Comparisons Between Ripple-Carry Adder and Carry-Look-Ahead Adder
Comparing Propagation Delays and Power Dissipation on CMOS-simulated Circuits in HSPICE

Tsung-Han Sher
Viterbi School of Engineering
University of Southern California
Los Angeles, California, USA
tsher@usc.edu

Shermin Arab
Viterbi School of Engineering
University of Southern California
Los Angeles, California, USA
arab@usc.edu

Abstract — A four-bit adder is simulated using HSPICE in two classic design methods: a ripple-carry adder (RCA) and a carry-look-ahead adder (CLA). All components of the adders are composed of PMOS, NMOS, and capacitors. The propagation delay and power dissipation were measured under different VDD values and different operating temperatures in HSPICE. A comparison of these two metrics were analyzed between the RCA and CLA, and concludes that both the CLA and RCA operates optimally at low temperatures and are unsuitable for high VDD environments, CLA is preferred for faster computation while RCA is preferred for smaller chip size, and the CLA is preferred for high-frequency adder usage situations, and the CLA is more suitable for low-VDD systems.

Index Terms—CMOS, HSPICE, Ripple-Carry Adder, RCA, Carry-Look-Ahead Adder, CLA, Power Dissipation, Propagation Delay

I. INTRODUCTION

THE adder is a central component of a central processing unit of a computer. One of the main considerations of designing a digital circuits is the trade-off between size, performance speed, and power consumption. The RCA and CLA have differing designs, and thus different benchmarks for the above stated characteristics. This paper will be using HPICE to simulate and analyze the propagation delay and power dissipation of the RCA and CLA to explore and conclude which adder design is optimal under certain situations.

II. CMOS COMPONENTS DESIGN

Each transistor of the adders are composed of logic gates, and each logic gate is composed of CMOS components. Each gate used in this paper’s design has been implemented at the CMOS level in HSPICE.

A. Design Constant Parameters

Each logic gates used in the adder design are composed of PMOS’s, NMOS’s, and capacitors. The major parameter constants are as follows: The length of all CMOS components are 45 nano-meters; the width of all NMOS transistors are 180 nano-meters; the width of all PMOS transistors are 360 nano- meters. The capacitors in the RCA are set at 0.1 pico-farads; the capacitors in the CLA are set at 10 femto-farads. The CMOS is simulated using a Predictive Technology Model of a high performance 45 nano-meters metal gates. A full list of the parameters used for this paper’s CMOS design is given in Appendix A.

B. Inverter

The inverter is designed using one NMOS and one PMOS. Refer to Appendix B for the schematic of the inverter.

C. NAND and AND Gates

For an n-input NAND gate, the gate is designed using n NMOS’s and n PMOS’s. Each PMOS is connected in series and each NMOS is connected in parallel. Each gate input line is connected to one PMOS and one NMOS. The AND gate is implemented by adding an inverter at the output of the NAND gate. The full schematic of the 2-input NAND and AND gate is given in Appendix B.

D. NOR and OR Gates

For an n-input NOR gate, the gate is designed using n NMOS’s and n PMOS’s. Each PMOS is connected in parallel and each NMOS is connected in series. Each gate input is connected to one PMOS and one NMOS. The OR gate is implemented by adding an inverter at the output of the NOR gate. The full schematic of the 2-input NOR and OR gate is given in Appendix B.

E. XOR Gate

Both the RCA and CLA uses a two-input XOR gate. The XOR gate is designed using four NMOS’s and four PMOS’s. Each input to the NMOS or PMOS is either the gate-input itself or the gate-input inverted via an inverter. Gate input combinations that are supposed to yield a high signal are put on the PMOS side, and input combinations that are supposed to yield a low singer are put on the NMOS side. The full schematic of the XOR gate is given in Appendix B.

III. RIPPLE-CARRY ADDER DESIGN

The sub-units of an RCA is a full-adder. A full-adder takes in two one-bit numbers as well as a carry-in bit, and outputs the corresponding one-bit sum as well as a carry-out bit.
A. Half-Adder

The half-adder is a sub-component of the full-adder. The half-adder has two inputs: two one-bit digits denoted as $A_H$ and $B_H$; the half-adder has two outputs: a one-bit sum denoted as $S_H$ and a one-bit carry-out denoted as $C_H$. The outputs are generated using an XOR gate and an AND gate (1). The full schematic of the half-adder is given in Appendix B.

$$S_H = A_H \oplus B_H$$

$$C_H = A_H \cdot B_H$$

B. RCA Full-Adder

The full-adder has three inputs: two one-bit digits denoted as $A_{RF}$ and $B_{RF}$ and a one-bit carry-in denoted as $CIN_{RF}$; the full-adder has two outputs: a one-bit sum denoted as $S_{RF}$ and a one-bit carry-out denoted as $COUT_{RF}$. The outputs are generated using a half-adder, an XOR gate, an AND gate, and an OR gate (2). The full schematic design of the full-adder is given in Appendix B.

$$A_H = A_{RF}$$

$$B_H = B_{RF}$$

$$S_F = S_H \oplus CIN_{RF}$$

$$COUT_{RF} = COUT_H + (S_H \cdot CIN_{RF})$$

C. Four-Bit RCA

The four-bit RCA has nine inputs: two four-bit binary numbers denoted as $A_{RF}[3:0]$ and $B_{RF}[3:0]$ along with a one-bit carry-in denoted as $CIN_{RF}$; the RCA has five outputs: a four-bit sum denoted as $S_{RF}[3:0]$ along with a one-bit carry-out denoted as $COUT_R$. The RCA is constructed by concatenating multiple full-adders in series by connecting the carry-out bit of one full-adder to the next successive full-adder’s carry-in bit. The carry-in bit of the RCA is connected to the carry-in line of the least significant full-adder’s carry-in bit; similarly, the carry-out bit of the RCA is connected to the carry-out line of the most-significant full-adder’s carry-out bit. Capacitors with a capacitance of 0.1 pico-farads are connected to each of the sum-bits as well as the carry-out bit. Refer to Fig. 1 for the outline schematic of the RCA. The full HSPICE implementation of the four-bit RCA is given in Appendix C.

D. RCA Correctness

To test for the correctness of the implemented RCA, two test cases were run for correctness. Table I displays the test cases, expected results, and the actual results from the RCA adder simulated in CosmosScope. The CosmosScope waveform results of the test cases are given in Appendix D.

