University of Alberta

On the Design and Testability of Analog Decoder Interfaces

by

Keith David Boyle ©

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2007

Your file Votre référence
ISBN: 978-0-494-29940-1
Our file Notre référence
ISBN: 978-0-494-29940-1

To Tianye

Abstract

Analog decoders are a class of soft-decision decoders that use probabilistic calculations to converge to a solution. Previous work has demonstrated the concept of analog decoders, including working prototypes. This thesis extends the physical design techniques of analog decoders by focusing on the interface with the remainder of the communication system, introducing a design methodology for and analyzing the testability of this interface.

This thesis analyzes the sample and hold and comparator circuits, and introduces analog test circuits for each one. A system-level testing circuit to allow observability of system variables is also introduced. The design methodology for this interface is demonstrated for an analog Fast-Fourier Transform chip. A portion of this interface is demonstrated on a chip implemented through CMC Microsystems.

Acknowledgements

I am grateful to my co-supervisors, Dr. Vincent Gaudet and Dr. Christian Schlegel, for their support and advice throughout my program. I would also like to thank Dr. Chris Winstead of Utah State University for hosting me as a short-term visiting student and for assisting in the design of both the system and the chip featured in this thesis. Finally, a thank you to Dr. Bruce Cockburn and Dr. Kris Iniewski for their assistance in publishing conference papers relating to this thesis.

Also a special thank you to my fiancée, Tianye Li, for spending so many late nights with me as I worked on this project.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

In Alphabetical Order

AD	Analog Decoder
ADC	Analog to Digital Converter
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BiCMOS	Bipolar-CMOS
BIST	Built-In Self Test
BJT	Bipolar Junction Transistor
BTC	Block Turbo Code
CLK	Clock
CMC	Canadian Microelectronics Corporation
CMOS	Complementary Metal-Oxide Semiconductor
CMRR	Common-Mode Rejection Ratio
CUT	Circuit Under Test
D/A	Digital to Analog
DAC	Digital to Analog Converter
DC	Direct Current
DFT	Design For Testability
DFT	Discrete Fourier Transform
DRC	Design Rule Check
FFT	Fast-Fourier Transform
GDS	Gerber Data Stream
HDL	Hardware Description Language
IC	Integrated Circuit
IDFT	Inverse Discrete-Fourier Transform
IFFT	Inverse Fast-Fourier Transform
LDPC	Low-Density Parity Check
LLR	Log-Likelihood Ratio

MiM	Metal-Insulator-Metal
MOSIS	Metal Oxide Semiconductor Implementation Service
MUX	Multiplexer
NDA	Non-Disclosure Agreement
NMOS	N-Channel Metal-Oxide Semiconductor
OFDM	Orthogonal Frequency-Division Multiplexing
PAR	Place and Route
PCB	Printed Circuit Board
pdf	Probability Density Function
PI	Primary Input
PLL	Phase Locked Loop
PMOS	P-Channel Metal-Oxide Semiconductor
PN	P-doped to N-Doped
PO	Primary Output
P-S	Parallel to Serial
RC	Resistive-Capacitive
RST	Reset
SH	Sample and Hold
SOR	Successive Over Relaxation
S-P	Serial to Parallel
SPICE	Simulation Package with Integrated Circuit Emphasis
SR	Shift Register
TG	Transmission Gate
TSMC	Taiwan Semiconductor Manufacturing Company
USU	Utah State University
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
W/L	Width / Length
WF	Weighting Factor
WTA	Winner Take All

List of Symbols

k_n	Transistor current factor
C_{OX}	Transistor oxide capacitance
W	Transistor channel width
L	Transistor channel length
η	Process-specific mismatch factor
κ	Process-specific constant
ρ	Resistivity
$D(t)$	Discrete Fourier transform of a signal
d_n	Discrete Fourier transform coefficient
P_{LOSS}	Mismatch power loss; the additional transmit power needed to overcome the effects of mismatch in an analog decoder
W_F^N	Weighting factor (or twiddle factor) in the Cooley-Tukey FFT algorithm
N_0	Channel noise power
σ	Standard deviation of a Gaussian random variable
Q_C	Injected charge (model of charge injection)
N,n,k	Used to represent integers in various equations and diagrams
n_{PINS}	Number of Pins
n	Fraction Denominator (value found experimentally)
n_C	Number of Comparators
M	Number of DACS used in the distribution system
V	Voltage
V_{DD}	Supply voltage
V_{SS}	Return voltage; electrical ground (in this thesis)
V_{IN}	Input voltage
V_{OUT}	Output voltage

v_{gs}	Transistor gate to source voltage
v_{ds}	Transistor drain to source voltage
V_T	Transistor threshold voltage
V_C	Capacitor voltage
V_{COMP}	Comparator output voltage
V_{DAC}	DAC output voltage
V_{TH}	Comparator input threshold voltage
V_{WTA}	Winner-take-all stage output voltage
V_{HIGH}	Positive portion of a differential signal
V_{LOW}	Negative portion of a differential signal
V_{REF}	Reference voltage
U_T	Device thermal voltage

R	Resistance
R_{DS}	Resistance from drain to source
R_{DAC}	DAC Output Resistance
R_{TG}	Transmission gate on-resistance

C	Capacitance
C_{TOTAL}	Total capacitance
C_{GD}	Transistor gate-drain capacitance
C_{WIRE}	Metal – Substrate capacitance for a wire

I	Current
I_D	Transistor drain current
I_U	Unit current supply
I_0	Process-specific current approximation
I_{LEAK}	Leakage current

I_{TEST}	Test current
I_{OUT}	Cell output current
P	Power
A	Silicon die area
$A_{256-FFT}$	Area of a 256-bit FFT
$A_{256-BIT}$	Area of a 64-bit FFT
A_{C}	Capacitor area
$A_{COMPARATOR}$	Comparator area
$A_{REGISTER}$	Register area
A_{SH}	Sample and Hold area
A_{TOTAL}	Total area
t	Time
$\mathrm{t}_{\mathrm{STOR}}$	Storage Time (over which an SH holds a value)
t_{N}	One time frame, of a total consisting of N time frames
$\mathrm{t}_{\mathrm{LOAD}}$	Time to load one value into an SH unit
t_{SYS}	System Period (length of time for a system to run one complete cycle)
τ	RC time constant
f	Frequency
f_{CLK}	Clock frequency
f_{SYS}	System frequency ($=1/t_{SYS}$)

Chapter 1

Introduction

The main goal of any communication system is to deliver information in a reliable manner. The goal of communications engineers, at its most basic definition, is to increase the amount of information that can be moved across a channel, while reducing the power required to do so. One method one can use to achieve this goal is to improve the reliability of the data transmitted: this reduces the need to re-transmit data, or reduces the power at which it needs to be transmitted. Reliability can be improved using channel coding techniques, which are methods of adding redundancy to a message to improve the likelihood of error-free reception. For an introduction to channel coding see [1, 2].

Many popular channel coding techniques make use of iterative decoding at the receiver. Iterative decoders use a parallel network of soft probability calculation nodes, which exchange information with each other. Presented at the input with Log-Likelihood Ratios (LLRs), values which represent the probability a given symbol is 1 divided by the probability the same symbol is zero, each node uses extrinsic information from neighboring LLRs (based on the structure of the code) to increase confidence in the symbol, ideally converging to a confident '1' or a confident '0'. Through this exchange, the local information of each channel sample contributes to the convergence of the estimate of the transmitted message as a whole.

These decoders are typically implemented using digital CMOS logic, using several bits to represent each LLR. However, these soft probability calculations also naturally map to simple analog circuits. Using these circuits is a potential approach for implementing iterative decoders. Previous work has demonstrated the concept of analog decoders (ADs), where measurements from fabricated chips have proven that the decoder core offers power and area gains over their pure-digital counterparts (due to, among other factors, reduction in overhead wiring and removal of the clock circuit) [3].

However, the demonstrated analog decoders are essentially proof-of-concept, designed to demonstrate specific aspects of analog decoding. Having proven these concepts, the new requirement is to demonstrate that analog decoders are viable options for integration into full receiver chains. This requires demonstrating system-level design: in particular, they must have an effective interface, and they must be testable.

Previous analog decoders used codes with extremely small block lengths (typically (8,4) Hamming), as they were concerned more with the analog design than the effectiveness of the code. However, longer block lengths lead to more effective channel coding, and common commercial block lengths are frequently in the order of thousands of bits. To be a competitive technology, then, ADs must be able to process much longer block lengths than they currently do.

