Design and Optimization of a 32-Bit Carry-Bypass Adder

Mitchell Hoffman, Matthew Nutt

***Abstract*—This document outlines the design flow of developing a 32-bit carry-bypass adder. Calculations, schematics, waveforms, and optimization reasoning are included. This is the report of our final project for the ELEC 423 course at Rice University, taught by Professor Kaiyuan Yang.**

Cādence Virtuoso was used for all circuit schematics and simulations.

For our optimizations, the unit capacitance of a minimum-sized NMOS is $C_U = 0.052\,\text{fF}$, and the capacitance of the input of a minimum-sized inverter is $C_{\text{inv}} = 0.13\,\text{fF}$.

I. Introduction

Arithmetic units are some of the most fundamental building blocks of all computing systems. The ability to quickly and accurately calculate mathematical results is critical for efficient algorithms. Adders are perhaps the most essential of such units, so extensive consideration is put into their design and optimization.

Usually, the primary limitation on the delay of an adder is the path of the carry bits. If the result of addition on a certain digit results in a carry bit, that carry bit must be considered in the next digit's result, and so on, all the way to the final bit and its carry bit. The simplest design that satisfies this requirement is the ripple-carry adder, which chains a one-bit adder's carry-out bit to the next one-bit adder's carry-in bit. However, this results in a slow carry path, as a carry from the lowest digit that propagates to the highest digit must first "ripple" through all adders in the chain.

The carry-bypass adder accounts for this suboptimal design by allowing a carry bit to "bypass" sections of the adder chain. If addition on a number of consecutive digits - a section - creates "propagate" signals for each digit (defined by the PGK formulation of binary addition), then the carry-in bit of the section will end up propagating through the adders; therefore, it can be prematurely passed on to the next section. This is accomplished with the augmentation of combinational logic to calculate the bypass signal, and a multiplexer that either passes the carry-out of the final adder or the carry-in of the section to the next section, depending on the value of the bypass signal.

Beyond the circuit topology, careful consideration must be taken in optimizing the size of transistors and gates to balance the current they provide with their inherent parasitic capacitance. In this design, this optimization is done via the "logic effort" method.

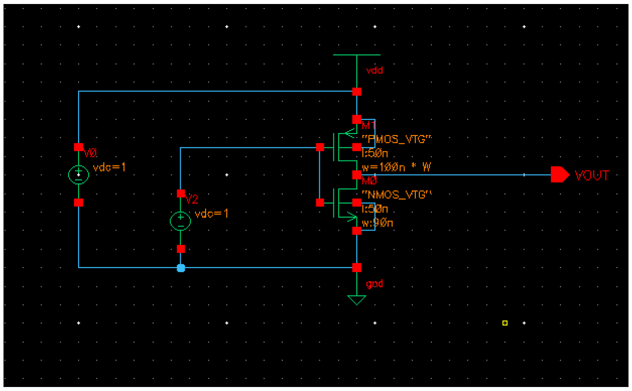
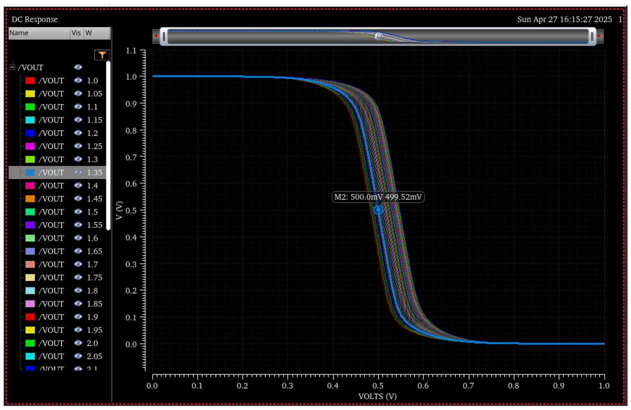


Fig. 1. Circuit schematic of the P/N ratio testbench for a minimum-sized inverter.



Name	Vis	W
/VOUT	👁	
🟥 /VOUT	👁	1.0
🟨 /VOUT	👁	1.05
🟩 /VOUT	👁	1.1
🟦 /VOUT	👁	1.15
🟦 /VOUT	👁	1.2
🟪 /VOUT	👁	1.25
🟩 /VOUT	👁	1.3
🟦 /VOUT	👁	1.35
🟥 /VOUT	👁	1.4
🟧 /VOUT	👁	1.45
🟩 /VOUT	👁	1.5
🟪 /VOUT	👁	1.55
🟩 /VOUT	👁	1.6
🟦 /VOUT	👁	1.65
🟥 /VOUT	👁	1.7
🟨 /VOUT	👁	1.75
🟦 /VOUT	👁	1.8
🟪 /VOUT	👁	1.85
🟥 /VOUT	👁	1.9
🟨 /VOUT	👁	1.95
🟩 /VOUT	👁	2.0
🟦 /VOUT	👁	2.05
🟦 /VOUT	👁	2.1

Fig. 2. Simulation results for the VTC of the minimum-sized inverter sweeping over the width of the PMOS.

