A GENERAL DESIGN METHODOLOGY FOR EMBEDDED HIGH SPEED A/D CONVERTERS

by

Selçuk Talay

B.S. in Electronics and Telecom. Eng., İstanbul Technical University, 1998

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

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

112100

Boğaziçi University

2001

A GENERAL DESIGN METHODOLOGY FOR EMBEDDED HIGH SPEED A/D CONVERTERS

APPROVED BY:

Assoc. Prof. Günhan Dündar
(Thesis Supervisor)

Assoc. Prof. Levent Akın

Prof. Ömer Cerid

DATE OF APPROVAL 13/09/2001

...to my dearest Defne

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Assoc. Prof. Günhan Dündar for his guidance, motivation, patience and continuous support.

I would also like to thank to Prof. Ömer Cerid for his kind interest and his valuable ideas through out the development of this thesis.

I would also like to thank Assoc. Prof. Levent Akın for his kind interest and being a member of my thesis committee.

I am indebted to BETA members for their continues support and for the humor we had. Also special thanks to Faik Başkaya for the motivation and for the hard times we shared.

I would like to thank to Dr. Nihal Sarıer for having inspired me the idea of attempting graduate education.

Heartfelt thanks to Defne Kayrak for her motivation in the toughest times of my study and for her support, which makes the problems easier for me. I would also like to thank her again for just being herself.

Finally I would like to thank to my family, to whom I owe my success.

ABSTRACT

A GENERAL DESIGN METHODOLOGY FOR EMBEDDED HIGH SPEED A/D CONVERTERS

There are various types of analog-to-digital converters available in the literature. The fast and efficient data converter design automation systems gaining more and more interest. Although there are tools available for the synthesis of specific ADC architectures, there's need for development of the methods for the systematic selection of the topology.

The first step in data converter design automation is the selection of the adequate architecture. This thesis proposes a methodology for the systematic selection of the topology. Different architectures have been modeled in this study. Using these models, the restrictions introduced have been calculated. These restrictions have been used for generating the specifications of the blocks of each topology. The optimization has been done for the specifications. Then, the library has been searched for an adequate block that satisfies the required specifications. If the methodology could not find a solution, the range of inputs have been swept. Then the area, the power and the speed performances have been calculated in order to find the optimum topology.

The methodology for optimum topology selection has been realized. The results of the methodology were tested with the previous work. The results were similar as expected but models need further improvement for higher accuracy. Also, new architecture models should be added in order to cover the solution space efficiently.

ÖZET

YÜKSEK HIZLI, GÖMÜLÜ ANALOG DİJİTAL ÇEVİRİCİLER İÇİN GENEL BİR TASARIM METODU

Günümüzde çok çeşitli türlerde analog-dijital çeviriciler bulunmaktadır. Bu çeviriciler için hızlı ve verimli tasarım otomasyon sistemleri artan bir ilgi çekmektedir. Bazı analog-dijital çevirici yapıları için sentez araçları mevcut olsa da, sistematik şekilde bu yapılardan uygununu seçmek için gerekli metotların geliştirilmesi gerekmektedir.

Veri çeviricileri otomasyonunda ilk adim uygun yapıyı seçmektir. Bu tezde, yapının sistematik bir şekilde seçilmesi için bir metodoloji önerilmektedir. Bu çalışmada değişik yapılar modellenmiştir. Bu modeller kullanılarak, yapının getirdiği kısıtlamalar belirlenmiştir. Bu kısıtlamalar da, her yapının içerdiği blokların özelliklerini bulmakta kullanılmıştır. Bu özelliklerin eniyilemesi de yapılmıştır. Bundan sonra, blok kütüphanesi, istenilen özellikleri sağlayan uygun bir blok için araştırılmıştır. Eğer metodoloji bir sonuç bulamazsa, bir sonuç bulunmak için girişteki değişkenler taranır. Sonra, yapının alan, güç ve hız performansları en iyi yapıyı bulmak için hesaplanmıştır.

En iyi yapıyı seçen metodoloji gerçeklenmiştir. Elde edilen sonuçlar daha önceki çalışmalarla karşılaştırılmıştır. Sonuçlar beklenildiği gibi benzer çıkmıştır. Fakat daha kesin sonuçlar için modellerin geliştirilmesi gerekmektedir. Ayrıca, yeni modeller de daha geniş çözüm uzayını kapsayabilmek için eklenmelidir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF TABLES x
LIST OF SYMBOLS/ABBREVIATIONS xi
1. INTRODUCTION 1
2. DESIGN AUTOMATION OF ANALOG SYSTEMS AND DATA CONVERTERS ... 3
2.1. Behavioral Design of ADC's 5
2.2. Previous Work 8
3. METHODOLGY 11
4. FLASH ADC 13
4.1. Power Estimation 14
4.2. Area Estimation 15
4.3. Error Sources 16
4.3. Speed Estimation 17
4.4. Methodology for Flash Converter 17
5. PIPELINE ADC 20
5.1. Error Sources 23
5.1.1. Sub-ADC (SADC) Errors 25
5.1.2. DAC Errors 25
5.1.3. SHA Errors 26
5.1.4. Optimum Stage Resolution 27
5.2. Error Sources in MDAC 28
5.2.1. Thermal Noise 29
5.2.2. Mismatch Error 30
5.2.3. Other Errors 31
5.3. Power Estimation 32
5.4. Area Estimation 33
5.5. Delay Estimation 34

5.6. Speed Estimation 34
5.7. Capacitor Sizing 34
5.8. Calibration and Correction 35
5.9. Methodology of Pipeline Converters 36
6. SIGMA-DELTA ADC 38
7. DESIGN EXAMPLES 41
8. CONCLUSION 48

LIST OF FIGURES

Figure 1.1. Block diagram of methodology..... 2

Figure 2.2. ADC decomposition example. 7

Figure 4.1. Flash ADC..... 14

Figure 5.1. General model of a pipeline ADC..... 21

Figure 5.2. The recycling two step converter [29]. 22

Figure 5.3. Pipeline architecture with MDAC..... 28

Figure 7.1. Optimum stage resolution 42

Figure 7.2. Relation between resistor power and the reference voltage..... 46

Figure 7.3. Relation between the total power and the supply current. 46

Figure 7.4. Relation between power and resolution. 47

LIST OF TABLES

Table 2.1. The previous works about ADC automation. 10

Table 7.1. Different configurations for 12 and 13 bits. 41

Table 7.2. The constraints for the resolutions. 43

Table 7.3. Effect of capacitor sizing on thermal noise. 44

Table 7.4. Comparison of architectures for eight bits. 45

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

LIST OF SYMBOLS/ABBREVIATIONS

$A_{cap.array}$	Area of capacitor array
A_{comp}	Total area of comparators
a_dff	Area of D flip-flop
A_{dig}	Total area of digital circuit
A_{FF}	Area of flip-flop
A_{flashi}	Area of sub-ADC at stage i
A_{MDACi}	Area of MDAC at stage i
$a_$min	Minimum standard cell area
$a_$max	Maximum standard cell area
A_o	Gain of the amplifier
a_opamp	Area of OPAMP
a_ota	area of OTA
A_{res}	Total area of resistors
A_{selfi}	Area of self calibration circuit at stage i
$A_{swithes}$	Area of switches
A_{synci}	Total area of latches at stage i
A_{tot}	Total area of ADC
BW	Bandwidth of the OPAMP
BW_{comp}	Bandwidth of the comparator
b_i	Digital word
C_{dig}	Total switching capacitance of digital circuit
C_f	Feedback capacitor
C_{MDACi}	Capacitance of MDAC at stage i
C_p	Parasitic capacitance
C_S	Total switching capacitance
C_{unit}	Unit capacitor
e_i	Noise of stage i
e_{in}	Input referred noise of stage i
e_{gain}	Gain error of noise of stage i
e_m	Mismatch error

e_{offi}	Nonlinearity error of noise of stage i
e_{org}	Quantization error
e_t	Thermal noise
e_0	Total quantization error of oversampled ADC
f	Frequency
f_o	oversampling frequency
f_s	sampling frequency
FS	Full scale
G	Stage gain
G_i	Total gain before noise of stage i
G_{Si}	Gain of stage i
I_{sup}	Supply current of comparator
I_{supa}	Supply current of amplifier
K	Boltzmann's constant
k	constant for switching probability
K_{ci}	Constant for parasitic capacitors
K_F	Constant for probability of switching
k_2	Constant for wiring
l	length
m	mismatch
M	Required accuracy of MDAC
n	Resolution
Ni	Resolution of stage i
n_gates	Number of gates
N_{MDACi}	MDAC noise of stage i
NS	Number of stages
NS_{cal}	Number of stages that need calibration
$N_{S/H}$	S/H noise of stage i
P_{comp}	Total power dissipation of comparators
P_{dig}	Total power dissipation of digital circuit
P_{flashi}	Power dissipated by sub-ADC at stage i
P_{FF}	Power of flip-flop
P_{MDACi}	Power dissipated by MDAC at stage i

P_{RAM}	Energy consumed per cycle in a flip-flop
P_{res}	Total power dissipation of resistors
P_{selfi}	Power dissipated by self calibration at stage i
P_{synci}	Power dissipated by latches at stage i
P_{tot}	Total power dissipated by ADC
R	Unit resistor
R_{EQ}	Pseudoresistance at one input of the amplifier
R_{ON}	On-resistance of the CMOS switches
R_S	Sheet resistance
SNR	Signal to noise ratio
T	Absolute temperature
TTN	Total thermal noise
V_{in}	Input voltage
V_{dd}	Supply voltage of digital circuit
vos	Offset voltage of the comparator
V_{RA}	Amplified residue
V_{ref}	Reference voltage
V_{sup}	Supply voltage of comparator
V_{supa}	Supply current of amplifier
w	width
uca	Unit cell area
x	Number of redundant bits
ε	Mismatch error of capacitors
Φ	Constant for differential or single mode operation
ADC	Analog-to-Digital Converter
CAD	Computer aided design
CPU	Central processing unit
DAC	Digital-to-analog converter
ENOB	Effective number of bits
IC	Integrated circuit
INL	Integral nonlinearity

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

LSB	Least significant bit
MDAC	Multiplying digital-to-analog converter
MSB	Most significant bit
OPAMP	Operational Amplifier
OTA	Operational Transconductance Amplifier
SADC	Sub analog-to-digital converter
S/H	Sample and hold
SHA	Sample and hold amplifier
VLSI	Very large scale integrated

1. INTRODUCTION

Analog-to-digital converters (ADC) provide the link between the analog world and the digital world of signal processing, computing, and other digital data collection or data processing systems. Electronic signals are processed increasingly in the digital world. Since, most of the signals are analog in the world we live, the importance of ADC's are increasing considerably.

For a long time, ADC's have been used widely in digital test equipment. Recently, the applications for ADC's have expanded widely as many electronic systems that used to be entirely analog have been implemented using digital electronics. Examples of such applications include digital telephone transmission, cordless phones and medical imaging. Furthermore, ADC's have found their way into systems that would normally be considered as being entirely digital as these digital systems are pushed to higher levels of performance. Data storage is one example of such a system. As storage density in disk drive systems is increased, the signals handled by the read circuitry have become increasingly analog in character. Presently, 6-bit ADC's are commonly used in the read circuits of disk drives [1].

An enormous number of specific ADC designs of various forms exist in the literature [1-5]. However, a general framework for the systematic selection of the appropriate structure needs to be developed. The selection of the topology should also be automated in order to cope with larger analog design automation systems. The details of design automation of the ADC's with the automation of topology selection and supplying required parameters to the lower synthesis level are given in the next sections.

The first goal of this dissertation was to develop a general design methodology for embedded high-speed ADC's. The necessary algorithms for each topology have been developed for this purpose. The algorithms utilize the inputs supplied by the user as well as the technology file, which can be constructed by the technology databook and the lookup tables of the available library elements (i.e. comparator, operational amplifier (OPAMP), digital decoder) created only once by the user. The general system requirements and

constraints such as the operating frequency range, speed, supply voltage, allowed chip area and power consumption, are used as inputs by the methodology developed. The algorithms developed select the appropriate topology for the converter (i.e. flash, pipelined, Sigma-Delta or oversampled ADC's [1-3]) so as to satisfy design specifications.

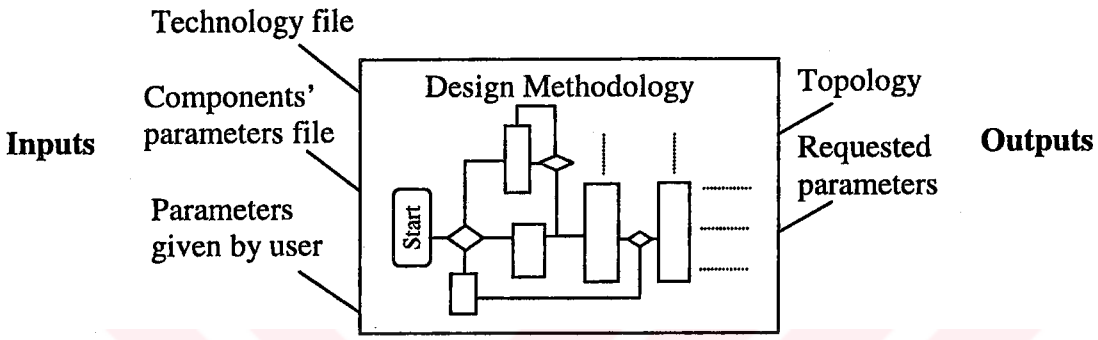

Figure 1.1. Block diagram of methodology

In the next section, the design automation of analog systems and data converters are presented. Also, the previous works are introduced. An overview of the developed methodology is given at section 3. The topologies available in the methodology are presented in sections 4 and 5. A design example is given at the section 6. The last section is the conclusion.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

2. DESIGN AUTOMATION OF ANALOG SYSTEMS AND DATA CONVERTERS

The lack of efficient automation tools for ADC's as well as the other building block of an analog system becomes a bottleneck in the very large scale integrated (VLSI) design. The increase in the need for the complex integrated circuits (IC) and decrease in the time to market time, places a demand for the automation and computer aided design of such systems.

