## SP 22.5: A Sub-Nanosecond 0.5μm 64b Adder Design

Samuel Naffziger

Hewlett-Packard Co., Fort Collins, CO

A sub-nanosecond 64b adder in 0.5μm CMOS forms the basis for the integer and floating point execution units. Integrating dual-rail dynamic CMOS and use of Ling's equations, the adder is composed of 7k FETs in 0.246mm² and performs a full 64b add, operands to result in <1ns (7 fanout of 4 inverter delays) under nominal conditions.

Addition time is important to CPU design. Add latency is in the critical path of such important areas as memory address calculation, ALU evaluation and floating point computation. Current state of the art addition hardware [1-3] is not adequate for 64b computing, and this prompted a new design that forms the core of the execution hardware. It incorporates architecture and circuit design techniques that enable it to achieve low latency for adders of width from 13b to 112b. These techniques for the 64b adder design are described here.

Conceptually, an adder is:

$$Sum = A \oplus B \oplus Carry\_in$$
$$Carry = (A * B) + (A + B) * Carry\_in$$

Rippling the carry one bit at a time however is clearly unacceptable for a 64b adder so a fusion of carry-look ahead and carry-select techniques is implemented. Carry-look-ahead techniques involve parallel calculation of groups of carries (in this case, 4 at a time) in a modular fashion that reduces the carry calculation time to $Log_r[n] + 2$ gate delays where r is the group size and n is the width of the adder. Most current adder designs are based on this scheme [1, 2]. The equations for a group of 4 carry look-ahead are:

$$C4 = G3 + G2*P3 + G1*P2*P3 + G0*P1*P2*P3$$

Where the G and P terms are the generate and propagate terms derived from the operands (G=A*B, P=A+B). These groups of carries can then be combined at another level to produce a longer carry look-ahead using the same equations. Hence, for a traditional group of 2 carry look-ahead adder delay:
1 (generate GPK terms) + $\log_2[64]$ + 1 (Final XOR) = 8 gate delays.
If the technology of implementation allows the greater fanout and fanin of group of 4 carry look-ahead, the gate delays are:

$$1 + \log_4[64] + 1 = 5 \text{ gate delays.}$$

A sum select method is often used that goes ahead and calculates two sums based on Carry_in = 0 and 1 for a group using duplicate carry chains. When the actual carry into that group becomes available via look ahead, the correct sum is selected.

Ling's equations can be used to reduce delay further. Ling developed a series of equations specifically to make use of the dot or capability of ECL logic [4]. The result is the definition of a psuedo-carry calculated quickly with dot-or circuits. The relevant equations are as follows for a normal 4b carry:

$$C4 = G3 + G2*P3 + G1*P2*P3 + G0*P1*P2*P3$$

For a Ling "psuedo carry" [5]

$$H4 = G3 + G2*P2 + G1*P1*P2 + G0*P2*P1*P0$$

The propagate terms are simply shifted by one so C4 = H4*P3. When we define P as the OR of the operands, the G2*P2 term is redundant since G2 implies P2. This redundancy can be used to simplify all of the terms for H4 yielding an equation based on the actual operands, not P and G terms and can be calculated in one gate delay with fanin 4 logic:

$$H4 = A3*B3 + A2*B2 + A2*A1*B1 + A1*B2*B1 + A2*A1*A0*B0 + A2*B1*A0*B0 + B2*A1*A0*B0 + B2*B1*A0*B0$$

This is simpler (8 terms, fanin of 4 vs.15 terms, fanin of 5) than the expansion of the C4 equation in terms of the operands. With careful use of X terms in the carry propagation path, the conversion of H4 into C4 can be accomplished with no penalty.

To perform a CLA, 4b propagate terms must be calculated in at least the same time as the H4 terms that can then be combined for higher level look-aheads. These 4b propagate terms (I4 = P3*P2*P1*P0) can be calculated using a wired-or to generate $P_n$. This wired-or capability can be duplicated in dynamic CMOS using multiple NFET pulldown legs on a precharged node. The resulting circuits that generate H4 and I4 directly off the input operands are shown in Figure 1. The H4 and I4 terms are combined in a distributed Manchester gate to produce the block of 16 carry signals:

$$C16 = (H0+I0)*I1*I2*I3 + H1*I2*I3 + H2*I3 + H3$$

Where the Hx Ix terms are for the x'th group of 4 in the 16b quadrant. These 16b carries are combined in 1 more gate to produce the final carry select signals to the upper quadrants of the 64b adder. In parallel with this long carry generation, a carry ripple is performed in each of the 16b quadrants that generates both values of the carry to be selected. To hide the psuedo-carry to real-carry conversion, the carry chain is shifted over one bit and a special C3 (or carry out of 3b) term is used from the H4 gate in combination with G0 (Figure 2). In this way the I4 terms can be used directly in the carry-ripple chain as well. Finally, the carry select and sum generate are performed in a single gate (Figure 3). The critical path in the 64b adder (Figure 4) involves 4 total gate delays: (H4/I4 generation) + (C16 gen) + (Long carry gen) + (Sum select). The fast fanout of 1 carry ripple that occurs in parallel with the look-ahead, produces the local carries just ahead of the long carry select signal at each sum gate.

The high gain and fanin capability of dynamic CMOS along with the flexibility of dual rail are key enablers to the design. The adder requires dual monotonic (DCVS) inputs and produces a dual monotonic sum. The number of transistors for a 64b sum is 6924, that when layed out in 0.6μm geometry occupies 96x2560μm or 0.246mm² in 3 layers of metal. This adder occupies several critical paths in a microprocessor whose frequency of operation confirms simulated time at nominal process and voltage conditions of 0.93ns data in to sum out (Figure 5) [6].