II. P/N Ratio of Minimum-Sized Inverter

The first consideration in optimizing the carry-bypass adder was finding the appropriate ratio for PMOS and NMOS sizing in a minimum-sized inverter. An "appropriate" ratio is one such that the voltage transfer curve (VTC) is symmetric, which is synonymous with equivalent current flow between the pull-up and pull-down networks. For the adder, we used the `PMOS_VTG` and `NMOS_VTG` cells from the provided `NCSU_Devices_FreePDK45` library, and VDD was set to $1\,\text{V}$. We created a testbench schematic (as seen in Fig. 1) with an inverter constructed of a minimum-sized NMOS (W: $90\,\text{nm}$, L: $50\,\text{nm}$) and minimum-sized PMOS (W: $100\,\text{nm}$, L: $50\,\text{nm}$) whose input was driven with a DC voltage source from the `analogLib` library.

We performed a DC analysis sweeping over the width of the PMOS to observe the changes in the VTC (as seen in Fig. 2). The VTC was most symmetrical with a P/N ratio of 1.5 (PMOS W: $135\,\text{nm}$, NMOS W: $90\,\text{nm}$).

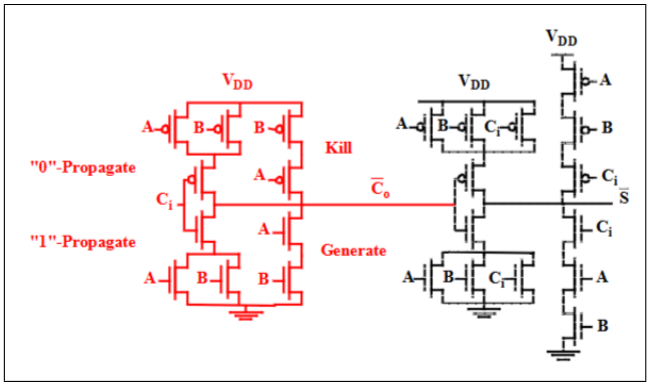


Fig. 3. Circuit diagram of a one-bit mirror adder, copied from Dr. Yang's lecture slides.

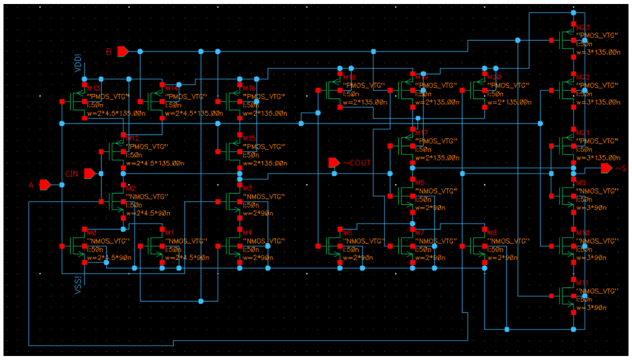


Fig. 4. Circuit schematic of the mirror adder, sized according to worst-case delay and optimal carry path stage effort.

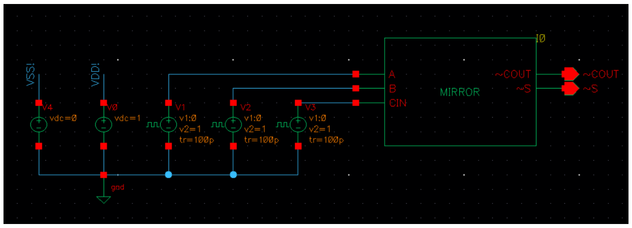


Fig. 5. Circuit schematic of the mirror adder testbench.

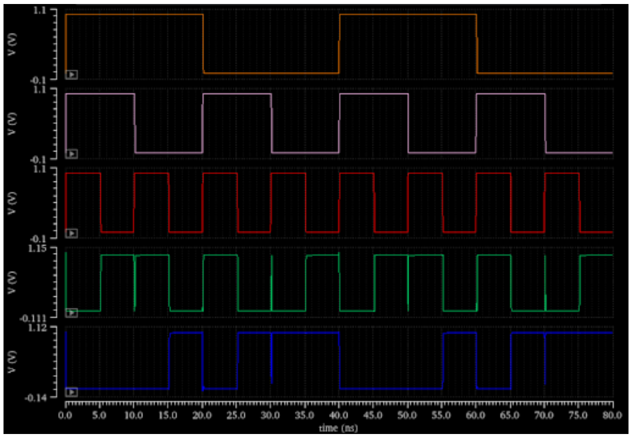


Fig. 6. Simulation results for the mirror adder's inputs (three waveforms above) and outputs (two waveforms below).

