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. Interver

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*.

$$\begin{aligned} S_H &= A_H \oplus B_H \\ C_H &= A_H \cdot B_H \end{aligned} \tag{1}$$

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*.

$$\begin{aligned} A_H &= A_{RF} \\ B_H &= B_{RF} \\ S_F &= S_H \oplus CIN_{RF} \\ COUT_{RF} &= COUT_H + (S_H \cdot CIN_{RF}) \end{aligned} \tag{2}$$

C. Four-Bit RCA

The four-bit RCA has nine inputs: two four-bit binary numbers denoted as $A_R[3:0]$ and $B_R[3:0]$ along with a one-bit carry-in denoted as CIN_R; the RCA has five outputs: a four-bit sum denoted as $S_R[3:0]$ 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 pervious 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.

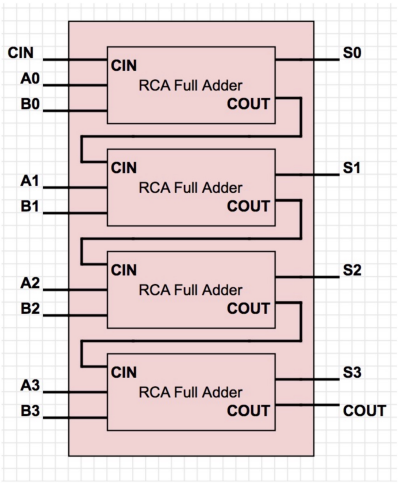


Fig. 1. Figure of the schematic layout of a RCA demonstrating the concatenation of the full adders

TABLE I
RCA Correctness Test Cases and Results

	Input			Expected		Actual	
Test	**A**	**B**	**CIN**	**S**	**COUT**	**S**	**COUT**
1	1111	0001	0	0000	1	0000	1
2	1110	0001	0	1111	0	1111	0

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).

$$G_i = A_i \cdot B_i \tag{3}$$

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).

$$P_i = A_i + B_i \tag{4}$$

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.

$$C_{i+1} = G_i + (P_i \cdot C_i) \tag{5}$$

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

$$\begin{aligned} CIN_0 &= CIN \\ CIN_1 &= G_0 + (P_0 \cdot C_0) \\ CIN_2 &= G_1 + (P_1 \cdot C_1) \\ CIN_3 &= G_2 + (P_2 \cdot C_2) \end{aligned} \tag{6}$$

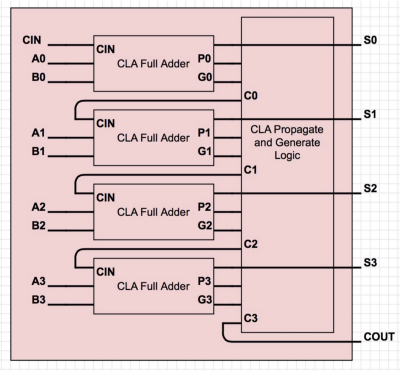


Fig. 2. Figure of the schematic layout of the CLA demonstrating the CLA full adders with the generating and propagating carry logic.

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*.

$$\begin{aligned} S_{CF} &= A_{CF} \oplus B_{CF} \oplus CIN_{CF} \\ G_{CF} &= A_{CF} \cdot B_{CF} \\ P_{CF} &= A_{CF} + B_{CF} \end{aligned} \tag{7}$$

E. Fout-Bit CLA

The four-bit CLA has nine inputs: two four-bit binary numbers denoted as $A_C[3:0]$ and $B_C[3:0]$ along with a one-bit carry-in denoted as CIN_C; the CLA has five outputs: a four-bit sum denoted as $S_C[3:0]$ along with a one-bit carry-out denoted as $COUT_C$. 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 are connected to each of the sum-bits as well as the carry-out 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

	Input			Expected		Actual	
Test	**A**	**B**	**CIN**	**S**	**COUT**	**S**	**COUT**
1	1111	0001	0	0000	1	0000	1
2	1110	0001	0	1111	0	1111	0

TABLE III
LOW-TO-HIGH PROPAGATION DELAY OF CRITICAL PATH RESULTS OF THE RCA AND CLA IN NANO-SECONDS

Adder	**VDD (Volts)**	**Temperature (Celcius)**		
		0	**25**	**90**
RCA	**0.6**	1.037	1.183	1.668
	0.9	0.313	0.354	0.510
	1.8	0.256	0.225	0.250
CLA	**0.6**	0.389	0.426	0.543
	0.9	0.129	0.145	0.194
	1.8	0.206	0.209	0.215

TABLE IV
HIGH-TO-LOW PROPAGATION DELAY OF CRITICAL PATH RESULTS OF THE RCA AND CLA IN NANO-SECONDS

Adder	**VDD (Volts)**	**Temperature (Celcius)**		
		0	**25**	**90**
RCA	**0.6**	1.121	1.248	1.632
	0.9	0.403	0.443	0.585
	1.8	0.114	0.123	0.255
CLA	**0.6**	0.544	0.607	0.774
	0.9	0.186	0.211	0.281
	1.8	0.011	0.010	0.008

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.

TABLE V
PROPAGATION DELAY ESTIMATIONS OF RCA AND CLA IN NANO-SECONDS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	1.079	1.216	1.650
	0.9	0.358	0.399	0.548
	1.8	0.185	0.174	0.238
CLA	0.6	0.467	0.517	0.659
	0.9	0.158	0.178	0.238
	1.8	0.109	0.110	0.112

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 (9). The time frame used to measure the average current will be defined in the following two sections.

$$P = I_{avg} \cdot V_{DD} \tag{9}$$

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.

$$P_D = \frac{\sum_{t=T_1}^{T_2} I(t)}{T_2 - T_1} \cdot V_{DD} \tag{10}$$

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 ϵ 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 ϵ will be 0.01 micro-seconds.

$$P_D = \frac{\sum_{t=T_1-\epsilon}^{T_2+\epsilon} I(t)}{(T_2 + \epsilon) - (T_1 - \epsilon)} \cdot V_{DD} \tag{11}$$

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.