The input interface of an analog decoder typically consists of switched-capacitance sample and hold (SH) circuits. They do not perfectly store their data, instead introducing signal degradation through nonidealities such as leakage currents and charge injection errors. At the output is a bank of comparators, which act as analog-to-digital converters, producing digital values; the precision of these comparators has a direct effect on the output accuracy; if the comparator makes an incorrect decision, the decoder processing is rendered useless. A well-designed interface is necessary for the continued growth of analog decoders.

Testability refers to the ease with which one can verify that a fabricated chip is structurally correct and will perform as designed; typically, the duration in which any one chip must be verified is on the order of milliseconds. Digital testing is generally based on fault models, with stuck-at faults being the most basic example. Analog testing is generally parametric, wherein a known signal is produced at the input and the output is measured for certain characteristics.

Since functional errors can result not only from fabrication errors such as dust contamination, but also from statistical variations such as mismatch, analog testing is frequently difficult and analog test circuits tend to be large. The current approach to testing analog decoders is to consider them as three sub-circuits: the input interface, the core, and the output interface. In 2005 researchers at the University of Alberta developed and demonstrated a Built-In Self Test (BIST) system to test the core circuits [4]. The BIST also ran simple tests on the interfaces, but could only find the most basic faults and did not consider parametric variation. A more powerful test system is required before ADs can be considered to be truly testable.

This thesis seeks to extend the standard interface of analog decoders in two directions: to expand analog decoder interfaces to interfaces of an analog processing system, where the AD is combined with another analog system, and to provide test circuits for this case. These test circuits, when combined with the test circuits in [4], create a test system for an analog decoder. The 'other analog system' used for this thesis consists of an analog current-mode Fast-Fourier Transform (FFT) circuit, one of the processing elements involved in an Orthogonal Frequency-Division Multiplexing (OFDM) receiver. The design and characterization of the FFT, and discussion of the integration of an analog FFT with analog decoders, is documented in [5], completed in tandem with this thesis.

Here, this thesis examines a (256,121) FFT-AD combined system, and then demonstrates a design methodology for an interface for the combined system.

This thesis is organized as follows. Chapter 2 introduces background information on OFDM receivers, focusing on FFT processors, analog decoders, SH circuits, comparator circuits, and analog testing. It also introduces the overall block diagram of an FFT-AD system in Figure 2.7; the main goal of this thesis is to design an interface and test system for the system shown in this diagram. Chapter 3 introduces the proposed circuits and test circuits for each block, including the AD, the FFT, SH, and comparator circuits. Chapter 4 examines a specific design of the interface for a combined (256,121) FFT and analog decoder system. This uses the schematics introduced in Chapter 3, examining specific device sizing for the (256,121) application. Chapter 5 demonstrates the layout of a 64-bit FFT chip that was fabricated to demonstrate the concept of an analog FFT; it uses the input interface and FFT test circuits designed in Chapter 4. Finally, Chapter 6 provides the conclusions and future outlook offered by this thesis.

Chapter 2

Background Information

This chapter provides the background information on the systems used in the remainder of the thesis. Each section is intended to provide a survey of pertinent literature surrounding each topic, and provide suitable grounding for the decisions and proposals in subsequent chapters. The topics in this chapter are organized from the most general to the most specific: Section 2.1 covers communication systems, OFDM receivers, and FFT processors. Then we focus solely on analog decoders in Section 2.2. The input and output interfaces of ADs are a primary focus on this thesis, so they are each examined in Sections 2.3 and 2.4 respectively. Finally, as another goal of this thesis is to improve ADs' testability, analog testing is briefly reviewed in Section 2.5.

2.1. OFDM Communication Systems and FFT Processors

This section is intended to give an idea of an analog decoders' place in a larger communication system. The basic model of a point-to-point communication system is shown in Figure 2.1. The common steps taken before transmission include source coding (data compression), channel coding (addition of redundancy), and modulation (physical representation of data). In the channel itself, noise is added. This model, and this thesis, assumes Additive White Gaussian Noise (AWGN). In this model, the channel noise is a Gaussian variable of zero mean and standard deviation σ. At the receiver, each of these steps is 'undone' in order. This thesis is concerned with the receiver, particularly the channel decoding step.

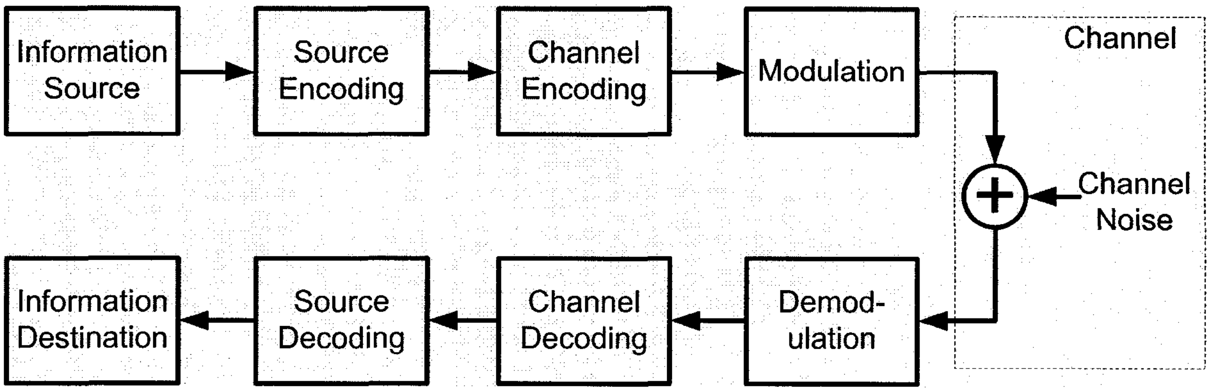


Figure 2.1: Basic Point-to-Point Communication System Model

In communication systems, channel coding is used to add redundancy to a signal to either increase the likelihood of proper reception for a given transmission power, or reduce the transmission power for a given likelihood of proper reception. The system at the receiver which uses this redundant information to check the quality of the received message is known as a channel decoder. The class of Analog Decoders is a subclass of 'soft-decision' iterative decoders that evaluate the probability that a given bit is a 1 or a 0. A review of iterative decoding is not offered here; for this the reader is encouraged to consult sources such as [2].

While it is academically convenient to consider the blocks in Figure 2.1 as self-contained systems, the truth is that design decisions in one block frequently affect the possible choices in other blocks. For this thesis, we will restrict consideration to an Orthogonal Frequency-Division Multiplexed (OFDM) system; this is introduced here.

Orthogonal Frequency-Division Multiplexing is a digital modulation scheme that is popular for wideband digital communication systems. As the name suggests, it makes use of Frequency-Division Multiplexing, which where the available frequency spectrum is divided into dozens or hundreds of narrowband subchannels, and each channel carries its own message. OFDM takes one more step by modulating the individual channels, of which there are usually dozens or hundreds, with orthogonal carriers. This offers considerable complexity improvements in a number of areas.

Since the modulated data streams are orthogonal, crosstalk between adjacent sub-channels is minimized, and guard tones between the sub-channels are unnecessary. Equalization at the receiver is simplified over schemes that require matched filtering; the data processing, to be discussed later, is easily implemented. This scheme also allows for easy adaptation to frequency-selective fading channels, and theoretically closely approximates the 'water-filling' technique for transmitting over such channels [6].

OFDM is widely considered to be one of the most important modulation schemes in the last 50 years; recently it has been included in wireless LAN standards such as IEEE 802.11a [75] and IEEE 802.11g (54 Mbps Wi-Fi [76]). It is also contained in the IEEE 802.16e (Wireless MAN) standard, on which the industrial standard Mobile WiMAX is based, and will be used in 4G cellular networks. Equally importantly, OFDM also receives a lot of attention for low-power networks, such as Wireless Sensor Networks or Smart Dust. This is important for this thesis, as these applications are those with the greatest applicability for analog decoders. It would not be unreasonable to expect that the first complete communication chain using analog decoders will be an OFDM system.

Consider, for a moment, how an FDM system might be implemented. The data to be transmitted is first coded, as in any other system, but is then split into N parallel channels in a Serial-to-Parallel (S-P) block. Each channel is then modulated by a separate filter and carrier and transmitted. At the receiver, matched filters each respond to their respective subchannels, and the resultant signal is reconstituted and decoded. If the carriers are chosen so that they are all mutually orthogonal, this can then be called an OFDM system. However, if these filters and carriers must be separately implemented, then this offers little hardware advantage over other, less complex systems. OFDM offers the greatest advantage if an efficient method can be found for generating all the orthogonal signals.

