## 24.2 A 240ps 64b Carry-Lookahead Adder in 90nm CMOS

Sean Kao, Radu Zlatanovici, Borivoje Nikolic

University of California, Berkeley, CA

Fast and energy-efficient single-cycle 64b addition is essential for today's high-performance microprocessor execution cores. The designer has several degrees of freedom to optimize the adder for performance and power. There is a choice of radix-2 or radix-4 trees, with full or sparse implementation of the conventional or Ling's carry-lookahead (CLA) equations as well as the circuit design style. In this paper, the impact of these design choices on the performance and power of the domino CMOS adder are analyzed and then confirmed using an optimization tool. An optimized topology, a sparse radix-4 Ling adder, is designed and fabricated in a general-purpose 90nm CMOS technology using standard $V_{TH}$ transistors. It performs single-cycle 64b additions in 240ps and consumes 260mW at a nominal supply voltage of 1V and room temperature.

In a high-performance integer execution unit, the microarchitecture sets the constraints for the adder design. The selected bit-slice height of 24 M1 tracks accommodates a wide-issue architecture. The bit-slice height and adder topology determine the length of the long 'prefix' wires in the carry-tree. In this study, the input capacitance per bit slice is limited to 27fF. The output loading capacitance of the adder equals its input capacitance, assuming that a buffer would be used to efficiently drive the local loopback bus. The carry tree (computing the final carry[63:0]) and the final sum-select are in the critical path while the sum-precompute is non-critical (Fig. 24.2.1). Using a circuit-optimization framework [1], representative 64b CLA adder design choices are evaluated under these design constraints. The optimization framework is configured to minimize the adder delay subject to a maximum energy constraint by tuning transistor sizes.

Ling's pseudo-carry equations [2] reduce transistor stack height in the first stage of the carry tree [3]; in turn, they increase the complexity of the sum pre-computation. Figure 24.2.2 shows the energy–delay tradeoff curves for conventional CLA adders and Ling adders implemented with full (Kogge-Stone) radix-2 and radix-4 carry trees. For low delay targets, Ling adders are about 5% faster because of the reduced transistor stack in the first stage. However, when energy constraints are tighter, more complex sum-precompute blocks appear in the critical path, making Ling adders less efficient. Figure 24.2.2 also shows that radix-4 adders are more efficient than radix-2 adders for this technology and chosen design constraints. Radix-4 adders have fewer stages in the critical path composed of more complex gates with more branching than radix-2. The radix-4 adders are closer to the optimal number of stages with low output loads [4].

Sparse trees have been explored recently [5,6] because of their potential energy savings. Sparse trees compute only every second or fourth carry signal and use a more complex sum pre-computation for the remaining bits. There are fewer "propagate" and "generate" gates in the tree, but with a higher fanout at the final carry. The input gates in the critical path can be upsized, reducing the effect of the internal wiring in the tree. As a tradeoff, there is a cost in the additional complexity and loading from the sum-precompute block. Figure 24.2.3 shows the energy-delay tradeoff curves for full and sparse domino Ling radix-2 and radix-4 trees. Because of their larger stage count, radix-2 adders benefit more from sparse trees. Sparseness of 4 reduces the delay by 9% or the energy consumption by 50%. Increasing the sparseness for radix-4 adders shows a more delicate balance of tradeoffs between increased critical-path gate sizes and more complex sum-precompute. The radix-4 tree with the sparseness of 2 has the lowest delay (7.3 FO4) with less than 20% energy reduction compared to the full radix-4 tree.

A single-rail domino radix-4 sparse-2 tree, implementing Ling's pseudo-carry equations in the critical path and a static CMOS sum-precompute block in the non-critical path, is implemented in 7M1P 90nm CMOS. The adder core consists of 6028 transistors whose sizing is based on the results of the optimization framework. The size of the adder core is 417µm × 75µm. Figure 24.2.7 shows a micrograph of a part of the chip containing two adder cores and the corresponding testing circuitry.

Figure 24.2.4 shows the block diagram of the adder, on-chip testing circuitry, and the corresponding timing waveforms. Delayed-precharge domino logic is used in the carry tree in order to hide the precharge phase from the overall cycle time. Most stages in the critical path use footless domino logic with stack node precharging, as shown in Fig. 24.2.5. Since the monotonicity of the global inputs a[63:0] and b[63:0] cannot be guaranteed, the first stage is implemented using footed domino logic. The inputs of the sum-select MUX, $S^0$[63:0], $S^1$[63:0], are outputs of a static block and non-monotonic, thus, psel must be a hard clock edge (Fig. 24.2.4). Critical timing edge arrivals can be fine tuned at runtime through the scan chain of the chip in order to ensure correct functionality and best performance.

The sparse-2 carry tree computes only even-order carries and each signal selects two sums (Fig. 24.2.5). The non-critical sum-precompute block has two types of paths: odd-order sums are precomputed as $a_i \oplus b_i$, and $(a_i \oplus b_i)'$; even-order sums are a function of the carry into the previous bit and are more complex. The layout of critical and non-critical paths are interleaved such that the more complex even-order sum-precompute gates fit exactly in the space freed by the eliminated carry gates of the sparse tree, resulting in a compact bit-sliced layout.