TABLE VI
LOW-TO-HIGH DYNAMIC POWER DISSIPATION IN RCA AND CLA IN MICRO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	2.153	2.165	2.231
	0.9	5.207	5.244	5.363
	1.8	407.5	408.5	406.9
CLA	0.6	0.535	0.553	0.653
	0.9	1.780	1.859	2.129
	1.8	595.1	590.9	586.8

TABLE VII
HIGH-TO-LOWS DYNAMIC POWER DISSIPATION IN RCA AND CLA IN MICRO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	7.712	7.696	7.695
	0.9	19.31	19.14	19.05
	1.8	613.8	600.9	572.5
CLA	0.6	1.251	1.277	1.376
	0.9	5.224	5.085	5.090
	1.8	766.2	743.3	700.0

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).

$$P_D = \frac{P_{DLH} + P_{DHL}}{2} \tag{12}$$

TABLE VIII
STATIC-LOW POWER DISSIPATION IN RCA AND CLA IN NANO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	8.843	17.97	84.98
	0.9	7.982	129.9	334.0
	1.8	340900	342000	346300
CLA	0.6	15.63	29.97	127.6
	0.9	46.09	211.4	570.4
	1.8	541400	540600	543700

TABLE IX
STATIC-HIGH DYNAMIC POWER DISSIPATION IN RCA AND CLA IN NANO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	2.892	19.43	67.95
	0.9	49.20	83.44	291.7
	1.8	387900	388500	393000
CLA	0.6	12.16	22.99	76.59
	0.9	94.85	144.8	418.7
	1.8	529900	529700	533600

TABLE X
DYNAMIC POWER DISSIPATION ESTIMATIONS OF RCA AND CLA IN MICRO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	4.933	4.931	4.963
	0.9	12.26	12.19	12.21
	1.8	510.7	504.7	489.7
CLA	0.6	0.893	0.915	1.015
	0.9	3.500	3.470	3.610
	1.8	680.7	677.1	643.4

TABLE XI
STATIC POWER DISSIPATION ESTIMATIONS OF RCA AND CLA IN NANO-WATTS

Adder	VDD (Volts)	Temperature (Celcius)		
		0	25	90
RCA	0.6	5.870	18.70	76.47
	0.9	28.60	106.7	312.9
	1.8	364400	365250	369650
CLA	0.6	13.90	26.48	102.1
	0.9	70.50	178.1	494.6
	1.8	535650	535150	538650

Similarly, the estimation of static power dissipation is the average of the static-high and-static low power dissipation (13).

$$P_D = \frac{P_{DLH} + P_{DHL}}{2} \tag{13}$$

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

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

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.

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.

