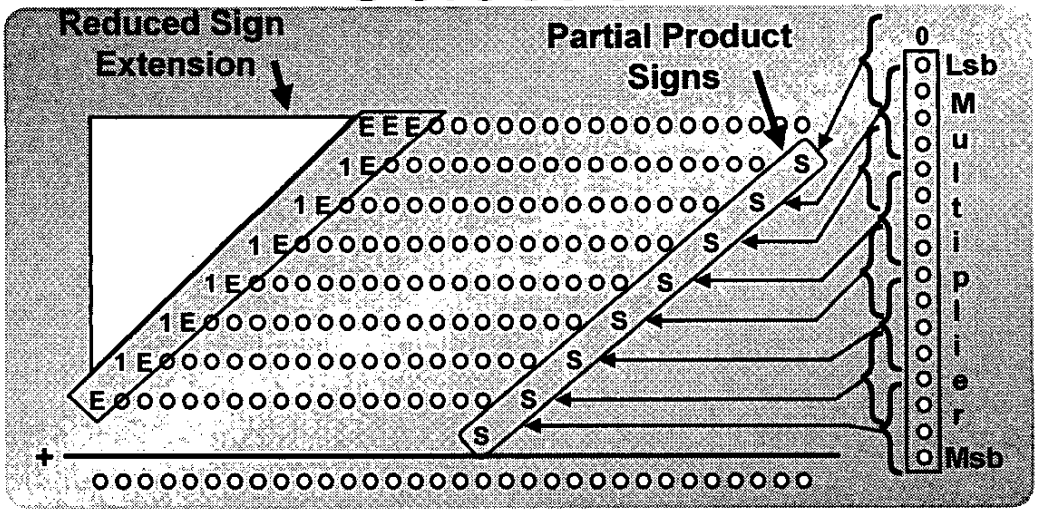
76

Multiplier Organization



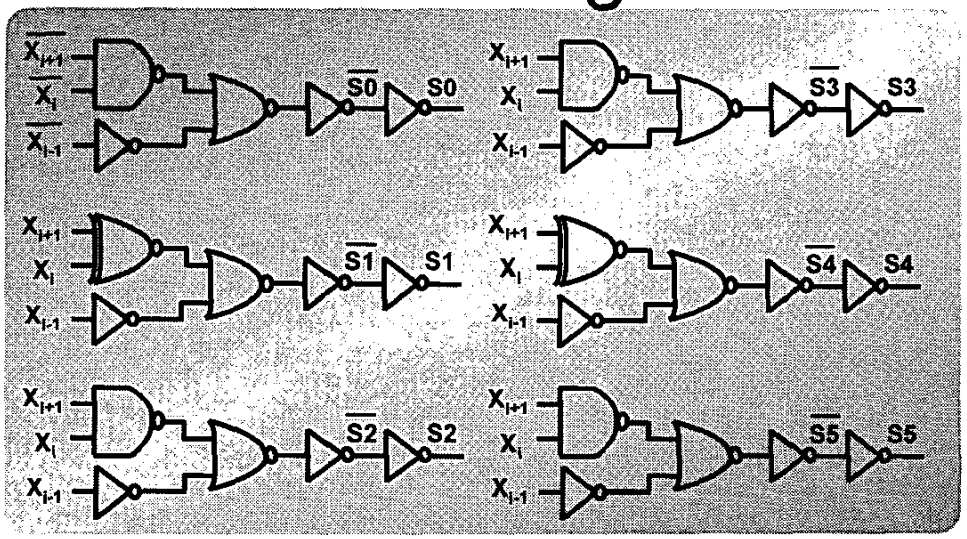
77

Booth Encoding and Selection



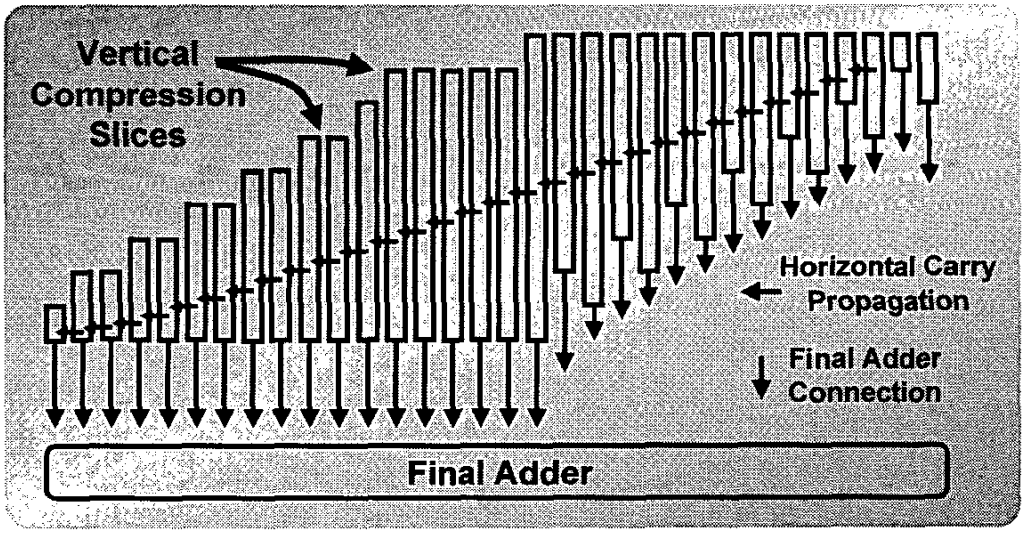
