HIGH PERFORMANCE 16 BIT MULTIPLIER DESIGN USING MOS CURRENT MODE LOGIC

by

Yavuz Delican

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2008

M.S., Electronics and Communications Engineering, Yıldız Teknik University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor Avni Morgül for his guidance and help throughout my research for this thesis and for providing me with all the prerequisites for a research mind and environment. I am also grateful for the helpful advices of Professor Günhan Dündar and Professor Sadri Özcan.

I would like to thank Uğur Çini for his guidance and help throughout my thesis. I would also like to thank for TUBITAK for its scholarship.

I am also grateful for the endless support and encouragement of my family.

ABSTRACT

HIGH PERFORMANCE 16 BIT MULTIPLIER DESIGN USING MOS CURRENT MODE LOGIC

In this work, MOS Current Mode Logic (MCML) is determined for low power and high speed applications. A small MCML cell library is developed and optimized for several different performance requirements. These requirements are voltage swing, voltage gain and current. The cells are then applied to the implementation of a 16 bit signed multiplier . In the literature, there are only two 8 bit MCML multipliers [12] [13] . We couldn't find any 16 bit MCML multiplier in the literature.

Modified radix 4 booth encoding is used to reduce the number of partial products by half for the multiplier architecture. Also, Wallace tree structure is used to sum the partial products in reduced time. In the Wallace tree, only full adders and half adders are used. For the completion adder, a fast and low power 32 bit hybrid adder is used.

Design is done in the 0.18µm UMC technology. A 16 bit non-pipelined MCML signed multiplier is designed and tested with 4 different reference voltages. The first one works at 800 MHz, consumes 55 mW. The second one works at 600 MHz, consumes 37mW. The third one works at 400 MHz, consumes 27mW. The last one works at 250 MHz, consumes 16 mW. The power of the third one is 25 % less than the CMOS multiplier with equivalent operating frequency, given in reference [14]. Also the speed of the second one is 50 % faster than the CMOS multiplier with equivalent power, given in reference [14]. The power supply current spike is only 4 % of the nominal current and the multiplier consists of 7268 transistors. The CMOS multiplier given in reference [14] consists of 13444 transistors.

ÖZET

AKIM MODLU MANTIK DEVRELERİ KULLANILARAK YAPILAN YÜKSEK PERFORMANSLI 16 BİT ÇARPICI DEVRESİ TASARIMI

Bu çalışmada düşük güç ve yüksek hızlı uygulamalar için mos akım modlu mantık devreleri kullanılmıştır. Daha sonra akım modlu mantık devreleri kullanılarak küçük bir kütüphane oluşturulmuş ve bu kütüphane içindeki devreler çeşitli performans ihtiyaçlarına göre (çıkış gerilimi, gerilim kazancı, akım) optimize edilmiştir. Bu devreler kullanılarak işaretli 16 bit çarpıcı devresi yapılmıştır.

Çarpıcı devresi mimarisinde, kısmi çarpanları yarı yarıya azaltmak için Booth kodlaması kullanılmıştır. Ayrıca kısmi çarpanların toplamasını kısa bir sürede yapmak için Wallace ağaç yapısı kullanılmıştr. Wallace ağaç yapısında tam toplayıcılar ve yarım toplayıcılar kullanılmıştır. En son olarak final toplayıcısında 32 bit hibrid toplayıcı kullanılmıştır.

0.18 mikrometre UMC teknolojisi kullanılmıştır. Dört değişik kaynak akımı ile çalışan çarpıcı devresi tasarlanmıştır. Birincisi 800 MHz'te çalışmakta, 55mW güç harcamaktadır. İkincisi 600 MHz'te çalışmakta, 37mW güç harcamaktadır. Üçüncüsü 400 MHz'te çalışmakta, 27mW güç harcamaktadır. Sonuncusu ise 250MHz'te çalışmakta ve 16mW güç harcamaktadır. Üçüncü çarpıcı devresi aynı teknolojide çalışan devreden, aynı frekansda % 25 daha az güç harcamaktadır [14]. Ayrıca ikinci modda çalışan çarpıcı devresi yine aynı teknolojide çalışan devreden aynı güç sarfiyatında % 50 daha hızlı çalışmaktadır [14]. Güç kaynağındaki akımın değişme oranı sadece % 4 dür ve çarpıcı devresi 7268 transistör içermektedir. Literatürdeki benzer CMOS çarpıcı devresi 13444 transistör içermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

mW	Milliwatt
ns	Nanosecond
μs	Microsecond
μ_n	NMOS carrier mobility
V_{dd}	Power supply voltage
ΔV	Voltage swing
MHz	MegaHertz
MCML	Mos Current Mode Logic
ECL	Emitter Coupled Logic
DSP	Digital Signal Processing
PDN	Pull Down Network
DCVSL	Differential Cascode Voltage Switch Logic
DSL	Differential Split-Level Logic
R	Resistance
C	Capacitance
ΔV	Voltage Swing
W	Width
L	Length
BDD	Binary Decision Diagram
V_{RFP}	Reference Voltage For The PMOS Load Transistors
V_{RFN}	Reference Voltage For The NMOS Current Source Transistors

1. INTRODUCTION

The recent advances in VLSI technology have allowed rapid growth in the area of portable electronic devices. Laptop computers, cellular phones, and personal desktop assistants have all become commonplace items in people's lives.

One of the primary consumer complaints of these devices is the short battery life and/or the extra weight of the batteries due to the high power consumption of the circuitry. As CMOS process technology scales down and demand for more processing power increases, it can be shown that the power consumption of future IC's will increase over time if significant architectural changes are not made. It is therefore critical in future circuits that power be minimized beyond the traditional constraints of packaging cost and heat dissipation. Also the speed should be minimized.

As device density increases, it is also extremely desirable to integrate analog and digital circuitry onto the same die for many DSP and communications systems. High levels of integration will be required in order to reduce total system area and decrease production costs. This integration has been delayed due primarily to the difficulty in designed high precision analog circuitry in the presence of extremely hostile digital noise. These difficulties will also increase as process technology scales due to fundamental challenges in high precision analog design at low supply voltages in digital CMOS technology. Either significant advances in analog design techniques will be required or digital designers will be forced to adapt their design style or process technology.

A digital circuit style that seems to be promising in both reducing power consumption, delay and providing an analog friendly environment is MOS Current Mode Logic (MCML). While bipolar CML, a derivative of emitter coupled logic (ECL), has been used for years in high performance applications, it has become less desirable over time due to its high static power consumption and reliance on bipolar processing [3].

MCML is preferred for mixed analogue-digital signal environments in order to reduce the digital inference between the analogue and digital blocks. MCML architectures provide higher immunity to supply noise due to their differential structure, lower cross talk due to the reduced output voltage swing and lower generated noise level due to the constant current flowing through the supply rails. The constant current source used in MCML is the reason for constant power consumption, which is independent from the frequency of operation or gate activity. The power consumption is independent from the frequency because the two branches are driven symmetrically and in opposition of phase [24].

MCML can be used multi-GHz communications systems. The applications of MCML are : MuX/Demux ICs for Sonet/Sdh optic – fiber links [26], [27], [28]; high-speed crosspoint switches for network (LAN/WAN) applications [29]; RF applications (PLL, prescalers, circuits for clock recovery, VCOs...) [30], [31], [32], [33], [34]; very high-speed buffers/links [35].

MCML is based on differential pair inputs and draws a constant current from the power supply. By applying a suitable voltage swing on the differential input, the constant current can be steered from one branch of the circuit to the other. Due to this constant current from supply to ground, the switching noise is reduced and since the operation of MCML is based on the differential pair, problems to supply voltage fluctuations are reduced as well.

In [1], MCML was analyzed and a 64-bit adaptively pipelined adder was developed and simulated. It was demonstrated in that paper that MCML could dissipate less power than equivalent CMOS circuitry as well as adjust for clock skew and environmental or process variations.

In this project, an analysis of MCML is presented. Near-minimum sized transistors are used in this project. A small MCML cell library is developed and it is used for implementation of a high performance 16 bit non-pipelined signed multiplier. The multiplier is tested with 4 different reference voltages.

For the multiplier architecture; modified radix 4 booth encoding is used to reduce the number of partial products by half. Also, Wallace tree structure is used to sum the partial products in reduced time. In the Wallace tree, only full adders and half adders are used. For the completion adder, a fast and low power 32 bit hybrid adder is used: ripple carry for bits (7:0), variable block carry-select for bits (23:8) and conditional carry for bits (31:24). This hybrid architecture enables power reduction with no performance penalty in the final adder compared to a conventional high-performance adder.

In the literature, there are only two MCML multipliers [12], [13]. Both of them are 8 bit multipliers. Since they are 8 bit multipliers, Booth encoding was not used in these multiplier structeres. But in this project, I design a new 16 bit MCML multiplier which uses Booth encoding.

Design is done in the 0.18µm UMC technology. A 16 bit non-pipelined MCML signed multiplier is designed and tested with 4 different reference voltages. The first one works at 800 MHz, consumes 55 mW. The second one works at 600 MHz, consumes 37mW. The third one works at 400 MHz, consumes 27mW. The last one works at 250 MHz, consumes 16 mW.

The power of the third one is 25 per cent less than the CMOS multiplier with equivalent operating frequency given in reference [14]. The speed of the second one is 50 per cent faster than the CMOS multiplier with equivalent power given in reference [14]. It can be seen that, using MCML in high frequencies is much more advantageous than CMOS. Because, power dissipation is independent from the frequency in MCML. So the power dissipation of MCML at high frequencies is lower than CMOS.

The power supply current spike is only 4 per cent of the nominal current. Because every MCML gates have a constant current from power supply to ground. My 16 bit MCML multiplier consists of 7268 transistors. The 16 bit CMOS multiplier [14] consists of 13444 transistors and the 16 bit Pass Transitor Logic multiplier [25] consists of 12349 transistors. So, the number of transistors is reduced by using MCML. Because some MCML gates such as XOR or MUX require less number of transistors than the CMOS gates. For example, a MCML full adder consists of 24 transistors (Figure 5.11 and Figure 5.12.) However a CMOS full

adder consists of 28 transistors [9]. 121 MCML full adders is used in my 16 bit multiplier. Also MCML booth selector consists of only 19 transistors (Figure 6.3 and Figure 5.10.) But a CMOS booth selector consists of 32 transistors [10]. I used 136 MCML booth selectors in my 16 bit multiplier.

2. MCML GATE DESIGN

2.1. Ideal Gate-Operation and Theory

In order to understand the issues in designing with real MCML gates, it is beneficial to first derive some of the properties and equations of a general, ideal MCML gate. This ideal gate is presented in Figure 2.1 below and consists of three main parts: the pull up resistors, the pull down network switch, and the current source.

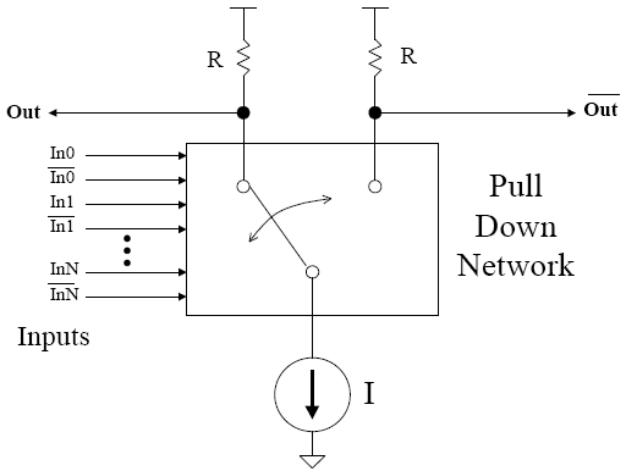

Figure 2.1. Basic MCML gate [3]

The inputs to the pull down network (PDN) are fully differential. In other words, the true and complement off all logical inputs must be presented to the gate. The PDN can implement any logic function but must have a definite value for all possible input combinations. In general, the design of the MCML pull down network is similar to other differential logic styles such as differential cascade voltage switch logic (DCVSL) or differential split-level logic (DSL) [2].

Unlike DCVSL or DSL, the pull down network in MCML circuits is regulated by a current source. The current, I, flows through only one of the resistor in Figure 2.1. and generates a voltage swing, $\Delta V = I \times R$. The other resistor will not have any current flowing

through it and its output node will be pulled up to V_{dd} in steady state. At the differential output voltage, the total voltage swing is set exclusively by the amount of current (I) and the value of the pull up resistance (R). This voltage swing is generally much smaller than V_{DD}, of the order of a few hundred millivolts.

With this simple model in mind, we can derive some basic transient properties for a circuit composed of MCML gates. For simplicity, let's assume that our circuit is a linear chain of N identical gates, all with identical load capacitance C on each output node. The total propagation delay of the chain of gates will be proportional to:

Delay: $$D_{MCML} = NRC = \frac{N \times C \times \Delta V}{I} \qquad (2.1)$$

The power consumption of a digital gate is typically composed of its static and dynamic components. In the case of MCML, the sum of the static and dynamic power dissipations is constant. Because the current from the voltage supply, V_{dd}, is a constant. Since the current is not a function of the switching activity and the switching speed, the amount of power consumption is always the same and hence the sum of the dynamic and the static power must be constant. The result of this fact is that $dI/dt \approx 0$ and no switching noise exists on the power lines ideally. With this assumption, we can write expressions for power, power-delay, and energy-delay [2]:

Dissipated Power: $$P_{MCML} = N \times I \times V_{dd}$$

Power-delay product: $$PD_{MCML} = NIV_{dd} \times \frac{NC\Delta V}{I} = N^2 \times C \times \Delta V \times V_{dd}$$

Energy-delay product: $$ED_{MCML} = N^2 C \Delta V V_{dd} \times \frac{NC\Delta V}{I} = \frac{N^3 \times C^2 \times V_{dd} \times \Delta V^2}{I}$$

(2.2)

For comparison, the delay, power, power-delay, and energy-delay for static CMOS logic are well known and approximated by [4]:

Delay: $$D_{CMOS} = \frac{N \times C \times V_{dd}}{\frac{k}{2} \times (V_{dd} - V_t)^{\alpha}}$$

Dissipated Power: $$P_{CMOS} = N \times C \times {V_{dd}}^2 \times \frac{1}{D_{CMOS}}$$

Power-delay product: $$PD_{CMOS} = N \times C \times {V_{dd}}^2$$

Energy-delay product: $$ED_{CMOS} = N^2 \times 2 \times \frac{C^2}{k} \times \frac{{V_{dd}}^2}{(V_{dd} - V_t)^{\alpha}} \quad (2.3)$$

where *k* and *a* are process and transistor size dependent parameters. Note that the above equations assume that the CMOS circuitry is being clocked at a frequency equal to the inverse of the propagation delay. So MCML dissipates a nearly constant amount of power, independent of the clock frequency or input switching activity.

One interesting property to note is that MCML circuits do not have a theoretical minimum to the energy-delay product whereas the CMOS circuits do. A designer can arbitrarily reduce the ED product by increasing the current for a given C, V_{DD}, and voltage swing. In reality, this is not possible for very large currents because the robustness of the circuitry will deteriorate if no other changes are made.

Possibly the most important conclusion from the above equations comes from the effect of logic depth, N. The performance of MCML gates in relation to CMOS decreases linearly with N. This is due to the fact that MCML consumes static power, even when not switching. It is very important therefore in MCML circuits to maintain a shallow logic depth. In slowly clocked circuits, CMOS will not consume as much power as MCML, but in circuits with high performance requirements, MCML can have significantly better power-delay or energy-delay.

Another interesting property is that the energy-delay is proportional to the square of the voltage swing for MCML. This fact encourages the use very low swing circuits. Once again, the limiting factor is the robustness of the circuitry.

For mixed signal environments, the constant current supplied by V_{DD} is extremely desirable. The dI/dt effects are negligible in comparison to CMOS circuits and the current variation is theoretically 0. There will be some current change during switching due to non-idealities, but the change is less than 5 per cent in circuits simulated. The circuits are also significantly more robust against power supply noise.

2.2. MCML Inverter

The first real circuit to analyze is the MCML Inverter/Buffer shown in Figure 2.2. Since MCML is a differential logic style, the buffer and inverter function are identical topologically and only require switching of the output or input sense.

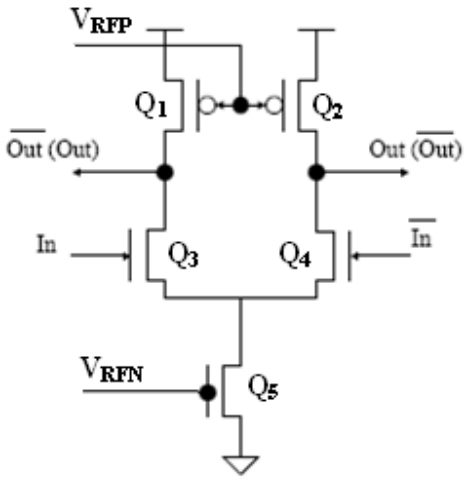

Figure 2.2. MCML inverter/buffer

The pull down network switch is implemented with a standard nmos differential pair controlled by the single input. The current source is an nmos device with a fixed gate voltage (V_{RFN}=Reference voltage for The NMOS current source) working in the saturation region. The load resistors are pmos devices with fixed gate voltages (V_{RFP}=Reference voltage for the PMOS load transistor) and are designed to be operated in the linear region in order to model resistors.

The goal of the NMOS differential pair is to switch the current provided by the current source from one side to the other. Ideally, all current will only travel down one path and the

"off" path will have zero current flowing through it under DC conditions. In reality, some current will always flow in the "off" path and cause a reduction in the true signal voltage from V_{dd}. The quality of current switching increases with larger input voltage difference (V_{id}) or larger W/L of the PDN transistors and decreases as larger currents are used.

The current source for MCML circuits is implemented with a single nmos device. While several different architectures are known for current sources [5] (e.g. cascoding), a single device implementation was decided upon for area efficiency. It is important to maintain a relatively small transistor so that total cell size is not dominated by the current source. It is also desirable to use a non-minimum length device for this current source in order to achieve higher output impedance and better current matching across gates.

The load resistances are implemented with single PMOS devices. It is desirable to make these devices as close to minimum size as possible, unlike standard analog circuits. Increasing the width of these devices will decrease the linearity and also increase the capacitance.

2.3. Other MCML Gate Topologies

All MCML gates have one current source device and two load devices. Different logic functions are implemented with different pull down networks. The pull down networks are identical to those used in ECL logic and are composed of sets of differential pairs.

The implementation of a logic function can be determined immediately from a creation of a Binary Decision Diagram (BDD). BDD's are used extensively in the area of logic synthesis and CAD to visualize boolean optimizations and can also be used in determining MCML gate structure. A general analysis of the formation and optimization of BDD's is beyond the scope of this thesis. Instead, we will look at a single example.