### IV. CARRY-LOOK-AHEAD ADDER DESIGN

For the CLA, the carry-in bits of each adder component do not have wait for the previous adder to compute the carry-out bit. This can be achieved by determining each carry-in bit from the cumulation of the CLA’s inputs as well as the previous calculations in determining the other carry-in bits. More formally, instead of producing a carry-out bit for each adder, we produce a generating and propagating carries used to determine the successive carry-in bits.

A. Generating Carry

The generating carry for the $i^{th}$ bit is produced from the AND of the two $i^{th}$ input of the adder (3).

$$G_i = A_i \cdot B_i$$

B. Propagating Carry

The propagating carry for the $i^{th}$ bit is produced from the OR of the two $i^{th}$ input of the adder (4).

$$P_i = A_i + B_i$$

C. CLA Carries

The CLA’s $i+1^{th}$ carry-in is generated by (5). Note that the $i^{th}$ carry expand into the previous bit’s carries. The carry-in for the least significant bit is $CIN$.

$$C_{i+1} = G_i + (P_i \cdot C_i)$$

Thus for the four-bit CLA implemented in this paper, the carry-in bits for the CLA are generated using equation (6).

$$CIN_0 = CIN$$

$$CIN_1 = G_0 + (P_0 \cdot C_0)$$

$$CIN_2 = G_1 + (P_1 \cdot C_1)$$

$$CIN_3 = G_2 + (P_2 \cdot C_2)$$

![Schematic layout of a RCA demonstrating the concatenation of the full adders](image-url)
D. CLA Full-Adder

The CLA’s full-adder has three inputs: two one-bit digits denoted as $A_{CF}$ and $B_{CF}$ and a one-bit carry-in denoted as $CIN_{CF}$; the full-adder has three outputs: a one-bit sum produced by the XOR of $A_{CF}$, $B_{CF}$, and $CIN_{CF}$ denoted as $S_{CF}$, a generating carry by using (4) denoted as $G_{CF}$, and a propagating carry by using (5) denoted as $P_{CF}$ (7). The full schematic of the CLA is given in Appendix B.

\[
S_{CF} = A_{CF} \oplus B_{CF} \oplus CIN_{CF} \tag{7}
\]

\[
G_{CF} = A_{CF} \cdot B_{CF}
\]

\[
P_{CF} = A_{CF} + B_{CF}
\]

E. Fout-Bit CLA

The four-bit CLA has nine inputs: two four-bit binary numbers denoted as $A\{3 : 0\}$ and $B\{3 : 0\}$ along with a one-bit carry-in denoted as $CIN\{3 : 0\}$; the CLA has five outputs: a four-bit sum denoted as $S\{3 : 0\}$ along with a one-bit carry-out denoted as $COUT\{3 : 0\}$. Each generating and propagating carries from each full-adder contribute to generate the successive carry-in bits. Capacitors with a capacitance of 10 femto-farads from each full-adder contribute to generate the successive carry-in bits. Refer to Fig. 2 for the outline schematic design. The full HSPICE implementation of the four-bit CLA can be found in Appendix C.

F. CLA Correctness

To test for the correctness of the implemented CLA, two test cases were run for correctness. Table II displays the test cases, expected results, and the actual results from the CLA adder simulated in CosmosScope. The CosmosScope waveform results of the test cases are given in Appendix D.

Table II
CLA Correctness Test Cases and Results

<table>
<thead>
<tr>
<th>Test</th>
<th>Input</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111</td>
<td>0001</td>
<td>0000</td>
</tr>
<tr>
<td>2</td>
<td>1110</td>
<td>0001</td>
<td>1111</td>
</tr>
</tbody>
</table>

V. PROPAGATION DELAY

In order to analyze the propagation delay differences between the RCA and the CLA, the propagation delays of the critical path of both designs will be used for the analysis. The critical path is from the $CIN$ to the $COUT$ for both four-bit adders. Both low-to-high ($t_{PLH}$) and high-to-low ($t_{PHL}$) propagation delays will be analyzed. Propagation delay will be measured at temperatures 0 Celsius, 25 Celsius, and 90 Celsius, as well as at voltage supply values 0.6 volts, 0.9 volts, and 1.8 volts.

A. Low-to-High Propagation Delay

To trigger a low-to-high signal change in $COUT$, input $A\{3 : 0\}$ is set at 1111 and $B\{3 : 0\}$ is set at 0000, and the $CIN$ signal will change from 0 to 1. This way the sum changes from 1111 to 0000 and the $COUT$ from 0 to 1. Table III presents the findings of the low-to-high propagation delay results.

B. High-to-Low Propagation Delay

To trigger a high-to-low signal change in $COUT$, input $A\{3 : 0\}$ is set at 1111 and $B\{3 : 0\}$ is set at 0000, and the $CIN$ signal will change from 1 to 0. This way the sum changes from 0000 to 1111 and the $COUT$ from 1 to 0. Table IV presents the findings of the propagation delay results.

C. Propagation Delay Estimation

The estimation of propagation delay ($t_p$) is the average of the low-to-high and high-to-low propagation delay(8).

\[
t_p = \frac{t_{PLH} + t_{PHL}}{2} \tag{8}
\]

Table V displays the propagation delay estimations of both the RCA and CLA in varying operating temperatures and $VDD$ values.
VI. POWER DISSIPATION

Two main types of power dissipation are measured for this paper: static and dynamic power dissipation. The power dissipation will be measured with the product of the average current and the supply power \( P = I_{avg} \cdot V_{DD} \) (9).

A. Dynamic Power Dissipation

Dynamic power dissipation \( (P_D) \) is a measurement of the average power dissipated over the time frame in which a signal rises or falls. Thus the time frame used to measure the average current will be between the start of a signal change, denoted as \( T_1 \), and the end of the signal change, denoted as \( T_2 \). Equation (10) describes the measurement of dynamic power dissipation.

\[
P_D = \frac{\sum_{t=T_1}^{T_2} I(t)}{T_2 - T_1} \cdot V_{DD}
\]

(10)

Determining \( T_1 \) and \( T_2 \) are done by analyzing the waveform in CosmosScope manually. Since it is not possible to determine the exact moment a signal change occurs, the time frame used for the final calculation will include an arbitrary small unit of time \( \epsilon \) to the left of \( T_1 \) and to the right of \( T_2 \) in order to ensure the entirety of the signal change is encapsulated in the calculation (11). For this paper, the value of \( \epsilon \) will be 0.01 micro-seconds.

\[
P_D = \frac{\sum_{t=T_1-\epsilon}^{T_2+\epsilon} I(t)}{(T_2 + \epsilon) - (T_1 - \epsilon)} \cdot V_{DD}
\]

(11)

Table V and Table VI displays the low-to-high \( (P_{DLH}) \) and high- to-low \( (P_{DHL}) \) dynamic power dissipation of the \( COUT \) signal in both the RCA and CLA.