78

Booth Encoding Circuits



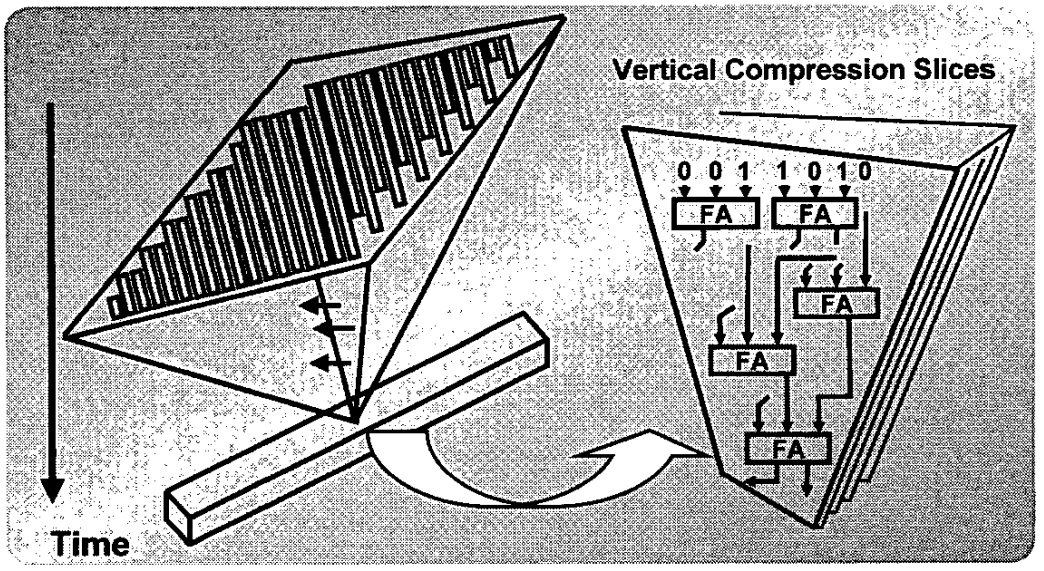
79

Vertical Partial Product Compression Tree



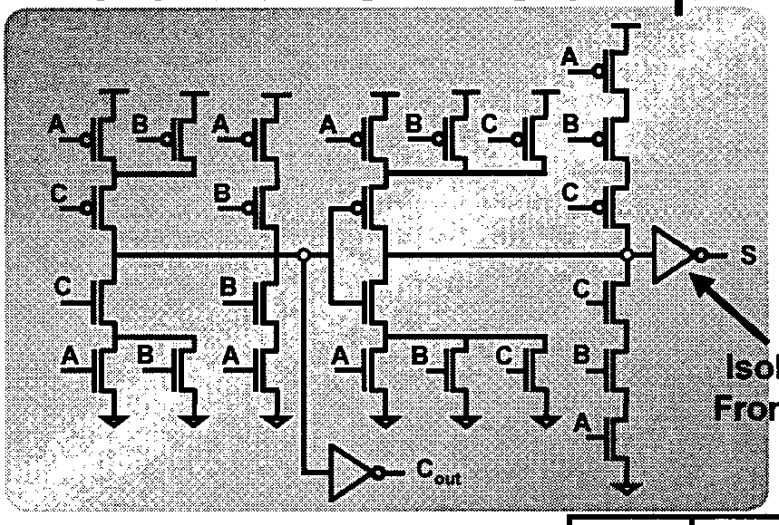
80

Partial Product Reduction Tree



