4-bit CMOS Transmission Gate Adder Module

David J. Grant
20116363

Xiuling Wang
20089932

ECE 637
Dr. Elmasry
Department of Electrical & Computer Engineering
University of Waterloo

April 14, 2003

Abstract

High-performance and low-power are the two main criteria in modern digital design. In this project, we have designed a high-speed, low power 32-bit conditional carry select adder using static CMOS Transmission Gate technology that overcomes the limitation of series-connected pass transistors in the carry propagation path. A 4-bit adder cell was created using 0.18 μm technology, which can be chained together to form wider-bit adders. A 16-bit and 32-bit adder was created and simulated. A different proposed 32-bit adder architecture enhances parallelism such that the delay of the 32-bit adder is the sum of the delay of the 16-bit adder and the delay of a single multiplexer. This feature enables the design to be expanded to 64-bit or 128-bit with little speed penalty. The design of the 4-bit module was done using some theoretical estimates, along with some experimental testing using an analog simulator. The sizing was optimized for delay first, while trying to keep the area low, which will reduce the power. This 4-bit module operates at voltages as low as 0.8 V which greatly reduces the power consumption. The design simulation on Cadence shows a worst-case delay of 0.50 ns for the 4-bit adder, 1.10 ns for 16-bit, and 1.22 ns for 32-bit. The total power consumption of the 32-bit adder is 200 μW. The 4-bit adder requires only 820 μm^2 of silicon area and has a very regular rectangular looking design.

Contents

List of Figures

List of Tables

Glossary

ALU Arithmetic Logic Unit

CBA Carry Bypass Adder

CCS Conditional Carry-Select

CLA Carry Lookahead Adder

CMC Canadian Microelectronics Corporation

CMOS Complementary Metal Oxide Semiconductor

CPL Complementary Pass Transistor Logic

CPU Central Processing Unit

CS Carry Select

CSA Contaional Sum Adder

DPL Double Pass Transistor Logic

DSP Digital Signal Processing

HA Half Adder

MUX Muliplexer

PDP Power-delay Product

EDP Energy-delay Product

RCA Ripple Carry Adder

TSMC Taiwan Semiconductor Manufacturing Company

TSPCL True Single-Phase Clocked Logic

TG Transmission Gate

VLSI Very Large Scale Integration

1 Introduction

The adder is the most commonly used arithmetic block of the Central Processing Unit (CPU) and Digital Signal Processing (DSP), therefore its performance and power optimization is of utmost importance. With the technology scaling to deep sub-micron, the speed of the circuit increases rapidly (doubling every 18 months assuming Moore's Law). At the same time, the power consumption per chip also increases significantly due to the increasing density of the chip. The design of an effective cooling system becomes a big major challenge in the VLSI circuit design. This in turn puts constraints on the speed improvement of circuit performance. Therefore, in realizing modern VLSI circuits, low-power and high-speed are the two predominant factors which need to be considered.

Like any other circuits' design, the design of high-performance and low-power adders can be addressed at different levels, such as architecture, logic style, layout, and the process technology. As the result, there always exists a trade-off between the design parameters such as speed, power consumption, and area.

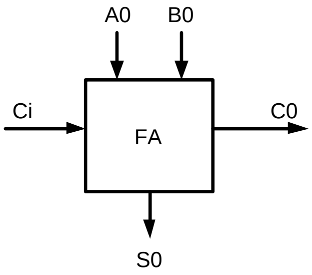
The objective of our project is to design a high-speed adder, along with lower-power and smaller area as a prime consideration. A Complementary Metal Oxide Semiconductor (CMOS) Transmission Gate (TG)-based Conditional Carry-Select (CCS) adder 4-bit block has been designed. This 4-bit block is designed to be connected to other 4-bit blocks to form higher bit adders. The CCS adder combines the advantages of Carry Lookahead Adder (CLA) and Contaional Sum Adder (CSA), and the further optimization enables the adder to expand into 32-bit, 48-bit or 64-bit with little overhead. The adder is implemented using 0.18 μm technology from Taiwan Semiconductor Manufacturing Company (TSMC).

2 High-Level Design

This section will give an introduction to the basic architecture of adders and go into the details of the high-level architecture of our design.

2.1 Adder architecture overview [1, 2]

A n-bit binary adder can be seen as the special arrangement of n full adder units that each take three one-bit inputs: A, B and carry C_i and which generate sum S and carry out C_o, as shown in Figure 1. The expression for S and C_o can be given as:



A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 1: Full adder and its true table.

$$S = A \oplus B \oplus C_i, \quad \text{and} \tag{1}$$

$$C_o = AB + BC_i + AC_i. \tag{2}$$

If the intermediate propagate P and generate G signals are used, the sum and the carry out can also expressed as:

$$S = P \oplus C_i, \quad \text{and} \tag{3}$$

$$C_o = G + PC_i = G\overline{C_i} + PC_i, \tag{4}$$

where,

$$G = AB \quad \text{(generate)} \qquad P = A + B \quad \text{(propogate)} \tag{5}$$

The characteristics of a few common adder architectures are summarized in the following subsections, which include: Ripple Carry Adder (RCA), Carry Bypass Adder (CBA), Carry Select (CS), CCS, CSA, and CLA.

2.1.1 Ripple Carry Adder (RCA)

An n-bit Ripple Carry Adder (RCA) is a simple cascading of n full adders as shown in Figure 2. In the RCA, the carry bit "ripples" from one stage of the adder chain to the next. The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

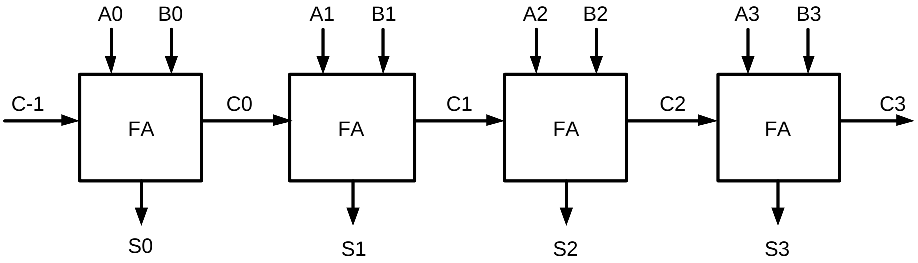


Figure 2: Ripple carry adder architecture.

$$t = (n-1)t_c + t_s, \tag{6}$$

where t_c is the delay through the carry stage of a full adder, and t_s is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly proportional to n, the number of bits, therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as a compact layout giving smaller chip area.

2.1.2 Carry Bypass Adder (CBA)

The Carry Bypass Adder (CBA) uses the additional bypass circuit controlled by propogate signals to speed up the carry propagation as shown in Figure 3. The delay of n-bit adder

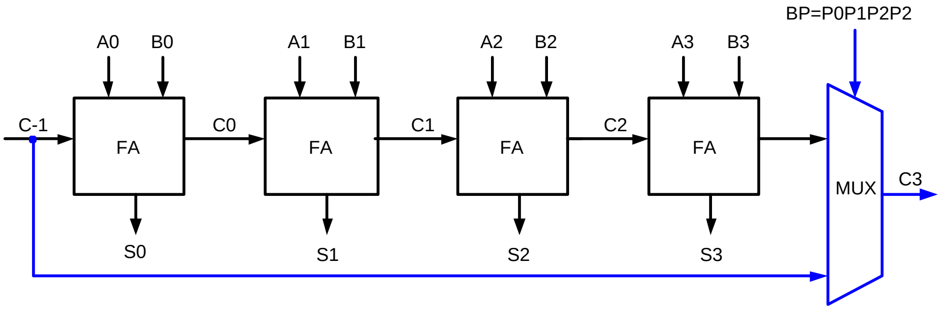


Figure 3: Carry Bypass Circuit (CBA) architecture

based on m-bit blocks of CBA rippled together can be given by:

$$t = t_{setup} + mt_{carry} + (n/m - 1)t_{carry} + t_{sum} \tag{7}$$

Comparing to the RCA, the CBA has slightly improved speed for wider-bit adders (still linear to n), but with higher active capacitance and the area overhead because of the extra bypass circuit.

2.1.3 Carry Select (CS)

The Carry Select (CS) realizes parallel computation of two results, each for different carry input assumptions (”0” or ”1”), as shown in Figure 4. The CS adder reduces the delay to a chain of Muliplexer (MUX)s, hence increases speed with a reasonable increase in area overhead. The delay of n-bit carry select adder based on an m-bit CBA blocks can be given by equation 8 when using constant carry number blocks and equation 9 when using successively incremented carry number blocks respectively. These equations are as follows:

$$t = t_{setup} + mt_{carry} + (n/m)t_{t_{mux}} + t_{sum}, \text{ and} \tag{8}$$

$$t = t_{setup} + mt_{carry} + (\sqrt{2n})t_{mux} + t_{sum}. \tag{9}$$

From equation 9, it is clear that for large values of n, the delay of the CS adder almost becomes a constant.

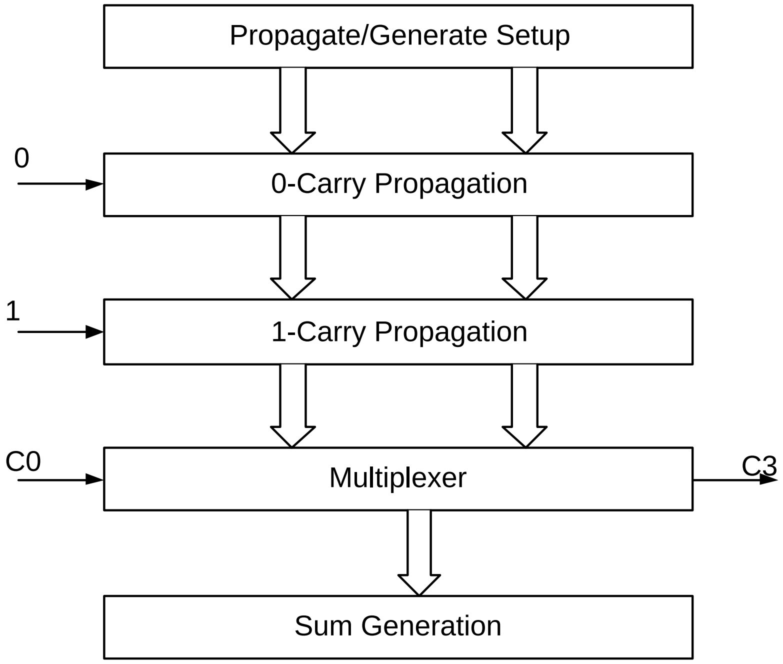


Figure 4: 4-bit CS adder architecture

The CCS adder and CSA are special types of the CS adder. It can be shown, that if well-optimized, the CSA outperforms other types of adders but with much higher area overhead due to the need to implement a large number MUXs in the selection circuits, while the CCS adder has better area efficiency, less power consumption, but slightly slower speed.

2.1.4 Carry Lookahead Adder (CLA)

The CLA removes the carry-ripple effect in the other types of adders described above. The architecture of a CLA is shown in Figure 5. Since the CLA generates a carry for each bit simultaneously, the delay is greatly reduced. The independent carry can be computed by expanding from equation 4:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}(\cdots + P_1(G_0 + P_0Ci, 0))) \tag{10}$$

In practice, it is not possible to use the CLA to realize constant delay for the wider-bit adders since there will be a substantial loading capacitance, and hence larger delay and larger power

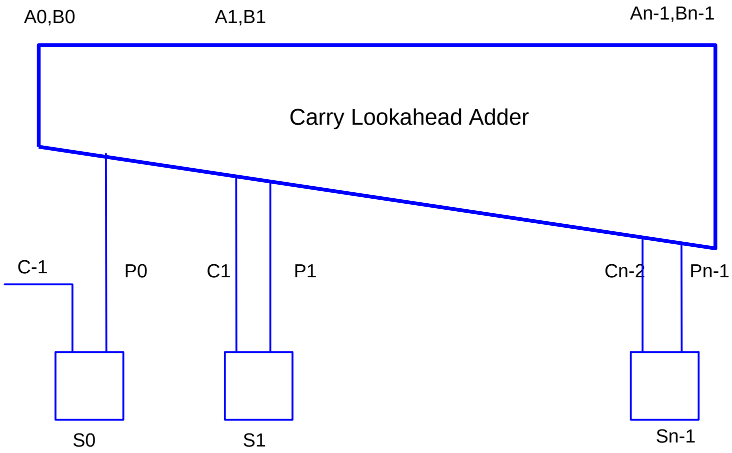


Figure 5: CLA architecture

consumption. The CLA has the fastest growing area requirements with respect to the bit size.