B. Static Power Dissipation

Static power dissipation \( (P_S) \) is the measurement of the average power dissipated of an unchanging signal. Since the signal is unchanging, the time frame in which to measure static power dissipation can be chosen arbitrarily. In this paper the time frame will be the exact same length as those used to measure the dynamic power dissipation in order to open the potential of observations between dynamic and static power dissipation. Table VII and Table VIII displays the static power dissipation of both static-high \( (P_{SH}) \) and static-low \( (P_{SL}) \) voltage of the \( COUT \) signal in both the RCA and CLA.

C. Power Dissipation Estimation

The estimation of dynamic power dissipation is the average of the low-to-high and high-to-low power dissipation (12).

\[
P_D = \frac{P_{DLH} + P_{DHL}}{2}
\]

(12)

TABLE V

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>1.079</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.358</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>0.185</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>0.467</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.158</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>0.109</td>
</tr>
</tbody>
</table>

TABLE VI

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>2.153</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>5.207</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>407.5</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>0.535</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.780</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>595.1</td>
</tr>
</tbody>
</table>

TABLE VII

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>7.712</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>19.31</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>613.8</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>1.251</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>5.224</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>766.2</td>
</tr>
</tbody>
</table>

TABLE VIII

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>8.843</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>7.982</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>340900</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>15.63</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>46.09</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>541400</td>
</tr>
</tbody>
</table>

TABLE IX

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>2.892</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>49.20</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>387900</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>12.16</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>94.85</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>529900</td>
</tr>
</tbody>
</table>
A. Propagation Delay

The propagation delay between a signal change in \( CIN \) and the corresponding change of the \( COUT \) signal are affected in different ways by the temperature and value of \( VDD \).

As the temperature increases, each transistor will experience a higher intrinsic resistance value, and therefore possibly result in the slower propagation delay time as the temperature increases. The exception to this trend is when \( VDD \) has a value of 1.8 volts. At this high of a voltage, changes in temperature seems to keep the propagation delay at the same relative value.

As the value of \( VDD \) increases, it has two effects on the propagation delay: (i) decrease the propagation delay; and (ii) decrease the change of propagation delay when the temperature varies. Once again the exception to this trend is when the \( VDD \) has a value of 1.8 volts. This may be due to the size of the CMOS transistors and it not being able to handle such high of a voltage since during HSPICE simulations there were warnings indicating a \( VDD \) value of 1.8 volts may cause unexpected behaviors.

The CLA has a notably less propagation delay than the RCA. In each corresponding comparison of the same temperature and \( VDD \) value, the CLA’s performance speed trumps the RCA’s. This is due to the significantly shorter critical path of the CLA compared to the critical path of the RCA. However the CLA’s high performance comes at the cost of a larger chip size due to the extra logic needed to be implemented with the generating and propagating carries.

B. Dynamic Power Dissipation

Both the RCA and the CLA increases in power dissipation linearly at a small rate as the temperature increases. However an increase to the \( VDD \) value will exponentially increase the power dissipation. From 0.6 volts to 1.8 volts there is an increase of two orders of magnitude. This applies to both the low-to-high and high-to-low dynamic power dissipation.

The CLA has notably less power dissipation for a \( VDD \) value of 0.6 and 0.9 volts. At 1.8 volts the CLA has a higher dissipation value. This means for low-\( VDD \) systems using the CLA benefits in both power dissipation and performance.

C. Static Power Dissipation

Both the RCA and CLA increases in power dissipation increases at a slight exponential as temperature increases. This increase is significantly greater than the increase dynamic power dissipation experiences as temperature changes. For increase in the \( VDD \) value, there is a great exponential increase in power dissipation as the \( VDD \) value increases. Between 0.6 and 1.8 volts, there is a five order of magnitude difference in the dissipation values.

At a \( VDD \) value of 1.8 volts, the static power dissipation value of both the RCA and the CLA are relatively similar. For a \( VDD \) value of 0.6 and 0.9 volts, RCA has considerably less power dissipation.

VII. DISCUSSION

After implementing and running simulations of the RCA and CLA in HSPICE, a number of notable conclusions can be made from the data results. These are as follows:

A. Propagation Delay

Table X and Table XI displays the power dissipation estimations of both the RCA and CLA in varying operating temperatures and \( VDD \) values.

VIII. CONCLUSIONS

From the analysis of the data and discussion above, this paper makes five main conclusions: (i) both the RCA and CLA operates optimally at lower temperatures; (ii) both the RCA and the CLA are unsuitable for high \( VDD \) value environments due to the unpredictable propagation delay as well as the immense power dissipation at this \( VDD \) value; (iii) if performance in the key concern the CLA is the more suited architecture due to the better performance in propagation delay; (iv) if the implementation using the adder is expecting a large number of computations, then the CLA is recommended due to the lower dynamic power dissipation for both low-to-high and high-to-low, however if computation tasks are infrequent, then the RCA is recommended due to the lower static power dissipation for both static-low and static-high; and (v) the CLA is more suitable for low-\( VDD \) systems due to the lower power dissipation and faster performance.

ACKNOWLEDGMENTS

T.S. would like to thank Rebecca Lee for the guidance on using and analyzing CMOS systems in HSPICE and CosmosScope.

