FPMA DESIGN IN SUBMICRON TECHNOLOGIES WITH DIGITAL ERROR CORRECTION

by

İsmail Kara

B.S., Electrical & Electronics Engineering,

Bogazici University , 2011

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

2013

ACKNOWLEDGEMENTS

First and foremost, I would like to express my appreciation to my thesis advisor, Assist. Prof. İ. Faik Başkaya for his guidance and support throughout my research and thesis. I would like to thank Prof. Günhan Dündar and Assoc. Prof. Alper Şen for sharing their knowledge and taking part in my thesis jury.

I owe many thanks to my dearest colleagues for their friendship, support and encouragement; Vahap Barış Esen, Bilgiday Yüce, Doğan Ulus, İpek Şen, İsmail Terkeşli, Gökhan Hacıahmetoğlu, Berkan Yaman, Selin Tolunay, Oğuz Karaduman, Uraz Çakacı, Ata Sarrafi, Abdullah Sarıduman, Simge Ay, Berk Çamlı and Can Doğa Kırbaç. I also would like to thank all the Beta Lab members for their friendship and helping me throughout my thesis. Besides, I especially would like to thank my flatmate Hüseyin Kaya for his friendship, supports and helping me to overcome the daily problems for the past eight years. I would like to thank my old but gold friends Berna Akbaş, Diren Demirci, Sercan Turan, Burhan Karahan, Uğur Koçak, Hakan Kaya, Murat Güneş, Haluk Mete Atalayın, Görkem Biçer, Mehmet Hüseyin Yılmaz, Merve Topselvi, Mehmet Çapın, Yavuz Ünal, Duygu Coşgun, Pınar Genç and Pınar Öztürk.

In addition, I would like to thank TÜBİTAK for supporting me through ARDEB since September 2012.

Finally, I would like to thank my family for their love and appreciation throughout my life.

ABSTRACT

FPMA DESIGN IN SUBMICRON TECHNOLOGIES WITH DIGITAL ERROR CORRECTION

The trend in VLSI systems towards System-On-Chip (SOC) leads to integration of digital and analog circuits on a single chip. The communication between the digital and analog blocks is employed by A/D and D/A converters. Also, digital error correction with the urge of submicron technologies is mostly used in these converters to achieve desired resolution, linearity, speed and low power. In this thesis, field programmable digital and analog array architectures and their applications are introduced. Digitally assisted architecture types for A/D and D/A converters are demonstrated and best suited architectures for TSMC 90nm technology are chosen. An 8-bit 1GSample/s current steering based DAC design is realized in TSMC 90nm technology. A self calibration technique is applied to the DAC as a digital error correction scheme. The performance improvement of the DAC after digital error correction is observed and the results are obtained using Mentor Graphics software tools and MATLAB.

ÖZET

MİKRONALTI TEKNOLOJİLERDE SAYISAL HATA DÜZELTİMİYLE FPMA TASARIMI

VLSI sistemlerdeki Yonga-üzerinde-Sistemlere doğru olan trend sayısal ve analog devrelerin tek bir yonda üzerinde birleşmelerine yol açmaktadır. Sayılsal ve analog bloklar arasındaki iletişim A/D ve D/A dönüştürücüler tarafından yapılmaktadır. Ayrıca, submikron teknolijilerinin kullanımı arttıkça bu dönüştürücülerde istenen çözünürlük, doğrusallık, hız ve düşük güç tüketimini sağlamak için sayısal hata düzeltim devreleri sıkça kullanılmaktadır. Bu tezde, alan programlanabilir sayısal ve analog dizi mimarileri ve bunların uygulama alanları tanıtılmıştır. A/D ve D/A dönüştürücüler için sayısal destekli mimari türleri gösterilerek bunlar arasından TSMC 90nm teknolojisi için en uygun olanları seçilmiştir. TSMC 90nm teknolojisi kullanılarak 8-bit 1GSample/s akım yönlendirmeli DAC tasarımı gerçekleştirilmiştir. Sayısal hata düzeltme mekanizması olarak DAC'a kendi kendini kalibrasyon etme tekniği uygulanmıştır. Sayısal hata düzeltiminden sonraki DAC performans iyileşimi Mentor Graphics yazılım araçları ve MATLAB ile elde edilerek gözlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A_β	Current mismatch parameter
A_{Vt}	Threshold voltage mismatch parameter
C	Capacitance
K	Number of inputs in a logic block
L	Channel length of a transistor
N	Number of bits in a data converter
V_{gs}	Gate to source voltage
V_t	Threshold voltage
W	Channel width of a transistor
σ	Variance

LIST OF ACRONYMS/ABBREVIATIONS

ADC	Analog-to-Digital Converter
CAB	Configurable Analog Block
DAC	Digital-to-Analog Converter
DNL	Differential Non Linearity
ENOB	Effective Number of Bits
FFT	Fast Fourier Transform
FPAA	Field Programmable Analog Array
FPGA	Field Programmable Gate Array
FPMA	Field Programmable Mixed Array
HSSL	High Speed Serial Link
IC	Integrated Circuit
INL	Integral Non Linearity
LUT	Look-up Table
LSB	Least Significant Bit
MOS	Metal Oxide Semiconductor
MSB	Most Significant Bit
NMOS	Negative-Channel Metal Oxide Semiconductor
OPAMP	Operational Amplifiers
OTA	Operational Tranconductance Amplifier
PLD	Programmable Logic Device
PLL	Phase-Locked-Loop
PMOS	Positive-Channel Metal Oxide Semiconductor
SC	Switched-Capacitor
SI	Switched Current
SOC	System-On-Chip
SNR	Signal-to-Noise Ratio

1. INTRODUCTION AND BACKGROUND

Electronic circuit design is an economic activity and greatly affected by the market factors. The evolution of the technology is quite fast which makes it very difficult to catch the market conditions. A good product, in other words a good electronic design, should be able to respond the consumer demands. These demands may come forward during or after the first design as new feature requests or performance improvements. So, one of the most effective solutions to overcome these difficulties has been reconfigurability. Field programmable gate arrays has been very successful considering the above problems. They provide fast and easy solutions to digital needs. There are many companies which manufacture and sell FPGAs with great numbers within the market. On the other hand, despite the domination of digital circuits in the market, analog circuits remain important in modern electronic systems, due to taking a role in interfacing between analog signals and digital electronic systems. The need and urge in the market lead to design and production of field programmable analog arrays (FPAA). But FPAAs couldn't reach the success of the FPGAs, since they have either been general purpose with more flexibility but poor performance, or less flexibility for a very narrow application range.

The evolution of the technology is quite beneficial for the digital circuits. Digital circuits will get far faster, occupy less area, consume less power with the scaling technology. However, collecting both analog and digital circuits in a single chip will be problematic for the analog circuits. The reasons behind are that non-ideal effects such as mismatch and non-linearity, reduced supply voltage hence reduced voltage room needed for some operations, etc. Apart from voltage requirements, the errors from mismatch and non-linearity can be well modeled and eliminated by either calibration of the analog circuit, or digital error correction. For FPAA designs, some digital circuitry may help the overcome some of the problems. This approach combines FPGA and FPAA in the same IC. Data converters (ADC & DAC) were designed in IC as interface between the FPAA and FPGA. This thesis focus on design of DAC and its digital error correction. In order to understand the functionality of these blocks, first

we need to analyze the digital and analog parts in detail, to see the requirements through the last two decades. For this purpose, rest of the chapter is focused on historical background of FPGA and FPAA. Then, in Chapter 2 and 3, digital and analog parts of the FPMA architecture are described in detail. In Chapter 4, interface circuit types are introduced. In Chapter 5, digitally assisted DAC block designs and simulation results are demonstrated. Finally, in Chapter 6, conclusions obtained from the simulation results are discussed.

1.1. FPGA

The architecture of a field-programmable gate array (FPGA) is shown in Figure 1.1 and consists of an array of logic blocks which are used to obtain different designs with programmable interconnections. An FPGA is similar to traditional programmable logic devices (PLDs) in terms of being programmed via electrically programmable switches. FPGAs have more complex architectures and logic implementations. These properties makes it superior to PLDs since PLDs have less level of integration compared to FPGAs. Although, PLD routing architectures are simple, they are inefficient crossbar-like structures that every output can be connected to every input through one switch. FPGA routing architectures are more efficient than PLDs and in this structure, each connection typically passes through several switches. In a PLD, logic is implemented using mostly two-level AND-OR logic with wide input AND gates. In an FPGA, multiple levels of lower fan-in gates, which is often much more compact than two-level implementations, is used to implement logic [4].

The simplicity of an FPGA logic is determined by the requirements of implemented circuit and can be as simple as a logic gate or as complex as a microprocessor. It has the ability of implementing many different sequential and combinational logic functions. Commercially available FPGAs contain one or more of the following logic blocks:

- Transistor pairs.
- Basic small gates such as two-input NAND's or exclusive-OR's.

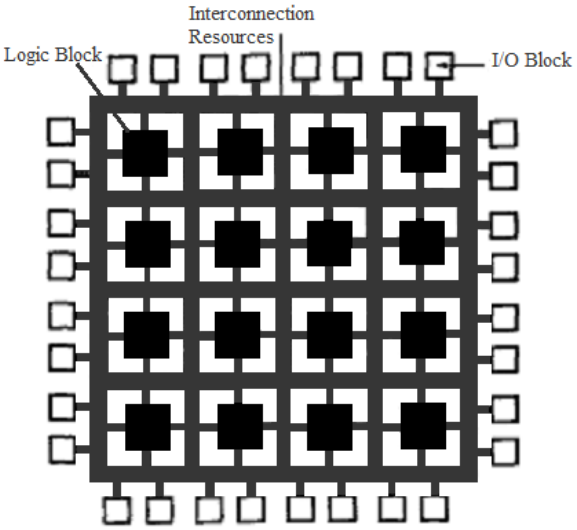


Figure 1.1. FPGA architecture [4].

- Multiplexers.
- Look-up tables (LUTs).
- Wide-fan-in AND-OR structures [4].

The routing architecture of an FPGA could be as simple as a nearest neighbor mesh [5] or as complex as the perfect shuffle used in multiprocessors [6]. More typically, an FPGA routing architecture combines different lengths of wire segments that can be interconnected via switches. Wire segment lengths are an important factor that affects the complexity of the FPGA routing architecture. If wire segments are insufficiently used, then logic blocks are under-utilized; conversely more than sufficient wire segment usage leads to wasted area due to remaining unused. A significant factor that affects both the density and the performance of the FPGA is the choice of wire segment lengths. For instance, choosing the long wire segments results in large area requirements and delays due to increasing cost of implementation of local interconnections. However, if the wire segments are chosen as short ones, for long interconnections, too many switches in series are needed, resulting in unacceptably large delays [4].

The programmable switches are implemented by using several different program-

ming technologies. There are three types of programmable switch technologies which are commonly used. These are:

- SRAM, where the switch is a pass transistor controlled by the state of a SRAM bit,
- Antifuse, which, when electrically programmed, forms a low resistance path, and
- EPROM, where the switch is a floating-gate transistor that can be turned off by injecting charge onto their floating-gate [4].

In all cases, a programmable switch occupies large area and exhibits high parasitic resistance and capacitance. Additional area is also required for programming circuitry.

The negative effects mentioned above can be reduced by careful architectural choices. The density and the performance can be optimized by careful choice of appropriate granularity and functionality of the logic block, and by achieving a good design of routing architecture, which minimizes the number of switches and has a high degree of routability. The programming technology highly restricts the architectural choices and implemented designs. Thus, no one architecture is likely to be best suited for all programming technologies and for all designs.

The complexity of FPGAs has passed the point where manual design is used. The modern FPGAs require automated logic and synthesis tools to completely utilize the FPGA architecture. An effective logic synthesis tool is needed to highly utilize a complex block. Also an effective placement and routing tool is required for a good utilization of an FPGA [4].

1.2. FPAA

Digital circuits are dominating the market for VLSI circuits. However, modern electronic circuits still require analog integrated circuits. One of the essential roles of the analog systems is interfacing digital electronics to the real world in applications

such as analog signal processing, motion control, biomedical measurements, etc. Also, analog circuits are still in race with digital circuits. They compete in dense, low-power, high-speed applications in low-precision signal-processing. The main advantage of digital integrated circuits over analog circuits is their ease of design [7].

In analog system design, there are a large number of different analog functions and different levels of signal complexities such as frequency, parasitics, signal levels and time. The time pressure over the analog integrated design cycles has pushed a new need for the development of high performance reconfigurable analog circuits. This motivation has resulted in research in the area of Field-Programmable Analog Arrays (FPAAs). The research focuses on finding low-cost, accurate, rapid prototyping techniques for analog and mixed analog-digital circuits- a long awaited development for circuit designers. Some results are recently obtained and already there are some commercial products. Also some university research labs are working on this subject, which is an indication of renewed interest over this goal [7].

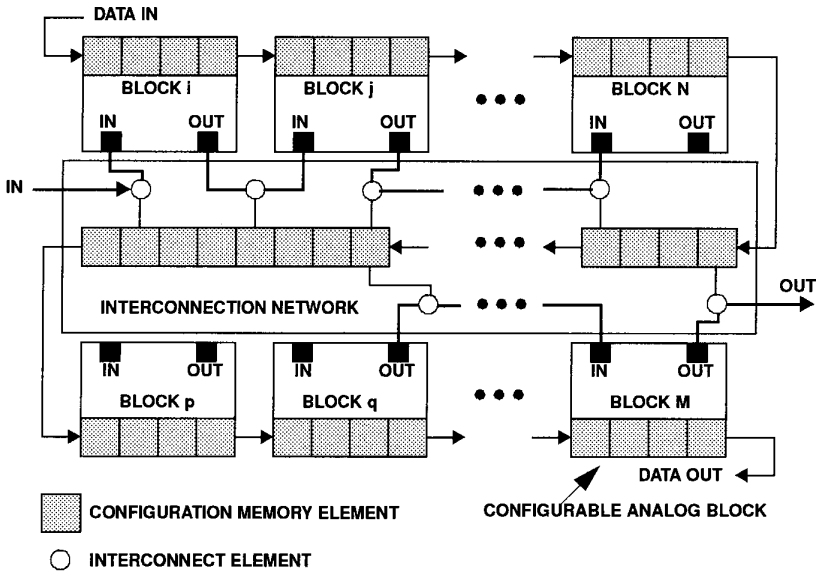


Figure 1.2. FPAA conceptual block diagram [7].

In its most general form, an FPAA is a collection of analog building blocks, a routing network which is user controllable and passes signals between the building blocks and some memory elements to define the structure and function. Alternatively,

the structure can be designed by other programming technologies such as antifuse programming. Figure 1.2 shows a conceptual block diagram of an FPAA which includes a set of Configurable Analog Blocks (CAB) and a routing network. Configuration memory is provided for the blocks and interconnect [7].

FPAA Design Issues:

- Discrete-time vs. Continuous-time. A significant choice in the implementation of an FPAA is whether to operate in discrete-time or continuous-time.
- Voltage-mode vs. Current-mode. Another important design choice is whether to use current or voltage as the signal parameter in the FPAA implementation.
- CAB Design. The design of the Configurable Analog Block (CAB) is usually dependent on some factors. These factors affect the functionality and performance features of circuits. Another factor due to used semiconductor technology and CAB design is area efficiency of routing resources. Routing resources and switches in the signal path requirement is more for fine grain FPAA architectures than a coarser grain FPAA architecture. On the other hand, the coarser architecture will be able to implement less number of circuit types than the fine architectures, since it will have larger blocks whereas finer grain FPAAs may be configured at the transistor level.
- Interconnect Architectures and Implementations. The routability of prototyped circuits and their performance are greatly dependent on the choice of an interconnection architecture and its implementation. Some problems such as fan-out, noise and the presence of switches in the signal path are more problematic on analog circuits compared to digital circuits. By considering these problems, either hierarchical or full crossbar interconnection architectures are used.
- Programmable Components. Some different methods are proposed to implement programmable resistors. Some of these are gate voltage controlled pass transistors, complex structures such as MOS transconductors. Also, programmable capacitor arrays are very common and especially used in switched-capacitor (SC) circuits [7].

1.3. FPMA

The trend in VLSI systems towards System-On-Chip (SOC) leads to integration of digital and analog circuits on a single chip. This trend is advantageous, since reduction in the number of chips will lead to less interconnections, area, and most importantly, cost. The programmable devices also follow the same trend. This is because of rapidly expanding market for mixed-analog-digital devices. Also, demand for mixed systems are growing faster than either analog or digital devices. Although commercial FPGAs are in the market far earlier than FPAAs, there has also been research into the area of Field- Programmable Analog Arrays (FPAAs) [1,8]. Also, there have been some attempts to use both digital and analog circuits on a single programmable chip to produce a single Field-Programmable Mixed-analog-digital Array (FPMA) [9].

Although large systems today are mostly digital, they still interface with a largely analog world. For a SOC, the interface systems must be included on the chip. The mixed systems are used in a large variation of applications such as instrumentation, control, telecommunications, etc. Two conceptual views of FPMA architecture are shown in Figure 1.3. In the first view, shown in Figure 1.3a, there are analog and digital arrays with some interconnections. Signal conversion, if required, is made from these resources available in the two arrays. Alternative is, shown in Figure 1.3b, consisting of data converters for signal conversions. The former view is better in terms of flexibility but costs in large area and reduction in the speed.

The architecture of an FPMA is determined by the requirements of the needs. For an FPMA design, a number of mixed-signal circuits should be collected and evaluated for the resources that are needed in a mixed-signal system. Some of these circuits are data converters: A/D converter, D/A converter. These data converters can be built from the available field programmable resources in the analog and digital arrays. Some other circuits are a noise generator, a phase-locked loop (PLL), an automatic gain controller (AGC), a simple data acquisition system, and a signal processor. To implement these data converters, some circuits are essential to include [9].

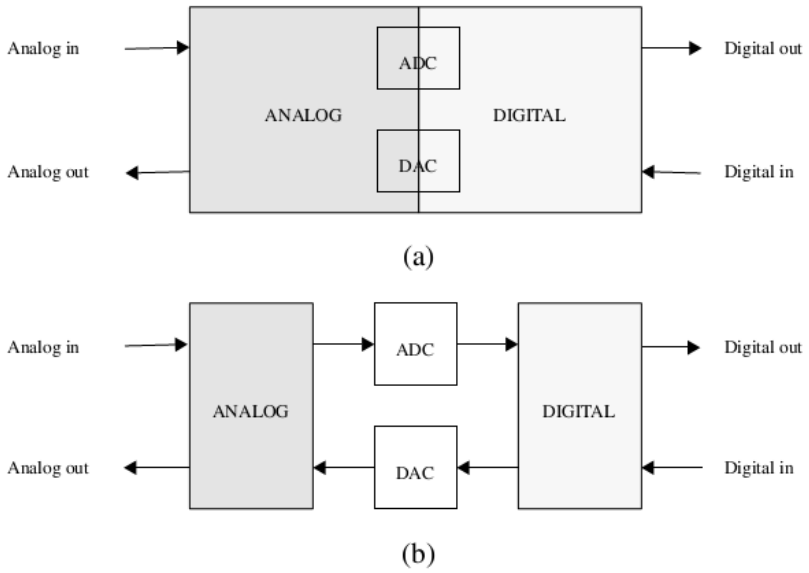


Figure 1.3. Conceptual views of mixed-signal architectures: (a) array architecture with field-programmable resources capable of constructing data converter as required; (b) array architecture with dedicated data converters [9].

1.4. Analog Circuits in Deep Submicron Technologies

The decreasing price-per-performance of digital circuits motivates the development of CMOS technology. The development speed is determined by Moore's Law [10]. To make digital circuits denser and to achieve low power consumption, the dimension shrinks with the lower nominal supply voltages. Although these are quite beneficial for digital circuits, the same is not true for analog circuits [11–13].

Novel integrated circuits (ICs) are type of mixed signal systems and consist of large digital core including a CPU or DSP and memory, generally surrounded by analog interface blocks such as D/A, A/D converters, I/O, RF front ends, etc. When both analog and digital circuits are on the same chip and development of CMOS technology is required for the performance of digital circuits, analog circuits also should be designed in that technology. In ultra-deep submicron CMOS technologies, several issues for analog circuits are observed and discussed for better performances in literature [13].

The development in manufacturing processes leads to reconsideration of analog circuit behaviors. A major problem is the decreasing supply voltage. The supply voltage has fallen down to 1.2V from 5V, and it is still possible to design analog circuits. But, more decrease in supply voltage is expected to result in serious performance problems for analog circuits due to decreasing signal headroom that prevents signal integrity for reasonable power consumption levels. Comparable bias conditions show that transistor properties do not get worse actually, but lower supply voltages lead to lower operating voltages and eventually worse transistor properties and lower circuit performance [14].

Another problem is gate leakage. Gate leakage will reach very high levels in novel technologies. When the gate oxide thickness is reduced as small as one atomic layer, the gate current increases one order of magnitude. Despite the technological developments, gate leakage will be a part of analog design. In literature, a bias insensitive frequency, f_{gate}, is introduced to estimate the effect of gate leakage. Gate leakage mismatch will dominate the long transistor mismatch. This effect limits the achievable maximum matching performance. This problem can be solved by either using more power or using active cancellation techniques [14].

1.5. Digital Error Correction

Analog and digital circuits have some properties that are scaled with migration to smaller technologies. Some of these properties are area and power consumption. However, the rate of change is slower for analog circuits compared to digital circuits. At the same time, as previously stated, as the technology continues to scale, the lower supply voltages and relatively higher process variations result in some difficulties for designing high performance analog circuits. Additionally, in ultra deep submicron technologies, the linearity of analog circuits is significantly affected by fast decaying maximum voltage rating. Process variations and design problems due to low supply voltages can be overwhelmed by using digitally assisted analog designs. These designs require on chip digital control and calibration circuits. Thus, either the dynamic operation of the analog system is adopted to environment or the errors are compensated

automatically with the help of digitally assisted circuits. These type of assisted circuits also have the advantage of the flexibility and power efficiency which help them to become more popular. Some good examples of this kind of circuit designs are all-digital PLLs, pipeline ADCs, RF image-reject receivers and adaptive equalizers in HSSL receivers. Digital signals have another advantage over analog ones: It is much easier to control and observe the digital signals. Using this property, analog signals can be easily controlled and significantly improved by digital adaptation unit [15].