In general, the designs of the complex analog signal processing function, the designs are decomposed into several parallel, manageable sub-design tasks by exploiting the hierarchical nature of analog circuits [6-7]. Unlike digital circuits, which have a well-defined hierarchical decomposition (e.g., the transistor level, the gate level, and the register transfer level), there are no precise standards for how analog systems can be decomposed into lower level blocks. In digital system design, these well-defined layers of hierarchy are very important. The design steps need only translate from one level to the next lower level, which is often a much simpler task than going from the specifications for the digital system directly to transistors.

If several decompositions can be found for a given analog system, and their quality can be compared for the entire sub-design tasks, proceeding without any information about the results of other sub-design tasks, then the hierarchical decomposition can be considered as adequate and all of the sub-design tasks can proceed in parallel. Unfortunately this can only be achieved rarely. Also selection of the appropriate architecture (also called topology) can be another problem.

Figure 2.1 shows a simple hierarchical decomposition of an analog system design process. System level synthesis is the root of the decomposition. The system specifications are passed to architectural level and the appropriate architecture, which should decrease the complexity of the lower level blocks and take more advantage of the parallel processing of them. The next level is mostly called circuit level synthesis. The blocks in this level are in more detail, which can supply more precise information about the circuit level parameters.

Also the optimization may be done at this level for subblocks in order to discard any mismatch with the specifications passed from the above level. The lowest level can be called as layout synthesis level, which mostly deals with the physical implementation considering the technology parameters. Each of these levels have been studied until present time in our research group [8-10].

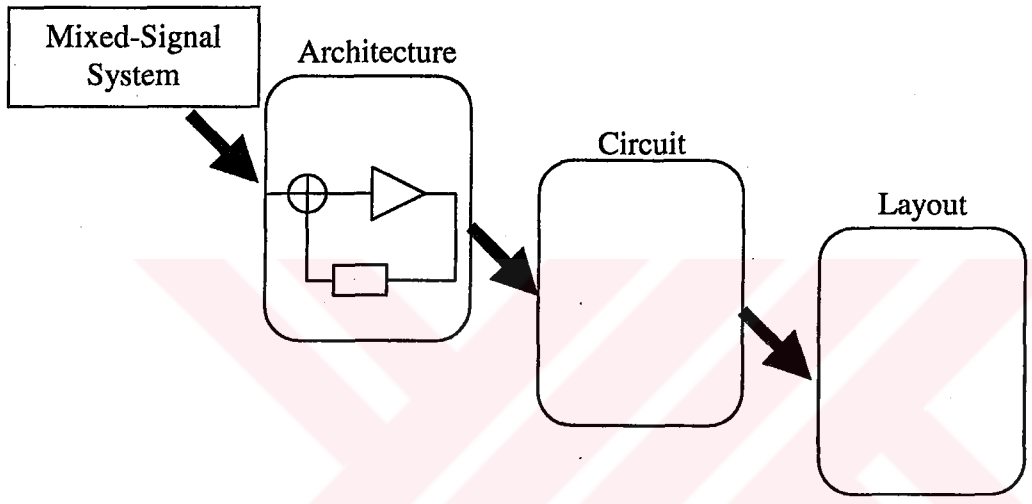

Figure 2.1. A simple decomposition of an analog system

In order to carry out a design in such a hierarchical environment, a top-down design methodology has to be applied. However, to achieve the desired accuracy between different levels can be a major problem. In that case, some design iterations should overcome these problems. Also, performance estimation tools help the designer to choose an adequate topology for a desired accuracy [11].

The top-down approach for analog design has some accepted steps; architecture selection, specification generation, behavioral verification. The architecture should be selected considering the specifications given at system level. The next step is generating the specifications of the next level. The blocks at that level may be created with the given specifications. If decomposition continues the specifications for the next level should be generated. After the generation of the specifications, the model must be verified. A behavioral simulator can be used in order to verify the whole architecture. If the specifications generated do not meet with the simulated ones than some iterations between the levels should be carried out.

2.1. Behavioral Design of ADC's

The top-down approach given above can be applied to the data converters [12-13]. However, the previous work in the literature mostly presents the top-down design of a specific or selected topology. This dissertation proposes a methodology, which deals with the system level topology selection and generating the specifications for the subblocks of the lower synthesis level in a data converter design.

First step is the selection of the topology of the converter. Depending on the given specifications (such as power, area, delay, technology parameters) an appropriate converter type and architecture has to be selected. In order to cover a wide range of applications and the corresponding variations in input specifications, different converter architectures have to be included in the design system's database.

There are several ways available to select the topology. One way is to write approximate analytical expressions that roughly predict the performance, die area, and power requirements of each topology. In this case it is possible to directly compare various topologies and choose one that meets the required performance with the minimum area or power. Unfortunately, such analytical expressions may not be available and the designer may have to extrapolate from experience with similar system designs. And, if sophisticated computer aided design (CAD) tools are available to help to speed the design process, another possibility is to completely design several different topologies and see which one is best.

The next step is the subblock specification generation. The parameters (such as gain, bandwidth) for the subblocks (comparators, capacitor arrays) within the selected architecture have to be determined. This specification generation (mapping) is a critical, nonunique process and it has to manage the trade-offs between the different subblocks. The mapping can be based on either architecture-specific mathematical descriptions or behavioral simulations, and the degrees of freedom in the design can be eliminated using heuristics and/or optimization techniques. The mapping must result in a realizable solution and, preferably, at the lowest overall implementation cost (minimum power/area for the complete architecture).

If simplified equations are used in the specification mapping, a more detailed verification of the converter's performance (e.g., verification of the architecture's integral and differential nonlinearity) can be obtained by means of behavioral simulation at this level. The architecture can be simulated in terms of the functional models of the composing circuit blocks (which should include nonidealities, the statistical variability of the parameters and noise) and their required performances as determined by the mapping process. When the converter architecture does not meet its input specifications, redesign iterations have to be carried out by performing another mapping of the subblock specifications, by selecting another architecture, or ultimately by changing the input specifications. On the other hand, when the desired input specifications are met, then a structured netlist is generated defining the synthesized data converter architecture in terms of the individual circuit blocks (such as amplifiers, comparators, capacitive arrays, etc...) together with their specifications.

The determined specifications of the blocks allow us to pass to the next lower level, circuit synthesis level. In this level, the different subblocks are synthesized and implemented. The overall converter layout can then be obtained by place and route process of the subblock layouts at the lowest level, layout synthesis level. At the circuit level, the netlist information can be passed to the lower-level tools. Flexibility in the design of the subblocks can be provided here by offering access to a variety of lower-level tools such as standard cell libraries, parameterized module generators (i.e. for capacitive arrays) or even circuit-level compilers if an optimum performance is needed (i.e. for OPAMP's or comparator). This flexibility allows for the widest range of input specifications to be covered with a given set of architectures contained in the design system's database.

The resulting subblock layouts are then placed and interconnected to obtain the overall converter layout. A layout extraction can then be carried out to back-annotate all layout parasitics and an ultimate verification of the converter's performance (such as the nonlinearity or the signal-to-noise ratio) can be performed by simulating the whole architecture with the exact parameter values as extracted from the layout. Finally, a simulation model (macromodel or behavioral model) for the complete converter can be constructed that allows simulating of the designed converter as part of a larger system.

In previous years, the above data converter design methodology is being followed by most automatic data converter synthesis programs [12], [14-15]. To allow large flexibility, this methodology should be implemented in an open, hierarchical and standardized framework for the computer-assisted design of analog integrated circuits. A general view of such an environment was proposed at [6], which clearly indicates the different hierarchical levels to be considered for data converters and the flow of information between these levels.

In Figure 2.2, an example for the steps given above is presented. With the given specifications, a topology is selected, which is pipeline in this case. The constraints for the subblocks then generated for each stage. The constraints in this level passed to the stage designer to generate the required specifications for the lower level.

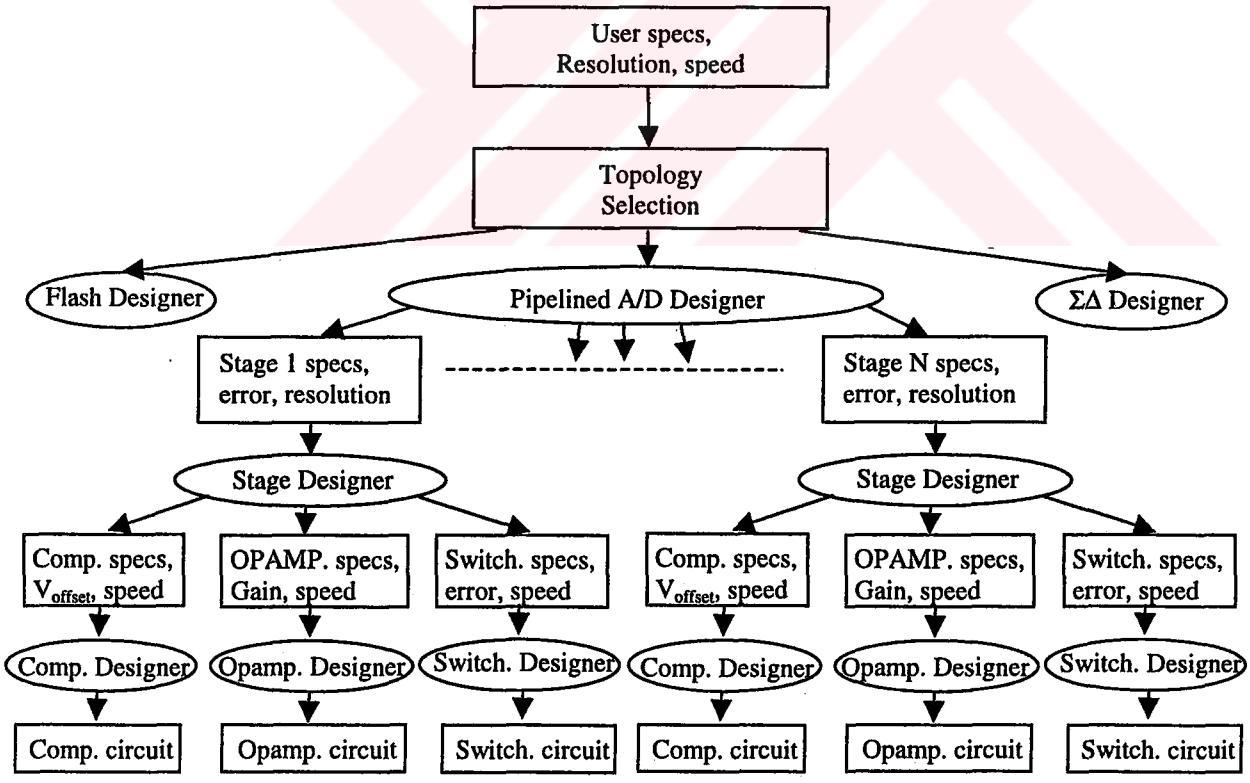

Figure 2.2. ADC decomposition example

2.2. Previous Work

The methodology proposed, basically has a database of different topologies. The topologies proposed in this dissertation are flash, pipeline and Sigma-Delta converter. Each of these architectures has been modeled with some simplified equations. However, these equations do not have enough accuracy to let us call the proposed methodology as a simulation method. The previous work about data converter design generally proposes methods for efficient simulation and/or synthesis of converters.

There are different approaches in the literature. One way is to use circuit level simulators. However, traditional circuit-level simulators cannot handle the complexity of most data converter circuits. For circuit level simulation, all subcircuits of the converter, both analog and digital, are modeled at the device level as an interconnection of basic devices, and the total architecture is simulated with a circuit level simulator like SPICE. The complexity of the present converter architectures, however, is beyond the limits of these simulators. For example, a 10-b flash ADC contains 1024 comparators, each of which can consist of more than ten transistors. If no simple expression for the integral nonlinearity (INL) is available, one general way to verify the nonlinearity specification is to simulate the entire INL characteristic of this converter. This requires many transient simulations, each on a circuit with more than 10^4 transistors. For oversampling converters, high oversampling ratio causes very long transient simulations. So, architecture selection or even verification of a single topology consumes unreasonable time. Alternative approaches have been proposed toward the modeling of the different subblocks. These alternative approaches basically intend to trade off small accuracy degradation for a large gain in simulation speed.

Instead of using circuit level simulation, mixed signal simulation can be used. Thus the simulation of digital circuitry can be replaced by fast logic simulation.

A solution for the above difficulties can be the use of macromodels for subcircuits such as operational amplifiers and comparators. These macromodels are equivalent circuits, with less and usually more ideal circuit elements, which approximate the behavior of the original circuit [16] but which can still be used in existing SPICE-like simulators.

This approach is useful, but is in general limited in that it is hard to find an appropriate equivalent topology for each circuit architecture which accurately matches the behavior of the original circuit, including all nonidealities.

Another and probably the most popular method is to use behavioral models. A behavioral model directly describes the performance behavior of each analog and digital subblock. For analog circuits, this must not only include the nominal input-output behavior but also the nonidealities. The model can be implemented in different ways, and it can be developed specifically for a particular circuit design, for a particular circuit architecture or, generically, for a whole class of circuits. A first approach consists of simulating the performance of a subcircuit with a circuit simulator and storing the resulting data in a look-up table, as is used in the ZSIM simulator for delta-sigma converters [17]. This is very efficient for final verification after completion of the converter design, but is inefficient during the design phase as the model is extracted for one particular subcircuit design. Changing the architecture or even changing some parameters in the same architecture requires the complete generation of a new model, and parameterizing the look-up tables is extremely expensive both in generation time and memory storage.