---

### Table X

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celsius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>4.933</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>12.26</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>510.7</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>0.895</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>3.500</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>680.7</td>
</tr>
</tbody>
</table>

### Table XI

<table>
<thead>
<tr>
<th>Adder</th>
<th>VDD (Volts)</th>
<th>Temperature (Celsius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>RCA</td>
<td>0.6</td>
<td>5.870</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>28.60</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>364400</td>
</tr>
<tr>
<td>CLA</td>
<td>0.6</td>
<td>13.90</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>70.50</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>535650</td>
</tr>
</tbody>
</table>

The CLA has a notably less propagation delay than the RCA. In each corresponding comparison of the same temperature and \( VDD \) value, the CLA’s performance speed trumps the RCA’s. This is due to the significantly shorter critical path of the CLA compared to the critical path of the RCA. However the CLA’s high performance comes at the cost of a larger chip size due to the extra logic needed to be implemented with the generating and propagating carries.
APPENDIX A

HSPICE SIMULATION PARAMETERS

The following is the parameter file used for the HSPICE simulations used in this paper:

```
.model nmos nmos level = 54
+version = 4.0 binunit = 1 paramchk= 1 mobmod = 0
+capmod = 2 lgbmod = 1 igbmod = 1 geomod = 1
+diomod = 1 rdsmod = 0 rbmmod= 1 rgateomod= 1
+permol = 1 acngsm= 0 trngsm= 0 
+tnom = 27 tox = 1.25e-009 toxp = 1e-009 toxm = 1.25e-009
+dtox = 2.5e-010 eproxx = 3.9 wint = 5e-009 lint = 3.75e-009
+l1 = 0 w1 = 0 llm = 1 wln = 1
+l2 = 0 w2 = 0 lwm = 1 wwn = 1
+x1 = -20e-9

+kth0 = 0.46893 k1 = 0.4 k2 = 0 k3 = 0
+k3b = 1 w0 = 2.5e-006 dvt0 = 1 dvt1 = 2
+dvtl = 1 dvt2 = 0 dvtlw = 0 dvtl = 0
+dsub = 0.1 minv = 0.05 voffl = 0 dvtl = 0
+dvtp = 0.1 lpe0 = 0 lpeb = 0 xj = 1.4e-008
+ngate = 1e+023 ndep = 3.24e+018 nsd = 2e+020 phin = 0
+nsrc = 0 cdscb = 0 cscd = 0 cit = 0
+pox = 0.12 u = 0.054 ua = 6e-010 ub = 1.2e-018
+uc = 0 vsat = 170000 a0 = 1 ags = 0
+a1 = 0 a2 = 0 b1 = 0 plm = 0.02
+keta = 0.04 dwg = 0 dwb = 0 
+pdiblcl = 0.001 pdiblc2 = 0.001 pdiblc3 = -0.005 drout = 0.5
+pvag = 1e-020 delta = 0.01 pscbrel = 8.14e+008 pscbrel2 = 1e-007
+prout = 0.2 pdits = 0.08 pdis = 0.23 pdits1 = 2300000
+rsx = 5 rdw = 155 rsw = 80 rdx = 80
+rdwmin = 0 rdwmin = 0 rdwmin = 0 
+prw = 0 wr = 0 0.074 wr = 0
+beta0 = 30 agid = 0.0002 bgid1 = 2.1e+009 cgid = 0.0002
+egid = 0.8 aigbb = 0.012 bigbb = 0.0028 cibbb = 0.0002
+bigb = 1 eibbb = 0 bigbb = 0.004 cigbb = 0.0004
+biggb = 1.1 eibgb = 0 biggb = 0.002 biggb = 0.0025
+biggb = 0.002 biggb = 0.002 biggb = 0.0002
+eigc = 1 eicgc = 1 eicgc = 1 
+nigc = 1 eicgc = 1 eicgc = 1
+xicc = 0 eicgc = 1 eicgc = 1
+xrcrc = 12 ercrg = 5
+cqso = 1.1e-010 cgo = 1.1e-010 cgo = 2.56e-011 cgd = 2.65e-010
+cpp = 2.653e-010 ckkab = 0.03 ckkpad = 0.03 acce = 1
+cmo = 15 noff = 0.9 voffcv = 0.02
+ktl = -0.11 ktl = 0 ktl = 0.022 ute = -1.5
+ual = 4.31e-009 ub1 = 7.61e-018 uc1 = -5.6e-011 prt = 0
+at = 33000

+fnoimod = 1 toimod = 0
+jss = 0.0001 jssw = 1e-011 jswg = 1e-010 
+lthswd = 0.01 jthwrev = 0.001 jtsw = 1e-011 jsbw = 1
+jsd = 0.0001 jsbw = 1e-011 jsbw = 1e-011 jsbw = 1
+lthswd = 0.01 jthwrev = 0.001 jtsw = 1e-011 jsbw = 1
+pbs = 1 cjs = 0.0005 mjs = 0.5 pbsws = 1
+csjs = 0.33 mjsws = 1 csjs = 3e-010
+pmjs = 0.33 pbsd = 1 cjd = 0.0005 mjd = 0.5
+pmb = 1 cjsd = 5e-010 mjswd = 0.33 pbsd = 1
+pmb = 0.085 tcsjd = 0.001 tcsjd = 0.001
tscp = 0.005 tscp = 0.005
tscp = 0.005 tscp = 0.005
tscp = 0.005 tscp = 0.005
+xst = 3 xtid = 3
+dnmcg = 0 dmcg = 0 dmcg = 0
+dwj = 0 dwj = 0 dwj = 0
+rsh = 0.4 gbmin = 1e-010 rbpp = 5 
+rbps = 15 rbpp = 15 rbsb = 15 ngcon = 1
.model pmos pmos level = 54
+version = 4.0 binunit = 1 paramchk= 1 mobmod = 0
+capmod = 2 lgbmod = 1 igbmod = 1 geomod = 1
+diomod = 1 rdsmod = 0 rbmmod= 1 rgateomod= 1
+permol = 1 acngsm= 0 trngsm= 0 
+tnom = 27 tox = 1.3e-009 toxp = 1e-009 toxm = 1.3e-009
+dtox = 3e-010 eproxx = 3.9 wint = 5e-009 lint = 3.75e-009
+l1 = 0 w1 = 0 llm = 1 wln = 1
+l2 = 0 w2 = 0 lwm = 1 wwn = 1
+x1 = -20e-9
```

