

CMOS 16-bit Adder

1. Introduction & Review

This report presents the design and simulation of a CMOS 16-bit adder targeting an addition time of $t_{add} \leq 1$ ns across all process corners and temperatures. A ripple carry adder and a carry lookahead adder were designed and compared at the transistor level before selecting the final architecture.

Ripple Carry Adder:

The ripple carry adder is the simplest and most area efficient of the standard adder architectures. 16 full adder cells are cascaded so that the carry out of each stage drives the carry in of the next, and the carry ripples from the least significant bit to the most significant bit. Each FA cell is a 28-transistor static CMOS full adder implementing $Cout = AB + Ci$ ($A + B$) and $S = A \oplus B \oplus Ci$. The worst case delay grows linearly with word length as $t_{add} \approx (N-1) t_{carry} + t_{sum}$ because a carry generated at bit 0 must traverse all 15 stages. This linear delay is the fundamental limitation of the architecture, while in exchange the RCA offers the lowest transistor count, area, and power of the candidates. Functional simulation at TT and 27 °C confirms correct operation. With vector A (FFFF + 0001) the sum bits are overturned one by one as the carry arrives, producing a staircase of glitches that directly shows the rippling carry.

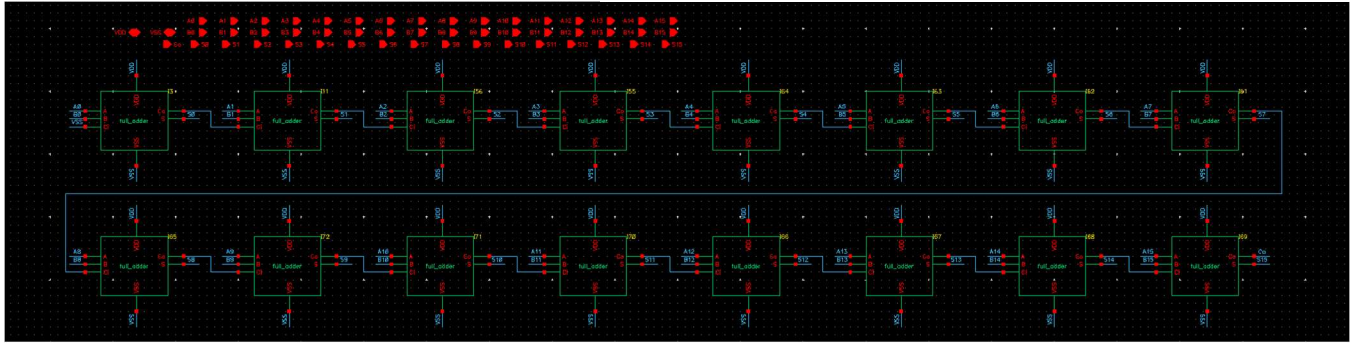
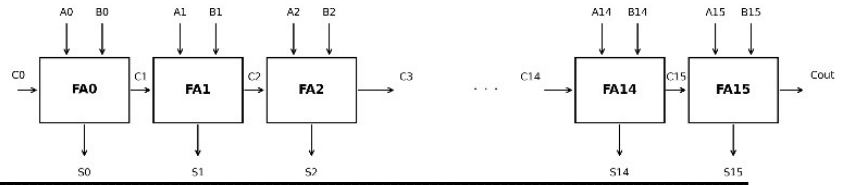