III. 1-Bit Mirror Adder

After determining the ideal P/N ratio for a symmetrical VTC, the rest of the transistor-level logic units can be designed with the ratio in mind. The first of these is the core of the design, a 1-bit mirror adder.

A ”mirror adder” is a modified version of the traditional full adder: By designing an adder that leverages the inversion property of adders, the carry path can be easily reduced, independent of the adder-chain topology. A diagram for the mirror adder can be seen in Fig. 3.

The sizing of the transistors in a mirror adder must be done with consideration for the carry path; the carry-out bit of an adder will be passed to the carry-in bit of the subsequent - identical(!) - adder. Let the sizing of the transistors on the carry-in path be variable, and to maintain consistent worst-case delay, also the transistors in their pull-up/pull-down units. Then the sizes of the NMOS transistors is $2x$, and the size of the PMOS transistors is $3x$, where ”size” is with respect to the minimum-sized NMOS (W: 90nm), and x is a size multiplier. With logic effort g, fanout f, and stage effort $h = gf$, the logic effort of the carry-in bit is $g = \frac{C_{\text{in}}}{C_{\text{inv}}} = \frac{5x}{2.5x} = 2$. For the conventionally-optimal stage effort $h = 4$, the necessary fanout is therefore $f = 2$. $f = \frac{C_{\text{load}}}{C_{\text{in}}}$, so the final requirement is that $C_{\text{load}} = 2C_{\text{in}}$. This is expressed with further detail in (1), and after sizing the rest of the transistors with worst-case delay in mind, with numerical values in (2).

$$C_{\text{diff}} + C_{\text{next}} = 2C_{\text{in}} \tag{1}$$

$$(5+5) + (5x + 5 + 7.5) = 2 \cdot 5x \tag{2}$$

Therefore, $x = 4.5$, and the final sizing of the variable transistors can be determined. This sizing is implemented in our final circuit schematic, as seen in Fig. 4.

We verified the correctness of this design with a simple testbench, whose schematic can be seen in Fig. 5. The resulting waveforms (as seen in Fig. 6) indicates that the adder behaves as expected.

IV. Miscellaneous Logic Cells

A. Transmission Gate

Our multiplexer design relies on the use of two transmission gates. We created our own schematics and symbols for transmission gates of various sizes (x1, x2, x4, x8, and x16). For the minimum-sized transmission gate, we used a minimum-sized PMOS and NMOS - each with W: 90 nm and L: 50 nm - resulting in a P/N ratio of 1, as seen in Fig. 7.

B. Multiplexer

For our multiplexers, we elected to use a design with transmission gates to minimize transistor count and delay, as seen in Fig. 8. Since the multiplexers are on the critical path for the carry-bypass adder, we created a testbench file consisting of two mirror adders with a multiplexer between them to measure the delay from the C_{in} of one mirror adder to the C_{out} of the other, as seen in Fig. 9. We found that implementing the multiplexer using x8 transmission gates resulted in the minimum delay, 38.12 ps, as seen in Table I and Fig. 10.

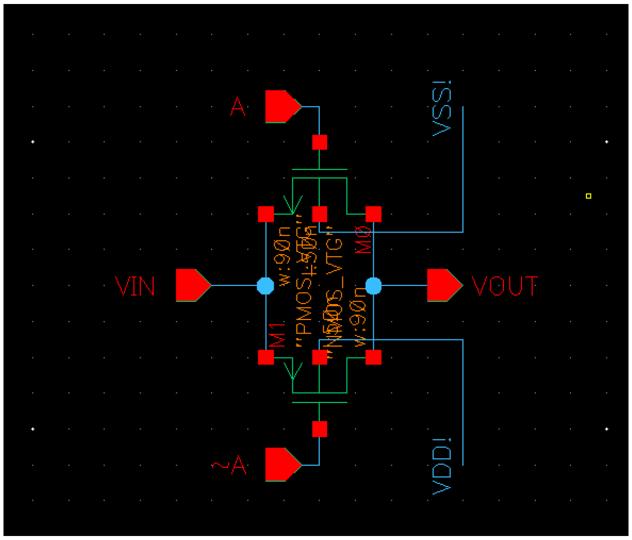


Fig. 7. Circuit schematic of a minimum-sized transmission gate.