In practice, the mixture of architectures listed above are often used in the design of wider-bit adders to realize better optimization in terms of design metrics such as performance, power consumption, Power-delay Product (PDP), Energy-delay Product (EDP), area, and noise margin, etc.

2.2 Logic style overview

2.2.1 Static logic style

- CMOS consists of pull-up (PUN) and pull-down (PDN) networks. It uses the largest number of gates ($2N$ for an N-input gate), hence it has large capacitance and higher delay. The advantages of complementary CMOS is that it has the best power efficiency [3]. Furthermore, it has high noise margin and is more robust for voltage scaling and transistor sizing.

- Pseudo-NMOS reduces the number of gates to $N + 1$ by replacing the pull-up block with single PMOS transistor, therefore reducing the capacitance and improving the speed. The drawbacks of Pseudo-NMOS are lower and asymmetrical noise margin as well as higher standby power consumption.

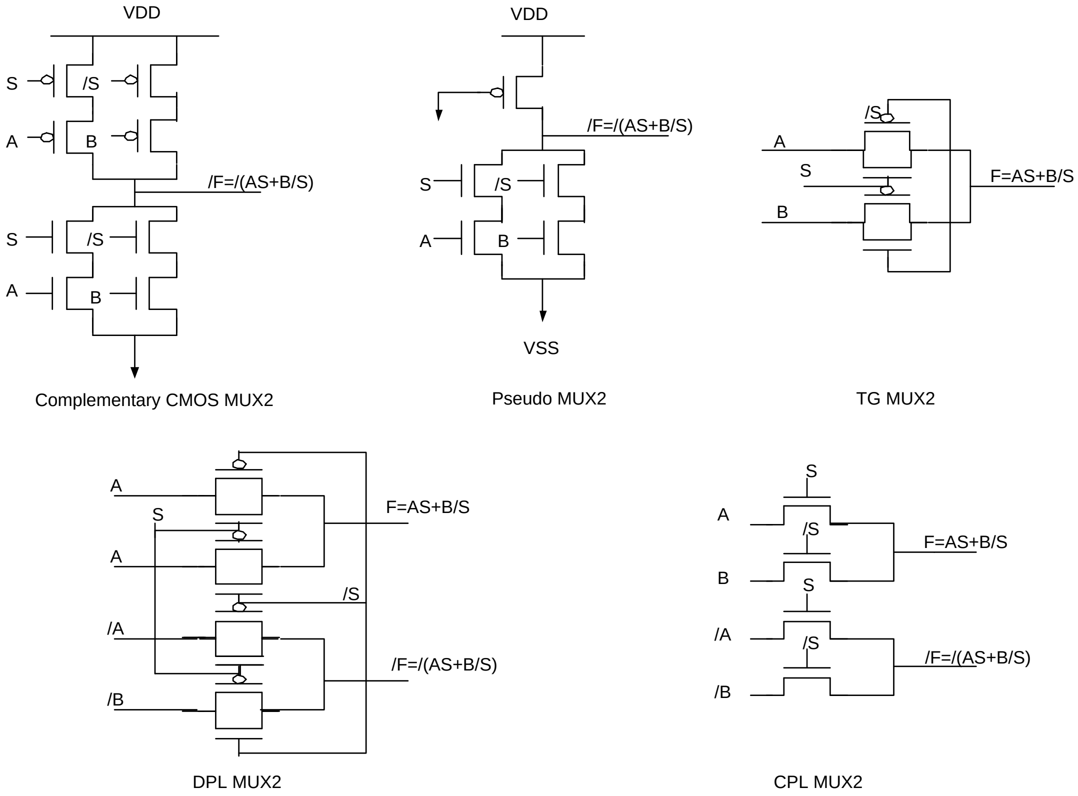
- TG (also called CMOS CPL logic in many cases) has the ability of a high-quality switch with low resistance and capacitance. It is one of the members of the ratioless logic family as the DC characteristics are independent of the input levels. Sizing is also not necessary in general, as the resistance and capacitance decrease and increase respectively as the gate W/L ratio is increased. TG is commonly used to implement of XORs and MUXs with the minimum number of transistors.

- Complementary Pass Transistor Logic (CPL) implements logic functions with NMOS-only. CPL is associated with several advantages including differential inputs/outputs availability, circuit simplicity and modularity. It can be efficiently realized in the implementation of certain types of circuits (such as XORs and adders) with small number of transistors. The disadvantages of CPL is reduced noise margin and higher static power consumption.

- Double Pass Transistor Logic (DPL) is a modified version of CPL that is suitable for low-voltage applications. DPL has balanced input capacitances, therefore reducing the dependence of the delay on the input data. Also, DPL provides full logic swing due to the use of PMOS gates as well as NMOS, and the dual current driving ability of DPL compensates for the additional PMOS gates [1]. The disadvantages of DPL is the higher number of transistors, hence higher area and higher power dissipation. A two-input MUX in complementary CMOS, pseudo-NMOS, TG, CPL, and DPL are shown in Figure 6.

2.2.2 Dynamic logic style

Dynamic logic uses a sequence of pre-charging and conditional evaluation phases to realize complex logic functions in a single NMOS pull-down or PMOS pull-up network, hence this requires less transistors and also has no static power consumption. The reduced overall ca-



Complementary CMOS MUX2

Pseudo MUX2

TG MUX2

DPL MUX2

CPL MUX2

Figure 6: Two-input multiplexer in CMOS, Pseudo-NMOS, TG, CPL, DPL

pacitance results in significantly improvement in the speed. The disadvantages of dynamic logic is the high dynamic power dissipation due to clock switching. Furthermore, the dynamic logic has clock skew and charge-sharing problems. The dynamic logic style would be implemented in applications where the performance is the primary concern.

The main types of dynamic logic are Domino, NP-CMOS, and True Single-Phase Clocked Logic (TSPCL). The C^2MOS latched NP-CMOS (also called NORA-CMOS) can be used in the effective implementation of pipelined circuits. The structure of Domino, NP-CMOS, and TSPCL are shown in Figure 7.

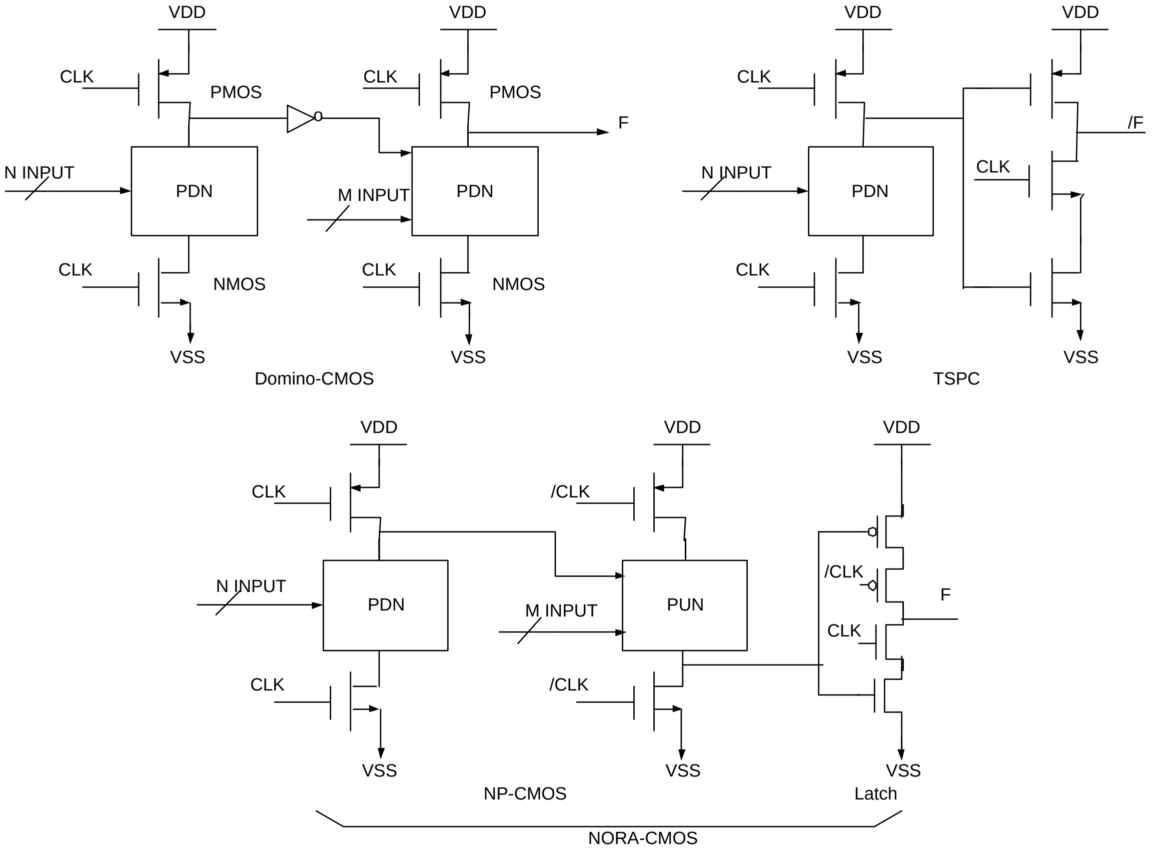


Figure 7: Dynamic logic styles.

2.3 High level architecture of the chosen adder

Among the various adders described above, the CCS adder and the CSA are shown to have superior speed performance [1, 4]. With the consideration of lower power consumption and higher area efficiency, the static CCS adder has been chosen in this project.

The CCS adder is realized in a 4-bit CCS modules as shown in Figure 8. Each 4-bit module consists of:

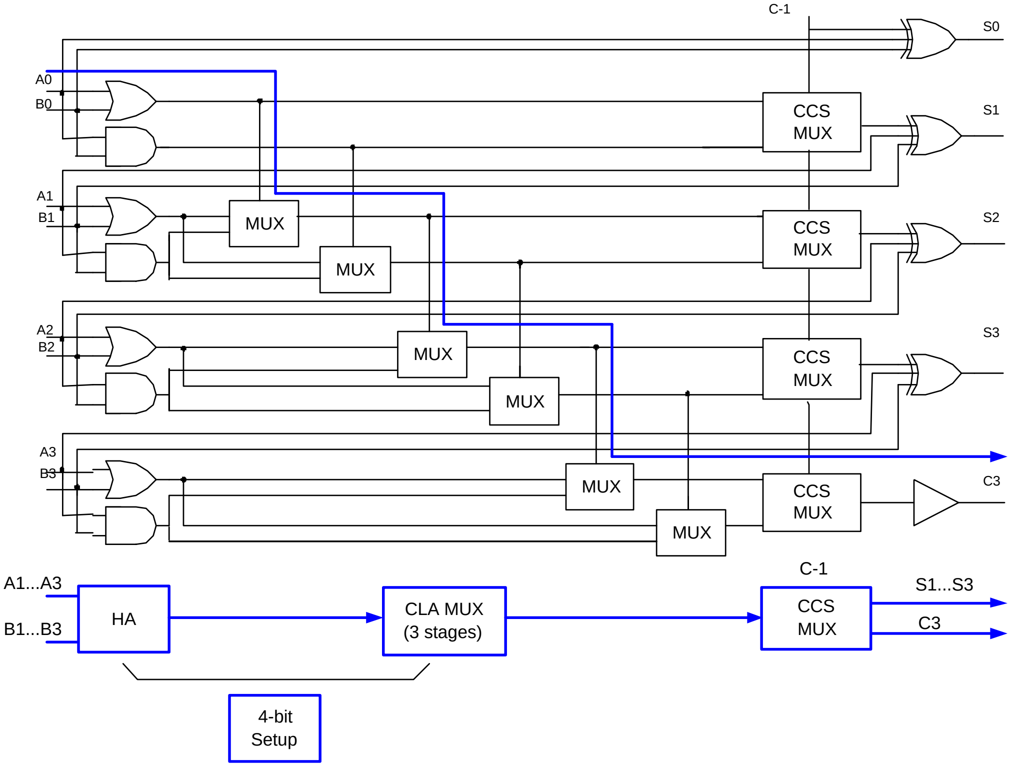


Figure 8: High-level architecture of 4-bit CCS adder.