+vth0 = -0.49158  k1 = 0.4  k2 = -0.01  k3 = 0
+k3b = 0  w0 = 2.5e-006  dvt0 = 1  dvt1 = 2
+dvt2 = 0.032  dvt0w = 0  dvt1w = 0  dvt2w = 0
+dsub = 0.1  minv = 0.05  vsffl = 0  dvtpp0 = 1e-011
+dvtp1 = 0.05  le0 = 0  lepb = 0  xj = 1.4e-008
+ngate = 1e+023  ndep = 2.44e+018  nsd = 2e+020  phin = 0
+cdsc = 0  cdscb = 0  cdscd = 0  cit = 0
+voff = -0.126  nfactor = 2.1  eta0 = 0.0055  etab = 0
+vfb = 0.55  u0 = 0.02  ua = 2e-009  ub = 5e-019
+uc = 0  vsat = 1500000  a0 = 1  ags = 1e-020
+a1 = 0  a2 = 1  b0 = 0  bl = 0
+keta = -0.047  pdiblc1 = 0.001  pdsb = 0.01  pclm = 0.12
+pidb1cl = 0.001  pidb2 = 0.001  pdiblc = 3.4e-008  drout = 0.56
+pvag = 1e-020  delta = 0.01  pscbel = 8.14e+008  pscbel = 9.58e-007
+efpout = 0.2  pdits = 0.08  pditrd = 0.23  pditsl = 2300000
+rsh = 5  rsd = 155  rsw = 75  rdw = 75
+rdsm = 0  rdsmin = 0  rsmin = 0  prw = 0
+prwb = 0  wr = 1  alpha0 = 0.074  alpha1 = 0.005
+beta0 = 30  agidl = 0.0002  bigd0 = 2.1e+009  cgid0 = 0.0002
+egd0 = 0.8  aigbacc = 0.012  bigbacc = 0.0028  cgbacc = 0.0002
+nibgacc = 1  aigbinc = 0.014  bigbinc = 0.004  cgbinc = 0.0004
+eigbinc = 1.1  nignb = 3  aigc = 0.010687  cgc = 0.0008
+cigc = 1  poxedge = 1  pigcd = 1  ntox = 1
+xrcrg2 = 12  xrcrg2 = 5
+cgso = 1.1e-010  cgdo = 1.6e-010  cgbo = 2.56e-011  cgdl = 2.653e-010
+cgs1 = 2.653e-010  ckgp = 0.03  ckpad = 0.03  acde = 1
+main = 15  noff = 0.9  voffcv = 0.02
+k1 = -0.11  ktl = 0  ktr = 0.022  ute = -1.5
+ual = 4.31e-009  ubl = 7.61e-018  ucl = -5.6e-011  prt = 0
+at = 33000
+fnoi = 0  tnoi = 0
+jjs = 0.0001  js = le-011  js = le-010  njs = 1
+jlth = 0.01  jlthev = 0.001  bvs = 10  xjbv = 1
+j = 0.0001  jsd = le-011  jsd = le-010  njd = 1
+jlthd = 0.01  jlthdev = 0.001  bvd = 10  xjbv = 1
+pbs = 1  cjs = 0.005  mjs = 0.5  pbs = 1
+cm = 5e-010  cm = 0.33  pbs = 1  cm = 3e-010
+m = 0.33  pbd = 1  cjd = 0.005  mj = 0.5
+pbd = 1  cm = 5e-010  mjswd = 0.33  tpc = 0.005  tcj = 0.001
+cm = 5e-010  cm = 0.33  tcp = 0.005  tcj = 0.001
+at = 33000  xtd = 3
+dmcg = 0  dmci = 0  dmcg = 0  dmcgt = 0
+dwj = 0  xgw = 0  xgl = 0
+ph = 0.4  gmin = 1e-010  rpb = 5  rpb = 15
+rbps = 15  rbdb = 15  rbs = 15  ngc = 1
APPENDIX B
CMOS AND ADDER COMPONENT SCHEMATICS

A. CMOS Implementation Schematic of an Inverter

B. CMOS Implementation Schematic of an NAND Gate
C. CMOS Implementation Schematic of an AND Gate

D. CMOS Implementation Schematic of an NOR Gate
E. CMOS Implementation Schematic of an OR Gate

F. CMOS Implementation Schematic of an XOR Gate
A. Ripple-Carry Adder Implementation in HSPICE

```plaintext
*library setup
.include "45nm_HP.pm"

*define parameters
.param Supply = 0.9V
*.define VDD supply voltage
.global VDD

*** add voltage source for VDD
.VDD VDD Gnd Supply

******************************************************************************
* Inverter
******************************************************************************
.SUBCKT INV IN OUT
.m0 OUT IN VDD VDD pmos l = 45n w = 360n
.m1 OUT IN Gnd Gnd nmos l = 45n w = 180n
.ENDS

******************************************************************************
* NAND
******************************************************************************
.SUBCKT NAND A B OUT
*** P-MOS
.m0 OUT A VDD VDD pmos l = 45n w = 360n
.m1 OUT B VDD VDD pmos l = 45n w = 360n

*** NMOS
.m2 OUT A x Gnd nmos l = 45n w = 180n
.m3 x B Gnd Gnd nmos l = 45n w = 180n
.ENDS

******************************************************************************
* AND
******************************************************************************
*** P-MOS

*** NMOS

.ENDS

******************************************************************************
* XOR
******************************************************************************
.SUBCKT XOR A B OUT
*** A' ***
.m0 A_NOT A VDD VDD pmos l = 45n w = 360n
.m2 A_NOT A Gnd Gnd nmos l = 45n w = 180n

*** B' ***
.m1 B_NOT B VDD VDD pmos l = 45n w = 360n
.m3 B_NOT B Gnd Gnd nmos l = 45n w = 180n

*** XOR ***
*** P-MOS
.m4 w B_NOT VDD VDD pmos l = 45n w = 360n
.m5 x B VDD VDD pmos l = 45n w = 360n
.m6 OUT A w VDD pmos l = 45n w = 360n
.m7 OUT A_NOT x VDD pmos l = 45n w = 360n

*** NMOS
.m8 OUT B y Gnd nmos l = 45n w = 180n
.m9 OUT B_NOT z Gnd nmos l = 45n w = 180n
.m10 y A Gnd Gnd nmos l = 45n w = 180n
.m11 z A_NOT Gnd Gnd nmos l = 45n w = 180n
.ENDS

******************************************************************************
* NOR
******************************************************************************
*** NMOS

.ENDS

******************************************************************************
* p-mos ***
```
m2  OUT  B  x  VDD  pmos  l = 45n  w = 360n
m3  x  A  VDD  VDD  pmos  l = 45n  w = 360n
.ENDS