To do this, consider a frame of binary data, N bits long; this can be represented as a vector d_n. If an inverse discrete Fourier transform (IDFT) is applied, the result is a time-domain signal $D(t)$, described by

$$D(t) = IDFT(d_n) = \sum_{n=1}^{N} d_n e^{j(\frac{2\pi}{N})nt} . \tag{2.1}$$

The set of exponential signals

$$\left\{ e^{j(\frac{2\pi}{N})nt}, (n = 0, \pm 1, \pm 2, \ldots) \right\} \tag{2.2}$$

are mutually orthogonal (see a proof in [7]), so $D(t)$ is necessarily a set of orthogonal vectors as well. This is an exciting result: a set of orthogonal vectors can be easily generated, simply by performing an IDFT. At the receiver, this can be undone by performing a Discrete Fourier Transform (DFT), and the signal d_n is recovered. In practice, the IDFT and DFT are usually implemented using the inverse Fast-Fourier Transform (IFFT) and Fast-Fourier Transform (FFT).

The transmitter is shown in Figure 2.2. The IFFT block replaces a set of filters and modulators (one each would otherwise be required for each channel).

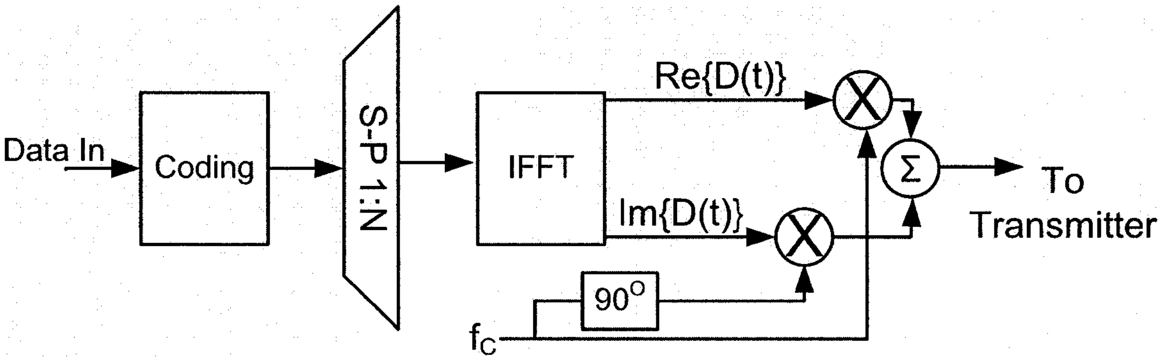


Figure 2.2: OFDM Transmitter Block Diagram

The receiver is shown in Figure 2.3. The incoming signal is demodulated, digitized, and then processed with an FFT to generate the signal values. These N values are then converted parallel to serial (P-S), decoded, and sent to the remainder of the communications chain.

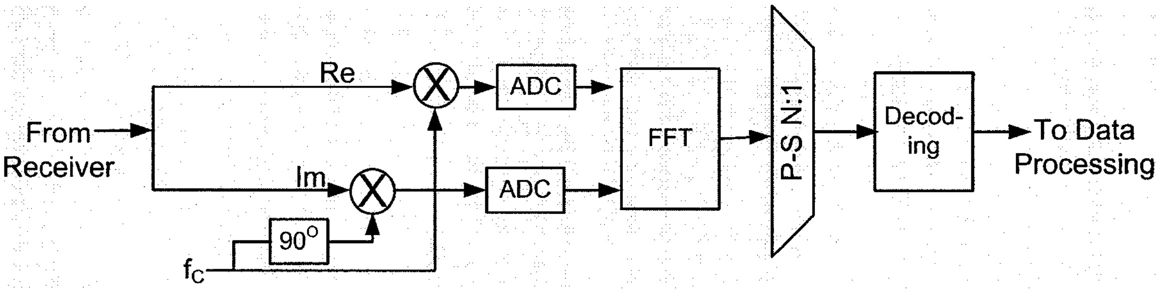
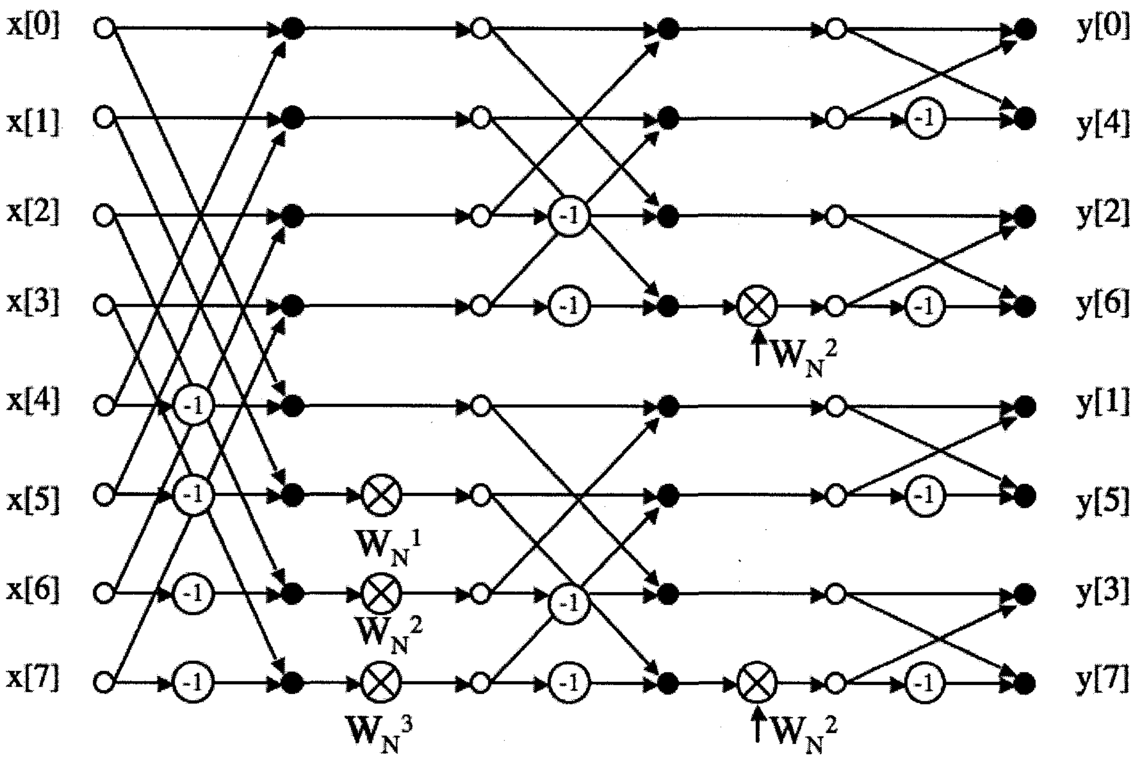


Figure 2.3: OFDM Receiver Block Diagram

As this thesis focuses on the receiver chain, the FFT block will be considered further. 'Fast Fourier Transform' describes algorithms to quickly find the Discrete Fourier Transform (DFT). The most popular method was introduced in 1965 by Cooley and Tukey [8]; it recursively breaks down the DFT into smaller DFTs which in turn consist of smaller DFTs. This allows for the size and complexity of DFTs to slowly grow as the size of the computation grows. A common method used to represent FFTs is the Butterfly Diagram, for which an 8-bit version is shown in Figure 2.4. Filled-in circles represent addition, and empty circles represent duplication of a value. The circles represent multiplication, either by 'twiddle factors' (also called 'weighting factors') or by -1.

Analog decoding is based on the observation that analog circuits can implement a factor graph as effectively as can a digital circuit. Implementation of an FFT processor is the same situation: given a representation of a data flow graph, the system can be implemented using analog circuits [10]. Each node is implemented using current mirrors, creating a full analog current-mode FFT. For this implementation, the system would look like that in Figure 2.5. In contrast to Figure 2.3, this receiver has no ADCs. Also, the data delivery from the FFT to the decoder happens in parallel; the parallel-to-serial conversion happens in the shift register at the output of the decoder.



Error! Figure 2.4: Eight-Bit FFT Connection Diagram (based on [9])

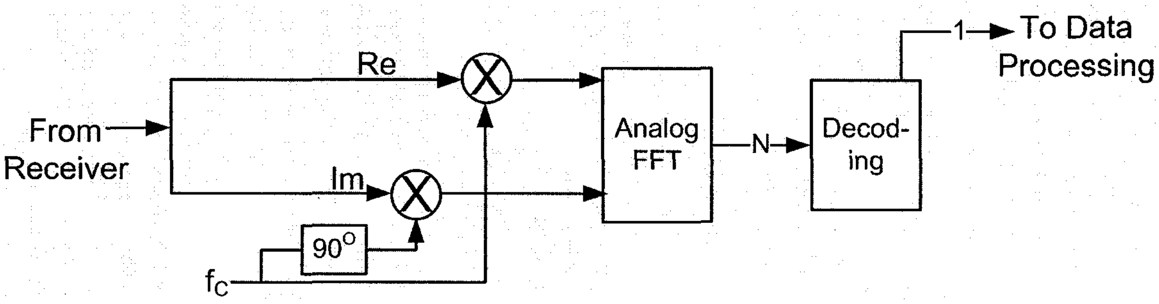


Figure 2.5: Analog OFDM Receiver System

The focus of this review will now shift to the ‘Decoding’ block as shown in Figure 2.5. The next section examines Analog Decoders, including their history and current research.