- Four Half Adder (HA) blocks (HAS: four pairs of OR/AND gates to generate propogate/generate signals respectively).
- Three-stage carry lookahead generation blocks (CLA MUXs: three-pairs of two-input multiplexers, for a total of six MUXs).
- Four carry selection blocks (CCS MUXs: four two-input MUXs).
- Four sum generation blocks (four three-input XOR gates).

The idea of the CCS adder is to process generate and propogate signals produced by the

HAs to generate the carry signals. The carry signal from equation 4 can reorganized as

$$C_i = \overline{C_{i-1}} G_i + C_{i-1} P_i. \tag{11}$$

For example,

$$\begin{gathered} C_0 = \overline{C_{-1}} G_0 + C_{-1} P_0 \\ => C_0 = P_0 (\text{if} C_{-1} = 1), \qquad C_0 = G_0 (\text{if} C_{-1} = 0) \end{gathered} \tag{12}$$

$$\begin{gathered} C_1 = \overline{C_0} G_1 + C_0 P_1 \\ => C_1 = \overline{P_0} G_1 + P_0 P_1 (\text{if} C_0 = 1), \qquad C_1 = \overline{G_0} G_1 + G_0 P_1 (\text{if} C_0 = 0) \qquad . \end{gathered} \tag{13}$$

The equations for the other carry bits can be derived in a similar fashion.

The architectures of for 16-bit and 32-bit CCS adders are shown in Figure 9. In each 4-bit carry block of the 16-bit architecture, four MUXs controlled by C_{i-1} produce four output

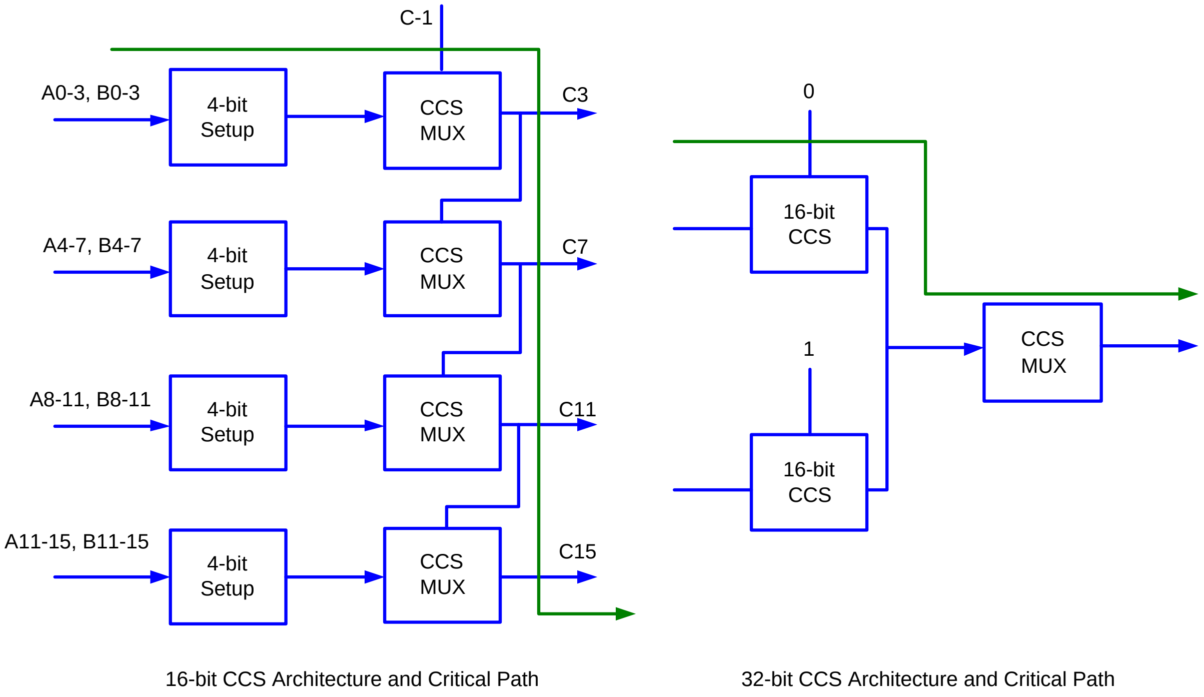


Figure 9: High-level architecture of 16-bit and 32-bit CCS adders.

carry signal C_i, C_{i+1}, C_{i+2}, and C_{i+3}, where,

$$C_i = \overline{C_j}C_i\,(\text{if}\,C_j = 0) + C_jC_i(\text{if}C_j = 1), i = j+1 \sim j+4, \quad j = -1, 3, 7, 11 \tag{14}$$

The 32-bit based architecture is further optimized by applying the additional CCS architecture (CCS MUX) twice, with parallel computation of two results based on carry input assumption ('0' or '1') (see Figure 9 and 10). Because the carry signals of the upper 16 bits are calculated in parallel with those of the lower 16 bits, this architecture results in faster operation. Comparing to the delay of 16-bit CCS adder, the additional delay of 32-bit CCS adder is the delay of a single MUX, plus delay due to wiring parasitics.

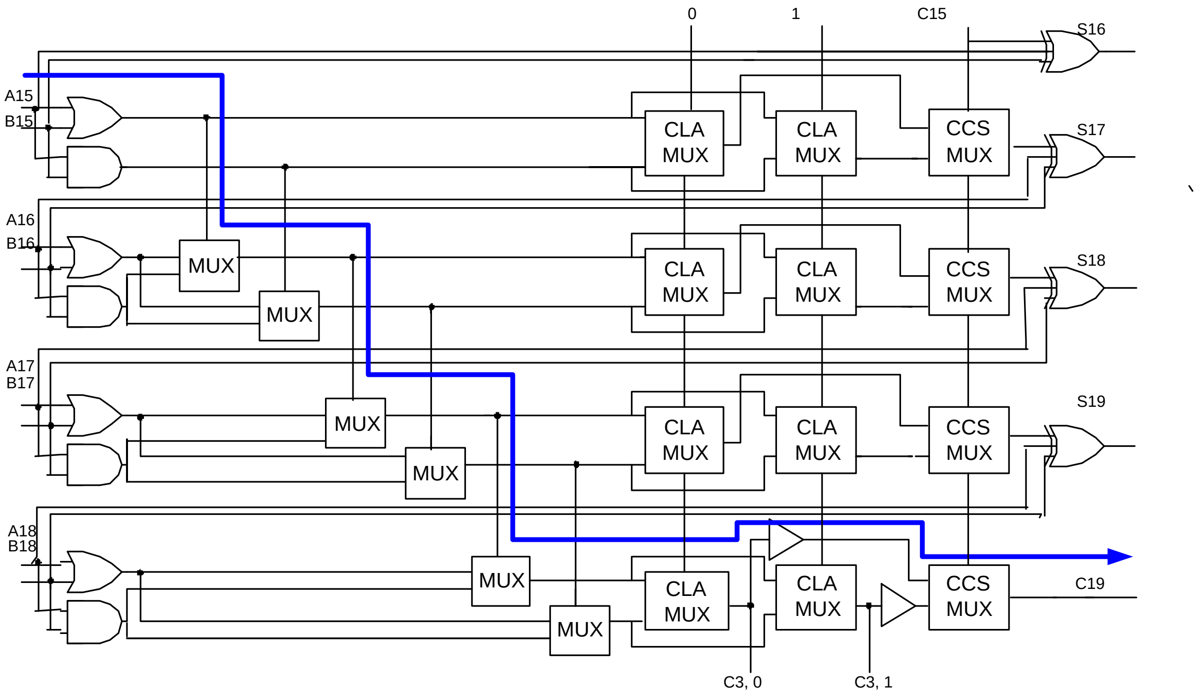


Figure 10: High-level architecture of 4-bit block in the upper 16 bits of 32-bit CCS adder.

3 Circuit Design

3.1 Low-Power Design

Designing low-power devices is done through many techniques. The techniques used in this design, to reduce the power consumption are [1]:

1. Use of a more static style over dynamic style.
2. Reduction of V$_{DD}$.
3. Minimize area of all capacitances, as well as minimizing the total area.

These techniques are mostly inherant in our design. Our design style is 100% static through the use of static CMOS and static TG gates. TG can have substantial lowering of V$_{DD}$ from the nominal value because PMOS and NMOS devices are both used, thus there is full-logic swing. Minimization of area is possible with TG gates because they can be mostly made to be minimum sized.

3.2 Capacitance Estimation

3.2.1 Estimation of C_{in}

The input capacitance of any circuit can be approximated by simply taking the sum of all the gate capacitances which are attached to a particular input signal,

$$C_{in} = \sum_{i=0}^{N} C_{gate,i} \tag{15}$$

The C_{gate} for a given gate with n transistors can be calculated from the formula,

$$C_{gate} = C_{ox} \sum_{i=1}^{n} (WL)_i, \tag{16}$$

where C_{ox} is a process parameter, and it is the capacitance (in Farads) per unit of square area. It is calculated as

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}} \tag{17}$$

where t_{ox} is the oxide thickness. For cmosp18 technology from Canadian Microelectronics Corporation (CMC), t_{ox} is around 4.1 nm and the dielectric constant for silicon dioxide is $\epsilon = 3.85 \times 8.85 \times 10^{-12}$. Thus we have

$$C_{ox} = \frac{\epsilon}{t_{ox}} = \frac{(3.85)(8.85 \times 10^{-12})F/m}{4.1 \times 10^{-9}m} = 8.31 \times 10^{-3}F/m^2 = 8.31fF/\mu m^2 \tag{18}$$

For a specific gate, the input capacitance can be calculated by multiplying C_{ox} by W and L as shown above in equation 18. For a minimum sized device of 0.18 $\mu m \times$ 0.22 μm, the input gate capacitance would be $C_{in} = 0.33\ fF$.

3.2.2 Output parasitic capacitances

In a standard CMOS gate, the output parasitic capacitances consist of the gate to drain overlap capacitance, C_{ov} and the drain junction capacitance, which is made up of the junction-to-body capacitance C_{jb} and the drain sidewall capacitance, C_{jsw}. For a transmission gate, the story is a bit different. One needs to consider the junction capacitance of the source and the drain, as well as the overlap capacitance of the source and drain.

The overlap capacitance will be equal for the drain overlap and source overlap capacitances. It is calculated from:

$$C_{ov} = C_{GD0} \times W, \tag{19}$$

where C_{GDO} is a SPICE parameter. For an NMOS transistor in cmosp18, $C_{GDO} = 3.665 \times 10^{-10}F/m$ and for a PMOS transistor in cmosp18, $C_{GD0} = 3.28 \times 10^{-10}$. Thus for a minimum sized device with $W = 0.22\ \mu m$, $C_{ov} = 1.5 \times 10^{-16}\ fF$ or $C_{ov} = 0.07\ fF$, where an average between NMOS and PMOS devices has been used. In a standard CMOS inverter, for example, the overlap capacitance is actually not connected to ground, but it experiences the Miller effect, so it can be replaced with a capacitance to ground, with double the original value.