APPENDIX A
HSPICE SIMULATION PARAMETERS

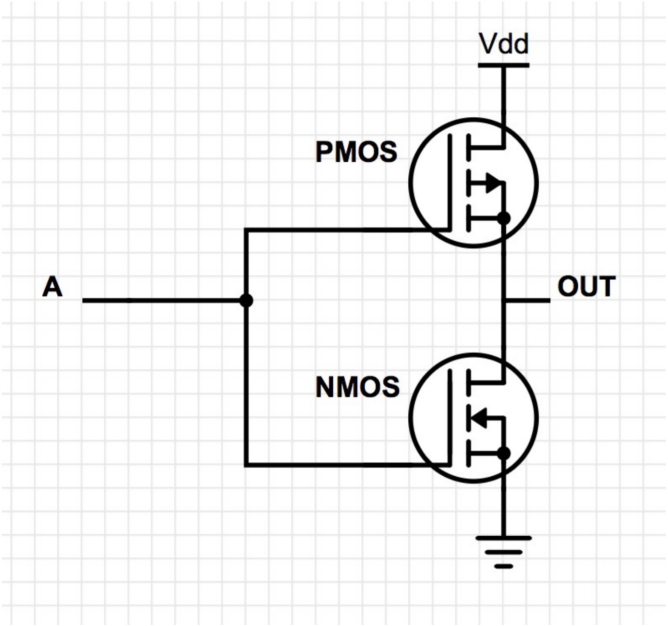
The following is the parameter file used for the HSPICE simulations used in this paper:

```
.model  nmos  nmos  level = 54

+version = 4.0            binunit = 1             paramchk= 1             mobmod  = 0
+capmod  = 2              igcmod  = 1             igbmod  = 1             geomod  = 1
+diomod  = 1              rdsmod  = 0             rbodymod= 1             rgatemod= 1
+permod  = 1              acnqsmod= 0             trnqsmod= 0

+tnom    = 27             toxe    = 1.25e-009     toxp    = 1e-009        toxm    = 1.25e-009
+dtox    = 2.5e-010       epsrox  = 3.9           wint    = 5e-009        lint    = 3.75e-009
+ll      = 0              wl      = 0             lln     = 1             wln     = 1
+lw      = 0              ww      = 0             lwn     = 1             wwn     = 1
+lwl     = 0              wwl     = 0             xpart   = 0             toxref  = 1.25e-009
+xl    = -20e-9

+vth0    = 0.46893        k1      = 0.4           k2      = 0             k3      = 0
+k3b     = 0              w0      = 2.5e-006      dvt0    = 1             dvt1    = 2
+dvt2    = 0              dvt0w   = 0             dvt1w   = 0             dvt2w   = 0
+dsub    = 0.1            minv    = 0.05          voff1   = 0             dvtp0   = 1e-010
+dvtp1   = 0.1            lpe0    = 0             lpeb    = 0             xj      = 1.4e-008
+ngate   = 1e+023         ndep    = 3.24e+018     nsd     = 2e+020        phin    = 0
+cdsc    = 0              cdscb   = 0             cdscd   = 0             cit     = 0
+voff    = -0.13          nfactor = 2.22          eta0    = 0.0055        etab    = 0
+vfb     = -0.55          u0      = 0.054         ua      = 6e-010        ub      = 1.2e-018
+uc      = 0              vsat    = 170000        a0      = 1             ags     = 0
+a1      = 0              a2      = 1             b0      = 0             b1      = 0
+keta    = 0.04           dwg     = 0             dwb     = 0             pclm    = 0.02
+pdiblc1 = 0.001          pdiblc2 = 0.001         pdiblcb = -0.005        drout   = 0.5
+pvag    = 1e-020         delta   = 0.01          pscbe1  = 8.14e+008     pscbe2  = 1e-007
+fprout  = 0.2            pdits   = 0.08          pditsd  = 0.23          pditsl  = 2300000
+rsh     = 5              rdsw    = 155           rsw     = 80            rdw     = 80
+rdswmin = 0              rdwmin  = 0             rswmin  = 0             prwg    = 0
+prwb    = 0              wr      = 1             alpha0  = 0.074         alpha1  = 0.005
+beta0   = 30             agidl   = 0.0002        bgidl   = 2.1e+009      cgidl   = 0.0002
+egidl   = 0.8            aigbacc = 0.012         bigbacc = 0.0028        cigbacc = 0.002
+nigbacc = 1              aigbinv = 0.014         bigbinv = 0.004         cigbinv = 0.004
+eigbinv = 1.1            nigbinv = 3             aigc    = 0.02          bigc    = 0.0025
+cigc    = 0.002          aigsd   = 0.02          bigsd   = 0.0025        cigsd   = 0.002
+nigc    = 1              poxedge = 1             pigcd   = 1             ntox    = 1
+xrcrg1  = 12             xrcrg2  = 5

+cgso    = 1.1e-010       cgdo    = 1.1e-010      cgbo    = 2.56e-011     cgdl    = 2.653e-010
+cgsl    = 2.653e-010     ckappas = 0.03          ckappad = 0.03          acde    = 1
+moin    = 15             noff    = 0.9           voffcv  = 0.02

+kt1     = -0.11          kt11    = 0             kt2     = 0.022         ute     = -1.5
+ua1     = 4.31e-009      ub1     = 7.61e-018     uc1     = -5.6e-011     prt     = 0
+at      = 33000

+fnoimod = 1              tnoimod = 0

+jss     = 0.0001         jsws    = 1e-011        jswgs   = 1e-010        njs     = 1
+ijthsfwd= 0.01           ijthsrev= 0.001         bvs     = 10            xjbvs   = 1
+jsd     = 0.0001         jswd    = 1e-011        jswgd   = 1e-010        njd     = 1
+ijthdfwd= 0.01           ijthdrev= 0.001         bvd     = 10            xjbvd   = 1
+pbs     = 1              cjs     = 0.0005        mjs     = 0.5           pbsws   = 1
+cjsws   = 5e-010         mjsws   = 0.33          pbswgs  = 1             cjswgs  = 3e-010
+mjswgs  = 0.33           pbd     = 1             cjd     = 0.0005        mjd     = 0.5
+pbswd   = 1              cjswd   = 5e-010        mjswd   = 0.33          pbswgd  = 1
+cjswgd  = 5e-010         mjswgd  = 0.33          tpb     = 0.005         tcj     = 0.001
+tpbsw   = 0.005          tcjsw   = 0.001         tpbswg  = 0.005         tcjswg  = 0.001
+xtis    = 3              xtid    = 3

+dmcg    = 0              dmci    = 0             dmdg    = 0             dmcgt   = 0
+dwj     = 0              xgw     = 0             xgl     = 0

+rshg    = 0.4            gbmin   = 1e-010         rbpb    = 5             rbpd    = 15
+rbps    = 15             rbdb    = 15            rbsb    = 15            ngcon   = 1

.model  pmos  pmos  level = 54

+version = 4.0            binunit = 1             paramchk= 1             mobmod  = 0
+capmod  = 2              igcmod  = 1             igbmod  = 1             geomod  = 1
+diomod  = 1              rdsmod  = 0             rbodymod= 1             rgatemod= 1
+permod  = 1              acnqsmod= 0             trnqsmod= 0

+tnom    = 27             toxe    = 1.3e-009      toxp    = 1e-009        toxm    = 1.3e-009
+dtox    = 3e-010         epsrox  = 3.9           wint    = 5e-009        lint    = 3.75e-009
+ll      = 0              wl      = 0             lln     = 1             wln     = 1
+lw      = 0              ww      = 0             lwn     = 1             wwn     = 1
+lwl     = 0              wwl     = 0             xpart   = 0             toxref  = 1.3e-009
+xl    = -20e-9
```