Let's try to implement the following function in MCML:

F = ABC + B'D + ACD + ABC'

We can begin to factor this expression until we have a completely specified and fully factored equation:

F = A[BC+B'D+CD] + A'[B'D+BC']
F = A[B(C+CD) + B'(D+CD)] + A'[B(C') + B'(D)]
F = A[B(C) + B'(D)] + A'[B(C') + B'(D)]

The BDD for this expression is shown in Figure 2.3a. and the implementation of the pull down network is shown in Figure 2.3b.

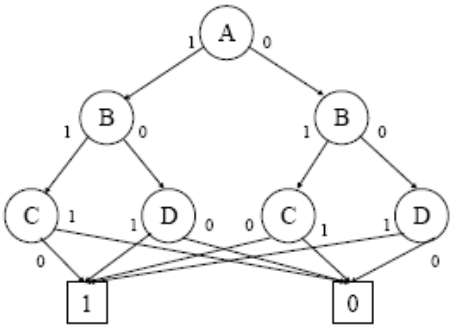

Figure 2.3a. Binary decision diagram [3]

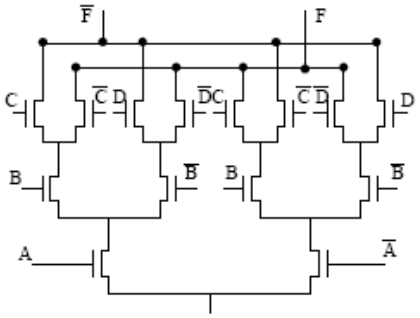

Figure 2.3b. MCML pull down network [3]

Since it is desirable to reduce the logic depth of the nmos pull down circuitry to preserve both DC and transient properties, only functions of three levels or less are considered. While general BDD algorithms can achieve optimized trees for any logic function, many of the well known functions can be easily created by hand. Several of these functions are shown in Figure 2.4 with their corresponding current sources and pull up devices included.

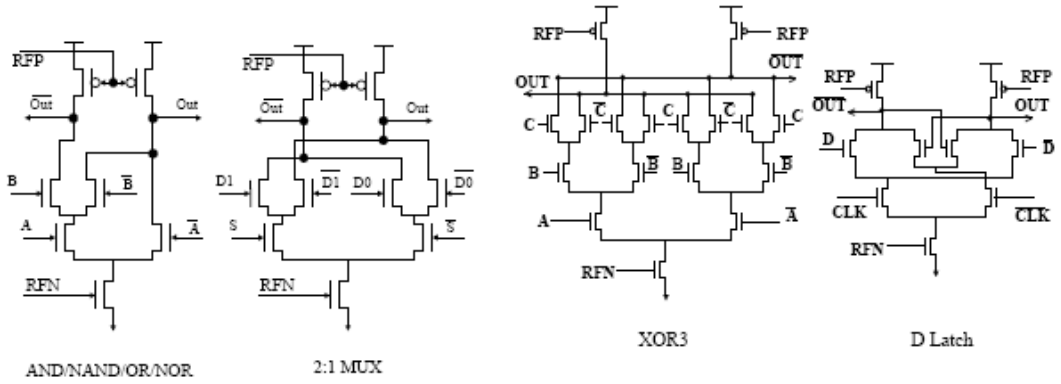

Figure 2.4. MCML gate examples [3]

One interesting property to note is the relative homogeneity of gate topologies. If we look at the leftmost gate in Figure 2.4, we can see that the AND, OR, NAND, and NOR functions all have the exact same topology and therefore the same sizing, delay, power, etc. The only difference in implementing these functions is the ordering of the inputs and outputs. This uniformity leads to more predictability in the timing and area of cells and reduces the need for boolean manipulation in order to transform into inverting logic.

The basic storage element used for MCML is the D-Latch shown above. The XOR and MUX gates shown above have a fairly compact structure compared to equivalent static CMOS implementations and are expected to perform well.

3. MCML GATE OPTIMIZATION

An optimization problem exists for static CMOS design but is much better understood. The only two real parameters which effect gate performance in CMOS are V_{dd} and transistor sizes. These two parameters are typically chosen independently and general guidelines for transistor sizing are well known. In contrast, MCML optimization has many more degrees of freedom in the parameter selection. Furthermore, the parameters tend to be tightly coupled and do not allow independent selection.

3.1. Simulation Methodology

I did simulations at the transistor schematic level in Mentor Graphics using the UMC 0.18μm process. Schematics were then netlisted and all simulations were performed in Eldo.

3.2. Hard Constraints

Hard constraints place a limit on some performance metric which must not be violated. The voltage gain, V_{RFN} and V_{RFP} voltage limits are the hard constrainst in my thesis.

3.2.1. Gain

In standard CMOS circuits, one of the main qualities of robustness to noise is the mid-swing DC voltage gain [3]. Digital logic can only function correctly if there exists a point in the DC transfer curve where the gain is larger than 1. There are two primary reasons for the gain to be made larger than the absolute minimum: regeneration and bi-stability. Regeneration is the ability for a gate to produce an output voltage closer to the ideal voltage level than its input voltage. Bi-stability is a requirement in latches and flip-flops and assures that there are only two stable logic states in the system. Both of these metrics are satisfied by high DC voltage gains.

Standard CMOS circuits naturally have large mid-swing voltage gains. In simple circuits simulated, gains of greater than 60 can be achieved with no additional effort. In constrast, MCML circuits do not naturally have high gains. High gains can be achieved but at a tremendous cost in area and performance. Therefore, it is critical to design at or near the minimum requirements for voltage gain.

Furthermore, MCML circuits do not suffer from the same noise constraints as CMOS circuits. Most of the noise which adversely affects CMOS circuits is common mode noise. However, in MCML this noise is rejected by the differential logic. MCML circuits also generate significantly less switching noise than CMOS circuits and the environment will therefore be more suitable to low gain operation.

The lower limit on voltage gain for this project was set at 2. The requirement is really that the gain should be greater than 1 for all process and voltage conditions. So I chose the voltage gain of all MCML circuits in my thesis as 2. This is a sufficient value for these variations [42]. I chose the sizes of the transistors of MCML circuits to give a voltage gain of two.

3.2.2. V_{RFN} and V_{RFP} Voltage Limits

V_{RFN} signal sets the gate voltage on the nmos current source. Therefore it needs to be kept a few hundred millivolts from both V_{dd} and from ground. V_{RFP} voltage is used to set the pmos load gate voltages. The constraint on V_{RFP} is that the total V_{sg} of the pmos devices must be kept below the process limit of 1.8V.

3.3. Design Parameters

3.3.1. V_{dd}

The only true upper bound on the supply voltage is due to the process limits (1.8V for our selected technology) but it is desirable to lower V_{dd} in order to reduce the power

consumption. The power consumption of the circuitry is linearly proportional to the supply voltage and it should therefore be reduced as much as possible.

The main lower limit on the power supply voltage comes from the nmos current source. Reducing V_{dd} too far reduces the output impedance of the current source and eventually pushes it out of the saturation region. One effect of this degradation is that the current in the gate is reduced. Another effect is that the voltage gain (A_v) is reduced. I chose V_{dd} as 1.8V.

3.3.2. Voltage Swing (ΔV)

As seen before, it is extremely desirable to reduce the voltage swing as much as possible in order to reduce the propagation delay of MCML. The lower limit on the voltage swing is determined by the gain and current switching requirements. As the voltage swing is reduced, the mid-transition output voltages become closer to V_{dd} and reduce the output impedance of the pmos loads. The steepness of the current switching will also be reduced. These degradation of the gain can be fixed by adjusting other parameters such as transistor sizes or V_{dd}.

The possible circuit mismatch effects must be also taken into account to determine the lover bound of ΔV. The upper bound on the voltage swing comes from the nonlinearity of the pmos loads. As the voltage swing is increased, the pmos device on the side is pulled down and moves closer to its Vdsat voltage. This leads to eventual entering to the saturation region and extreme nonlinearity. This can be adjusted by increasing the length of the pmos device but, this increases the propagation delay. Another upper bound on voltage swing comes from the nmos current source. If the voltage swing is too large, the pull down side will approach to the ground and force the current source out of saturation. So there is a tradeoff among linearity, gain, and speed when setting the voltage gain.

I did not apply any optimization program to find ΔV and other design parameters. But in reference [44] an automated optimization-based design strategy is proposed for MCML circuits to overcome the complexities of the design procedure. The proposed

design methodology in reference [44] determines the values of the design variables that achieve minimum power dissipation while attaining the high speed. I used the results of this optimization program as reported in [44] applying my hard constraints.

I chose the voltage swing as 400 mV for all gates by looking at the results of the optimization program in [44] applying my hard constraints. Also this ΔV value enables high speed for the current values in Section 3.3.3.

3.3.3. Current (I)

The upper bound of current is set by the maximum transistor sizes allowed for a "minimum" sized cell. The current is controlled by V_{RFN}. Increasing the current, increases power, but decreases the delay. So there is a tradeoff among power and delay.

I set the currents 44μA, 31μA, 21μA and 13μA by changing V_{RFN} voltage. Because 44μA is the maximum current for my transistor sizes. My MCML circuits do not work at larger currents than 44μA. Also 13μA is the minumum current for my transistor sizes. My MCML circuits do not work at smaller currents than 13μA. I chosed the other current values atbitrarily.

3.3.4. Differential Pair Transistor Sizes

The sizing of transistor lengths and widths is the design parameter with the greatest degree of freedom and effects almost all performance criteria. In order to limit the scope of the discussion, we will first make a few assumptions. First, while each transistor width and length could be independently varied, we assume that all differential pair transistors are matched to be the same size. The second assumption is that the length of all transistors in the pull down network will be kept to the minimum (0.18μm) since there is almost no benefit from increasing the length.

In general, increasing the width of the differential pair transistors will increase the voltage gain but it will also increase the input and output capacitance. This leads to a direct

tradeoff between performance (both delay and area) and robustness (voltage gain). It is desirable to use the smallest transistors possible in order to achieve enough voltage gain.

It is evident that gain must be kept at a minimum in order to preserve performance. By changing the transistor widths from 0.5µm to 1.5µm, the input capacitance increases dramatically (~3x) and the performance of the gate driving this gate decreases.

In multi-level gates (the gates with two or three inputs), the number of differential pairs to be sized also increases. The definition of voltage gain also changes in multi-level gates and we define the overall voltage gain as the gain from the worst case input combination. It is very difficult to come up with general design rules for multi-level gates due to the fact that the effects on gain are extremely inter-dependent among levels. So I did not change the sizes of differential transistors for multi-level gates.

I chose the width (W) of the differential pair transistors as 0.7µm for the gates with one input (buffer,inverter), 1µm for the gates with two inputs (2 inputs and/or/xor, 2 to 1 mux) and 1.2µm for the gates with 3 inputs (xor3,carry output of full adder). I set the lenght (L) 0.18µm for all gates. I did not design gates with four or more inputs. By chosing the widths and lenghts as here for differential pair transistor sizes, I achieved a voltage gain of 2 and high speed operation.

3.3.5. PMOS Load Transistor Sizes

Optimizing the size of the pmos load transistors is one of the most difficult and nonlinear tasks in creating a good MCML gate. Besides the obvious area tradeoffs, the main performance criteria affected by the sizing of these transistors are the voltage gain and the propagation delay.

The voltage gain is increased by increasing the length of the pmos devices. This effect is especially strong when increasing from minimum length, and therefore, non-minimum length transistors are used if possible for these devices.

The ratio of the width and length (W/L) of the devices also affects several criteria. Increasing the W/L, either by increasing W or decreasing L, decreases the effective resistance of the load devices and therefore improves the propagation delay. If the width is increased, the capacitance is also increased at the output node and the propagation delay may in fact stay the same. The actual effect will be heavily dependent on the amount of load capacitance.

Increasing the W/L of the devices also reduces the Vdsat voltage and increases the non- linearity of the resistance. The W/L must be kept large enough so that enough DC resistance can be achieved for a given voltage swing.

I chose width (W) of the pmos devices as 0.55µm and lenght (L) as 0.24µm. I changed the resistance value of pmos devices by changing V_{RFP}. I achieved enough DC resistance for a 400mV voltage swing, minumum delay and a voltage gain of 2 by setting the sizes of the pmos load transistors in my MCML circuits given in Table 3.1.

3.3.6. NMOS Current Source Transistor Sizes

The principal tradeoff in the selection of the current source transistor sizes is between area and robustness. It is desirable to use non-minimum length devices for the current sources to increase the output impedance. It is also desirable to have a large W/L to decrease the Vdsat voltage and allow for further reduction in V_{dd}. The limit on increasing both the length and width is that the area of the gate begins to grow dramatically.

The current source devices used in this project were chosen to have W = 1.6µm and L = 0.5µm. The currents are changed from 13µA to 44µA by changing V_{RFN}. These parameters enable my MCML circuits to have a reasonable area and robustness.

As a result, design parameter optimization results are tabulated in Table 3.1. My hard constaints are voltage gain of 2 and V_{RFN} and V_{RFP} Voltage limits.

Table 3.1. The optimization results of the design parameters

Parameter Name	Value
V_{dd}	1.8V
ΔV	400mV
Currents	44µA for V_{RFN} = 0.76V and V_{RFP} =80mV 31µA for V_{RFN} = 0.7V and V_{RFP} =0.4V 21µA for V_{RFN} = 0.65V and V_{RFP} =0.64V 13µA for V_{RFN} = 0.6V and V_{RFP} =0.82V
Differential Pair Transistor Sizes	W=0.7µm L=0.18µm for the gates with one input W=1µm L=0.18µm for the gates with two inputs W=1.2µm L=0.18µm for the gates with three inputs
PMOS Load Transistor Sizes	W=0.55µm L=0.24µm
NMOS Current Source Transistor Sizes	W=1.6µm L=0.5µm

4. MCML vs CMOS

It is developed both classical CMOS Logic Inverter and MOS Current-Mode Logic Inverter to see the differences between these circuits. It is a useful and appropriate example since the inverter is a basic element for every technology used in IC design.

4.1. Classical CMOS Logic Inverter

This inverter uses two MOS transistors, a PMOS and an NMOS, coupled as shown in Figure 4.1. These two transistors operate as switches that have a small resistance when on and an infinite resistance when off.

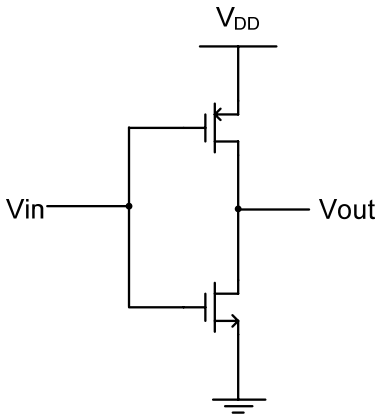

Figure 4.1. Classical CMOS logic inverter

When V_i is equal to V_{DD}, which is a logic 1, the PMOS transistor is cut-off , and we can consider the NMOS transistor as a resistor, as it operates in the linear or triode region. We can calculate the output resistance of the NMOS if we know the size and transconductance of it.

$$r_n = 1/\left[k_n\left(\frac{W}{L}\right)\left(V_{dd} - V_{tn}\right)\right] \tag{4.1}$$

In equation 4.1, r_n is the output resistance of the NMOS transistor, k_n is the transconductance of the NMOS transistor, W is width the of the NMOS transistor, V_{dd} is the power supply voltage and V_{tn} is the threshold voltage of the NMOS transistor.

When V_i is connected to ground, which is a logic 0, the NMOS transistor is cut-off, and the PMOS transistor is operating in the triode region. This time we also calculate the output resistance of the PMOS [6].

$$r_p = 1/\left[k_p\left(\frac{W}{L}\right)\left(V_{dd} - \left|V_{tp}\right|\right)\right] \tag{4.2}$$

Figure 4.4. shows the graph of V_0 (output voltage) versus V_i (input voltage) of a CMOS inverter. In this figure, sat means saturation region, triode means resistive region of transistors.

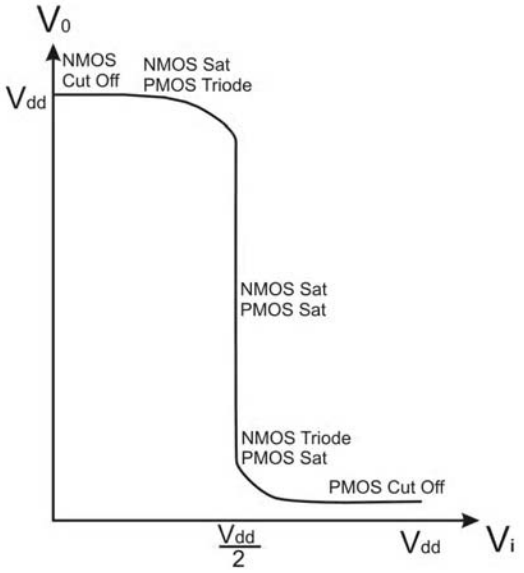

Figure 4.2. Transfer characteristic of a classical CMOS inverter [6]

The dynamic power disipation can be calculated if we determine the switching frequency and the transistor size:

$$P_{dynamic} = fCV_{dd}^{2} \tag{4.3}$$

Figure 4.3. shows the graph of power supply current, I, versus input voltage,V_i of a CMOS inverter.

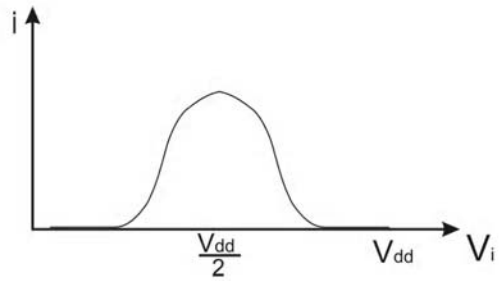

Figure 4.3. Current to input voltage for the classical CMOS inverter [6]

We can summarize the basic properties of a classical CMOS inverter as follows :

- V_{SWING} =V_{DD} (the maximum).
- Ideally the static power dissipation is zero.
- The dynamic power dissipation depends on the transistor sizes, frequency and V_{DD}.
- Output signal is 0 or 1 and it is independent of W/L.

4.2. MOS Current-Mode Inverter

This inverter uses five MOS transistors, two PMOS and three NMOS, connected as shown in Figure 4.4. The inputs and outputs are voltages. Here the current source (I) is made of a single NMOS transistor (Q_5) and a reference voltage (V_{RFN}). Also the resistance, R is generated by two PMOS transistors (Q_1 and Q_2) and a reference voltage (V_{RFP}). Therefore the voltage swing :

$$\Delta V = R \times I \tag{4.4}$$

The remaining two NMOS transistors (Q_3 and Q_4) are used to construct the differential amplifier to obtain the logic function.