****************************************************
* OR
****************************************************
* A -----
* | NOR  | -- x --| INV  | -- OUT
* B -----
****************************************************
.SUBCKT OR A B OUT
Xnor A B x NOR
Xinv x  OUT INV
.ENDS
****************************************************
* Half_Adder
****************************************************
* A ------|
* | | XOR  | ------ S
* B ------|
* | | ------|
* | | ------|
* | | ------| AND  | --- C
* | | ------|
****************************************************
.SUBCKT HALF_ADDER A B S C
Xxor A B S XOR
Xand A B C AND
.ENDS
****************************************************
* Full_Adder
****************************************************
* A ------|  ---S---x--- x| |
* | | | XOR  | ---S
* B ------| |     | Half_Adder | Cin |
* | | | ---C---y--- x| |
* | Cin  | | | ------| |
* | | | z| |
* | | | OR  | ---COUT
* | | | y| |
****************************************************
.SUBCKT FULL_ADDER A B Cin S COUT
Xhalf_adder A B x y HALF_ADDER
Xxor x  Cin S XOR
Xand x  Cin z AND
Xor y z  COUT OR
.ENDS
****************************************************
* 4-bit Full-Adder
****************************************************
* A0 ------|  ------| ---S0------
* B0 ------| Half_Adder  | ---COUT0--
* Cin ------|  ---S1------
* A1 ------| Half_Adder  | ---COUT1--
* COUT0------|  ---S2------
* A2 ------| Half_Adder  | ---COUT2--
* COUT1------|  ---S3------
* A3 ------| Half_Adder  | ---COUT--
* COUT2------|  ---COUT---
*
.SUBCKT RCA A0 A1 A2 A3 B0 B1 B2 B3 CIN S0 S1 S2 S3 COUT
Xfull_adder0 A0 B0 CIN S0 COUT0 FULL_ADDER
Xfull_adder1 A1 B1 COUT0 S1 COUT1 FULL_ADDER
Xfull_adder2 A2 B2 COUT1 S2 COUT2 FULL_ADDER
Xfull_adder3 A3 B3 COUT2 S3 COUT FULL_ADDER
.ENDS

******************************************************************************
Test that RCA gate is correct
******************************************************************************
X4_bit_full_adder A0 A1 A2 A3 B0 B1 B2 B3 CIN S0 S1 S2 S3 COUT RCA
* Add capacitors
C0 S0 Gnd 0.1p
C1 S1 Gnd 0.1p
C2 S2 Gnd 0.1p
C3 S3 Gnd 0.1p
Ccout COUT Gnd 0.1p
* SYNTAX: V <name> <node1> <node2> pulse( <value1>,<value2>, <delay>,<rise time>,<fall time>,<ON time>, <period>)
  *** define input signals for A and B PULSE
  VA0 A0 Gnd pulse( 0, 0.9V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VA1 A1 Gnd pulse( 0, 0.9V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VA2 A2 Gnd pulse( 0, 0.9V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VA3 A3 Gnd pulse( 0, 0.9V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VB0 B0 Gnd pulse( 0.9V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VB1 B1 Gnd pulse( 0.9V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VB2 B2 Gnd pulse( 0.9V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VB3 B3 Gnd pulse( 0.9V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
  VCIN CIN Gnd pulse( 0.9V, 0V, 1n, 0.001u, 0.001u, 7.999u, 8u )
* SYNTAX .TRAN <time step> <final time>
  .TRAN 0.049u 17u
  * .MEASURE FALL_DELAY
  +TRIG V(CIN) VAL='Supply/2' FALL = 1
  +TARG V(COUT) VAL='Supply/2' FALL = 1
  .MEASURE RISE_DELAY
  +TRIG V(CIN) VAL='Supply/2' RISE = 2
  +TARG V(COUT) VAL='Supply/2' RISE = 2
  * .MEASURE TRAN lowtohigh AVG i(VDD) FROM=3.99u TO=4.01u
  * .MEASURE TRAN low AVG i(VDD) FROM=3.97u TO=3.99u
  * .MEASURE TRAN hightolow AVG i(VDD) FROM=7.99u TO=8.01u
  * .MEASURE TRAN high AVG i(VDD) FROM=7.97u TO=7.99u
  * .MEASURE TRAN supplycurrent AVG i(VDD) FROM=3.99u TO=4.01u
  * .MEASURE TRAN lowtohighpower PARAM='lowtohigh * 0.6' **0.6 is supply voltage
  * .MEASURE TRAN lowpower PARAM='low * 0.6' **0.6 is supply voltage
  * .MEASURE TRAN hightolowpower PARAM='highto low * 0.6' **0.6 is supply voltage
  * .MEASURE TRAN highpower PARAM='high * 0.6' **0.6 is supply voltage
  .TEMP 90
  .OPTION Post
.end
B. Carry-Look-Ahead Adder Implementation in HSPICE

*library setup
.include "45nm_HP.pm"

*define parameters
.param Supply = 1.8V  +VDD supply voltage
.defines VDD as global node
.global VDD

*** add voltage source for VDD
VDD VDD Gnd Supply

****************************************************************************************
* Inverter
****************************************************************************************
.SUBCKT INV IN OUT
m0 OUT IN VDD VDD pmos l = 45n w = 360n
m1 OUT IN Gnd Gnd nmos l = 45n w = 180n
.ENDS

****************************************************************************************
* NAND
****************************************************************************************
.SUBCKT NAND A B OUT
*** P-MOS
m0 OUT A VDD VDD pmos l = 45n w = 360n
m1 OUT B VDD VDD pmos l = 45n w = 360n
*** NMOS
m2 OUT A x Gnd nmos l = 45n w = 180n
m3 x B Gnd Gnd nmos l = 45n w = 180n
.ENDS

****************************************************************************************
* NAND3
****************************************************************************************
.SUBCKT NAND3 A B C OUT
*** P-MOS
m0 OUT A VDD VDD pmos l = 45n w = 360n
m1 OUT B VDD VDD pmos l = 45n w = 360n
m2 OUT C VDD VDD pmos l = 45n w = 360n
*** NMOS
m3 OUT A x Gnd nmos l = 45n w = 180n
m4 x B y Gnd nmos l = 45n w = 180n
m5 y C Gnd Gnd nmos l = 45n w = 180n
.ENDS

****************************************************************************************
* NAND4
****************************************************************************************
.SUBCKT NAND4 A B C D OUT
*** P-MOS
m0 OUT A VDD VDD pmos l = 45n w = 360n
m1 OUT B VDD VDD pmos l = 45n w = 360n
m2 OUT C VDD VDD pmos l = 45n w = 360n
m3 OUT D VDD VDD pmos l = 45n w = 360n
*** NMOS
m4 OUT A x Gnd nmos l = 45n w = 180n
m5 x B y Gnd nmos l = 45n w = 180n
m6 y C z Gnd nmos l = 45n w = 180n
m7 z D Gnd Gnd nmos l = 45n w = 180n
.ENDS