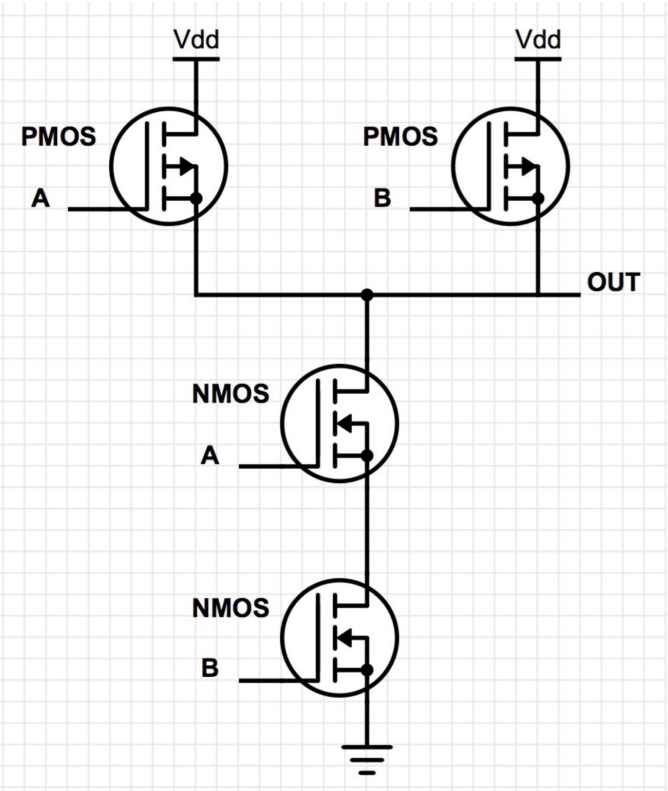
```
+vth0    = -0.49158        k1      = 0.4             k2      = -0.01           k3      = 0
+k3b     = 0               w0      = 2.5e-006        dvt0    = 1               dvt1    = 2
+dvt2    = -0.032          dvt0w   = 0               dvt1w   = 0               dvt2w   = 0
+dsub    = 0.1             minv    = 0.05            voff1   = 0               dvtp0   = 1e-011
+dvtp1   = 0.05            lpe0    = 0               lpeb    = 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    = 150000          a0      = 1               ags     = 1e-020
+a1      = 0               a2      = 1               b0      = 0               b1      = 0
+keta    = -0.047          dwg     = 0               dwb     = 0               pclm    = 0.12
+pdiblc1 = 0.001           pdiblc2 = 0.001           pdiblcb = 3.4e-008        drout   = 0.56
+pvag    = 1e-020          delta   = 0.01            pscbe1  = 8.14e+008       pscbe2  = 9.58e-007
+fprout  = 0.2             pdits   = 0.08            pditsd  = 0.23            pditsl  = 2300000
+rsh     = 5               rdsw    = 155             rsw     = 75              rdw     = 75
+rdswmin = 0               rdwmin  = 0               rswmin  = 0               prwg    = 0
+prwb    = 0               wr      = 1               alpha0  = 0.074           alpha1  = 0.005
+beta0   = 30              agidl   = 0.0002          bgidl   = 2.1e+009        cgidl   = 0.0002
+egidl   = 0.8             aigbacc = 0.012           bigbacc = 0.0028          cigbacc = 0.002
+nigbacc = 1               aigbinv = 0.014           bigbinv = 0.004           cigbinv = 0.004
+eigbinv = 1.1             nigbinv = 3               aigc    = 0.010687        bigc    = 0.0012607
+cigc    = 0.0008          aigsd   = 0.010687        bigsd   = 0.0012607       cigsd   = 0.0008
+nigc    = 1               poxedge = 1               pigcd   = 1               ntox    = 1
+xrcrg1  = 12              xrcrg2  = 5

+cgso    = 1.1e-010        cgdo    = 1.1e-010        cgbo    = 2.56e-011       cgdl    = 2.653e-010
+cgsl    = 2.653e-010      ckappas = 0.03            ckappad = 0.03            acde    = 1
+moin    = 15              noff    = 0.9             voffcv  = 0.02

+kt1     = -0.11           kt11    = 0               kt2     = 0.022           ute     = -1.5
+ua1     = 4.31e-009       ub1     = 7.61e-018       uc1     = -5.6e-011       prt     = 0
+at      = 33000

+fnoimod = 1               tnoimod = 0

+jss     = 0.0001          jsws    = 1e-011          jswgs   = 1e-010          njs     = 1
+ijthsfwd= 0.01            ijthsrev= 0.001           bvs     = 10              xjbvs   = 1
+jsd     = 0.0001          jswd    = 1e-011          jswgd   = 1e-010          njd     = 1
+ijthdfwd= 0.01            ijthdrev= 0.001           bvd     = 10              xjbvd   = 1
+pbs     = 1               cjs     = 0.0005          mjs     = 0.5             pbsws   = 1
+cjsws   = 5e-010          mjsws   = 0.33            pbswgs  = 1               cjswgs  = 3e-010
+mjswgs  = 0.33            pbd     = 1               cjd     = 0.0005          mjd     = 0.5
+pbswd   = 1               cjswd   = 5e-010          mjswd   = 0.33            pbswgd  = 1
+cjswgd  = 5e-010          mjswgd  = 0.33            tpb     = 0.005           tcj     = 0.001
+tpbsw   = 0.005           tcjsw   = 0.001           tpbswg  = 0.005           tcjswg  = 0.001
+xtis    = 3               xtid    = 3

+dmcg    = 0               dmci    = 0               dmdg    = 0               dmcgt   = 0
+dwj     = 0               xgw     = 0               xgl     = 0

+rshg    = 0.4             gbmin   = 1e-010          rbpb    = 5               rbpd    = 15
+rbps    = 15              rbdb    = 15              rbsb    = 15              ngcon   = 1
```

