Slide Supplement

A Subnanosecond 0.5µm 64b Adder

S. Naffziger

Hewlett Packard Company, Fort Collins, CO

A subnanosecond 64b adder implementation in 0.5µm CMOS forms the basis for integer and floating-point units of a microprocessor. Integrating dual-rail dynamic logic and Ling's equations, the adder contains 7k transistors in 0.246mm^2. It executes a full 64b add and latches the result in under 1ns at nominal conditions.

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Motivation

Why are designers always working on faster adders?

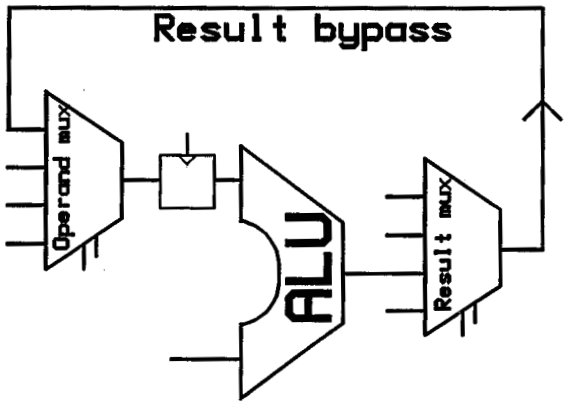
Adders occupy the critical path in many key areas of microprocessor operation

- Arithmetic Logic Units
- Memory address generation
- Floating point calculations

Continued...

Motivation: ALU

ALU operation in a typical CPU

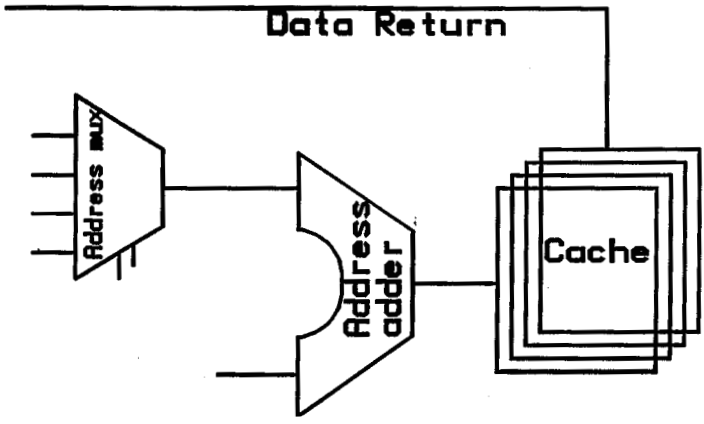


Operand to result bypass is often the cycle time limiter
☞A fast ALU (adder) is essential

Continued...

Motivation: Address Add

Time from address selection to cache data return will often be the chip frequency limiter. The address adder occupies a significant portion of this budget.



Continued...

Motivation: FPU

Adders occupy numerous critical paths in an FMAC.

- Generation of the 3X multiple for the Booth re-coded multiplier array
- Summation of the carry–save multiplier results (112 bit addition required here)
- Final IEEE rounding increment

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Design Goals for the PA8000 Adder

Given that:

- Over 20 Adders are used on the processor
- Many are situated in frequency limiting paths

The adder must:

- Be as fast as possible
- Be area efficient
- Be compatible with reliability and composition goals of the rest of the processor

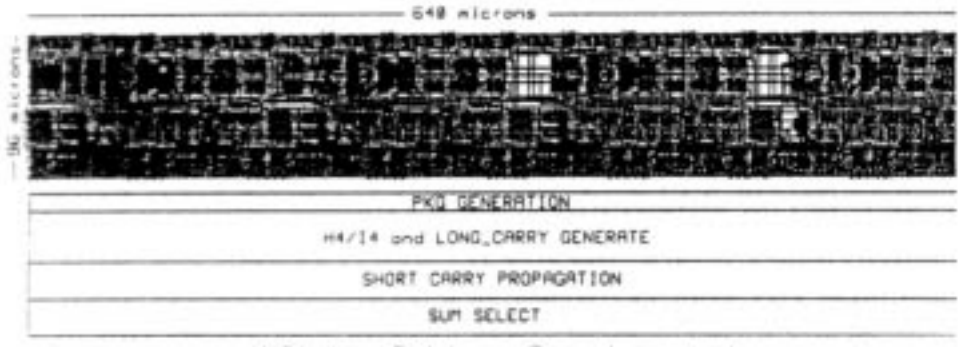
Continued...