Measured results are presented in Fig. 24.2.6. At the nominal supply of 1V, the adder core runs at an average speed of 4.2GHz (240ps, approx. 7.7FO4) for the slowest input vector and consumes 260mW in the worst case, with 2.3mW of leakage at room temperature. The power measurements include the adder core and clock-generation circuitry, but not the on-chip testing circuitry. By increasing the supply voltage to 1.3V the delay decreases to 180ps with 606mW of active power and 4.9mW of leakage.

*Acknowledgments:*
This work was supported in part by NSF grant ECS-0238527, UC Micro and Intel Corp. The authors thank ST Microelectronics for test chip fabrication, Yong Yang for the help in the design of the test circuitry, Ryan Roberts for the board design.

*References:*
[1] R. Zlatanovici, B. Nikolic, "Power – Performance Optimization for Custom Digital Circuits," *Proc. PATMOS*, pp. 404-414, Sept., 2005.
[2] H. Ling, "High Speed Binary Adder," *IBM J. R&D*, vol. 25, no. 3, pp. 156-166, May, 1981.
[3] S. Naffziger, "A Sub-Nanosecond 0.5µm 64b Adder Design," *ISSCC Dig. Tech. Papers*, pp. 362-363, Feb., 1996.
[4] I. Sutherland, R. Sproull, H. Harris, "Logical Effort," Morgan Kauffman 1999.
[5] S. Mathew, et al, "A 4GHz 300mW 64b Integer Execution Unit ALU with Dual Supply Voltages in 90nm CMOS," *ISSCC Dig. Tech. Papers*, pp. 162-163, Feb., 2004.
[6] Y. Shimazaki, R. Zlatanovici, B. Nikolic, "A Shared-Well Dual-Supply Voltage 64-bit ALU," *ISSCC Dig. Tech. Papers*, pp.104-105, Feb., 2003.

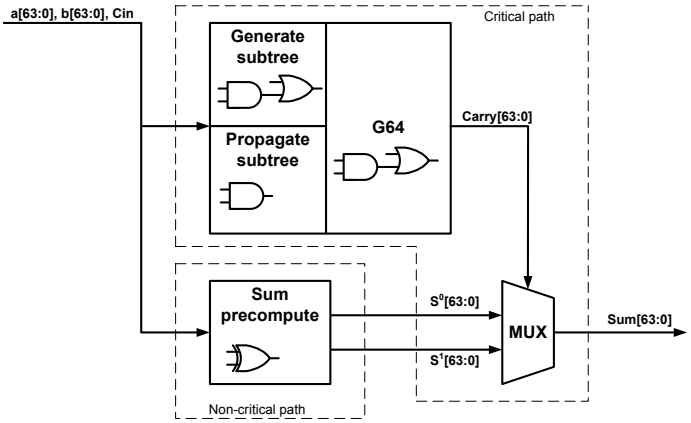



Figure 24.2.1: 64b carry-lookahead adder structure.

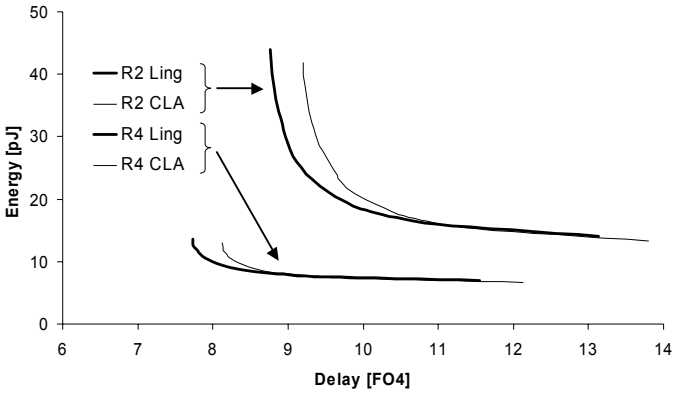



Figure 24.2.2: Ling versus CLA: Energy versus delay.

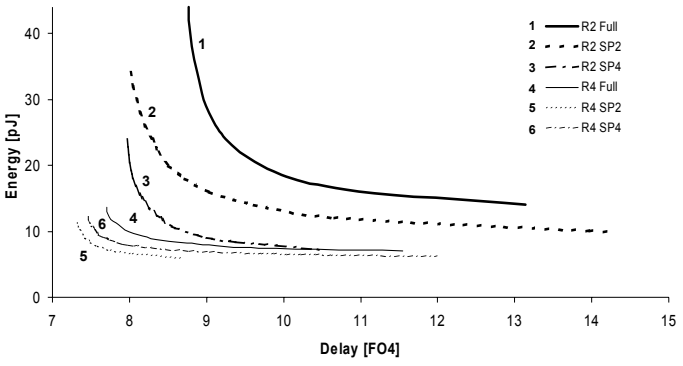



Figure 24.2.3: Full versus sparse trees: Radix-4 and radix-2 domino Ling adders in the energy–delay space.