APPENDIX B
CMOS AND ADDER COMPONENT SCHEMATICS

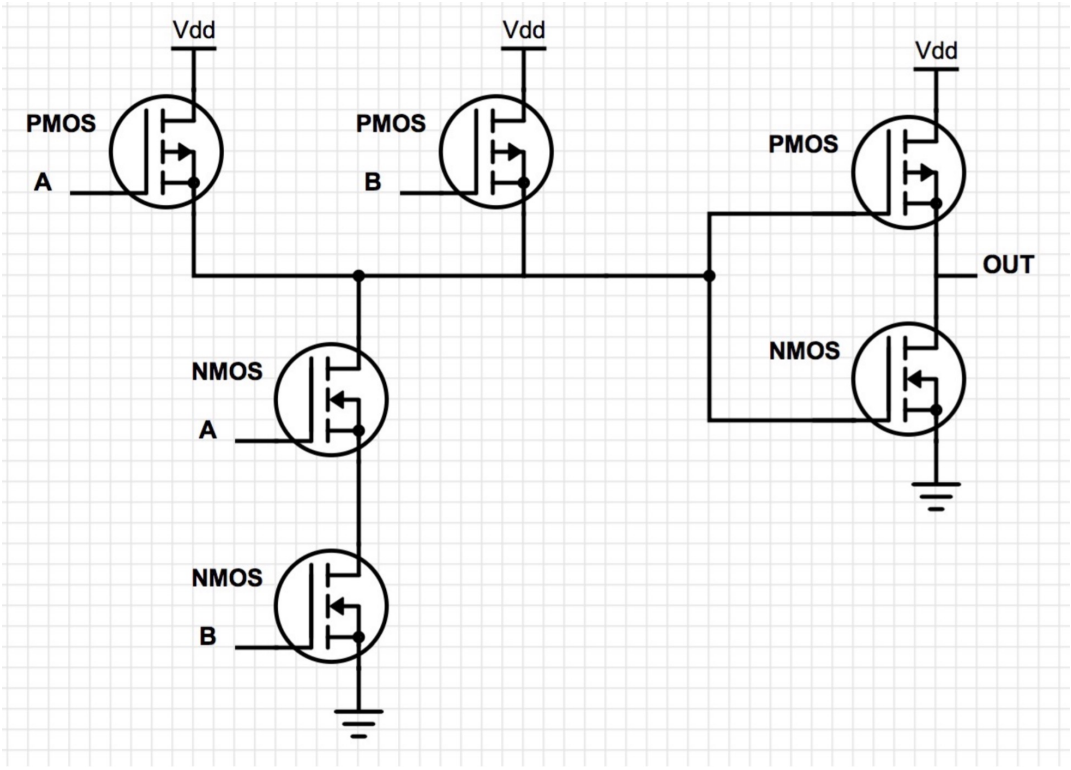
A. CMOS Implementation Schematic of an Inverter



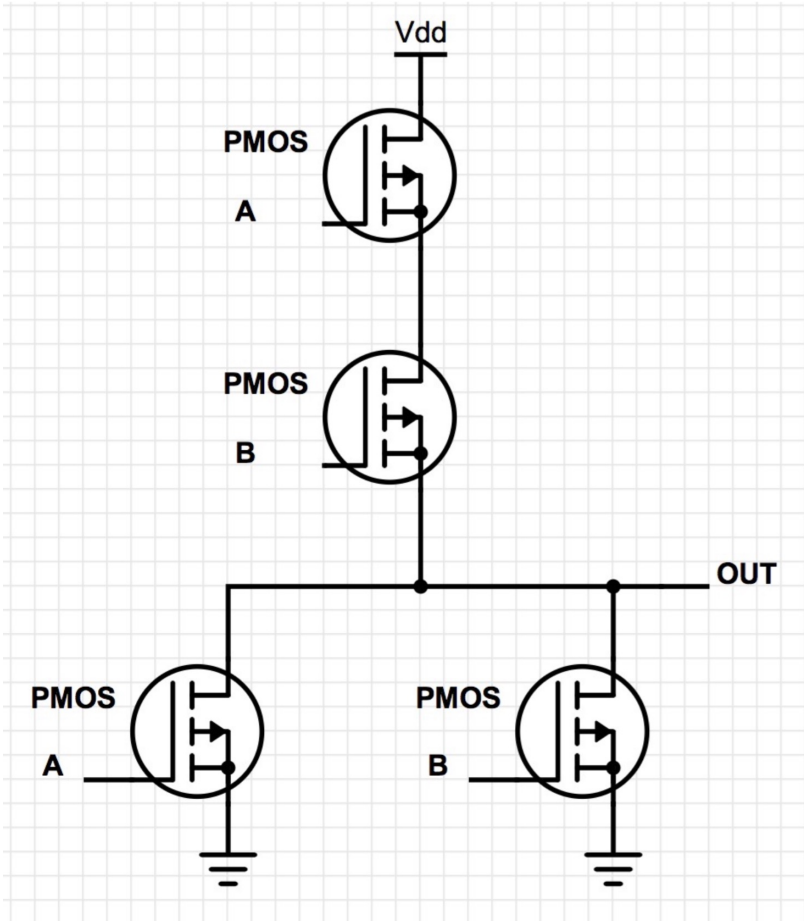
B. CMOS Implementation Schematic of an NAND Gate