Key Features of the Adder Design

- Dual rail dynamic logic
  - Fastest logic family for this CMOS technology
  - Enables seamless interface to both dynamic and static logic downstream
- Distributed carry chain and logical reductions enable a dense layout
- Use of Ling's equations in dynamic CMOS enable a 4 gate delay 64b add

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger



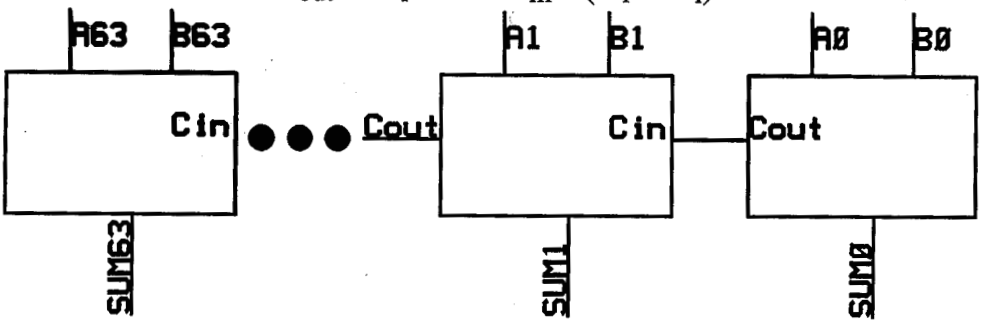
16bit Adder Quadrant Layout
In 3 layers of metal, .6u geometry

Continued...

Adder Design

$$Sum = A \oplus B \oplus C_{in}$$

$$C_{out} = A_i \cdot Bi + C_{in} \cdot (A_i + B_i)$$



Problem: Carry ripple from LSB to MSB for a 64 bit adder requires 65 gate delays.

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Speeding Up Addition

- Carry look ahead and sum select techniques
- Ling's equations
- Fast circuit techniques
  - Pass gate logic
  - Low voltage swing
  - Bipolar (BiCMOS) logic
  - Dynamic CMOS

Continued...

Carry Lookahead Techniques

Calculate groups of carries (in our case, 4 at a time) in **parallel** to reduce the carry calculation time to **$2*\log_r[n]+2$** gate delays for fanin 4 gates.

r = group size, **n** = number of bits to add

$C2 = G1 + G0 \cdot P0$

$C4 = G3 + G2 \cdot P3 + G1 \cdot P2 \cdot P3 + G0 \cdot P1 \cdot P2 \cdot P3$

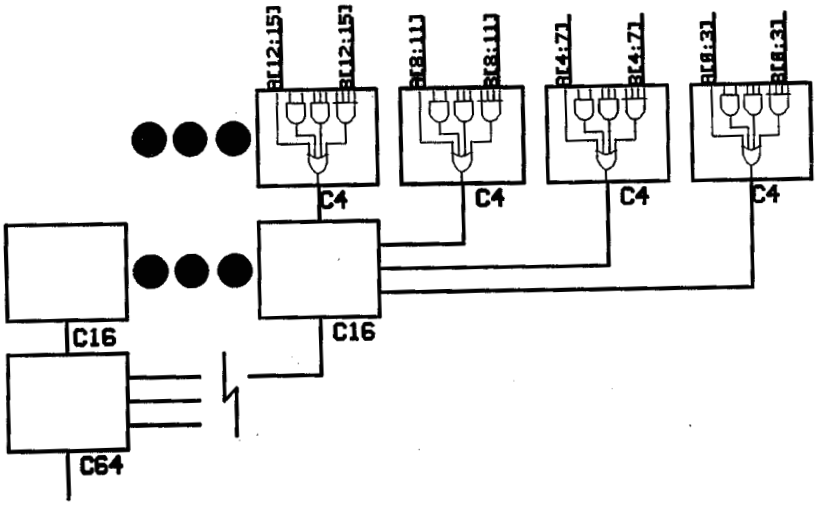
A	B	P	G	K
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	0 or 1	1	0

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Carry Lookahead Techniques

These group carries are combined at another level to **produce** a long carry and so on.