2. DIGITAL PART OF FPMA ARCHITECTURE

2.1. Logic Blocks

In digital systems, a logic block is used for storage elements and basic computations. The first programmable gates were implemented using transistors as the most simple way. Storage elements and basic gates were built from these transistors. This type of fine grained architectures require large amount of programmable interconnect. This will lead FPGA to suffer from area-inefficiency, low performance, and high power consumption. On the other hand, in some commercial products a logic block is a complete processor with some fine grained small logic blocks. However in this case performance won't be obtained from customizable hardware and implementing a 2-input AND gate will be extremely inefficient. Between these two cases, logic block choices vary from fine to coarse grain architectures. These logic blocks usually contain transistors, NAND gates, multiplexers, look-up tables, and PAL-style wide-input blocks. Additionally, modern FPGAs include a set of different type of blocks such as dedicated memory blocks or multipliers. These kind of structures are highly efficient at implementing specific functions [16].

As previously stated, FPGA logic blocks are different from each other in terms of size and implementation capability. For example, a logic block, which consists of tho transistors and can be used to implement only an inverter, is used in the Crosspoint FPGA. Whereas, Xilinx 3000 series FPGA is much larger and can be used to realize any five-input logic function. In the sight of these, logic blocks can be classified by their granularity. Commercial logic blocks are classified as fine-grain and coarse-grain [4].

2.1.1. Fine-Grain Logic Blocks

The most fine grain logic block would consist of few transistors that can be programmably interconnected. A single transistor pair in the logic block is used in the Crosspoint FPGA , as illustrated in Figure 2.1. This FPGA also has another type of

logic block, that is used for the implementation of RAM.

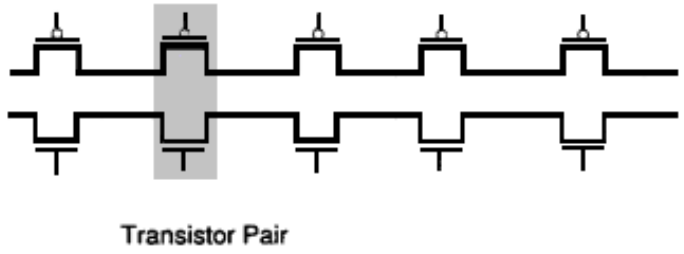


Figure 2.1. Transistor pair tiles in Crosspoint FPGA [4].

Another example of fine-grain FPGA architecture is Plessey FPGA. Its basic logic block consists of 2-input NAND gate as illustrated in Figure 2.2. The desired function is obtained by connecting the NAND gates [4].

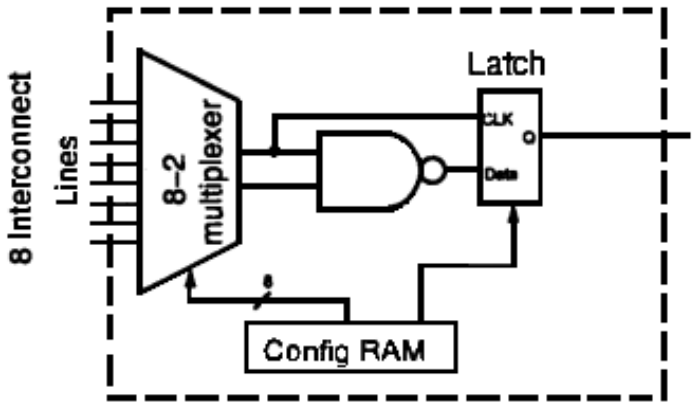


Figure 2.2. The Plessey logic block [4].

The most significant property of the fine-grain architectures is that almost all of the logic blocks can be utilized. This stems from the ease of efficiently using the small logic blocks. The main disadvantage of fine-grain logic blocks is the requirement of large wire segments and the number of programmable switches. This resource requirement for routing leads to large area and delay, which makes fine-grain logic block slower and less dense compared to coarse-grain blocks as a result.

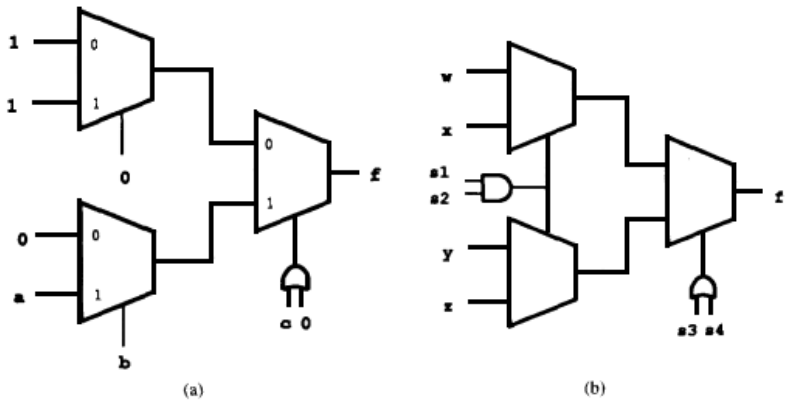


Figure 2.3. The Actel logic blocks (a)Act-1 (b)Act-2 [17, 18].

2.1.2. Coarse-Grain Logic Blocks

The Actel logic block utilizes multiplexers to implement different logic blocks. This is achieved by connecting each of its inputs to a constant or to a signal. The Actel ACT-1 logic block [17] is illustrated in Figure 2.3a and consists of three multiplexers and one logic gate, has a total of 8 inputs and one output. Different input sequences lead to realization of 702 logic functions. The Act-2 logic block [18] is similar to Act-1, except that the separate multiplexers on the first row are joined and connected to a 2-input AND gate, as shown in Figure 2.3b. 766 different functions can be implemented using the Act-2 logic block.

The Quicklogic FPGAs [19], which is illustrated in Figure 2.4, has four to one multiplexers whose inputs are fed by AND gates in the logic blocks.

The advantage of Multiplexer based logic blocks is having a large degree of functionality using small number of transistors. However, this case requires large number of inputs which results in excessive amount of routing resource usage. Thus, this kind of designs are best suited to FPGAs with antifuses like small size programmable switches.

Xilinx logic blocks which are called look-up tables (LUT) are commonly used in

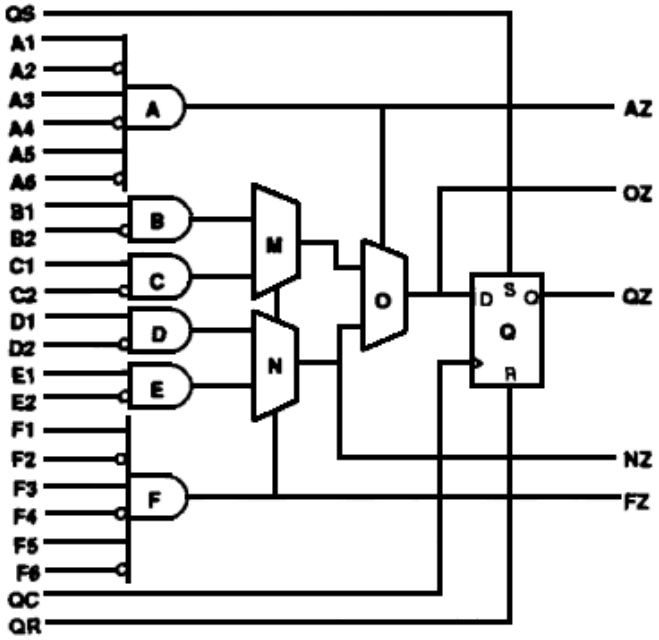


Figure 2.4. The Quicklogic logic block.

modern structures. LUTs are SRAMs which can provide any logic function. The most advantageous feature of a k-input LUT is that it can provide any function of k inputs and there are 2^{2^k} such functions. The disadvantage of k-input LUT comes out when k is more than five since it will take too much area to implement and most probably it will be underutilized due to lack of sufficient logic synthesis tool. The Xilinx 3000 series logic block [20] has one 5-input 1-output LUT as illustrated in Figure 2.5a. It can be configured in such a way that there are two 4-input LUTs for up to five distinct inputs. This makes LUT flexible and better utilized since generally five inputs is enough for many logic functions. The Xilinx 4000 series logic block [21] consists of two 4-input LUTs connected to a 3-input LUT as illustrated in Figure 2.5b. All of the inputs are distinct and externally changeable. Using two different sized LUTs provides a better trade-off between performance and logic density. Another property of this logic block is direct connection of two 4-input LUTs' outputs to the 3-input LUT. With a proper use, this may ensure a great performance increase, but possibility of underutilization of LUTs may occur and is not desired at all. Also each LUT can be used as an SRAM block for small amount of memory needs [4].

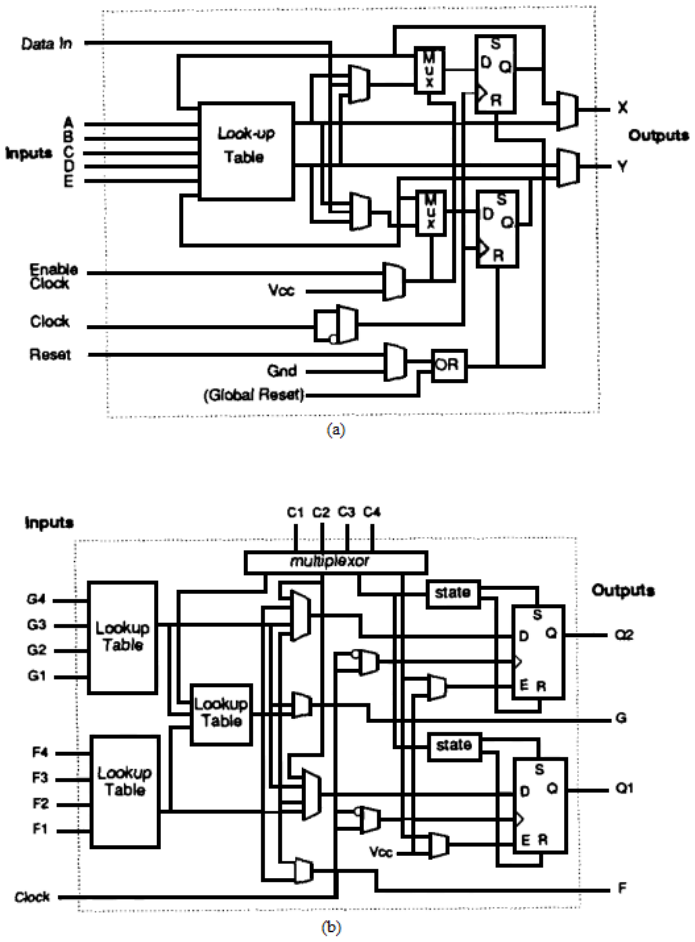


Figure 2.5. (a)The Xilinx 3000 logic block. (b)The Xilinx 4000 logic block.

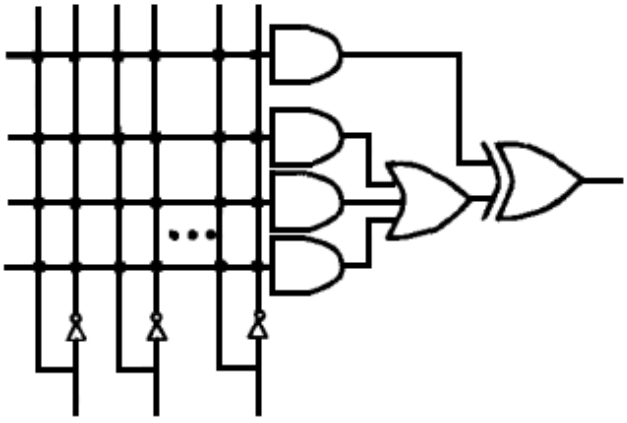


Figure 2.6. The Altera 5000 series logic block.

Altera is another well-known FPGA manufacturer and its FPGAs use another architecture different than LUTs but wired AND-OR gates which allow large number of inputs (20 to over 100 inputs of AND gates feeding into an OR gate with three to eight inputs). Altera MAX 5000 series logic block, which is illustrated in Figure 2.6, has the advantage of less requirement to the programmable interconnect due to using wide AND gate to obtain desired logic function with few levels of logic block. But, it is difficult to make efficient use of all of the inputs resulting in a lower density. Another disadvantage is static power consumption of wired-AND configuration due to pull-up device usage. This drawback has been overcome with the MAX 7000 series at the expense of an increase in delay. MAX 7000 series also has two more product terms and more flexibility compared to MAX 5000 series.

2.1.3. The Effect of Logic Block Functionality on Area Efficiency

The logic block functionality is an important factor and can be defined as the number of logic blocks required to implement a set of circuits in an FPGA architecture. The functionality of the logic block determines the utilization and the amount of the required area. If it has sufficient functionality, then it will probably be underutilized and some part of the active area will be wasted. If it has insufficient functionality, then a large amount of routing resources will be used. Also DFF usage in the logic blocks affects the overall area [22].

A simple thought is that if the functionality of the logic block increases, then implementing a desired circuit will require less number of logic blocks. However, increase in the functionality leads to more pin usage that means a significant amount of area increase. Thus, a high area-efficient block is determined by looking at the functionality per connected pin. In [22], it's shown that an increase in the logic block functionality leads to a decrease in the number of logic blocks, which leads to a larger number of pin connections per block that results in an increase in routing area. Since the routing area dominates the active area, and regardless of the used programming technology, the best choice for number of the inputs of the LUT is determined by looking at the functionality per connected pin, which is between three and four, and

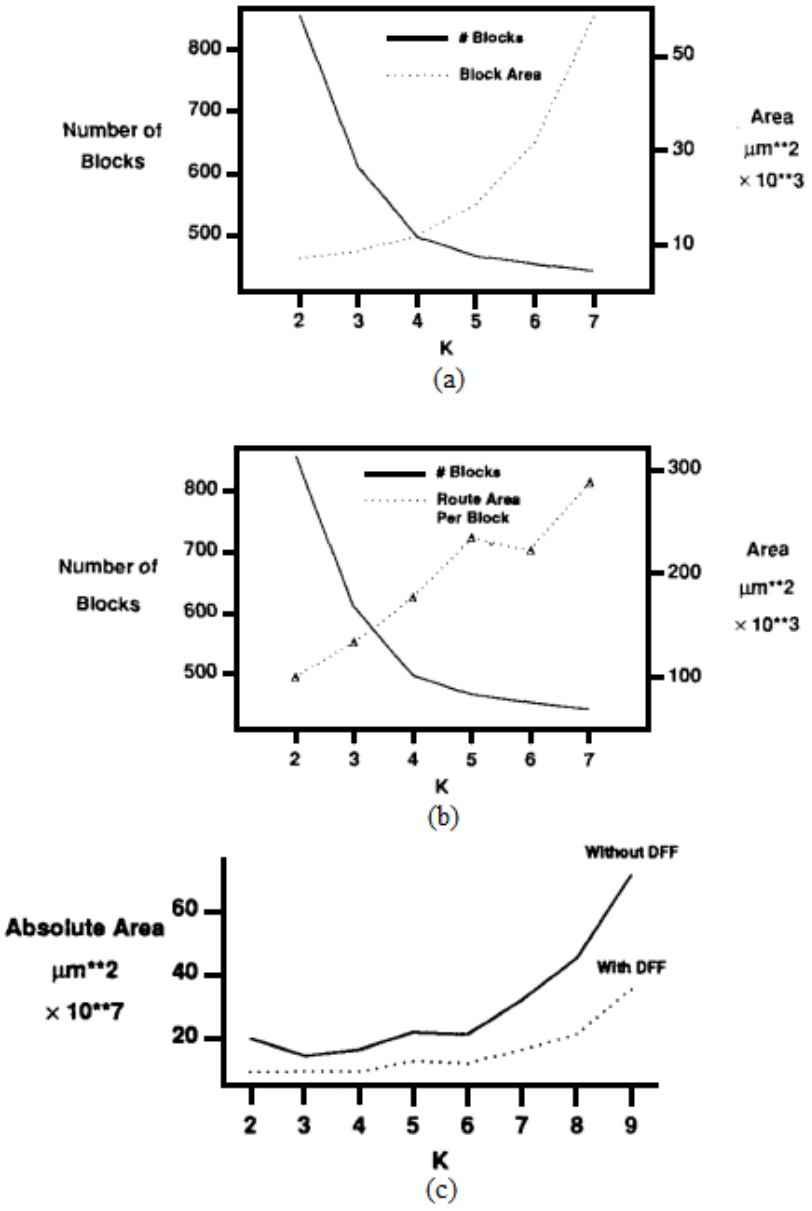


Figure 2.7. For K input (a)Number of blocks and block area versus K (b)Number of blocks and route area per block versus K (c)Area with and without of DFF.

that is always beneficial to include a D flip-flop in the logic block as shown in Figure 2.7c.

2.2. Switch Blocks

An FPGA is programmed using electrically programmable switches. There are a number of different programming technologies with different properties such as size, on-resistance, capacitance, and complexity of manufacturing. These properties are the keys for choosing the FPGA architecture. Widely used programming technologies are SRAM, antifuse and flash(EPROM, EEPROM, and floating-gate) in modern FPGAs [4].

2.2.1. SRAM Programming Technology

Static memory cells are used to control pass gates or multiplexers as shown in Figure 2.8. Some of the leading manufacturers such as Xilinx, Plessey, and Lattice use this technology. Two main functionalities of SRAMs are (i)controlling the signal flow and (ii)store the data in the LUTs. This technology has two main advantages that make it the main choice for the manufacturers; fast re-programmability and requirement of no special integrated circuit processing steps unlike other technologies. But it also has some drawbacks such as large area due to need of six transistors, volatility of the SRAM cell, high on-resistance because of using pass transistors, and high switching capacitance.

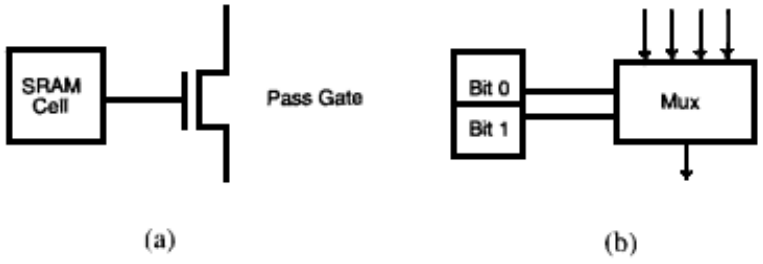


Figure 2.8. SRAM controlling (a)pass gate (b)multiplexer.

2.2.2. Antifuse Programming Technology

An antifuse is a two terminal device whose terminals have very high resistance before programming. To programme the antifuse, a high voltage is applied to its terminals to make a link between terminals. The programmed link has a very low on-resistance and is permanent. The most important advantage of antifuse programming technology is its low area. Beside low on-resistance and low area, another advantage of antifuse is low parasitic capacitance compared to the other technologies. Disadvantages of this technology are one time programming, and need for extra circuitry to obtain high voltages.

2.2.3. Flash (EPROM, EEPROM, and floating-gate) Programming Technology

Another alternative to SRAMs is the use of floating-gate programming technology that is like flash or EEPROM memory cells. These cells are non-volatile meaning that information is always kept as a charge on the transistor without power requirement. This approach is more area efficient than SRAM technology and it can be programmed unlimited times. However, they cannot be reprogrammed infinitely. It requires some other processes to manufacture and has high on-resistance and high static power consumption due to the pull up resistor as shown in Figure 2.9.

Comparison of programming technologies is given in Table 2.1 [16] .

2.3. Routing Architectures

The programmable routing in an FPGA provides the connection between logic blocks and I/O blocks. It consists of programmable switches and wiring segments that form the desired connections. Figure 2.10 shows a general routing architecture model. A *wire segment* is a wire between any two switches. A *track* is a number of wire segments in a line. A *routing channel* is a group of tracks [16].

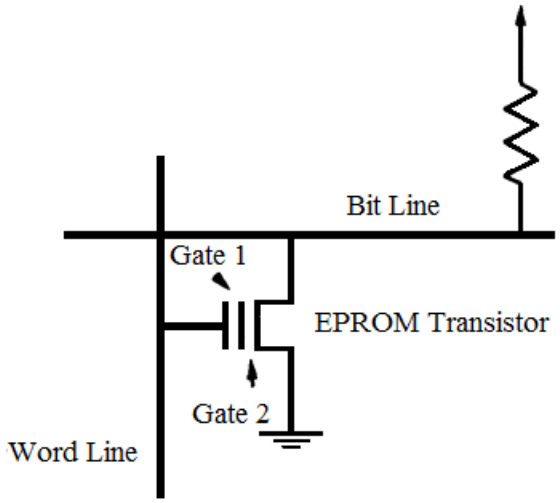


Figure 2.9. Floating-gate programming technology.

Table 2.1. Comparison of programming technologies.

	SRAM	Flash	Anti-fuse
Volatile?	Yes	No	No
Reprogrammable?	Yes	Yes	No
Area (storage element size)	High (6 transistors)	Moderate (1 transistor)	Low (0 transistor)
Manufacturing process?	Standard CMOS	Flash Process	Needs special development
In-system programmable?	Yes	Yes	Yes
Switch resistance	∼500-1000 Ω	∼500-1000 Ω	20-100 Ω
Switch capacitance	∼1-2 fF	∼1-2 fF	<1 fF
Programming yield	100%	100%	>90%

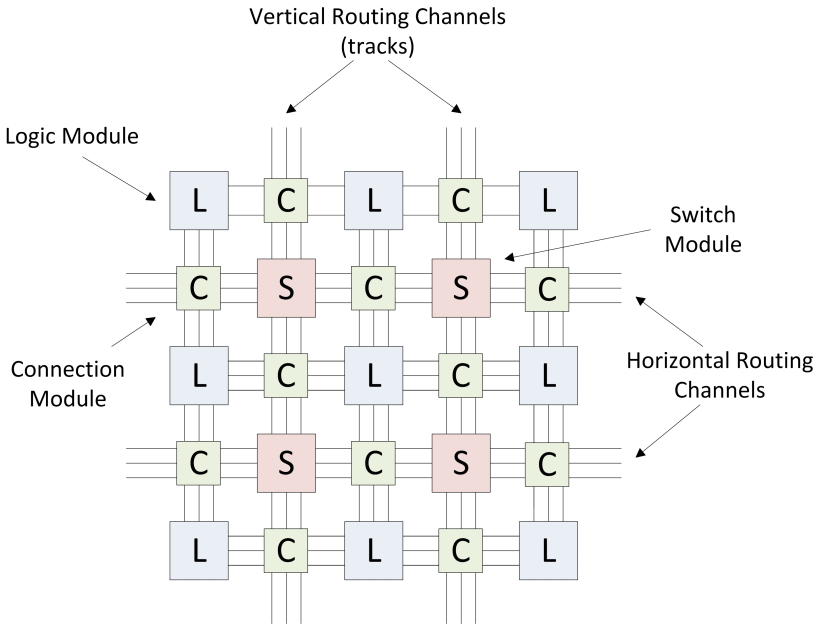


Figure 2.10. General FPGA routing architecture.