2.2. Analog Decoders

Previous to this section, ADs were treated only as blocks, without discussion of their functionality or design. This section offers such a discussion: it reviews the general concept of analog decoders, major research in the field to date, and design issues currently facing the field.

First, consider translinear circuits [11]. Translinear circuits make use of an idealized device with an exponential transconductance, or

$$I_D \propto e^{v_{gs}}. \tag{2.3}$$

Circuits have been developed that use this simple transfer function to calculate various mathematical and geometric equations [11, 12]; the most well-known of these is the Gilbert Multiplier [13]. Translinear circuits were originally implemented using Bipolar Junction Transistors (BJTs), but in as early as the 1970s, it was shown that the same idealized relationship could be made from CMOS transistors operating in their subthreshold region, when the gate-to-source voltage is less than the device's threshold voltage, or

$$|v_{gs}| < |V_T|. \tag{2.4}$$

As a result, complex calculations, such as channel decoding, can be implemented using only subthreshold CMOS transistors; because of the requirements of (2.4) they must be operated at an extremely low supply voltage and at very low current levels. This subthreshold operation has been the focus of much study in both digital and analog circles, as it allows for operation while consuming very low power. Analog decoders consisting of CMOS transistors operating in this region operate at a lower overall power consumption than their purely digital counterparts [14], though perhaps with speed limitations. For further analysis on the applicability and limitations of CMOS translinear circuits, see [11] or [14].

Analog decoders should not be viewed as a revolutionary idea unto themselves, but a natural consequence of work in iterative decoding methods and analog Viterbi decoding (see [15] for an example analog Viterbi decoding, or [16] for an early diode-based decoder for trellis codes). The theoretical analysis is identical to that of digital decoding methods, including use of Tanner graphs to describe the connections. The only system-level difference between analog and digital decoding is that the analog process is continuous (as opposed to being synchronous in typical digital applications), and the quantization error in the digital case is replaced by mismatch error in the analog case. Loeliger's 2001 paper [17] is widely used as an introductory presentation of analog decoders.

The first documentation of analog decoders is [18], a patent submitted by J. Hagenauer in Germany in 1997. In 1998, Hagenauer [19] and H.A. Loeliger et al [20] independently presented conference papers proposing the use of analog electronic networks to decode error correcting codes. In contrast to previous work on analog Viterbi decoders, the work both by Hagenauer and by Loeliger was inspired by turbo-style decoding of codes described by graphs. Large gains, in terms of speed or power consumption, over digital implementations were forecast.

The first working analog decoder to be published was presented in [3]. It introduced a current-mode BiCMOS decoder for memory channels; the chip consisted of discrete check and variable nodes, which were wired together across a PCB to create the circuit. This provided the essential first demonstration of the effectiveness of analog decoders. It is also the first paper to make the claim that, "A comparison [of their circuit] to an equivalent digital implementation exhibits more than two orders of magnitude less power and/or more speed", a statement often used in arguments supporting analog decoders.

The next analog decoder was presented in 2000 by Hagenauer's group [21]. This too was implemented in BiCMOS. At the time, the authors observed that "CMOS devices operating below threshold also show exponential behaviour, but it is difficult to ensure this operating mode for all devices in a circuit."

Hagenauer's proposed CMOS solution was first implemented by Winstead and Schlegel at the University of Utah, who implemented the first CMOS-only decoder (an (8,4) Hamming decoder) [22]. This can also be considered to be the first analog decoder system – it offered a full interface and core integrated on a single chip. CMOS circuits in subthreshold operation are appropriate for building translinear circuits as they have the requisite exponential transfer function [23]. While designed for operation in the weak-inversion region, this decoder continued to work in the strong inversion region, and with an allowed for a throughput of up to 20 Mbit/s.

Since 2001, dozens of chips have been produced, demonstrating analog decoding for Hamming [4, 22, 24], Turbo [25, 26] (including work on programmable interleavers in [27]), and LDPC [28] codes.

2.2.1. AD System-Level Simulation

Any system must be effectively simulated before it can be manufactured; however, simulating large-scale analog systems, such as analog decoders, is extremely computationally intensive.

The digital iterative decoders that analog decoders are based on can easily be simulated in any software language, such as C, VHDL, or Verilog HDL. These can be then extended, using simple analytical analog models, for system-level analog design. However, they are only valid to a first order degree of accuracy, so are not, on their own, sufficient for design. Far better would be to run transistor-level circuit simulations, such as SPICE [29] (Simulation Package with Integrated Circuit Emphasis) simulation. However, this is computationally intensive. In particular, design questions such as mismatch and system characterization issues such as Bit Error Rate (BER), are extremely difficult to simulate as they are of a statistical nature that may only be evident after millions of iterations. For this reason, many researchers have attempted to create high-level models of analog decoders, for quick design and verification. This section introduces approaches towards modeling different portions of AD systems.

Mathematical Modeling

Hemati [30] presented a model for analog decoding, arguing that the processing of analog decoders is analogous to the Successive Over Relaxation (SOR) numerical method for solving nonlinear functions. Furthermore, he argues that an analog decoder is superior to a synchronous digital decoder based on its underlying dynamics.

Delay Modeling

It is commonly agreed that an iterative decoder continues processing until it has converged to a solution. However, in the case of analog decoders the stopping criterium is vague – the decoder must continue processing until the values are suitably diverged that the comparators can make a correct decision on them (this spurring strong emphasis on comparator precision and accuracy). Even if firm stopping criteria were known, there is not a strong method to predict the time the decoder core would need to converge in every case. In [31] Hemati et al. present a model where each check and variable node in the decoder consists of an instantaneous transfer function followed by an RC low pass filter. However, an analysis presented in [32] found that this model was accurate for cases with no errors, but not when the input codeword had several errors; as these are the cases of most interest, this model is not yet sufficient.

Monte Carlo Simulation with Importance Sampling

Monte Carlo Simulation is a statistical simulation method, designed to allow for variation in a given variable. To run the simulation, the variables must be declared as statistically varying values, such as a Gaussian variable with a given mean and standard deviation. In one iteration, the simulator randomly selects values for each variable, then runs the simulation and records the results. It then randomly selects new values and runs the simulation for the second iteration. This is continued until the stopping point (which may be as simple as a given number of iterations) is reached. This is a well-known and popular technique, but suffers from a major problem: to accurately simulate a Gaussian distribution, millions of samples may be required to achieve a significant number of samples in the 'tails' of the distribution, beyond 2σ from the mean. In addition, the large numbers of samples that fall close to the mean are generally not valuable, since the purpose of Monte Carlo simulation is to view the statistical outliers. It is of little use for the designer to be informed that the circuit works when the variance of a value from its expected value is small; the cases of interest are when variance is large.

Importance Sampling, first explored in [33], was then applied in [34] as a means of improving Monte Carlo simulation. Under importance sampling, when the simulator selects the values, the selection process is weighted so that the error cases are more likely. This reduces the cases that convey little information, and allows the designer to focus on the outliers. The number of samples required to achieve meaningful results, of importance sampling is often superior to that of Monte Carlo sampling by several orders of magnitude [35], and in the published case for analog decoders allowed BER simulations to the order of $BER = 10^{-4}$ to be completed with only a few thousand samples [34]. This comes at some cost in accuracy; a review of the costs and benefits associated with this technique is available in [36].

Density Evolution

A second technique for modeling mismatch uses Density Evolution. In this approach, device parametric variation is modelled as a probability density function (pdf). This is processed by the transfer function in each node, shaping the pdf until it too is referred to the input. Winstead et al. [37] assumed mismatch to be a Gaussian variation in current at the output of each transistor; they found that mismatch had a negligible effect in the overall decoding efficiency as long as mismatch was less than 20%, a result achievable by analog designers in most processes. This is an encouraging result, though the assumption of current variation is overly simplistic; the most popular mismatch model [38] also considers mismatch of the device's threshold voltage.