The junction capacitance is made up of the junction-to-body capacitance, C_{jb} and the side-wall capacitance, C_{jsw}. C_{jb} is calculated as follows:

$$C_{jb} = W \cdot D \cdot \frac{1}{V_{DD}} \int_0^{V_{DD}} \frac{CJ0}{[1 + V_j/V_b]^{mj}} dV_j, \tag{20}$$

where $CJ0$ is a SPICE parameter and is equal to $CJ0 = 1.1 \times 10^{-3} F/m^2$ for cmosp18 (averaging the CJ parameter for NMOS and PMOS)[1]. V_j is the voltage on the drain or source to body junction, and V_b is the built-in voltage across the said junction (0.69 V for NMOS and 0.90 V for PMOS), and V_{DD} is the supply voltage. mj is the grading coefficient and is equal to 0.36 for NMOS and 0.45 for PMOS. D is the length of the drain/source contact. The formula above calculates the average drain-to-body junction capacitance, as it changes depending on the drain/source voltage, which alters the junction width. Assuming $V_{DD} = 1.8\ V$, and taking an average between PMOS and NMOS devices, the integral evaluates to $5.2 \times 10^{-4} F/m^2 \cdot W \cdot D$. For a minimum sized device of $D = 0.48\mu m$ and $W = 0.22\mu m$, $C_{jb} = 0.05\ fF$.

The sidewall capacitance is given by:

$$C_{jsw} = (2D + W) \cdot \frac{1}{V_{DD}} \int_0^{V_{DD}} \frac{CJSW0}{[1 + V_j/V_b]^{mjsw}} dV_j, \tag{21}$$

which is the same as equation 20 above, only the SPICE parameters $CJ0$ and MJ have been replaced with $CJSW0$ and $MJSW$ respectively. $CJSW0 = 2.3 \times 10^{-10} F/m^2$, $MJSW = 0.2$ for NMOS and $MJSW = 0.37$ for PMOS, and V_b is the same as for the junction to body capacitance calculations. D in equation 21 above is the length of the drain or source junction. For minimum source drain lengths in cmosp18, $D \approx 0.48\mu m$. Assuming $V_{DD} = 1.8\ V$, the

[1] Unless otherwise stated, the values will be averaged for NMOS and PMOS devices. This approximation is appropriate since our design is completely complementary and uses an equal number of NMOS and PMOS transistors.

integral in equation 21 evaluates to $C_{jsw} \approx 1.2 \times 10^{-10} F/m \cdot (2D + W)$. Thus $C_{jsw} \approx 0.14 fF$ for a minimum sized transistor.

A summary of the capacitances as calculated above is given in Table 1.

Table 1: Summary of capacitances for $0.18\mu m$ technology

Capacitance	Formula	Value for $0.18\mu m \times W\mu m$ transistor
C_{gate}	$8.3\ fF/\mu m^2 \cdot W \cdot L$	$1.5\ fF/\mu m \times W$
C_{ov}	$0.35\ fF/\mu m \cdot W$	$0.35\ fF/\mu m \times W$
C_{jb}	$0.52\ fF/\mu m^2 \cdot W \cdot D$	$0.25\ fF/\mu m \times W$
C_{jsw}	$0.12\ fF/\mu m \cdot (2D + W)$	$0.12\ fF + 0.12\ fF/\mu m \times W$

3.3 Sizing of Multiplexer

The MUXs is the most important part in the critical path of our circuit. There are two main MUXs in our circuit, they will labelled as CLA_MUX and CCS_MUX from this point forward, as shown in Figure 11. This figure shows the main building blocks involved in our design.

3.3.1 Sizing of Inverter Inside Multiplexers

The “S” (select) input signal loads an inverter as well as two transmission gates. Initially assuming a minimum-sized inverter (with $W_n = 0.22\ \mu m$ and with $W_p \approx 2 \times W_n$)[2], the S input will have a load capacitance of $1.5\ fF \times (2W_n + 2W_p) = 1.9\ fF$. The output of $\overline{S}$ will have a load of $2.6\ fF$ including the load of two transmission gates, and the output parasitic load of the single inverter. So $\overline{S}$ will lag behind the S input by one gate delay, so it makes sense to make the inverter minimum sized. This will reduce the overall delay for the S signal to transition due to the decreased load, so that the $\overline{S}$ can switching as soon as possible. Make this inverter as small as possible also reduces the capacitance, and thus reduces power. This logic applies to both CLA_MUXs and CCS_MUXs

[2] Note: the ratio of W_p/W_n is always chosen to be somewhere around 2.

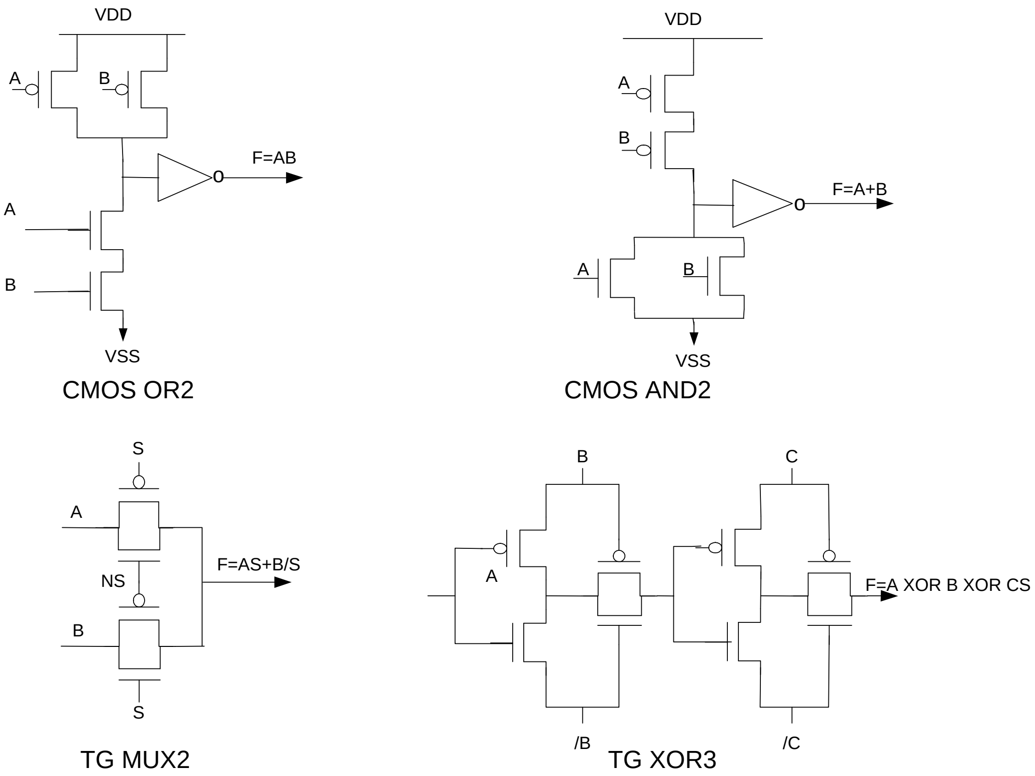


Figure 11: The building blocks of the CCS adder.

3.3.2 Sizing of Transmission Gates Inside Multiplexers

The transmission gates make up the crucial part of our critical path in the first 4-bit stage. The CLA_MUX blocks are very important MUXs because they comprise the critical path in the first 4-bit block of any adder made of 4-bit CCS adder stages. In a 32-bit adder made up of 8 cascaded 4-bit cells, the first 4-bit block still accounts for about $\approx$ 40% of the total delay of the circuit.

The worst-case delay of a chain of pass transistors can be modelled by the linearized RC network [2]. The linearized on-resistance of the transmission gate is determined, and is represented by a resistor. The linearized diffusion capacitance is also determined for each transistor. One then ends up with a circuit like that shown in Figure 12. The propagation delay of this circuit is:

$$t_p = 0.69 \sum_{i=1}^{N} C_i \left(\sum_{j=1}^{i} R_j \right) , \tag{22}$$

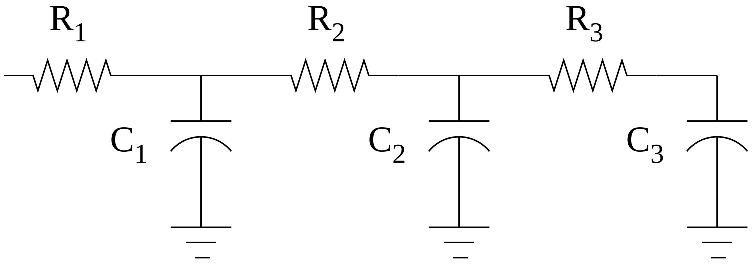


Figure 12: Equivalent network to determine delay of transmission gate chain.

so for the example shown in Figure 12, the delay will be approximately,

$$t_p = 0.69(C_1R_1 + C_2(R_1 + R_2) + C_3(R_1 + R_2 + R_3)). \tag{23}$$

As explained in [2], since R_1 occurs six times in Equation 23, it is a good idea to minimize R_1. We have three MUXs in each bit-slice, however, two of the MUXs are really in parallel to each other. So really, we just have a network of two MUXs. The first mux, (a CLA_MUX) is represented R_1, with a diffusion and overlap capacitance of C_1. The second mux, (a CCS_MUX) is represented by R_2, with a diffusion and overlap capacitance of C_2.

Minimizing the resistance of the first transmission gate in the chain will also increase its diffusion capacitance, so clearly, to a first-order approximation, increasing the W of a transmission does not change the delay at all. Increasing the W of a TG will also increase the gate capacitance which will load the select signal, S, more heavily. Based on the logic above, we can infer that to design for low power, the transmission gates should be made as small as possible. And to design for high performance, the first transmission gate in the chain should be made larger than the second transmission gate. The following sizes were determined by trial and error and are shown in Table 2. The table shows two types of sizing strategies. In the first strategy, all MUXs were sized equally, and in the second strategy, the first CLA_MUXs were sized larger than the CCS_MUXs. Table 2 shows that the 920/420 [3] and 420/220 sizing gives the best delay, with equal rise times. This is what was used in the

[3] These are units are in nanometers, and will be assumed to be in nanometers for sizing of W from this point forward.

Table 2: Experimental results for sizing of CLA_MUX and CCS_MUX

CLA_MUX W/L	CCS_MUX W/L	t_{PLH}	t_{PLH}	Power	PDP
($\mu m/\mu m$)	($\mu m/\mu m$)	(ps)	(ps)	(10^{-5} W)	(10^{-14} $J \cdot s$)
220/220	same	686	491	1.41	8.29
420/220	same	584	587	1.44	8.43
420/420	same	599	526	1.56	8.74
920/420	same	564	615	1.70	10.02
920/920	same	627	605	1.94	11.95
920/420	420/420	577	542	1.59	8.88
920/420	420/220	560	575	1.50	8.53
750/330	500/220	554	591	1.50	8.56
920/920	920/420	582	533	1.65	9.17

circuit described in this document. The power is slightly greater than for the equal sizing scheme, however, in this instance we opted for lower delay, since the PDP did not vary by too much, as seen in Table 2

If we consider this 4-bit adder block being used in a higher-bit design, such as a 16-bit adder or 32-bit adder, it makes sense to reduce the size of the CCS_MUXs as much as possible. These MUXs load the carry-out signals from all the previous 4-bit blocks, and this makes up the critical path for higher-bit designs. So using 420/220 as determined experimentally above for the CCS_MUX makes very good sense.

3.4 Sizing of other gates

The other gates in the design are not as important as the MUXs. However, they do play an important role. In a 32-bit adder made of of 4-bit blocks of this adder, the first 4-bit block makes up 40% of the total delay time. So although the critical path in the later 4-bit blocks only includes the CCS_MUXs, in the first 4-bit block, the critical path includes an AND and OR gate, and the XOR gate provides some loading of the signals (see Figure 9 for illustration of the critical path).

3.4.1 Sizing of 3-input XOR Gate

The XOR gate simply loads the critical path circuit, and creates the sum outputs. The sum outputs are not part of the critical path, however, except on the last 4-bit block of a large adder made with multiple 4-bit blocks. It is optimal for performance and power considerations, to make the XOR minimum sized. Also, to make the falling edges equal to the rising edges, $W_p \approx 2 \times W_n$.