Connection block shown in Figure 2.10 appears in all FPGA architectures. It provides the connection of any logic block to the other logic blocks or I/O blocks via wire segments and tracks. Another structure is switch block which provides the connectivity between the horizontal and vertical wire segments. Some architectures have only one block replacing both connection and switch blocks [4, 16].

Routing architectures are chosen by considering the requirements of overall design of the FPGA. Following subsections will illustrate some examples of commercial routing architectures.

2.3.1. The Xilinx Routing Architecture

The Xilinx 3000 series routing architecture is shown in Figure 2.11. Logic blocks are connected to the channels by connection blocks. Area restrictions force each pin to connect to only two or three of five tracks. The connections are implemented with pass transistors for the output pins and multiplexers for the input pins. This reduces the number of SRAM cells needed per pin. Switch blocks connect wire segments

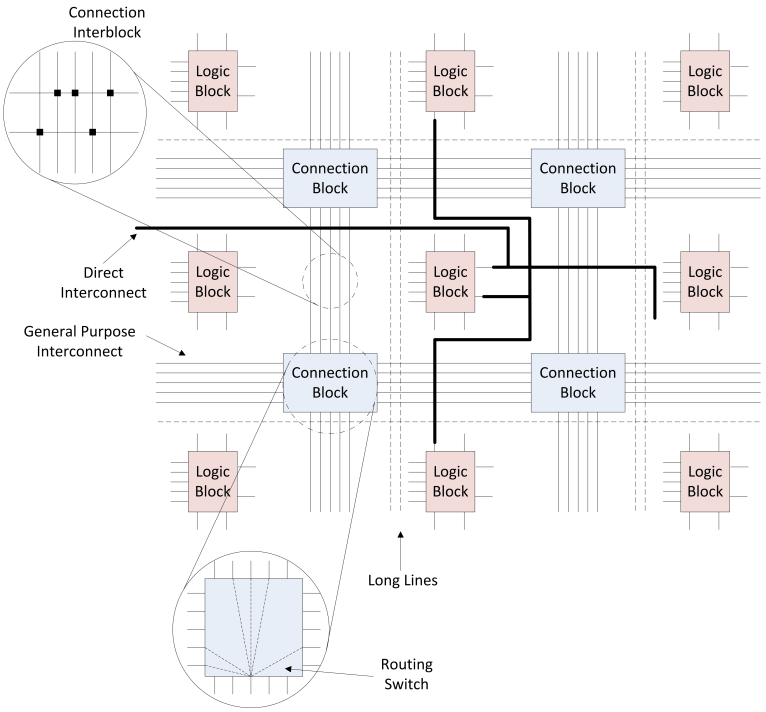


Figure 2.11. Xilinx 3000 routing architecture.

of horizontal and vertical channels. Different types of wire segments are used due to area and performance restrictions. *General purpose interconnects* are the wire segments which pass through switch blocks. *Direct interconnects* are used for direct communication of neighboring logic blocks. *Long lines* span whole chip and provide high fanout uniform delay connections. *A clock line* spans whole chip and is connected to the clock inputs of flip-flops [4, 16].

The Xilinx 4000 series architecture is similar to the 3000 series. In the 4000 series, there are many more general purpose tracks. All pins are connected to all tracks. Four of the tracks pass through a switch only every second switch. These tracks are shown in Figure 2.12. These long wire segments have the advantage of less series resistance when traveling the same distance compared to the shorter wire segments.

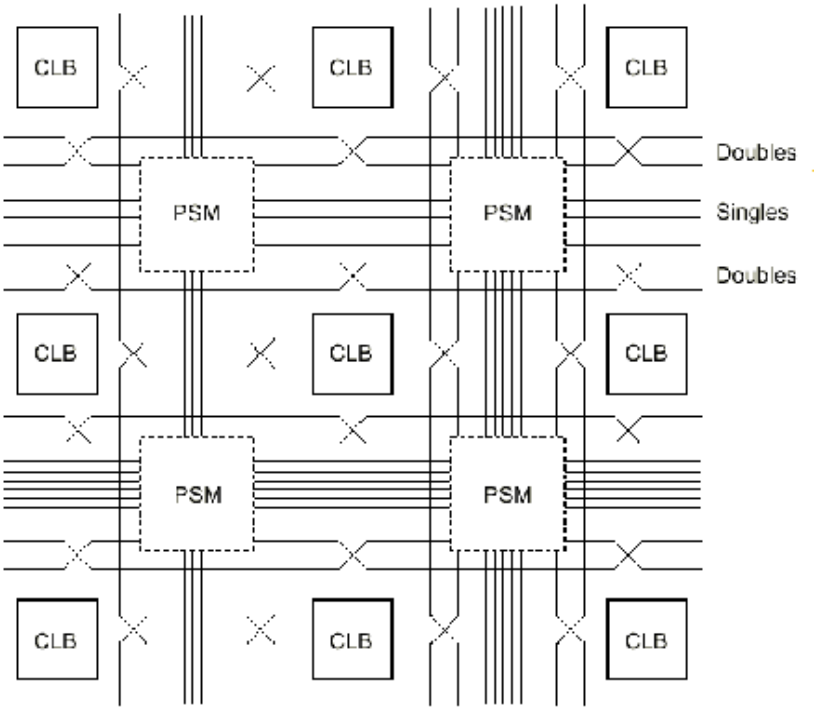


Figure 2.12. Xilinx 4000 routing architecture.

2.3.2. The Actel Routing Architecture

General routing architecture of Actel FPGAs is shown in Figure 2.13. The routing architecture is not symmetric and switch blocks are not separable as a block. This is due to existence of more general purpose horizontal wire segments. The connectivity of the input and output pins are different. Each input pin can connect to all the tracks that are on the same side with the pin. Output pins can connect to two channels above and below the logic block. Also all vertical tracks can make a connection with every incident horizontal track. This provides much more connectivity than Xilinx architectures. But this also leads to more switch requirement that contributes extra capacitive loading. Each horizontal channel consists of 22 routing tracks. These tracks are in different lengths providing that less number of switches are needed. Switches are the antifuse type and make area requirements minimum. The routing architecture of the Crosspoint FPGA is similar to that of Actel. Also Quicklogic architecture uses antifuses but has some different segments [4, 16].

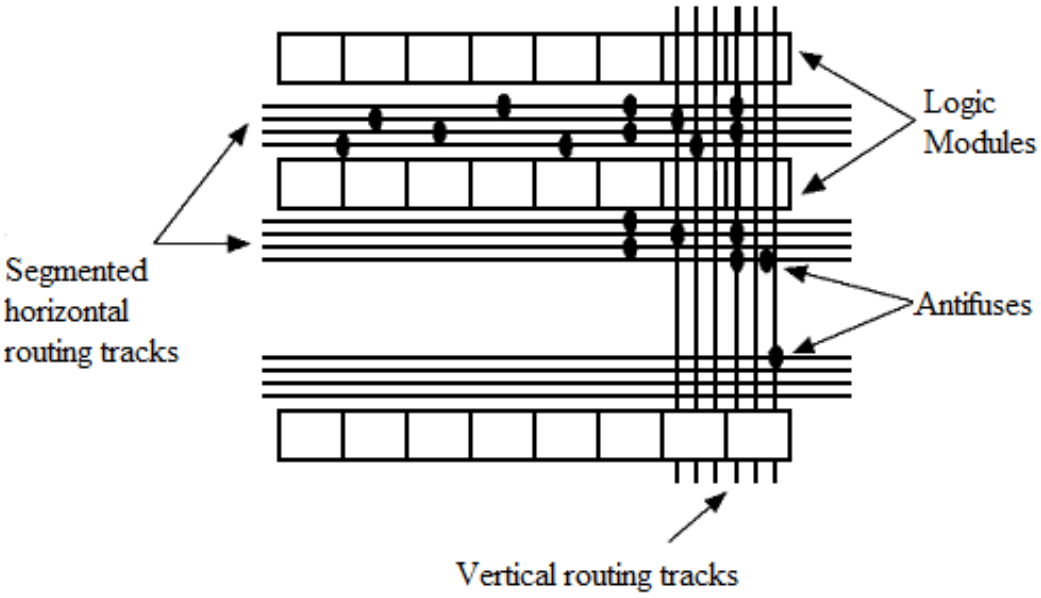


Figure 2.13. General Actel FPGA routing architecture.

2.3.3. The Altera Routing Architecture

The routing architectures of the Altera is shown in Figures 2.14 and 2.15. This architecture differs from the others since it has a hierarchical structure. It is beneficial in terms of utilizing the logic blocks to obtain better density and performance. The details of local routing can be seen from Figure 2.14 that shows an array of logic blocks and x are denoting connection points. These points are floating-gate transistors as described in Section 2.2.3.

Vertical tracks make the connections to the I/O blocks, logic expanders, outputs of all logic blocks in this logic block array and outputs of other logic blocks from other logic block arrays. This connection scheme has the advantage of a very simple and regular routing design. But, this also requires many switches and must be in an efficient order. Another important advantage is that the delay through any track is predictable since tracks have the same lengths. Hence, overall performance of the system becomes easy to predict. The drawback is that segmented wire architectures are much faster than this architecture.

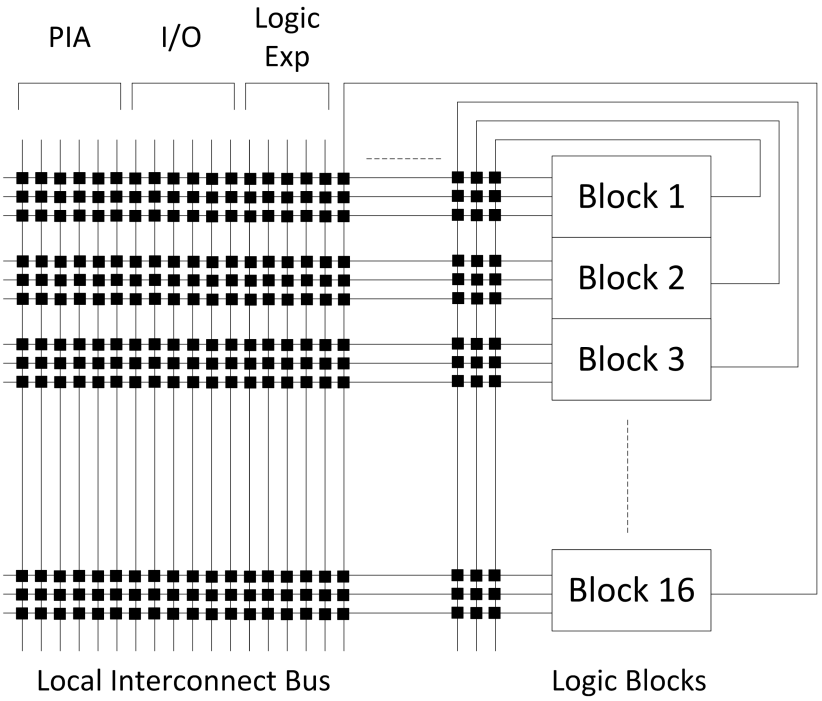


Figure 2.14. Altera MAX 5000 local routing architecture.

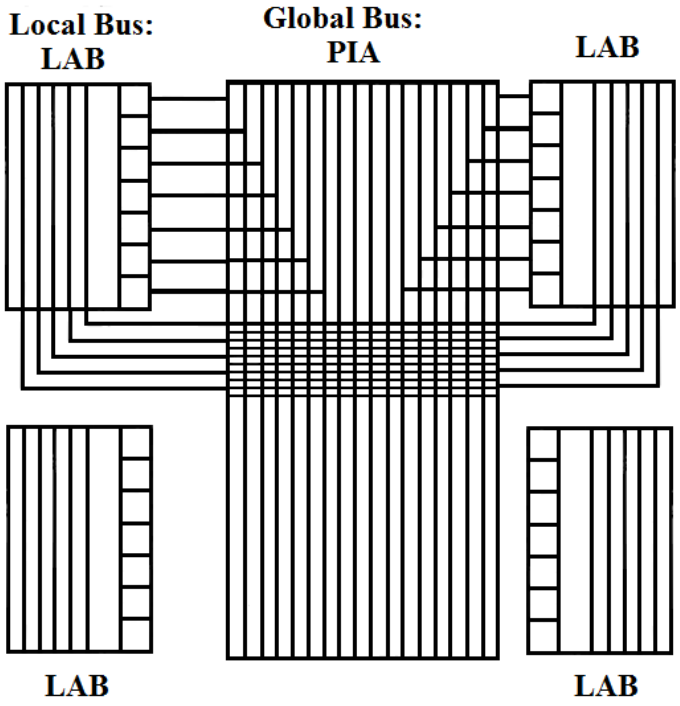


Figure 2.15. Altera MAX 5000 global routing architecture.

3. ANALOG PART OF FPMA ARCHITECTURE

In Section 1.2, the factors that urge the development of FPAAs are discussed. This task is quite difficult to realize as fast as its digital counterpart, FPGA. Analog circuits use less power and occupy less area to realize some important signal processes, such as multiplication and integration, compared to digital circuits. As previously discussed, it is impossible to eliminate analog circuits from electronic design. Hence, designers should easily be able to use them in their designs. Complexity of analog circuits stems from large number of design trade-offs compared to digital circuits. And they are restricted to analog designer's choice for the level of complexity. But, a circuit design with different functions may help these problems and provide a number of solutions to any user if the system is an FPAA design. In this case, user does not need to have a good analog knowledge [23].

In literature, a number of FPAA designs are presented. However, some of these have limited bandwidth due to circuit topologies that they use. Some of these topologies are switched-capacitor techniques [24], subthreshold MOS operation [1], extensive use of global interconnections [8]. A more detailed FPAA review is presented in [7].

Another design choice that greatly affects the performance and tuneability is whether being based on switched-capacitor (discrete time) or continuous time design. If the choice is switched-capacitor, then tuneability is easy and can be controlled digitally. Its drawbacks are limited bandwidth and area used for digital control. On the other hand, continuous time topologies have the advantage of high bandwidth but less configurability compared to switched-capacitor ones [25].

There is also another important design choice: whether to use voltage-mode or current-mode signals in the FPAA. The circuit designs for voltage mode signals are well developed and these signals also have high fanout. These properties make them first choice for some FPAA architectures [8, 26–28]. However, the simplicity of current mode circuits for implementing some functions with high accuracy and high bandwidth

make them preferable to the voltage-mode circuits [25, 29, 30].

Granularity is another design choice for the FPAA architectures. For example, in switched-capacitor designs, main elements are operational amplifiers (OPAMPs), switches and capacitor arrays. The granularity is about how largely these components are grouped. The size of the groups such as macro block or building block is the level of the granularity. In macro blocks, a CAB will contain its own OPAMP, switches, programmable capacitor arrays and a control unit. This macro block can perform a large variation of functions such as integration, summing, sample-and-hold, programmable gain amplification, etc. Most of the switched-capacitor based FPAAs [24, 28, 31] have their CABs implemented at this granularity [32]. On the other hand, building blocks may be consisting of transistor pairs, simple switches, operational tranconductance amplifiers (OTAs), resistors, capacitors, transmission gates, etc. Using both macro blocks or building blocks the following applications can be obtained:

- sample-and-hold circuits,
- gain stages,
- comparators,
- sum and difference amplifiers,
- integrators,
- differentiators,
- limiters,
- peak detectors,
- zero crossing detectors,
- low and high Q biquad filters,
- ladder filters,
- full wave rectifiers,
- relaxation oscillators,
- sine wave oscillators,
- PLLs,
- variable gain amplifiers,
- voltage control oscillators,

- multipliers,
- reconfigurable neural networks,
- vector-matrix multipliers [7, 33].

Also, some comparisons between the FPAAs can be found from the literature [7, 34, 35].

Last but not least, the switch and routing architecture choices are quite important for FPAA design, because the number of switches in the signal path and length of the wire tracks significantly affect the performance of the analog circuits.To keep the signal quality, as few as possible switches should be used. Longer wire segments can be used to minimize the number of switches and hence distortions due to the switches. Routing architectures related to wire segments are similar to ones for FPGAs which are introduced in Section 2.3. The switch types for FPGA are introduced in Section 2.2. These switches are also used in FPAA architectures. Antifuse type of the switches are generally used for high reliability, high performance circuits that are used in space applications for instance. Floating-gate switches are generally used for the reconfigurability part of the design since their voltages can be adjusted externally. Mostly, SRAMs are used in designs for their simplicity. Also, they do not require additional manufacturing process steps.

3.1. Configurable Analog Blocks (CABs)

As previously mentioned in Section 1.2, CAB design choice is greatly dependent on some factors such as routability, functionality and performance. In literature, various design choices can be found. Some of them find it easier to choose switched-capacitor architecture since this architecture has the advantage of fine tuneability due the being digitally controllable. Some of them seek high performance and high bandwidth, and choose continuous time architecture, generally current mode circuits in their CAB. Some architectures will need no switches and achieve to obtain very high bandwidth compared to other architectures.

In this section, the CAB types used in the FPAA architectures will be introduced. A general FPAA concept is shown in Figure 1.2. Their contents, functions, routing architectures, switch types and signal modes will be represented.

In [1], a subthreshold-based FPAA utilizing current-mode techniques is presented. Pass transistors, as illustrated in Figure 3.1b, were used as switches. Current-mode cells, as shown in Figure 3.1a, used in the CABs and obtainable functions are given in Table 3.1. Overall CAB design is given as shown in Figure 3.2. A four-quadrant Gilbert multiplier and a neural network were implemented with the elements of this CAB.

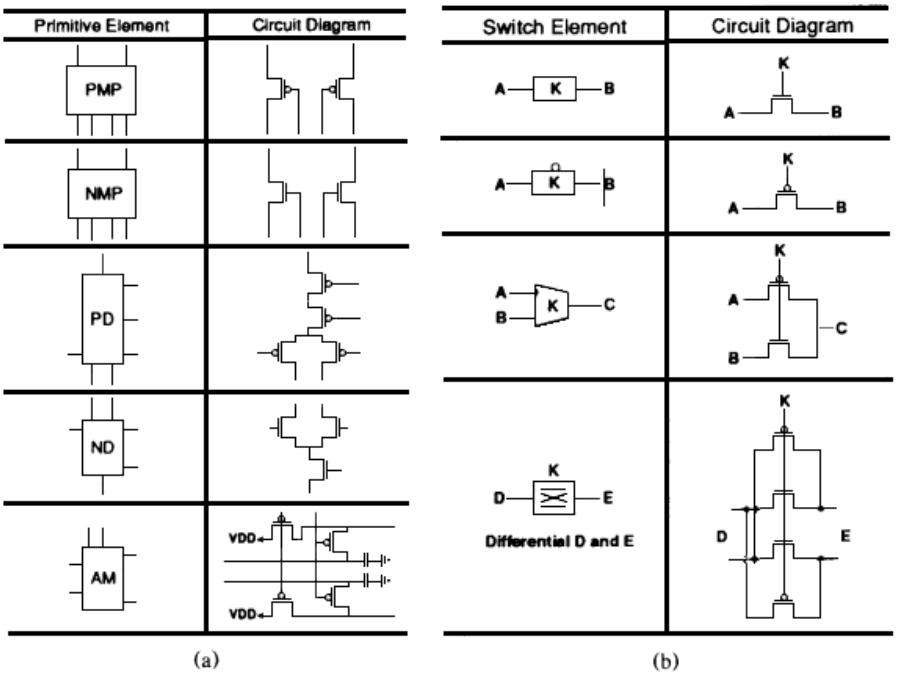
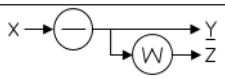
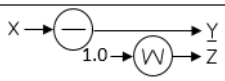
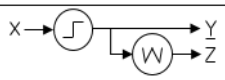
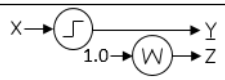
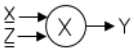
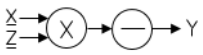


Figure 3.1. (a)Current-mode cells (b)Pass transistors as switch elements.

The author of the [1] changed his architecture in [8] to MOS tranconductor based FPAA. The underlying reason of this change is the parasitic effects of the pass transistors. These effects are restricting linear operation of analog circuits. By using MOS transconductor structure as switches and interconnection network, these effects are eliminated and high linearity, noise immunity and area efficiency are obtained. Four-transistor transconductor is shown in Figure 3.3a. Biquad and VCO are implemented

Table 3.1. Configurations of the CAB in [1].

Configuration bits			Description	Symbol
1	2	3		
1	1	0	Buffer from X to Y Z = X * W	
0	1	0	Buffer from X to Y Z = W	
1	1	1	Comparator from X to Y Z = X * W	
0	1	1	Comparator from X to Y Z = W	
0	0	1	Signal multiplier Y = X * Z	
0	0	0	Signal multiplier (buffered) Y = X * Z	
1	0	1	Constant multiplier Y = X * W(Z)	
1	0	0	Loading weight W from G1 and G2 Loaded at AM as $\overline{W}$ = -3V and stored as $\overline{W}$ = +3V	

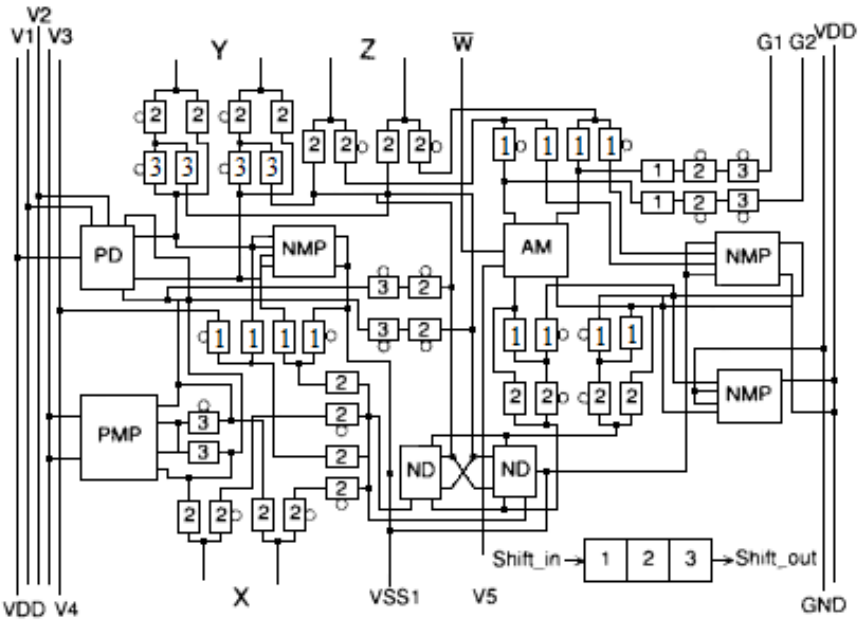


Figure 3.2. Schematic diagram of the CAB in [1].

using this FPAA. A slight modification, as illustrated in Figure 3.3b, were applied to the transconductor to be able to prevent the under-utilization of interconnection network and obtain a series of programmable linear resistors. Biquad, full-wave rectifier and analog multiplier are some applications implemented using this FPAA [2].