81

PPRT 3:2 Compressor



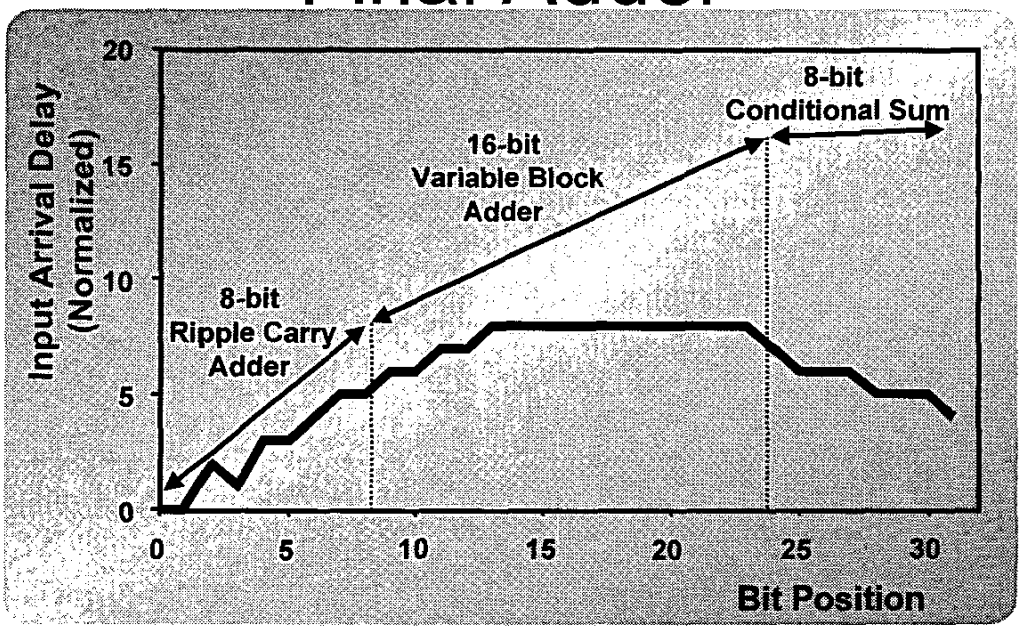
Isolates 3-stack From output load

0.8V 110˚C

Input	Worst-case Delay
Sum	240ps (1x)
Carry	165ps (0.69x)