2.2.2. Design Issues & Techniques

This section introduces the major design issues surrounding analog decoders, and offers a survey of approaches researchers have taken towards solving or accounting for these design issues.

Mismatch

Device mismatch in analog circuits refers to a device's parameter variations from design and, within a single design, from each other. It usually has negative consequences for the circuit, and as such must be minimized or at least considered during the design of an analog circuit. The most popular model, first suggested in [38], assumes that, for the simple NMOS device current equation

$$I_D = k_n^{'} \frac{W}{L}\left(v_{gs} - V_T\right)^2, \tag{2.5}$$

that mismatch can be modeled with additive Gaussian variables:

$$I_D = \left(k_n \frac{W}{L} + \Delta(k_n \cdot W/L)\right)\left(v_{gs} - \left(V_T + \Delta V_T\right)\right)^2 \tag{2.6}$$

where $\Delta(W/L)$ and ΔV_T are Gaussian variables with a mean of zero and a process-specific variance. Critics of this model suggest that this model owes its popularity more to its simplicity rather than its accuracy, but its staying power is undeniable.

As analog decoders matured, mismatch was of increasing concern. Lustenberger first addressed the issue in [39], concluding that the overall accuracy of analog decoders is limited by device mismatch. In the same paper he offered an analytical approximation for use in decoder models. Since then, further attempts have been made to characterize the effect of mismatch, usually treating it as another source of systematic noise for the decoder to overcome.

In [26] Gaudet et al. modeled mismatch as a variation in voltage signals passed between nodes as

$$V_{ACTUAL} = (1+\varepsilon)V_{NOMINAL}, \tag{2.7}$$

where ε is a Gaussian variable of zero mean and some variance σ. Bit error rate (BER) simulations then showed that global mismatch, which is when devices vary over large distances, has little effect on the overall performance. However, local mismatch had a flattening effect on the BER curves, particularly with longer block lengths. Dai et al. [40] performed a similar analysis, but assumed a Gaussian variance in currents instead of voltages; they concluded that the power loss due to mismatch, as referred to the input of the decoder, is

$$P_{LOSS}(dB) = 10\log_{10}(1+\eta N_0 \sigma_m^2). \tag{2.8}$$

where N$_0$ is the channel noise power, σ is the variance of the device mismatch, and η is a process constant varying from 0.25 to 0.8 [37]. Using density evolution analysis (discussed later), these results were extended to place an upper limit on the variance of the current factor ($\Delta(k_n \cdot W/L)$ in (2.6) above) of 20%.

Testability

Testability refers to the ease with which one can verify that a fabricated chip is structurally correct and will perform as designed. Digital testing is generally based on fault models, with stuck-at faults being the most basic example. Analog testing is generally parametric, wherein a known signal is produced at the input and the output is measured for certain characteristics. Analog testing is frequently difficult (functional errors can result not only from fabrication errors, such as dust contamination, but also from statistical variations such as mismatch) and analog testing circuits tend to be large.

In 2005, Yiu [4] introduced a built-in self test (BIST) module for analog decoders. It considered the system as three sub-circuits: the input interface, the core, and the output interface. The BIST tested the core circuits, but only ran simple tests on the interfaces, but only finds the most basic faults. Furthermore, its 'digital test of analog circuits' methodology only finds structural faults, not parametric errors that may still fell an analog circuit.

Interfacing

In a typical implementation, such as that shown in Figure 2.6, all the inputs must be presented at the input of an analog decoder in parallel. Since the symbols arrive serially, they must then be stored until all the symbols arrive. In digital systems this is easily achieved using a shift register, but in an analog system this requires switched-capacitance sample and hold circuits. Likewise, at the output there must be a parallel-to-serial as well as an analog-to-digital conversion, usually implemented using comparators and a shift register.

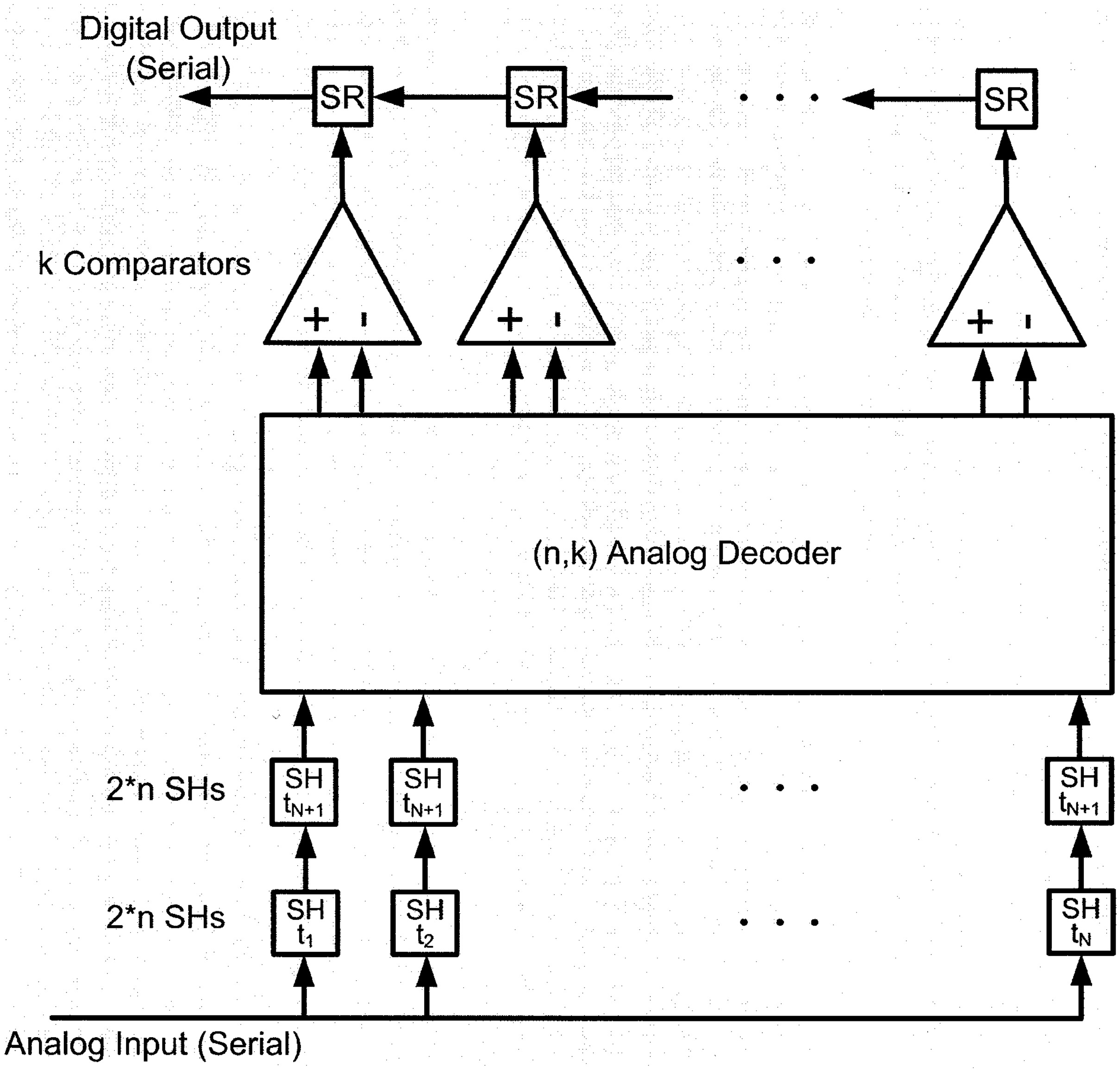


Figure 2.6: Analog Decoder Interface [14]

Unlike a shift register, capacitance-based circuits do not perfectly store their data, instead introducing signal degradation through leakage currents and charge injection errors. Since analog decoders use differential signals, there are twice as many values to be stored as there are symbols.

The question of interfacing was first addressed in [41]; it proposed three separate interfaces for a current-mode code, including an all-digital interface using on-chip D/A converters at the input and comparators at the output. In his design, Winstead [22] used an ‘analog-in digital-out’ design which used sample and hold circuits and comparators, but assumed an off-chip D/A converter at the input, as shown in Figure 2.1. In this interface, the input is analog; the DAC is not considered to be a part of the system. The values are loaded into a bank of Sample and Hold (S/H) units, then (when all the values have been loaded) are transferred into a second bank that feeds the core. t_n refers to the clock; each of the first bank of comparators is loaded at the same time, and the second bank is all driven from the same clock. For the FFT values to be formatted correctly for use in the AD, they must undergo a conversion from imaginary values to real ones; this block is included in Figure 2.7. This is explained in [10]; the system-level effect is to halve the number of wires leading into the AD. On the output side there is a bank of comparators, who make the final decision on a bit’s value. The comparators’ digital output is loaded into a shift register (SR) that is then shifted into the next portion of the communication chain.