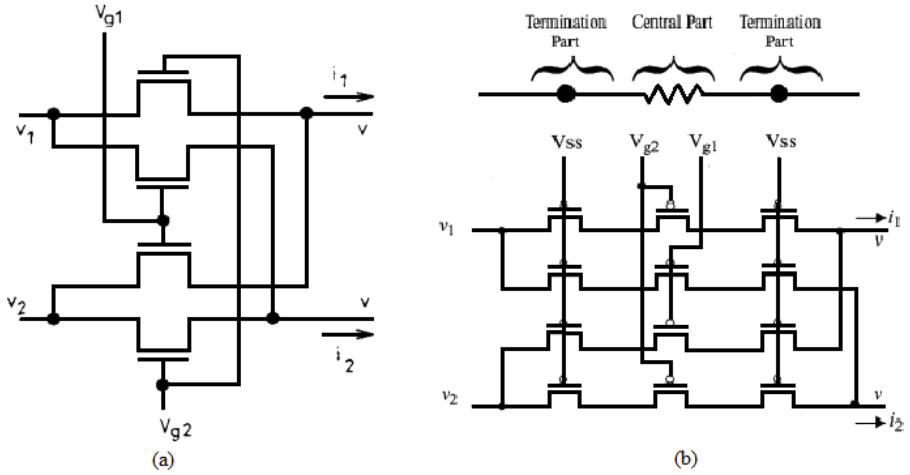


Figure 3.3. Transconductor used in (a) [1], in (b) [2].

Another current-mode approach is introduced in [25]. This architecture uses current conveyors, as shown in Figure 3.4, to obtain some basic operations such as current amplification, differentiation, integration, etc (see Figure 3.5). The interconnection network and switches are also four-transistor transconductors as in [8]. The resistors and capacitors seen in Figure 3.4b are programmable and provide a sufficient range of RC constant for the applications. Non-inverter amplifier, inverter amplifier, first order low-pass filter, analog multiplier, comparator and rectifiers are some applications that can be build using this FPAA.

A current-mode based FPAA for signal processing applications is presented in [36]. This architecture uses Configurable Analog Cells (CAC) as a configurable subblock. It is configured to perform three functions. These are integrator, amplifier and attenuator. Current-mode is chosen to use for low voltage applications. Also, it provides wideband frequency response and is simple to implement. The proposed topology and schematic of CAC is given in Figure 3.6. Biquads and some other filters were implemented using this topology.

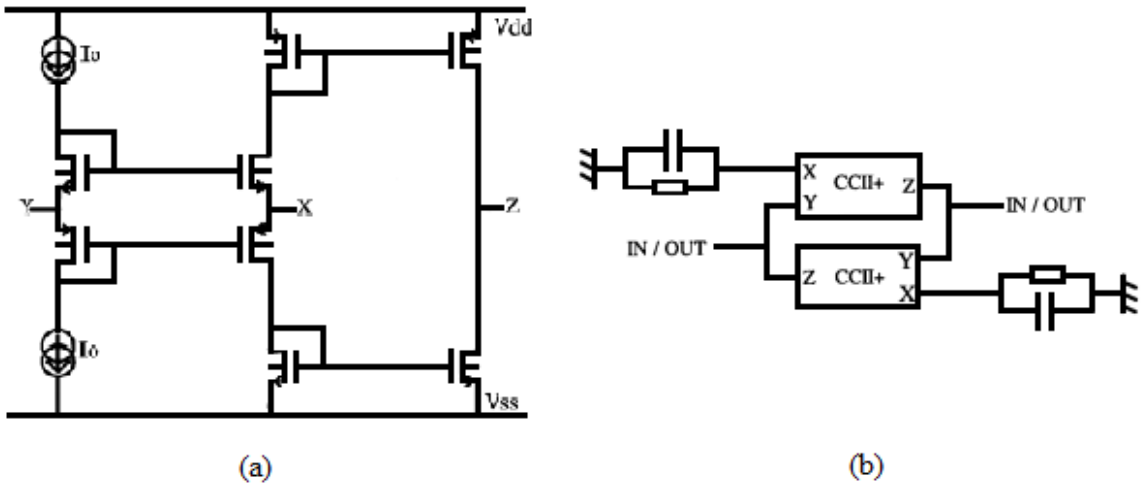


Figure 3.4. (a)Schematic of current conveyor, (b)analog elementary cell as a CAB.

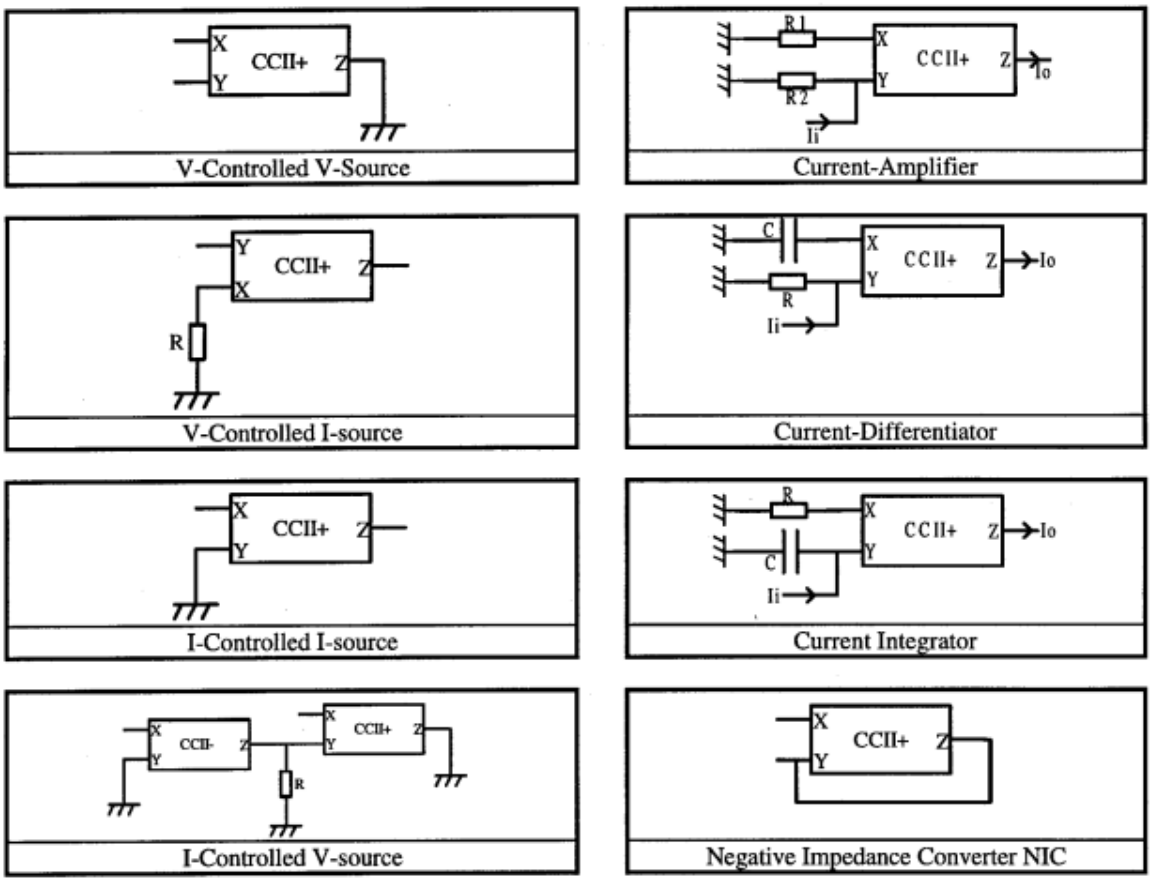


Figure 3.5. Current conveyor applications.

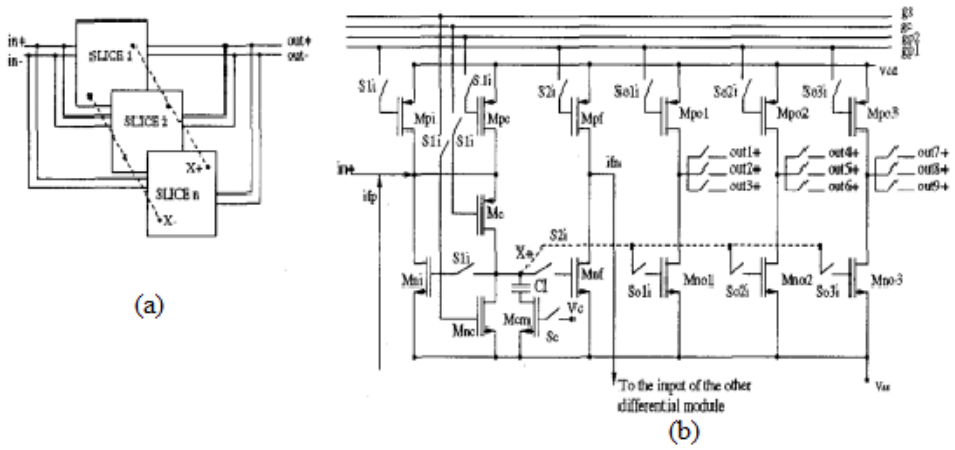


Figure 3.6. (a)The proposed topology (b)Circuit schematic of the CAC.

Another FPAA topology using current feedback operational amplifier (CFOA) is presented in [3]. The CAB used in this architecture is shown in 3.7. This architecture is used to implement grounded-C second order low, high, band and all pass filters.

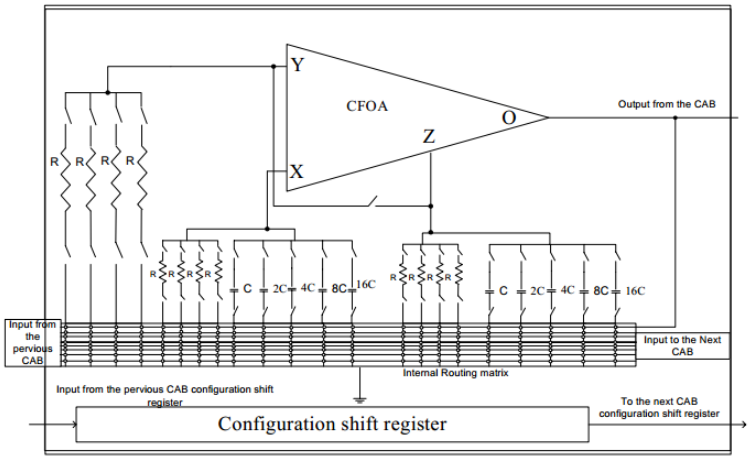


Figure 3.7. Proposed CAB in [3].

A multiple input translinear element (MITE) is chosen in [37] as the core of the programmable analog array. The main properties of these translinear elements are having a near infinite input impedance and an exponential voltage to current relationship when operated in deep subthreshold. A field programmable MITE array is shown in Figure 3.8. A squaring circuit, a square root circuit, a second order translinear

loop, a vector magnitude circuit and a first order log-domain filter is implemented and demonstrated.

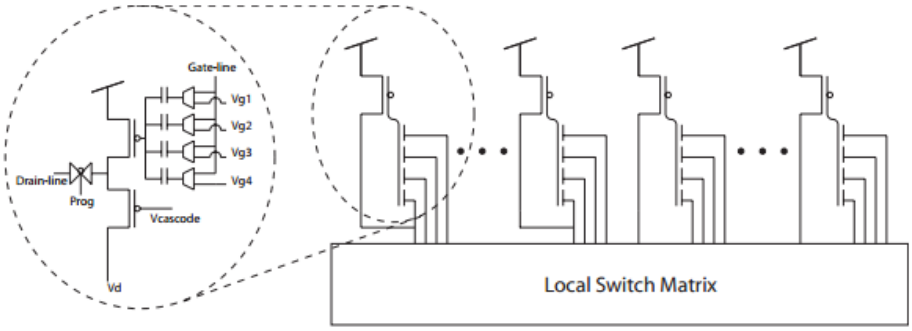


Figure 3.8. Architecture of a field programmable MITE array.

Also, programmable analog vector multipliers are demonstrated in the literature [38]. Matrix multiplication operation finds application in neural networks and signal processing algorithms. Advantage of multiply-add circuit in current mode is the ease of adding current signal by just joining them in a node. Its implementation is shown in Figure 3.9 and provides low power, high density and fast performance compared to digital counterparts.

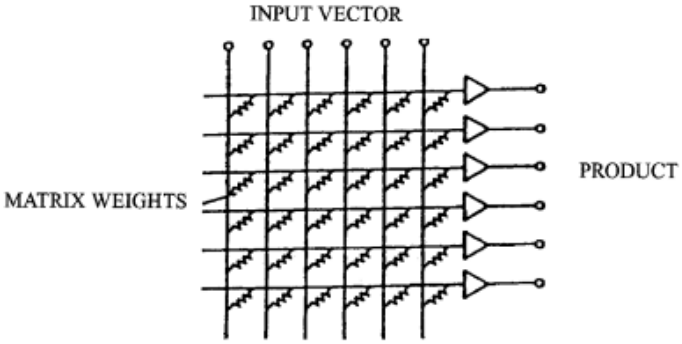


Figure 3.9. Circuit implementation of matrix-vector multiplication.

Another current-mode based FPAA is introduced by Analogix [39]. It is intended to realize continuous, fuzzy, and multi-valued logic circuits. It uses bipolar transistor array and works up to several hundred MHz. The implemented FPAA has an array of analog homogeneous analog cells which are locally connected to four neighboring

cells without switches. Programmable cell functions are shown in Figure 3.10. Some detailed architecture design and applications are given in [23, 40], respectively. Some of the applications are linear filters, nonlinear signal processing applications including rank filter, PLL, etc.

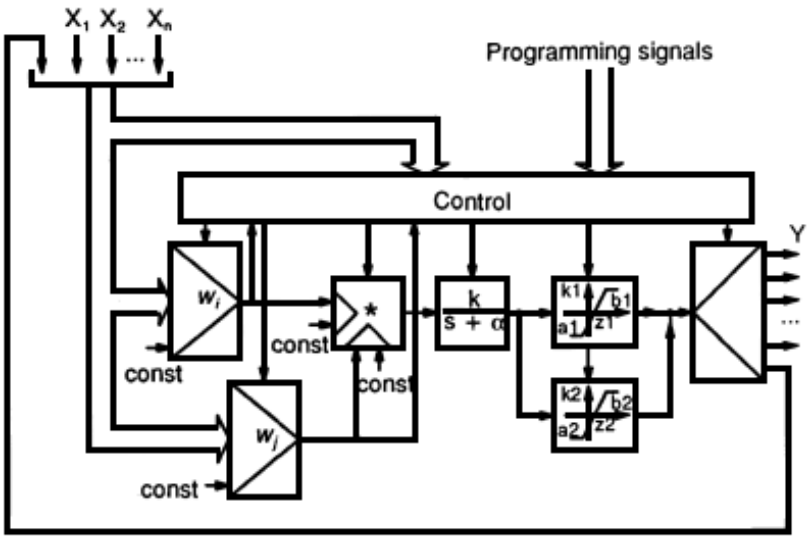


Figure 3.10. Functional diagram of the programmable cell.

A programmable high-frequency OTA is presented in [41]. In Figure 3.11, structure of the CAB and proposed programmable OTA schematic are shown. It is used for analog signal processing applications such as addition, subtraction, amplification, attenuation, integration and filtering of signals of frequencies from several kilohertz up to a few megahertz.

A good example of switched-capacitor architecture is presented in [32]. Each of the CABs contain an OPAMP with feedback switches. Interconnection networks are used to connect programmable resistor arrays and programmable capacitor arrays. The switches in the signal path are also used as charge transfer mechanism for switched-capacitor circuits. This scheme helps achieving high speed operations. The proposed switched-capacitor (SC) CAB and programmable capacitor array are shown in Figure 3.12. This SC technique offers very high flexibility. A/D and D/A converters, interpolation filters, VCO and some filters can be implemented using this

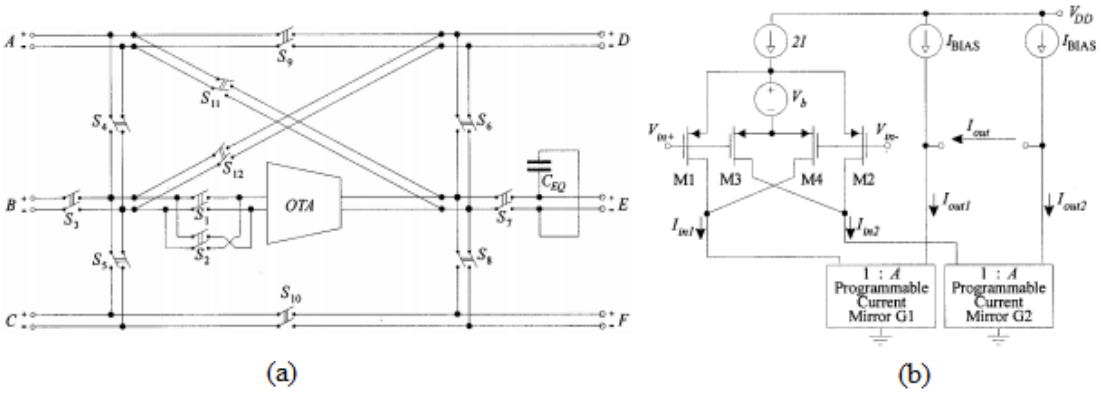


Figure 3.11. (a)Structure of CAB (b)Schematic of the CMOS programmable OTA .

architecture. Embedding of these circuits in the FPAA is described in the article.

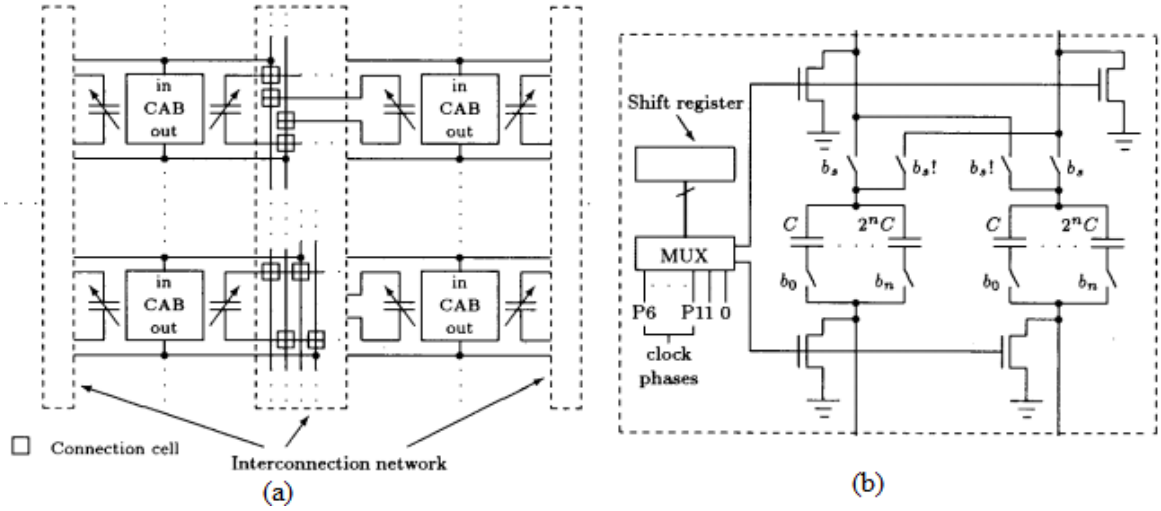


Figure 3.12. (a)Proposed SC based FPAA architecture (b)Programmable capacitor array .

Another example of switched-capacitor circuit is presented in [42]. It is a voltage-mode switched-capacitor based architecture. SC architecture is chosen for the following reasons: (i) SC circuits are easy to configured externally, (ii) the clock frequency help to obtain different values capacitors without changing the physical dimensions, (iii) accurate adjustibility of capacitors is quite possible compared to other architectures. The CAB structure is illustrated in Figure 3.13. Interconnection network between two

CABs is shown in Figure 3.14. The two clocks are adjusted for the desired operation mode and for the bandwidth. Low pass, band pass and biquad filters, programmable amplifiers, balanced modulators and signal generators are implemented in this architecture.

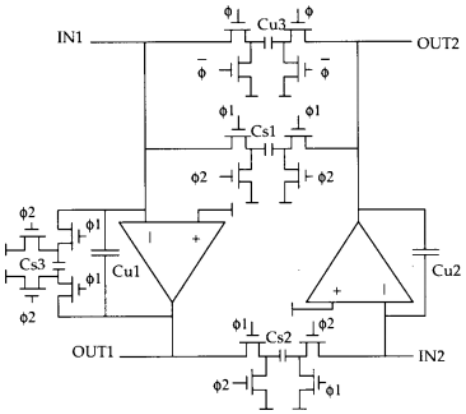


Figure 3.13. Switched-Capacitor CAB.

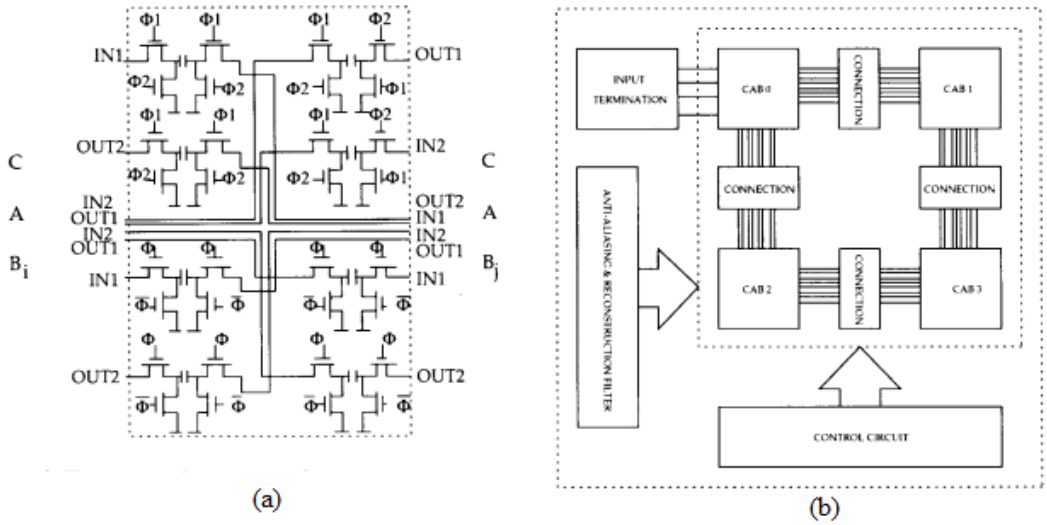


Figure 3.14. (a) Interconnection scheme of the FPAA (b)overall structure of the FPAA.