3.4.2 Sizing of AND and OR Gates

Since the input circuit is not specified, it has been assumed that the AND and OR gates should present the least possible load to the input circuit. The AND and OR gates are made up of NAND and NOR gates followed by an inverter for each. The NAND and NOR gates were made with W_{eff} of an equivalent 920/420 inverter. It was decided to use 920/420 instead of 420/220 since there was enough room in the layout to do this efficiently, and because it improved the performance slightly. The input from the test-bench also had unlimited driving capability, so the effect of the increased loading was not noticed. Standard CMOS sizing was then used, thus the pull-down NMOS section of the NAND gate needed to have its W doubled, and the pull-up PMOS section of the NOR gate needed to have its W doubled.

Each NAND and NOR gate needed an inverter at each output, in order to create an AND and OR gate. The inverter acts as buffer to drive lots of fan-out. The size of this inverter was determined experimentally. The best size was determined to be approximately 2840/1420. The numbers were rounded up or down in order to make the drain/source contacts fit more optimally into each device.

3.4.3 Sizing of Buffer

This buffer is placed after every 4-bit adder stage. It helps restore the rise time and fall time and buffer the signal, which needs to provide enough to drive for 4 TG gates in the following 4-bit block. The load capacitance of S input to a minimum-sized TG gate inside the

CCS_MUX is 1.5 $fF/\mu m \times (0.22 + 0.42)\ \mu m = 0.96\ \mu m$. There are two of these gates, plus a minimum sized inverter, so the total input capacitance of one CCS_MUX select signal, S, is 2.88 fF. There are four of these gates in one 4-bit block which the C_{out} signal has to load, so the total load is approximately 11.5 fF. It was decided to make the buffer with two different sized inverters, the second inverter being larger than the first. This would make the delay more optimal [2]. Based on the output load of approximately 11.5 fF (neglecting output parasitic and wiring capacitances, and assuming that the first inverter is made to have W/L of 920/420 (this corresponds to a load of approximately 2.0 fF), the second inverter inside the buffer's desired input capacitance can be calculated as follows:

$$C_{buf,in} = 2.0\ fF, \qquad C_{1,buf} = \alpha \times 2.0\ fF, \qquad C_{2,buf} = 11.5\ fF$$

$$\alpha = \sqrt{\frac{11.5}{2.0}} = 2.4 \tag{24}$$

$$C_{buf,2} = 2.0\ fF \times \alpha = 2.0\ fF \times \alpha = 4.8\ fF. \tag{25}$$

This sizing for the two inverters will make each inverter have approximately equal delays. This corresponds to sizing of approximately 2200/1100 the second inverter; however, using analog simulations, it was determined that the optimal value, optimizing delay, was in fact around 2840/1420 [4]. It makes sense that the simulations showed that this second inverter needed to be made larger. This is because in the initial estimation, output parasitic capacitances were all neglected.

3.5 Summary of Transistor Sizing

A summary of all the transistor sizings is given in table 3.

[4] These widths were rounded down/up a bit in order to make the layout simpler, to fit the contacts more optimally

Table 3: Summary of all transistor sizings in the design

Element	PMOS size (*nm*)	NMOS size (*nm*)
NAND	920	920
AND inverter	1420	2840
NOR	1920	420
OR inverter	1420	2840
CLA_MUX	920	420
CLA_MUX inverter	880	330
CCS_MUX	420	220
CCS_MUX inverter	880	330
XOR (all transistors)	420	220
Buffer (first inverter)	920	420
Buffer (second inverter)	1420	2840

4 Physical Layout

The layout was done in $0.18 \mu m$ technology provided by TSMC through CMC. The design of the 4-bit CCS adder module was done entirely from scratch and no "standard cells" were used in the design. The main objectives of the layout are:

- Minimize all internal capacitances, by minimizing the active area of the transistors, and thus minimizing power.
- Minimize total area to reduce manufacturing costs.
- Maximize area efficiency.
- Minimize drain/source junction resistances by using fingered layouts and joining common source/drains on same nets.
- Minimize parasitic capacitance and resistance, especially along critical path, and at high activity nodes.

4.1 Basic NMOS and PMOS devices

The first layouts created were the basic layouts of NMOS and PMOS devices. All our devices use minimum length channel, in order to get the best speed advantage out of this sub-micron technology. A minimum sized layout of an NMOS device with a substrate contact is shown in Figure 13 with $W = 0.22\ \mu m$. A PMOS device is shown in Figure 14 with $W = 0.42\ \mu m$. Note: $W = 0.42\ \mu m$ is considered minimum sized for PMOS, since making is this size makes gives it the same ON-resistance as the $0.22\ \mu m$ NMOS transistor. The NMOS device and PMOS device shown in Figures 13 and 14 have a substrate contact and n-well contact respectively. Of course every single transistor does not need its own substrate contact. For cmosp18 technology, a substrate or n-well contact is only needed within $5\ \mu m$ of all active gate regions.

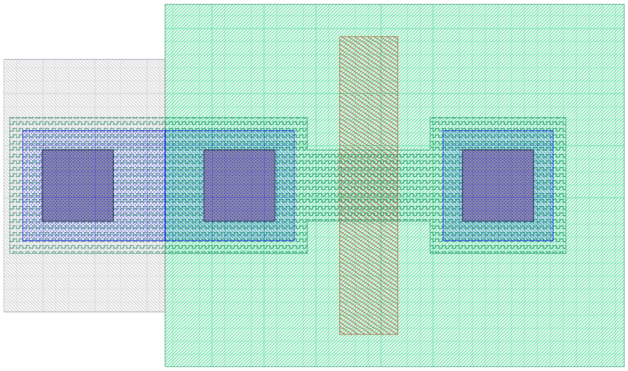


Figure 13: Layout of minimum-sized NMOS transistor with substrate contact in 0.18 μm technology.

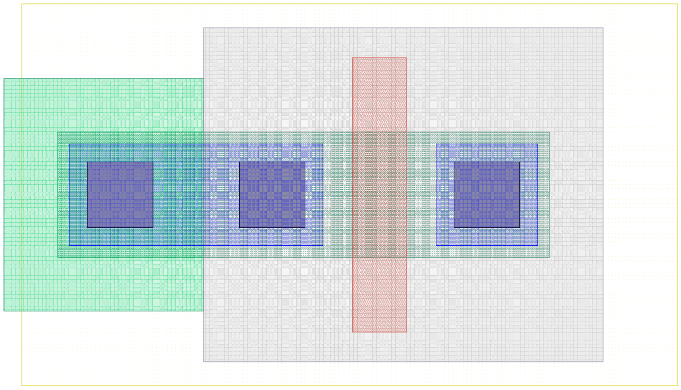


Figure 14: Layout of minimum sized PMOS transistor with n-well contact in 0.18 μm technology.

There is one drawback to making the width of the NMOS transistor as low as 0.22 μm. The drain and source areas remain constant between the 0.22 μm and 0.44 μm design, only the channel width changes. This means that the source and drain junction capacitances will remain the same, while the effective resistance in the channel has increased. Also, in order to satisfy design rule PO.C.1, which says that the *active*-to-*poly* spacing must be 0.10 μm [5], the contacts must actually be moved further away from the active region by 0.08 μm on each side. This further increases the resistance, as well as the junction capacitance. The gate capacitance is reduced, however, by decreasing the capacitance. Since our transmission gates are driving the gates of other transmission gates, the reduced driving capability of the smaller 0.22 μm size is compensated by the smaller size of the gate of the TG it drives.

In order to maximize the number of contacts to the source and drain, to minimize source and drain contact resistance, each transistor's W value can only be increased in 0.5 μm increments. This is due to the design rules which state that the contacts must be sized with 0.22 μm in all dimensions, and also that they must be spaced apart by 0.28 μm [5]. An example of a scaled NMOS transistor is shown in Figure 15 and a scaled PMOS transistor is shown in Figure 16.

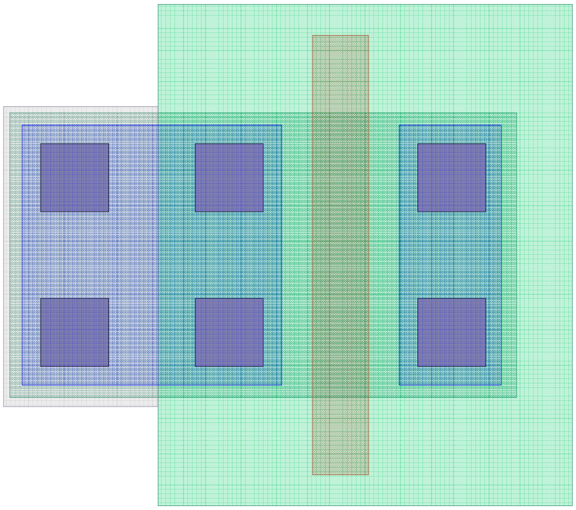


Figure 15: Layout of $W = 0.92$ μm sized NMOS transistor with substrate contacts in 0.18 μm technology.

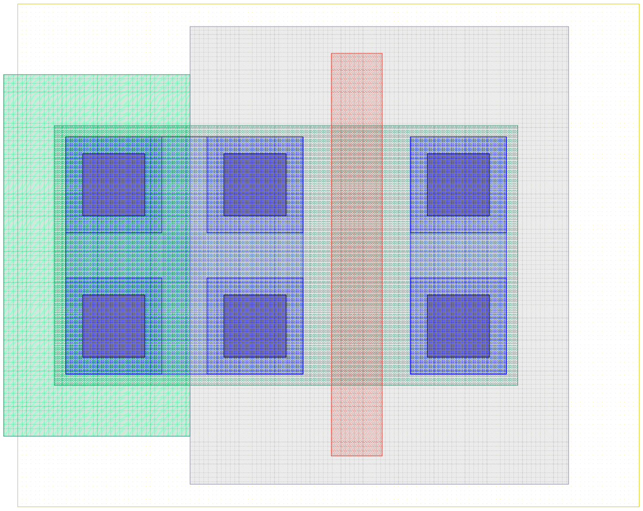


Figure 16: Layout of $W = 0.92\ \mu m$ sized PMOS transistor with substrate contacts in 0.18 μm technology.

4.2 2:1 Transmission Gate (TG) Multiplexer (MUX)

The MUX is the most important element in our CCS adder. It should be as fast as possible, and small, in order to reduce the parasitic resistance and capacitance of signal paths flowing from MUX to MUX. One of our MUX layouts is shown in Figure 17.

4.3 AND/OR layout

The AND and OR gates required a very large inverter. We thus used a fingered layout, as shown by the AND gate in Figure 18.