Fig. 1. Schematic of the 16-bit RCA

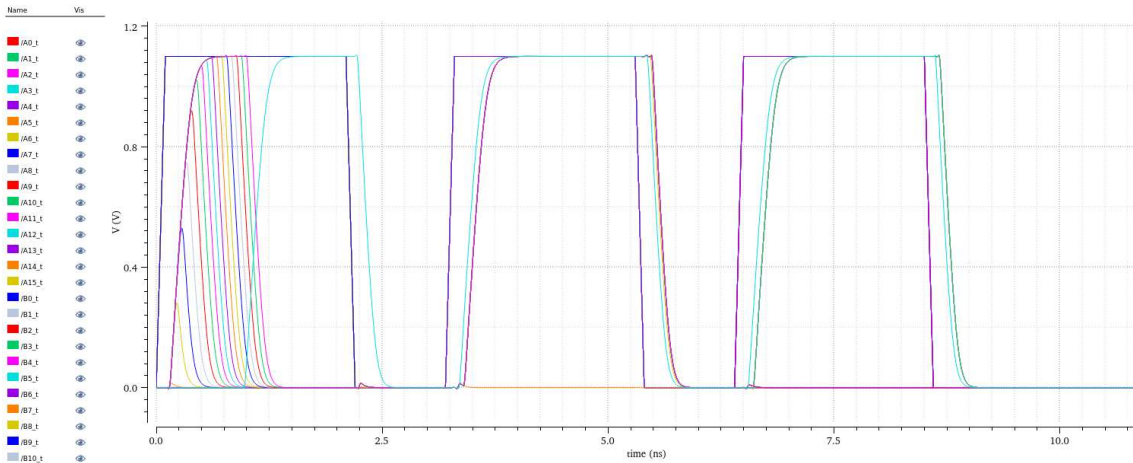


Fig. 2. Functional waveforms of vectors A, B, and C for RCA at TT, 27 °C (launched at 0, 3.3, and 6.5 ns)

Carry Lookahead Adder:

The carry lookahead adder improves on the ripple carry adder by computing all carries in parallel from generate and propagate terms, so a carry no longer has to walk through 16 sequential full-adder stages. This 16-bit design uses four 4-bit CLA blocks, each with a setup stage producing Pi/Gi signals and a group stage producing PGi/GGi signals, while an intermediate block supplies inter-block carries instead of rippling. Sum bits are formed from Pi and the resulting carries. The carry path is therefore a fixed number of gate levels rather than a chain of 16 stages, so the delay is far less sensitive to the input pattern than the RCA, though not entirely independent of it, as the all-propagate vector A remains the slowest of the three. This costs roughly twice the transistor count, 988 in static CMOS against 448 for the RCA.