A second approach consists of describing the subcircuit performance as a mathematical subroutine using a language, like C [18-20]. But this method is limited with the ability of the representing the formulas explicitly. In other words, the input output behavior of the ADC should be modeled precisely. This method reduces the CPU time considerably. In a behavioral model, the change in any parameter directly affects the output.

Behavioral modeling, with the increasing capabilities of modeling languages such as VHDL-A or VHDL-AMS, is gaining more and more interest. The advantage of constructing architectures with predefined behavioral models in a HDL is very hot topic in recent research [21].

Together with the above simulation methods, synthesis systems for specific topologies are presented. Most of these synthesis systems are academic. Although these systems are restricted with a limited architecture set, several successful automatically

designed and fabricated data converters are available. And some data converters designed with MIDAS [18] are available commercially.

The notable earlier work in this area can be found in [22]. The other recent synthesis tools are CADIS [23], HiFaDiCC [24], CATALYST [25], OASYS [26]. The capabilities of these tools are given in Table 2.1 [6], [13]. Also, there are other efforts for algorithm driven synthesis [27]. In this work, the architecture is modeled by means of signal flow graph and a pattern recognition process is executed which maps the recognized graphs to library blocks.

Table 2.1. The previous works about ADC automation

Tool	CADIS	HiFaDiCC	AZTECA	CATALYST	OASYS	MIDAS
architectures						
technology	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS
optimization	yes	yes	yes	yes	yes	yes
layout	using UCB layout tools	yes	no	no	no	by Philips layout tools
verification	behavioral simulator	general simulator	behavioral simulator	behavioral simulator	dedicated formulas	behavioral simulator
working Si	yes	not reported	not reported	not reported	not reported	yes

The previous work about data converter design can be classified into two; simulators and synthesizers. The available tools simulate or synthesize the selected topologies by a top-down methodology are presented earlier in this section. This dissertation may be placed above all of these tools in such a top-down design. The proposed methodology selects the proper architecture by using simplified behavioral models, which includes nonidealities such as offset, gain error, nonlinearity. Since the methodology should explore more than one architecture space, the models should be as simple as possible in order to achieve reasonable solution in reasonable CPU time. The tools presented in Table 2.1, synthesize the selected architecture and use more precise noise models since it only deals with one topology. Although the proposed methodology can be evaluated as a performance estimation tool more than a synthesis tool, with the libraries plugged in the databases of the methodology allow the proposed tool to function like synthesizer. Since subblocks are determined, the only process is to connect the subblocks and verify the whole architecture.

3. METHODOLGY

The proposed methodology selects the architecture by means of the given constraints. With the given constraints, the solution space is explored. Also, the methodology uses a database of different topologies to generate the solution set. Currently, two topologies are modeled and added to the database. Each topology takes the inputs from the user and estimates the performance of the ADC which results in the generation of subblock specifications such as OPAMP gain, comparator offset, resistor values, speed of the combinational logic. In order to give the requested values, process parameters with which the ADC will be implemented should be available. The code, which evaluates the methodology, reads the parameters from a parameter file. Also, the component parameters, such as comparator offset or number of gates used in the encoder, have to be read from a file.

The power of this methodology is that it is not limited with the available library. If no solution is available after the solution space or the library database has been searched, then the methodology widens the solution space, which, is only limited by the parameters given by the user. The solution can be found by sweeping these parameters. The limiting parameters for each topology are given in corresponding sections. For flash ADC architecture these parameters are total power, total area, speed, comparator offset, comparator and area. The methodology developed generates a solution set for each topology, which consists of different ranges of requested parameters. Also, a cost function has been implemented in order to choose the optimum solution in a given solution set. Another advantage of this methodology is that the user can limit the solution space in any dimension. In other words, the user gives the ranges of whole parameters. One drawback of this method is that the complex expressions of topologies may lead to a very long CPU time. But since the analytical expressions of input output relations are not derived (because it is extremely hard and complex in many topologies) this method gives reasonable solutions in acceptable CPU time. Since the increments of the parameters are determined by user the first run can be made by using large increment values in order to get an idea of the range of the solution. Then a refinement of the solution can be done by using small increments.

In order to keep the run time of this methodology reasonable, the equations used in this methodology are simplified. Hence, the solution is not as accurate as a simulation tool. However the methodology should give a solution as accurate as it can be.

The methodology gives a solution set for the given constraints. Also a cost function has been implemented in order to present the optimum solution by means of four specifications: power, area, delay, speed. The user may adjust the constants for the cost function.

4. FLASH ADC

The fastest of all types of high-speed analog-to-digital converters is the flash or parallel type of converter. The flash converter is considered to be the fastest ADC because the conversion takes place in a single cycle, hence the name "Flash". The resolution of flash converters tend to be limited to eight bits due to the fact that the amount of circuitry doubles every time the resolution is increased by one bit. A block diagram of a flash converter is shown in Figure 3.1. The analog input signal is applied simultaneously to the 2^n-1 latching comparators. They are arranged in a "thermometer" code fashion with each comparator's reference biased one least significant bit (LSB) higher than that of the preceding comparator. The individual reference voltage is derived from a resistive voltage divider string. With the analog signal applied to the flash converter, each comparator will compare the signal level to the reference level. If the signal level is higher than the reference level, the comparator will output a logic "1". Also, the comparator will assume a logic "0", if the signal amplitude is lower than the reference. The resulting thermometer code is then converted to a binary output in the logic encoder. Even though the design of flash converters is highly repetitive in its structure, it demands a high level of matching between the parallel comparators. One of the major contributors to the nonlinearities is the comparator input offset voltage. The offset should be less than ±1/2 LSB not to degrade the monotonicity of the converter. Similar effects could be caused by the bias and input offset currents of the comparators. Together with the resistance of the reference ladder, they will be added to the offset voltage. The reference voltage resistive ladder contributes as a secondary effect to the error.

The flash module in the methodology, which generates the solution set for the flash ADC, developed based on a structure given in Figure 2.1. The flash ADC is partitioned into three sections: the resistor string, comparators and the digital circuitry. The methodology evaluates each of these partitions separately. The methodology estimates the power or area, as well as relations of these parameters with others for each section.

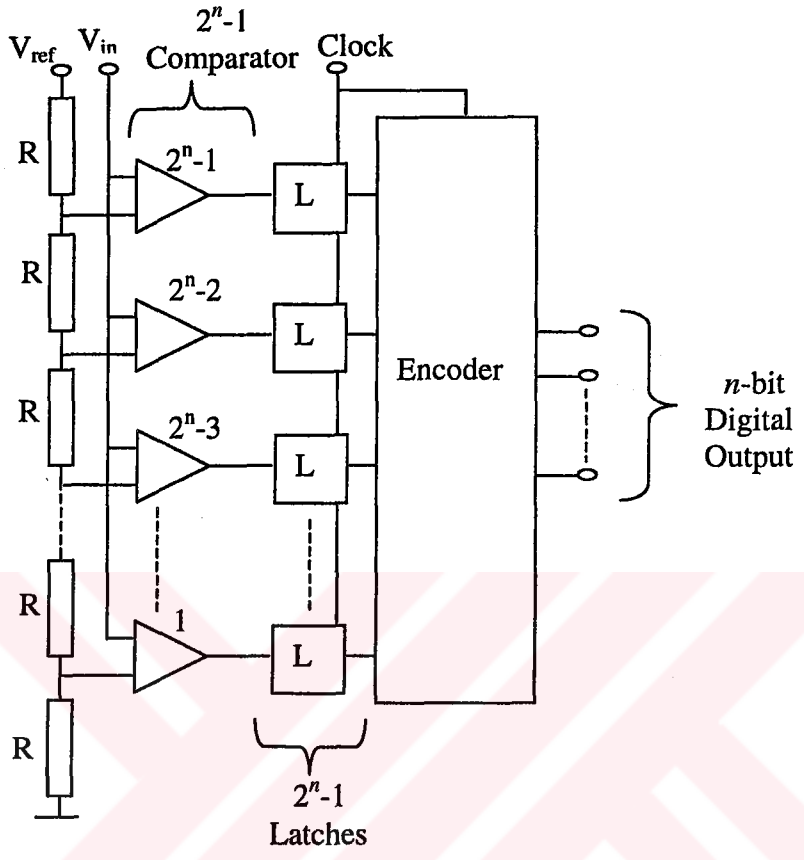

Figure 4.1. Flash ADC

4.1. Power Estimation

Power of the resistor string calculated as follows:

$$P_{res} = \frac{V_{ref}^{\ 2}}{2^n \ R} \tag{4.1}$$

where P_{res} represents the power dissipated at the resistor string. The powers dissipated at the other partitions of the ADC are:

$$P_{comp} = (2^n - 1).(I_{sup}).V_{sup} \tag{4.2}$$

$$P_{dig} = C_{dig} V_{dd}^{\ 2} f \quad k \tag{4.3}$$

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

where, P_{comp} and P_{dig} are powers of comparators and digital circuitry. The f value represents the frequency of the encoder and k is the switching probability of the encoder. C_{dig} is the overall capacitance of the encoder.

4.2. Area Estimation

Like the power estimation, the area estimation of the resistor string also requires technology parameters. From the resistance values, the ratio of w and l are calculated. Since the length of the resistor is indirectly proportional with the resistance and the smaller resistance area is desired to decrease total area, the length has been set to the minimal value. The minimum allowed length for the technology used in the examples is 0.7µm. The poly resistance is used to implement resistances. The sheet resistance of the poly read from technology file. The resistance can be calculated as follows:

$$R = \frac{w}{l} R_s \tag{4.4}$$

The resistance can be one of the desired outputs. In that case, the width value is calculated from the given resistance. The area of the resistor string is simply:

$$A_{res} = wl2^n \tag{4.5}$$

The area of the comparator is an input parameter. The total area of the comparators can be calculated as follows:

$$A_{comp} = (2^n - 1)a_opamp \tag{4.6}$$

The area of the OPAMP is critical for the total area since for increasing n, A_{comp} increase exponentially.

The third stage is the digital part, in other words, an n bit encoder. The parameters of the encoder, like number of gates for n bit encoder should be available. The methodology uses a parameter file for encoder block. The parameter file is a look-up table generated for

n bits. A generic C code has been written in order to generate the VHDL code of the *n* bit encoder. Then, the VHDL code synthesized with Autologic II. Also, available optimizations are applied and the number of gate value for each *n* has written to the parameter file. The synthesized designs are simulated with QuickSim II and the delay values are observed.

The area of the digital circuitry is directly proportional with the number of gates used. The area of each gate can be found from technology data book. The areas are specified in terms of *uca* (Unit Cell Area). Since different synthesizers may use different number of gates and structure, the area is calculated with an average gate area. The minimum and maximum gate areas must be specified in the technology file. The number of D flip-flops can be easily calculated, so, the area of D flip-flop calculated separately. The area of the digital circuitry is as follows:

$$A_{dig} = k_2 \left[\left(\left(n + 2^n - 1 \right) a_dff \right) + \left(\frac{a_\min + a_\max}{2} n_gates \right) \right] uca \qquad (4.7)$$

The total area of gates cannot represent the total area of the circuitry. The area used for routing should be added. The coefficient k_2 is used for this purpose.

4.3. Error Sources

The first stage of the flash ADC is a resistor string. In order to show the thermal noise in the resistor string, a voltage source added to the string for each resistor. The thermal noise generates a voltage e_t.

$$e_t = \sqrt{4KTR}\ x\ BW_{comp} \qquad (4.8)$$

Also, the mismatches of the resistors contribute another voltage source. The mismatch is the standard deviation of the relative differences of two identical resistors. The models of the resistors, which are at most 10μm apart from each other, are used in the algorithms to calculate the mismatch. For more accurate estimation, the resistors should not exceed this range. The resistance values are functions of *w*, *l* and the sheet resistance

(R_s). The maximum deviation from the standard value can be calculated by using the 3 sigma design method [7]. The deviation equals to 3 sigma in this method. As a result, mismatch contributes a voltage difference:

$$e_m = \frac{V_{ref}}{2^n R} 0.03.m.R \tag{4.9}$$

The sum of these voltages with the offset voltage of the comparator should not exceed the half of the voltage, which is equal to the one LSB.

$$e_m + e_t + vos < \frac{v_{ref}}{2^{n-1}} \tag{4.10}$$

4.3. Speed Estimation

The flash architecture is fast enough to give the digital output in one clock cycle. Hence, the delay for this architecture is one clock cycle. The clock frequency is the only factor in speed estimation. The major factor in defining the clock frequency is the speed of the comparator. The comparator, mostly a clocked one which include a latch at the output, should be fast enough to process in several MHz range. Various comparator designs are available in the literature. The most important specification of the comparator is the slew-rate. In the methodology, the slew-rate is calculated from the rise or fall time. Half period is reserved for the latch and the digital circuitry. Hence, the comparator should give the correct output at the other half of the clock period. One other factor which should be taken into account is the minimum and maximum voltage for the logical "1" and "0". Since the output of the comparator exceeds the logical "0" or logical "1" threshold, the latch can fetch the right logical value. There is no need to wait until the comparator output reaches the maximum or minimum value.

4.4. Methodology for Flash Converter

The module for flash converter operates in the same order as it was presented above. First of all, the desired *SNR* should be given in order to estimate the effective number of

bits (ENOB) for the topology. The *SNR* that was limited from the bottom by the quantization noise. Which means that the quantization noise determines the minimum number of bits for the desired *SNR*. The expression below shows the mathematical expression for the quantization noise [28]. The n represents the number of bits required to achieve the desired *SNR*.

$$SNR = n.6.02 + 1.76 \text{ dB} \tag{4.11}$$

The given *SNR* limits the number of bits. The methodology, proposed for flash converter in this dissertation is limited by eight bits. This is mainly because the pipeline architecture offers better results by means of complexity, area and power. Also, knowing the least number of bits limits the solution space considerably.