Another interesting FPAA architecture is developed and implemented in [34] and is based on floating-gate structure. The programmable interconnections and some of the circuit elements are implemented using floating-gate transistors. Two kind of CABs with different granularities are used and shown in Figure 3.16c. Some of the

implemented circuits are FGOTA, current-mode translinear circuits, programmable offset buffer, folded gilbert multiplier, vector matrix multiplier, analog speech processor and AM receiver. Architecture of the FPAA is shown in Figure 3.15.

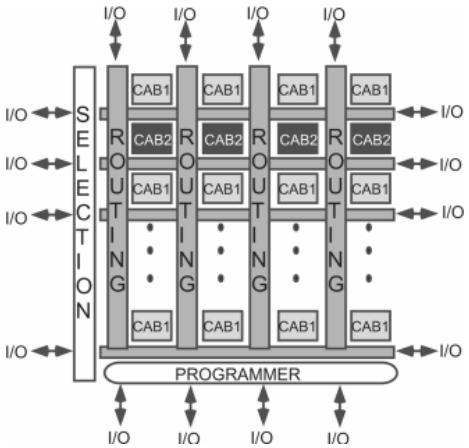


Figure 3.15. Overall structure of the FPAA.

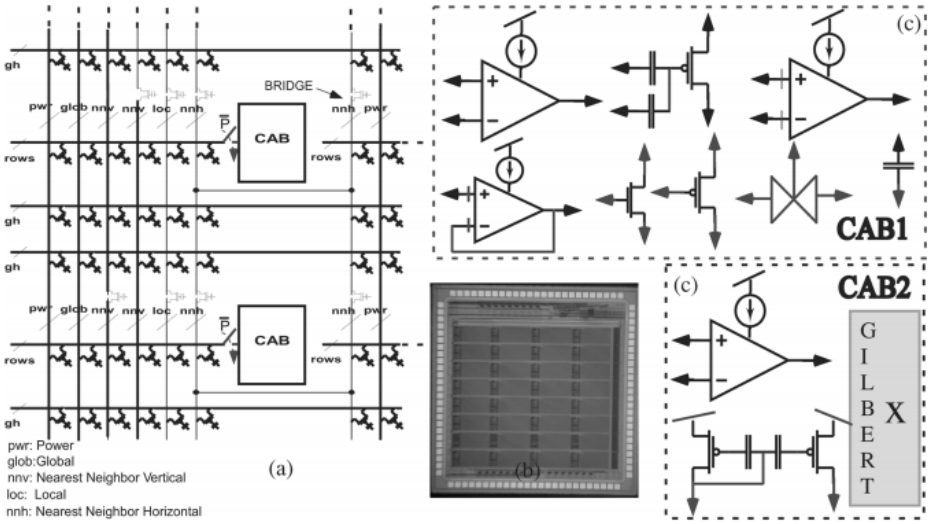


Figure 3.16. (a)The different categories of routing lines interconnecting the CABs. (b) The die photograph of the chip. (c) The components in the two types of CAB.

In [35], tunable OTAs are used as the main circuits in the CABs. Seven CABs are connected to each other without switches. Filters are designed and simulated up to a hundred MHz in this architecture. OTAs can be turned on and off, hence they behave like switches. The array topology and schematic of the CAB is shown in Figure

3.17a and 3.17b, respectively.

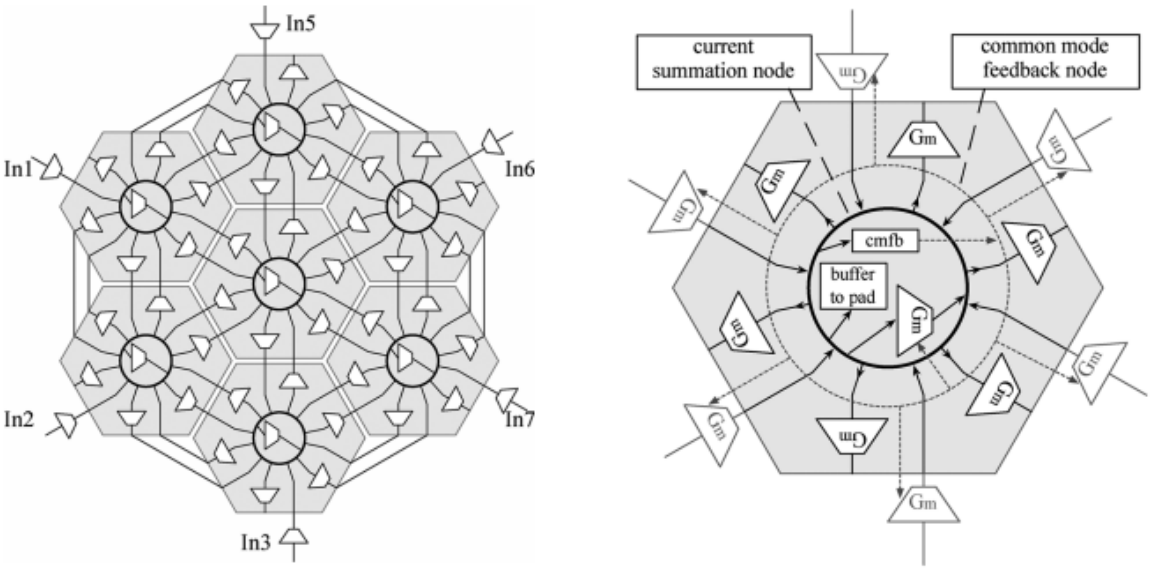


Figure 3.17. (a)Array topology of the implemented FPAA (b)Schematic of the CAB.

3.2. Analog Blocks in FPMA

With the evolving technology, scaling of the transistor has already reached the ultra submicron dimensions (less than 100nm). Thus, very high performances can be obtained for digital circuits. On the other hand, smaller transistors will cause some problems for analog circuits. These are increased leakage current, reduced intrinsic gain, reduced voltage room and poor matching. These problems may cause functionality loss in analog circuits. It is very difficult to maintain signal integrity. In Section 1.4, the decreased supply voltage and gate leakage problems are introduced. In [43], manufacturing variations are defined and device mismatch is defined as random differences between the device characteristics that are stemmed from device-to-device variations for identical MOS transistors. The mathematical expressions are given and they show that accuracy can be improved by increasing the total transistor area, which also means increasing power consumption. These mathematical expressions are given for down to 180nm technology. On the other hand, in [14] it is shown that although good matching is obtained by increasing the area, gate leakage problem will dominate scaling down from 130nm technologies. In other words, increasing the area will also

increase the gate leakage for the technologies below 100nm. Hence, the intrinsic gain of the transistors will be very low compared to old technologies and analog performance of the circuits will get worse. Also, as previously explained, a bias insensitive frequency f_{gate} would estimate the effect of the gate leakage. This frequency will show the MOS behavior according to the frequency of the input signal at the gate. If the frequency of the input signal is larger than f_{gate}, then input impedance is capacitive and the MOS transistor behaves normally. Otherwise, below f_{gate} input impedance is resistive and gate leakage is dominant. In [14], it is also shown that input impedance is resistive below 0.1 Hz in 180nm technology and 1 MHz in 65nm technology. According to [14], the effect of the leakage current can be eliminated by using either high supply voltages for critical parts or thick-oxide transistors.

Determining the blocks in FPMA is a tradeoff between performance and functionality. Therefore, it is very important to choose the most appropriate circuits with the satisfying performances. For this purpose, a literature survey is done to choose the blocks to be used in FPMA. All the papers in the last three years are searched within the journals; IEEE Journal of Solid State Circuits (JSSC), IEEE Transactions on VLSI Systems (TVLSI) and IEEE Transactions on Circuits and Systems (TCAS), and within the conference papers; International Solid-State Circuits Conference (ISSCC), IEEE International Symposium on Circuits and Systems (ISCAS) and IEEE International System-on-Chip Conference (SOCC). The mostly used analog blocks are listed in the Table 3.2.

Amplifiers are the most important and essential building blocks in analog designs. Most of the amplifiers in the articles above are general purpose operational amplifiers such as OPAMPs and OTAs, the remaining ones are used for specific applications. For example, the signals coming from the outside of IC may require amplifying before being affected from the circuit noises. Hence, low noise amplifiers (LNAs) are needed as a preamplifier stage. Variable gain amplifiers can be obtained using OTAs, which have transconductances controlled by bias currents.

Some other frequently used blocks are comparators, oscillators, filters, ADCs and

Table 3.2. Most used analog blocks in last three years.

Block	Number of times appearing in an article
Amplifiers	30
OPAMP, OTA, etc	236
VGA	41
LNA	40
Power amplifier	13
Comparators	67
Oscillators	61
Filters / Integrators	60
ADC	57
DAC	57
Mixers / Multipliers	38
Biasing circuits	27
Charge pumps	24
Buffer circuits	15
PLL	15
V/I converters	11

DACs. In the digital part of the FPMA, digital signal processing implementations will be done, therefore ADC and DAC blocks should be designed as a priority. Filters will not be designed as separate blocks. They will be implemented by connecting the amplifiers and capacitors. The expected operating frequencies may be low, since these filters will be active filters. But, obtained frequencies will be enough for low frequency applications, such as audio applications.

Some of the remaining blocks in Table 3.2 will not be added to overall FPMA design since some will be implemented with the CABs and some will be unnecessary to use. For example, charge pumps are not needed since a system which require energy production will not be implemented. Buffer circuits can be obtained by operational amplifiers which are connected in negative feedback configuration. Biasing circuits also will not be included since DAC circuit can produce necessary biasing voltages. Also, DACs with biasing purposes will not require filtering the output signal, hence it will occupy a low area and can be used many places. Two input analog multipliers

that can also work as mixers and coefficient multipliers, which are commonly used in signal processing, will be included in the overall FPMA design. Finally, inputs and outputs of the analog blocks may be in the form of voltage or current; this situation will require V/I and I/V converters.

In this thesis, comparator, VCO and DAC are designed and simulated. Comparator and VCO circuits are demonstrated in this chapter while DAC design is represented in detailed in Chapter 5.

3.2.1. Comparator

One of the analog blocks that will be used in the FPMA design is comparator. In Figure 3.18, schematic diagram of the comparator is shown. This comparator will be used for comparing two different analog voltages. The transistors M3 and M4 are cross-connected from their gates. This connection scheme creates positive feedback and leads to hysteresis in the output. This hysteresis is adjusted to 20 mV sensitivity and can be seen from Figure 3.20. This value can be changed with different bias values. The comparator circuit has two input values, either logic HIGH (high = supply voltage) or logic LOW (low = ground). When the input voltage value is smaller than reference voltage value, then transistors M1, M4 and M5 are on and M2, M3, and M6 are off. In this case output is logic HIGH. When the input voltage value is larger than reference voltage value, then transistors M2, M3, and M6 are on and M1, M4 and M5 are off. Hence, logic LOW is observed at the output. Simulation results are shown in Figure 3.19.

3.2.2. Voltage Controlled Oscillator

Voltage controlled oscillators are generally a part of phase-locked loop design. PLLs plays an important role in electronic circuits. It is generally used to obtain a well defined time clock for synchronous digital circuits. It is advantageous over the LC type oscillator, since it has small area, wide tuning range and ease of implementation for different processes. Scaling to the new technologies with low supply voltages make

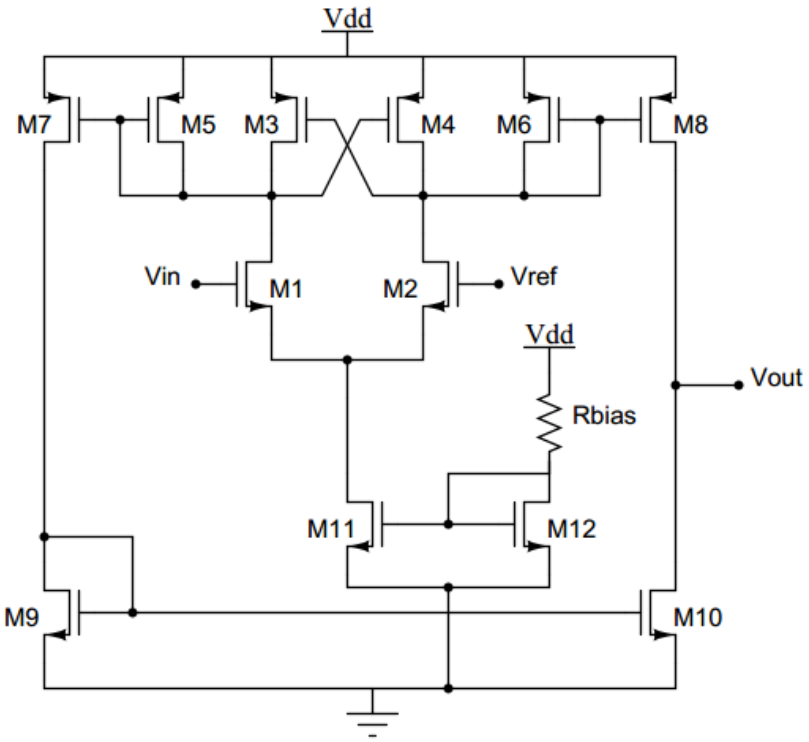


Figure 3.18. Schematic of the comparator.

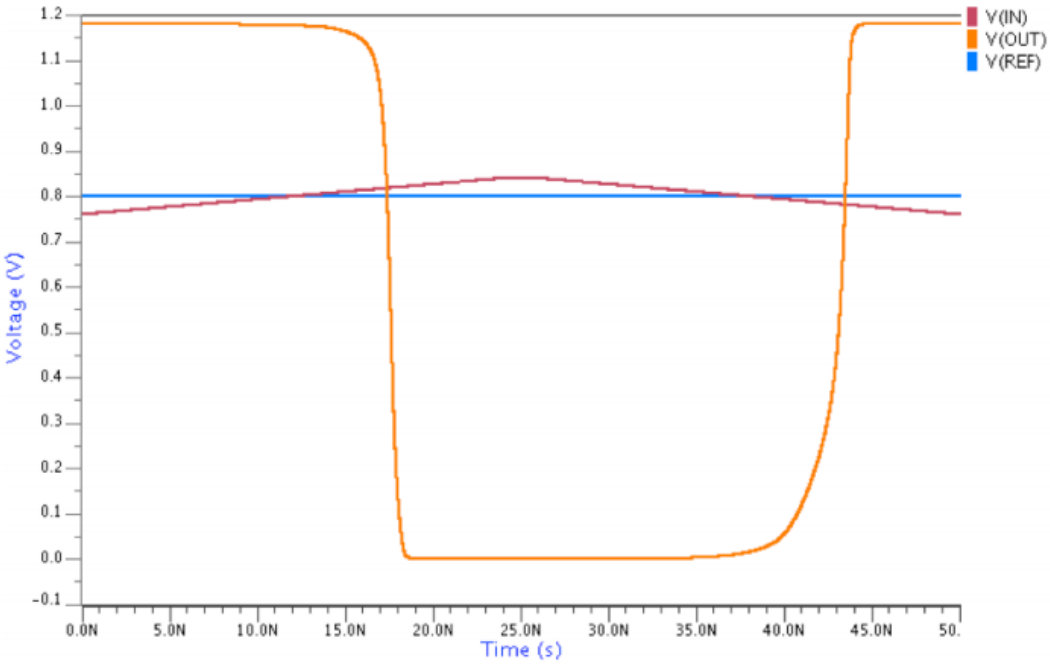


Figure 3.19. Simulation results of the comparator.

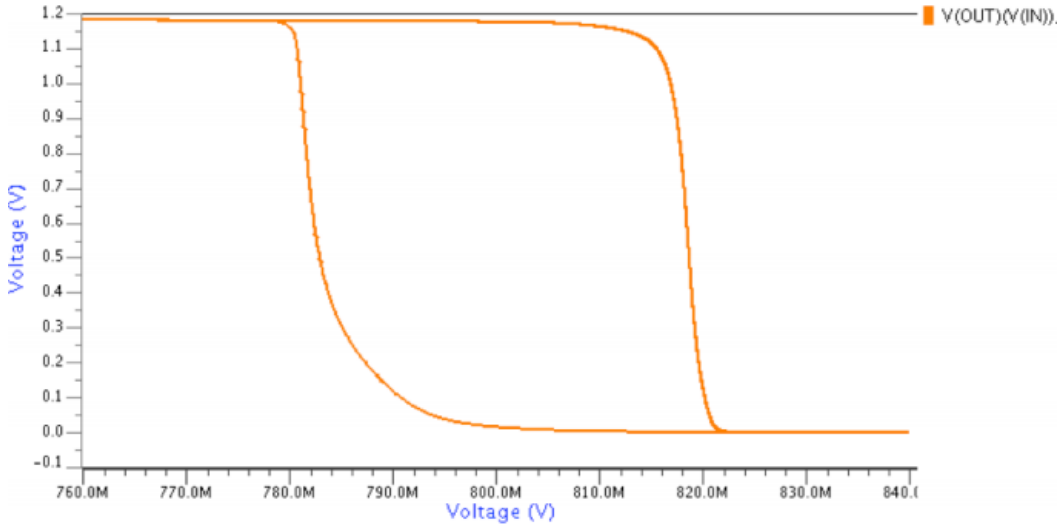


Figure 3.20. Hysteresis in comparator.

it difficult to obtain desired VCO outputs since switching activities under low supply voltages causes to distortions at the output. To eliminate these undesired behavior, the supply-regulated VCOs are used commonly. VCO tuning is also an important part of PLL loop bandwidth apart from low supply sensitivity designs.

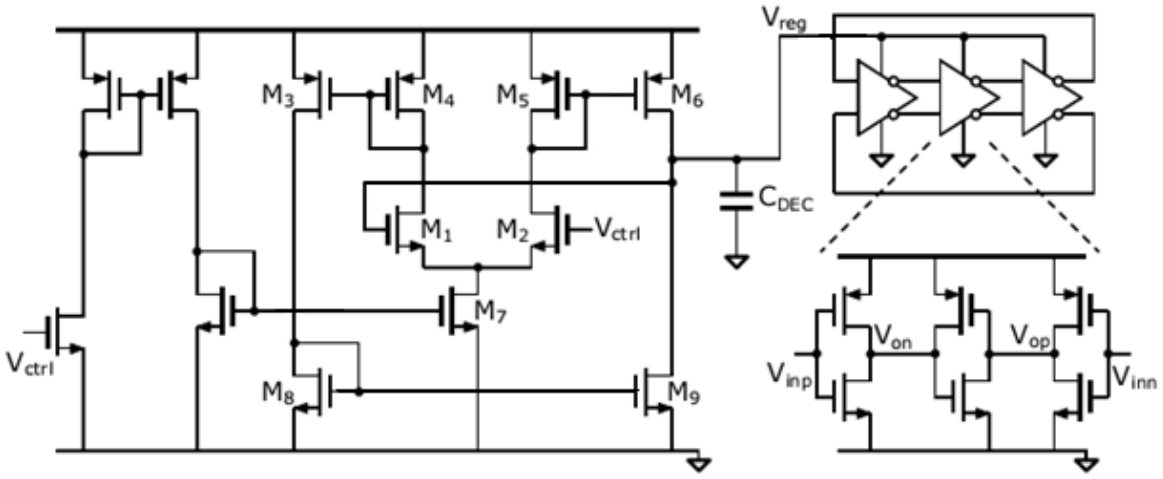


Figure 3.21. Voltage controlled oscillator.

For our FPMA project, the design in [44] is used. Figure 3.21 shows the schematic

of the design. The design consists of a supply regulator and a ring oscillator. Ring oscillator part is obtained using dual inverter delay cells. Each delay cell has a pair of inverters followed by a cross-coupled latch. The propagation delay of these delay cells are controlled by the regulated voltage V_{reg}. Supply regulator part has a bias circuit and a current mirror based OPAMP M1-M9. C_{dec} is used to filter the high-frequency supply noise. The low-frequency supply noise is eliminated by the loop gain of the amplifier. V_{ctrl} controls the bias current of the amplifier hence the bandwidth of the regulator. The change in VCO output frequency with changing V_{ctrl} is shown in Figure 3.23. By changing the V_{ctrl} values from 0.27 V to 1.2 V, an output frequency ranging from 833kHz to 48.5MHz is obtained. The frequency change curve is shown in Figure 3.22.

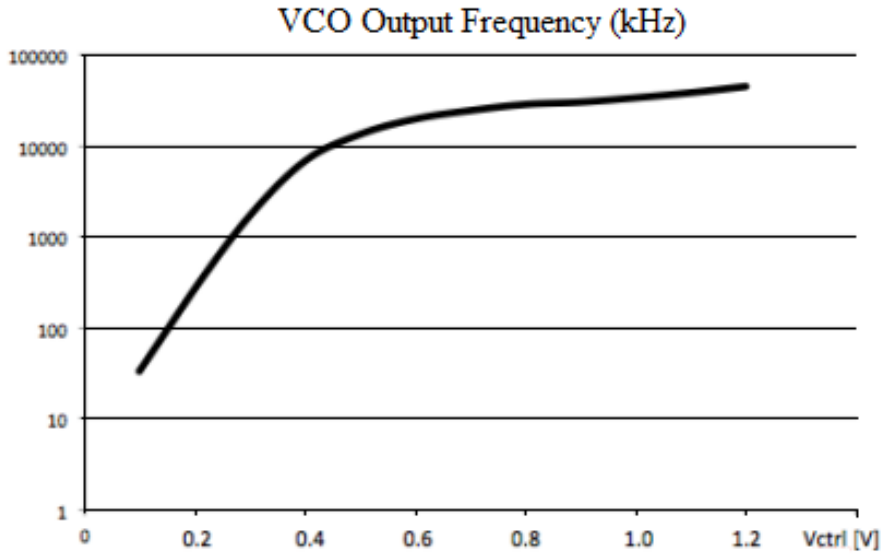


Figure 3.22. Frequency vs V_{ctrl} curve.