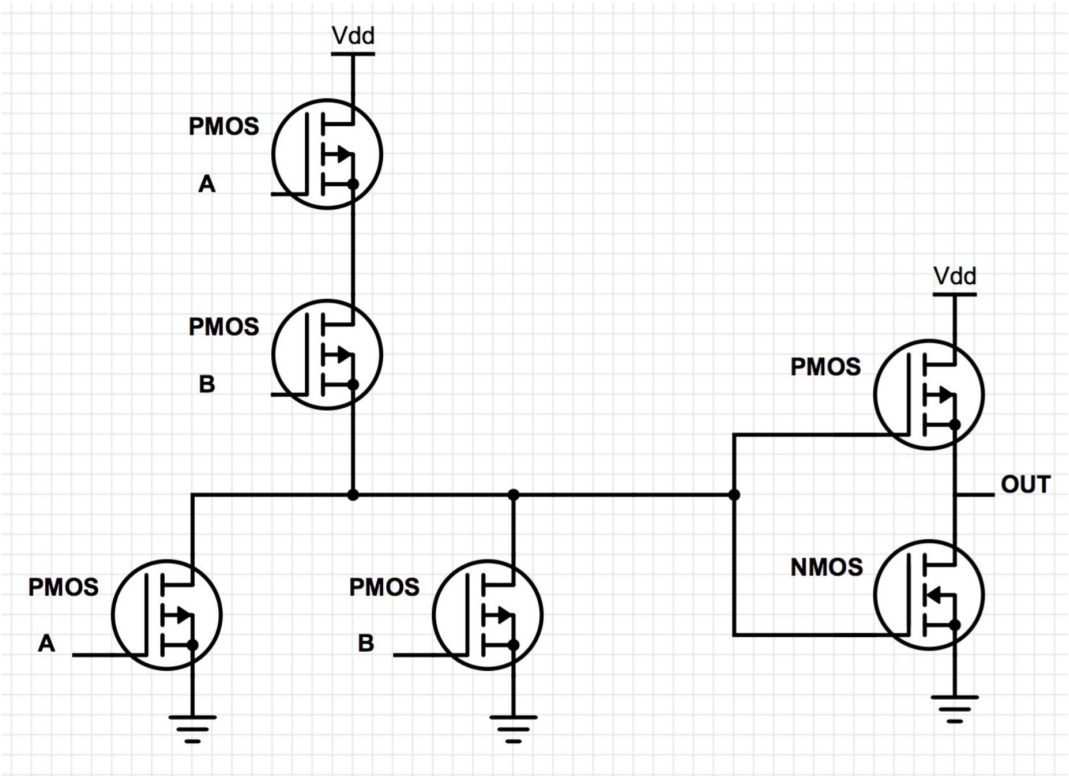
C. CMOS Implementation Schematic of an AND Gate



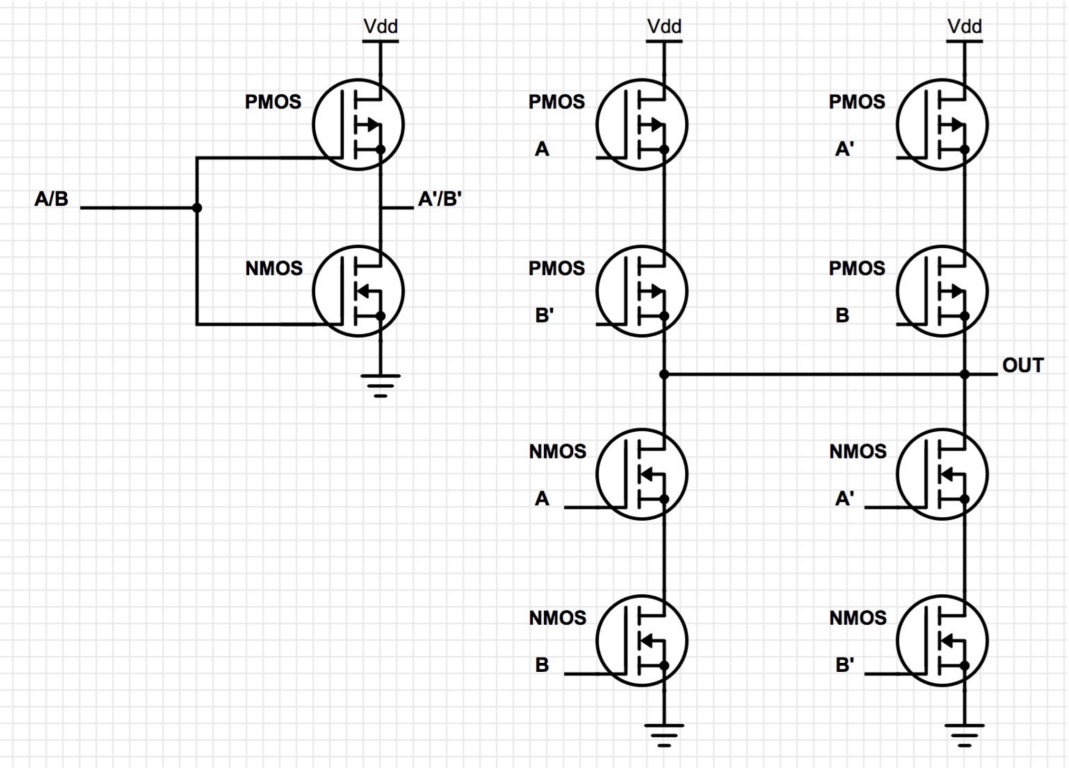
D. CMOS Implementation Schematic of an NOR Gate



E. CMOS Implementation Schematic of an OR Gate



F. CMOS Implementation Schematic of an XOR Gate



APPENDIX C
RCA AND CLA HSPICE IMPLEMENTATION

A. Ripple-Carry Adder Implementation in HSPICE

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

*define parameters
.param Supply = 0.9V  *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

*****************************************************
* AND
*****************************************************
* A --|
*     | NAND | -- x --| INV | -- OUT
* B --|
*****************************************************

.SUBCKT AND A B OUT
  Xnand   A B   x   NAND
  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    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 ***
.SUBCKT NOR A B OUT
  m0    OUT     A     Gnd     Gnd nmos l = 45n  w = 180n
  m1    OUT     B     Gnd     Gnd nmos l = 45n  w = 180n

*** 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
*           | Half_Adder  |               Cin|     |
*           |             |
*    B -----|_____________|---C---y---      x|     |
*                                            | AND |---z
*  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  -----|_____________|
*             _____________
*   A1 -----|             |
*           |             |---S1-----
*   B1 -----| Half_Adder  |
*           |             |---COUT1---
* COUT0-----|_____________|
*             _____________
*   A2 -----|             |
*           |             |---S2-----
*   B2 -----| Half_Adder  |
*           |             |---COUT2---
* COUT1-----|_____________|
*             _____________
*   A3 -----|             |
*           |             |---S3-----
*   B3 -----| Half_Adder  |
*           |             |---COUT---
* COUT2-----|_____________|
*
*****************************************************
```