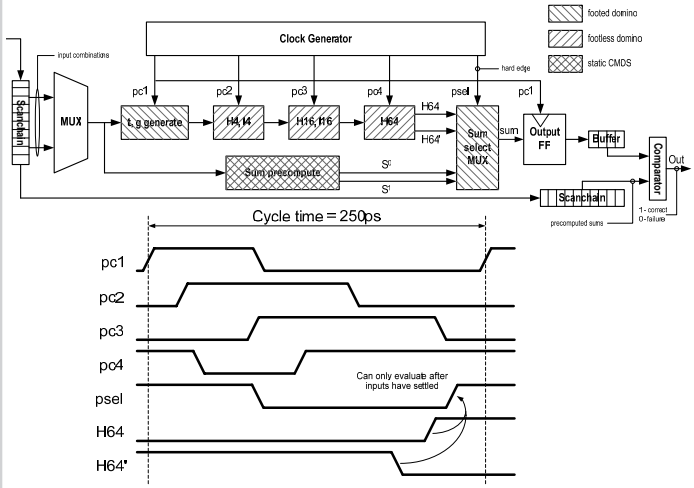



Figure 24.2.4: Adder and testing circuitry block diagram and timing waveforms.

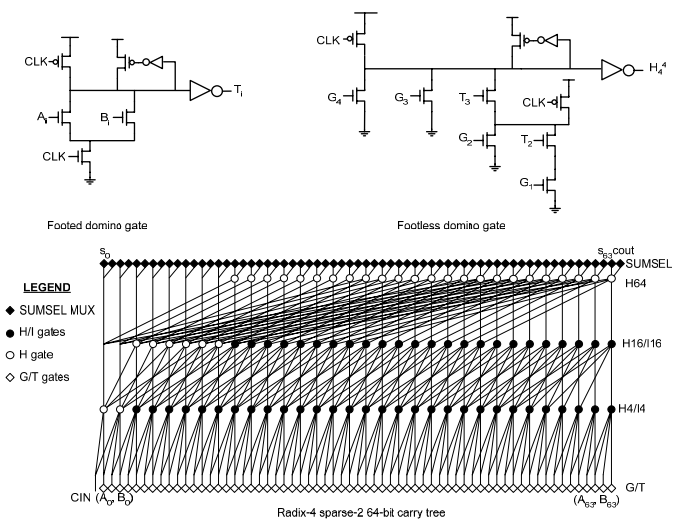



Figure 24.2.5: Adder circuit details and carry tree.

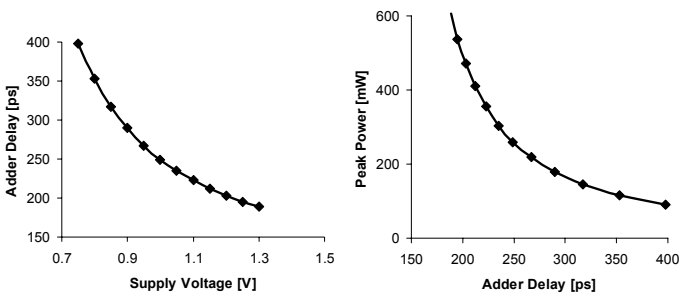



Figure 24.2.6: Measured results.