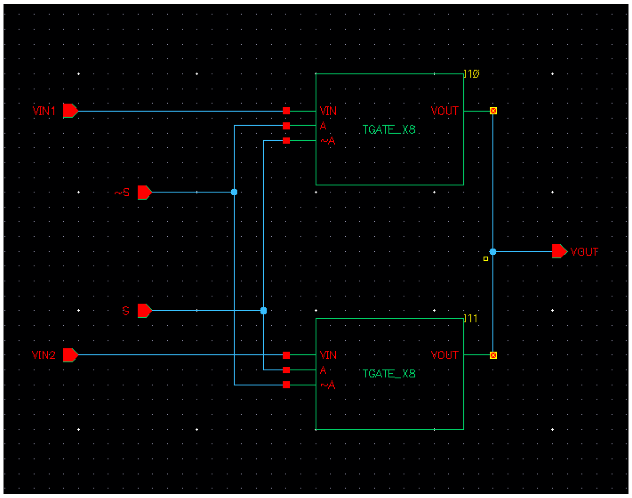


Fig. 8. Circuit schematic of the final multiplexer design.

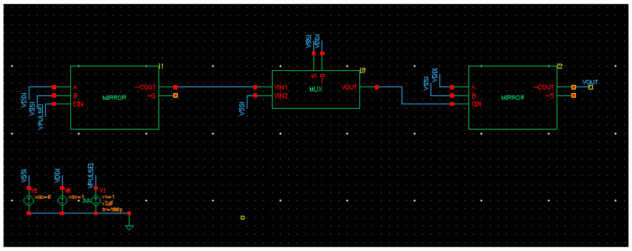
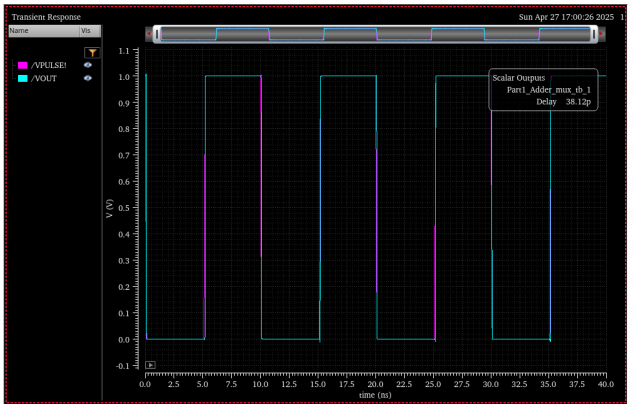


Fig. 9. Circuit schematic of the transmission gate testbench for sizing to minimize delay across the multiplexer.

TABLE I
MULTIPLEXER DELAY IN RELATION TO TRANSMISSION GATE SIZING.

Transmission Gate Sizing	Delay (in ps)
x1	55.39
x2	43.74
x4	39.46
x8	38.12
x16	42.42


Transient Response
Sun Apr 27 17:00:26 2025 1

Scalar Outputs
Part1_Adder_mux_tb_1
Delay 38.12p

Fig. 10. Simulation results of the multiplexer testbench using x8 transmission gates.

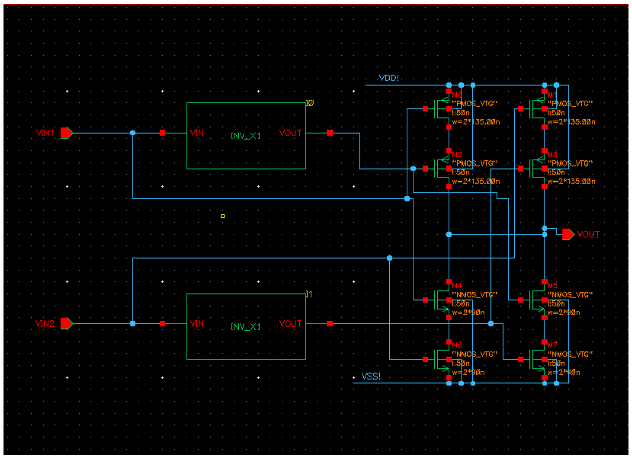


Fig. 11. Circuit schematic of a minimum-sized XOR gate.

C. XOR Gate

We used XOR gates to create the propagate signals for each bit of the inputs. We sized the transistors such that, in the worst case, the pull-up and pull-down networks of the XOR gate drive the same current as a minimum-sized inverter. In this case, each PMOS has a width of $2 \times 135\,\mathrm{nm} = 270\,\mathrm{nm}$ and each NMOS has a width of $2 \times 90\,\mathrm{nm} = 180\,\mathrm{nm}$. The schematic can be seen in Fig. 11.