Consider each block in Figure 2.7. The ‘Analog Decoder’ block has been well-examined, as shown in this section. The ‘FFT Processor’ and ‘Complex to Real Conversion’ blocks are analyzed in [5]. This leaves only the interfaces to be analyzed: they are considered in the following two sections.

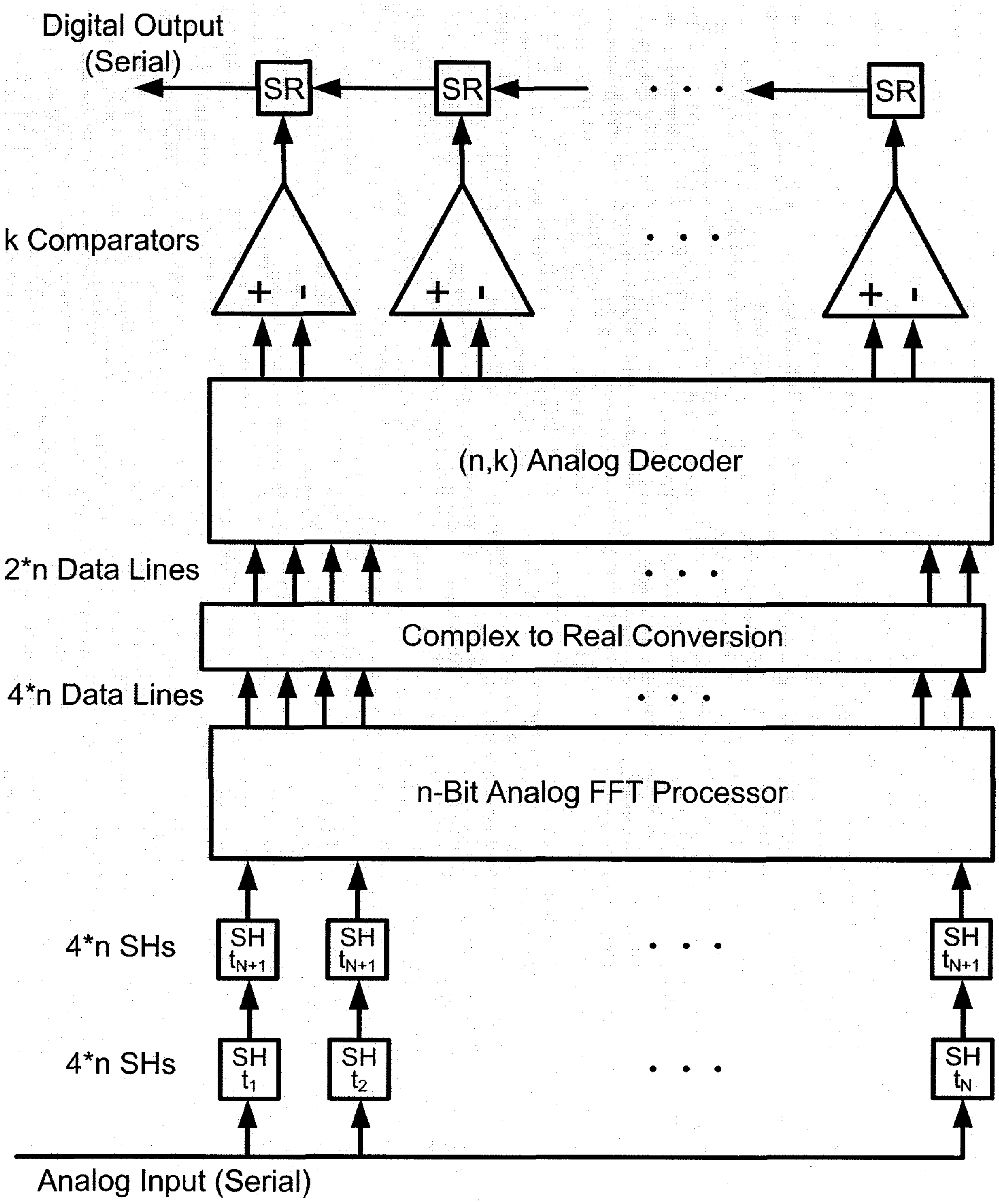


Figure 2.7: Combination Analog FFT and Analog Decoder System

2.3. Input Interface

As shown in Figure 2.6 and 2.7, the typical input interface of an AD consists of two banks of Sample and Hold (SH) elements. This section summarizes the basic parameters that need to be considered when designing both the circuit itself and its test method.

A sample-and-hold circuit is designed to hold the value of an analog signal for a given period of time, usually one clock pulse. Frequently used in analog-to-digital converters (ADCs), they are used to hold data that is to be analyzed or processed in some way. The simplest sample-and-hold circuit is a capacitor and a switch (implemented with a PMOS transistor), as shown in Figure 2.8 [42]. When the switch is closed (digital signal CLK is low), the circuit is in the tracking phase, where the voltage on the capacitor matches V_{in}. When CLK goes high, the capacitor is isolated from V_{in}, so it holds its voltage at what ideally is a constant level (this is known as the hold phase).

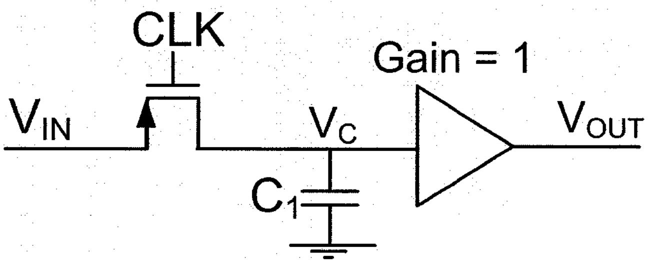


Figure 2.8: Basic Sample and Hold

Sample and hold elements use some combination of three basic elements: a storage element, input control, and an output amplifier (optional). These three elements are discussed below.

Sample and Holds require some method to store a charge. Voltage-mode techniques generally use a capacitor to store the charge. A well-designed capacitive storage unit can be very precise, but the designer must be aware of certain design parameters [43].

Switched Current is a current-mode equivalent; it uses diode-connected transistors to mirror input current. This method is still effectively capacitance-based, but can be effective at mirroring input current levels [44]. This has a derivative called Switched Voltage; it uses the Switched Current circuits for storage, but then the output signal is read as a voltage across a load [45]. However, it should be noted that these circuits are not free of the issues surrounding capacitive storage; some cells ([45][46]) still use capacitance to maintain voltage levels at certain nodes. These nodes are subject to all the issues surrounding capacitive storage, as will be discussed later.

To connect the storage unit to the value being sampled, a switch must be used. This is usually implemented using a single transistor or a transmission gate (one NMOS and one PMOS in parallel, controlled from inverted signals). The transmission gate has lower input impedance and a greater range, though is at least twice as large and complex as a single gate. It also requires complementary control signals, whereas a single gate only requires one signal.

A unity-gain output amplifier is generally shown so that the capacitor sees a high input impedance into the next circuit element. This also avoids the charge-sharing problem: if an output switch is used to connect to another element, when the switch is closed the charge on the capacitor is spread among the capacitor and the input capacitance of the next element. This will reduce the output voltage. If no output amplifier is chosen, this must be suitably planned for.

The main parameters to be considered include the acquisition time, droop voltage, and pedestal error [43]. These occur at different points in the circuit's operation, and so will be divided into the following sections: Sample Interval, Sample-to-Hold

Transition, Hold Interval, and Hold-to-Sample Transition. These are illustrated in Figure 2.9. The signals are as labeled in Figure 2.8. The input voltage, V_{IN}, changes during the hold interval. When CLK goes high, the capacitor charges to V_{IN}, taking t_{LOAD} seconds to do so. During the transition to the hold interval, pedestal error occurs, and then during the hold interval voltage droop occurs.