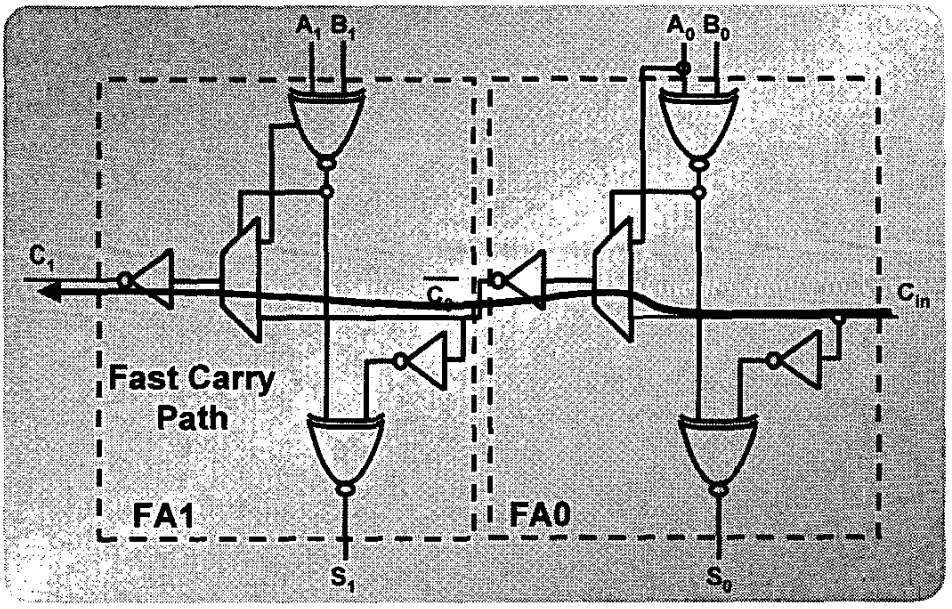
82

Final Adder



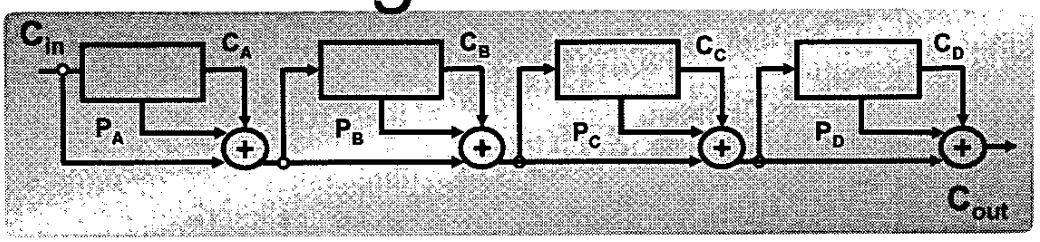
83

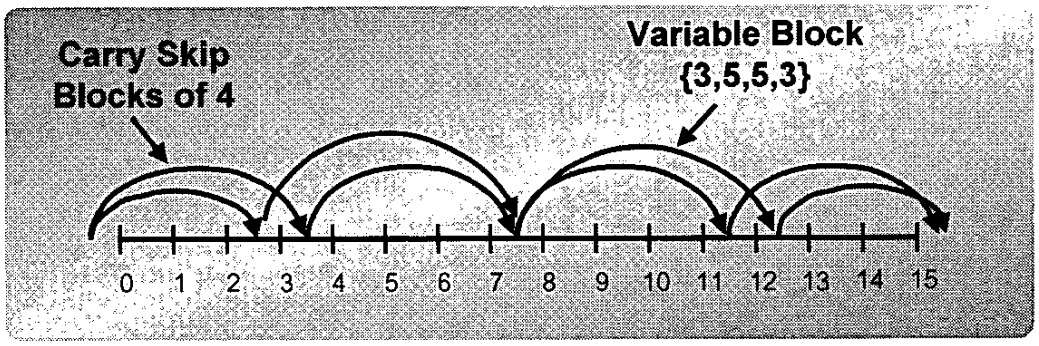
Fast Ripple Full Adders



84

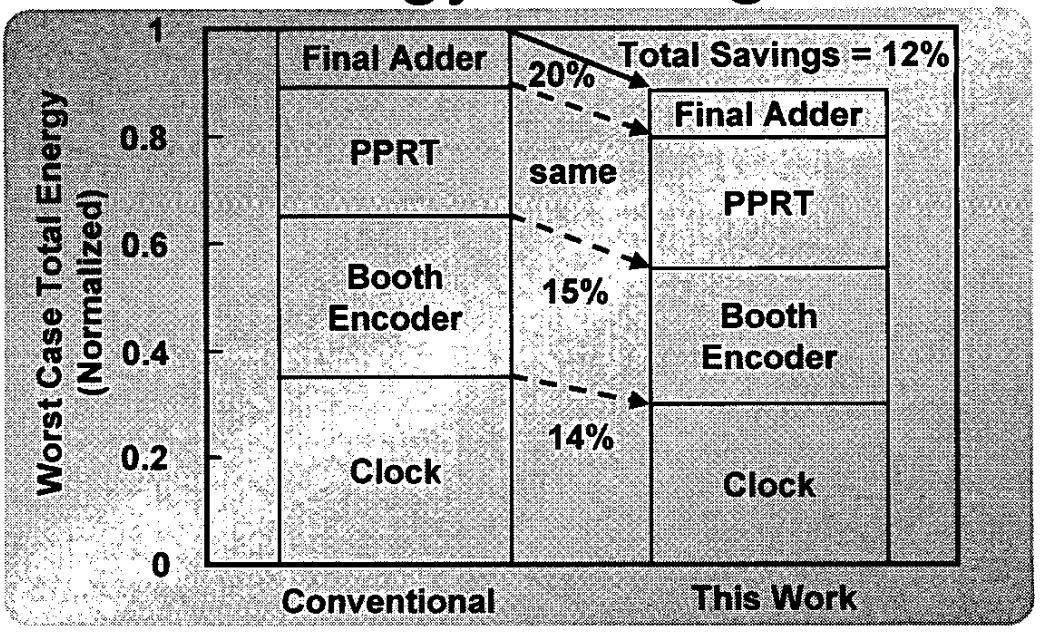
Carry Skip Adder Organization





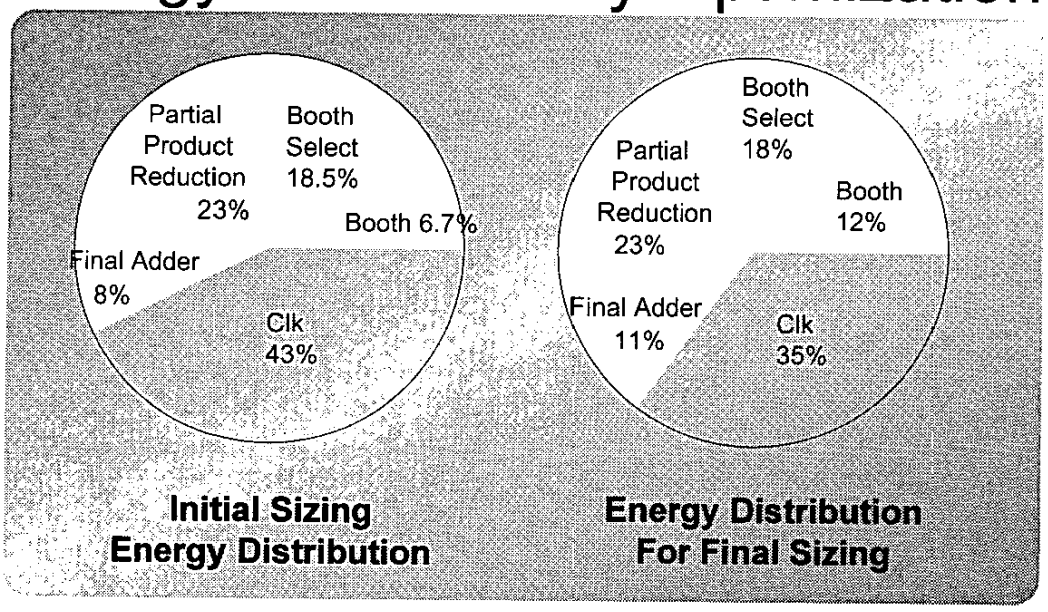