Continued...

Sum Select

- Calculate two sums per block, one with LSB carryin of 0, another with carryin of 1
- When the long carry comes, select the correct sum.

Total gate delays for CLA/sum select group of 4:

$1 + 2*\log_4[64] + 1 = 8$ gate delays

Continued...

Using Ling's Equations

- Define a group of 4 "psuedo-carry"

$$H_4 = G_3 + G_2 \cdot P_2 + G_1 \cdot P_1 \cdot P_2 + G_0 \cdot P_2 \cdot P_1 \cdot P_0$$

VS.

$$C_4 = G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_3 \cdot P_2 \cdot P_1$$

- $C_4 = H_4 \cdot P_3$

Continued...

Using Ling's Equations

Why is the H_4 equation a good thing?

- If $P_x = A_x + B_x$, then $G_x \Rightarrow P_x$
- Use this reduction to define H_4 in terms of the in put operands, not P and G

$$H4 = A_3B_3 + A_2B_2 + A_2A_1B_1 + B_2A_1B_1 + A_2A_1A_0B_0 + A_2B_1A_0B_0 + B_2A_1A_0B_0 + B_2B_1A_0B_0$$

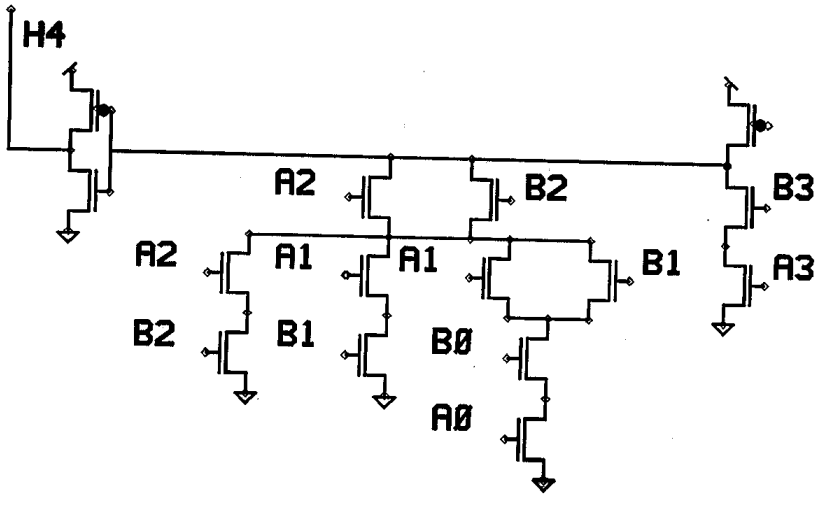
Calculate H_4 in **8 terms, fanin of 4** vs. **15 terms fanin 5** for a similarly flattened C_4, or **one** gate delay vs. **three**

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Using Ling's Equations

$$H4 = A_3B_3 + A_2B_2 + A_2A_1B_1 + B_2A_1B_1 + A_2A_1A_0B_0 + A_2B_1A_0B_0 + B_2A_1A_0B_0 + B_2B_1A_0B_0$$



Continued...

Using Ling's Equations

A fast group of 4 propagate is also necessary to combine for the next level (16) of carry generate:

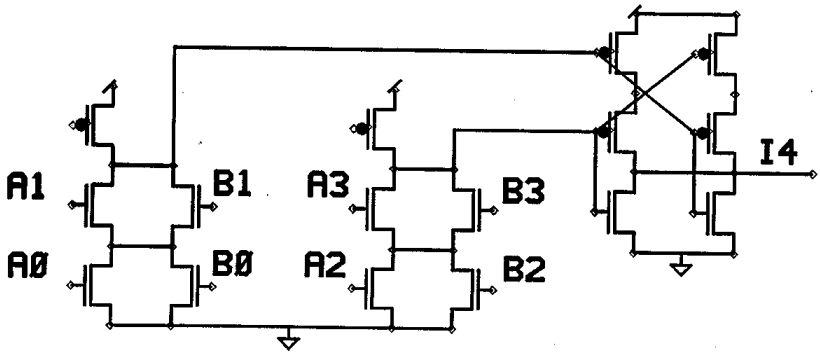
$$I_4 = P_0P_1P_2P_3$$

If P = A + B, this can be done in one gate delay using "wired–or" techniques also.