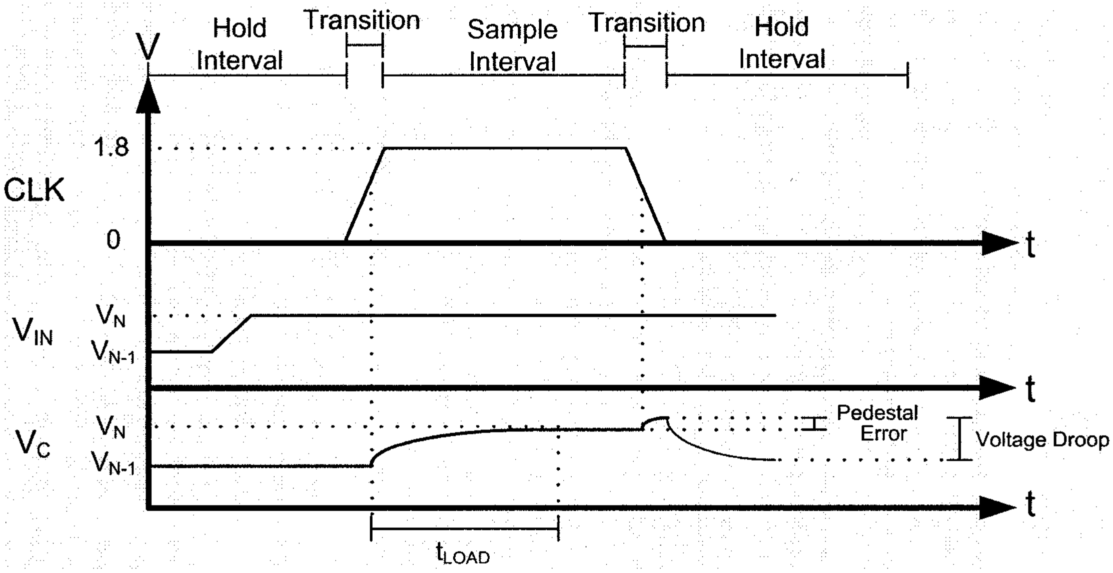


Figure 2.9: SH Timing Diagram, with Major Parameters

Sample Interval

The main parameter considered here is the acquisition time, here denoted t_{LOAD}, defined as the time required for the capacitor to charge to the data line's value [47]. This is dominated by the circuit's RC time constant; thus a smaller capacitance results in a faster acquisition time. Lesser factors include switching delay time, multiplexer settling time, and capacitive 'ringing', which is a long settling time due to poor damping of the RC circuit.

Also measured is the throughput offset, resulting from any nonlinearity in the device or the following amplifier (this includes offset/nonlinearity of the buffer amplifier, if used). Circuit linearity can be improved by using a fully differential circuit [45].

Sample-to-Hold Transition

The main concern during this phase is charge injection from the CMOS switch(es). This is termed the pedestal error, defined as the ratio of the induced voltage error to the full-scale input voltage range [48]. Unfortunately, this problem is difficult to face as available simulators do not accurately model charge injection errors. However, the general model for charge injection is [49]:

$$Q_C = \frac{1}{n} \cdot WLC_{OX}(V_{DD} - V_{IN} - V_T) \tag{2.9}$$

where W is the pass-transistor's width, L its effective length, C_{OX} and V_T are fabrication parameters, and 1/n refers to the fraction of the charge emitted from the source of the pass-transistor (n is typically selected to be 2). Recalling the simple equation

$$Q_C = V_C \cdot C, \tag{2.10}$$

and modeling the case where an unknown fraction of the charge is emitted from each of the source and the drain, and the source is connected to the capacitor of the S/H unit, the net effect on the capacitor is:

$$\Delta V_C = \frac{1}{n} \cdot \frac{WLC_{OX}(V_{DD} - V_{IN} - V_T)}{C} \tag{2.11}$$

From this equation one can see that using a larger capacitor compensates for this problem, as does minimizing overall transistor size.

Another important effect is that of using a transmission gate (with similarly-sized transistors) instead of a pass transistor. Because the charge injection is dependent on transistor geometry, but not charge mobility, the charge injection will likely be similar for adjacent NMOS and PMOS transistors. The injection of electrons from an NMOS and holes from a PMOS will then cancel each other out (assuming no mismatch; even when mismatch is considered, the total charge injected would be less than if a single gate were to be used). Similarly, extra devices known as 'dummy transistors' can be added between the pass transistor and the capacitor [49]. Such a device usually has half the size of the pass transistor, uses the same charge carrier as the pass transistor, but uses a clock signal that is complementary to the pass transistor. For example, an NMOS pass transistor will have an NMOS dummy transistor. When CLK goes high, the pass transistor is opened and the dummy transistor is closed. When this happens, the charge injected by the pass transistor is absorbed by the dummy transistor.

Using a differential topology is also useful for reducing charge-injection errors. However, the layout engineer must ensure the S/Hs are laid out such that the process parameters do not widely vary between the two devices.

Hold Interval

Voltage Droop Error occurs during the hold phase. This is the slow discharge of the capacitor through the transistors (which are ideally open circuits, but in fact have some finite resistance) Approximating the discharging capacitor as a linear circuit, the droop voltage expression is given in [50]:

$$\Delta V_C = t_{STOR} \frac{I_{LEAK}}{C} \tag{2.12}$$

where I_{LEAK} is the aggregate leakage current and t_{STOR} is the time over which the capacitance C must maintain a value. Leakage through the transistor could be through substrate leakage, which is leakage through the reverse PN junction between the drain and the base, and subthreshold current through the channel itself. Substrate leakage was found to be negligible for ADs [50]; subthreshold current can be described [14] as

$$I_D = I_0 * e^{\frac{\kappa V_{gs}}{U_T}} \left(1 - e^{-\frac{\kappa V_{ds}}{U_T}} \right) \tag{2.13}$$

where I_0 is a process and device dimension parameter, κ is a fabrication constant, and U_T = 0.025 V is the well-known thermal voltage. Typically κ = 0.7 (unitless), but κ may lie in a range between 0.5 and 0.99 [14]

Leakage current can be minimized by layout techniques such as designing a guard ring around the capacitor [47], but cannot be completely avoided. For this reason, it is best to be in the 'hold' interval for as short a period of time as possible. This in turn means maximizing the system's clock frequency. Use of a differential configuration is also helpful here because the capacitors tend to discharge at similar rates.

In multiplexed circuits, there is also the concern that 'data feedthrough', or capacitive coupling between the storage capacitor and the data line, will affect the stored value. For this reason it is important that the designer separate the data line and capacitor as far as possible. A design possibility is to use a distribution tree like that in [26]. The typical setup is that all SH circuits are fed from the same data line. This, here termed the "Bus Distribution Method" is shown in Figure 2.10.

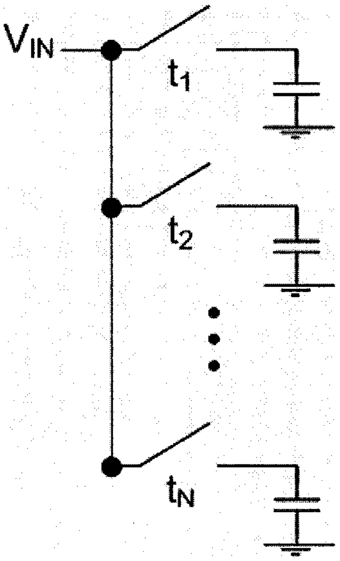


Figure 2.10: Bus Distribution Method

The method in [26], here termed the "Multiplexer Distribution Method", uses a distribution tree (as shown in Figure 2.11), where each node has only two switches, and several switches must be activated to access any one capacitor. This can greatly reduce data feedthrough, as there could be several switches between the data node and a capacitor, thoroughly insulating it. However, this requires more transistors and tends to be slower than the Bus Method (this is explored further in Chapter 4).

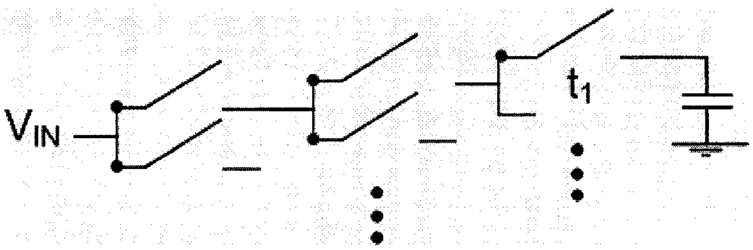


Figure 2.11: Multiplexer Distribution Method

Clock feedthrough is also a concern. Unfortunately, conventional models do not successfully predict the effects of clock feedthrough [51]. The designer must be sure to route clock lines as far from the storage node as possible.

Hold-to-Sample Transition

This section generally has few errors, as the data is neither being sampled nor read. It is generally not an area of concern to designers.

The major design and test S/H parameters are summarized in Table 2.1 on the next page. The review of AD interfaces is now half complete; we will now move to the output interface.