The second step of the methodology is to read the technology parameters. Since these parameters do not change while the functions of the topologies are executed, they should be read only once per each program run. Also the other parameters used by the methodology are read from the files. For example, the parameters for the comparator are read at the beginning and stored at the memory for fast processing. However this may be not efficient for a large library database.

After the parameters are read, the power of the available solutions is generated. This solution is limited by the maximum power given by the user. Area is the next step in the methodology. Since the power estimation reduces the number of solutions in the design space, the area estimation deals with a narrowed design space. The area estimation again, decreases the number solutions. At this time, since more precise values are determined by the previous stages, errors in the flash architecture are calculated in order to eliminate the designs, which will not function appropriately.

The last step is to estimate the speed of the architecture and check if the solutions fit with the given specifications. After the speed estimation the solution space is available. However, the optimum solution should be determined by means of area, power and speed. The optimum solution can be achieved by a cost function, which is the case in this methodology. As the functions are reducing the number of solutions, a cost function also

processes the solutions in order to determine the optimum one. The optimum solution may be adjusted by the parameters given by the user.

5. PIPELINE ADC

A pipeline ADC consists of a number of consecutive stages. The stages are similar in their function and each stage generally resolves only one or two bits. Each individual stage consists of a sample and hold, a low resolution flash ADC, a low resolution digital-to-analog converter (DAC) and a summing stage including an interstage amplifier for providing gain. The outputs of each stage are combined in the output latch. Stage one takes a sample of the input voltage and makes the first coarse conversion. The result is then the most significant bit (MSB) and its digital value is fed to the first latch. As the residue of the first stage gets resolved in the subsequent n-stages the MSB value is rippled through the n number of latches in order to coincide with the end of the conversion of the last stage. Then all data bits are latched in the output register and then the output is available.

Figure 5.1 shows a typical pipeline ADC architecture. The structure is highly repetitive. There are *NS* number of identical stages, each quantizes *Ni* bits. So the overall resolution is *NS* times *Ni*. Each stage samples the output from the previous stage and quantizes to *Ni* bits digital codes using flash ADC architecture. The codes then are converted to analog signal by *Ni*-bit DAC and subtracted from the sampled signal. The residue then is amplified by a gain of $G=2^{Ni}$. The output register combines the output bits from each stages and gives the final digital codes. Stage one gives MSB's, stage *NS* gives the LSB's.

Due to the small dimensions and low power consumption, the pipeline architecture is more suitable for high-resolution applications than flash converters, but is also easily affected by circuit imperfections. The errors caused by these imperfections are presented below. Although many error sources are available, correction circuitry compensate these errors efficiently.

Pipeline architecture offers relatively high speeds and approximately linear hardware cost with resolution. A very important advantage of pipeline architecture is that it allows digital error correction and digital gain calibration (for offset/gain errors, and nonlinearities) and power minimization through capacitor scaling. Pipeline ADC is most

popular in high speed (several MHz~50MHz) and high resolution (above eight bit) application where latency is not a concern.

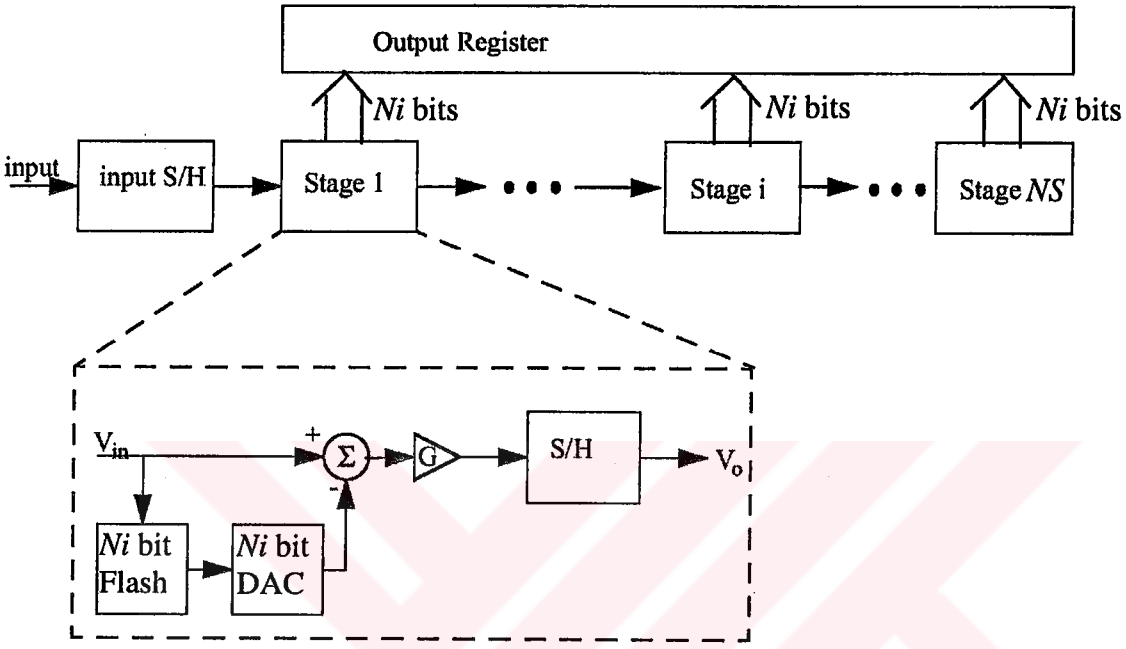

Figure 5.1. General model of a pipeline ADC

The main advantage of pipeline ADC's is that they can provide a high throughput rate with moderate design complexity and low power consumption. This is because of the concurrent operation of the *NS*-stages. The associated data latency is not a limitation in most applications. Two main clock phases are required per conversion; because the pipeline ADC uses flash converters. Therefore, the maximum throughput rate can be high. After the initial data latency time, the data representing each succeeding sample is output with every following clock pulse.

In recent studies, multiplying DAC's (MDAC) are used for the function of three components: sample and hold (S/H), DAC, residue amplifier. The use of MDAC for recycling or pipeline ADC's firstly presented by [29]. Figure 5.2 shows the architecture of the two step recycling ADC. In the following years, the studies mainly concerned with pipelined designs with low stage resolution (2-bit or 1.5-bit) instead of two stage designs with high stage resolution.

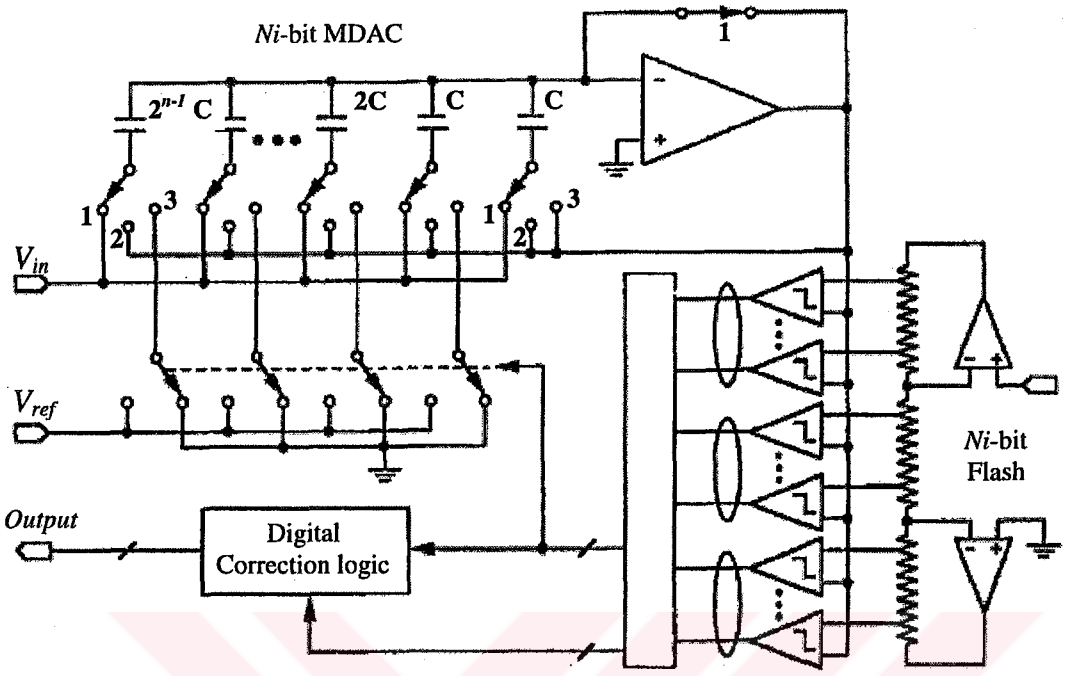

Figure 5.2. The recycling two step converter [29]

The residue calculation of the MDAC presented above is as follows. During the sample-phase (1), the input signal is applied to all the capacitors connected in parallel while the OPAMP is reset. During the hold phase (2), the full set of capacitors closes the feedback loop around the OPAMP to complete the S/H sequence. The output voltage of the OPAMP is then the opposite of the input voltage. Since it is applied to the flash converter, the latter outputs the coarse code of the input signal. Residue calculation then takes place (3). The bottom plates of all the capacitors except the last one, which is the only one remaining across the OPAMP, are connected either to the ground or to the reference V_{ref} depending of the outcome of the flash conversion. The charge stored in the capacitor's bank now represents the analog voltage created by the output code of the flash converter. Since it is the same as the input signal, the remainder is transferred to the integrating unit-capacitor around the OPAMP. Hence, the MDAC outputs the residue multiplied by the interstage gain, and the output of the flash converter is now the fine code.

Recycling converters may easily be changed into pipelined converters. Instead of being recycled, the residue is fed to the next stage. The architecture is very straightforward for every MDAC already has its own built-in S/H. One of the first multi-bit pipelined

converters of this kind was the 5 MS/s, 180 mW, 9-bit device [30] that consisted of four, 3-bit wide, stages which uses one extra redundancy bit for error correction. Much effort has been devoted since then to design high performance pipelined converters. Present pipelined converters display sampling rates in the range of 20 to 100 MS/s while their resolutions are mosty 15 bits. The power consumption becomes large when the sampling rate exceeds 50 MS/s. The values between 800 mW and 1.1 W have been reported. An exception is the one with 250 mW in [31], which uses low power design techniques and reduces the number of OPAMP's from seven to three by sharing the same amplifiers between adjacent stages. Fast converters [32] achieve 100 MS/s sampling rate with two stages. In [33] the same sampling rate achieved using a 4-bit front-end that distributes the residue to several identical time interleaved fine channels. Also recent research about parallel pipeline architecture extends the performance by means of resolution and speed [34]. A 50 MS/s, 10-bit, 900-mW converter is described by [35]. Pipelined converters in the 50 MS/s range currently reach 12 bits [36].

5.1. Error Sources

In this section the errors in a general pipeline architecture are presented. The methodology proposed, uses a MDAC, which combines and eliminates some of these errors. The dominant error sources of a MDAC are presented in the following sections.

The main error sources in a general pipeline architecture are gain, offset and nonlinearity errors in sub-ADC (SADC), DAC and sample and hold amplifiers (SHA). The other errors such as settling errors can be modeled as a combination of these errors [37]. A model for errors in a pipeline ADC has been introduced by S.H. Lewis [37]. Since then, the introduced model became very popular and used by other researchers [35], [38-39].

The model of an analog path in a pipelined ADC presented below. It consists of *NS* stages, and each stage contains a SHA and an error source, e_i. In general, each e_i represents the offset, gain, nonlinearity, and quantization errors in that stage. The input-referred error, e_{in}, that is equivalent to the contributions of all the individual error sources is,

$$e_{in} = e_1 + \sum_{i=1}^{NS-1} \frac{e_i}{G_i} \tag{5.1}$$

where, G_i stands for the the product of gains (G) of the stages before the stage i. The general expression for the gain is as follows:

$$G = 2^{Ni-x} \tag{5.2}$$

In Equation (5.2), x represents the number of bits for redundancy. One bit redundancy is used in this methodology, which can handle errors as much as the amplified residue. Equation (5.1) shows that SHA gains greater than one reduce the effects of nonidealities in all stages after the first on the error of the entire conversion. For each nonideality, the reduction factor is the product of all the SHA gains before the nonideality except the SHA gain in the first stage. As a result, to limit the ADC error arising from each error source to less than 1/2 LSB, Equation (5.3) should be confirmed.

$$e_i \le \frac{FS}{2^{n+1}} G_{i-1} \tag{5.3}$$

The FS stands for full-scale conversion range. If errors are similar then,

$$e_i = e, \text{ for } 1 \le i \le NS \tag{5.4}$$

In this case Equation (5.1) can be rewritten as:

$$e_{in} = e\left(1 + \sum_{i=1}^{NS-1} \frac{1}{G_i}\right) = e.F \tag{5.5}$$

Equation (5.5) shows that the combined effect of identical errors in all stages is greater than the effect of only the first-stage error by a factor, F, that depends on the SHA gains. If $G = 1$, $F = NS$, and if $G >> 1$, $F \approx 1$. A boundary between these two limiting cases occurs when $G = 2$ and $F \approx 2$. Therefore, to limit the effect of errors in all stages after the first so that the first-stage error dominates the total ADC error, the stage resolution should

be chosen so that $G \geq 2$. We can conclude from that, large stage resolution reduces the error of multistage ADC's.

The errors of each functional block can be modeled in similar way. An overview of these blocks with the constraints they introduce, is presented in the following sections

5.1.1. Sub-ADC (SADC) Errors

A sub-ADC is used in the pipeline architecture. A flash quantizer can be preferable among others since 2-bit stage resolution is very common in pipeline architecture. SADC offset, gain, and nonlinearity errors move the SADC decision levels. Define the correction range as the amount of SADC decision-level movement that can be tolerated without error in the ADC output code. In all stages except the last, redundancy and digital correction can eliminate the effects on the ADC linearity of all SADC errors up to the correction range. Since the output of the last stage is not corrected, however, SADC errors there do cause ADC nonlinearity but in an amount that is diminished by the combined interstage gain before the last stage. The stage redundancy, x is usually selected to provide enough correction range to eliminate the effects of these SADC errors in all stages except the last. Through out this dissertation one bit stage redundancy, which doubles the output range of SADC has been selected.