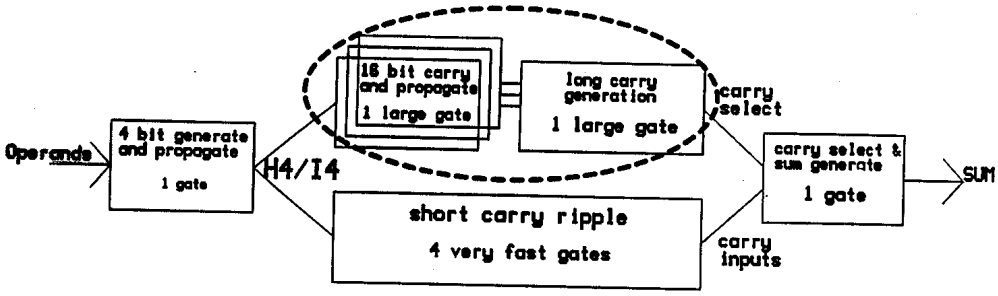
Continued...

Using Ling's Equations

$$I_4 = P_0P_1P_2P_3$$



Continued...



64 bit add critical path

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

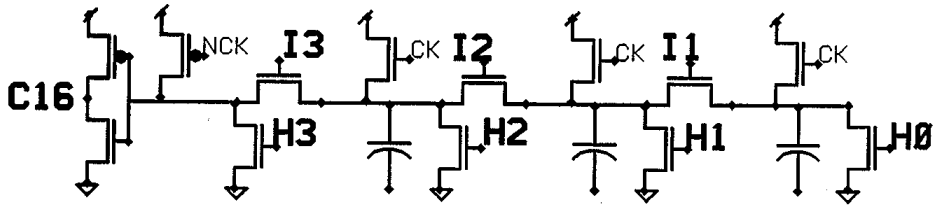
Long Carry Propagate

Combine I_4 and H_4 to produce C_{16}

- ❑ As fast as possible since this is directly in the critical path
- ❑ Minimum area
- ❑ Minimum wires
  - ▶ Use a combination of Dynamic, Pass gate, and Low Voltage techniques to optimize for these three parameters

Continued...

$$C_{16} = H_3 + H_2 \cdot I_3 + H_1 \cdot I_2 \cdot I_3 + H_0 \cdot I_1 \cdot I_2 \cdot I_3$$