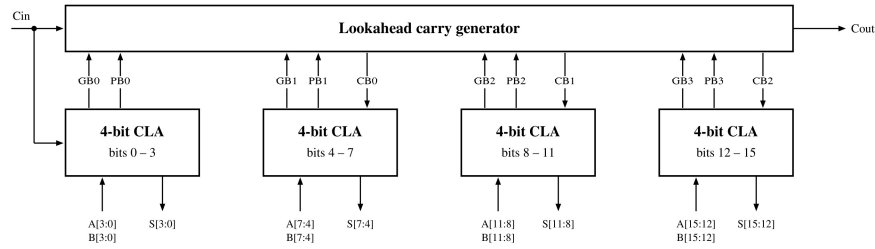


Fig. 3. Block diagram of the 16-bit carry lookahead adder

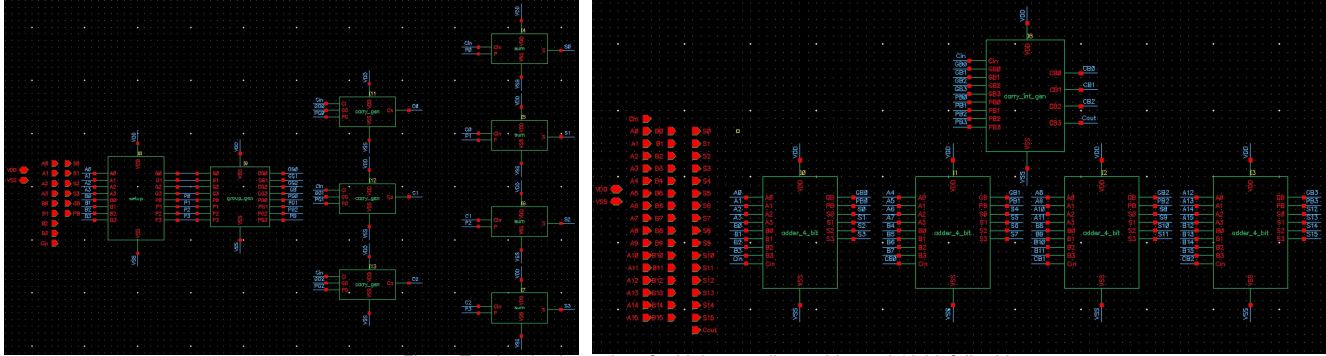


Fig. 4. Top level schematics of 4-bit intermediate adder and 16-bit full adder

2. Comparative analysis

Compares the two implementations at TT and 27 °C with vector A (worst-case carry propagation):

Architecture	Transistor Count	Worst Case Delay (ps)	Power (μ W)	PDP (fJ)
Ripple Carry	448	1008	123.1	124.1
Carry Lookahead	988	456	300.1	136.9

The two architectures sit at opposite ends of the delay and power trade-off. The RCA has the lowest transistor count and power, but its delay grows linearly with word length and is strongly input-dependent, reaching 1.01 ns for vector A at nominal conditions. It therefore already exceeds the 1 ns budget at nominal conditions, and any shift toward a slow corner or high temperature only widens the violation. The CLA reduces the carry path to a fixed number of gate levels, giving a shorter and nearly pattern-independent delay at the cost of roughly twice the transistors and higher power. The CLA is chosen because it is the only design that meets the timing specification across all corners and temperatures.

3. Simulation Results

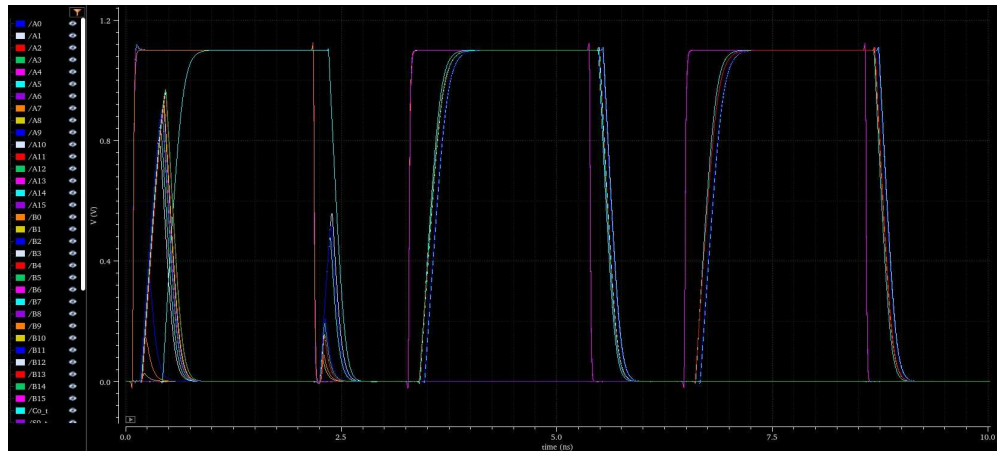


Fig. 5. Functional waveforms of vectors A, B, and C for CLA at TT, 27 °C (launched at 0, 3.3, and 6.5 ns)

Process corner and temperature analysis:

Vector	Corner	Temp (°C)	Delay (ps)	Power (μ W)	PDP (fJ)
A	FF	-25	312.7	295.5	92.4

Vector	Corner	Temp (°C)	Delay (ps)	Power (μW)	PDP (fJ)
A	TT	27	456.0	300.1	136.9
A	SS	85	676.5	301.2	203.8
B	FF	-25	216.9	372.7	80.8
B	TT	27	312.0	370.3	115.5
B	SS	85	459.2	362.5	166.5
C	FF	-25	214.4	199.4	42.7
C	TT	27	308.0	198.1	61.0
C	SS	85	453.3	191.2	86.7