D. 4-Input NAND Gate

We used a 4-input NAND gate in conjunction with an inverter to create the bypass signal from the propagate results. We sized the transistors such that, in the worst case, the pull-up and pull-down networks of the NAND-4 gate drive the same current as a minimum-sized inverter. In this case, each PMOS has a width of $1 \times 135\,\mathrm{nm} = 135\,\mathrm{nm}$ and each NMOS has a width of $4 \times 90\,\mathrm{nm} = 360\,\mathrm{nm}$. The schematic can be seen in Fig. 12.

V. 4-BIT ADDER STAGE

For the 32-bit carry-bypass adder, a longer stage length reduces the probability that the bypass logic is put to use, as well as necessitating exponentially larger logic to determine

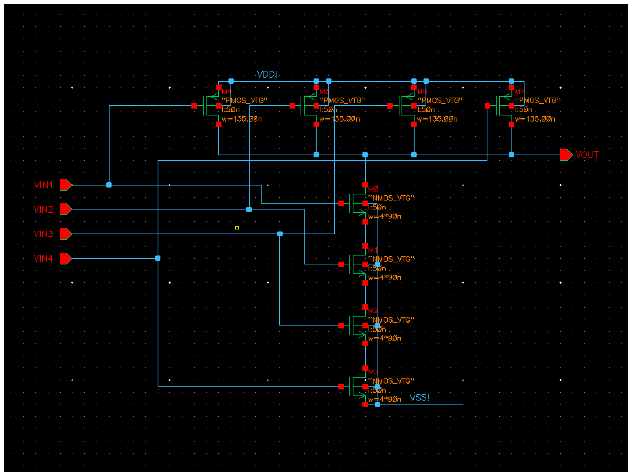


Fig. 12. Circuit schematic of a minimum-sized NAND-4 gate.

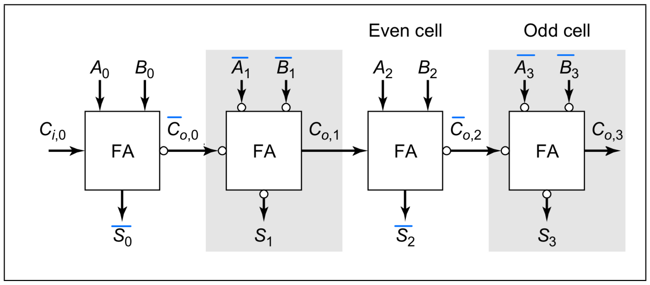


Fig. 13. Block diagram illustrating the inversion property in a chain of mirror adders, copied from Dr. Yang's lecture slides.

the bypass signal. In contrast, a shorter stage length encumbers the carry path with an excessive number of multiplexers. We hope to minimize both of these effects with a moderate stage length.

We opted for a stage size of 4 bits, meaning that every four bits of the adder chain is equipped with the logic to bypass the carry path of its constituent mirror adders. Additionally, we required that the number of bits in each stage was equal, to avoid redesigning the stages for multiple stage lengths.

Since mirror adders exploit the inversion property to remove redundant inverters on the critical path, the A and B input bits of odd cells in our 4-bit adder need to be inverted, and the output sum bits of even cells in our 4-bit adder need to be inverted (as seen in Fig. 13).

A. *Input Inverters*

In order to size the inverters driving $\overline{\text{A}}$ and $\overline{\text{B}}$ on the odd cells, we first look at the gate capacitance of the mirror adder for these inputs: $40C_U$. This means that the input inverters need to drive a load capacitance $C_{\text{load}} = 40C_U$. In order to achieve a stage effort $h = 4$, the inverters should have an input capacitance $C_{\text{in}} = 10C_U$ such that $H = \frac{40C_U}{10C_U} = 4$. This means that the inverters should be scaled by $\frac{10C_U}{2.5C_U} = 4$. The input inverters can be seen in Fig. 14.

B. *Output Inverters*

In order to identify the appropriate sizing for the inverters that drive the sum bit outputs, we start with the fixed input capacitance of the mirror adder's sum bits, $12.5C_U$, and the fixed load capacitance of the sum bit lines, $10\,\text{fF} \approx 192.3C_U$. Then the path fanout is $F = \frac{C_{\text{load}}}{C_{\text{in}}} = \frac{192.3C_U}{12.5C_U} \approx 15.4$. The logic effort of the mirror adder through the sum bit path is $g = \frac{C_{\text{in}}}{C_{\text{inv}}} = \frac{12.5C_U}{2.5C_U} = 5$. Since the rest of the path will consist of inverters, the path effort is therefore $H = GF \approx 76.9$.