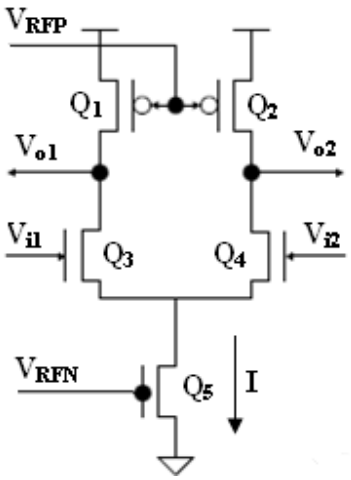

Figure 4.4. MOS current-mode logic inverter

When the inverter is operating, PMOS transistors work in the triode region, as resistors. The voltages of PMOS transistors are:

$$V_{sg} = V_{dd} - V_{RFP}$$
$$V_{sd} < V_{sg} \times I \qquad (4.5)$$

Here V_{sg} is source to gate voltage of the PMOS transistors, V_{sd} is source to drain voltage the PMOS transistors and V_{sg} is source to gate voltage of PMOS transistors.

The current (I) will be divided between the two NMOS transistors. The amount of current flowing through the transistor [7] :

$$i_{D1}(v_i) = \begin{cases} 0 & \text{if } v_i < -\sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \\ \dfrac{I}{2} + \dfrac{v_i}{2}\sqrt{\mu_n C_{ox}\dfrac{W_n}{L_n} I - \left(\mu_n C_{ox}\dfrac{W_n}{L_n}\dfrac{v_i}{2}\right)^2} & \text{if } |v_i| \le \sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \\ I & \text{if } v_i > \sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \end{cases}$$

$$i_{D2}(v_i) = I - i_{D1}(v_i) \tag{4.4}$$

Here, W_n, and L_n are the effective NMOS transistor channel width and length, C_{ox} is the oxide capacitance per area, μ_n is the NMOS carrier mobility, v_i is V_{i1}- V_{i2}, i_{D1} is the current which is passed through the NMOS transistor Q_3 and i_{D2} is the drain current of the NMOS transistor Q4.

As PMOS transistors work in the linear region, their equivalent resistance will be written as follows :

$$R = \frac{V_{sd}}{i_{D1} + i_{D2}} \tag{4.6}$$

The output voltage transfer characteristics of this inverter can be calculated by R. The output voltage V_0 is :

$$V_o = V_{o1} - V_{o2} = -R \times (i_{D1} - i_{D2}) \tag{4.7}$$

We stated the transistor current equations before, so we can evaluate the transfer characteristic of a MCML inverter as follows :

$$V_0(v_i) = \begin{cases} RI & \text{if } v_i < -\sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \\ -v_i RI\sqrt{\mu_n C_{ox}\dfrac{W_n}{L_n} I - \left(\mu_n C_{ox}\dfrac{W_n}{L_n}\dfrac{v_i}{2}\right)^2} & \text{if } |v_i| \le \sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \\ -RI & \text{if } v_i > \sqrt{\dfrac{2I}{\mu_n C_{ox}\dfrac{W_n}{L_n}}} \end{cases} \tag{4.8}$$

So the transfer characteristic of MCML inverter is shown in Figure 4.7. Here V_i is V_{i1}-V_{i2} and V_o is V_{o1}-V_{o2}. V_{swing} is the voltage swing (ΔV) and NM is the noise margin.

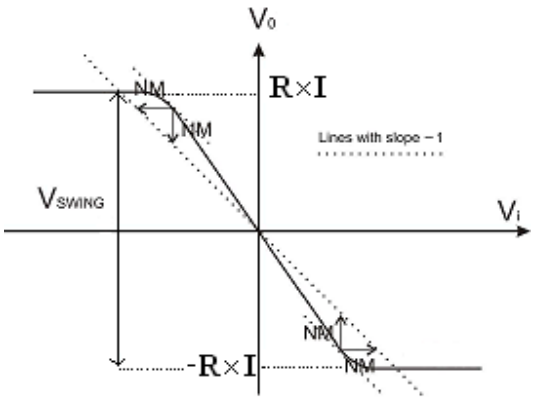

Figure 4.5. Transfer characteristic of a MCML inverter [7]

We can summarize the basic properties of the MCML inverter as follows :

- $V_{SWING} = RI$.
- Static power dissipation $P = V_{dd}I$.
- Output signal is dependent of W/L, because ΔV depends on the parameters of PMOS load and MOS current source transistors.

4.3. Advantages and Disadvantages of MCML

4.3.1. Advantages:

- Less number of transistors in some logic gates due to differential circuit topology.
- Faster than CMOS due to low V_{SWING}.
- Adopted in multi-GHz communications systems.
- Weak dependence of propagation delay on fanout load capacitance.
- Robust performance and better noise immunity.
- Does not generate significant switching noise.
- More power efficient than CMOS at high operating frequencies.
- Suitable for low-power DSP ICs [36].
- Very low switching noise (di/dt noise ≈0).
- Used in mixed-signal high-accuracy ICs [37].
- Less sensitive to supply noise, better signal integrity.

4.3.2. Disadvantages:

- Static power dissipation is worse than CMOS due to a constant current from power supply to ground.
- More elaborated design process due to adjusting a desired voltage gain and V_{SWING}.
- Larger interconnect area due to differential routing.
- Lower voltage gain than CMOS.

5. MCML GATES LIBRARY

In order to design arithmetic circuits, I first created a library cell which consists of mos current mode logic gates (and, mux, xor etc). I drew schematics of these gates in Mentor IC Studio and simulated in Eldo Spice. I used UMC 0.18µm technology and sized the transistors to the optimization rules which are explained in section 3.

After simulation in Eldo, I looked at the input and output waves in EZwave program and found the delays using measurement tools of EZwave. This tool finds the t_{pHL} (high-to-low output transition), t_{pLH} (low-to-high output transition and delay.

I simulated gates for different currents. I changed the currents by changing the V_{RFN} and adjusted the voltage swing by changing V_{RFP}. Because V_{RFP} is used for adjusting the resistance and V_{RFN} is used for adjusting the current. Multiplication of resistance and current gives us voltage swing. I used 400mV voltage swing (ΔV) for all the gates taking into consideration design optimization rules. Higher current means higher power but less delay.

The currents are:

- 44µA for V_{RFN} =0.76V and V_{RFP} =80mV
- 31µA for V_{RFN} =0.7V and V_{RFP} =0.4V
- 21µA for V_{RFN} =0.65V and V_{RFP} =0.64V
- 13µA for V_{RFN} =0.6V and V_{RFP} =0.82V

I drawed the schematics of MCML gates library and I used these gates except D latch and 4 to 1 mux gates to design my 16 bit multiplier. I tabulated the delays of all these gates and showed the input and output voltage and current waves of the buffer for 44µA in Figure 5.2. I used the full adder which is in reference [8] for my library. Because in this full adder, the carry input of the sum circuit is connected to the lowest logic level to reduce the input load of the carry signal to get high performance.

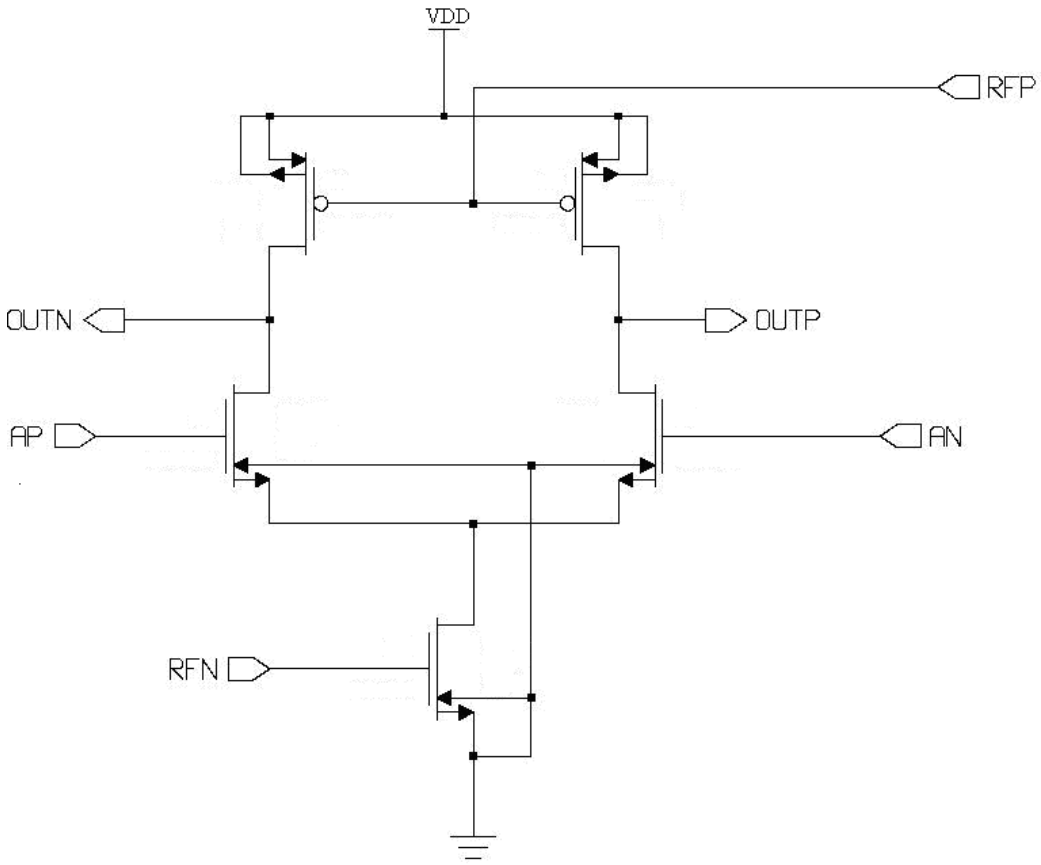

Figure 5.1. The schematic of the MCML buffer

Table 5.1. The propagation delays of the MCML inverter/buffer

	Rise Time	Fall Time	Delay
The Inverter/Buffer for 44µA	51ps	71ps	18ps
The Inverter/Buffer for 31µA	67ps	95ps	23ps
The Inverter/Buffer for 21µA	79ps	153ps	32ps
The Inverter/Buffer for 13µA	119ps	301ps	44ps

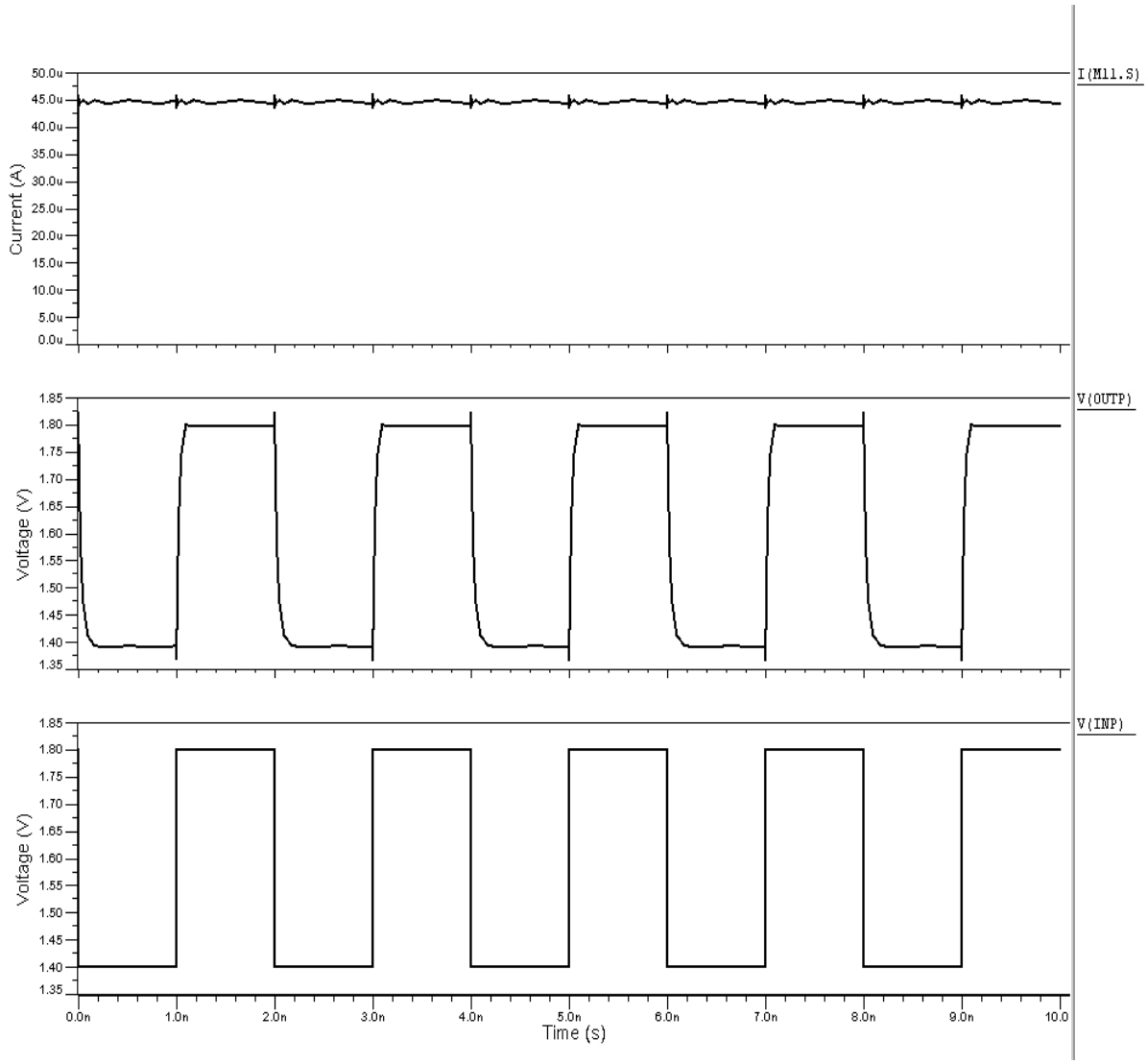

Figure 5.2. The input and output signals of the MCML buffer for 44μA

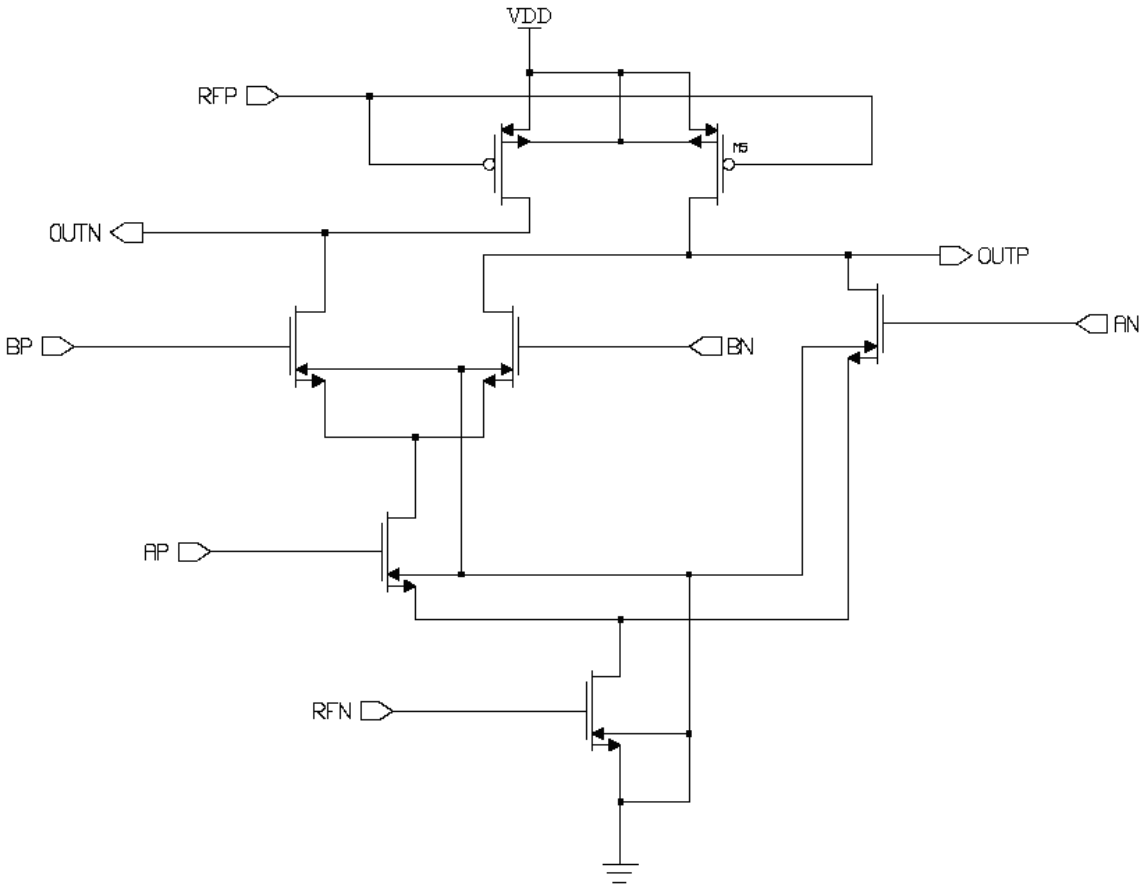

Figure 5.3. The schematic of 2 input AND

Table 5.2. The propagation delays of 2 Input AND/NAND/OR/NOR

	Rise Time	Fall Time	Delay
2 Input AND/NAND OR/NOR for 44µA	54ps	114ps	20ps
2 Input AND/NAND OR/NOR for 31µA	68ps	137ps	25ps
2 Input AND/NAND OR/NOR for 21µA	83ps	219ps	34ps
2 Input AND/NAND OR/NOR for 13µA	122ps	394ps	46ps