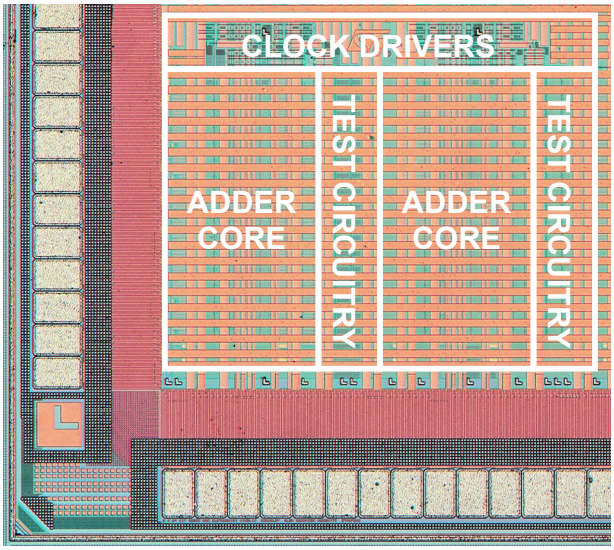



**Figure 24.2.7: Micrograph of two adder cores and test circuitry.**

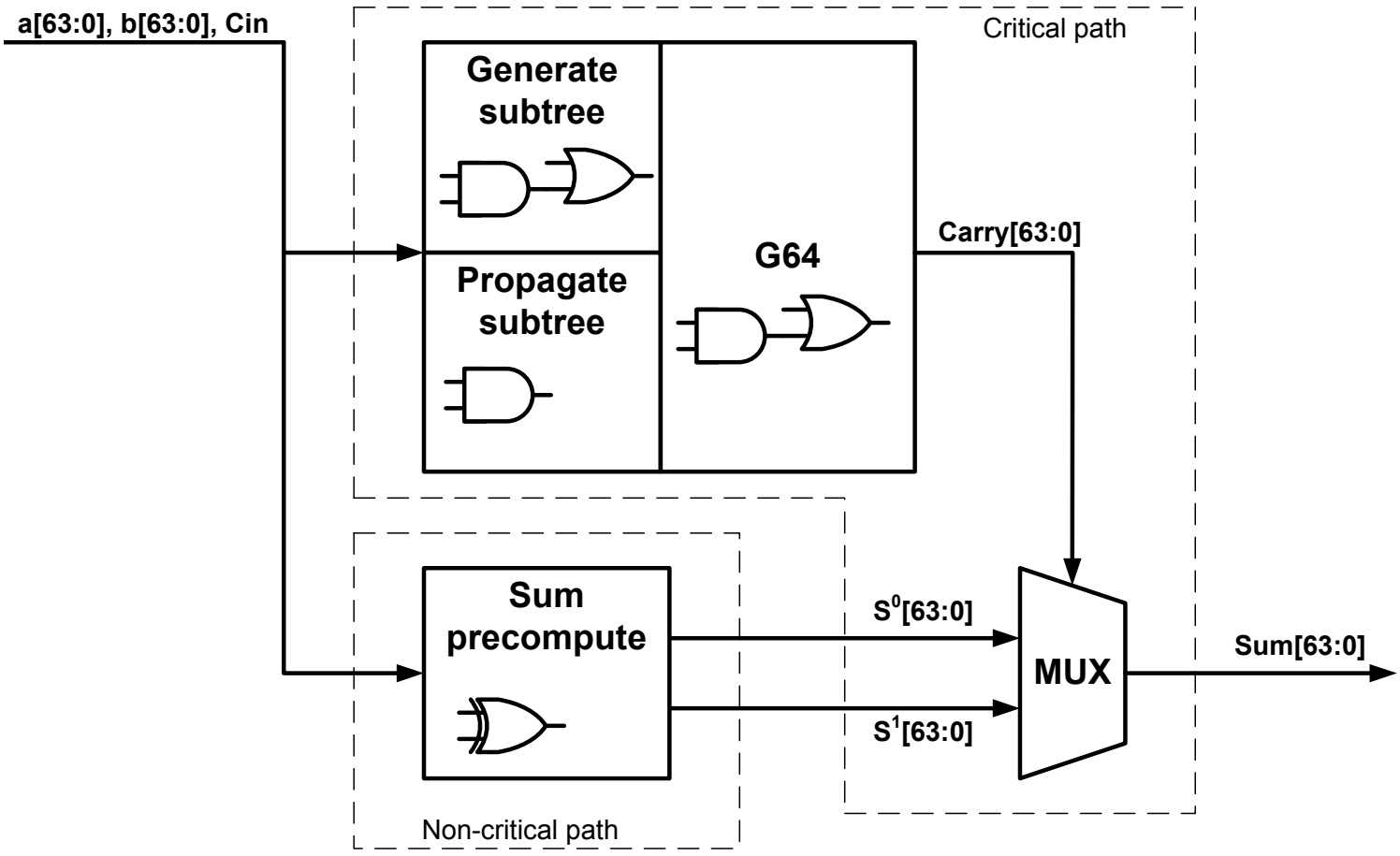



**Figure 24.2.1: 64b carry-lookahead adder structure.