*References:*

[1] Inoue, A., et al., "A 0.4mm 1.4ns 32b Dynamic Adder using Non-precharge Multiplexers and Reduced Precharge Voltage Technique," Symp. VLSI Circuits Digest of Tech. Papers, pp. 9-10, June, 1995.

[2] Dobberpuhl, et al., "A 200 MHz 64b dual-issue CMOS microprocessor," IEEE J. of Solid-State Circuits, Vol. 27, No. 11, Nov., 1992.

[3] Suzuki, M., et al., "A 1.4ns 32b CMOS ALU in Double Pass-Transistor Logic," IEEE J. of Solid-State Circuits, Vol. 28, No. 11, Nov., 1993.

[4] Ling, H., "High Speed Binary Adder," IBM J. Reasearch. Dev., Vol. 25, No. 3, p.156, May, 1981.

[5] Flynn, M., "Topics in Arithmetic For Digital Systems Designers," (Preliminary Second Edition) pp. 104-105, 1995.

[6] Heikes, C., G. Colon-Bonet, "Dual floating-Point Coprocessor with an FMAC Architecture," ISSCC Digest of Technical Papers, pp. 354-356, Feb., 1996.



0-7803-3136-2 / 96 / $5.00 / © IEEE

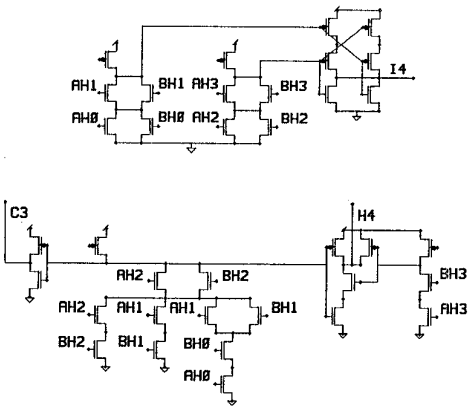


Figure 1: H4 and I4 generation circuits.

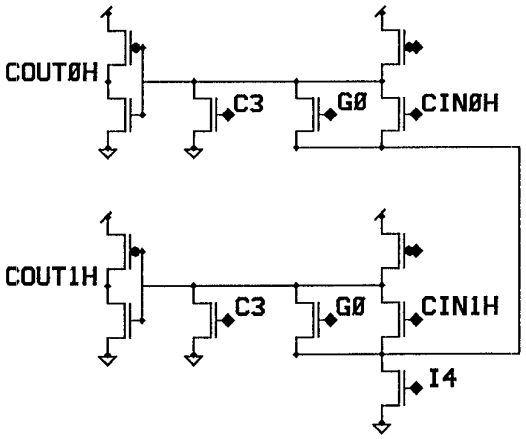


Figure 2: Carry ripple circuit.

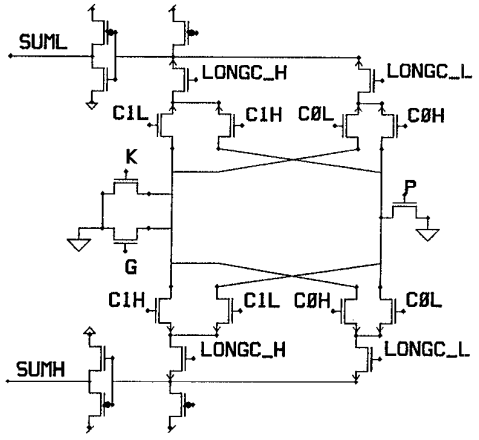


Figure 3: Sum select gate.

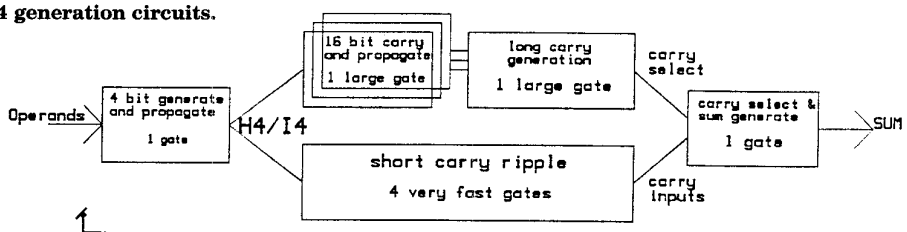


Figure 4: 64b add critical path.

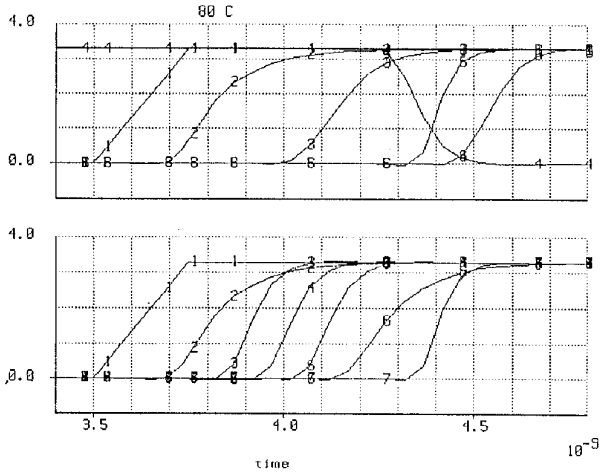


Figure 5: Critical path SPICE waveforms:

Top key: (1) operands, (2) I4 , (3) C16, (5) long carry select, (6) sum.

Bottom key: (1) operands, (2) I4, (6) short carry ripple, (7) long carry select.