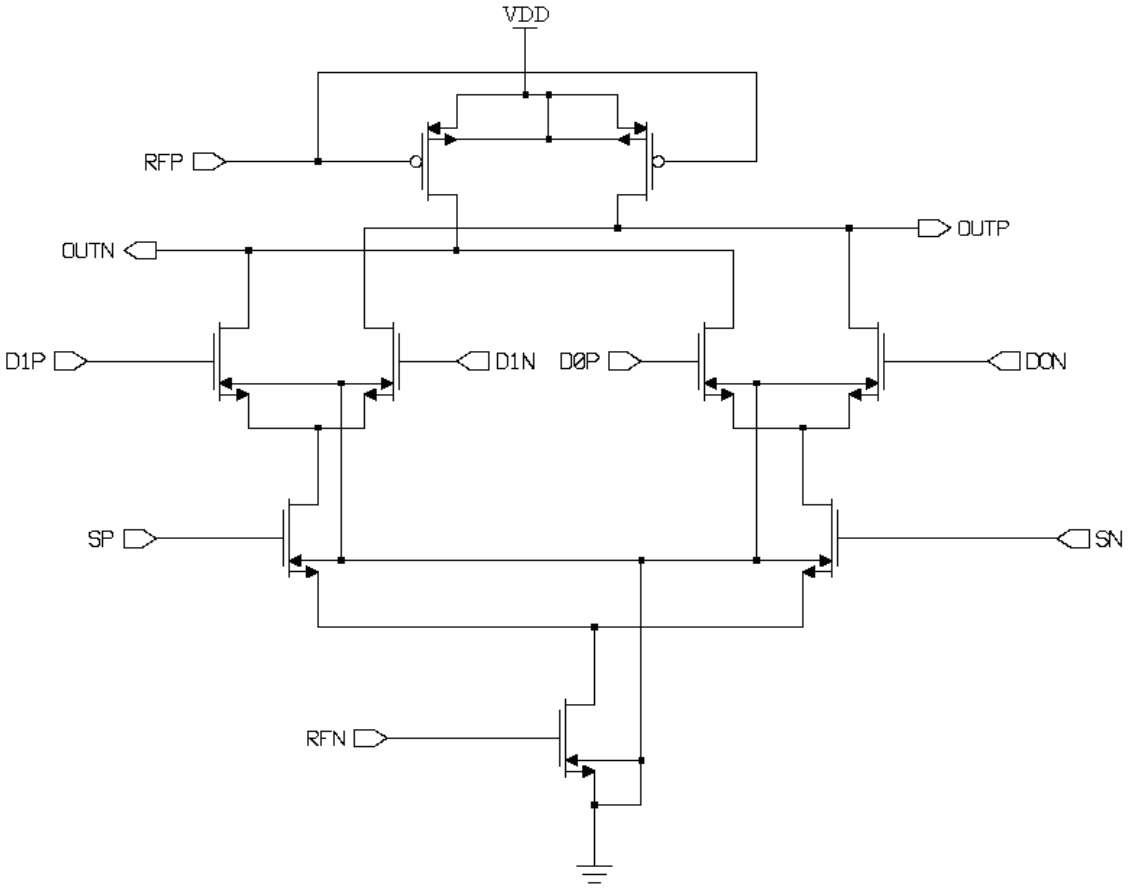

Figure 5.4. The schematic of 2 to 1 MUX

Table 5.3. The propagation delays of 2 to 1 MUX

	Rise Time	Fall Time	Delay
2 to1 MUX for 44µA	74ps	125ps	28ps
2 to1 MUX for 31µA	98ps	186ps	37ps
2 to1 MUX for 21µA	137ps	284ps	51ps
2 to1 MUX for 13µA	175ps	586ps	75ps

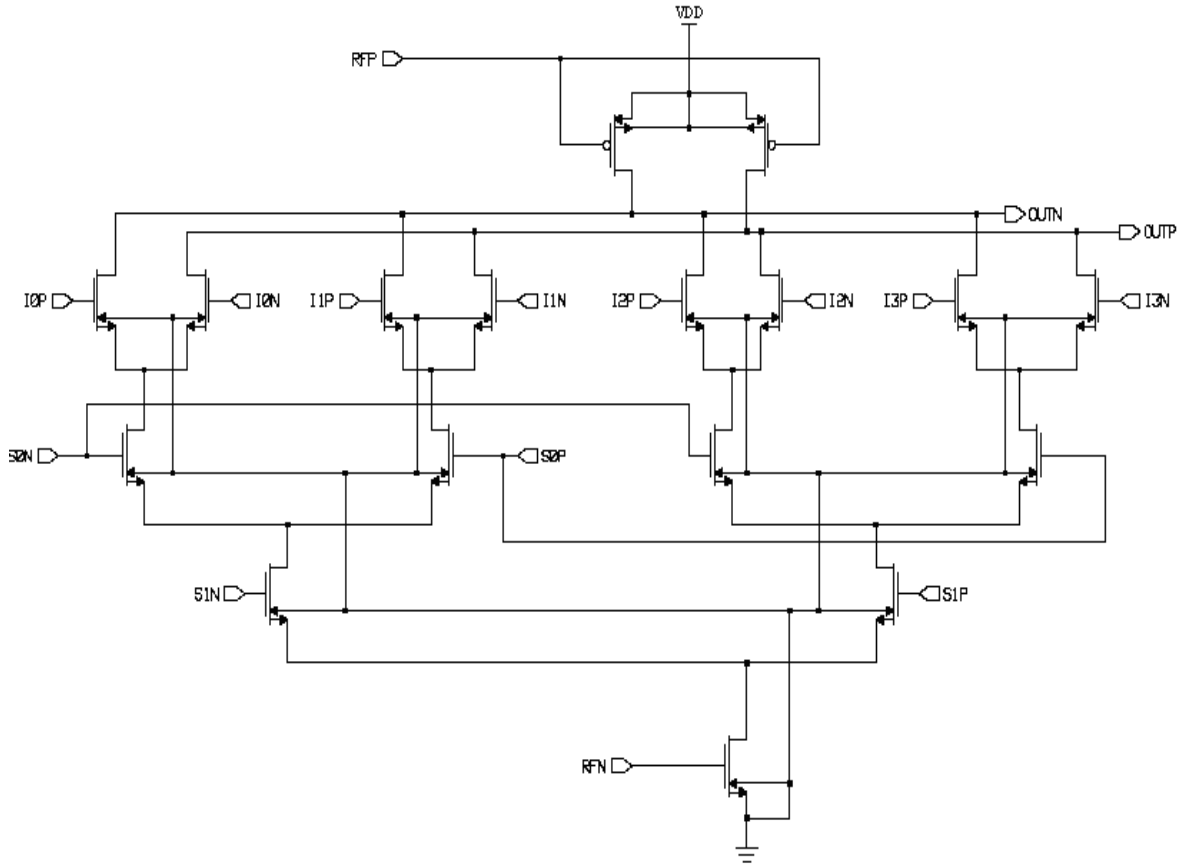

Figure 5.5. The schematic of 4 to 1 MUX

Table 5.4. The propagation delays of 4 to 1 MUX

	Rise Time	Fall Time	Delay
4 to1 MUX for 44μA	165ps	215ps	109ps
4 to1 MUX for 31μA	219ps	306ps	169ps
4 to1 MUX for 21μA	282ps	477ps	262ps
4 to1 MUX for 13μA	391ps	822ps	427ps

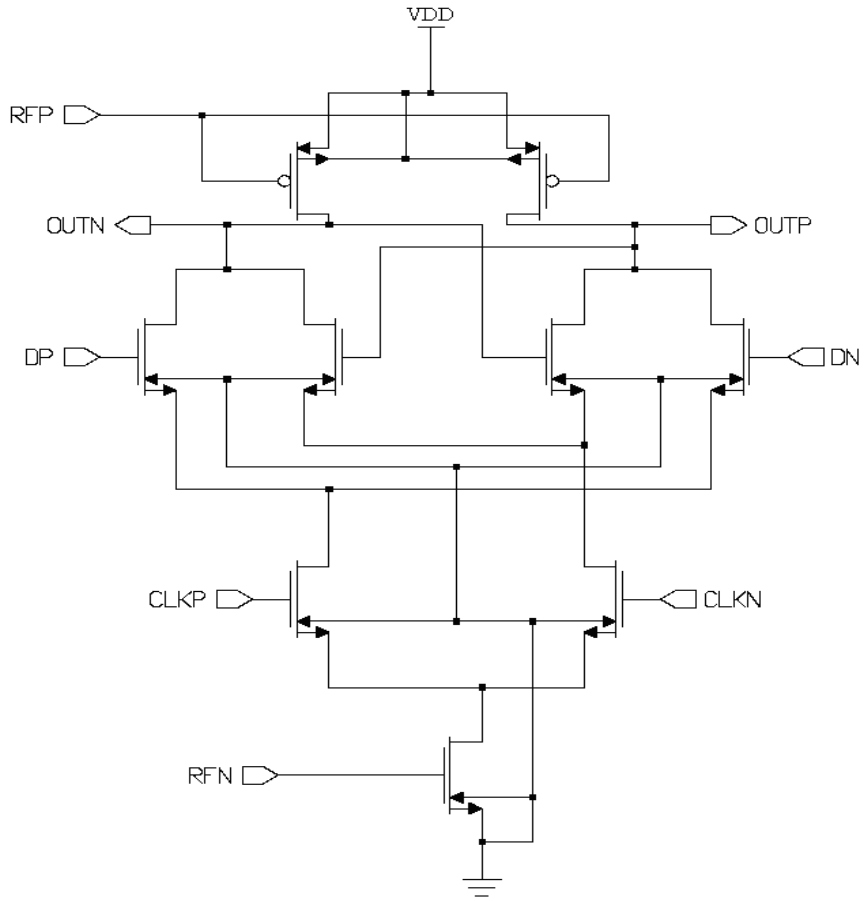

Figure 5.6. The schematic of the D latch

Table 5.5. The propagation delays of the D latch

	Rise Time	Fall Time	Delay
The D latch for 44μA	114ps	148ps	40ps
The D latch for 31μA	142ps	217ps	51ps
The D latch for 21μA	167ps	322ps	70ps
The D latch for 13μA	263ps	599ps	93ps

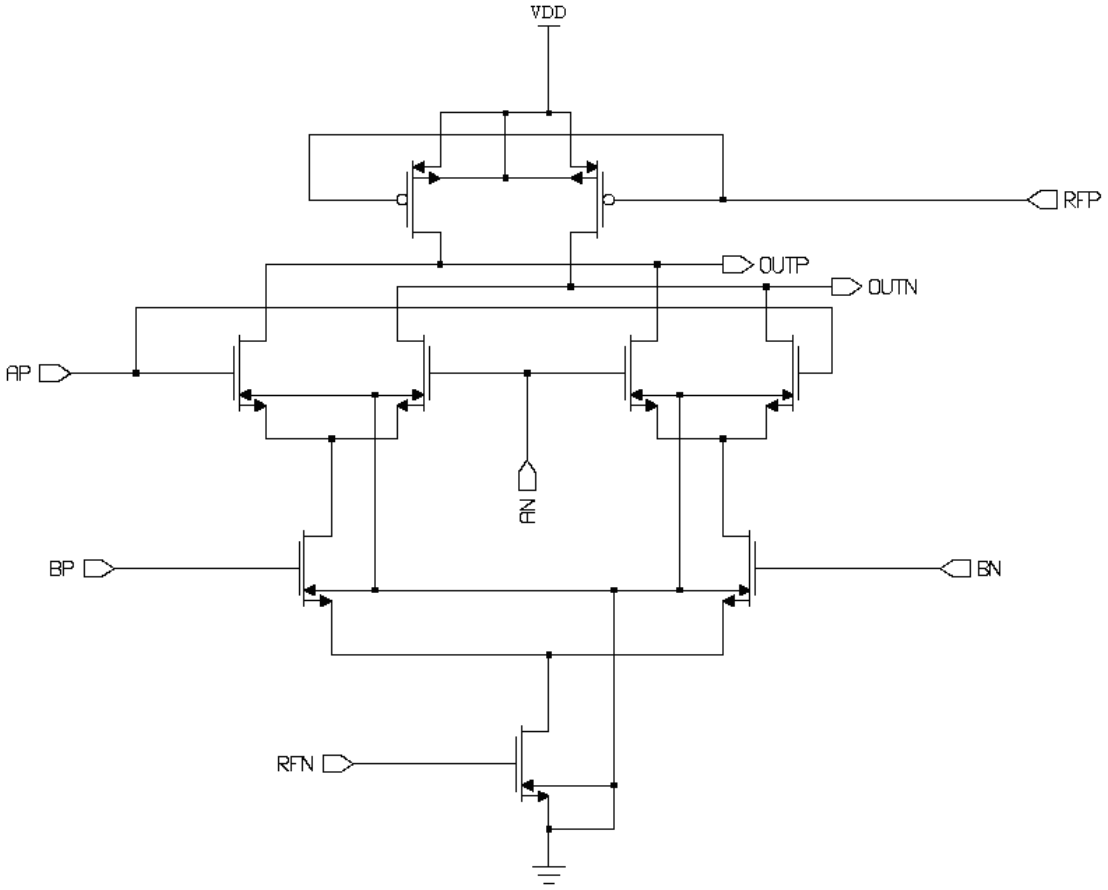

Figure 5.7. The schematic of 2 input XOR

Table 5.6. The propagation delays of 2 Input XOR

	Rise Time	Fall Time	Delay
2 Input XOR for 44μA	77ps	129ps	25ps
2 Input XOR for 31μA	97ps	186ps	33ps
2 Input XOR for 21μA	138ps	285ps	43ps
2 Input XOR for 13μA	169ps	581ps	63ps

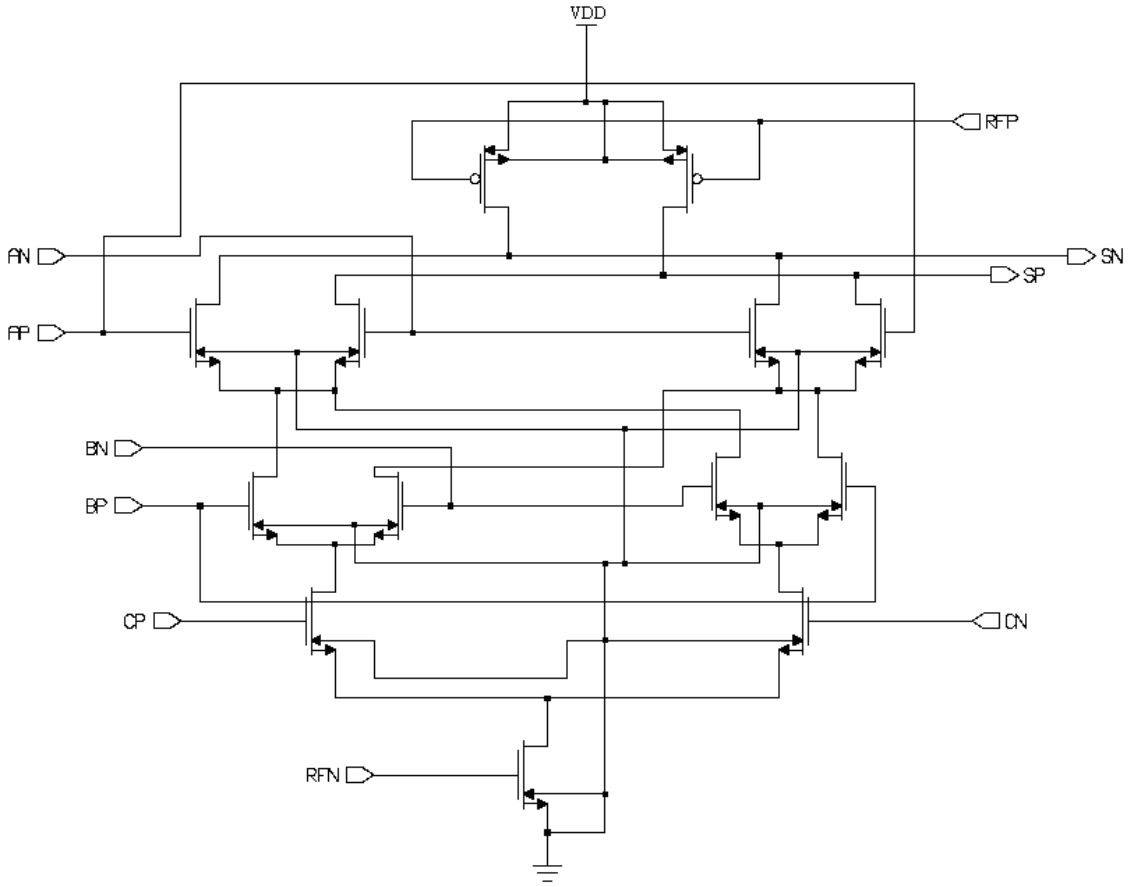

Figure 5.8. The schematic of 3 input XOR

Table 5.7. The propagation delays of 3 input XOR

	Rise Time	Fall Time	Delay
3 input XOR for 44μA	146ps	200ps	69ps
3 input XOR for 31μA	223ps	270ps	106ps
3 input XOR for 21μA	260ps	438ps	168ps
3 input XOR for 13μA	364ps	893ps	304ps

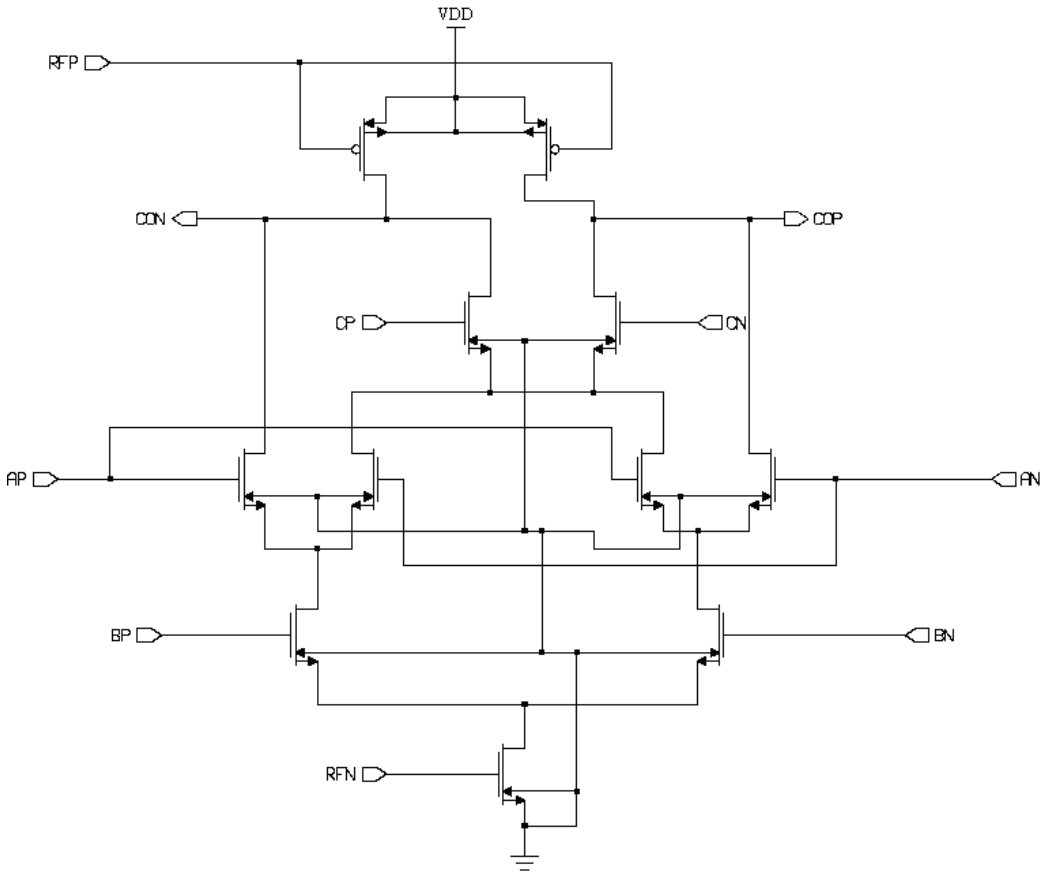

Figure 5.9. The schematic of the cary output of the full adder (AC+BC+AB)

Table 5.7. The propagation delays of the carry output of the full adder

	Rise Time	Fall Time	Delay
The carry output for 44µA	81ps	247ps	80ps
The carry output for 31µA	112ps	350ps	121ps
The carry cutput for 21µA	144ps	466ps	185ps
The carry output for 13µA	205ps	929ps	317ps

6. MULTIPLIER STRUCTURE

Digital multiplication is one of the most basic functions in a wide range of algorithms. The ubiquity of this operation in computing has given rise to a large number of multiplier implementations, each with different specifications and goals. Some applications require wide dynamic range, others need high precision, while in some cases, neither of these characteristics are very tightly specified. Digital multiplication is used as opposed to analog when high precision is an issue; it is fairly straightforward to make digital multipliers as accurate as the application requires. Precision required for multiplication varies by function. At the low end, 8 bits are needed,e.g., in image compression algorithms or 16 bits in more precise DSP tasks .At the high end, we see 53 bit and 64 bit multiplication (IEEE double precision standard). Typically, we see 16 bit multipliers used for digital signal processing and 53/64 bit multipliers used in microprocessors.