**

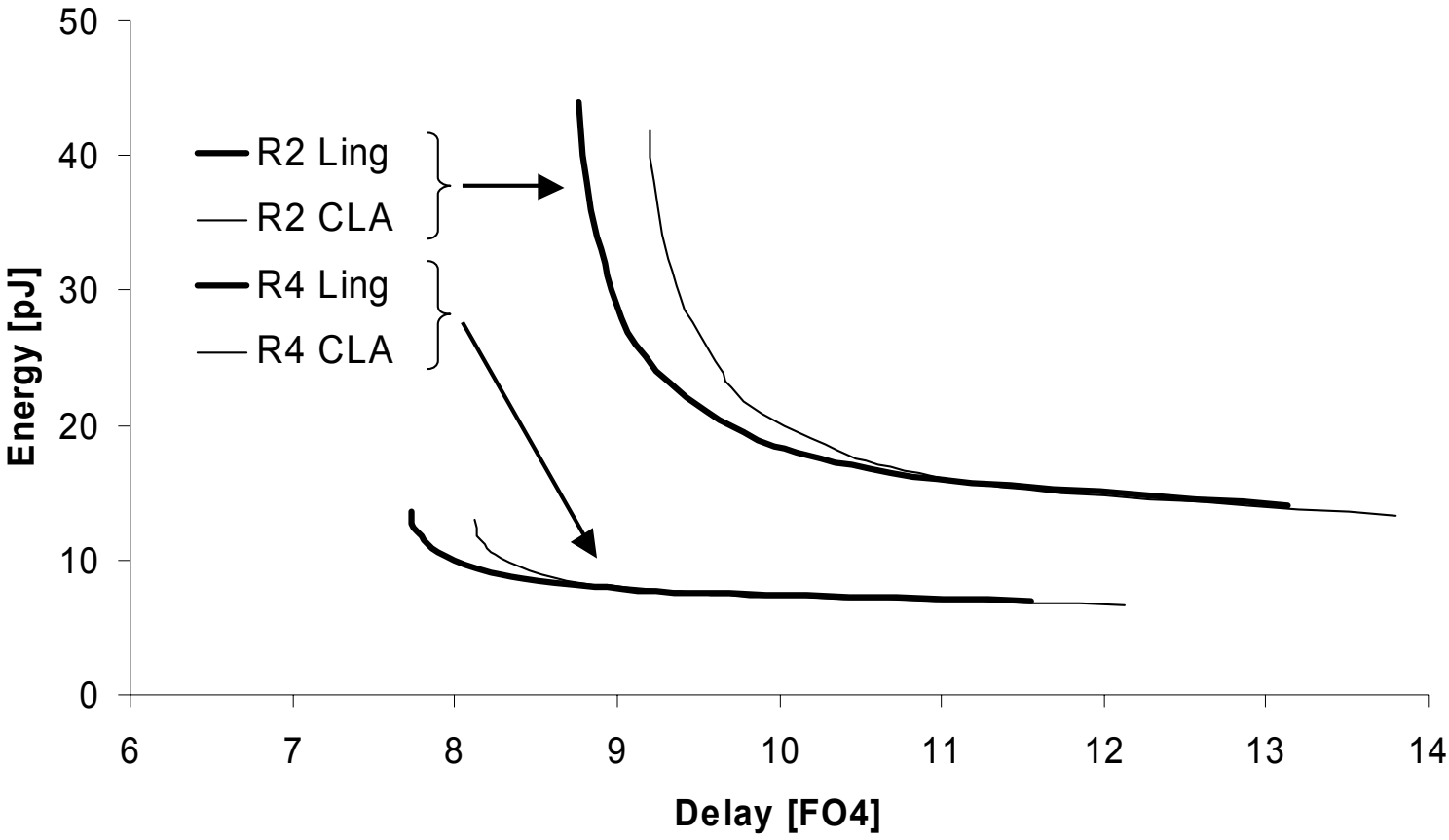



**Figure 24.2.2: Ling versus CLA: Energy versus delay.**

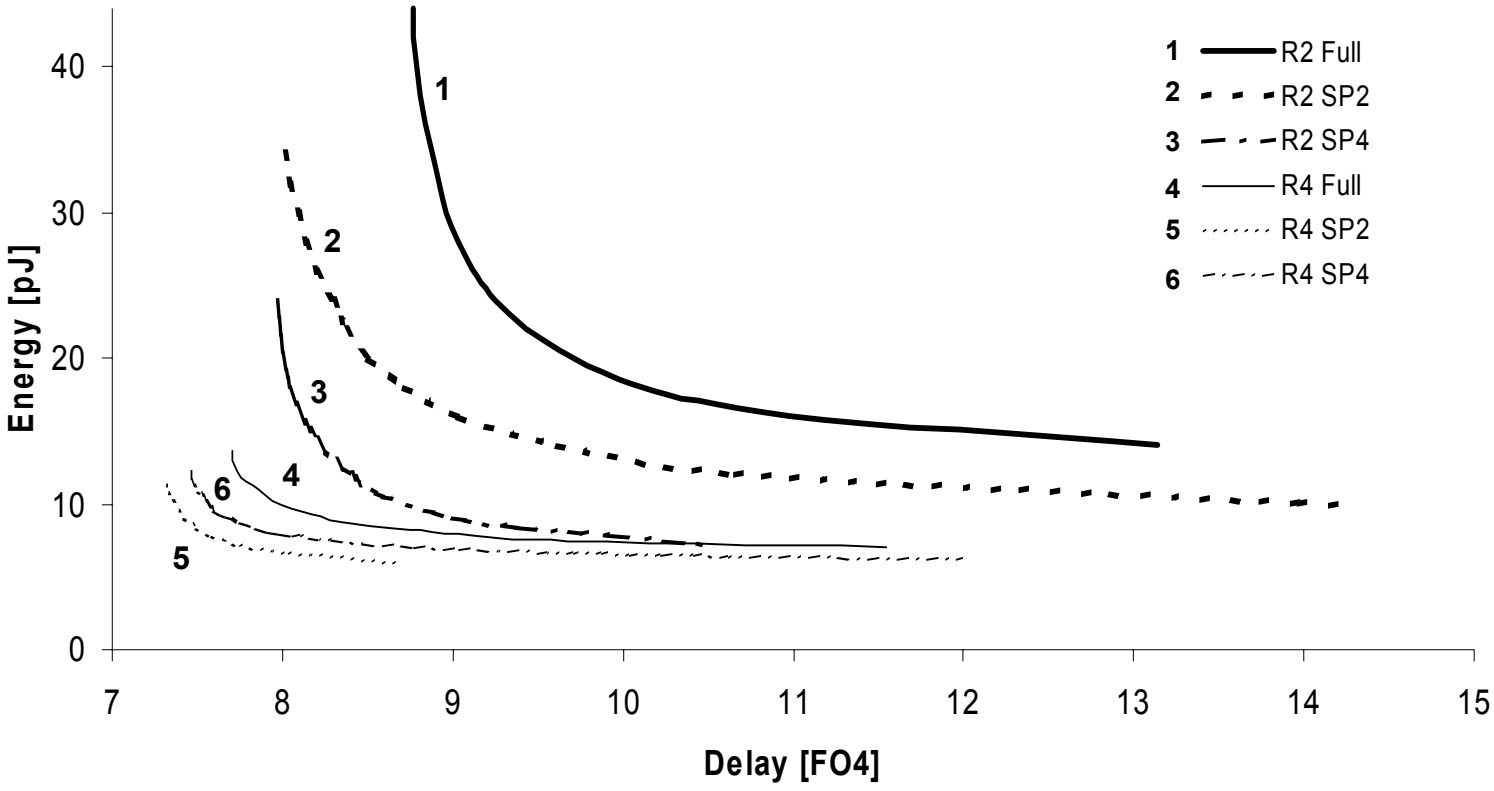



**Figure 24.2.3: Full versus sparse