4.4 Routing

Some initial routing was decided arbitrarily near the beginning of the layout task. It was decided that all transistors should have their channel length running horizontally and their width vertically, as shown in the Figures 13 and 14. This leads to very short *poly* (dark red) lines connecting the gates of the transistors, and also leads to a minimal number of corners on the *poly* lines. Since polysilicon has a lower conductivity than metal, it is a good idea to

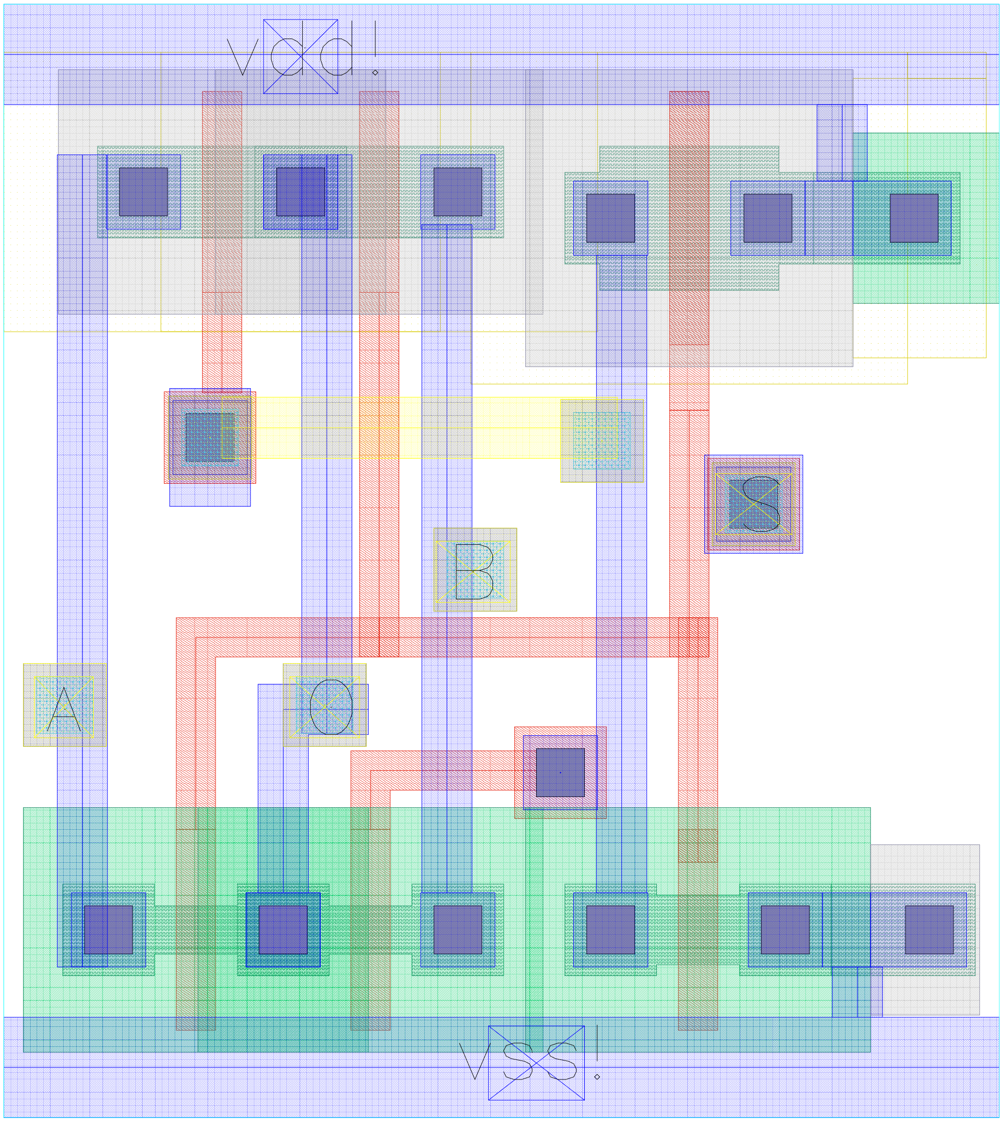


Figure 17: Layout for 420/220 transmission gate multiplexer

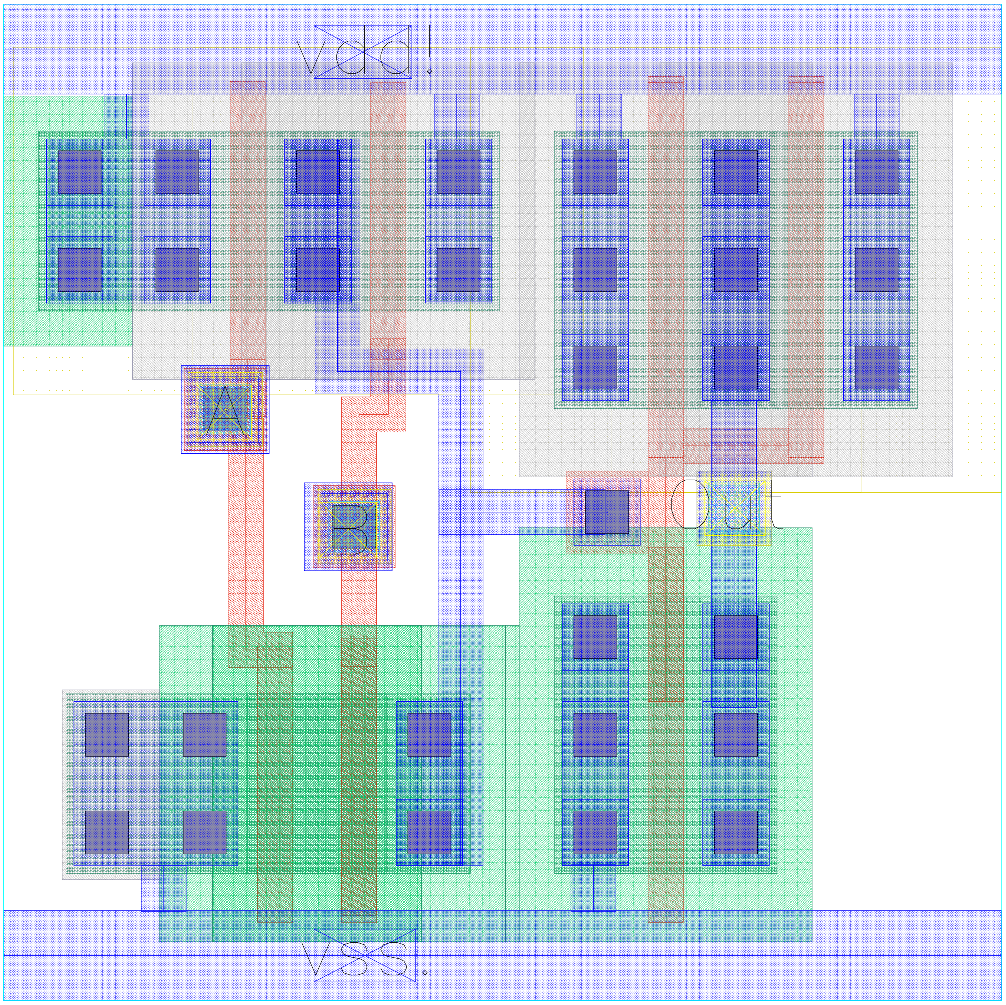


Figure 18: Layout of CMOS AND gate with fingered gate for inverter

keep the polysilicon/*poly* area minimized.

Many of the transistors' drains and sources need to be connected to each vertically, like the drain contacts of a CMOS inverter. For these connections it makes the most sense to use the *metal1* layer (blue) to avoid unnecessary contacts to higher layers, which can take up area. The *metal1* can easily be routed horizontally overtop of the *poly* layer over short distances.

All cells are arranged horizontally in each bit-slice, and are connected horizontally using the *metal2* layer (yellow) layer. This is convenient because some output signals may be in the middle of a cell, and the *metal2* can go over top of the *metal1* and *poly* if need be, although this should not be done very much, to reduce parasitic capacitance between these layers.

Most connections in our circuit will be in the horizontal direction within each bit-slice of our adder, connecting subsequent blocks to each other. Some connections go between bit-slices, namely, the outputs of the CLA_MUXs, which are connected to the select signals of the CLA_MUXs of the higher-order bit slice. The *metal3* (light red) layer should be used for these connections. This is advantageous for several reasons: 1) It can cross over top of the horizontal *metal2* layer, 2) part of our critical path must flow in the vertical direction, so this lower-capacitance layer is a good choice, and 3) it is a better choice than the *metal4* layer, because the resistance of our many vias on the critical path will be less by using *metal3*.

There are a few horizontal connections which must get from the inputs (A_i, B_i) to the output 3-input XOR gate. This is accomplished using the *metal4* layer (green). This is a good choice because this layer will have low capacitance, although it doesn't really matter too much because these signals are not on the critical path. Although these metal lines can cause coupling capacitance with the signals on layers beneath it, so having it on the lower capacitance *metal4* layer is a good choice.

4.5 V$_{DD}$ supply and Ground Routing

The V$_{DD}$ and V$_{SS}$ lines have been routed to the individual transistors horizontally using the *metal1* layer. This makes for simple connections to the body and/or source terminals of the transistors, without the need for contacts. The main V$_{DD}$ and V$_{SS}$ lines will be routed on the *metal5* or *metal6* layers. This is where the power and ground will be routed across the chip.

At 110^o C, the *metal5* layer can withstand 1.0 $mA/\mu m$ of current before it melts. The *metal6* layer can withstand 1.6 $mA/\mu m$ of current before it melts. We have measured the peak current of our circuit and found it to be approximately $I_{peak} = 1.5\ mA$. This current provided by the supply lines is actually reduced by a factor of four, since there are two V_{DD} lines per 4-bit block and these are connected to the *metal6* V_{DD} lines in two places. This would mean our V$_{DD}$ and V$_{GND}$ lines to the individual transistors (*metal1* layer) should be able to withstand $\approx 0.4\ mA$. Thus we made the V_{DD} and V_{SS} lines 0.46 μm wide.

The vias can all withstand 0.28 mA/via. So if the current will exceed this value in one via, then another via needs to be added in parallel, to reduce the peak current. We have assumed that all our vias have less than 0.28 mA flowing through them. Except for possible the *metal1* to *metal5*/*metal6* vias, so in order to reduce the current in these connections, many vias were added.

4.6 Entire 4-bit Layout

The 4-bit layout is not an entirely custom design. Each bit-slice is custom designed, and then copied and pasted to the subsequent bit slice. All the odd bit-slices were flipped about a horizontal axis. This allowed the n-wells of each bit-slice to be joined, thus saving some area. The main savings came from the joining of the V_{DD} and V_{SS} metal lines on the *metal1* layer. This allowed a 50% reduction in the number of V_{DD} and V_{SS} lines, and a reduction in area. The entire 4-bit layout is shown in Figure 19.

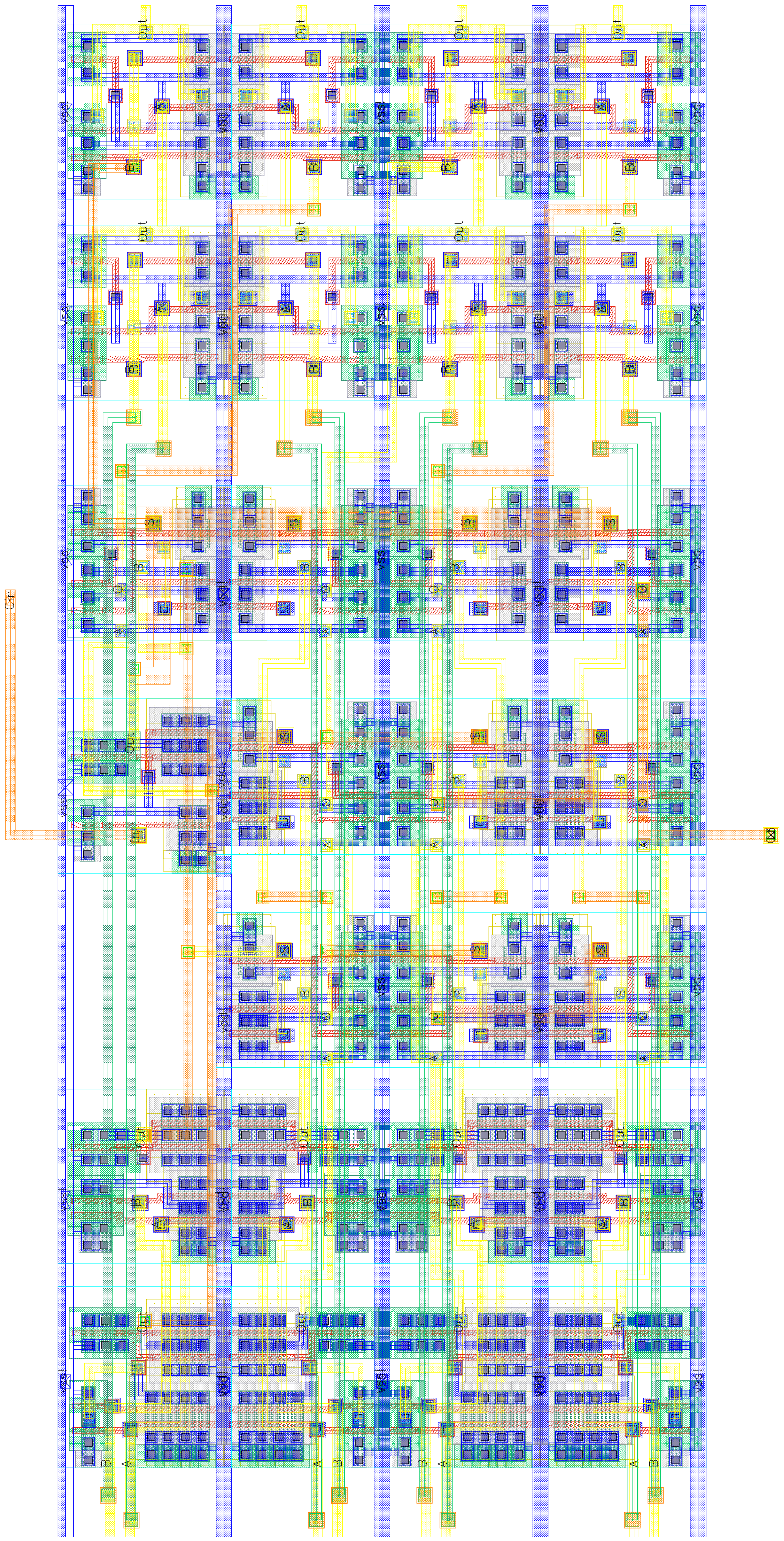


Figure 19: Entire 4-bit adder layout (V_{DD} and V_{SS} lines are omitted due to space constraints)

4.7 Entire 16-bit Layout

The 16-bit layout can easily be formed just by cascading the 4-bit modules together and connecting their C_{out} and C_{in} lines together. This can be done by simply copying and pasting. This makes it easy for designers to make higher-bit layouts. The entire 16-bit layout is shown in Figure 20.