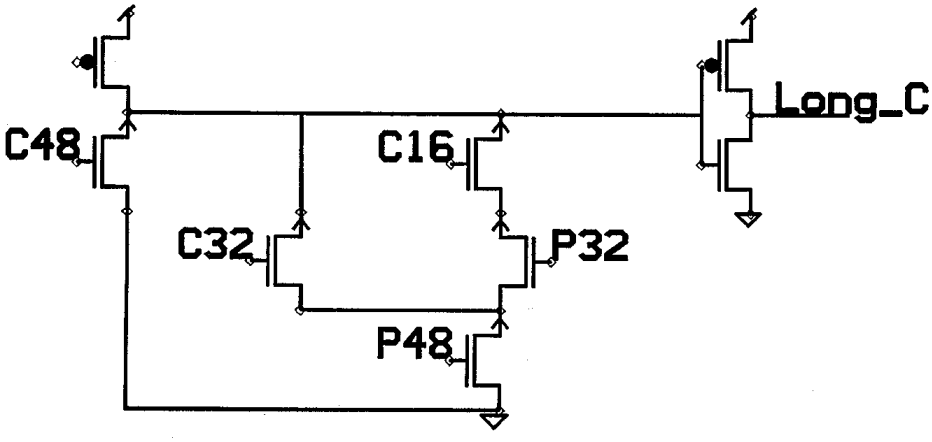
Long Carry Propagate

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

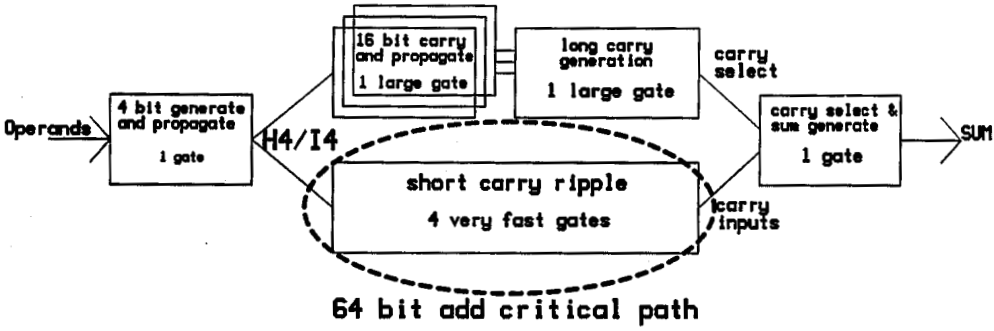
Combine group of 16 carries for final long carry:

$$\mathbf{Long\_C = C_{48} + C_{32} \cdot P_{48} + C_{16} \cdot P_{32} \cdot P_{48}}$$



Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

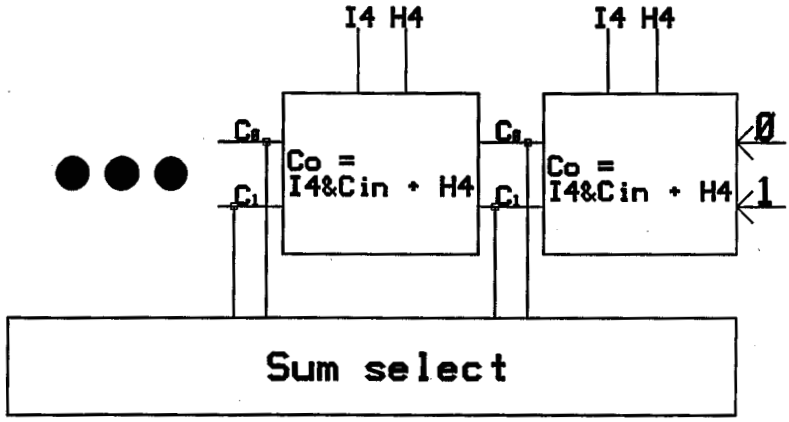


Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Both flavors of Local Carry must be generated

- Before the long carry arrives
- In very little area due to high replication factor

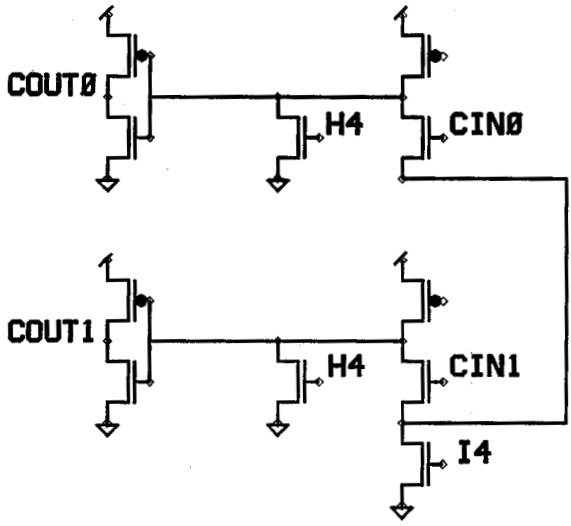


Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Short Carry Generate

Use a by–four scheme $C_o = I_4 \cdot C_{in} + H_4$



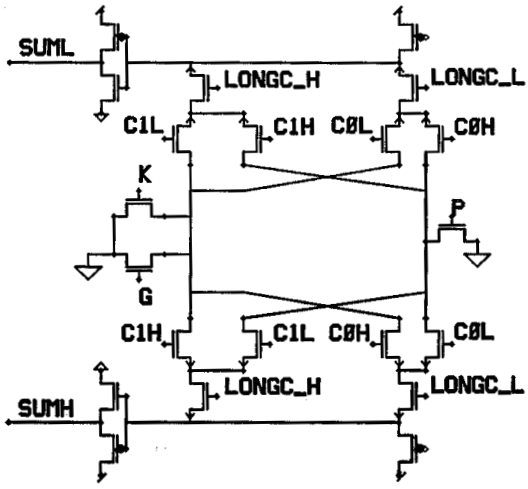
Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Sum Generate and Select