trees: Radix-4 and radix-2 domino Ling adders in the energy–delay space.**

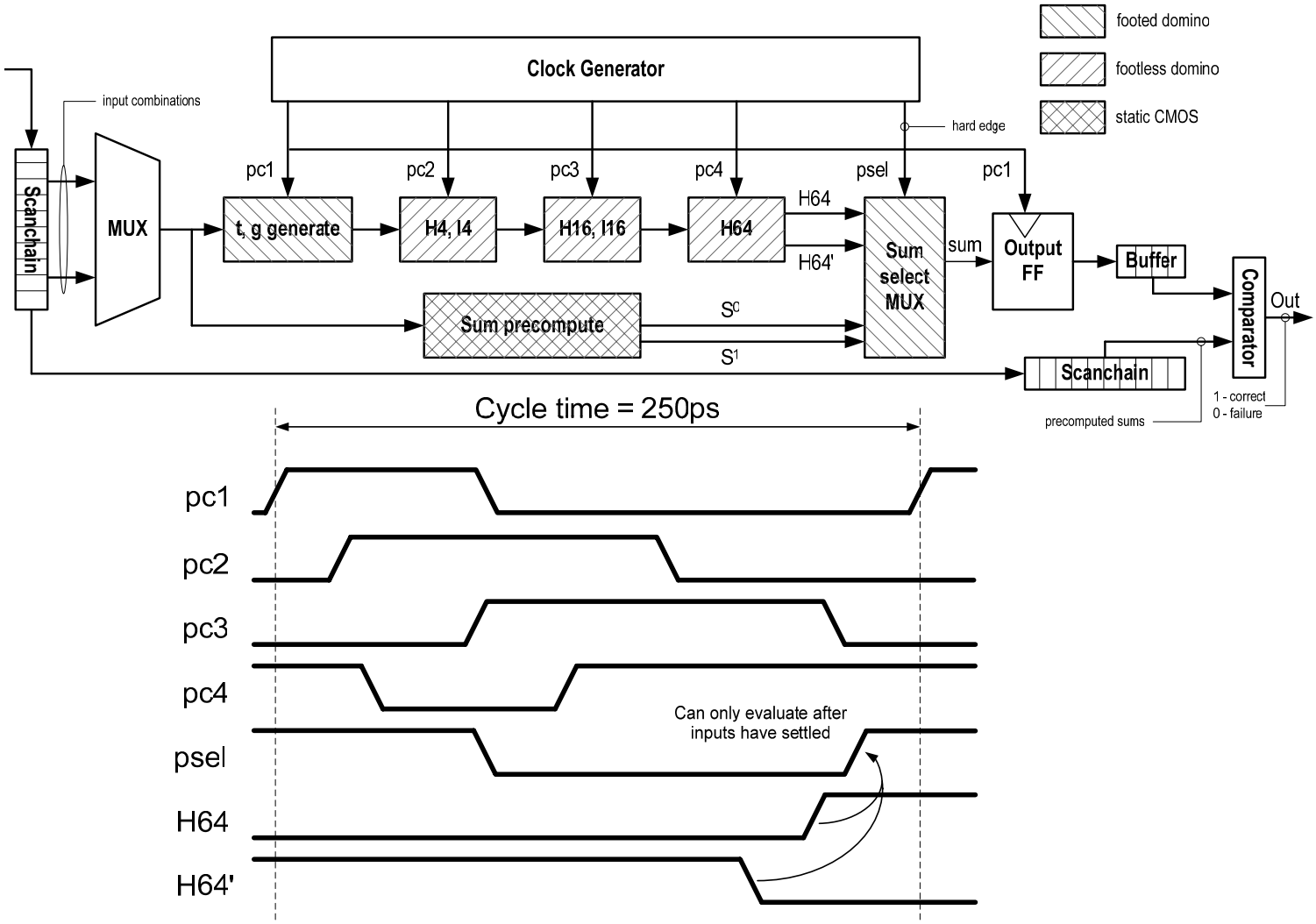
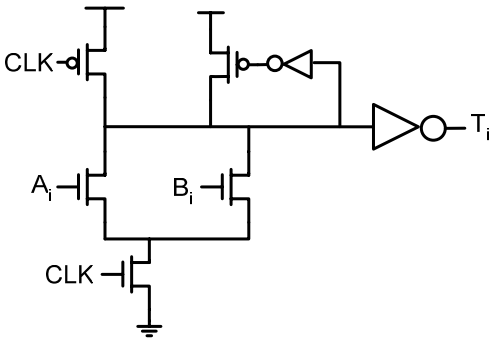
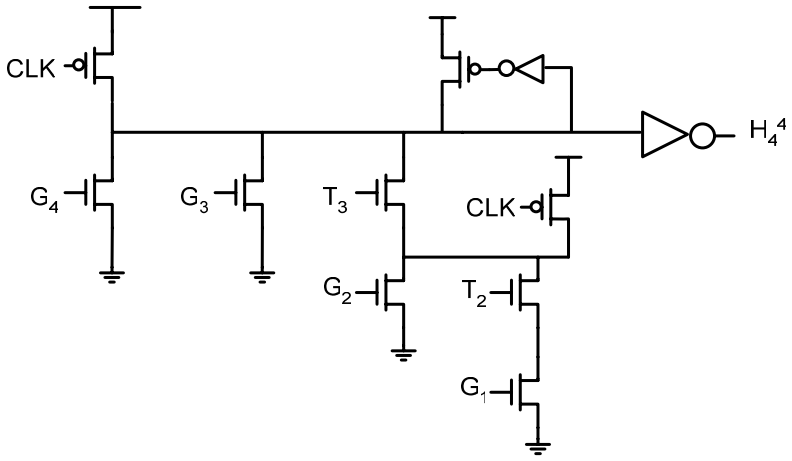