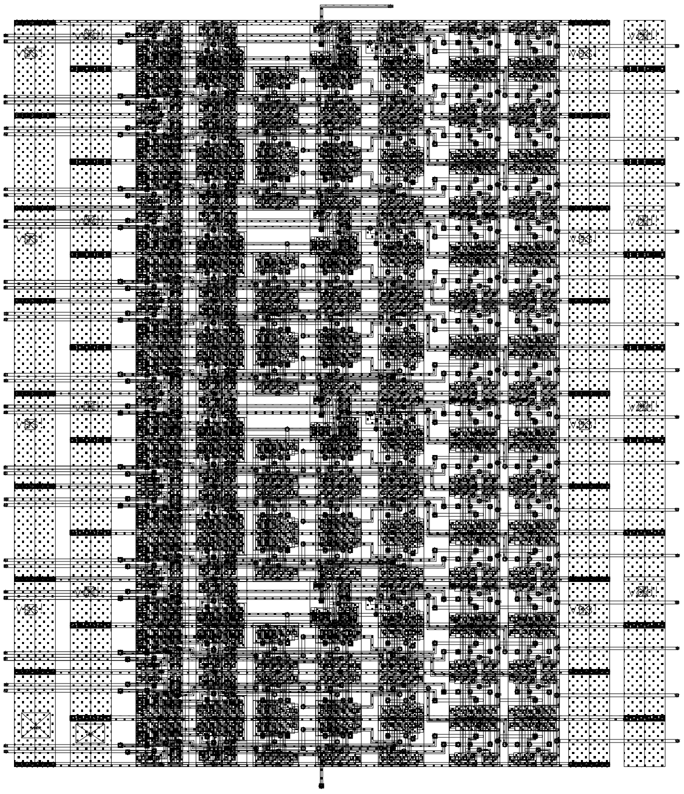


Figure 20: Entire 16-bit adder layout (with V_{DD} and V_{SS} lines shown)

4.8 Recommendations for Future Layout

These are some recommendations. They may include things that could be optimized better, if the layout were to be done again, or things which were unintentionally or intentionally neglected in the present layout.

Vias/contacts There is one thing that the authors of this report did not take into consideration, the reliability and the resistance associated with vias and contacts. When connecting most metal lines by vias, usually only one contact was used. However, if this design were being sent to CMC for processing, it is necessary to add as many vias as possible between interconnecting metal layers. Sometimes vias do not get processed properly and so it is wise to have a "backup" via. Also, adding more vias reduces the resistance of the particular connection. According to the design rules document for cmosp18, ". . . If space permits, it is preferable to have more contacts or vias than required by EM rules. This will reduce interconnect resistance and also improve reliability. Avoid using only one contact or via in one metal line unless it is absolutely necessary and allowed by rules" [5].

5 Simulation Results

Using the 4-bit adder blocks, the CCS adders with various widths (up to 32-bit) have been constructed and simulated using spectre SPICE models (in Cadence design environment). The parameters from both schematic and extracted versions have been collected. To summarize, the 4-bit and 32-bit CCS adders have the following characteristics, as shown in table 4.

Table 4: Summary of Parameters for Adder Under Consideration

Architecture	Conditional Carry Select
Circuit	TG-based CCS CLA
Technology	0.18 μm
Supply Voltage	1.8 V
Worst Case Delay	0.50 ns(4-bit), 1.10 ns(16-bit), and 1.22 ns(32-bit)
Power Consumption	37 μW(4-bit), 110 μW(16-bit) and 197μW(32-bit)
Lowest Voltage Supply	0.8 V
Area	820 μm(4-bit), 3150 μm(16-bit), and 6500 μm(32-bit)

The worst-case delay happens when one input of the adder is all ones and another input's least significant bit is one (for example, in the 16-bit case, A = 0xFFFF, B = 0x0001, C = '0') so that the input carry needs to propagate through all stages of the adder. With 0.18 μm technology under supply voltage of 1.8 V, the worst delay for the 4-bit, 16-bit, and 32-bit adders (with output buffered) are plotted in Figure 21, 22, 23, and 24. The average propagation delay, the average power consumption, the PDP and the EDP are computed as follows:

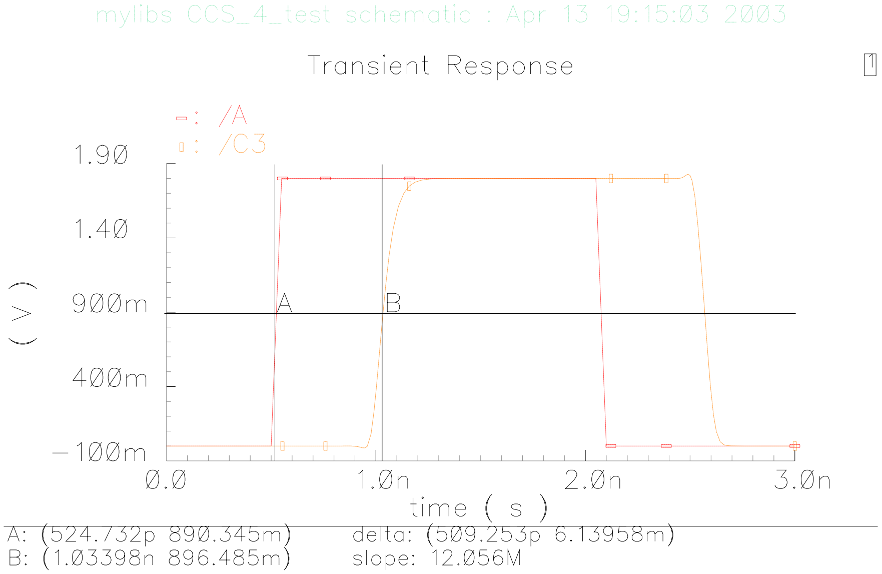
$$t_p = \frac{T_{PLH} + t_{PHL}}{2} \tag{26}$$

$$P_{av} = \frac{V_{DD}}{T} \int_0^T idt \tag{27}$$

$$PDP = P_{av} \cdot t_p \tag{28}$$

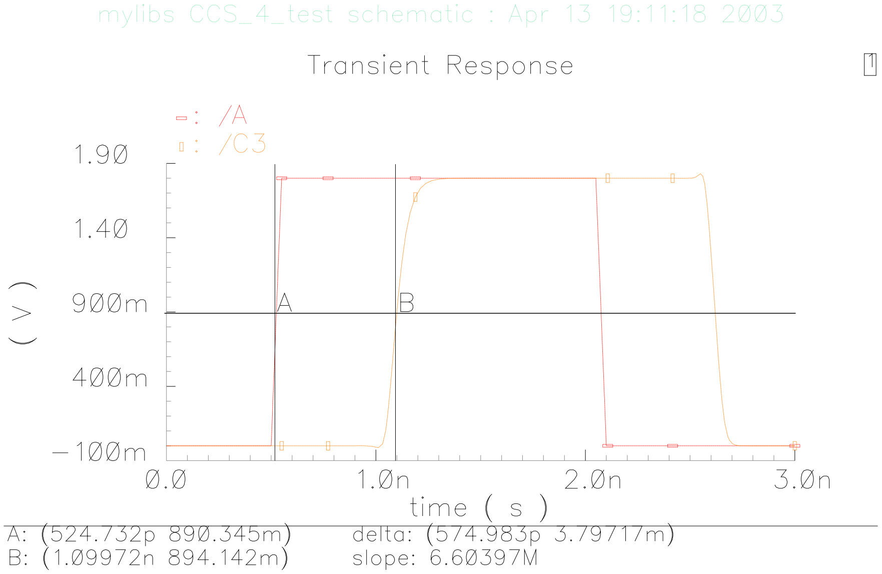
$$EDP = PDP \cdot t_p \tag{29}$$

The extractor in CMCs 0.18 μm technology can only extract parasitic capacitances, or para-



```
A: (524.732p 890.345m)      delta: (509.253p 6.13958m)
B: (1.03398n 896.485m)      slope: 12.056M
```

Figure 21: The worst delay of 4-bit CCS adder (schematic).



```
A: (524.732p 890.345m)      delta: (574.983p 3.79717m)
B: (1.09972n 894.142m)      slope: 6.60397M
```

Figure 22: The worst delay of 4-bit CCS adder (extracted).

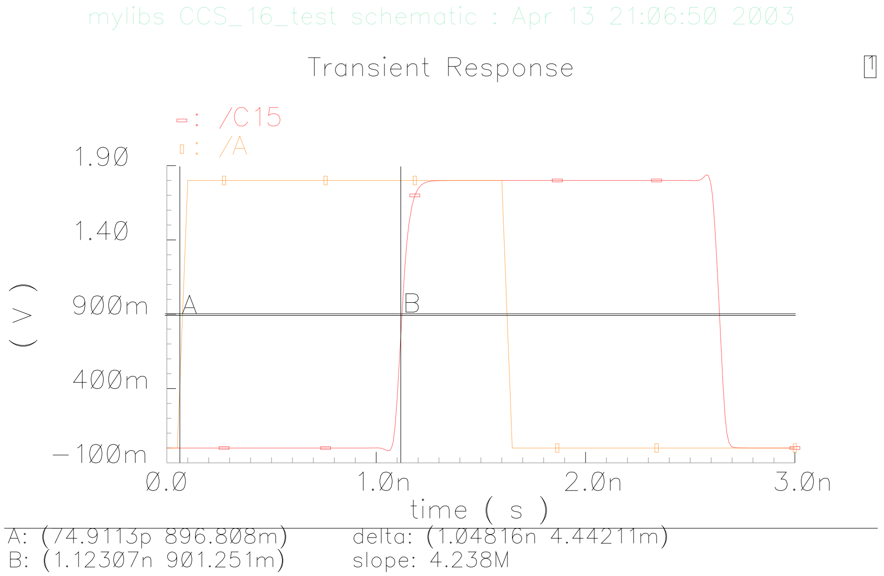


Figure 23: The worst delay of 16-bit CCS adder (extracted).