85

Energy Savings



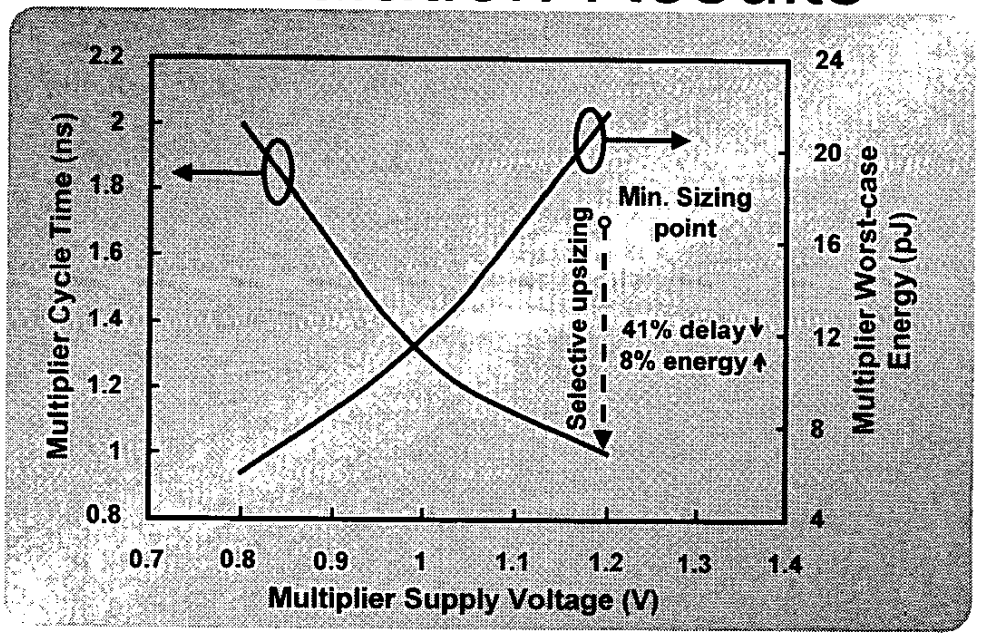
86

Energy Based Delay Optimization



87

Simulation Results

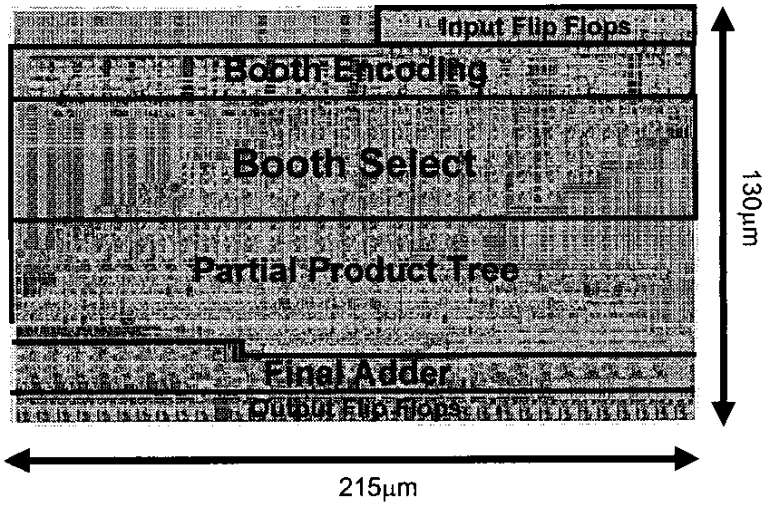


88

Multiplier Layout

0.8V: 500MHz, 3mW

1.2V: 1GHz, 22mW



89

Summary

- 16-bit multiplier features:
  - **Efficient Booth Encoding and Select**
  - **Delay and Area Optimized Partial Product Reduction Tree**
  - **Signal Profile Optimized Final Adder**
  - **Energy Optimized Sizing**
- Enables multi-mode operation
  - **1GHz at 1.2V 22mW**
  - **500MHz at 0.8V 3mW**

90