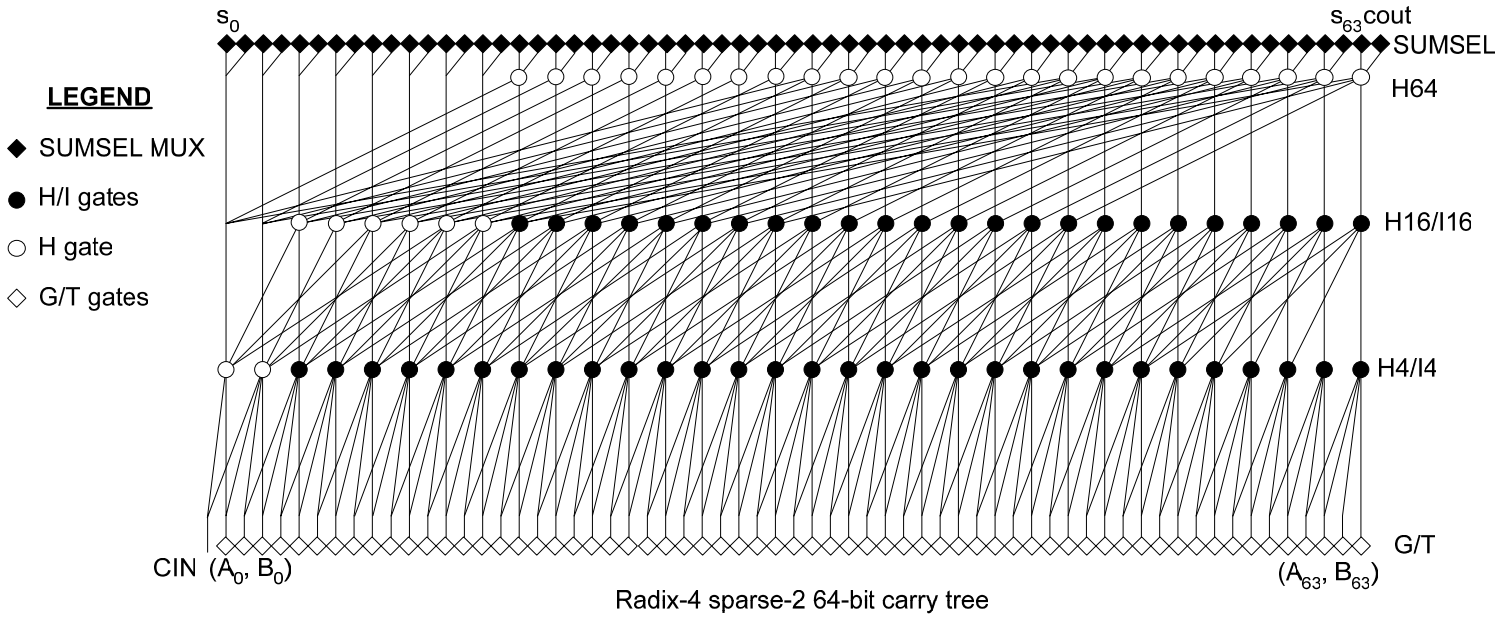

**Figure 24.2.4: Adder and testing circuitry block diagram and timing waveforms.**




Footed domino gate

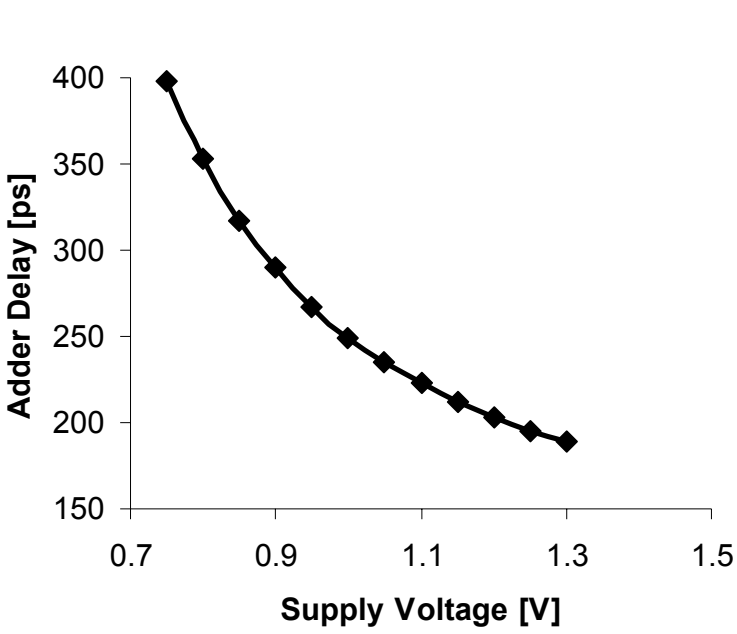



Footless domino gate




Radix-4 sparse-2 64-bit carry tree