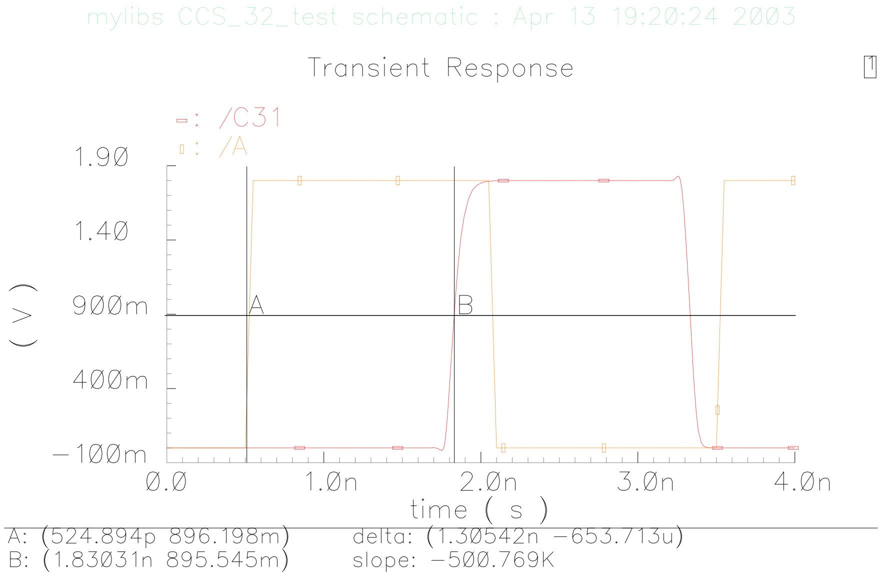


Figure 24: The worst delay of 32-bit CCS adder (schematic).

sitic resistances, but not both at the same time. We used extraction of parasitic capacitances only.

To demonstrate the overall performance trade-off between design metrics such as speed and delay, the power consumption was also measured under various supply voltages. The smallest supply voltage without causing the adder circuit to malfunction is 0.8 V, which is close to the sum of threshold voltages of PMOS and NMOS as expected. The PDP and EDP, which can be used to measure the power efficiency and the energy efficiency of the design, are also calculated and are shown in Table 5 and are plotted in Figures 25 and 26.

Supply (V)	4-bit CCS Adder				32-bit CCS Adder			
	Delay (ps)	Power (W)	PD (10^{-14}J)	ED (10^{-23}Js)	Delay (ns)	Power (W)	PD (10^{-13}J)	ED (10^{-22}Js)
1.8	463	34.1	1.57883	0.730998	1.22	177	2.1594	2.634468
1.7	496	30.1	1.49296	0.740508	1.31	155	2.0305	2.659955
1.6	538	26.4	1.42032	0.764132	1.41	135	1.9035	2.683935
1.5	587	22.9	1.34423	0.789063	1.54	120	1.848	2.84592
1.4	660	19.7	1.3002	0.858132	1.69	99	1.6731	2.827539
1.3	753	16.8	1.26504	0.952575	1.94	84	1.6296	3.161424
1.2	878	14.2	1.24676	1.094655	2.25	70	1.575	3.54375
1.1	1050	11.8	1.239	1.30095	2.7	58	1.566	4.2282
1	1340	9.6	1.2864	1.723776	3.4	46	1.564	5.3176
0.9	1790	7.7	1.3783	2.467157	4.55	37	1.6835	7.659925
0.8	2830	6.0	1.698	4.80534	6.76	28	1.8928	12.79533

Table 5: Simulation results.

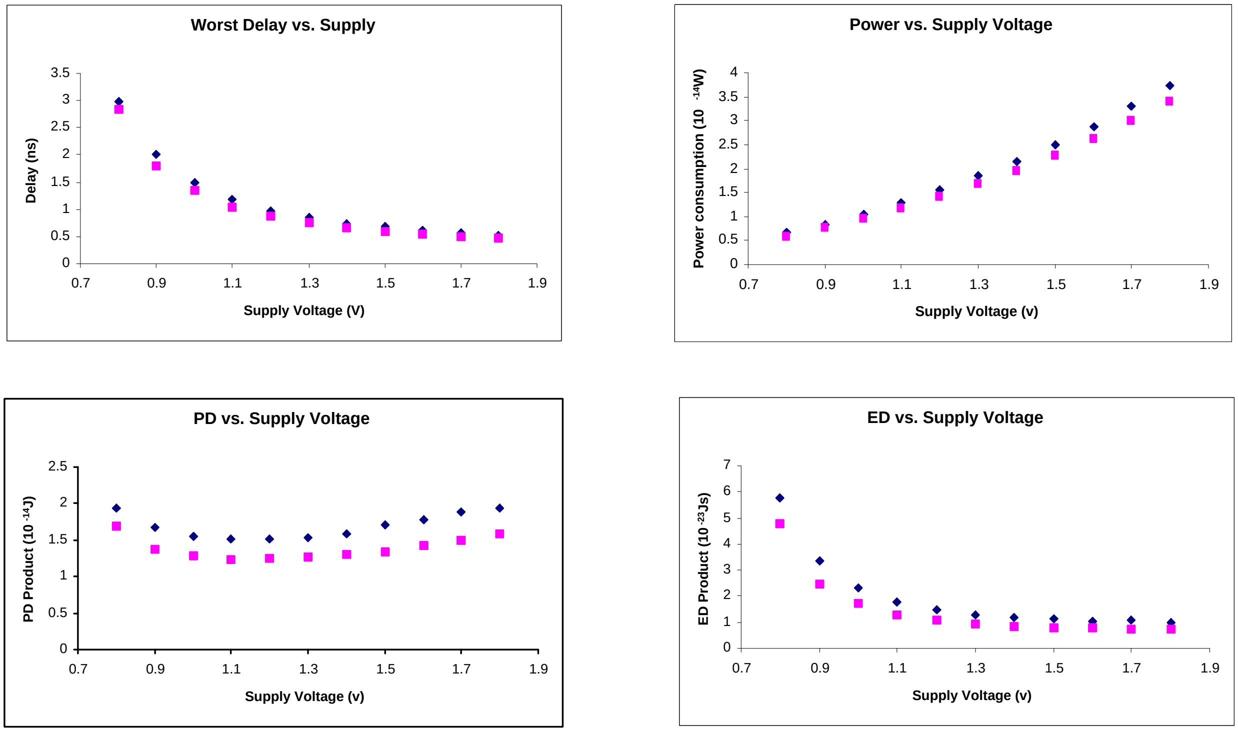


Figure 25: 4-bit CCS adder simulation parameters.

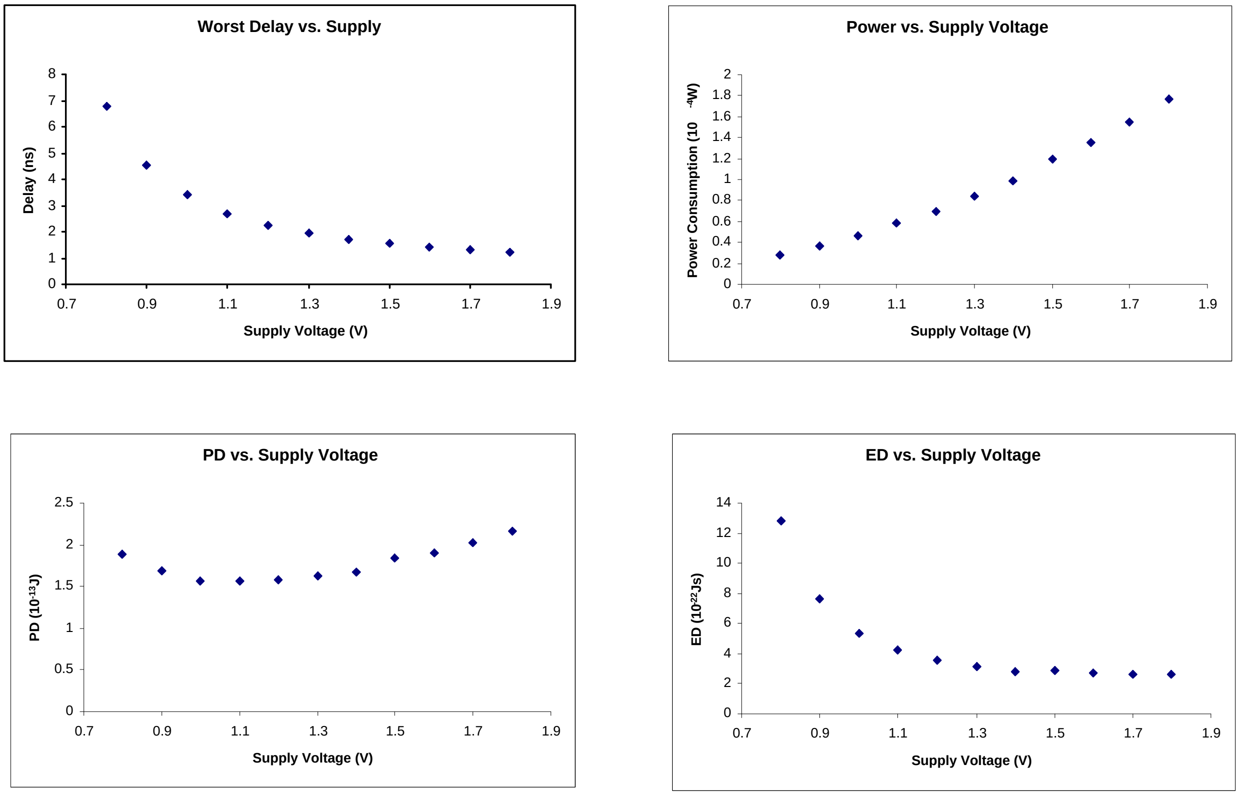


Figure 26: 32-bit CCS adder simulation parameters.

6 Conclusion

In this project, the CCS adder had been designed using static CMOS techniques (both complementary CMOS and Transmission Gate logic). The transmission-gate based carry-selection blocks (CLA_MUXs and CCS_MUXs) have the following advantages:

- The output of one stage is connected to the gate of the next stage instead of the source/drain in the conventional carry lookahead adder circuit (such as Manchester-chain CLA circuit). Therefore, the effective capacitance is reduced and the speed is increased. [4]

- There is no series critical path involved in this circuit as in Manchester-chains, therefore it is especially usefully for low-power consumption.

- The carry-select architecture can achieve better speed optimization when the width of the adder grows. As it shown, the delay of the 32-bit CCS adder is the sum of the delay of 16-bit CCS adder and the additional delay of a signal MUX.

- The static CMOS architecture has higher noise margin and thus is more robust against supply-voltage scaling. The smallest value of the supply voltage is the 0.8 V, which is close to the sum of V_{tn} and V_{tp} in 0.18 μm technology.

The adder is realized in 0.18 μm technology. The schematic and after-layout extracted simulation suggests that the adder has better overall performance and a good balance between high-speed, low-power consumption, and area efficiency. With 1.8 V supply voltage, the worst delays of the 4-bit, 16-bit, and 32-bit adder are 0.50 ns, 1.10 ns, and 1.22 ns respectively. The total power consumption of the 32-bit adder is 200 μW.

In conclusion, the CCS adder balances well the speed-performance and low-power consumption. It is therefore suitable for low-power application with relative high-speed requirement; however, for the extremely high-speed applications, the alternative dynamic logic should be used.

References

[1] Abdellatif Bellaouar and Mohamed I. Elmasry. *Low-Power Digital VLSI Design*. Kluwer Academic Press, Boston, 1995.

[2] Jan M. Rabaey. *Digital Integrated Circuits*. Prentice Hall Electronics and VLSI Series. Prentice Hall, New Jersey, 1996.

[3] Reto Zimmermann and Wolfgang Fichtner. Low-power logic styles: CMOS versus padd-transistor logic. *IEEE Joural of Solid State Circuits*, 32(7):1079–1090, July 1997.

[4] Yen-Mou Huang and James B. Kuo. A high-speed conditional carry select (CCS) adder circuit with a successively incremented carry number block (SICNB) strucure for low-voltage VLSI implementation. *IEEE Transactions on Circuits and Systems*, 47(10):1075–1079, October 2000.

[5] TSMC 0.18 μm logic 1P6M salicide 1.8 V/3.3 V design rule. PDF, May 2001. `file:/CMC/kits/cmosp18/doc/CMOSP18designRulesLogic.pdf`.