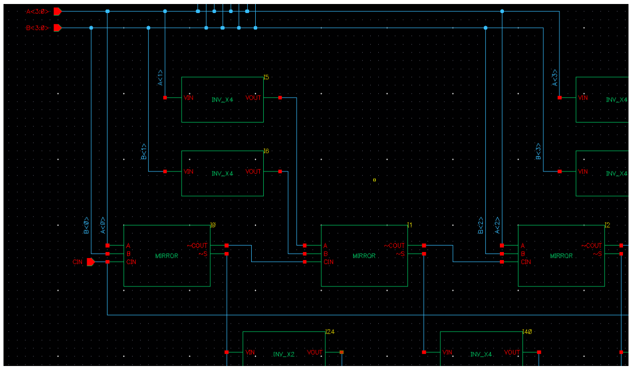


Fig. 14. Section of the 4-bit adder circuit schematic showing the input inverters.

The optimal number of stages, assuming an optimal stage effort of 4, is $\log_4 H \approx 3$. This includes the mirror adder, so ideally, two inverters would be added. However, this only works for the non-inverting outputs; for the even cells, which require inverting, three inverters will be used.

1) *Odd cells:* For the odd cells - those with two inverters - the optimal stage effort is $\sqrt[3]{H} \approx 4.25$; we round to 4. The stage effort of the first stage is $gf = 5f = 4$, so $f = \frac{4}{5} = \frac{C_{\text{load}}}{C_{\text{in}}}$, and $C_{\text{load}} = 10C_U$. Then the optimal inverter sizing for the first stage is x4.

The logic effort of inverters is always 1, so for a second stage with the same stage effort 4, a simple size step-up of ×4 can be used, resulting in an inverter size of x16.

2) *Even cells:* For the even cells - those with three inverters - the optimal stage effort is $\sqrt[4]{H} \approx 2.96$; we round to 3. In the same manner as before, $f = \frac{3}{5}$, and $C_{\text{load}} = 7.5C_U$. Then the optimal inverter sizing for the first stage is x3.

For a second and third stage with the same stage effort 3, a simple size step-up of ×3 can be used each time, resulting in inverter sizes of x9 and x27. These odd sizes are achieved with the use of multiple standard inverter cells in parallel. The output inverters can be seen in Fig. 15.

C. *Propagate Signal*

Each of the multiplexers in the carry-bypass adder is controlled by the bypass signal $P_0P_1P_2P_3$, where $P_i = A_i \oplus B_i$ represents the propagate signal for bit i. The path for producing these control signals consists of a 2-input XOR gate, a 4-input NAND gate, and an inverter driving the multiplexer. The switching signal for our multiplexers goes to the gates of a 90 nm NMOS and a 90 nm PMOS. Therefore, $C_{\text{load}} = 16C_U$ for the path. The first stage is a size 1 XOR gate, so $C_{\text{in}} = 7.5C_U$ and $F = \frac{16C_U}{7.5C_U} = 2.13$. The XOR gate has a logic effort of $\frac{7.5}{2.5} = 3$ and the NAND-4 has a logic effort of $\frac{5.5}{2.5} = 2.2$, so $G = 3 \times 2.2 = 6.6$ and $H = GF = 14.08$.

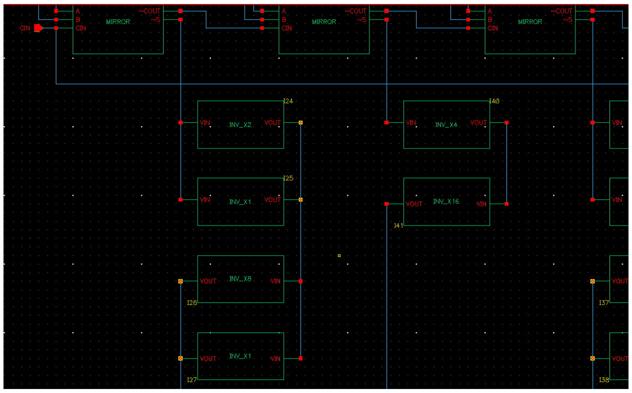


Fig. 15. Section of the 4-bit adder circuit schematic showing (a portion of) the output inverters.

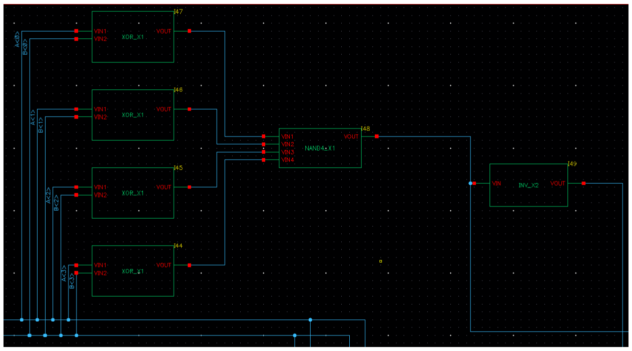


Fig. 16. Section of the 4-bit adder circuit schematic showing the propagate signal path.