The following results showcase the effect of process and temperature variations on consumed power, propagation delay, and the power-delay-product across all three test vectors. The TT_27°C case represents typical operating conditions and sets the baseline for comparison. The FF_-25°C case represents the best-case scenario, with propagation delay reduced by roughly 31-36% and power staying nearly unchanged from baseline, producing a substantial improvement in PDP. The SS_85°C case represents the worst-case scenario, with delay increasing by roughly 44-49% while power again stays close to baseline, producing the worst PDP across all vectors. Since power shifts by only a few percent across corners while delay shifts by 30-50%, PDP variation is driven almost entirely by propagation delay, indicating this measurement is dominated by dynamic switching power rather than the leakage effects typically associated with process/temperature corners.

Power Supply Variation:

The following plot illustrates how the power-delay-product varies with the power supply voltage. Power consumption increases quadratically with VDD, while the propagation delay is roughly proportional to $VDD/(VDD-V_t)^\alpha$. For extremely low VDD, the delay term dominates and pulls the PDP up as a result. After rising to the local maximum at VDD = 850mV, the PDP begins to drop, as the propagation delay is shrinking faster than power consumption is growing. This trend continues until the PDP finds its local minimum at around 1.2V (slightly higher than the nominal voltage), where the growth rate of the power consumption starts to outpace the propagation delay's decay rate. The PDP begins to increase after this point, approaching a roughly quadratic relationship with VDD as the propagation delay approaches a constant value.

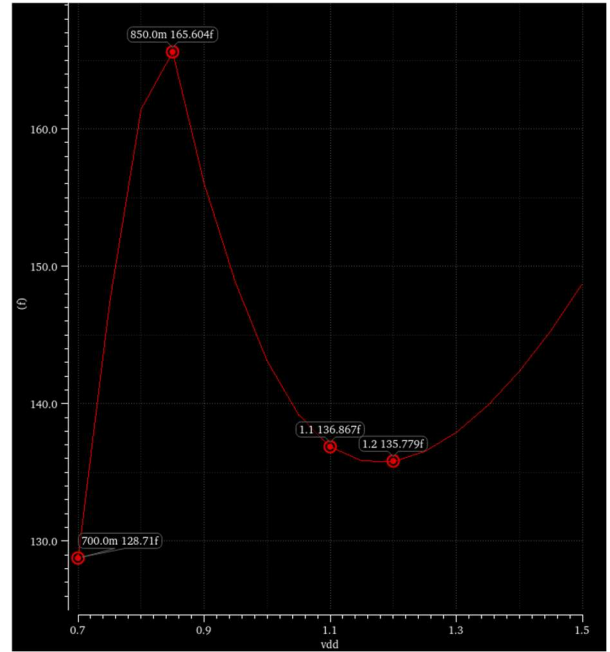


Fig. 6. PDP versus supply voltage for vector A at TT, 27 °C

4. Conclusion

Two 16-bit adders were designed at the transistor level in static CMOS and compared under identical testbench conditions. The ripple carry adder is the smaller and lower power design, but its carry path grows linearly with word length and it already exceeds the 1 ns budget at nominal conditions, before any corner shift is applied. The carry lookahead adder computes all carries in parallel and more than halves the delay for roughly twice the transistor count, at only about 10 % higher PDP, so it was chosen. The chosen design adds correctly for all three test vectors and meets timing at every corner, with a worst case of 676.5 ps at SS and 85 °C. Average power is almost corner independent, since switching energy is set by CV^2 , so PDP tracks delay and SS at 85 °C is the worst corner on both counts. Power instead varies strongly between vectors, since it follows the total switching activity including glitching on the internal carry and sum nodes, rather than the process corner. Vector A draws more than vector C despite producing an all-zero sum, because the upper 15 sum bits glitch. PDP passes through a local minimum of 135.8 fJ near 1.2 V, above which the quadratic growth of switching energy overtakes the shrinking delay, leaving the nominal 1.1 V point within 1 % of that minimum. At lower supply the addition time increases sharply as the devices approach threshold, so the 1 ns budget rather than the energy sets the practical lower limit on supply voltage.