5.1.2. DAC Errors

The main errors of a DAC are offset, gain and nonlinearity. These errors can be modeled as an input referred error. For DAC offsets, the error can be replaced by an input-referred stage offset and an offset in series with the SADC. If the correction range is not exceeded by the combination of all errors that shift the SADC decision levels, the effect of the SADC offset is eliminated by the digital correction. The remaining input-referred stage offset is equivalent to a DAC offset in the $(i - 1)$ st stage. Using this process repetitively, the offsets in all the DAC's can be referred to the input of the ADC. Therefore, to eliminate the effect of DAC offsets on ADC linearity, the stage resolution, n, and redundancy, x, are usually selected so that the correction range is not exceeded by the combination of all errors that shift the SADC decision levels.

DAC gain error can be replaced by three gain errors: one in series with the stage input, one in series with the SADC, and one in series with the stage output. If the correction range is not exceeded by the combination of all errors that shift the SADC decision levels, the effect of the gain error in series with the SADC is eliminated by the digital correction. The two remaining gain errors contribute interstage gain errors, which have the same effect on ADC linearity as the SHA gain errors that are presented later in this section.

A nonlinearity error in the i th DAC can be modeled by e_{off} that depends on the DAC output. Although the digital outputs of this stage are correct with such an error, the residue output of this stage, is incorrect exactly by the amount of the DAC nonlinearity. To limit the resulting nonlinearity to ± 1/2 LSB,

$$e_{offi} \leq \frac{FS}{2^{n+1}} G_{i-1} \tag{5.6}$$

The first stage again is the dominant source in error. If the stage resolution is selected so that only two-level DAC's are required, the DAC's are inherently linear, and the accuracy of the SHA gain determines the ADC linearity. If the DAC's have more than two output levels, however, they are not inherently linear. In order to overcome ADC nonlinearity errors caused by component mismatches in such DAC's, the self-calibration techniques in pipelined, multistage ADC's with more than 10-bit resolution can be applied.

5.1.3. SHA Errors

An offset in the i th SHA is equivalent to an offset in the (i - 1)st DAC. Therefore, the conclusions made earlier about the effects of DAC offsets apply also for SHA offsets.

The expression of gain error e_{gain} can be reduced to Equation (5.7) [37].

$$|e_{gain}| \leq \frac{G_{i-1}}{2^n} = \frac{1}{2^{n-(Ni-x)}} \tag{5.7}$$

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Equation (5.7) shows that each SHA gain (after the first) must be accurate enough to preserve the combined linearity of the resolution remaining after the SHA. For $2 \leq i \leq NS$, residue is maximum when $i = 2$. Therefore, the allowable SHA gain error is minimum for the second-stage SHA (which is the first interstage SHA). The equation also shows that increasing n reduces the required gain accuracy because the resolution remaining after the first interstage SHA decreases as Ni increases.

A nonlinearity error in the i th SHA can be modeled similar to the gain error which is dependent on the residue generated by the previous stage. Because the ADC operates on the SHA outputs, SHA nonlinearity causes ADC nonlinearity in an amount that depends on the ADC resolution remaining after the SHA. Therefore, to limit the resulting nonlinearity to ±1/2 LSB, the fractional variation in the gain of the input SHA must be less than one part in 2^{n+1}. In practice, SHA linearity does not usually limit ADC linearity except in ADC's that use correction techniques to overcome interstage gain errors and DAC nonlinearities.

5.1.4. Optimum Stage Resolution

Some of the errors presented above effect the number of stages in a pipeline. But the errors are not the only factor for the optimum stage number and also in stage resolution. For example, contribution of gains in the input referred error expression lead us to the stage resolution of two bits. However the factors like area, power and speed (conversion rate) must be taken into account. In [37] it is shown that, the effect of conversion rate is limited by the stage resolution. In order to maximize the conversion rate the minimum stage resolution should be used. Similar to these, the area and power factors add new constraints. In [37] it is shown that the problem of minimizing the total area can be reduced to minimizing the total number of comparators in an ADC, which is done by using the minimum stage resolution. Therefore, the minimum stage resolution minimizes the required die area. By using similar approach it can be shown that reducing the stage resolution can minimize the power [37].

As a result, 2-bit stage resolution decreases the area, power and increases the conversion rate for pipelines up to 10 bits.

5.2. Error Sources in MDAC

A common pipeline architecture is given below in order to show the usage of MDAC in a multistage pipeline ADC.

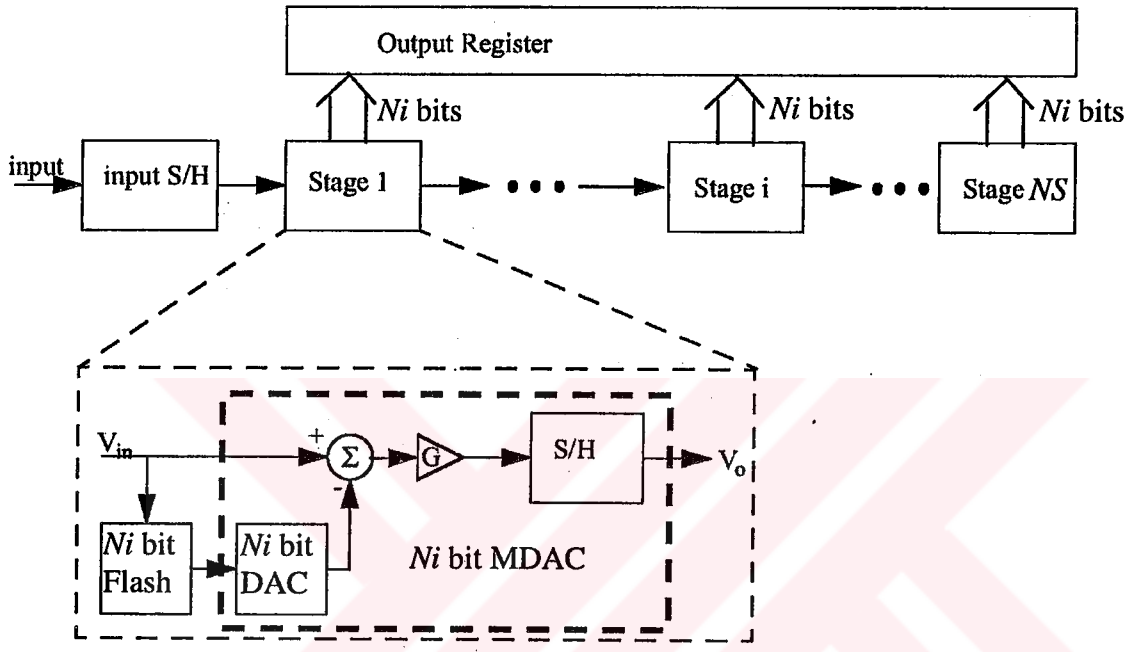

Figure 5.3. Pipeline architecture with MDAC

In Figure 5.3 the MDAC replaces the blocks DAC, SHA and residue amplifier. The structure of MDAC is shown in Figure 5.2. The basic building blocks of an MDAC are an amplifier and a capacitive array. The MDAC has three phases: sample, hold, residue amplification. Because of the capacitive nature of the load (the input capacitance of the SADC and the input capacitance of the MDAC of the next stage), the amplifier employed in the MDAC can be implemented using single-stage operational transconductance amplifier (OTA).

If we consider the recent technologies, where chip sizes continue to shrink, the capacitor array of the MDAC introduces a new source of error, thermal noise. Since the capacitance values are getting smaller and smaller, thermal noise becomes comparable to or even dominant with respect to the other error source.

The maximum resolution achieved with the MDAC introduced above is limited mainly by thermal noise, by the nonlinearities of the MDAC's produced by capacitor

mismatches, and also by the residue amplification error that is due to the amplifier nonidealities [38-40]. The thermal noise can be reduced by appropriate sizing of the capacitors and the nonlinearities in the MDAC can be reduced by self-calibration techniques and by an appropriate amplifier design.

Error sources other than MDAC's are SADC's. But the errors from the flash quantizer can be digitally corrected if they are kept within the range covered by the redundancy created between consecutive stages. For this purpose, an extra redundancy bit can be added to the architecture [28], [41-42].

5.2.1. Thermal Noise

$$TTN = \sqrt{\frac{N_{S/H}^2}{1} + \frac{N_{MDAC1}^2}{G_{S1}^2} + \frac{N_{MDAC2}^2}{G_{S1}^2 G_{S2}^2} + \ldots\ldots + \frac{N_{MDAC(NS-1)}^2}{G_{S1}^2 \ldots G_{S(NS-1)}^2}} \tag{5.8}$$

Considering that the main sources of thermal noise are the on-resistances of the switches and the OPAMP's of the MDAC's and of the front-end S/H (can be implemented by using a MDAC), then the rms value of the total thermal noise (*TTN*) referred to the input of the converter is given by Equation (5.8), shown at the bottom of the page, where $N_{S/H}$ and N_{MDACi} respectively, are the output referred RMS noise contributions of the S/H and of the MDAC and G_{Si} represents the closed-loop gain of each stage during residue amplification. According to the simplified noise model shown in [38], the mean square noise contribution of each MDAC, $N_{MDACi,}$ will be a sum of three terms.

$$N_{MDACi}^2 = V_{S/H}^2 + V_1^2 + V_2^2 \tag{5.9}$$

The first term is a sampled-and-held component introduced by the on-resistance of the CMOS switches during the sampling phase. The second term is a broadband contribution due to the on-resistance of the CMOS switches during the residue amplification phase. The third term is also a broadband contribution introduced by the amplifier itself during residue amplification. Thus, the output referred mean square noise introduced by each MDAC can be expressed approximately as:

$$N_{MDACi}^2 = \phi\left[\frac{KT}{C_{MDACi}} + \left(4KTR_{ON} + + KTR_{EQ}\right)BW\right]G_{Si}^2 \quad (5.10)$$

where C_{MDACi} is the total input capacitance of the MDAC, K is the Boltzmann's constant, and T is the absolute temperature. The constant Φ can take the values of either one or two depending on, respectively, whether the circuit is implemented in a single-ended or fully differential configuration. R_{ON} and R_{EQ}, respectively, are the on-resistance of the CMOS switches during the residue amplification phase and the pseudoresistance at one input of the amplifier caused by switch capacitors [38].

In order to obtain a pipeline with n effective bits of resolution, one should guarantee that the total thermal noise is below the quantization noise and, thus, for a given reference voltage, the condition [28], [38],

$$TTN < \frac{\phi V_{ref}}{2^n\sqrt{12}} \quad (5.11)$$

must be satisfied. We can conclude that, the capacitor sizes must be carefully chosen in order to limit the thermal noise contribution.

5.2.2. Mismatch Error

One of the main sources of the errors is mismatch errors of the capacitor array [43-45]. The errors can cause gain and nonlinearity errors [46]. In order to show the effect of the mismatch, the ideal amplified residue expression and the expression of the output of the MDAC affected by the mismatches at the capacitor array, are presented below.

$$V_{RA} = 2^{Ni-1}\left(v_{in} - V_{ref}\sum_{i=1}^{Ni} 2^{i-Ni-1}bi\right) \quad (5.12)$$

The ideal amplified residue of a MDAC is presented at Equation (5.12). A feedback capacitor with the twice the capacitance of a unit capacitor gives a gain of 2^{Ni-1}.The

weighted sum of the capacitors determine the voltage which will be subtracted from the input. If a mismatch, shown by ε, is present, the relative capacitor values are,

$$C_f = (1+\varepsilon)C_{unit}, \quad C_i = (2^{i-1}+\varepsilon_i)C_{unit} \quad i = 1,2,.....Ni \tag{5.13}$$

With the capacitor values above, an error contributes the expression (5.12).

$$V_{RA} = 2^{Ni-1}\left(v_{in}\left[1+\frac{1}{2^{Ni}}\left(\varepsilon+\sum_{i=1}^{Ni}\varepsilon_i\right)\right]-\frac{V_{ref}}{2^{Ni}}\sum_{i=1}^{Ni}\left[2^{i-1}+\varepsilon_i\right]bi\right) \tag{5.14}$$

By comparing Equation (5.12) with Equation (5.14), we may conclude that mismatch error contributes a gain error and another error that depends on the code applied by SADC. The first error term in Equation (5.14) represents the gain error. The second error term shows the nonlinearity error of the MDAC. The gain error that affects the input voltage can be referred as an analog error, where as the nonlinearity error depends on the digital code applied. But the calibration techniques offer many solutions to these errors. The errors both in analog and digital domain may be calibrated with efficient digital correction mechanisms.

There are some layout techniques available, which reduces the systematic mismatch error significantly. One technique for improving the matching of capacitors is called the "common centroid" layout style [7]. Basically this technique prefers big capacitors made up of many, small, unit capacitors.

In this methodology, calibration techniques are not implemented. However, the information about need of calibration for a stage is important. The methodology gives the number of stages that require calibration.

5.2.3. Other Errors

The limited gain of the amplifier generates an error. In order to limit the effect of this error,

$$A_0 \geq 2^{M-Ni}\left(2^{Ni}+\frac{C_p}{C_{unit}}\right) \tag{5.15}$$

must be satisfied. The M is the accuracy needed by MDAC. C_p represents the parasitic capacitance at the input of the amplifier.

5.3. Power Estimation

The power estimation for the pipeline can be computed in a straightforward way as:

$$P_{tot} \cong \sum_{i=0}^{NS-1} P_{MDACi} + \sum_{i=1}^{NS} P_{Flashi} + \sum_{i=1}^{NS} P_{synci} + \sum_{i=1}^{NScal} P_{selfi} \tag{5.16}$$

The functional blocks in pipeline ADC can be classified into four. The blocks are, MDAC, flash SADC, synchronization circuitry and if available the self-calibration circuit. However, the first two terms usually dominate the total power.