****************************************************************************************
* NAND5
****************************************************************************************
.SUBCKT NAND5 A B C D E OUT
*** P-MOS
m0 OUT A VDD VDD pmos l = 45n w = 360n
m1 OUT B VDD VDD pmos l = 45n w = 360n
m2 OUT C VDD VDD pmos l = 45n w = 360n
m3 OUT D VDD VDD pmos l = 45n w = 360n
m4 OUT E VDD VDD pmos l = 45n w = 360n
*** NMOS
m5 OUT A w Gnd nmos l = 45n w = 180n
m6 w B x Gnd nmos l = 45n w = 180n
m7 x C y Gnd nmos l = 45n w = 180n
m8 y D z Gnd nmos l = 45n w = 180n
m9 z E Gnd Gnd nmos l = 45n w = 180n
.ENDS
* AND

.SUBCKT AND A B OUT
   Xnand A B x NAND
   Xinv x OUT INV
.ENDS

* AND3

.SUBCKT AND3 A B C OUT
   Xnand3 A B C x NAND3
   Xinv x OUT INV
.ENDS

* AND4

.SUBCKT AND4 A B C D OUT
   Xnand4 A B C D x NAND4
   Xinv x OUT INV
.ENDS

* AND5

.SUBCKT AND5 A B C D E OUT
   Xnand5 A B C D E x NAND5
   Xinv x OUT INV
.ENDS

* XOR

.SUBCKT XOR A B OUT
   *** A' ***
   m0 A_NOT A VDD VDD pmos l = 45n w = 360n
   m2 A_NOT A Gnd Gnd nmos l = 45n w = 180n
   *** B' ***
   m1 B_NOT B VDD VDD pmos l = 45n w = 360n
   m3 B_NOT B Gnd Gnd nmos l = 45n w = 180n
   *** XOR ***
   *** P-MOS
   m4 OUT A w VDD pmos l = 45n w = 360n
   m5 OUT A_NOT x VDD pmos l = 45n w = 360n
   m6 w B_NOT VDD VDD pmos l = 45n w = 360n
   m7 x B VDD VDD pmos l = 45n w = 360n
   *** NMOS
   m8 y A Gnd Gnd nmos l = 45n w = 180n
   m9 z A_NOT Gnd Gnd nmos l = 45n w = 180n
   m10 OUT B y Gnd nmos l = 45n w = 180n
   m11 OUT B_NOT z Gnd nmos l = 45n w = 180n
.ENDS

* NOR

.SUBCKT NOR A B OUT
   *** NMOS
   m0 OUT A Gnd Gnd nmos l = 45n w = 180n
   m1 OUT B Gnd Gnd nmos l = 45n w = 180n
   *** P-MOS
   m2 OUT A x VDD pmos l = 45n w = 360n
   m3 x B VDD VDD pmos l = 45n w = 360n
.ENDS

* NOR3

.SUBCKT NOR3 A B C OUT
   *** NMOS
   m0 OUT A Gnd Gnd nmos l = 45n w = 180n
   m1 OUT B Gnd Gnd nmos l = 45n w = 180n
   m2 OUT C Gnd Gnd nmos l = 45n w = 180n
   *** P-MOS
   m3 OUT A x VDD pmos l = 45n w = 360n
   m4 x B y VDD pmos l = 45n w = 360n
m5 y C VDD VDD pmos l = 45 n w = 360 n
.ENDS

******************************************************************************
* NOR4
******************************************************************************
.SUBCKT NOR4 A B C D OUT
*** NMOS
m0 OUT A Gnd Gnd nmos l = 45 n w = 180 n
m1 OUT B Gnd Gnd nmos l = 45 n w = 180 n
m2 OUT C Gnd Gnd nmos l = 45 n w = 180 n
m3 OUT D Gnd Gnd nmos l = 45 n w = 180 n
*** P-MOS
m4 OUT A x VDD pmos l = 45 n w = 360 n
m5 x B y VDD pmos l = 45 n w = 360 n
m6 y C z VDD pmos l = 45 n w = 360 n
m7 z D VDD VDD pmos l = 45 n w = 360 n
.ENDS

******************************************************************************
* NOR5
******************************************************************************
.SUBCKT NOR5 A B C D E OUT
*** NMOS
m0 OUT A Gnd Gnd nmos l = 45 n w = 180 n
m1 OUT B Gnd Gnd nmos l = 45 n w = 180 n
m2 OUT C Gnd Gnd nmos l = 45 n w = 180 n
m3 OUT D Gnd Gnd nmos l = 45 n w = 180 n
m4 OUT E Gnd Gnd nmos l = 45 n w = 180 n
*** P-MOS
m5 OUT A w VDD pmos l = 45 n w = 360 n
m6 w B x VDD pmos l = 45 n w = 360 n
m7 x C y VDD pmos l = 45 n w = 360 n
m8 y D z VDD pmos l = 45 n w = 360 n
m9 z E VDD VDD pmos l = 45 n w = 360 n
.ENDS

******************************************************************************
* OR
******************************************************************************
.SUBCKT OR A B OUT
Xnor A B x NOR
Xinv x OUT INV
.ENDS

******************************************************************************
* OR3
******************************************************************************
.SUBCKT OR3 A B C OUT
Xnor3 A B C x NOR3
Xinv x OUT INV
.ENDS

******************************************************************************
* OR4
******************************************************************************
.SUBCKT OR4 A B C D OUT
Xnor4 A B C D x NOR4
Xinv x OUT INV
.ENDS

******************************************************************************
* OR5
******************************************************************************
.SUBCKT OR5 A B C D E OUT
Xnor5 A B C D E x NOR5
Xinv x OUT INV
.ENDS

******************************************************************************
* CLA Full-Adder
******************************************************************************
*}
xnor A B x NOR
xinv x OUT INV
.ENDS

******************************************************************************
* CLA Full-Adder
******************************************************************************
* A ------| x| | x| |
* B | XOR |------x--- | XOR |---- S
* Cin ------| | Cin |
* A| |
* |
} AND |--- G
* Cin ------------------------------- B|
* A| |
* |
} OR |--- P
* B| |
* SUBCKT CLA_FULL_ADDER A B CIN S G P
  Xxor1 A B x XOR
  Xxor2 x CIN S XOR
  Xand A B G AND
  Xor A B P OR
.ENDS