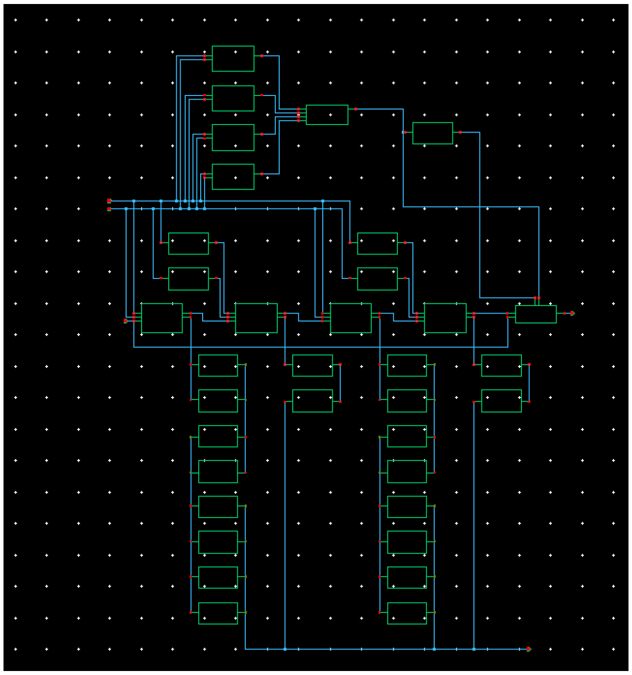


Fig. 17. Full circuit schematic for each 4-bit adder stage.

TABLE II
SUMMARY OF FINAL MEASUREMENTS

Metric	Value
$T_{p,\text{critical}}$	639.4 ps
I_avg	132.7 µA
P	132.7 µW
PDP	8.485×10^{-14} W · s
EDP	5.425×10^{-23} W · s^2

Since $\log_4 H = 1.91$, it is optimal to keep the number of stages to a minimum and not add any buffer stages.

The optimal stage effort is $h = \sqrt[3]{H} = 2.4$. The stage effort of the XOR gate is $gf = 3f = 2.4$, so $f = \frac{2.4}{3} = \frac{C_\text{load}}{C_\text{in}}$, and $C_\text{in} = 7.5C_U$. This means that optimal the $C_\text{load} = 6C_U$, and a x1 NAND-4 gate has an input capacitance of $5.5C_U$, therefore the optimal sizing for the NAND-4 gate is x1.

The stage effort of the NAND-4 gate is $gf = 2.2f = 2.4$, so $f = \frac{2.4}{2.2} = \frac{C_\text{load}}{C_\text{in}}$, and $C_\text{in} = 5.5C_U$. This means that optimal $C_\text{load} = 6C_U$ and an x1 inverter has an input capacitance of $2.5C_U$. Therefore, the optimal sizing for the inverter is x2.

This means that our propagate signal path consists of an x1 XOR gate, an x1 NAND-4 gate, and an x2 inverter. The propagate signal path can be seen in Fig. 16.

D. Final Stage Design

The schematic for our 4-bit adder stage can be seen in Fig. 17. In accordance with the standard carry-bypass adder design, the input carry bit can optionally be output through the multiplexer if each adder generates a propagate signal; otherwise, the adder behaves as a ripple-carry adder.

VI. 32-Bit Adder

Our complete 32-bit adder consists of eight 4-bit stages, linked to each other by their carry-in and carry-out lines, as seen in Fig. 18. An adder testbench was provided by the instructor, including the power supply and the load capacitance of each output line, as seen in Fig. 19. Test vector files were also provided. Final results are described below, and listed in Table II.

A. Function Verification

We simulated a transient analysis lasting 30 ns, with step size 50 ps, and with the vector file `sim_vec_function.vec` providing input signals. An `input0.err` file with no errors indicated that our design performed as expected (as seen in Fig. 20).

B. Critical path delay

We simulated a transient analysis lasting 30 ns, with step size 50 ps, and with the vector file `sim_vec_critical_path.vec` providing input signals. Delay measurements from the simulation indicated a critical path delay of 639.4 ps.

C. Current, P, PDP, and EDP Measurements

We simulated a transient analysis lasting 150 ns, with step size 100 ps, and with the vector file `sim_vec_I_P_PDP_EDP.vec` providing input signals.