The power dissipated in each MDAC is the sum of the static power dissipated in the OTA and a dynamic contribution corresponding to switching of the capacitors at the sampling frequency, and can be approximately given by:

$$P_{MDACi} = \left(I_{supa} V_{supa}\right) + \phi C_{MDACi}(1+K_{Ci}) V_{ref}{}^2 f \tag{5.17}$$

where the constant K_{Ci} reflects the parasitic capacitors mainly because of switches.

The power dissipated in the flash can be determined by the static contribution of the preamplifiers of the comparators, by the dynamic contribution, which depends on the energy consumed by the latch per comparison and on the switched capacitances in each comparator, and by the static contribution of the R-string. This is approximately given by:

$$P_{flash} = (I_{sup} V_{sup}) 2^{Ni-1} + C_S V_{ref}{}^2 f K_F + \frac{V_{ref}{}^2}{2^{Ni} R} \tag{5.18}$$

The C_S represents the total switching capacitance in the flash quantizer. The constant K_F represents the probability of switching. R is the unit resistance in the R-string.

$$P_{synci} = Ni(NS+1-i)P_{FF}.f \tag{5.19}$$

The P_{FF} is the energy consumed per cycle in a flip-flop.

Also, if available, the self calibration circuitry consumes some power which is directly related to the power consumed in each fetch of the RAM [38], [41].

$$P_{self} \approx P_{RAM}.f \tag{5.20}$$

The P_{RAM} stands for the power consumed by the RAM in each fetch operation.

5.4. Area Estimation

The total are can be found by:

$$A_{tot} = \sum_{i=0}^{NS-1} A_{MDACi} + \sum_{i=1}^{NS} A_{flashi} + \sum_{i=1}^{NS} A_{sync} + \sum_{i=1}^{NScal} A_{self} \tag{5.21}$$

Again area can be classified into four blocks.

$$A_{MDACi} = a_ota + \phi A_{CAP.ARRAY} + \phi A_{SWITCHES} \tag{5.22}$$

The functional block of the MDAC are an amplifier(generally an OTA), a capacitor array and the switches which determine the phase of the MDAC.

$$A_{flashi} = A_{res} + A_{comp} + A_{dig} \tag{5.23}$$

The expression for the area of the flash quantizer is given in Equation (5.23). The area of the comparator is the dominant factor in the expression. The detailed expression is given in Equation (4.6).

$$A_{synci} = Ni(NS+1-i)A_{FF} \tag{5.24}$$

The area of the synchronizing can be calculated by the number of flip-flops and the area of a flip-flop. The last contribution to the expression is the area of the self-calibration circuitry. The area of the calibration techniques mostly determined by the area of the RAM, which is used to store calibrating codes. Hence the area of the calibration circuitry can be reduced to the area of the RAM.

5.5. Delay Estimation

The delay estimation is straightforward. The delay of a pipeline with *NS* stage is *NS* times the clock. Actually the delay is determined by the first stage. The resolved bits in the first stage are delayed until the last stage gives its output. Thus the synchronization between stages can be obtained.

5.6. Speed Estimation

The speed limitation determined by the settling time of the amplifier. The same approach, which is presented in section 3.1, is used in generating the speed requirement. Since the MDAC has different phases the dominant phase should be calculated. Earlier studies [6], [38], [40] show that the residue amplification phase brings tightened constraints. Hence the speed is determined by the slew-rate of the amplifier.

5.7. Capacitor Sizing

Capacitor sizing is basically a low-power technique that reduces the power stage by stage by decreasing the capacitor values which increases the thermal noise [47-48]. So, the downsizing of the capacitors must be carefully processed. In [48] it is shown that the two

ways to reduce the power are the capacitor scaling and pipeline resolution scaling. The choice of capacitor scaling is implemented in proposed methodology.

The capacitor scaling can also be done as we apply resolution scaling. However previous work shows that this method actually does not affect the total power. With resolution scaling, the stage resolution of the input stages can be as much as five bits. Hence, the power is mainly consumed by these stages. The downsizing of the latter stages does not reduce the power significantly.

On the other hand, if identical stage resolutions are used, the capacitor scaling may decrease power considerably. Earlier studies show that global minimum for capacitor scaling factor is two [48]. This value is used in the methodology where capacitor scaling is requested.

5.8. Calibration and Correction

One of the advantages of the pipeline architecture is that it allows calibration and correction. The digital correction is used in order to relax the constraints, mainly on the SADC. The correction methods can be found in [14], [49]. Previous work proves the necessity of digital correction.

On the other hand, self-calibration methods are very efficient and should be used in pipeline architectures over 10 bit resolution [41], [50]. They correct the most common causes of nonlinearity, such as incorrect DAC levels and gain errors. A general mathematical description can be found in [39].

Calibration methods can process in analog or in digital domain. The advantage of analog calibration is it can process fast enough to handle the clock speed. Early digital calibration techniques offer accurate calibration, but slow down the operation. However, some new calibration free techniques are presented recently [51].

5.9. Methodology of Pipeline Converters

The methodology proposed in this dissertation, contains a module for the pipeline ADC's. The design complexity of a pipeline architecture is higher than the flash architecture. The pipeline architecture, as its name implies, has stages connected to each other in order to form a pipeline. These stages can differ from each other with their resolutions. The number of stages and the resolution of each stage create a huge design space to be explored. This design space should be limited.

After getting the first parameter, which is the *SNR*, the methodology asks for some other parameters. These are maximum power, maximum area, maximum resolution, maximum capacitor array size. The design space is limited mainly by the maximum resolution and the minimum resolution calculated from the given *SNR*. The next step is to generate all available design configurations. Each configuration has different stage resolutions. The design space created by this process should be limited or reduced. Otherwise, reasonable CPU times cannot be achieved. The methodology uses some heuristic in order to limit design space. The methodology rejects the configurations, which have low resolutions at the early stages and high resolutions at the latter stages, because low stage resolutions contribute higher thermal noise and the effect of noise at the early stages is crucial. The input referred noise model Equation (5.1) shows that the noise at the early stages is dominant. Therefore, the early stages should have higher, at least equal, stage resolutions.

After the design space is explored and limited by the possible stage resolutions, the thermal noise is calculated and checked by using Equation (5.8) and Equation (5.11). This check may reduce the number of solutions, especially for low reference voltages.

The next step in the methodology is to calculate the required accuracy of each MDAC. Stage resolutions and the maximum matching between the capacitors determine this accuracy. Stage by stage this accuracy requirement decreases. Hence, the limitation for the residue amplifier loosens. These values are calculated by the methodology for each design and each stage. The result gives the number of stages that requires calibration. Also the accuracy limits determines the minimum gain by using Equation (5.12). The

methodology then searches its database in order to check that, if the gain requirements are met. Actually, this step and the following steps basically try to find a library element that does not violate the limits. After this database search, the methodology estimates the power, the area and the speed.

Finally, the values calculated are passed to a cost function in order to find the optimum solution. Also, the other results are written to a file for further inspection.

6. SIGMA-DELTA ADC

Sigma-delta converters are good alternatives to pipeline converters resolutions. They are very suitable to achieve resolutions up to 20 bits. They exchange the loss of accuracy inherent to analog circuits in digital processes for faster signal processing and more digital circuitry. In other words, more accurate ADC's can be achieved by complex and fast digital circuitry.

Many papers and books deal with the theory, design and simulation of Delta-Sigma converters. One of them [52] is a collection of key references published in specialized journals. Also the book [28] contains sections devoted to noise shapers and Delta-Sigma modulators.

A major problem of high-resolution converters is that thermal noise becomes the dominant noise source. Any increase in resolution causes an exponential increase in power consumption. For example, a reduction of six decibels of the thermal noise, which also means an extra bit at resolution, increases the capacitor size by a factor of four. Hence the power consumption exceeds the reasonable power limits at high resolutions.

Once the resolution exceeds six bits, the quantization noise appears increasingly random. Hence, Sigma-Delta converters use stochastic processing to perform conversion [13-14], [49].

The Delta-Sigma converter uses oversampling and noise shaping techniques to lower the quantization noise and consequently improve the resolution. Oversampling occurs whenever a signal is being sampled at a frequency larger than twice its bandwidth, the so-called baseband f_o. According to the Nyquist theorem, the higher sampling rate does not add information to the sampled signal. It is not only inefficient but also wrong since the spectrum is unnecessarily widened. Yet, increasing the bandwidth obtains positive results. Considering a quantized signal, the quantization noise power remains unchanged regardless of the sampling rate therefore, the noise power density must go down as the spectrum widens. If we restrict the bandwidth of the oversampled signal to the baseband no

information will be lost, but a lower noise power will be collected in the baseband. The bandwidth of the quantization noise spectral density encompasses the baseband f_o exactly. For example, if the sampling frequency is multiplied by four, the bandwidth of the quantization noise density is multiplied by four, whereas the magnitude is divided by four. If the bandwidth of the oversampled signal is restricted to the baseband, the quantization noise power in the baseband is four times smaller. This leads to a quantization noise:

$$e_0 = e_{org}\sqrt{\frac{2f_o}{f_s}} \tag{6.1}$$

Since the noise power in the baseband is divided by four, the *SNR* improves by 6 dB, which is the same as adding one bit to the quantized signal. More than one bit is also feasible, but the oversampling ratio rapidly becomes very large. An *SNR* improvement of 10 bits implies that the sampling frequency be one million times larger than the Nyquist frequency. This is too much unless the signal baseband does not exceed 100 Hz. To improve the *SNR* without compromising the bandwidth, the benefit obtained from oversampling is supplemented by a filtering operation that shifts part of the noise to high frequency, leaving less noise in the baseband. This is called noise shaping.

Noise shaping is achieved by putting a feedback loop around the quantizer [13], [49]. The loop controls the quantizer input in such a way that the output tracks the input signal as closely as possible. The role of the low-pass high gain filter controlling the quantizer is to minimize the differences between the input signal and its quantized one. Since the quantized signal is oversampled, the output looks like high frequency digital noise superposed on the input signal. This increases the high frequency noise of the quantized signal, hence lowers the low frequency noise, that is, the noise remaining in the baseband.

Another essential function considering the architecture is the low-pass filtering of the quantized signal in order to gather the benefits of oversampling. This is done using a filter, which ideally cuts everything above f_o and passes everything below. Therefore, a high order low-pass filter is needed, which would be very costly if it is implemented in the analog domain. However, this is not the case because the noise shaper delivers coded data

and the filtering takes place in the digital domain thus where sharp cut-off characteristics can be achieved.

The combination of the noise shaper and low-pass digital filter proves that Delta-Sigma converters take the best of analog and digital worlds. The noise shaper is a fast, relatively inaccurate analog circuit and the filter is a high precision digital signal processor. Therefore, Sigma-Delta converters are good candidates for mixed analog digital circuits, in a world that is dominated by digital technologies.

However, the processing in both digital and analog domain makes it difficult to model the architecture. The effects of filtering in the digital domain have to taken into account. Hence, previous work focused on simulation type behavioral modeling instead of constraint type modeling.

Sigma-Delta converters have different architectures. The choice of the order of the filter and the number of stages to be cascaded enlarges the solution space. A clever method has to developed in order to select the optimum solution.

The modeling of Sigma-Delta converters is left for further work.

7. DESIGN EXAMPLES

In this section some design examples are presented. The methodology can accept resolution values up to 19 bits. Up to this resolution the flash and the pipeline architectures should find a solution.

The design example is limited by 13 bits. The desired *SNR* is 75dB. This ratio determines the minimum resolution by using the expression Equation (4.12). The technology is 0.7μm mietec technology. The matching between capacitors is six bits. The conversion range is selected as 20MS/s. A reference voltage of five volts used for design example.

The number of available configurations is 33 for 12-bit design and 37 for 13-bit design. Some of the configurations are given at Table 7.1.

Table 7.1. Different configurations for 12 and 13 bits

Resolution	Configuration	Number of amplifier	Number of comparator	Stages that need calibration
12	22222222222	11	33	6
12	332222222	9	35	4
12	3333222	7	37	3
12	43222222	8	40	3
12	44332	5	47	2
12	54322	5	59	2
13	222222222222	12	36	6
13	333222222	9	39	3
13	3333322	7	41	3
13	443322	6	50	2
13	5443	4	68	2

In Table 7.1, some of the configurations are presented. The configuration column shows the stage resolutions from left to right. The right most number shows the resolution

of the last stage. The sum of the stage resolutions exceeds the desired resolution because of redundancy used. The methodology uses one extra bit for each stage. Thus, digital correction can be performed in each stage.

As we can see from Table 7.1, different configurations for stage resolutions may lead to very different designs. The power and area may vary significantly. Hence, the optimum stage resolution should be calculated for optimum stage resolution. The methodology calculates the power and the area values for each configuration. The optimum stage resolution can be determined from the graph.