```
.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,   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 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='-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
```

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 = 45n w = 360n
.ENDS

*****************************************************
* NOR4
*****************************************************
.SUBCKT NOR4 A B C D 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
  m3 OUT  D   Gnd   Gnd   nmos l = 45n w = 180n

  *** P-MOS
  m4 OUT  A   x     VDD   pmos l = 45n w = 360n
  m5 x    B   y     VDD   pmos l = 45n w = 360n
  m6 y    C   z     VDD   pmos l = 45n w = 360n
  m7 z    D   VDD   VDD   pmos l = 45n w = 360n
.ENDS

*****************************************************
* NOR5
*****************************************************
.SUBCKT NOR5 A B C D E 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
  m3 OUT  D   Gnd   Gnd   nmos l = 45n w = 180n
  m4 OUT  E   Gnd   Gnd   nmos l = 45n w = 180n

  *** P-MOS
  m5 OUT  A   w   VDD   pmos l = 45n w = 360n
  m6 w    B   x   VDD   pmos l = 45n w = 360n
  m7 x    C   y   VDD   pmos l = 45n w = 360n
  m8 y    D   z   VDD   pmos l = 45n w = 360n
  m9 z    E   VDD VDD   pmos l = 45n w = 360n
.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
*****************************************************
*
*    A -----|     |                        x|     |
*           | XOR |-------x---              | XOR |--- S
*    B -----|     |                      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
****************************************************
.SUBCKT CLA A0 A1 A2 A3 B0 B1 B2 B3 CIN S0 S1 S2 S3 COUT

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

*** Bit 1 ***
* C1 = G0 + (P0 * 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 * P1) + (CIN * P0 * P1)
  Xand2a        CIN P0 P1   X2        AND3
  Xand2b        G0 P1       Y2        AND
  Xor2          G1 X2 Y2    C2        OR3
  Xfull_adder2  A2 B2 C2    S2 G2 P2  CLA_FULL_ADDER


*** Bit 3 ***
* C3 = G2 + (G1 * P2) + (G0 * P1 * P2) + (CIN * P0 * P1 * 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 * P3) + (G1 * P2 * P3) + (G0 * P1 * P2 * P3) + (CIN * P0 * P1 * P2 * 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 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='-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
```

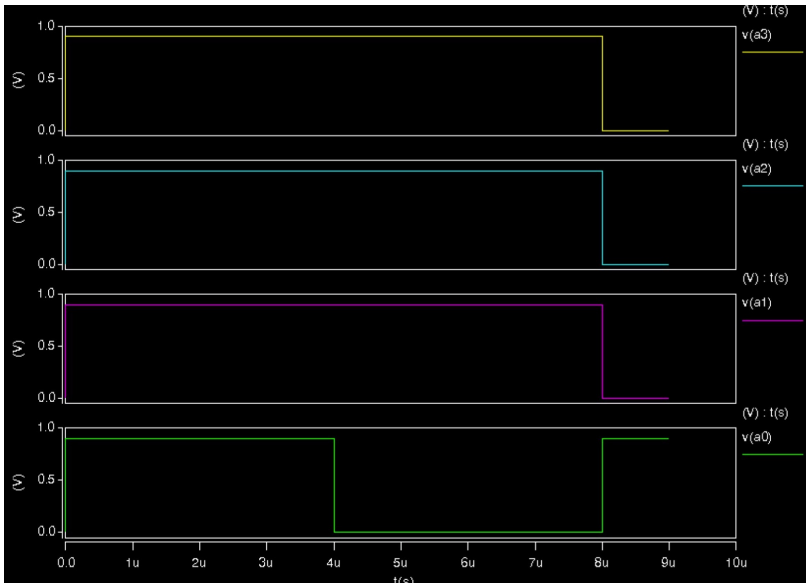
APPENDIX D
COSMOSSCOPE WAVEFORMS

A. RCA Waveforms

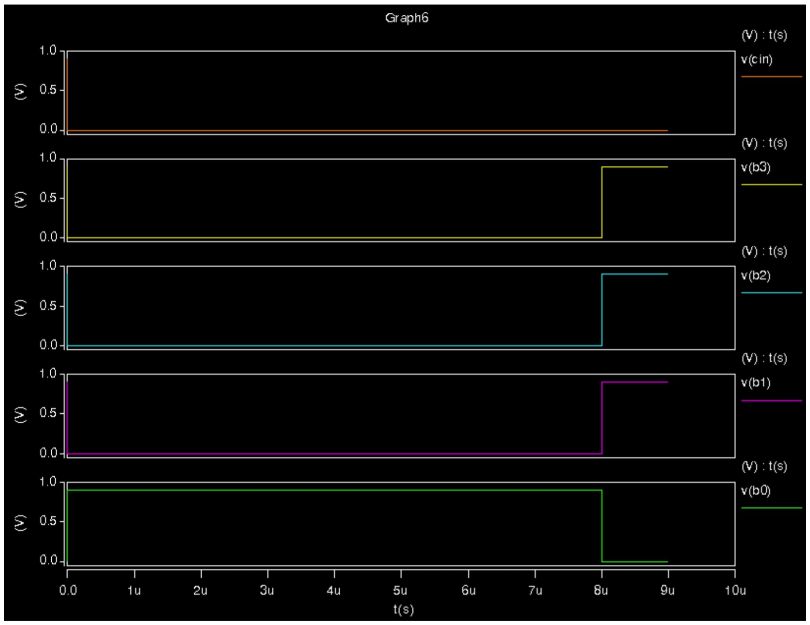
The following three waveforms are from the same simulation for the RCA. Waveforms 1, 2, and 3 in this section show the four-bit input A, four-bit input B as well as the CIN bit, and four-bit output sum and one-bit COUT result respectively. The waveforms are the results of two test cases:

Test Case	Input A	Input B	CIN	Test Begin	Test End
				micro-seconds	
1	1111	0001	0	0.0	0.4
2	1110	0001	0	0.4	0.8

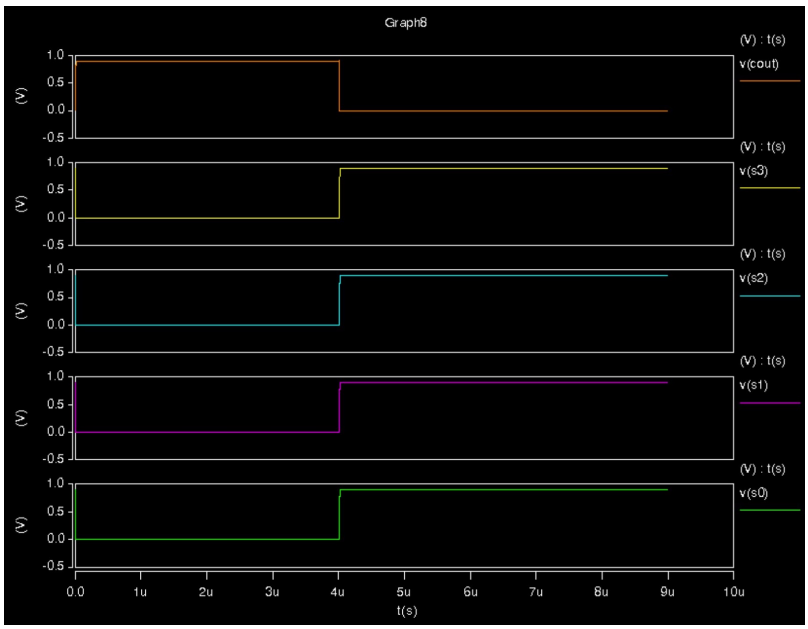
1) RCA Simulation: Waveform of the Four-Bit Input A



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

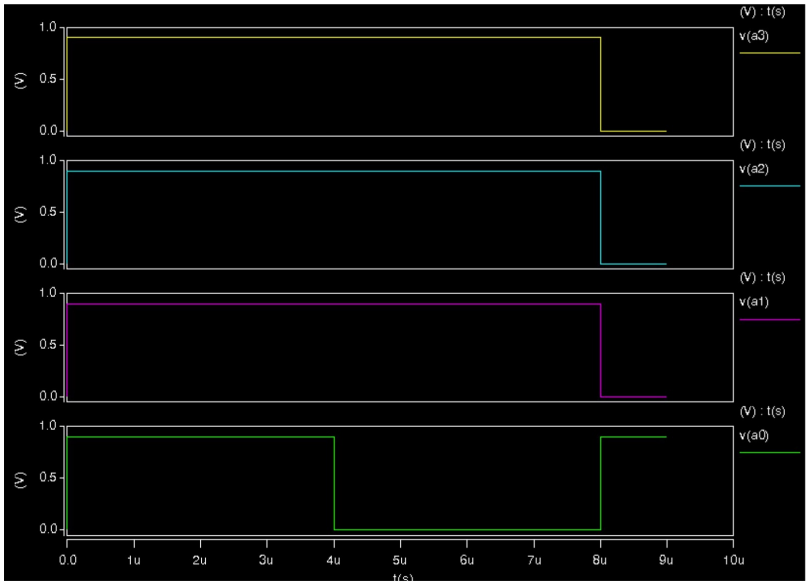


B. CLA Waveforms

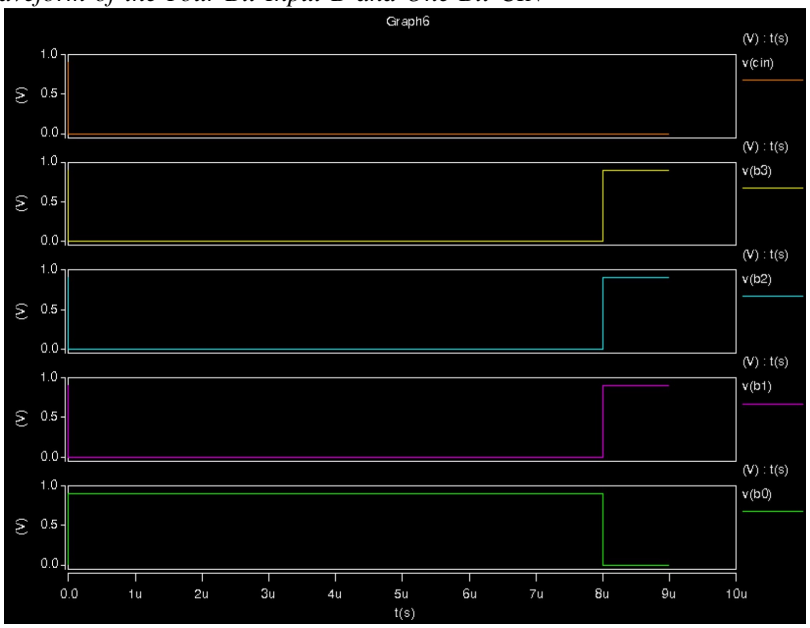
The following three waveforms are from the same simulation for the CLA. Waveforms 1, 2, and 3 in this section show the four-bit input A, four-bit input B as well as the CIN bit, and four-bit output sum and one-bit COUT result respectively. The waveforms are the results of two test cases:

Test Case	**Input A**	**Input B**	**CIN**	**Test Begin**	**Test End**
				micro-seconds	
1	1111	0001	0	0.0	0.4
2	1110	0001	0	0.4	0.8

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



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