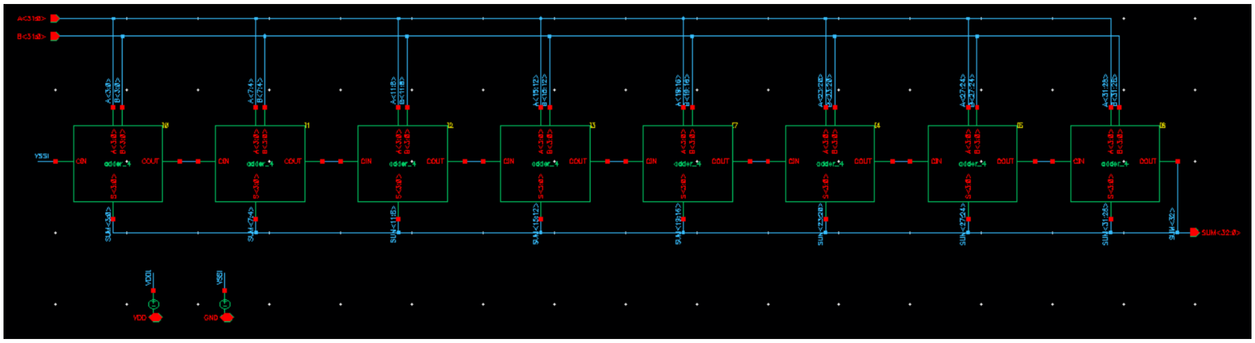


Fig. 18. Circuit schematic of the 32-bit adder.

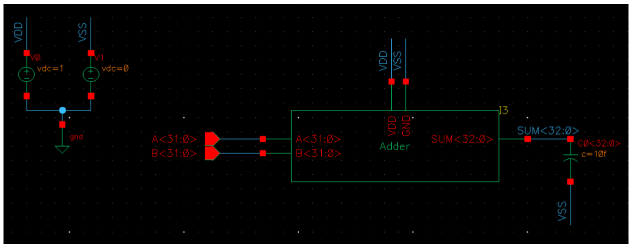
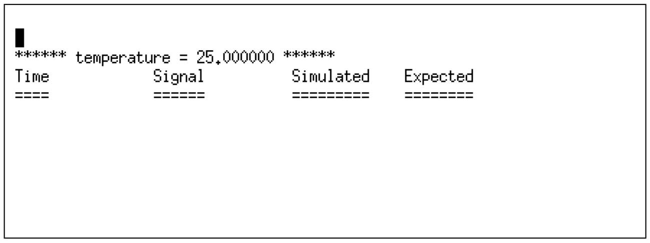


Fig. 19. Circuit schematic of the 32-bit adder testbench.



```
****** temperature = 25.000000 ******
Time            Signal          Simulated     Expected
====            ======          =========     ========
```

Fig. 20. Error log for the function verification test of our carry-bypass adder.

With the delay measurements from before, power measurements from the simulation indicated an average power consumption of $132.7\,\mu\mathrm{W}$, a Power-Delay Product (PDP) of $8.485 \times 10^{-14}\,\mathrm{W} \cdot \mathrm{s}$, and an Energy-Delay Product (EDP) of $5.425 \times 10^{-23}\,\mathrm{W} \cdot \mathrm{s}^2$.

D. Area Estimation

We estimated the area of our carry-bypass adder by summing up the total width of all NMOS ($\mathrm{W}_{\mathrm{N,\ total}}$) and all PMOS ($\mathrm{W}_{\mathrm{P,\ total}}$) transistors. W_{N} and W_{P} for each cell in the cell hierarchy is recorded in Table III (Note: $W_n = 90\,\mathrm{nm}$ represents the width of a minimum-sized NMOS).

The total area of the carry-bypass adder is $\mathrm{W}_{\mathrm{N,\ total}} + \mathrm{W}_{\mathrm{P,\ total}} = 3200W_n + 4592W_n = 7792W_n = 701.28\,\mu\mathrm{m}$.

TABLE III
AREA ESTIMATION OF EACH CELL IN THE CELL HIERARCHY, WITH RESPECT TO $W_n = 90\,\mathrm{nm}$

Cell Name	W_{N}	W_{P}
INV_X1	W_n	$1.5W_n$
TGATE_X1	W_n	W_n
NAND4_X1	$16W_n$	$6W_n$
mirror_adder	$48W_n$	$72W_n$
MUX	$16W_n$	$16W_n$
XOR_X1	$10W_n$	$15W_n$
adder_4	$400W_n$	$574W_n$
Adder	$3200W_n$	$4592W_n$

VII. Conclusion

With the application of logic design methodology to a common adder circuit topology, we have developed an efficient 32-bit carry-bypass adder fit for use in conventional computing systems. However, better topologies exist, such as the carry-select adder, carry-lookahead adder, or the entire family of tree adders. Further optimization could also be achieved on the current topology with more exhaustive simulations and design tweaking.

Working on this project, we acquired valuable experience with the logic effort method of delay optimization, hierarchical design and testbench simulations in Cādence Virtuoso, and the compilation of our process and results in a detailed report.