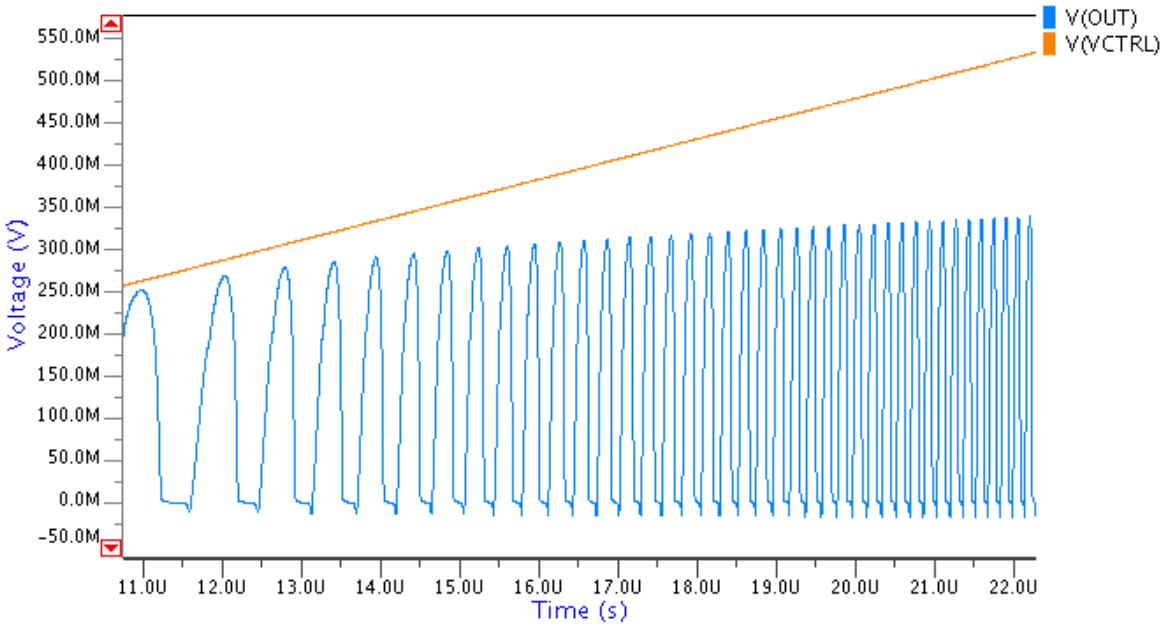


Figure 3.23. The change in VCO output frequency with changing V_{ctrl}.

4. INTERFACE PART OF FPMA ARCHITECTURE

4.1. ADC

FPMA design will consist of analog and digital cores which will require direct communication between each other. Data transmission will be done all the time. Therefore, ADC requirement is inevitable. Other from the FPMA design, ADCs are very common in all electronic systems. Since the digital complex circuits are available and they usually process the analog signals, they need ADCs as front end circuitry [45]. System requirements will be a determining factor for the choice of topology. Although there are two many ADC topologies, they can be classified into main categories. These are Nyquist rate and oversampling ADCs. The speed of the Nyquist rate ADCs are far better than the oversampling ones. On the other hand, oversampling ADCs are more accurate than the Nyquist rate ADCs. Other than these ADCs, there are also some other subcategories. These subcategories are shown in Figure 4.1.

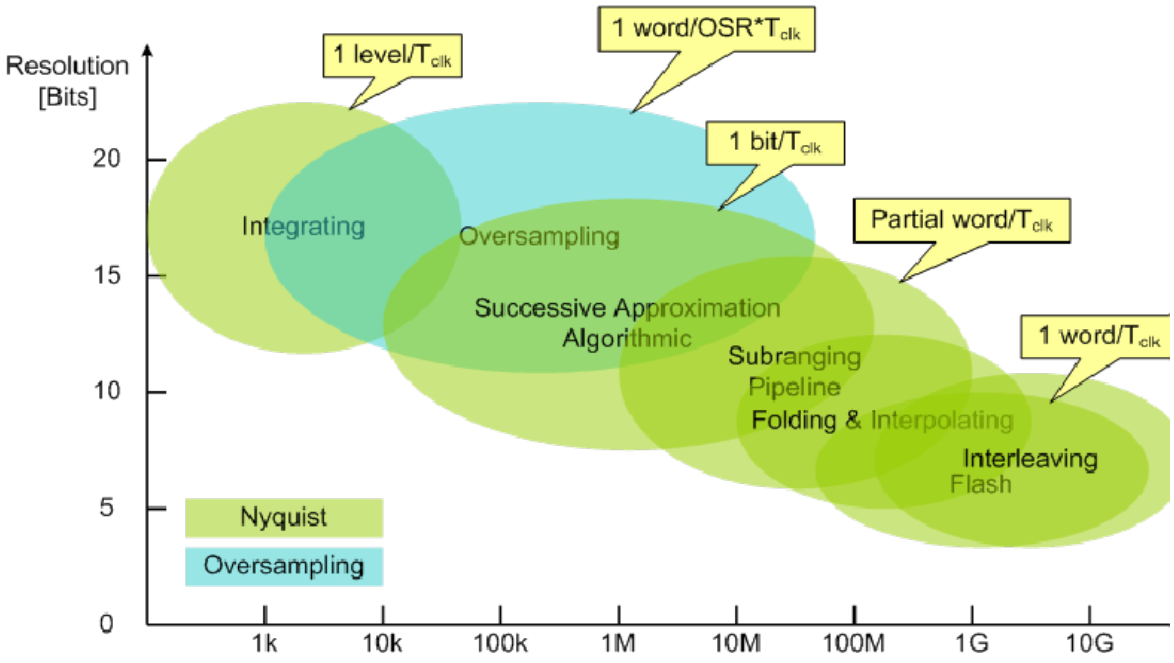


Figure 4.1. Accuracy-speed trade off graph for ADCs [46].

Pipeline topologies use OPAMPs and switched-capacitor structures to generate

residual. Also, they consist of $2^n - 1$ comparators, which consumes very large amount of power. The scaling through sub-100nm technologies also make using these topologies very hard since reduction in the supply voltage will lead to more power consumption. However, SAR ADC architecture is not affected too much because its analog part consists of a comparator and a capacitive DAC array. The speed of a SAR ADC is determined by the time required by the DAC to settle within 1/2-LSB. Large number of capacitor requirements due to increasing number of bits will lead reference generator to consume an important part of the overall power. The other important source of power consumption is dynamic power of the DAC.

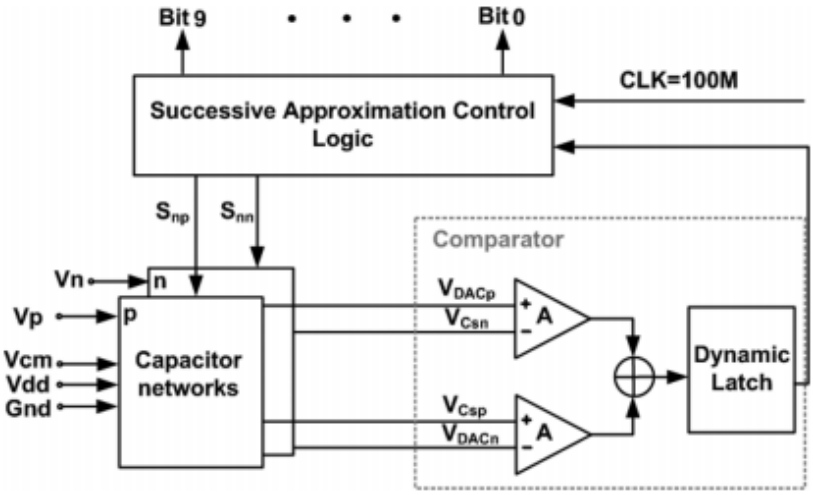


Figure 4.2. Overall schematic diagram of the ADC structure [47].

ADC design, as shown in Figure 4.2, achieves low power consumption for medium resolution and high conversion speeds [47]. Some methodologies are applied to realize the low power consumption. The architecture avoids reference generator to using power supply and saves the switching energy during the SA conversion. Using the supply voltage as the reference voltage of the ADC will lead to reduction in the effective SNR since this signal swing is larger than the typical signal range. This drawback can be eliminated by using a technique that doubles the input. This relaxes the limits of noise and offset, and also reduces the power consumption of the comparator. The overall architecture contains differential capacitor networks, dynamic comparator and SA logic

control. The differential capacitor networks are composed of 10-bit split schemes and sampling capacitors C_s which are needed for passive doubling. The comparator consists of four inputs preamplifier and dynamic latch and determine the value of each bit. The SA control logic will contain shift registers, bit registers and a switching logic block. SA control logic control the DAC operation to realize binary-scaled feedback mechanism during the successive approximation cycle. Sampling frequency is used in SA block.

4.2. DAC

Similar to ADCs, DACs are needed for data transmission and conversion between analog and digital circuits. An electronic definition is given in [48]: an electronic linear D/A converter is an electronic circuit that accepts at its input a set of electrical signal, that represent a digital code, and yields at its output an analog signal; i.e., in proportion to a reference electrical quantity as the input numeric code is to the full range of possible codes. In literature, there are many topologies that are used for D/A conversion. These topologies can be classified as follows:

- Resistive voltage division architectures,
- Capacitive voltage and charge division architectures,
- Current division based architectures.

4.2.1. Resistive voltage division architectures

In voltage divider topologies, a reference voltage is divided to M=2^n steps by using a resistor network. The main consideration for this type of architecture is to obtain necessary limits for INL and DNL, which are hard to achieve due to process mismatches between resistors due to temperature. A resistive voltage division based DAC is shown in Figure 4.3. This architecture has three stages for conversion: the first is a resistive divider, second is a network of electronic switches, and the third is an impedance adaptation buffer.

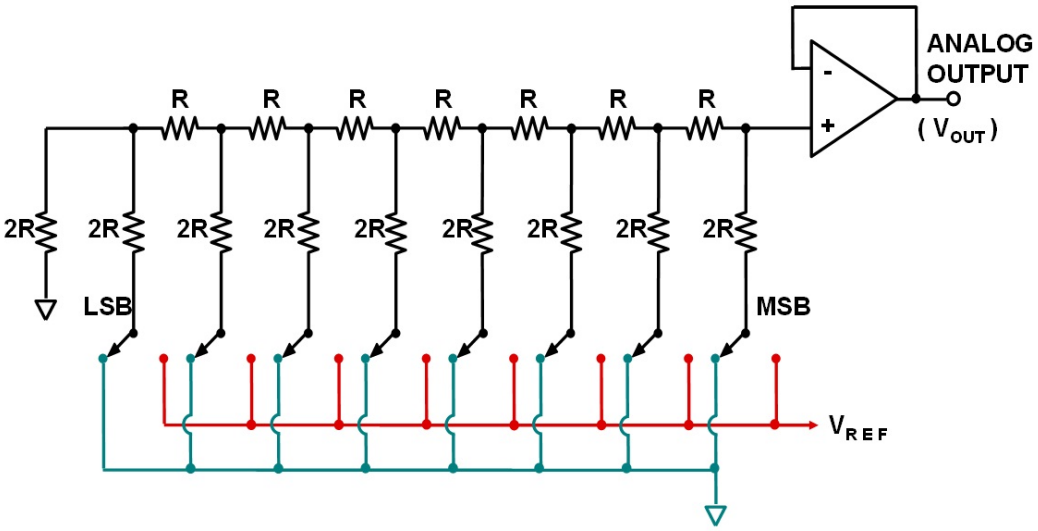


Figure 4.3. Resistive voltage division based DAC.

4.2.2. Capacitive voltage and charge division architectures

In these architectures, D/A conversion is done by using switched-capacitors based on the concept of charge re-distribution. In Figure 4.4, a binary weighted DAC schematic is shown. Switched-capacitor networks consist of capacitors, switches and amplifiers. An additional capacitor C is added to obtain $2^N C$ capacitance for an N bit converter at the common capacitor terminal. Before each conversion, the capacitor array is discharged in every conversion step via the switch S_d. Then, all the capacitors are precharged by reference voltage in their individual terminals. At the same time, the additional capacitor is grounded. Therefore, a total charge $Q = V_{ref}.C.2^N$ is deposited on the top plates of the capacitors. When the conversion starts, all the capacitors are either grounded or resumed to V_{ref} depending on their bits. The charge conversion law makes the stored charge in the top plate to re-distribute forcing a voltage at the top plate which is a fraction of V_{ref} according to the code [49].

SC based conversion is limited by:

- The matching accuracy of the capacitors,

- The speed and linearity of the voltage buffer,
- The large capacitance present in the node of the top plates of the capacitors,
- Thermal noise considerations, that dictate large capacitance, thus large size.

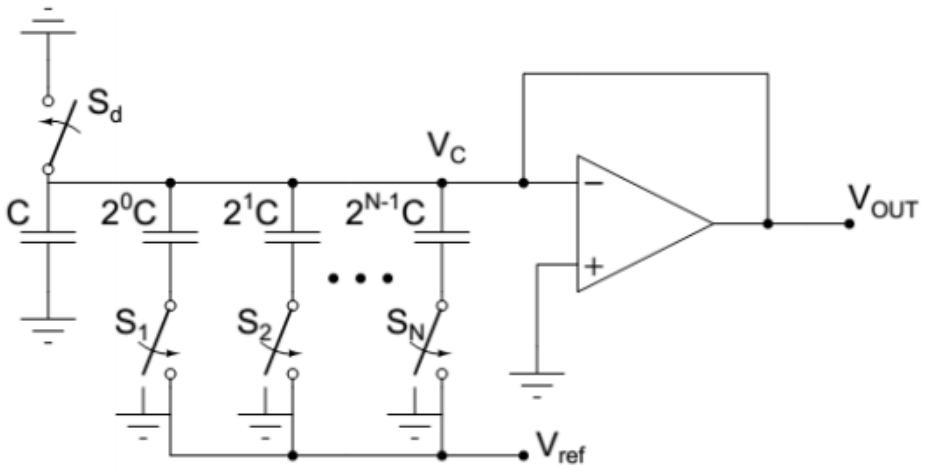


Figure 4.4. Capacitive voltage and charge division based DAC.

4.2.3. Current division based architectures

A parallel DAC schematic based on current division is shown in Figure 4.5. This architecture is known as current steering architecture. The circuit consists of a reference current replication network, a network that combines binary weighted currents to generate the output value and a current to voltage converter. The inventor of this very popular architecture is B.M. Gordon [48]. This architecture has proven itself as a high speed design because the current steering nature of the circuit is very fast. Current steering DACs are main candidates for the high speed and high resolution applications. In this thesis, a high speed current steering DAC with digital error correction scheme is presented.

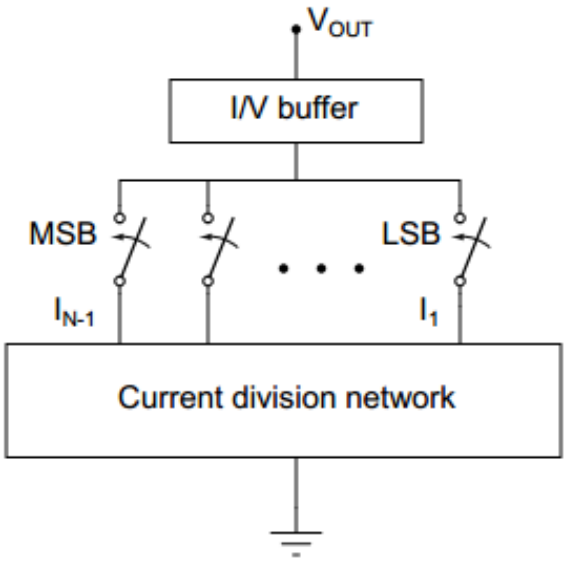


Figure 4.5. Current division based DAC.

5. DIGITALLY ASSISTED DAC DESIGN IN FPMA

In Section 4.2, the most commonly used DAC architectures are introduced. The evolving technology requires high speed and high resolution mixed-signal blocks. One of them is, as previously stated, the digital-to-analog converter (DAC). For this reason DAC needs to have the properties of being fast and having a great accuracy. Figure 5.1 shows that current steering architecture is best suited for the reasons mentioned above.

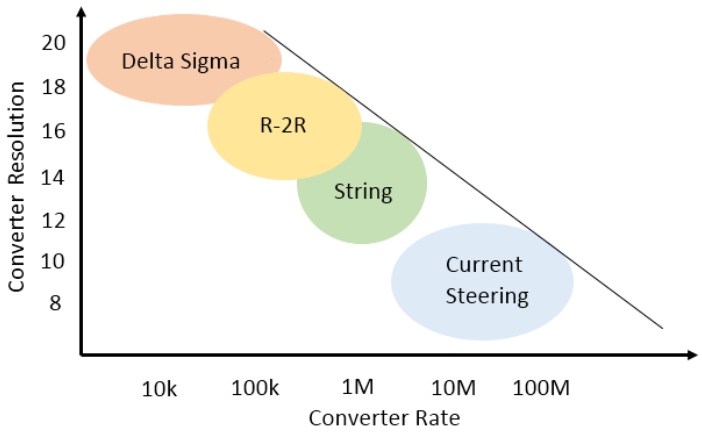


Figure 5.1. Resolution vs. conversion rate trade-off graph for DACs.

Current steering DAC basically consists of current replicas, electronic switches and output resistors as shown in Figure 5.2. Current replication networks generate the weighted currents, current switching network is controlled by the binary bits and resistor is used for the current to voltage conversion. The switch network combines the corresponding current in the output node and creates output value and this process is repeated for each new bit set.

A possible implementation of the current steering switches and current sources is shown on the right side of Figure 5.2. Current source for the LSB bit is implemented with MOS current source in cascode configuration. These transistors are biased with constant voltages. The other bits are implemented by scaling the LSB transistors

according to their bit weights. Also the switches are sized up according to bit weights. The switches are made by differential pair configurations. For high resolutions, sizing up the weighted bits may increase the mismatch pressure over the overall design. In practice, the concept of partitioning is applied to the weighted sources and each weighted current source is made of parallel connected LSB current sources.

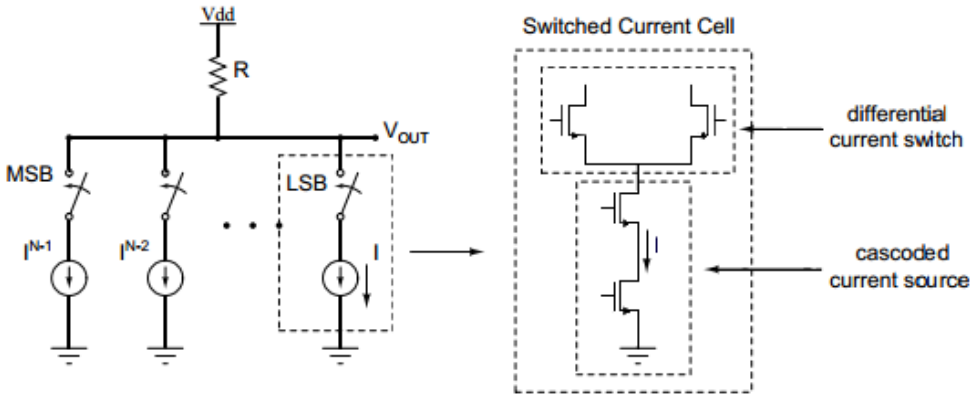


Figure 5.2. Current steering DAC architecture.

This very simple and compact architecture may obtain very high conversion rates with low power consumption. However, these advantages come at the cost of two severe problems. First, matching requirements for good accuracy are limited due to weighting the bits. The MSB current is 2^{N-1} times larger than the LSB current and must be matched to the LSB current with one LSB accuracy (DNL$<$ 1 LSB to guarantee monotonicity) which is very difficult to obtain. The other problem arises from the weighted impact of switching problems. One of these types of problems is MSB/LSB glitches that occur when the data waveforms, which control the current switches are imperfectly synchronized. The binary data waveforms arrive at the switches, with different timings. Consider as an example a 6 bit binary weighted DAC. At the midscale transition from 011111 to 100000, the MSB current source turns on and others turn off. If the MSB turns on slightly earlier than the others, then the code 111111 will appear for a short time before the 100000. This glitch creates harmonic distortion in the operation of the DAC. This is one of the major problems for years in the current steering DAC architecture [49].

5.1. DAC Design

In this thesis, an 8-bit PMOS current source based 4/4 thermometer/binary segmented SC DAC architecture is designed and simulated. The design also has a full self calibration technique to assure linearity. The overall design is shown in Figure 5.3.

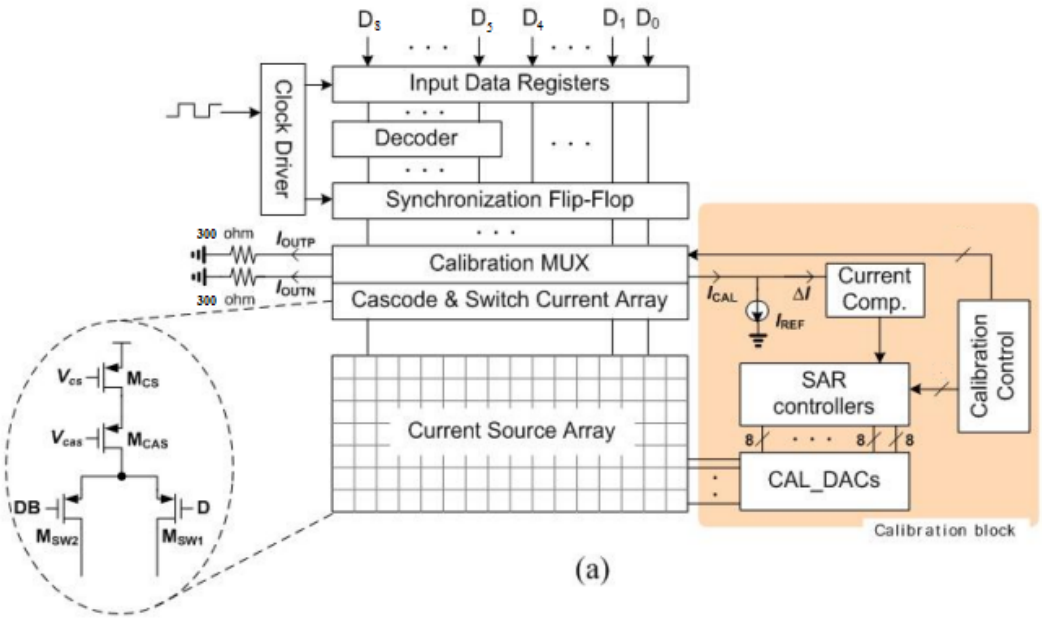


Figure 5.3. Overall current steering DAC architecture [50].