The basic operation in these design is integer multiplication; in floating point multipliers, integer multiplication units are sub-blocks of the greater floating point unit. Signed versus unsigned techniques have an impact on the design and some clever techniques have been suggested for manipulating the bit representation of numbers to generate power savings. However, the primary consideration in multipliers has been and continues to be the delay.

Short bit-width (<16 bit) 2's complement multipliers with single-cycle throughput and latency are essential ingredients of high-performance embedded processor and DSP execution cores. Key components of many DSP algorithms, such as finite-impulse (FIR) filters, infinite-impulse response (IIR) filters, discrete cosine transforms (DCTs), and fast Fourier transforms (FFTs), consist of repetitive multiplication operations that equate to over half of the total operations.These constraints require an energy-efficient multiplier that enables low compressor tree fan-outs.

Several traditional parallel multiplier schemes improve the speed proportional to the *log* of the operand length, such as Wallace and Dadda cary-save trees. A wallace tree [38]

requires $log_{3/2}$ levels of (3:2) counters to reduce the N inputs down to two carry-save redundant form outputs, where the (3:2) counter converts three inputs into two-count encoded outputs. In the Dadda tree [9], the number of counters in a compression tree is minimized. A higher order (4:2) compressor by Weinberger [10] requires $log_2(N/2)$ levels of compressor. This type of partial product tree simplifies the internal horizontal routing within the multiplier. Beyond (4:2) compressors, even higher order (9:2) compressor based partial produce trees show delay improvements [10]. Further improving the multiplier delay, Oklobdzija [11] devoloped a three-dimensional method (TDM) which appropriately connects fast/slow inputs and slow/fast outputs. This optimizes the tiles of a partial product tree as one N th-order compressor instead of individual smaller order compressors,finding a global optimum rather then a local optimum [9].

In my thesis, I designed a 16 bit signed multiplier using Wallace Tree and modified booth algorithm as can be seen in Figure 6.1.

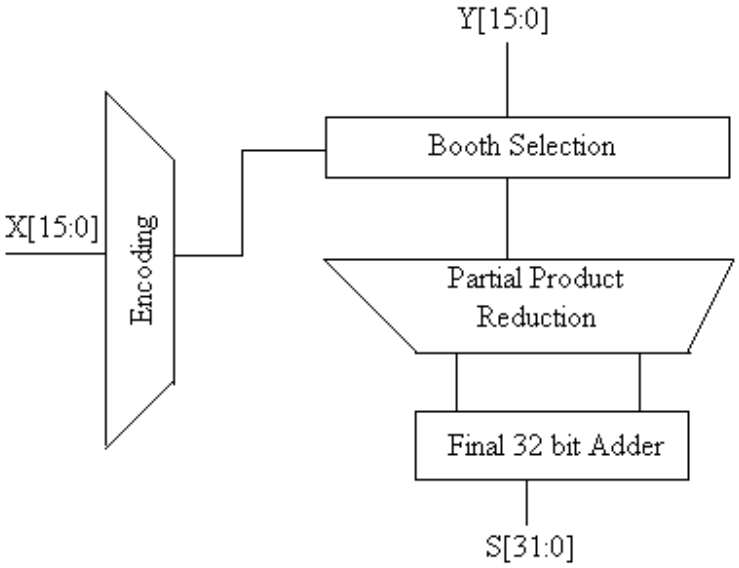

Figure 6.1. 16×16 bit multiplier organization

6.1. Booth Encoding

Booth encoding was originally proposed to accelerate serial multiplication in 1951 [39]. After then it was developed and found modified booth encoding by McScorley in 1996 [40]. Modified Booth encoding allows radix 4 parallel operation. Partial products are chosen by considering a pair of bits along with the most significant bit from the previous pair in modified booth algorithm. If the most significant bit from the previous pair is true, multiplicand bits must be added to the current partial product. If the most significant bit of the current pair is true, the current partial product is selected to be negative and the next partial product is incremented. Table 6.1. shows how the partial products are selected, based on bits of the multiplier. Negative partial products are generated by taking 2's complement of the multiplicand (possibly left-shifted by one column) [10].

Table 6.1. Radix-4 modified booth encoding values

Inputs			Partial Product	Booth Selects		
x_{2i+1}	x_{2i}	x_{2i-1}	PP_i	X_i	$2X_i$	M_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	2Y	0	1	0
1	0	0	-2Y	0	1	1
1	0	1	-Y	1	0	1
1	1	0	-Y	1	0	1
1	1	1	-0(=0)	0	0	1

In a radix-4 Booth-encoded multiplier, each group of three bits (a pair, along with the most significant bit of the previous pair) is decoded into several select lines (X_i, $2X_i$ and M_i given in the rightmost columns of table 6.1.) and driven across the partial product row as ashown in Figure 6.2.

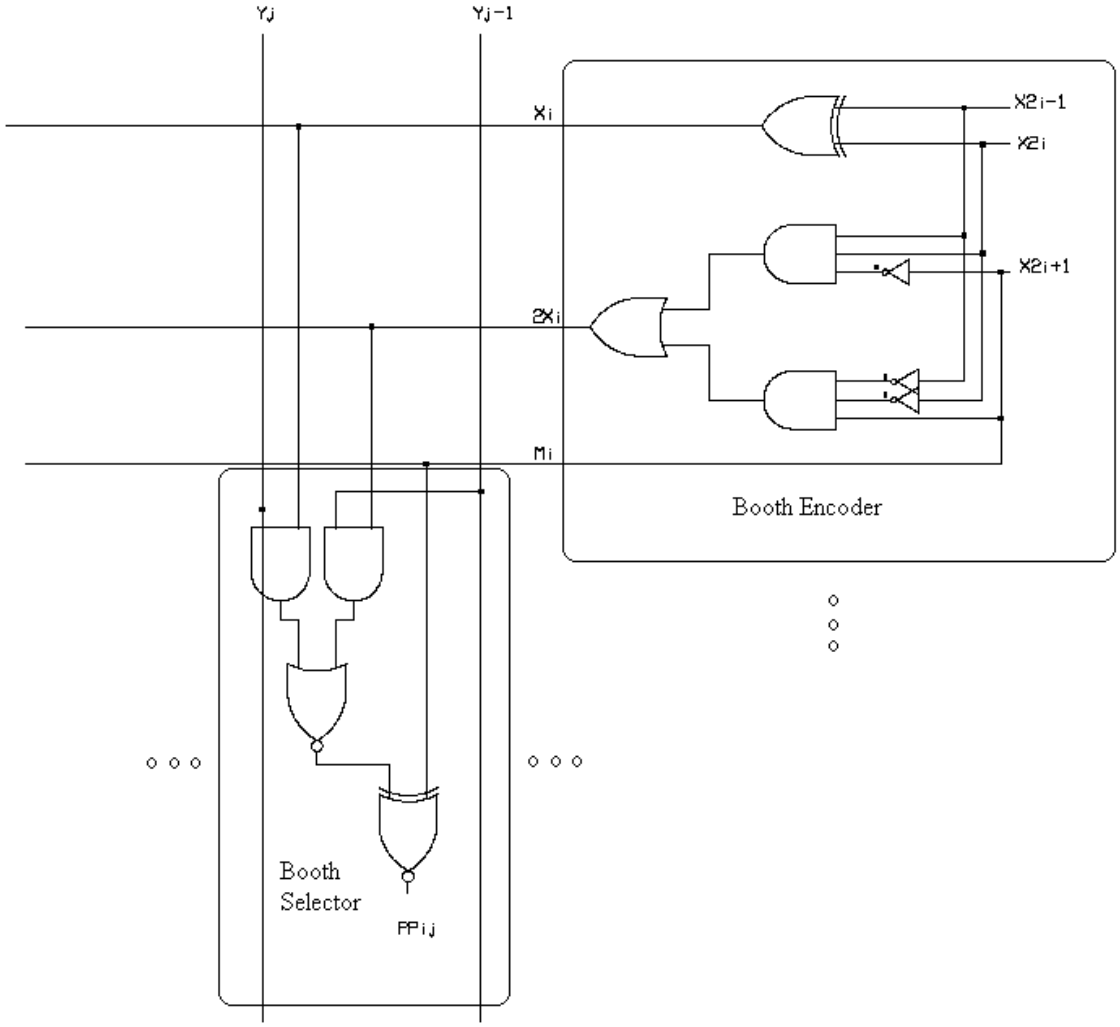

Figure 6.2. Radix-4 booth encoder and selector [10]

In the figure 6.2., the multiplier Y is distributed to all rows. The select lines control Booth selectors that choose the appropriate multiple of Y for each partial product. The Booth selectors substitute for the AND gates of a simple array multiplier. Figure 6.2. shows a conventional booth selector design that computes the *j*th partial product bit of the *i*th partial product. It was invented by Chandrakasan in 2001 [41]. Here, if the partial product has a magnitude of Y, y_i is selected. If it has a magnitude of 2Y, y_{i-1} is selected. If it is negative, the multiple is inverted (and a 1 is added to the least significant column elsewhere in the array to form the 2's complement).

In my signed 16 bit multiplier 136 booth selector and 8 booth encoder circuits is used. Booth selectors account for a subsantial portion of the area. So I optimized booth selector by using MCML. In Figure 6.3., booth selector uses 4 gates (2 AND, 1 NOR, 1 XNOR). But the same logic function which is done by using 2 AND and 1 NOR ($(y_j.X_i)+(y_{j-1}.2X_i)$) in conventional CMOS, can be done only 1 gates in MCML as can be seen in Figure 6.3. By using 1 gate instead of 3 gates, the power and area is reduced.

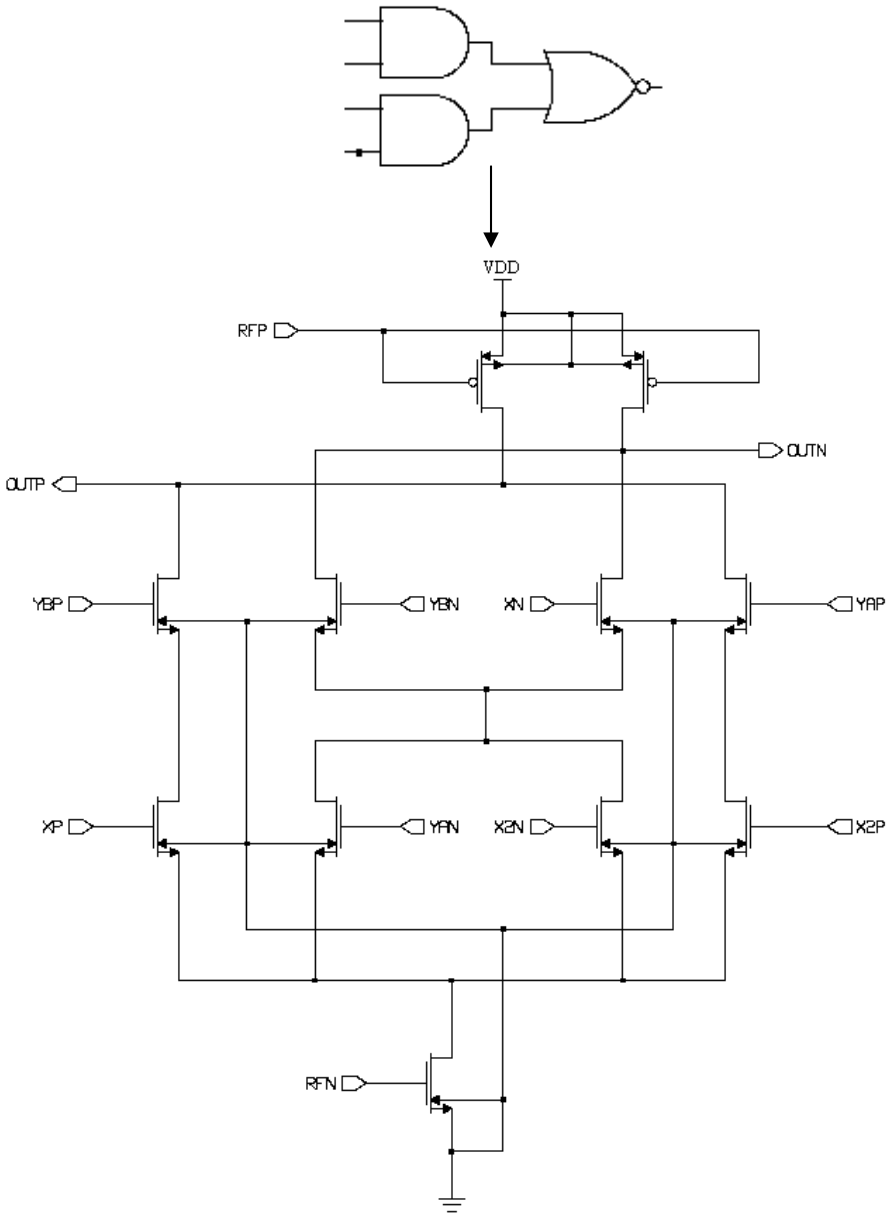

Figure 6.3. The schematic of the logic function of $(y_j.X_i)+(y_{j-1}.2X_i)$ in MCML

Table 6.2. The propagation delays of the circuit in Figure 6.3.

	Rise Time	Fall Time	Delay
Figure 6.3 for 44μA	72ps	188ps	78ps
Figure 6.3 for 31μA	96ps	248ps	114ps
Figure 6.3 for 21μA	136ps	386ps	168ps
Figure 6.3 for 13μA	174ps	702ps	283ps

I also did the booth encoder in MCML. In MCML, the logic function can be done only 1 gate and buffer, instead of using 3 gates (2 NAND and 1 OR). 8 booth decoders are used in my 16 bit multiplier.

Table 6.3. The delays of the circuit in Figure 6.4. with buffer gate

The Circuit in Figure 6.4.	Delay
Figure 6.4. for 90μA	125ps
Figure 6.4. for 62μA	170ps
Figure 6.4. for 42μA	232ps
Figure 6.4. for 26μA	356ps

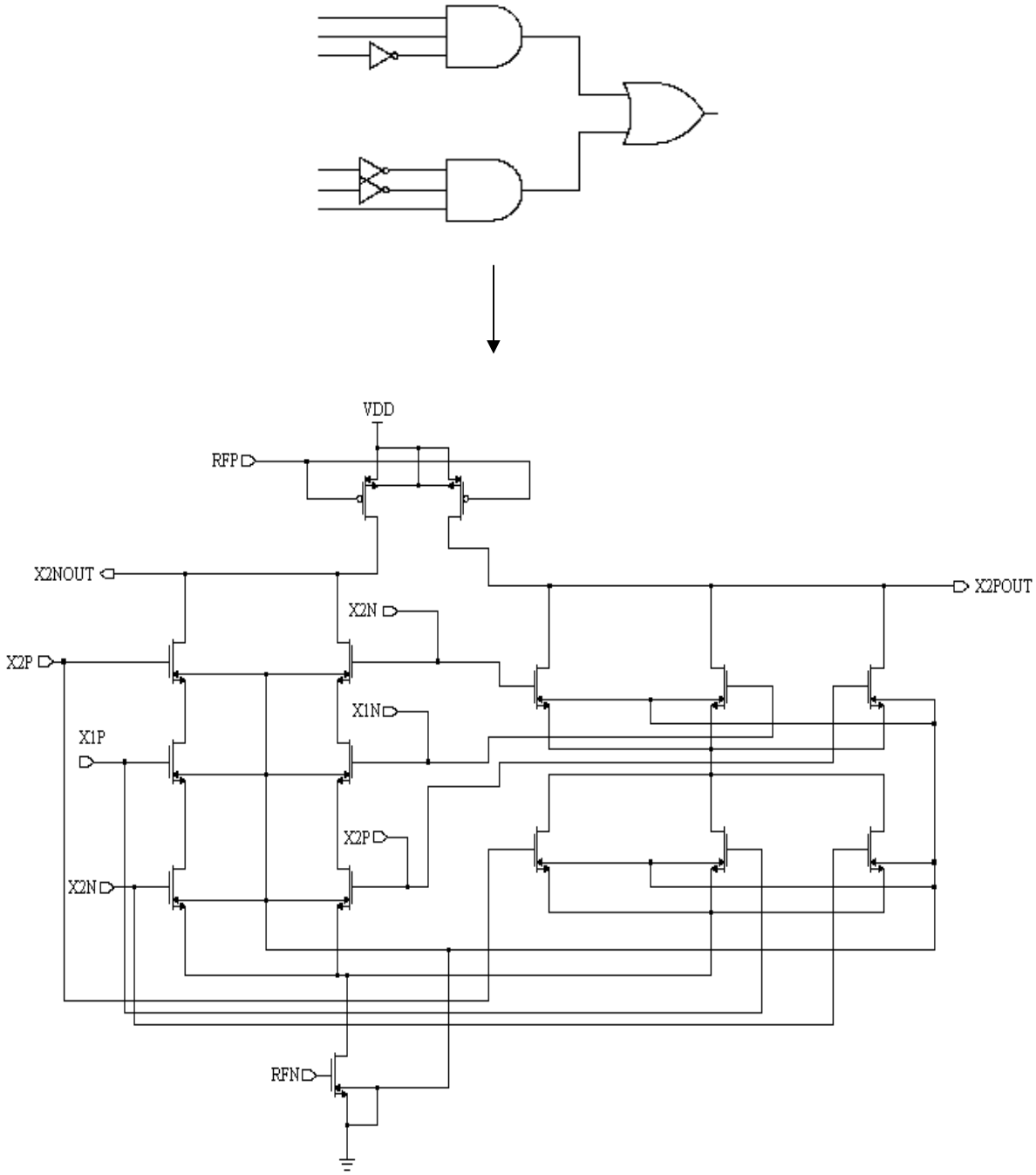

Figure 6.4. The schematic of the logic function in booth encoder

6.2. Reduced Sign Extension

Compression of the sign-extension bits in Figure 6.5. is achieved by merging the signs of the partial products with the multiplicand and pre-computing their sum (Table 6.1.) thereby removing the sign-extension bits from the critical path of the compressor tree. A partial product tree is formed when the Booth encoding generates eight sign-extended 17-bit partial products. In a conventional design, the sign extension bits are fully extended to the most significant bit of each of the eight partial product rows. This sign extension results in 30 per cent extra transistors across the boundary of the partial product tree and longer wire-loading on the Booth multiplexers. Compression of the sign-extension bits, represented by E, is achieved by first merging the signs of the partial products.S, with the multiplicand. Their sum is then pre-computed as shown in each partial product row. This removes the sign-extension bits from the critical path of the compressor tree.