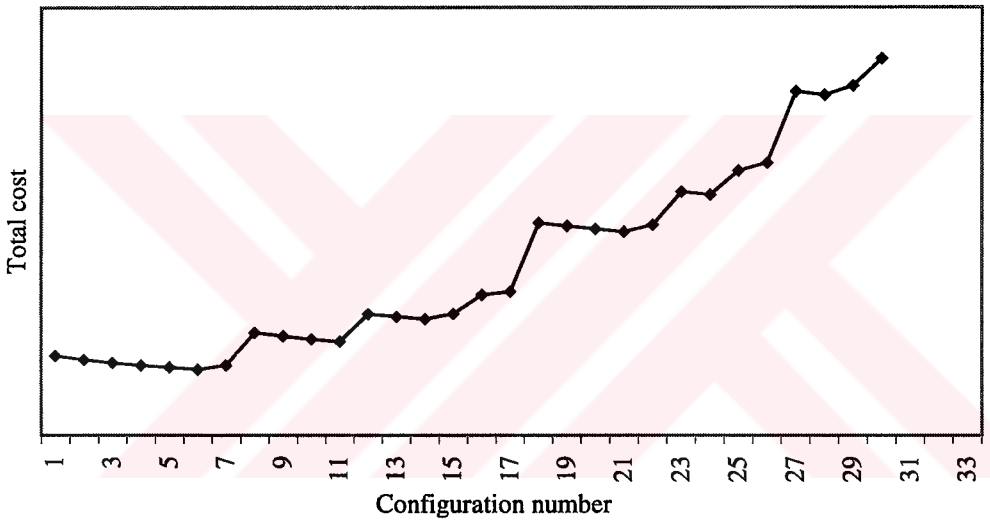

Figure 7.1. Optimum stage resolution

The methodology selects the configuration "3333322". Earlier studies about optimum stage resolution [37] conclude that minimum stage resolution is optimum for resolutions up to 10 bits. However, the recent studies show that above 10-bit resolution, medium resolutions gives better results [38]. The results obtained by the proposed methodology give similar results. This is because high resolutions at the input stages require large number of comparators. The exponential grow of comparator number increases the area significantly. That's why configurations like "54322 "or "5532 " are far from global minimum. On the other hand, low resolution per stage leads to large number of stages. Hence, exponential growth can be observed at the synchronization circuitry. Also number of stages that requires calibration increases. As a conclusion, optimum solution can be achieved by medium number of stages with medium resolution.

Better power related cost calculations could be achieved if the current of the amplifier can be calculated. However the circuit level models has to be derived in this case. Further work about circuit level modeling is needed for more accurate calculations.

Although the optimum number of stages may be determined, the other constraints have to verified. These constraints are generated by thermal noise and the error caused by the limited gain of the residue amplifier. Table 7.2 shows the required values for these errors.

Table 7.2. The constraints for the resolutions

Resolution	Configuration	Minimum amplifier gain For first three stages	Thermal noise
12	22222222222	78 dB, 72 dB, 66 dB	2.609e-05
12	332222222	78 dB, 66 dB, 54 dB	2.320e-05
12	43222222	78 dB, 60 dB, 42 dB	2.211e-05
12	54322	78 dB, 54 dB, 36 dB	2.157e-05
13	222222222222	84 dB, 72 dB, 66 dB	2.609e-05
13	333222222	84 dB, 66 dB, 54 dB	2.319e-05
13	443322	84 dB, 60 dB, 42 dB	2.209e-05
13	5443	84 dB, 54 dB, 36 dB	2.157e-05

Table 7.2. shows that the gain requirements for the amplifier decreases through the pipeline. Therefore, the amplifier selected from the library for the first stage may be unsuitable for the later stages. The cost function should take this property into account. This feature was built into the methodology. However, this feature totally depends on the available library elements.

Another feature of the methodology is capacitor scaling. The capacitor scaling results reduces the power dissipated stage by stage. On the other hand, the thermal noise increases with capacitor scaling. However, since the noise generated is reduced by a factor which is the total gain contributed by the early stages, the total thermal noise can be limited below the quantization noise.

Table 7.3. compares the total thermal noise values for different configurations. In the example, the noise stays below the limits determined by quantization noise. This is mostly because of the reference voltage and the resolution. Resolutions about 15 bits may decrease the boundaries significantly. In that case, the total thermal noise constraint may reject most of the configurations.

Table 7.3. Effect of capacitor sizing on thermal noise .

Resolution	Configuration	Total thermal noise with scaling	Total thermal noise without scaling
12	22222222222	3.167e-05	2.609e-05
12	332222222	5.179e-05	2.320e-05
12	43222222	5.092e-05	2.211e-05
12	54322	5.028e-05	2.157e-05
13	222222222222	3.167e-05	2.609e-05
13	333222222	5.167e-05	2.319e-05
13	443322	5.051e-05	2.209e-05
13	5443	5.028e-05	2.157e-05
Quantization noise		3.860e-04	3.860e-04

As it can be seen from the table, the quantization noise does not limit the number of realizable configurations. However, when resolutions of 15 bits are desired, the quantization noise forces us to use configurations with minimum stage resolution.

As we decrease the desired *SNR*, lower resolutions may be more advantageous. In that case the topology can be flash or pipeline. In this case the pipeline ADC gives better results considering the power and the area. However, pipeline architecture cannot compete with the delay and speed advantage of the flash. If high sampling rates are desired, than the pipeline architecture requires very large slew rate values. These high values can be obtained for a comparator but they may become very costly for an amplifier (generally an OTA in pipeline).

The flash architecture and the pipeline architecture are compared in Table 7.4. The number of comparators itself may be sufficient to show the high area and power ratios. On

the other hand, if high speed and minimum delay is a must, the flash architecture promises better performance.

Table 7.4. Comparison of architectures for eight bits

Type	Configuration	Number of amplifier	Number of comparator	Stages that need calibration	Delay (clockcycle)
8-bit pipeline	2222222	7	21	2	7
8-bit pipeline	3332	4	23	1	4
8-bit pipeline	442	3	33	1	3
8-bit flash	-	-	255	-	1

In this design, the methodology can find a solution from the design space created by the library elements. However, this is not the case every time. If a solution could not be found by searching the library elements, an input sweep operation can be executed.

The second design requires a *SNR* of 38dB. This leads us the designs with resolutions more than six bits. Since, the methodology limits the flash architecture by eight bits, the available resolutions are six, seven and eight bits. The desired sampling frequency selected above 50MS/s. Therefore, the methodology cannot find a solution by using library elements. Then the methodology selects another way to find a solution; parameter sweep. For the input parameters a sweep operation can be executed. The user also may exit from the methodology at this level.

Each parameter, which will be used for the sweep process, introduces another dimension. Hence, some parameters are kept constant to reduce the time spent by the methodology. In this example the sampling rate was kept constant. This method is also useful for observing the effect of the parameters on different designs.

Figure 7.2. shows the relation between the power consumed by the resistor string and the reference voltage.

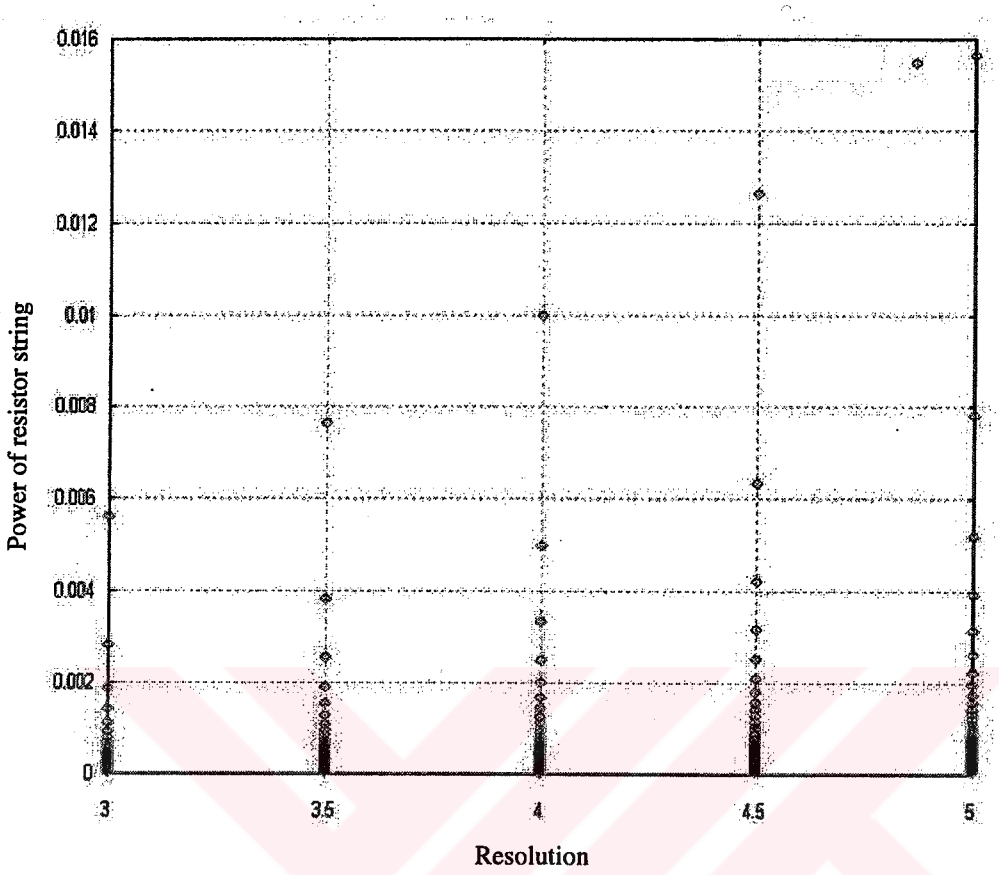

Figure 7.2. Relation between resistor power and the reference voltage

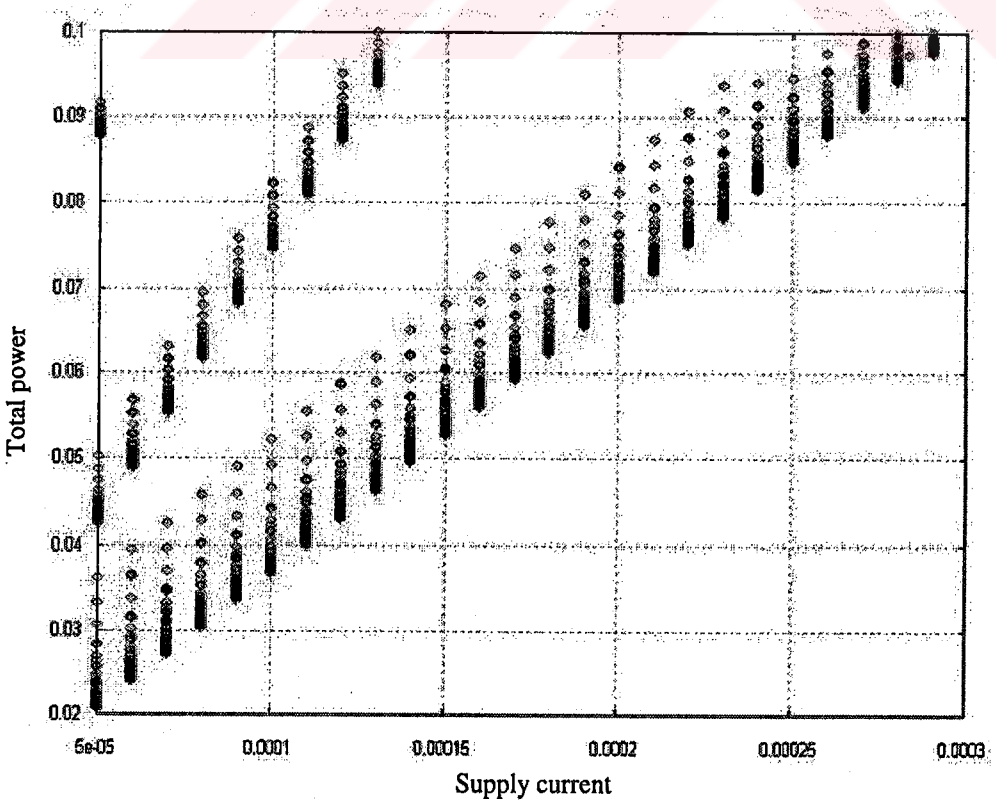

Figure 7.3. Relation between the total power and the supply current

We may see the effect of resolution in Figure 7.3. The three lines with different slopes represent the effect of resolution. The bottom line is the realizable designs with six bits. The line above six bits shows the designs with seven bits. The upper one, which has only one solution considering the supply current, is the design with eight bits. We may conclude that the currents larger than the one shown in the Figure, results with a power consumption above limits. The plot also shows the power consumption of the resistor string. If the trend line is subtracted from the graph, Figure 7.3. is the same as Figure 7.2.

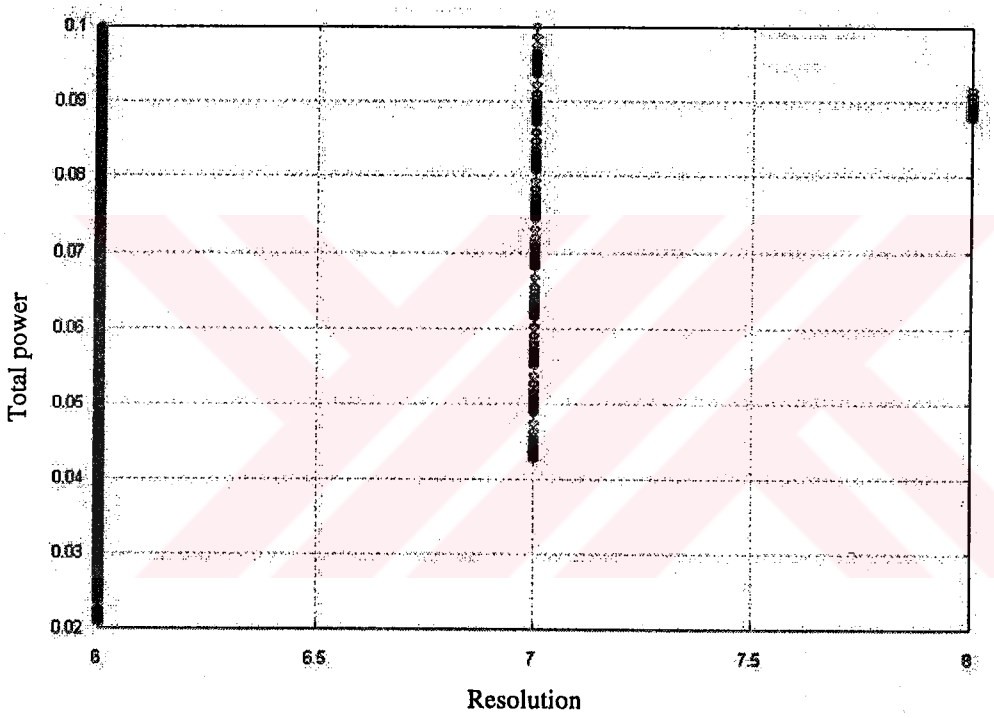

Figure 7.4. Relation between power and resolution

Figure 7.4 summarizes the relation of power and resolution. The graph shows that, low resolutions lead to a large number of solutions. With the reduction of the resolution, the number of solutions decreases. The eight bits resolution has limited solutions. Since only one value of supply current gives reachable solutions, the gap over the line occurs.