5.1.1. Partitioning and segmentation

To relax the matching requirements over the current sources, partitioning and segmentation is proposed in literature [51]. However, its impact is not very effective. Typical options is a thermometer code (some binary bits and some thermometer bits). In this case, MSB current cells are identical and relaxes the matching requirements of MSB and LSB sources. Also binary weighted switching problems are eliminated and monotonicity is guaranteed. However, using thermometer decoding for all of the bits is impractical since large number of current cells are required, which makes it hard to decode and interconnect to each other. The best option is segmentation of bits as a coarse thermometer part and a fine binary part. In this thesis, 4/4 segmentation scheme is realized. This means that 2^4-1 = 15 thermometer coded MSB current cells

Table 5.1. Binary to Thermometer Decoder.

b3	b2	b1	b0	s15	s14	s13	s12	s11	s10	s9	s8	s7	s6	s5	s4	s3	s2	s1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

and 4 LSB current cells are used. Thermometer decoding is given in Table 5.1.

5.1.2. Current switching network and current sources

A current switch and a current source form together a switched current (SI) cell. A general topology for an SI cell is shown in Figure 5.2 and it consists of a cascoded current source and a differential current switch. The input is a differential voltage signal and output is a differential current for a current cell. Generally, all of the cells use the same bias voltages. Current is driven to one or the other branch of the cell according to input bit values, thus current is steered all the time [49].

As previously stated, timing errors lead to glitches at the output of the DAC. Another source of glitches is charge feed-through phenomenon on switch common source node. This phenomenon occurs during the switching with the charge flow between the parasitic capacitances of the switches. Charge flow amount is proportional

to the channel and overlap capacitances of the switch transistors. The capacitances and the voltage swing in the inputs of the switches determine the charge deposited in the output node. Considering that switch transistors are not necessarily matched at all, keeping them minimum size will help the reduction of this charge flow and glitches at the output node.

Another problem is that output resistance of the DAC is a function of the number of switched-on cells, therefore it is modulated by the input signal. This will lead to changes in output resistance and hence will cause errors. A solution is increasing the output resistances of each current source. Thus, overall resistance will be less affected from the number of switched-on cells. To achieve this, cascode transistors are used in current sources. Although this attempt decreases the voltage head room at the output, a careful design will prevent additional errors.

Using thermometer code scheme will relax the matching properties of the current source cells up to some point. However, this will not help all the way to achieve very high accuracies. Careful sizing and biasing will do the rest. Using the INL-yield model proposed in [52] for a yield of 99.8% a $\sigma(\mathrm{I})/\mathrm{I} = 0.25\%$ is required for the LSB current. The area per LSB current source required to achieve this matching precision is calculated as defined in [52]:

$$WL = \frac{A_\beta^2 + 4A_{Vt}^2/(V_{gs} - V_t)^2}{2(\frac{\sigma(I)}{I})^2} \tag{5.1}$$

At this point, area vs voltage trade-off appears. Large overdrive reduces random amplitude errors and the area, but reduces the voltage room left for the rest of the current cell. Besides, in TSMC 90nm technology, it is very difficult to spare a large overdrive voltage. However, area increase will lead to increase in output capacitances and will affect the high frequency performance. As a result, considering the TSMC 90nm process, 200mV overdrive voltage is chosen and mismatch parameters are taken from [53]. Obtained area is scaled up according to the series 1,2,4,8,16 according to

bit weights.

5.1.3. Error correction

In literature, many calibration techniques have been published in order to improve the linearity of DAC. These approaches tried not to use large current sources, but instead intrinsic accuracy theorems [52], self calibration techniques [54] and dynamic averaging [55]. The approach used in [50] is best suited for the DAC presented in this thesis.

The calibration scheme consists of current comparators, calibration control logics, SAR controllers and calibration DACs, illustrated in Figure 5.3. All of the current cells are calibrated sequentially. A current cell chosen by calibration control logics is carried to the current comparator, SAR controller and the calibration DAC for current calibration. If there is a difference between reference current and the output current, then 8-bit SAR controller will try to find a solution for calibration DAC. Then, the calibrating DAC will generate a tuning bias voltage which is connected to the body terminals of the current source. When the calibration is converged, the required tuning bias voltage will be kept constant for each current cell.

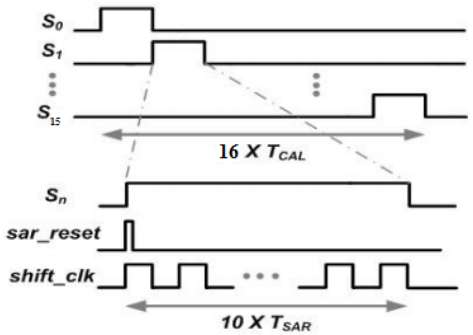


Figure 5.4. The timing diagram of the calibration scheme.

The timing diagram of the calibration scheme is shown in Figure 5.4. The calibration controller has 16 digital output bits. In each cycle, one output is set to 1 and the other set to 0 by a shift register. The SAR controllers [56] will be enabled accord-

ing to output of the calibration controller. For every calibration control clock, there exist 10 SAR controller clocks. The first SAR clock cycle will reset all the outputs of the SAR to 0. Then, the SAR controller searches the optimum voltage through a successive approximation approach by changing the bits of the calibration DAC. At first, the MSB bit of the SAR is set to 1, and calibration DAC produces an analog output that adjusts the body voltage of the current source. Then, the produced current is checked with the reference current at the current comparator [57] and results are fedback to the SAR controller. The same process continues until all the current sources are calibrated to reference current. After calibration, each calibration DAC's output is locked and desired currents are obtained from current sources.

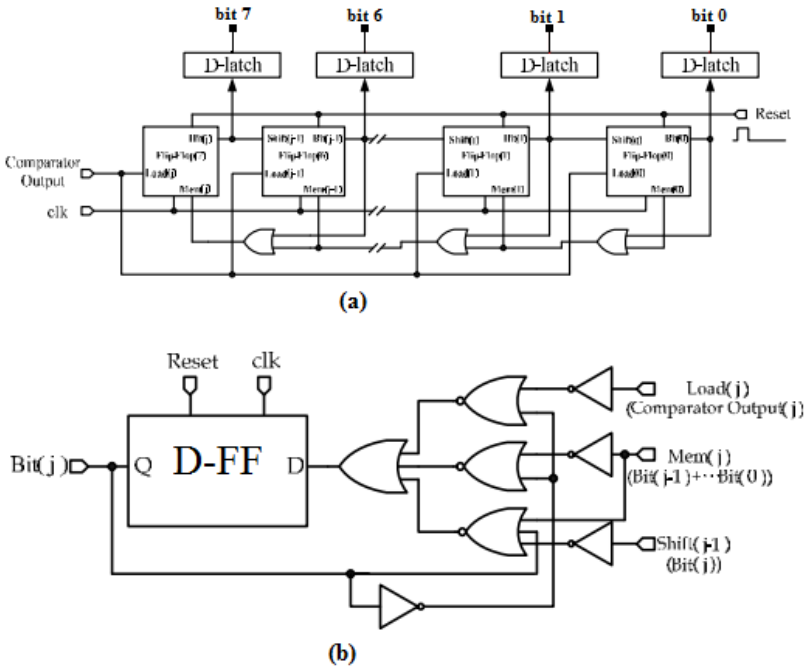


Figure 5.5. (a)8-bit SAR controller (b)Block diagram of the N_{th} control unit.

An 8-bit binary SI calibration DAC is shown in Figure 5.6. It consists of a load resistance and 8 cascode current sources. The output of the calibration DAC is connected to the body terminal of the corresponding PMOS current source. The output voltage is controlled by the binary weighted currents. These currents are controlled by the SAR output codes. Hence, a controllable body-source voltage is obtained for the main DAC architecture.

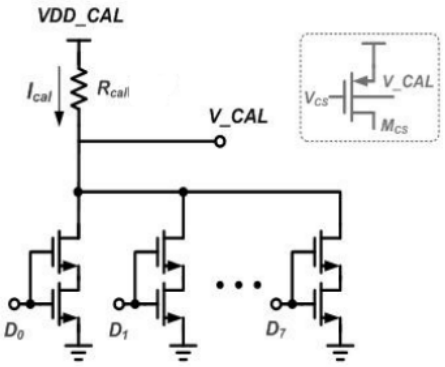


Figure 5.6. An 8-bit binary switch current calibration DAC.

5.2. DAC Simulation Results

The performance of the DAC is evaluated in this section. Effective number of bits, maximum obtainable sample rate and DNL and INL of DAC will be presented. Linearity of the DAC is very important and it is a figure of merit. DNL and INL of the DAC before and after the calibration are given in Figure 5.7 and 5.8, respectively. INL has improved from 3.6 to 0.4 LSB. DNL of the uncalibrated circuit is calculated as less than 0.06 LSB. DNL of the calibrated circuit is calculated as less than 0.16 LSB. Since the area of the current sources are adjusted well, DNL results of the DAC before and after calibration are very good as expected. A ramp output, as seen in Figure 5.9, is observed and data is collected to calculate DNL and INL.

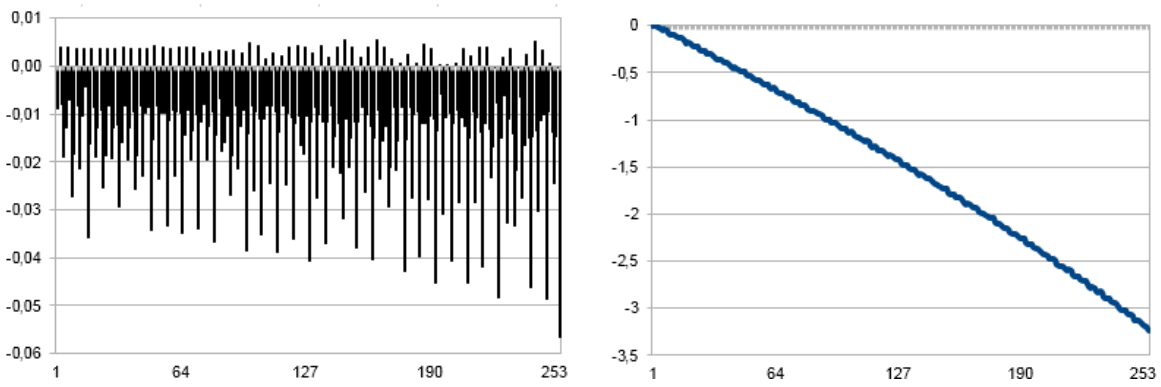


Figure 5.7. Before calibration, (a)DNL and (b)INL of the DAC.

The proposed DAC can work up to 1GSample/s. Transient simulation result for

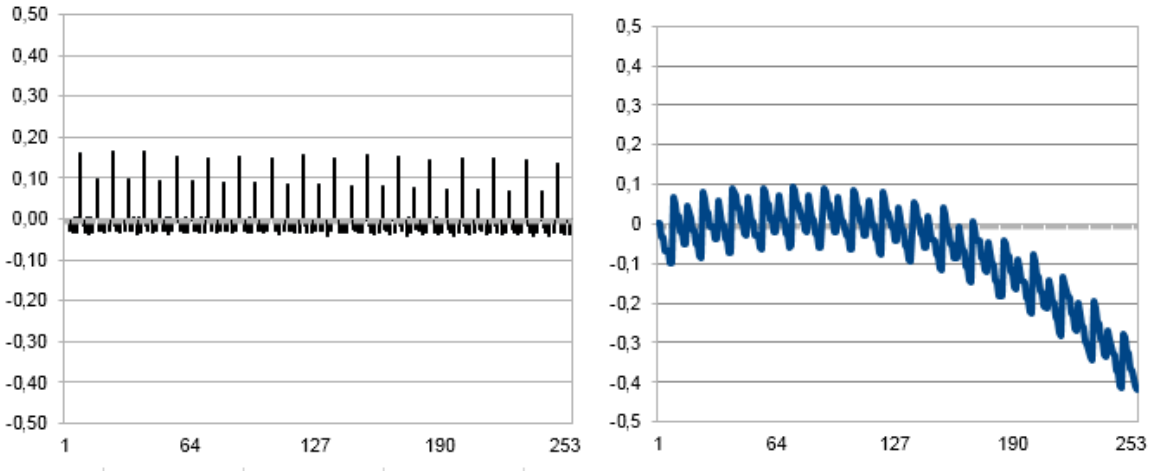


Figure 5.8. After calibration, (a)DNL and (b)INL of the DAC.

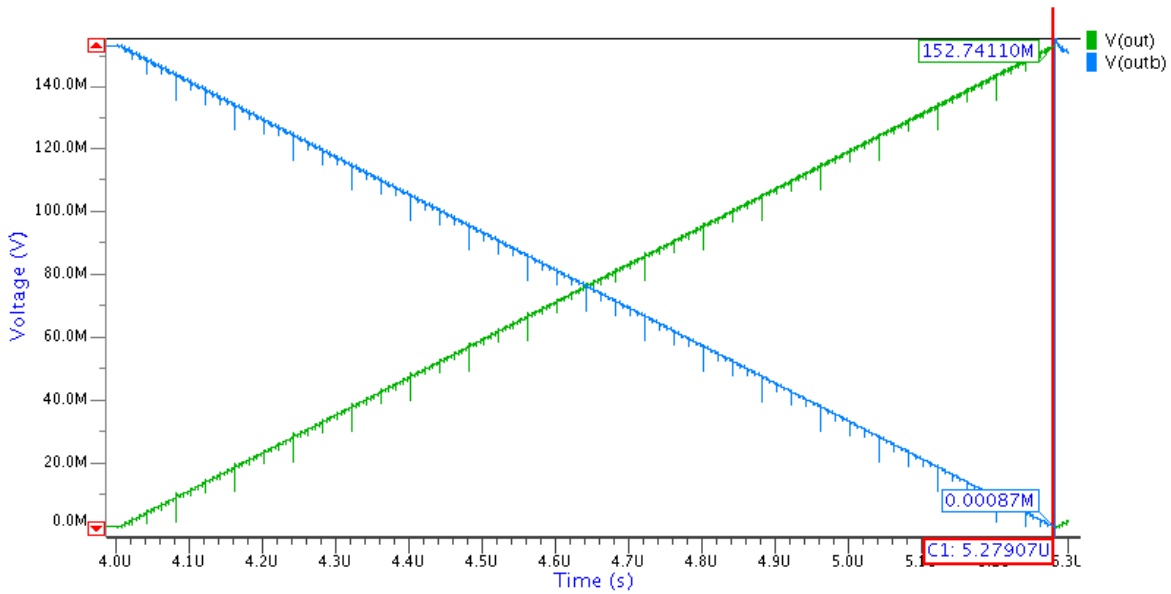


Figure 5.9. Ramp outputs of the DAC for DNL and INL calculation.

1GSample/s and 500MSample/s are shown in Figure 5.10 and 5.11, respectively.

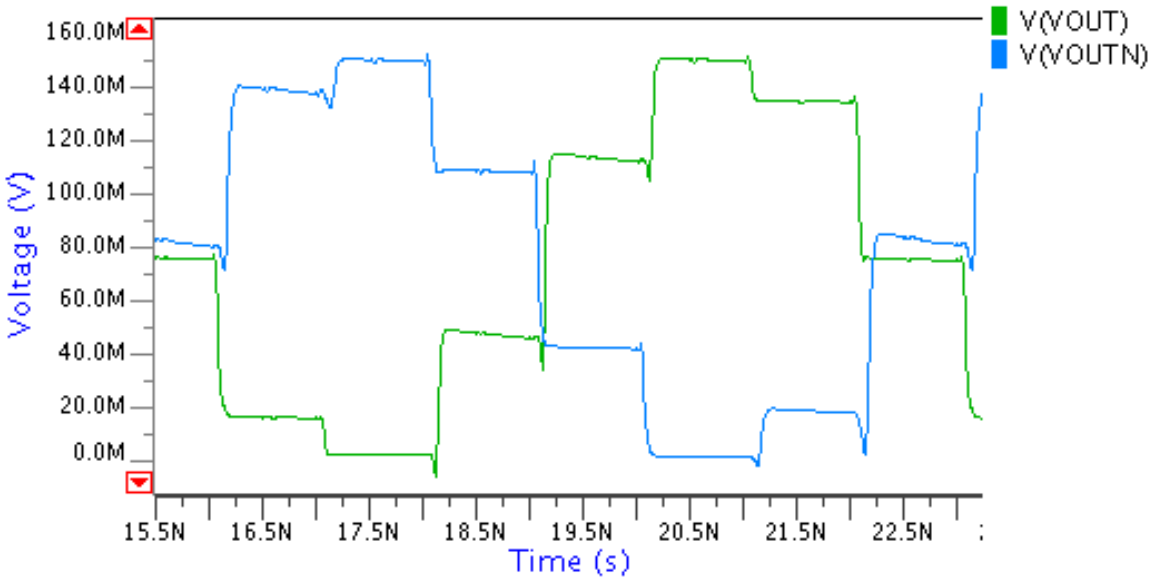


Figure 5.10. Transient simulation result for 1GSample/s.

Another performance criteria is ENOB which is commonly used to evaluate the performance of the DAC. ENOB is calculated as:

$$ENOB = \frac{SNR - 1.76}{6.02} \tag{5.2}$$

where SNR is given in dB and constant 1.76 comes from the assumption of sinusoidal input signal.

SNR is calculated through FFT that is taken by MATLAB. A sinusoidal input is used with a frequency that is not a submultiple of sampling frequency, hence same samples are not used. Number of samples taken within one period of input signal is chosen around seven. 1024 samples are used to take the FFT. ENOB is calculated as 7.882 bits with 49.206 dB SNR value for the calibrated DAC and 7.673 bits with 47.950 dB SNR value for the non-calibrated DAC for 200MHz clock signal. It is calculated 7.076 bits with 44.357 dB for 500MHz clock signal. SNR result for the 200MHz clock is given in Figure 5.12.

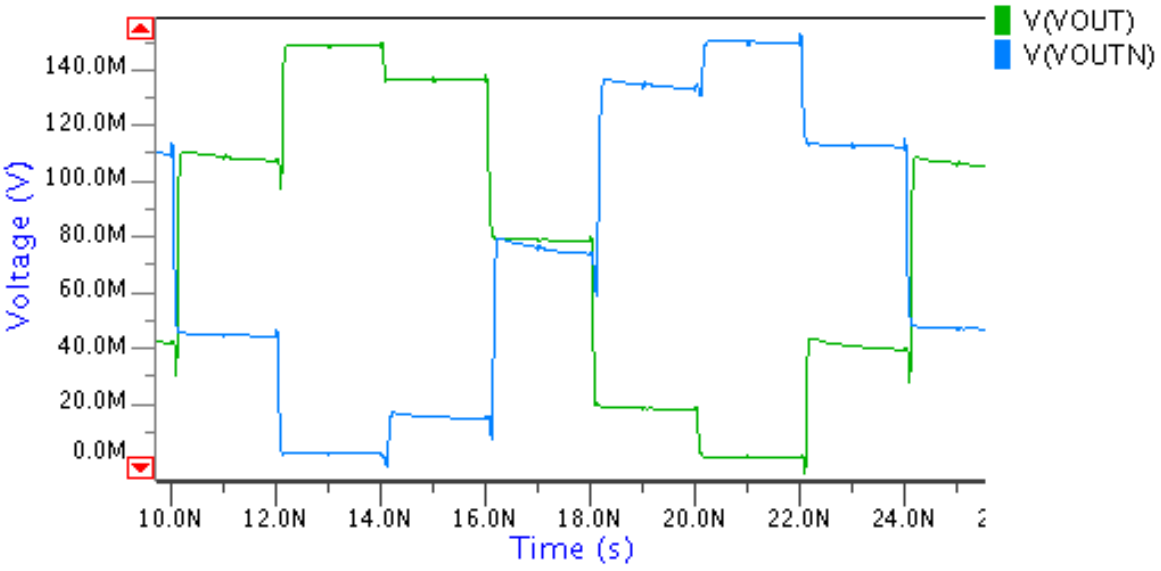


Figure 5.11. Transient simulation result for 500MSample/s.

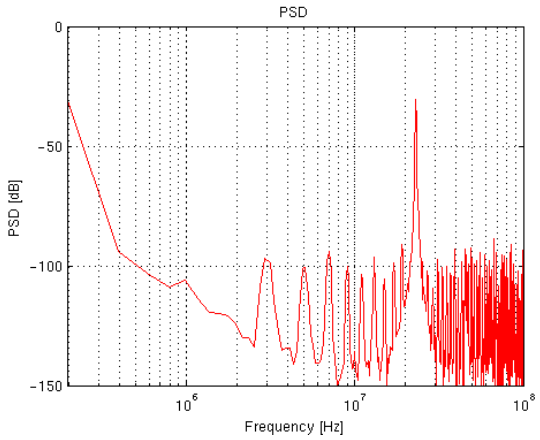


Figure 5.12. SNR result for the 200MHz clock.

The effect of calibration circuit on the currents of the DAC is shown in Figure 5.13. Currents are very close to the ideal values after the calibration is completed.

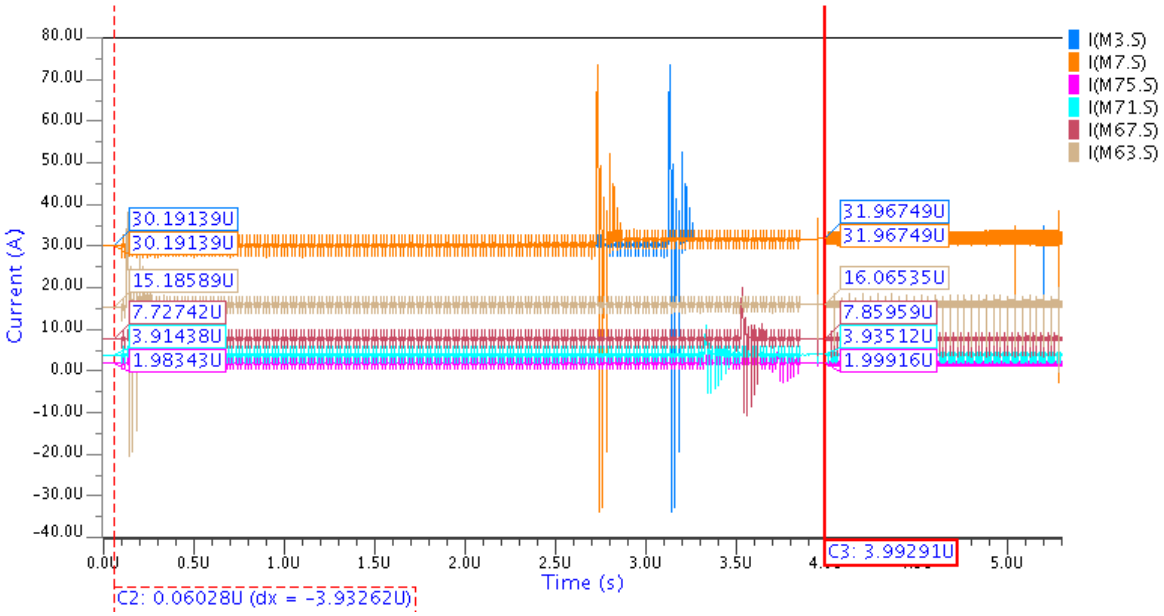


Figure 5.13. The effect of calibration circuit on the currents of the DAC.