The reduced sign extension results in 23 percent reduction in partial product bits (from 208 to 160), and a subsequent 15 per cent overall power reduction. The critical path through the partial product reduction tree involves the compression of 9 bits, implemented using seven (3:2) compressor circuits.

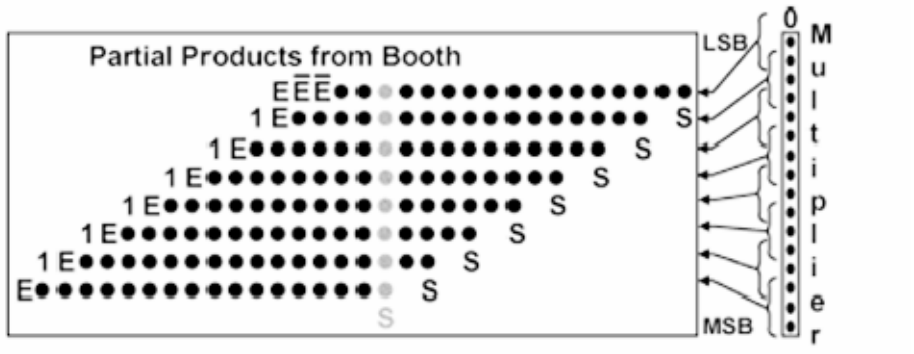

Figure 6.5. Radix-4 booth-encoded partial products for signed multiplication [9]

In this figure the sign extension bits $E = M_i \oplus y_{15}$. Here the sign bit $S=M_i=x_{2i+1}$ and y_{15} is the most significant bit of the multiplicand.

6.3. Partial Product Reduction Tree

The column addition of partial products is slow. So the partial-sum adders can also be rearranged in a treelike fashion, reducing both the critical path and the number od adder cells needed. In this tree, the sum is vertical and the carry is horizontal propogated. It is called Wallace tree.The tree multiplier realizes substantial hardware sawings for larger multipliers. The propagation delay is reduced as well. The Wallace tree requires $log_{3/2}(N/2)$ levels of (3,2) counters to reduce N inputs downt to 2 carry-save redundant form outputs. Unfortunately, the routing between levels becomes much more complicated. The longer wires have greater wire capacitance and the irregular tree is diffucult to lay out.

I used wallace tree adder for partial product reduction and I used only full adder (3:2 compressor) and half adder in wallace tree. Because compared to a high order compressor; a 3:2 compressor-based partial product tree provides the finest granularity. I tabulate the wallace tree in Table 6.4 and Table 6.5. Here HA is half Adder and FA is full adder. 2^0,2^1,2^2….. represent the partial products in Figure 6.5. At each stage the bits are compressed in the table and at last we achive 32 pieces carry-save outputs for the inputs of the final 32 bit adder.

Table 6.4. The wallace tree structure for the first 16 partial products

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
8	9	7	8	6	7	5	6	4	5	3	4	2	3	1	2
FA FA HA	FA FA HA	FA HA	FA HA	HA	HA										
6	6	6	6	6	6	5	6	4	5	3	4	2	3	1	2
FA FA	FA FA	FA FA	FA FA	FA FA	FA FA	FA HA	FA HA	HA	HA						
4	4	4	4	4	4	4	4	4	4	3	4	2	3	1	2
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	HA	HA				
3	3	3	3	3	3	3	3	3	3	3	3	2	3	1	2
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	HA	HA		
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2

Table 6.5. The wallace tree structure for the last 16 partial products.

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
1	2	2	3	3	4	4	5	5	6	6	7	8	8	8	8
									FA	HA HA	FA FA	FA FA HA	FA FA HA	FA FA HA	FA FA HA
1	2	2	3	3	4	4	5	6	6	6	6	6	6	6	6
					HA	FA	FA HA	FA FA	FA FA	FA FA	FA FA	FA FA	FA FA	FA FA	FA FA
1	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4
			HA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	HA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

HA consists of one 2 input XOR and one 2 input AND gate. FA consists of one 3 input XOR and the carry-out circuit. The outputs of this compressor adder-tree arrive at the final adder at different times in Figure 6.6.

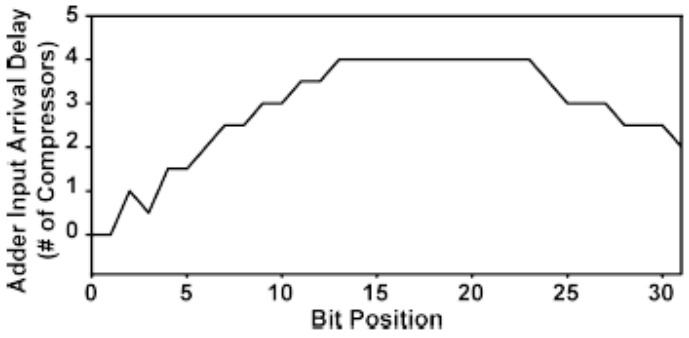

Figure 6.6. The completion adder input profile [9]

6.4. Final Adder

Figure 6.6. also shows the arrival delay variation for each input bit position to the 32-bit completion adder. The lower and upper order 8 bits arrive early, while the middle 16 bits are the most critical. By taking advantage of the uneven arrival-time profile of the compressor tree outputs, the energy consumed is minimized by the completion adder.

To exploit this delay profile, a hybrid adder architecture is used: ripple carry for bits (7:0), variable block carry-select for bits (23:8) and conditional carry for bits (31:24). This hybrid architecture enables power reduction with no performance penalty in the final adder compared to a conventional high-performance adder.

6.4.1. 8 Bit Ripple Carry Adder

I just connected the full adders for ripple carry adder. But at the first bit, I used a half Adder. Because we do not have carry input signals. Here HA is half Adder and FA is full adder.

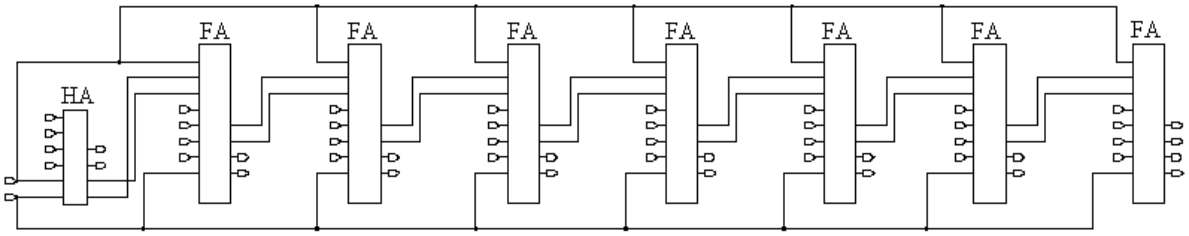

Figure 6.7. The schematic of 8 bit RCA

Table 6.6. The delays of 8 bit RCA

Ripple Carry Adder (RCA)	Delay
8 Bit RCA for 750µA	444ps
8 Bit RCA for 500µA	598ps
8 Bit RCA for 340µA	826ps
8 Bit RCA for 220µA	1200ps

6.4.2. 16 Bit Carry-Select Adder

A carry-select adder is divided into sectors, each of which – except for the least significant – performs two additions in parallel, one assuming a carry-in of zero, the other a

carry-in of one. The 16-bit carry-select adder (CSLA) in Figure 6.8., is divided into sectors of lengths 1, 2, 3, 4, and 6, proceeding from least-significant to most-significant bit to get faster. The 4-bit sector in Figure 6.9. illustrates the general principle.

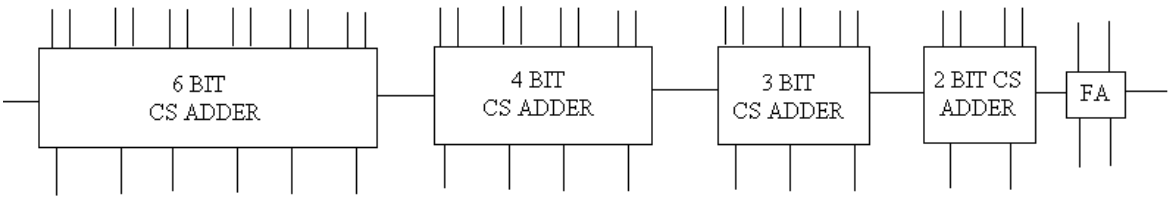

Figure 6.8. 16 bit carry-select adder

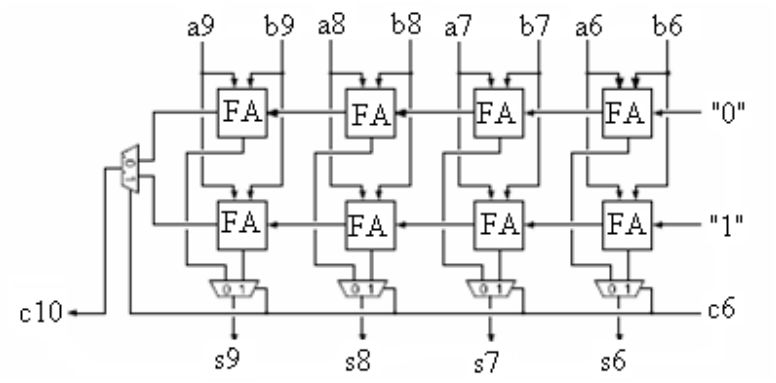

Figure 6.9. The schematic of a 4-bit sector

Within the sector in Figure 6.9., there are two 4-bit ripple-carry adders receiving the same data inputs but different carry-ins. The upper adder has a carry-in of zero; the lower adder a carry-in of one. The actual carry-in from the preceding sector selects one of the two adders. If the carry-in is zero, the sum and carry-out of the upper adder are selected. If the carry-in is one, the sum and carry-out of the lower adder are selected. Logically, the result is no different than if a single ripple-carry adder were used. The difference, of course, is in performance. Instead of having to ripple through four full adders, the carry now only has to pass through a single multiplexer.

Table 6.7. The delays of 16 bit CSLA

Carry-Select Adder (CSLA)	Delay
16 Bit CSLA for 3.7mA	511ps
16 Bit CSLA for 2.56mA	701ps
16 Bit CSLA for 1.76mA	982ps
16 Bit CSLA for 1.10mA	1471ps

6.4.3. 8 Bit Conditional-Carry Adder

I designed an 8 bit Conditional-Carry adder for the last 8 bit. Conditional-carry adder was invented by Kuo-Hsing Cheng and Shun-Wen Cheng in 2005. This adder is based on conventional conditional-sum adder but uses less muxes. So its power and delay is superior to the old one. Conditional-sum adder applies large portions of multiplexers to select the sum output data, thus making the multiplexer tree large and irregular. However the sum outputs are useless for the carry selection procedure during the multiplexer network reduction processes. So new addition rule was found to decrease the number of multiplexers by Cheng. For the 8 bit adder, classic conditional-sum adder uses 28 muxes, whereas this conditional-carry adder uses only 17 muxes in Figure 6.10. [11].

Table 6.7. The delays of 8 bit CCA

Conditional-Carry Adder (CCA)	Delay
8 Bit CCA for 1.8mA	210ps
8 Bit CCA for 1.25mA	285ps
8 Bit CCA for 0.86mA	396ps
8 Bit CCA for 0.54mA	591ps

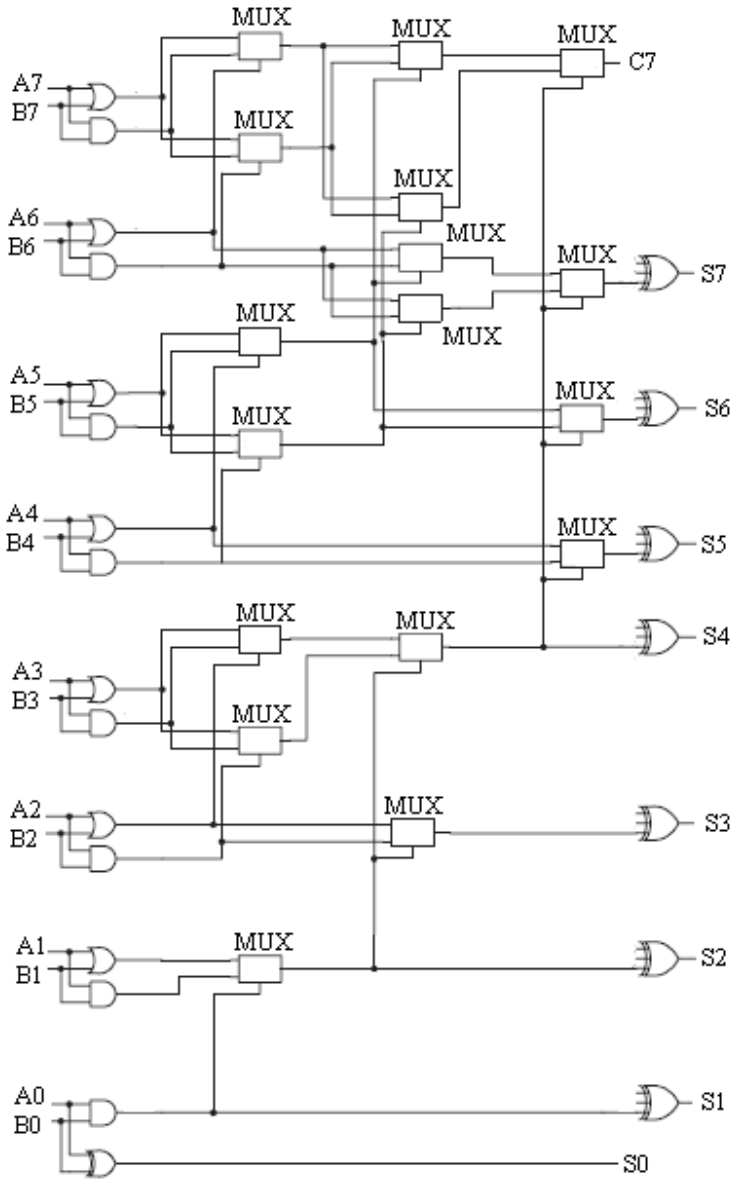

Figure 6.10. The conditional-carry adder schematic by Cheng [11]

7. SIMULATIONS AND COMPARISIONS

7.1. Simulations

I did simulations in Eldo Spice. I showed four examples of simulation for different currents. When we apply V_{RFN}=0.76V and V_{RFP}=80mV, the current which is passed through V_{dd} is in Figure 7.1. The difference between the maximum and minimum power supply current is only about 1.2 mA (31mA-29.8mA) during a switching event, which is only 4 per cent of the nominal power supply current (31mA). The power is (V_{dd}×I) 55.8mW.

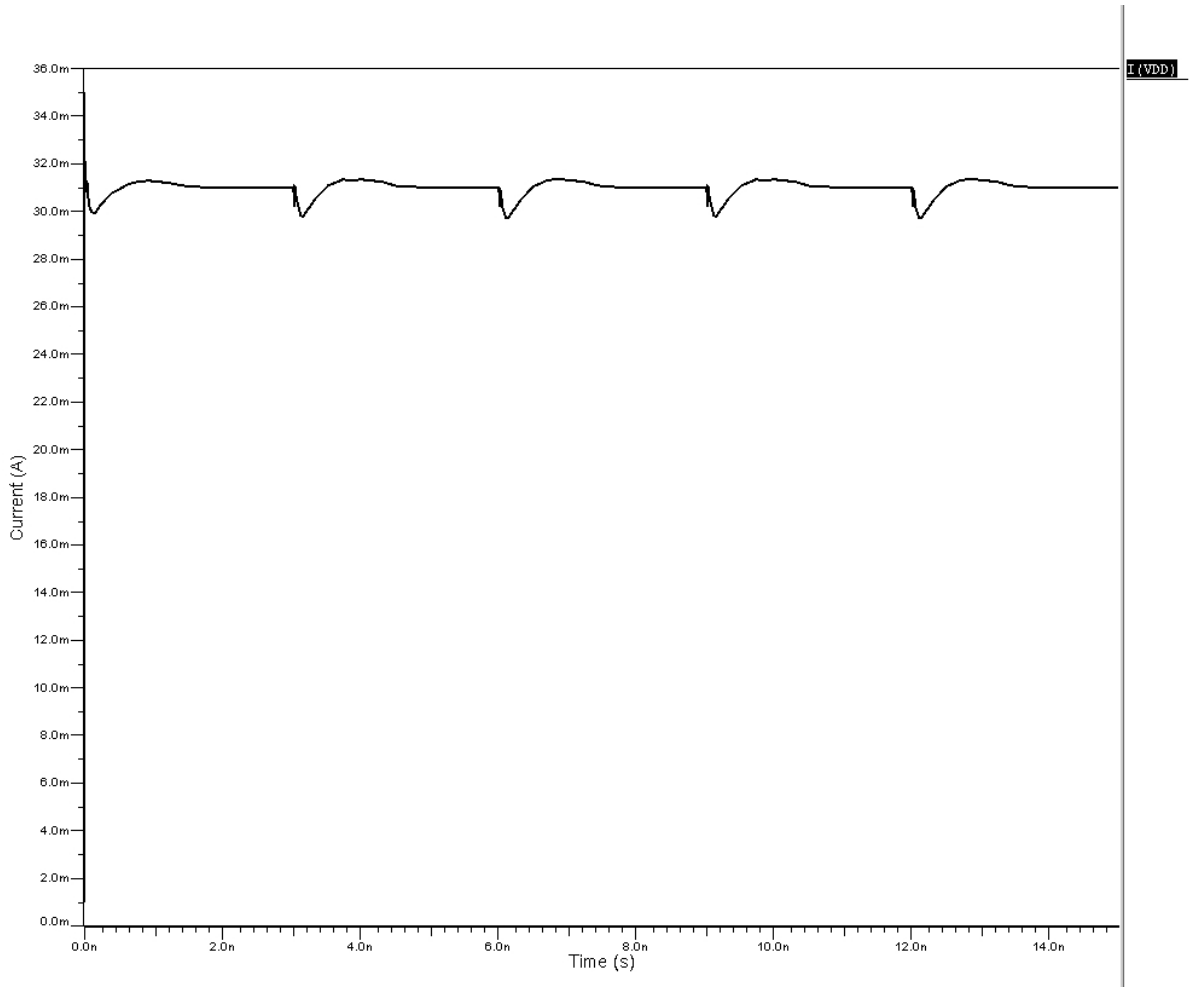

Figure 7.1. The power supply current of the 16 bit MCML multiplier for V_{RFN}=0.76V and V_{RFP}=80mV

I multiplyed two positive number in this mode (0101100101011010 and 0111001010111010) and the result is true (01010000000101011111111101100100) as can be seen in above Figures 7.2., 7.3., 7.4., 7.5., 7.6. and 7.7.