In this section two different examples were presented, one with pipeline architecture and one with the flash topology. Also, two different methods used by the methodology are shown.

8. CONCLUSION

In this dissertation, a methodology for ADC design was developed. The methodology operates at system level and generates the limit specifications for lower level blocks. The main purpose of such system is to guide the designer through a design process. Also, the developed methodology is crucial for an automated ADC design. This methodology is the first step in the design process. The circuit synthesizers and optimizers may be merged with this methodology in order to form an automated ADC design system.

The methodology is pretty fast if a solution can be found by using blocks available to the methodology. But the presented approach is not limited by the library. Setting the upper and lower limits of the parameters can widen the design space. In this case, the speed of the methodology is limited by the increments given by user. Also, the implemented architectures limit the performance of this methodology. In order to cover a wide range in the resolution vs. conversion range graph, Sigma-Delta architecture should be added.

The methodology uses functions, which calculates the possible errors. By using these error values, design constraints are determined. The next step is to search the database in order to find the set of solutions. If a solution cannot be achieved, a sweep operation is performed. The user specifies the constraints for the parameters. In other words, the design space to be explored is determined. Then, the parameters are swept within the given limits. In order to achieve better performance, each step in the evaluation cycle rejects the designs that can't be implemented and adjust the new boundaries for the following steps.

The methodology developed has two topologies available: flash, pipeline. Some heuristic rules are applied to the methodology, which performs a coarse architecture selection. The most important rule is the limitation of resolutions. The flash architecture is limited by eight bits. However, for some resolutions, the procedure presented above may be performed for each topology. Various solutions, especially when using sweep operation, are achieved. To select the most promising one, a cost function is implemented. As a result, the methodology generates an optimum solution considering area, power and speed, for each topology.

The presented design examples show that the solutions generated are consistent with the previous studies. The speed of the methodology needs further improvement. A generic algorithm that generates the configurations by using heuristic methods will increase the performance significantly.

REFERENCES

1. Tsukamoto, S., W. G. Schofield and T. Endo, "A CMOS 6-b, 400-MSample/s ADC with Error Correction", *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1939-1947, December 1998.

2. Ingino, J. M. and B. A. Wooley, "A Continuously Calibrated 12-b,10-MS/s, 3.3-V A/D Converter", *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1920-1931, December 1998.

3. Medeiro, F., B. Perez-Verdu, J. M. de la Rosa and A. Rodriguez-Vazquez, "Fourth-Order Cascade SC ΣΔ Modulators: A Comparative Study", *IEEE Transactions on Circuits and Systems I,* vol. 45, no. 10, pp. 1041-1051, October 1998.

4. Promitzer, G., "12-bit Low-Power Fully Differential Switched Capacitor Noncalibrating Successive Approximation ADC with 1 MS/s", *IEEE Journal of Solid-State Circuits*, vol. 36 no. 7, pp. 1138-1143, July 2001.

5. Erdogan, O. E., P. J. Hurst and S. H. Lewis, "A 12-b Digital-Background-Calibrated Algorithmic ADC with 90-dB THD", *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1812 -1820, December 1999.

6. Gielen, G. E. and J. E. Franca, "CAD Tools for Data Converter Design: An Overview", *IEEE Transactions on Circuits and Systems II,* vol. 43, no. 2, February 1996.

7. Ismail, M. and T. Fiez, *Analog VLSI, Signal and Information Processing*, McGraw-Hill, Singapore, 1994.

8. Erten, G., *High Level Synthesis of Analog Integrated Circuits using Macromodels and Equation Based Optimization Tools*, M.S. Thesis, Boğaziçi University, 1999.

9. Alpaydın, G., *A New Approach to Analog Integrated Circuit Optimization*, Ph.D. Thesis, Boğaziçi University, 2000.

10. Şimşek, A., *Analog Layout Generation for Silicon Compilation*, M.S. Thesis, Boğaziçi University, 1997.

11. Başkaya, F., *Performance Estimation in Analog Computer Aided Design*, M.S. Thesis, Boğaziçi University, 2000.

12. Chang, H., E. Liu, R. Neff, E. Felt, E. Malavasi, E. Charbon, A. Sangiovanni-Vincentelli, and P. Gray, "Top-Down, Constraint-Driven Design Methodology Based Generation of n-bit Interpolative Current Source D/A Converters," in *Proceedings of the Custom Integrated Circuits Conference,* pp. 369-372, 1994.

13. Medeiro, F., B. Perez-Verdu and A. Rodriguez-Vazquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, Netherlands, 1999.

14. Hoeschele, D. F., *Analog-to-Digital Digital-to-Analog Conversion Techniques,* J. Wiley, New York, 1994.

15. Horta, N., J. Franca, and C. Leme, "Framework for Architecture Synthesis of Data Conversion Systems Employing Binary-Weighted Capacitor Arrays", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 1789-1792, 1991.

16. Casinovi, G. and A. Sangiovanni-Vincentelli, "A Macromodeling Algorithm for Analog Circuits", *IEEE Transactions on Computer-Aided Design,* vol. 10, pp. 150-160, February 1991.

17. Brauns, G. T., R. J. Bishop, M. B. Steer, J. J. Paulos and S. H. Ardalan, "Table-based modeling of Δ-Σ modulators using ZSIM", *IEEE Transactions on Computer-Aided Design,* vol. 9, no. 2, pp. 142-150, February 1990.

18. Wooley, B., *MIDAS: Mixed Digital/Analog Sampled-Data Simulator,* 2000, http://www-cis.stanford.edu/icl/wooley-grp/midas.html

19. Liberali, V., V. Dias, M. Ciapponi and F. Maloberti, "TOSCA: A Simulator for Switch-Capacitor Noise-Sharping A/D Converters," *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 2677-2680, 1991.

20. Jusuf, G., P. Gray and A. Sangiovanni-Vincentelli, "CADICS—Cyclic Analog-to-Digital Converter Synthesis", pp. 286-289, 1990.

21. Peralias, E., A. J. Acosta, A. Rueda and J. L. Huertas, "VHDL-based Behavioral Description of Pipeline ADC's", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 681-686, 2000.

22. Degrauwe, M. G., O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. L. A. Goffart, E. A. Vittoz, S. Cserveny, C. Meixenberger, G. V. Stappen and H. J. Oguey, "IDAC: An Interactive Design Tool for Analog CMOS Circuits," *IEEE Journal of Solid-State Circuits,* vol. SC-22, pp. 1106-1116, December 1987.

23. Jusuf, G., P. Gray, and A, Sangiovanni-Vincentelli, "A Performance-Driven Analog-to-Digital Converter Module Generator", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 2160-2163, 1992.

24. Sabiro, S., P. Semi, and M. Tawfik, "HiFADiCC: A Prototype Framework of A Highly Flexible Analog to Digital Converters Silicon Compiler", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 1114-1117, 1990.

25. Vital, J. and J. Franca, "Synthesis of High-Speed A/D Converter Architectures with Flexible Functional Simulation Capabilities", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 2156-2159, 1992.

26. Harjani, R., R. A. Rutenbar and L. R. Carley, "*OASYS: A Framework for Analog Circuit Synthesis*", *IEEE Transactions on Computer-Aided Design,* vol. 8, no. 12, pp. 1247-1266, December 1989.

27. Nuno, C., N. Horta, and J. Franca, "Algorithm-Driven Synthesis of Data Conversion Architectures", *IEEE Transactions on Computer-Aided Design,* vol. 16, no. 10, pp. 1116-1135, October 1997.

28. Plassche, R. J., *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, Boston, 1994.

29. Song, B. S., S. H. Lee and M. F. Tompsett, "A 10 bit 15 MHz CMOS Recycling Two-Step A/D Converter", *IEEE Journal of Solid-State Circuits,* vol. 25, no. 6, pp. 1328-1338, December 1990.

30. Lewis, S. H. and P. R. Gray, "A Pipelined 5 Msample/s 9-bit Analog-to-Digital Converter", *IEEE Journal of Solid-State Circuits,* vol. SC-22, no. 6, pp. 954-961, December 1987.

31. Krishnaswamy, N., H. S. Fetterman, J. Anidjar, S. H. Lewis and R. G. Renninger, "A 250-mW, 8-b, 52 Msamples/s Parallel-Pipelined A/D Converter with Reduced Number of Amplifiers", *IEEE Journal of Solid-State Circuits,* vol. 32, no. 3, pp. 312-320, March 1997.

32. Kazuya, S., Y. Nishida and N. A. Nakadai, "10-b 100 Msample/s Pipelined Subranging BICMOS ADC", *IEEE Journal of Solid-State Circuits,* vol. 28, no. 12, pp. 1180-1186, December 1993.

33. Kim, K.Y., N. Kusanayagi and A. A. Abidi, "A 10-b, 100-MS/s CMOS A/D Converter", *IEEE Journal of Solid-State Circuits,* vol. 32, no. 3, pp. 302-311, March 1997.

34. Sumanen, L., M. Waltari and K. A. I. Halonen, "A 10-bit 200-MS/s CMOS Parallel Pipeline A/D Converter", *IEEE Journal of Solid-State Circuits,* vol. 36, no. 7, pp. 1048-1055, July 2001.

35. Yotsuyanagi, M., E. Toshiyuki and H. Kazumi, "A 10-b 50-MHz Pipelined CMOS A/D Converter with S/H", *IEEE Journal of Solid-State Circuits,* vol. 28, no. 3, pp. 292-300, March 1993.

36. Pan, H., M. Segami, M. Choi, J. Cao, and A. A. Abidi, "A 3.3-V 12-b 50-MS/s A/D Converter in 0.6-μm CMOS with over 80-dB SFDR", *IEEE Journal of Solid-State Circuits,* vol. 35, no. 12, pp. 1769-1780, December 2000.

37. Lewis, S. H., "Optimizing the Stage Resolution in Pipelined, Multistage, Analog-to-Digital Converters for Video-Rate Applications", *IEEE Transactions on Circuits Systems. II,* vol. 39, pp. 516-523, August 1992.

38. Goes, J., J. C. Vital and J. E. France, "Systematic Design for Optimization of High-Speed Self-Calibrated Pipeline A/D Converters", *IEEE Transactions on Circuits Systems. II,* vol. 45, no. 12, pp. 1513-1525, December 1998.

39. Soenen, E. G. and R. L. Geiger, "An Architecture and an Algorithm for Fully Digital Correction of Monolithic Pipelined ADC's", *IEEE Transactions on Circuits Systems. II,* vol. 42, no. 3, pp. 143-153, March 1993.

40. Goes, J., C. Vital and J. E. Franca, *Optimum Resolution-per-Stage in High-Speed Pipelined A/D Converters Using Self-Calibration, 1995,* http://www.dee.fct.unl.pt/electronica/OptimumResolution.doc.

41. Lin, Y. M., B. Kim, and P. R. Gray, "A 13-b 2.5-MHz Self-Calibrated Pipelined A/D Converter in 3-μm CMOS", *IEEE Journal of Solid-State Circuits,* vol. 26, no. 4, pp. 628-636, April 1991.

42. Lewis, S. H., H. S. Fetterman, G. F. Gross, R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s Analog-to-Digital Converter", *IEEE Journal of Solid-State Circuits,* vol. 27, no. 3, pp. 351-358, March 1992.

43. McNutt, M. J., S. LeMarques and L. Dunkley, "Systematic Capacitance Matching Errors and Corrective Layout Procedures", *IEEE Journal of Solid-State Circuits,* vol. 28, no. 5, pp. 611-616, May 1994.

44. McCreary, J. L., "Matching Properties, and Voltage and Temperature Dependence of MOS Capacitors", *IEEE Journal of Solid-State Circuits,* vol. SC-16, no. 6, pp. 608-616, December 1981.

45. Shyu, J., G. C. Temes and F. Krummenacher, "Random Error Effects in Matched MOS Capacitors and Current Sources", *IEEE Journal of Solid-State Circuits,* vol. SC-19, no. 6, pp. 948-955, December 1984.

46. Goes, J., C. Vital and J. E. Franca, "A CMOS 4-bit MDAC with Self-Calibrated 14-bit Linearity for High-Resolution Pipelined A/D Converters", *IEEE Custom Integrated Circuits Conference*, May 1996, San Diego, U.S.A, pp. 105-108, 1996.

47. Plassche, R. J., W. M. C. Sansen and J. H. Huijsing, *Analog Circuit Design*, Kluwer Academic Publishers, Boston, 1995.

48. Kwok, P. T. F. and H. C. Luong, "Power Optimization for Pipeline Analog-to-Digital Converters", *IEEE Transactions on Circuits Systems. II*, vol. 46, no. 5, pp. 549-553, May 1999.

49. Gray, P. and A. Jespers, *Integrated Converters,* Oxford University Press, New York, 2000.

50. Lee, S. H. and B. S.Song, "Digital-Domain Calibration of Multistep Analog-to-Digital Converters", *IEEE Journal of Solid-State Circuits,* vol. 27, no. 12, pp. 1679-1688, December 1992.

51. You, S., K. Lee, H. C. Choi, H. Park, J. Kim and P. Chung, "A 3.3V 14-Bit 10MSPS Calibration-Free CMOS Pipelined A/D Converter", *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 435-439, 2000.

52. Candy, J. C. and G. C. Temes. *Oversampling Delta-Sigma Data Converters*, IEEE Press, Piscataway, NJ, 1992.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