DAC is designed to operate between 0V and 153mV. This voltage range is not enough for most of the applications. To extend this range, multiples of full scale current can be added to the output load resistors. Figure 5.14, shows up to 7 times of the multiple of full scale current can be added. But satisfied linearity is obtained only when up to 3 times of the full scale current is added.

Also, Monte Carlo analysis is done for the calibrated DAC. Results show that 80% of the outputs are between the 0.5 LSB INL range. It is demonstrated in Figure 5.15.

The overall DAC design consumes 2.5 mW. The calibration circuitry of the DAC consumes the important part of the power which is around 1.9mW, and the remaining power is consumed in the current cells.

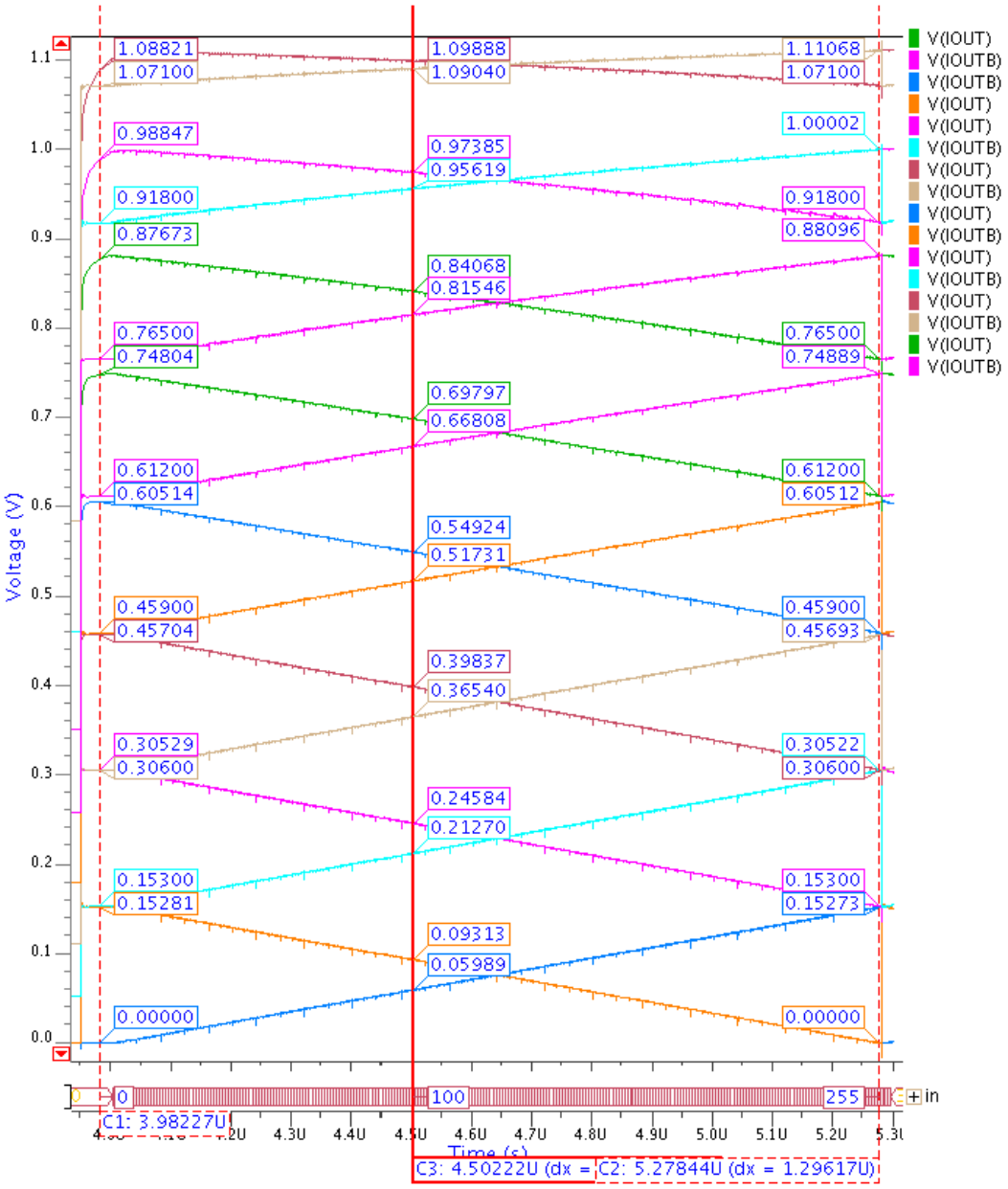


Figure 5.14. Different voltage levels obtained by additional full scale currents.

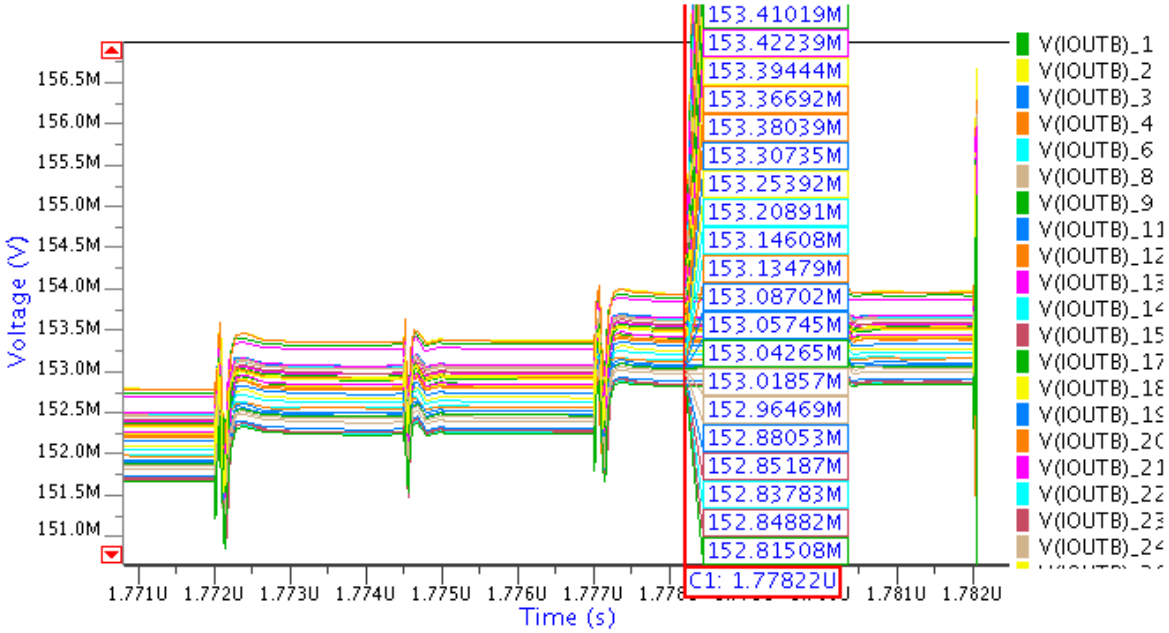


Figure 5.15. Results of the Monte Carlo analysis for calibrated DAC.

6. CONCLUSION AND FUTURE WORK

In this thesis, some issues for FPMA design in submicron technologies are introduced. Possible problems in FPMA design due to submicron technologies are explained and some solutions in the literature are demonstrated. The evolution of programmable analog and digital arrays are explained in detail. Since the FPMA will consist of digital and analog parts, different architectures and their complete properties are introduced.

The communication of the digital and analog parts within the FPMA is realized by DAC and ADC circuits. The designs in literature are introduced and, the architectures that will be used in the FPMA design are chosen. These architectures, which are best suited for the 90nm technology, are SAR ADC and current steering DAC. The SAR ADC architecture that will be used in the FPMA design is introduced. Also a VCO circuit which is one of the most used circuits in electronic designs is designed and simulated.

The main work in this thesis is the design of the current steering DAC and its digital error correction. A PMOS 8-bit 1GSample/s current steering based DAC is designed and simulated. The design works between 0V and 153mV with the accuracy of 0.4 LSB INL and 0.16 LSB DNL. The digital error correction scheme is a self calibration technique and improves the INL from 3.5 LSB to 0.4 LSB. Also FFT tests prove that digital error correction scheme has a great impact on the circuit. The effective number of the bits of the circuit is improved to 7.881 bits from 7.673 bits. SNR is calculated as 49.206 dB for a 23 MHz sinusoidal input and 200MHz clock. The overall DAC design consumes 2.5mW. The calibration part of the DAC consumes an important part of the power, which is around 1.9mW, the remaining part is consumed in the current cells.

In Figure 5.14, it is shown that DAC can operate up to 0.6V without significant errors by adding full scale current to output branches. Above 0.6V, the error is hard to tolerate and as a future work an NMOS counterpart of the DAC is planned to design.

REFERENCES

1. Lee, E. K. and P. G. Gulak, "A CMOS Field-Programmable Analog Array", *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 12, pp. 1860–1867, 1991.

2. Lee, E. K. and P. G. Gulak, "A Transconductor-Based Field-Programmable Analog Array", *Solid-State Circuits Conference, 1995. Digest of Technical Papers. 42nd ISSCC, 1995 IEEE International*, pp. 198–199, IEEE, 1995.

3. Madian, A. H., S. A. Mahmoud and A. M. Soliman, "Field Programmable Analog Array Based on CMOS CFOA and Its Application", *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, pp. 1042–1046, IEEE, 2008.

4. Rose, J., A. El Gamal and A. Sangiovanni-Vincentelli, "Architecture of Field-Programmable Gate Arrays", *Proceedings of the IEEE*, Vol. 81, No. 7, pp. 1013–1029, 1993.

5. Carter, W., K. Duong, R. H. Freeman, H. Hsieh, J. Y. Ja, J. E. Mahoney, L. T. Ngo and S. L. Sze, "A User Programmable Reconfigurable Gate Array", *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 233–235, 1986.

6. Stone, H. S., "Parallel Processing with the Perfect Shuffle", *IEEE Transactions on Computers*, Vol. 100, No. 2, pp. 153–161, 1971.

7. D'Mello, D. R. and P. G. Gulak, "Design Approaches to Field-Programmable Analog Integrated Circuits", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 7–34, 1998.

8. Lee, E. and P. Gulak, "Field Programmable Analogue Array Based on MOSFET Transconductors", *Electronics Letters*, Vol. 28, No. 1, pp. 28–29, 1992.

9. Chow, P., P. G. Gulak and P. Chow, "A Field-Programmable Mixed-Analog-Digital Array", *Field-Programmable Gate Arrays, 1995. FPGA'95. Proceedings of the Third International ACM Symposium on*, pp. 104–109, IEEE, 1995.

10. Moore, G. E. *et al.*, "Cramming More Components Onto Integrated Circuits", *Proceedings of the IEEE*, Vol. 86, No. 1, pp. 82–85, 1998.

11. Taur, Y., "CMOS Design Near the Limit of Scaling", *IBM Journal of Research and Development*, Vol. 46, No. 2.3, pp. 213–222, 2002.

12. Buss, D. D., "Technology in the Internet Age", *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, Vol. 1, pp. 18–21, IEEE, 2002.

13. Annema, A., B. Nautal, R. van Langevelde and H. Tuinhout, "Designing Outside Rail Constraints", *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, pp. 134–135, IEEE, 2004.

14. Annema, A., B. Nautal, R. van Langevelde and H. Tuinhout, "Analog Circuits in Ultra-Deep-Submicron CMOS", *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 1, pp. 132–143, 2005.

15. Abbas, M., K.-T. Cheng, Y. Furukawa, S. Komatsu and K. Asada, "Signature-Based Testing for Digitally-Assisted Adaptive Equalizers in High-Speed Serial Links", *Test Symposium, 2009 14th IEEE European*, pp. 107–112, IEEE, 2009.

16. Kuon, I., R. Tessier and J. Rose, *FPGA Architecture: Survey and Challenges*, Foundations and Trends in Electronic Design Automation, Now Publishers Inc., 2008.

17. El Gamal, A., J. Greene, J. Reyneri, E. Rogoyski, K. A. El-Ayat and A. Mohsen, "An Architecture for Electrically Configurable Gate Arrays", *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 2, pp. 394–398, 1989.

18. Ahrens, M., A. El Gamal, D. Galbraith, J. Greene, S. Kaptanoglu, K. Dharmarajan, L. Hutchings, S. Ku, P. McGibney, J. McGowan *et al.*, "An FPGA Family Optimized for High Densities and Reduced Routing Delay", *Custom Integrated Circuits Conference, 1990., Proceedings of the IEEE 1990*, pp. 31–5, IEEE, 1990.

19. Birkner, J. a., A. Chan, H. Chua, A. Chao, K. Gordon, B. Kleinman, P. Kolze and R. Wong, "A Very-High-Speed Field-Programmable Gate Array Using Metal-to-Metal Antifuse Programmable Elements", *Microelectronics Journal*, Vol. 23, No. 7, pp. 561–568, 1992.

20. Hsieh, H.-C., K. Dong, J. Ja, R. Kanazawa, L. Ngo, L. Tinkey, W. Carter and R. Freeman, "A 9000-Gate User-Programmable Gate Array", *Custom Integrated Circuits Conference, 1988., Proceedings of the IEEE 1988*, pp. 15.3/1–15.3/7, IEEE, 1988.

21. Hsieh, H.-C., W. S. Carter, J. Ja, E. Cheung, S. Schreifels, C. Erickson, P. Freidin, L. Tinkey and R. Kanazawa, "Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays", *Custom Integrated Circuits Conference, 1990., Proceedings of the IEEE 1990*, pp. 31.2/1–31.2/7, IEEE, 1990.

22. Rose, J., R. J. Francis, D. Lewis and P. Chow, "Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Effciency", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 5, pp. 1217–1225, 1990.

23. Pierzchala, E. and M. A. Perkowski, "A High-Frequency Field-Programmable Analog Array (FPAA) Part 1: Design", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 143–156, 1998.

24. Kutuk, H. and S.-M. Kang, "A Field-Programmable Analog Array (FPAA) Using Switched-Capacitor Techniques", *Circuits and Systems, 1996. ISCAS'96., Connecting the World., 1996 IEEE International Symposium on*, Vol. 4, pp. 41–44, IEEE, 1996.

25. Premont, C., R. Grisel, N. Abouchi and J.-P. Chante, "A Current Conveyor Based Field Programmable Analog Array", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 105–124, 1998.

26. Bratt, A. and I. Macbeth, "Design and Implementation of a Field Programmable Analogue Array", *Proceedings of the 1996 ACM Fourth International Symposium on Field-Programmable Gate Arrays*, pp. 88–93, ACM, 1996.

27. Vallancourt, D. and Y. P. Tsividis, "Timing-Controlled Fully Programmable Analogue Signal Processors Using Switched Continuous-Time Filters", *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 8, pp. 947–954, 1988.

28. Klein, H., "The EPAC Architecture: An Expert Cell Approach to Field Programmable Analog Circuits", *Circuits and Systems, 1996., IEEE 39th Midwest Symposium on*, Vol. 1, pp. 169–172, IEEE, 1996.

29. Embabi, S., X. Quan, N. Oki, A. Manjrekar and E. Sanchez-Sinencio, "A Field Programmable Analog Signal Processing Array", *Circuits and Systems, 1996., IEEE 39th Midwest symposium on*, Vol. 1, pp. 151–154, IEEE, 1996.

30. Pierzchala, E., M. A. Perkowski, P. Van Halen and R. Schaumann, "Current-Mode Amplifier/Integrator for a Field-Programmable Analog Array", *Solid-State Circuits Conference, 1995. Digest of Technical Papers. 42nd ISSCC, 1995 IEEE International*, pp. 196–197, IEEE, 1995.

31. Zhang, C., A. Bratt and I. Macbeth, "A New Field Programmable Mixed-Signal Array and Its Application", *4th Canadian Workshop on Field-Programmable Devices*, pp. 13–14, 1996.

32. Lee, E. K. and W. L. Hui, "A Novel Switched-Capacitor Based Field-Programmable Analog Array Architecture", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 35–50, 1998.

33. Anderson, D., C. Marcjan, D. Bersch, H. Anderson, P. Hu, O. Palusinski, D. Gettman, I. Macbeth and A. Bratt, "A Field Programmable Analog Array and Its Application", *Custom Integrated Circuits Conference, 1997., Proceedings of the IEEE 1997*, pp. 555–558, IEEE, 1997.

34. Basu, A., S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg and P. Hasler, "A Floating-Gate-Based Field Programmable Analog Array", *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 9, pp. 1781–1794, 2010.

35. Becker, J., F. Henrici, S. Trendelenburg, M. Ortmanns and Y. Manoli, "A Field-Programmable Analog Array of 55 Digitally Tunable OTAs in a Hexagonal Lattice", *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 12, pp. 2759–2768, 2008.

36. Quan, X., S. Embabi and E. Sanchez-Sinencio, "A Current-Mode Based Field Programmable Analog Array Architecture for Signal Processing Applications", *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, pp. 277–280, IEEE, 1998.

37. Abramson, D. N., J. D. Gray, S. Subramanian and P. Hasler, "A Field-Programmable Analog Array Using Translinear Elements", *System-on-Chip for Real-Time Applications, 2005. Proceedings. Fifth International Workshop on*, pp. 425–428, IEEE, 2005.

38. Kub, F. J., K. K. Moon, I. A. Mack and F. M. Long, "Programmable Analog Vector-Matrix Multipliers", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 1, pp. 207–214, 1990.

39. Pierzchala, E., M. A. Perkowski and S. Grygiel, "A Field Programmable Analog Array for Continuous, Fuzzy, and Multi-Valued Logic Applications", *Multiple-Valued Logic, 1994. Proceedings., Twenty-Fourth International Symposium on*, pp. 148–155, IEEE, 1994.

40. Pierzchala, E. and M. A. Perkowski, "A High-Frequency Field-Programmable Analog Array (FPAA) Part 2: Applications", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 157–169, 1998.

41. Pankiewicz, B., M. Wojcikowski, S. Szczepanski and Y. Sun, "A Field Programmable Analog Array for CMOS Continuous-Time OTA-C Filter Applications", *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 2, pp. 125–136, 2002.

42. Kutuk, H. and S.-M. S. Kang, "A Switched Capacitor Approach to Field-Programmable Analog Array (FPAA) Design", *Analog Integrated Circuits and Signal Processing*, Vol. 17, No. 1-2, pp. 51–65, 1998.

43. Kinget, P. R., "Device Mismatch and Tradeoffs in the Design of Analog Circuits", *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 6, pp. 1212–1224, 2005.

44. Lai, C.-M., M.-H. Shen, Y.-D. Wu, K.-H. Huang and P.-C. Huang, "A 0.24 to 2.4 GHz Phase-Locked Loop with Low Supply Sensitivity in 0.18-μm CMOS", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 981–984, IEEE, 2011.

45. Esen, V. B., *10-Bit 60 MS/s Two-Step Flash ADC Design*, Master's Thesis, Bogazici University, 2013.

46. Chiu, Y., *Data Converter Basics*, 2012, `http://www.utdallas.edu/ỹxc101000/courses/7327/handout.html`, Accessed at May 2013.

47. Zhu, Y., C.-H. Chan, U.-F. Chio, S.-W. Sin, U. Seng-Pan, R. P. Martins and F. Maloberti, "A 10-bit 100-MS/s Reference-Free SAR ADC in 90 nm CMOS", *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 6, pp. 1111–1121, 2010.

48. Gordon, B., "Linear Electronic Analog/Digital Conversion Architectures, Their Origins, Parameters, Limitations, and Applications", *IEEE Transactions on Circuits and Systems*, Vol. 25, No. 7, pp. 391–418, 1978.

49. Doris, K., *High-speed D/A Converters: from Analysis and Synthesis Concepts to IC Implementation*, Ph.D. Thesis, Eindhoven University of Technology, 2004.

50. Chi, J.-H., S.-H. Chu and T.-H. Tsai, "A 1.8-V 12-bit 250-MS/s 25-mW Self-Calibrated DAC", *ESSCIRC, 2010 Proceedings of the*, pp. 222–225, IEEE, 2010.

51. Schoeff, J. A., "An Inherently Monotonic 12 bit DAC", *IEEE Journal of Solid-State Circuits*, Vol. 14, No. 6, pp. 904–911, 1979.

52. Bastos, J., A. M. Marques, M. S. Steyaert and W. Sansen, "A 12-bit Intrinsic Accuracy High-Speed CMOS DAC", *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 12, pp. 1959–1969, 1998.

53. Croon, J. A., W. M. Sansen and H. E. Maes, *Matching Properties of Deep SubMicron MOS Transistors*, Vol. 851, Springer, 2005.

54. Cong, Y. and R. L. Geiger, "A 1.5-V 14-bit 100-MS/s Self-Calibrated DAC", *IEEE Journal of Solid-State Circuits*, Vol. 38, No. 12, pp. 2051–2060, 2003.

55. Chen, T. and G. G. Gielen, "A 14-bit 200-MHz Current-Steering DAC with Switching-Sequence Post-Adjustment Calibration", *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 11, pp. 2386–2394, 2007.

56. Lee, S.-Y., C.-J. Cheng, C.-P. Wang and S.-C. Lee, "A 1-V 8-bit 0.95 mW Successive Approximation ADC for Biosignal Acquisition Systems", *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 649–652, IEEE, 2009.

57. Traff, H., "Novel Approach to High Speed CMOS Current Comparators", *Electronics Letters*, Vol. 28, No. 3, pp. 310–312, 1992.