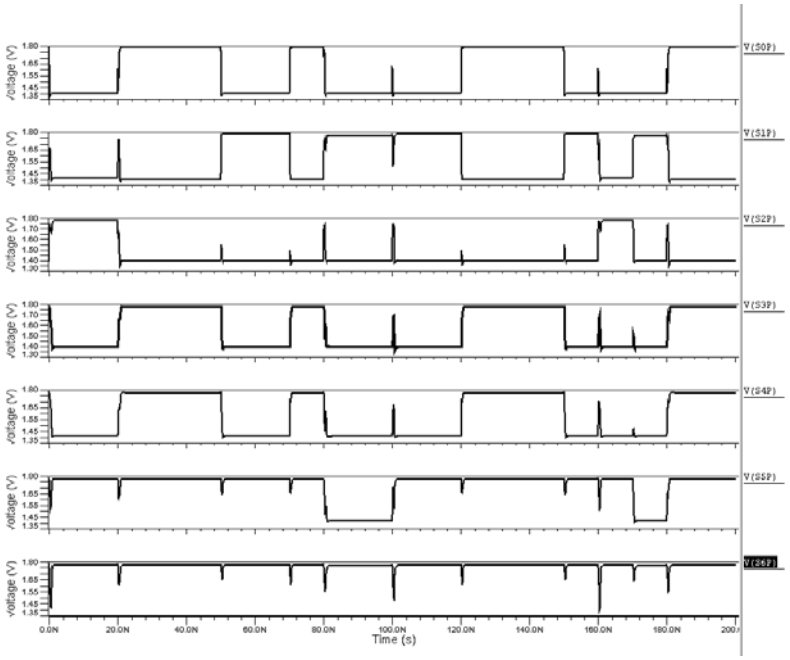

Figure 7.2. The first 7 bits of the result

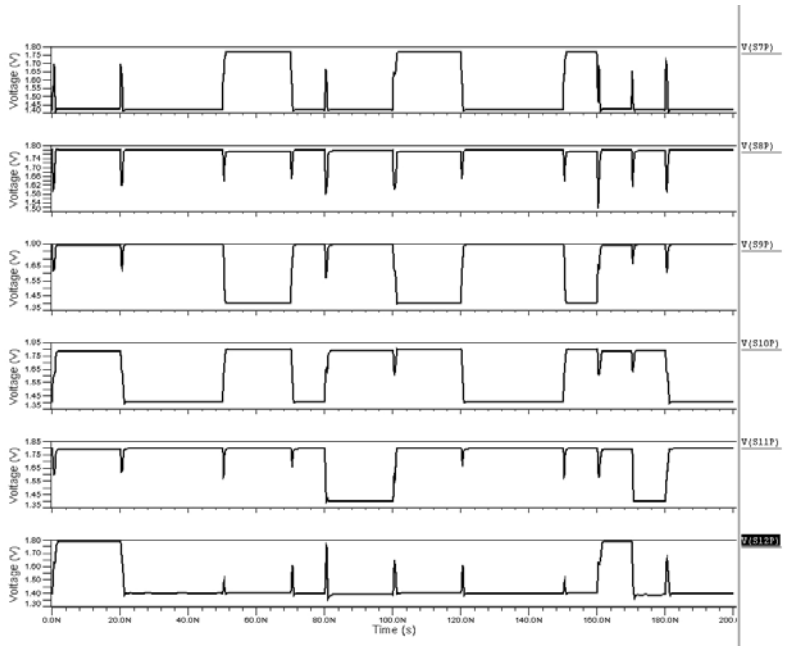

Figure 7.3. The second 6 bits of the result

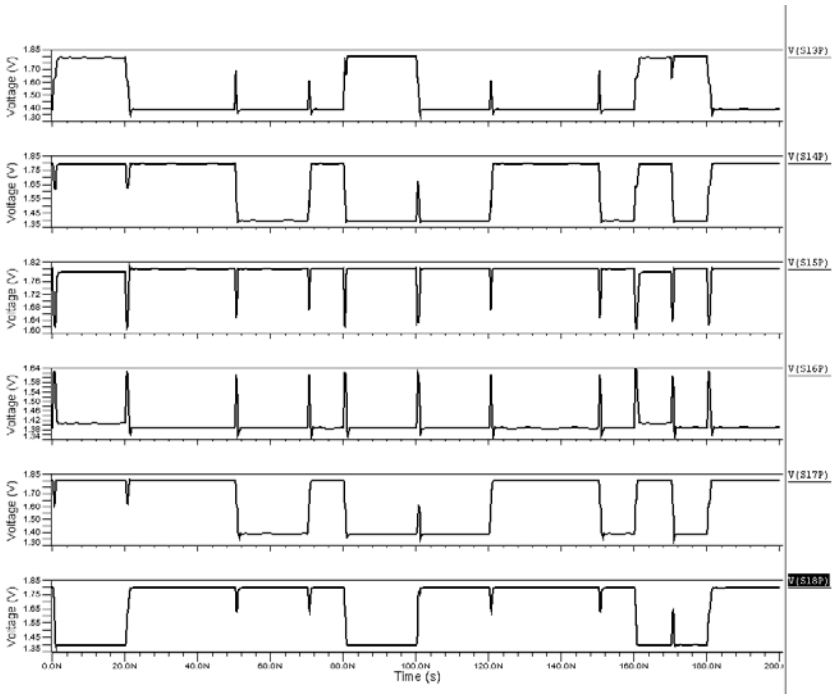

Figure 7.4. The third 6 bits of the result

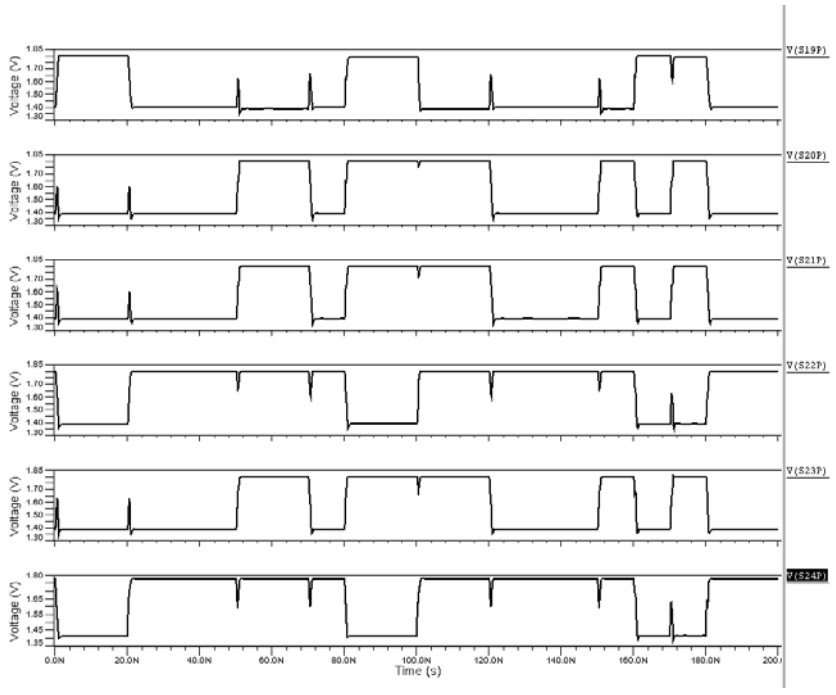

Figure 7.5. The fourth 6 bits of the result

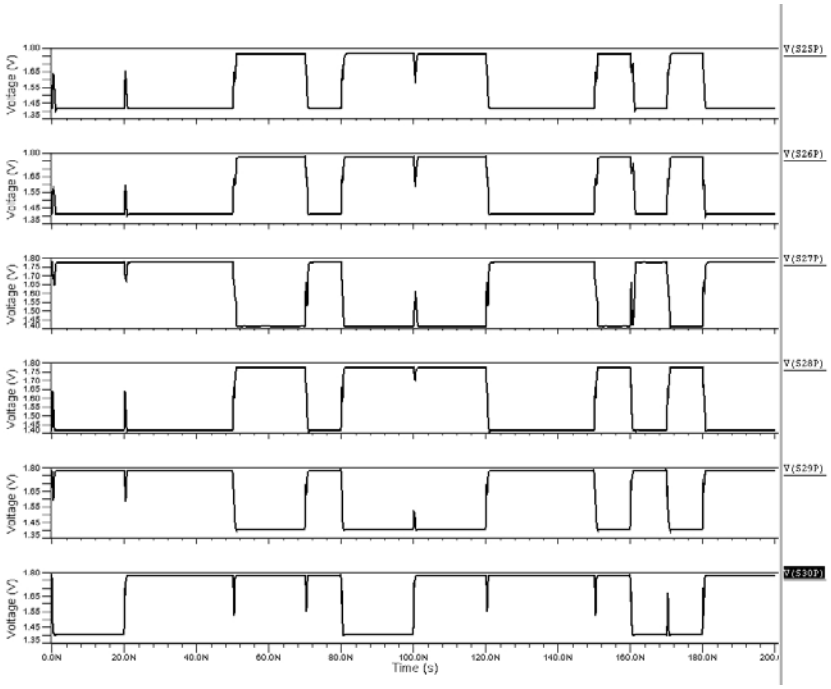

Figure 7.6. The last 6 bits of the result

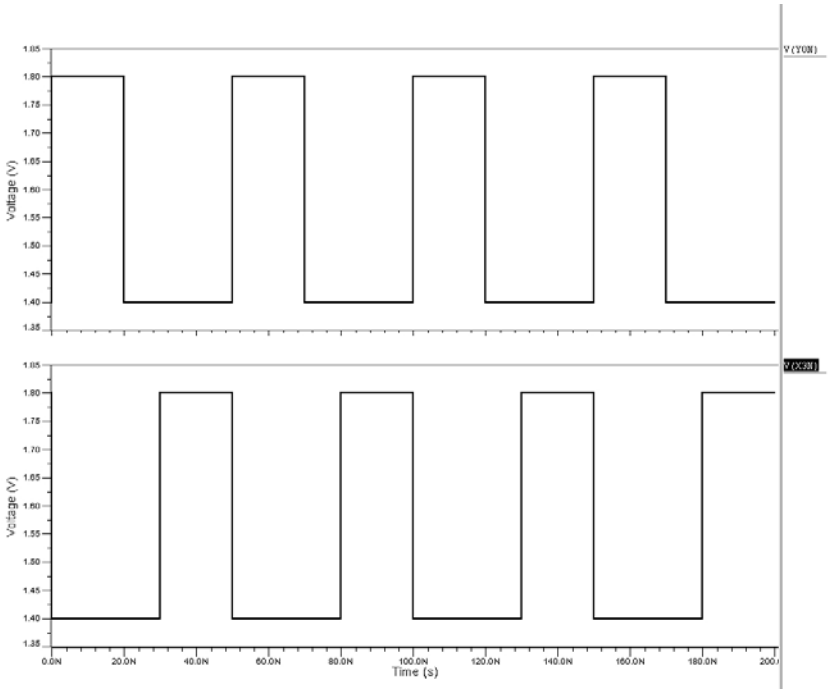

Figure 7.7. The input signals

Secondly, I set V_{RFN}=0.7V and V_{RFP}=0.4V in Figure 7.8. The difference between the maximum and minimum power supply current is only about 0.9 mA (21.5mA-20.6mA) during a switching event, which is only 4 per cent of the nominal power supply current (21.5mA). The power is (V_{dd}×I) 37.8mW.

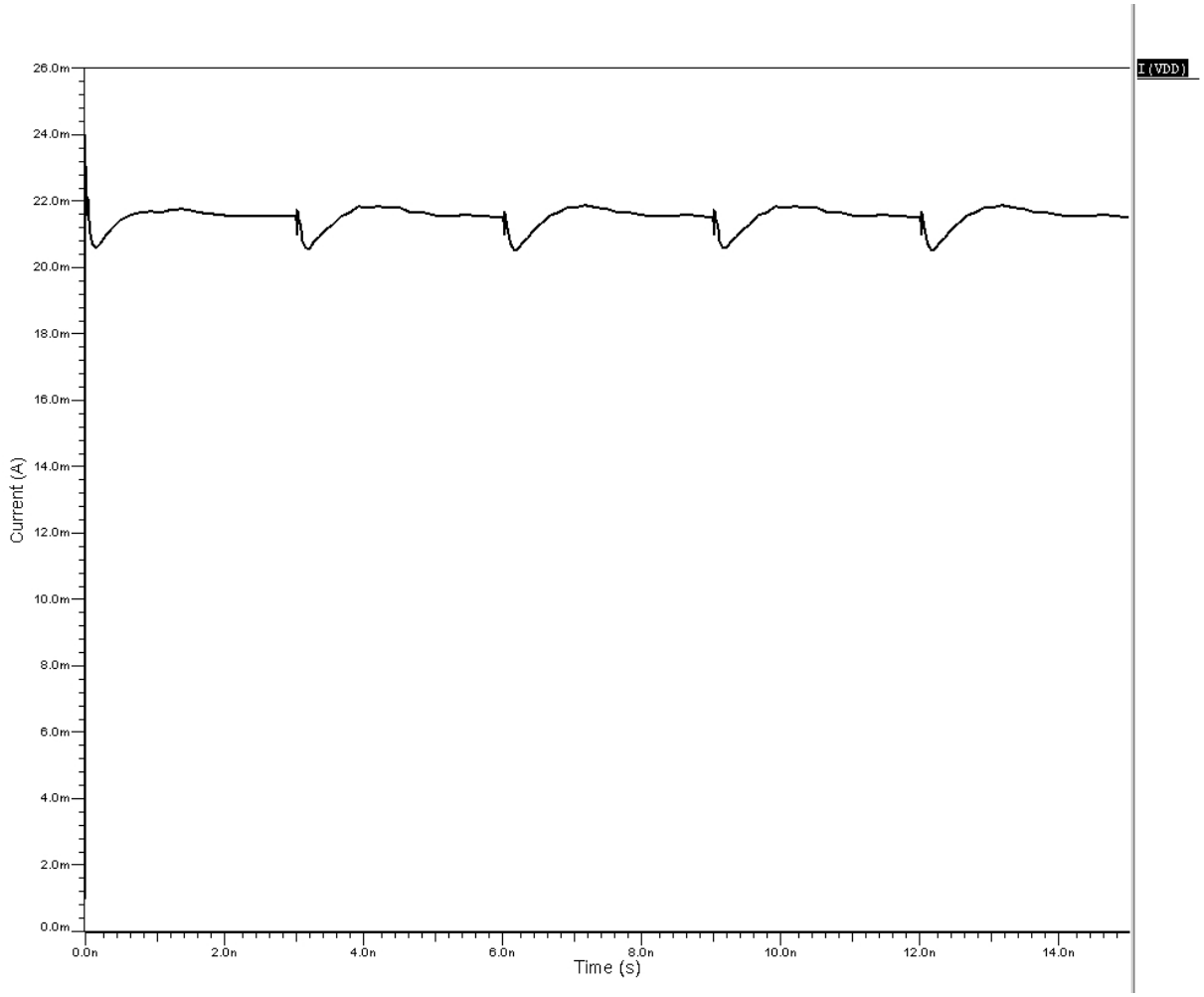

Figure 7.8. The power supply current of the 16 bit MCML multiplier for V_{RFN}=0.7V and V_{RFP}=0.4V

Thirdly, I set V_{RFN}=0.65V and V_{RFP}=0.64V in Figure 7.9. The difference between the maximum and minimum power supply current is only about 0.7 mA (15mA-14.3mA) during a switching event, which is only 4 per cent of the nominal power supply current (15mA). The power is (V_{dd}×I) 27.6mW.

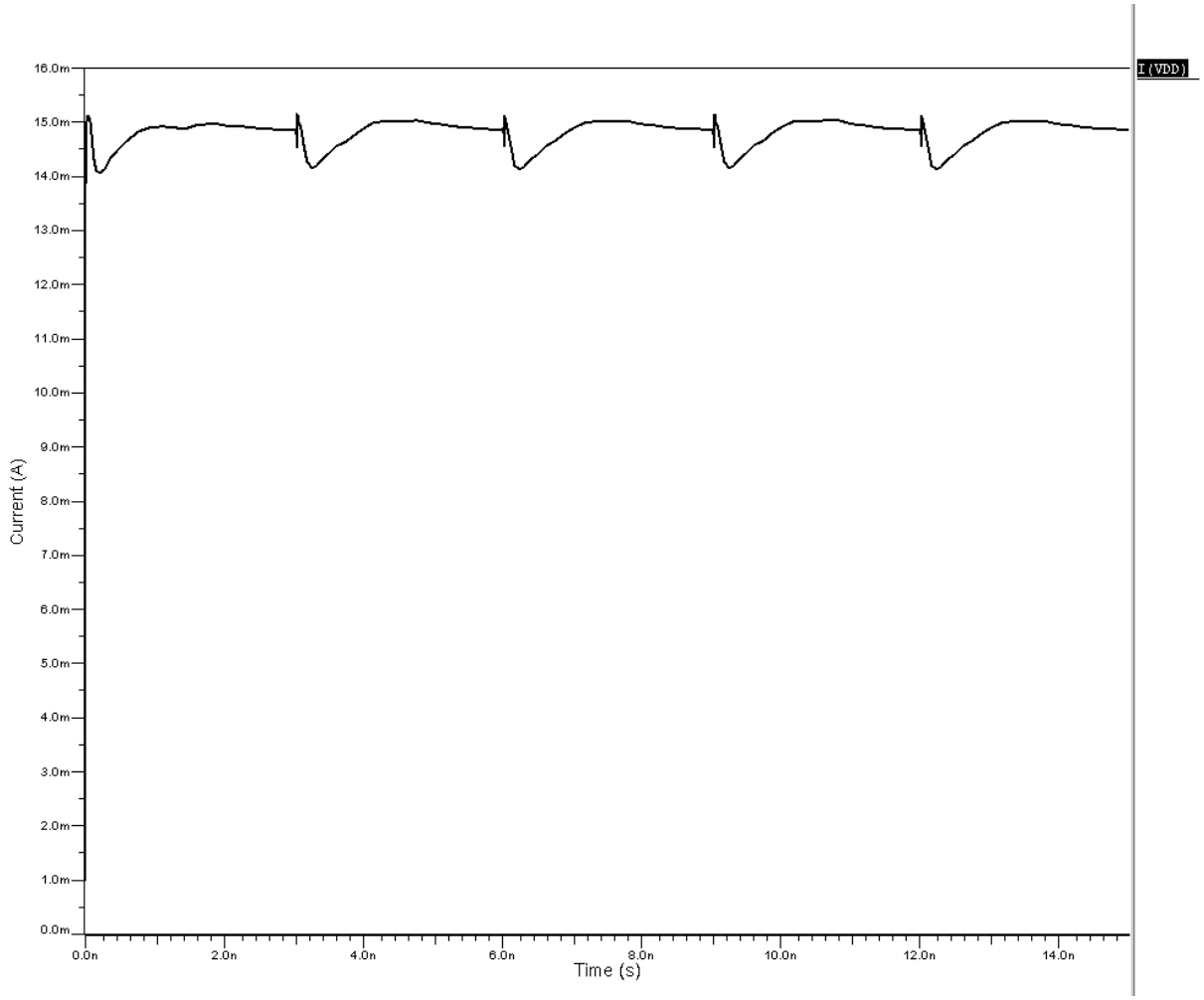

Figure 7.9. The power supply current of the 16 bit MCML multiplier for V_{RFN}=0.65V and V_{RFP}=0.64V.