****************************************************

* CLA
****************************************************

* Bit 0 ***
Xfull_adder0 A0 B0 CIN S0 G0 P0 CLA_FULL_ADDER

* Bit 1 ***
  C1 = G0 + (P0 x CIN)
  Xand1 P0 CIN X1 AND
  Xor1 G0 X1 C1 OR
  Xfull_adder1 A1 B1 C1 S1 G1 P1 CLA_FULL_ADDER

* Bit 2 ***
  C2 = G1 + (G0 x P1) + (CIN x P0 x P1)
  Xand2a CIN P0 P1 X2 AND3
  Xand2b G0 P1 Y2 AND
  Xor2 G1 X2 Y2 C2
  Xfull_adder2 A2 B2 C2 S2 G2 P2 CLA_FULL_ADDER

* Bit 3 ***
  C3 = G2 + (G1 x P2) + (G0 x P1 x P2) + (CIN x P0 x P1 x P2)
  Xand3a CIN P0 P1 P2 X3 AND4
  Xand3b G0 P1 P2 Y3 AND3
  Xand3c G1 P2 Z3 AND
  Xor3 G2 X3 Y3 Z3 C3 OR4
  Xfull_adder3 A3 B3 C3 S3 G3 P3 CLA_FULL_ADDER

*** COUT ***
  C4 = G3 + (G2 x P3) + (G1 x P2 x P3) + (G0 x P1 x P2 x P3) + (CIN x P0 x P1 x P2 x P3)
  Xand4a CIN P0 P1 P2 P3 W4 AND5
  Xand4b G0 P1 P2 P3 X4 AND4
  Xand4c G1 P2 P3 Y4 AND3
  Xand4d G2 P3 Z4 AND
  Xor4 G3 W4 X4 Y4 Z4 COUT OR5
.ENDS

****************************************************

* Test that FOUR_BIT_FULL_ADDER gate is correct
****************************************************

XCLA A0 A1 A2 A3 B0 B1 B2 B3 CIN S0 S1 S2 S3 COUT CLA
* Add capacitors
C0 S0 Gnd 10f
C1 S1 Gnd 10f
C2 S2 Gnd 10f
C3 S3 Gnd 10f
Ccout COUT Gnd 10f

* SYNTAX: V <name> <node1> <node2> pulse( <value1>,<value2>, <delay>,<rise time>,<fall time>,<ON time>, <period>)

*** define input signals for A and B PULSE
VA0 A0 Gnd pulse( 0, 1.8V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VA1 A1 Gnd pulse( 0, 1.8V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VA2 A2 Gnd pulse( 0, 1.8V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VA3 A3 Gnd pulse( 0, 1.8V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VB0 B0 Gnd pulse( 1.8V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VB1 B1 Gnd pulse( 1.8V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VB2 B2 Gnd pulse( 1.8V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VB3 B3 Gnd pulse( 1.8V, 0V, 1n, 0.001u, 0.001u, 7.999u, 16u )
VCIN CIN Gnd pulse( 1.8V, 0V, 1n, 0.001u, 0.001u, 3.999u, 8u )

* SYNTAX .TRAN <time step> <final time>
.TRAN 0.049u 17u

.MEASURE FALL_DELAY
+TRIG V(CIN) VAL='Supply/2' FALL = 1
+TARG V(COUT) VAL='Supply/2' FALL = 1
.MEASURE RISE_DELAY
+TRIG V(CIN) VAL='Supply/2' RISE = 2
+TARG V(COUT) VAL='Supply/2' RISE = 2

* .MEASURE TRAN lowtohigh AVG i(VDD) FROM=3.99u TO=4.01u
* .MEASURE TRAN low AVG i(VDD) FROM=3.97u TO=3.99u
* .MEASURE TRAN hightoclow AVG i(VDD) FROM=7.99u TO=8.01u
* .MEASURE TRAN high AVG i(VDD) FROM=7.97u TO=7.99u
* * .MEASURE TRAN supplycurrent AVG i(VDD) FROM=3.99u TO=4.01u
* .MEASURE TRAN lowtohighpower PARAM='lowtohigh * 0.6' *0.6 is supply voltage
* .MEASURE TRAN lowpower PARAM='low * 0.6' *0.6 is supply voltage
* .MEASURE TRAN hightolowpower PARAM='hightolow * 0.6' *0.6 is supply voltage
* .MEASURE TRAN highpower PARAM='high * 0.6' *0.6 is supply voltage

.TEMP 90
.OPTION Post
.end
A. RCA Waveforms

The following three waveforms are from the same simulation for the RCA. Waveforms 1, 2, and 3 in this section show the four-bit input A, four-bit input B as well as the CIN bit, and four-bit output sum and one-bit COUT result respectively. The waveforms are the results of two test cases:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Input A</th>
<th>Input B</th>
<th>CIN</th>
<th>Test Begin</th>
<th>Test End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111</td>
<td>0001</td>
<td>0</td>
<td>0.0</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>1110</td>
<td>0001</td>
<td>0</td>
<td>0.4</td>
<td>0.8</td>
</tr>
</tbody>
</table>

1) RCA Simulation: Waveform of the Four-Bit Input A

2) RCA Simulation: Waveform of the Four-Bit Input B and One-Bit CIN
3) **RCA Simulation: Waveform of the Four-Bit Output Sum and One-Bit COUT**

![Graph of RCA Simulation](image)

**B. CLA Waveforms**

The following three waveforms are from the same simulation for the CLA. Waveforms 1, 2, and 3 in this section show the four-bit input A, four-bit input B as well as the CIN bit, and four-bit output sum and one-bit COUT result respectively. The waveforms are the results of two test cases:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Input A</th>
<th>Input B</th>
<th>CIN</th>
<th>Test Begin</th>
<th>Test End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111</td>
<td>0001</td>
<td>0</td>
<td>0.0</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>1110</td>
<td>0001</td>
<td>0</td>
<td>0.4</td>
<td>0.8</td>
</tr>
</tbody>
</table>

1) **CLA Simulation: Waveform of the Four-Bit Input A**

![Graph of CLA Simulation](image)
2) CLA Simulation: Waveform of the Four-Bit Input B and One-Bit CIN

3) CLA Simulation: Waveform of the Four-Bit Output Sum and One-Bit COUT