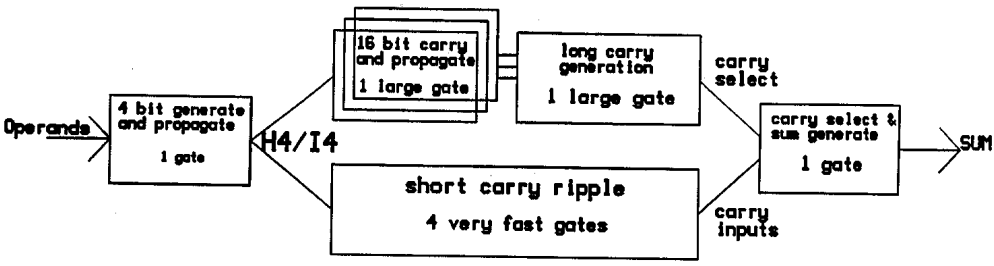
Sum = A ⊕B⊕C_{in}

C_{in} is selected from C_0 or C_1 by C_{long}



Continued...

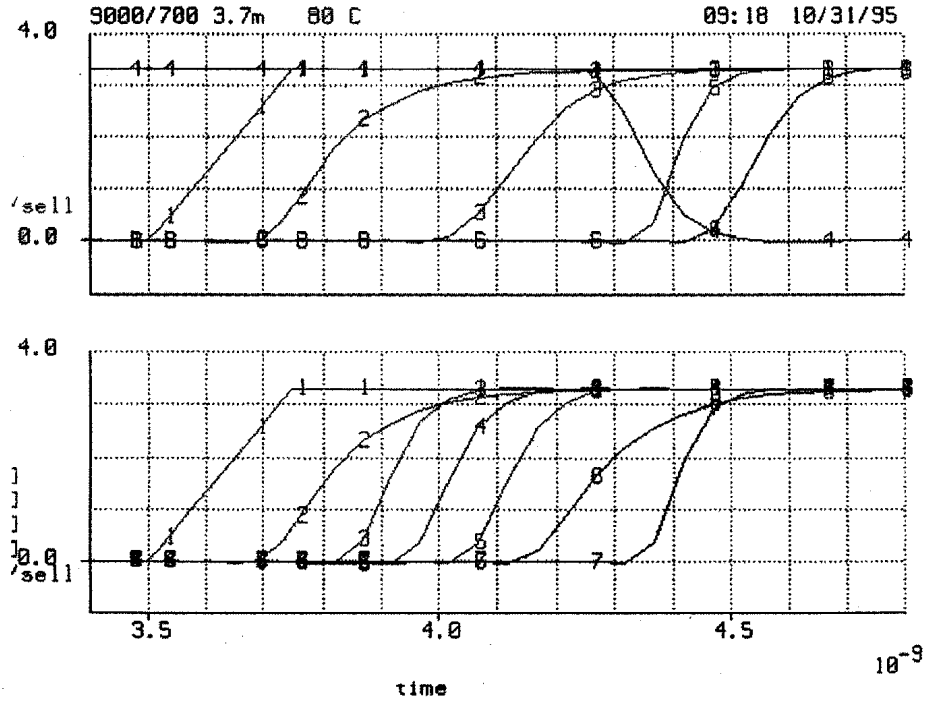
Summary



64 bit add critical path

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger



Continued...

Performance Analysis

- SPICE analysis indicates that the delay from operands to sum is **.930 nanoseconds** under Nominal voltage (3.3), Nominal FETs, 80C
- Silicon confirms simulated speed based on characterization of numerous paths involving the adder

Continued...

A Sub-Nanosecond 0.5mm 64b Adder Design
Samuel Naffziger

Source

1996 IEEE International Solid-State Circuits Conference
1996 Slide Supplement, pp. 290-291