Lastly, I set V_{RFN}=0.6V and V_{RFP}=0.82V in Figure 7.10. The difference between the maximum and minimum power supply current is only about 0.4 mA (9.4mA-9mA) during a switching event, which is only 4 per cent of the nominal power supply current (9.4mA). The power is (V_{dd}×I) 16.5mW.

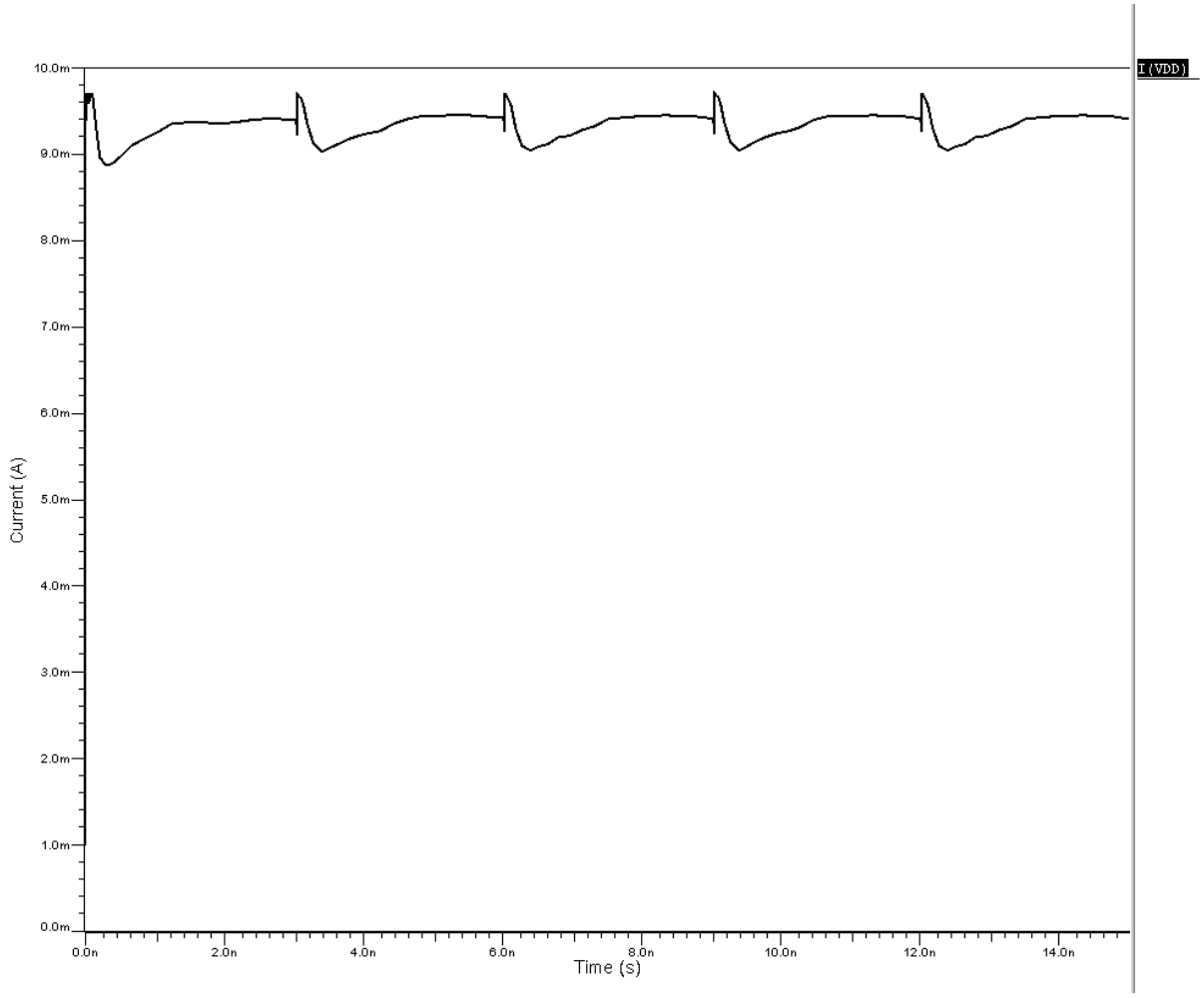

Figure 7.10. The power supply current of the 16 bit MCML multiplier for V_{RFN}=0.6V and V_{RFP}=0.82V

I multiplyed one positive number (011001010110010) with one negative number (1101100101001010) and the result is true (11110001101011101000100101110100).

I also multiplyed two negative numbers (1001001010010100), (11100110011001011) and the result is true (00101011110001100101011010111100).

I simulated this multiplier about 15 times in each mode for different input vectors. In each time, the output delay values change only per cent 5. Since the number of simulations are not large enough to obtain the real worst case value, I also add an extra per cent 10 delay value myself in Table 7.1. For example, the actual delay value for the 16 bit multiplier for 31mA is 1.1ns in simulation. But I take this delay value as 1.2ns to guarantee the worst case.

Table 7.1. The delays of multipliers for 4 different power supply currents

Multiplier	Delay
16 Bit Multiplier for 31mA	1.2 ns
16 Bit Multiplier for 21mA	1.6 ns
16 Bit Multiplier for 15mA	2.5 ns
16 Bit Multiplier for 9mA	4 ns

Also I simulated this multiplier with different input frequencies and found the maximum operating frequency in Table 7.2. Here the frequency is maximum operating frequency and the power is Current$\times V_{dd}$.

Table 7.2. The maximum operating frequency and the power dissipation of multipliers for 4 different reference voltages

Multiplier	Power	Frequency
16 Bit multiplier for V_{RFN}=0.76V and V_{RFP}=80mV	55mW	800 MHz
16 Bit multiplier for V_{RFN}=0.7V and V_{RFP}=0.4V	37mW	600 MHz
16 Bit multiplier for V_{RFN}=0.65V and V_{RFP}=0.64V	27mW	400 MHz
16 Bit multiplier for V_{RFN}=0.6V and V_{RFP}=0.82V	16mW	250 MHz

7.2. Comparisons

I compared my multiplier with the other multipliers in the literature in Table 7.3. There is no 16 bit MCML multiplier in the literature.

The first two multiplier in [12] and [13] are 8 bit MCML multipliers. Since they do 8 bit multiplication, my 16 bit multiplier results are better in respects of power and delay.

The third multiplier in [25] does 16 bit multiplication using pass transitor logic in 0.35µm technology. It can be seen that my multiplier results in Table 7.3. are better than this multiplier in respect of power and delay. Also the number of transistors in my 16 bit MCML multiplier is less than this multiplier (7268 to 12349).

The fourth multiplier in [14] does 16 bit multiplication using CMOS in 0.18µm technology. It can be seen that my multiplier results in Table 7.3. are better than this multiplier in respect of power and frequency. Also the number of transistors in my 16 bit MCML multiplier is less than this multiplier (7268 to 13444).

The other multipliers in [15] and [16] are 16 bit multiplier in different technologies (1.3µm and 90nm). It can be seen that my multiplier is not better, the results in Table 7.3 are comparable to these multipliers.

Also the power supply current spike of my 16 bit MCML multiplier is only 4 per cent of the nominal current and this value is not given in the other multipliers in Table 7.3. For example the difference between the maximum and minimum power supply current of my 16 bit MCML multiplier for V_{RFN}=0.76V and V_{RFP}=80mV is only about 1.2 mA during a switching event, which is only 4 per cent of the nominal power supply current. It is obvious that the power supply current spike of the other CMOS multipliers is worse than my 16 bit MCML multiplier.

Table 7.3. The comparison of the multipliers

Reference	Technology	Number of Bits	Delay	Frequency	Power	Number of Transistors
[12]	0.35µm MCML	8	4.53ns		4.86mW	
[13]	0.18µm MCML	8	3.8ns	4.76 GHz	261mW	
[25]	0.35µm CMOS	16	7.4ns	100 MHz	33.7mW	12349
[14]	0.18µm CMOS	16		400 MHz	35.8mW	13444
[15]	0.13µm ASD-CMOS	16	1.57ns		12.58mW	
[45]	90nm CMOS	16		1 GHz	22mW	
My multiplier for 4 different reference voltages	0.18µm MCML	16	1.2ns 1.6ns 2.5ns 4ns	800 MHz 600 MHz 400 MHz 250MHz	55mW 37mW 27mW 16mW	7268

8. CONCLUSIONS AND FUTURE WORK

8.1. Conclusions

A digital circuit style that seems to be promising in both reducing power consumption and delay, providing an analog friendly environment, is MOS Current Mode Logic (MCML). In this thesis, I firstly determined the design parameters of MCML. I created a cell library by using MCML. In this library, I optimized the MCML circuits by looking at the results of the optimization program in [44].

I created high performance arithmetic circuits by using this MCML library. I designed a 16 bit high performance signed multiplier.

In this multiplier, I used modified radix 4 booth encoding to reduce the number of partial products by half. I designed the booth encoder and booth selector circuits in MCML. By using MCML, only two gates are used in booth selector instead of four gates in CMOS. 136 booth selector circuits are used in my 16 bit multiplier.

I used Wallace-tree structure for speed and power improvement. In this tree, I used only full adders and half adders.

For the final 32 bit completion adder, I used a hybrid structure: ripple carry for bits (7:0), variable block carry-select for bits (23:8) and a conditional carry for bits (31:24). This hybrid architecture enables power reduction with no performance penalty in the final adder compared to a conventional high-performance adder.

In conclusion, I created a 16 bit non-pipelined signed multiplier using MCML and tested for 4 different supply currents. The power supply current spike is only 4 per cent of the nominal current and the multiplier consists of 7268 transistors.

Table 8.1. The performance of my 16 bit MCML multiplier

Multiplier	Power	Frequency
16 Bit Multiplier for V_{RFN}=0.76V and V_{RFP}=80mV	55mW	800 MHz
16 Bit Multiplier for V_{RFN}=0.7V and V_{RFP}=0.4V	37mW	600 MHz
16 Bit Multiplier for V_{RFN}=0.65V and V_{RFP}=0.64V	27mW	400 MHz
16 Bit Multiplier for V_{RFN}=0.6V and V_{RFP}=0.82V	16mW	250 MHz

I compared the simulation results with the other published multipliers in Table 7.3. and showed that my 16 bit signed multiplier using MCML is better or comparable with them.

8.2. Future Work

The layout should be drawn and the simulations should be repeated with layout extracted parameters. The chip may be manufactured if the results are satisfactory.

In the layout MCML to CMOS and CMOS to MCML conversion circuits can be designed for the inputs and outputs. Also the DC biasing circuits should be added in the layout to generate V_{RFN} and V_{RFP}.

REFERENCES

1. Mizuno, M., M. Yamashina, K. Furuta, H. Igura, H. Abiko, K. Okabe, A. Ono and H. Yamada, " A GHz MOS, Adaptive Pipeline Technique Using MOS Current-Mode Logic", *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 6, pp.784-791, June 1996.

2. Rabaey, J., *Digital Integrated Circuits: A Design Perspective*, Prentice Hall, 1996.

3. Musicer, J., *An Analysis of MOS Current Mode Logic for Low Power and High Performance Digital Logic*, Research Project, Berkeley University, 2000.

4. Chandrakasan, A. P. and R. W. Broderson, " Minimizing Power Consumption in Digital CMOS Circuits", *Proceedings of the IEEE*, Vol. 83, No. 4, pp498-523, April 1995.

5. Gray, P. R. and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits,* John Wiley and Sons, 1993.

6. Smith, A. S. and K. C. Smith, *Microelectronic Circuits*, Oxford University Press, 1997.

7. Alioto, M. and G. Palumbo, *Model and Design of Bipolar and Mos Current-Mode Logic : CML, ECL and SCL digital circuits*, Kluwer Academic Publishers, 2005.

8. Leblebici, Y. and E. Brauer, " Sub-70ps Full Adder in 0.18µm CMOS Current Mode Logic", *Proceedings of the IASTED International Conference*, December 2004.

9. Steven, K. Hsu, S. K. Mathew, M. A. Anders, B. R. Zeydel, V. G. Oklobdzija, R. K. Krishnamurthy and S. Y. Borkar, "A 110 GOPS/W 16-bit Multiplier and Reconfigurable PLA Loop in 90-nm CMOS", *IEEE Journal Of Solid-State Circuits*, 2006.

10. Neil, H.E W., D. Harris, *CMOS VLSI Design*, Pearson Education, 2005.

11. Cheng, K.H, W. Cheng , " 64 Bit High Performance Power - Aware Conditional Carry Adder Design", *IEICE Trans. Electron*, June 2005.

12. Kim, J. and Y. S. Lee, "Design of a low-power 8×8 - bit parallel multiplier using MOS current mode logic circuit", *International Journal of Electronics*, October 2007.

13. Srinivasan, V ., D. S. Ha and J. B. Sulistyo, " Gigahertz - Range MCML Multiplier Architectures", *ISCAS*, 2004.

14. Zhentao, L., C. Shuming, L. Zhaoliang and L. Conghua , " The Algorithm and Circuit Design of a 400MHz 16-Bit Hybrid Multiplier", *ACSAC* , 2006.

15. Morimoto, M., M. Nagata and K. Taki, " High - Speed Digital Circuit Design Using Differential Logic with Asymmetric Signal Transition", *IEICE Trans. Electron*, 2005.

16. Pereira, A. R., P. Alvodara and H. Wolfwang , " Design of a MCML Gate Library Applying Multiobjective Optimization", *ISVLSI*, 2007.

17. Shahnam, K. and S. Maitham, " Implementation of MCML Universal Logic Gate For 10GHz-Range in 0.13µm CMOS Technology", *ISCAS*, 2004.

18. Hassan, H and A. M. Elmasry , " MOS Current Mode Circuits : Analysis, Design and Variability", *IEEE transactions on VLSI systems*, 2005.

24. Toumazou, C . and C . Francesco , " Current Mode Digital Circuits " , *International Application Published Under the Patent Cooperation Treaty*, 2007.

25. Hirofumi S., K. I. Uda and B . Y. Lee, "A 16 - bit Redundant Binary Multiplier Using Low-Power Pass-Transistor Logic SPL" , *IEEE*, 2000.

26. Razavi, B., "Prospect of CMOS Technology for High - Speed Optical Communication Circuits", *IEEE Solid-State Circ.*, Vol. 37, No. 9, pp. 1135-1145, September 2002.

27. Razavi, B., *Design of Integrated Circuits for Optical Communications*, McGraw-Hill, 2003.

28. Tanabe, A., M. Umetani, I. Fujiwara, T. Ogura, K. Kataoka, M. Okiara, H. Sakuraba, T. Endoh and F. Masuoka, "0.18-μm CMOS 10 - Gb/s Multiplexer / Demultiplexer ICs Using Current Mode Logic with Tolerance to Threshold Voltage Fluctuation", *IEEE J. of Solid-State Circuits*, Vol. 36, No. 6, June 2001.

29. Haykin, S., *Communication Systems,* John Wiley and Sons, 1994.

30. Razavi B., "Monolithic Phase-Locked Loops and Clock Recovery Circuits (Theory and Design)", *IEEE Press*, 1996.

31. Hung , C., B. Floyd and B. Park , " Fully Integrated 5.35 - GHz CMOS VCOs and Prescalers", *IEEE Trans. Microwave Theory and Techniques*, Vol. 49, No. 1, Jan. 2001.

32. Lam, C. and B. Razavi, "A 2.6-GHz/5.2-GHz Frequency Synthesizer in 0.4 - μm CMOS Technology", *IEEE Jour. of Solid-State Circ.*, Vol. 35, No. 5, pp. 788-794, May 2000.

33. Nosaka, H., K. Isshii, T. Enoki and T. Shibata, " A 10-Gb/s Data-Pattern Independent Clock and Data Recovery with a Two-Mode Phase Comparator", *IEEE Jour. of Solid-State Circuits*, Vol. 38, No. 2, pp. 192-197, February 2003.

34. Yan, T. and H. Luong, " A 3-V 1.3-to-1.8-GHz CMOS Voltage – Controlled Oscillator with 0.3-ps Jitter", *IEEE Trans. on Circuits and Systems part II*, Vol. 45, No. 7, pp. 876-880, July 1998.

35. Rabaey, J., A. Chandrakasan and B. Nikolic, *Digital Integrated Circuits (A Design Perspective)*, Prentice Hall, 2003.

36. Musicer, J. and J. Rabaey, " MOS Current Mode Logic for Low Power, Low Noise CORDIC Computation in Mixed-Signal Environments", *ISLPED 2000*, 2000.

37. Fujimori I., " A 5-V Single Chip Delta - Digma Audio A/D Converter with 111 dB Dynamic Range", *IEEE J. Of Solid State Circuits*, Vol. 32, pp. 329-336, Mar. 1997.

38. Wallace, C. "A Suggestion for a Fast Multiplier", *IEEE Trans. Electron. Comput.*, Vol. EC-34, pp. 14–17, Feb. 1964.

39. Booth, A. D., "A Signed Binary Multiplication Technique *"*, *Appl. Math., Vol. IV*, 1951

40. McSorley, O.L., "High-Speed Arithmetic in Binary Computers ", *Proc. IRE*, Vol. 49, pp. 67-91, 1961.

41. M. Bhardwaj, R. Min, and A.P. Chandrakasan, "Quantifying and enhancing power-awareness of VLSI systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., vol.9, no.6, pp.757–772, Dec. 2001**.**

42. Bruma, S., "Impact of on-chip process variations performance on MCML [MOS current mode logic]", *IEEE International [Systems-on-Chip] SOC Conference*, 2003.

43. Brauer, E. J. and Y. Leblebici, "Low Noise MCML Prefix Adders Using 0.18 µm CMOS Technology", *Proceedings of the IASTED International Conference*, December 2004.

44. Hassan, H., M. Anis and M. Elmasry, " Design and optimization of MOS current mode Technology", *Elsevier Integration VLSI journal 38,* 2005.

45. Zeydel, B., V. G. Oklobdzija, S. Mathew, R. K. Krishnamurthy and S. Borkar, "A 90nm 1GHz 22mW 16x16-bit 2's Complement Multiplier for Wireless Baseband", *Symposium on VLSI Circuits Digest of Technical Papers*, 2003.