**Figure 24.2.5: Adder circuit details and carry tree.**



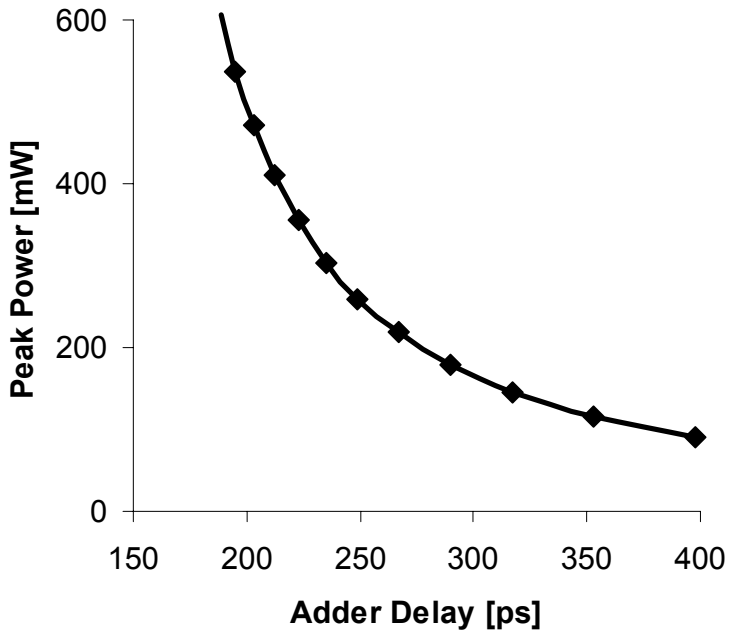




**Figure 24.2.6: Measured results.**

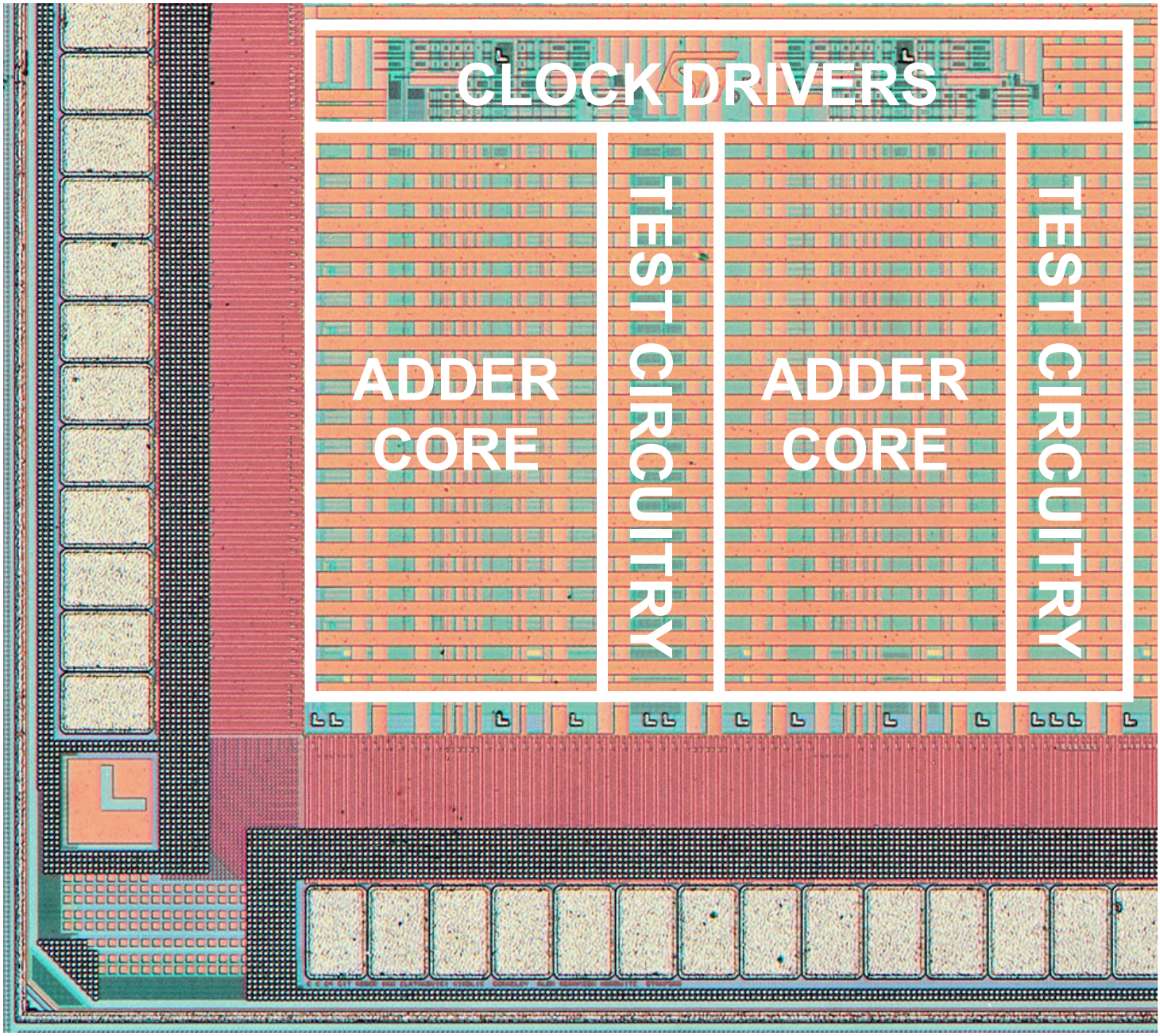



**Figure 24.2.7: Micrograph of two adder cores and test circuitry.**