Table 2.1: Sample and Hold Design Parameters

Parameter	Optimization Method
Pedestal Error	Increase C Use differential signals Decrease W*L of transmission gate Use transmission gate, not pass-transistor
Data Feedthrough	Wire data line far from capacitors Use a distribution tree instead of one line
Clock Feedthrough	Wire clock line far from capacitor
Voltage Droop	Increase C Reduce leakage current Minimize storage time Use Differential Signals

2.4. Output Interface

As shown in Figure 2.7, the output interface for analog decoders generally consists of comparators and a shift register. The design of a digital shift register is trivial (with the exception of mixed-signal placement issues, such as ensuring that it is far from noise-sensitive sensitive analog circuits); here the design of comparators will be considered.

Comparators are simple circuits designed to implement the ideal transfer function

$$Vout = \begin{cases} V_{DD} & if\, Vin_1 \geq Vin_2 \\ V_{SS} & if\, Vin_1 < Vin_2 \end{cases}. \qquad (2.14)$$

Current-mode circuits implement a similar transfer function, but with current inputs (see [52-54] for examples). Although the output signals are digitized, comparators themselves are inherently analog devices; it is appropriate to think of comparators as extremely high-gain differential amplifiers, or as 1-bit analog-to-digital (A/D) converters [55].

In general, modern comparators use some combination of three basic elements: a differential amplifier, usually at the input, which magnifies the difference in the input signal, a positive feedback element to reinforce and magnify this difference, and a metastable output latch, which is driven by the first two elements to a final state. This element is used in clocked designs (simple comparators may only have the first two elements), and must be reset in between reads.

It should be stressed that these are not necessarily three distinct stages. However, these elements are all included somewhere within the circuit. These elements are discussed in further detail below.

Differential Amplifier

The simplest comparator can be a differential amplifier with such high gain that the output is driven to V_{DD} or V_{SS} by even the slightest difference in input voltage. Some modern designs essentially do this (see [56] for an example), and regardless of other components almost all comparators start with a differential amplifier at the input [57]. This is so that the later stages, which switch to a 0 or a 1 based on the input, have a clear signal for maximum precision. However, errors resulting from device mismatch between the differential pair is a serious cause of error, and in circuits requiring high sensitivity must be taken into account. This is discussed further below.

Positive Feedback

Positive feedback is important for magnifying the output of the differential amplifier into a large output signal. Through this stage, some small difference at the output of the differential stage will be quickly taken up by the positive feedback and magnified into a much larger value. This increases the overall speed of the output as a larger signal can be more quickly resolved by the output stage.

Metastable Output Latch

The output stage, frequently a basic latch such as a pair of cross-connected inverters, makes the final decision as to whether the output will be a 1 or a 0. As it must be reset (returned to its metastable state) after each decision, this stage is not included in all designs but is generally used in dynamic, or clocked, comparators. This is not as limiting as it may seem; most modern designs are clocked. In addition, use of a dynamic comparator is an effective method of reducing static power consumption, because the 'clock' transistors can be used to cut off current paths when not in use [58]. The remainder of this survey assumes a latch.

Comparators are primarily analog devices, derived from analog amplifiers, so are generally measured using the same characteristics: speed, power consumption, input offset error, hysteresis, and common-mode rejection [57]. These factors are briefly explored below.

The speed of a comparator is measured in terms of its propagation delay, the total time from when an input changes to the time the output changes. Delays are usually on the order of tens of nanoseconds, with the fastest circuits having a total delay of less than 1ns [59]. In order to maximize a comparator's speed, one must minimize the voltage swing of the amplification stage [60]. The comparator in [61], for example, uses diode-connected transistors to minimize total voltage swing. Speed can be optimized using positive feedback, but positive feedback applied at the input reduces the circuit's overall sensitivity. As a result, the most common practice among high-speed comparators is to use a positive feedback latch at the output which is reset before each read [62]. Comparator outputs are digital, so a slower-than-design speed manifests itself at the output as excessively slow pull-up or pull-down times. Some attention must be given to logical effort/fanout considerations, which is whether the output stage is capable of driving its load capacitance. If required an inverter chain with increasingly large inverters could be applied to reduce this problem [63].

Power is measured as total power dissipated, at a given frequency, with a given input signal. A comparator's power consumption is generally at odds with speed requirements: as with many circuits, high speed means being able to quickly charge or discharge parasitic and load capacitance, and doing so requires being able to source a large current. A study in [64] compared the effect of pre-charging the latch to V_{DD}, $V_{DD}/2$, or just above the transistor's V_T. It found that pre-charging to $V_{DD}/2$ consumes the least power of the 3 options.

Offset is the problem receiving the most attention from comparator designers. It is generally caused by mismatch between transistors in current mirrors used in designs. For a differential pair (typically the first stage in a comparator), there are 3 major values that cause error: differences in the load resistances, differences in the W/L ratios and differences in the V_T in the input transistors. This has the effect of changing the input threshold, modifying the transfer function in (2.11) to:

$$V_{OUT} = \begin{cases} V_{DD} & if\, Vin_1 \geq Vin_2 + V_{OS} \\ V_{SS} & if\, Vin_1 < Vin_2 + V_{OS} \end{cases} \qquad (2.15)$$

where V_{OS} is the offset voltage, and can be positive or negative. The effects of input offset can also be mitigated by using a fully differential design [62]. This has the added benefit in being effective against noise from the clock and power supply [55].

Circuits requiring high precision can use an external compensation circuit (for examples see [65] [62] [66]). The most popular circuit uses a switched capacitance design, which uses charged capacitors as a means of correcting for offset (for an example, see [67]). In this method there is a 'calibrate' phase and an 'evaluate' phase. During the calibrate phase, the load transistors are switched into a diode configuration, and charge a set of capacitors Because each transistor charges its own capacitor, each capacitor is charged according to the particular characteristics of that transistor. During the evaluate phase, these capacitors are connected to the output of the amplifier, and in doing so compensate for the mismatch of the transistors. This is an effective method of compensation, but adds complexity because it frequently requires multiple, precisely timed, clock signals, and the switches required introduce a charge injection error into the circuit. This can be met through conventional charge-injection mediation, but is nonetheless a source of error. For applications that require a large number of comparators, such as ADs or flash ADCs, these techniques are unpopular because of the huge area requirements. However, it is undeniable that some form of mismatch compensation is required. A different compensation approach for flash ADCs can be seen in [68], but this does not easily translate to ADs.

Resolution is the comparator's precision – the range beyond which it can reliably differentiate between the input signals. Resolution is also a parameter of the input amplifier. Like input offset, this is effectively a property of the input amplifier. General amplifier design is not covered in this thesis.

As a differential circuit, an ideal comparator does not react to a signal present at both inputs, DC or otherwise. The Common-Mode Rejection Ratio (CMRR) is a measure of this quality. This is generally an important measure for comparators, but not for comparators being used in a 'slicer' configuration, in which one input is tied to a constant value. CMRR is also a property of the input amplifier.

Hysteresis is the dependence of the input threshold on the comparator's previous input [42]. Hysteresis is typically a concern in comparator design, but can be mitigated by using a clocked design that resets the input values in between reads. As most modern designs are clocked, hysteresis is no longer considered to be a major concern.

The major comparator design parameters are summarized in Table 2.2.

Table 2.2 – Comparator Design Parameters

Parameter	Optimization Method
Speed	Minimize swing Include reset capabilities
Power Consumption	Precharge to $V_{DD}/2$
Input Offset	Reduce Mismatch of input stage Use switched-capacitance compensation
Resolution	No feedback to input stage Higher gain
Common-mode rejection	Common-mode feedback element
Hysteresis	Use clocked design with reset capabilities

This completes the review of analog decoder interfaces. The final section offers a brief review of analog testing, as this thesis seeks to add testability features to the design of AD systems.

2.5. Analog Testing

This thesis introduces test circuits for analog circuits and systems. Before this can be done, an introduction to analog test methods is required.

In [69], the authors divide analog test methods into specification-oriented testing and waveform-oriented testing. Specification-oriented testing means that the test engineer chooses limits on a circuit's parameters, such as an amplifier's resolution. If the circuit parameter fails to fall within these limits, then it is assigned a 'fail'. An easy extension of this concepts is to introduce 'binning', wherein a circuit will receive a grade, such as A, B, or C, based on its performance. This allows circuits to be classified for sale while keeping yield high. This is generally extremely time consuming and requires extensive testing equipment.

Waveform-oriented testing involves using a waveform generator and observing the circuit's response to that waveform. For simple signal processing elements, such as PLLs or amplifiers, this is best because much can be determined based on the circuit's response to a given input. Waveform based testing could involve generation of a sine wave to test the performance of a PLL, but could be as simple as generation of these simple ramp voltage sweep to test the input of a comparator. This method can be considered to be functional testing; the tester attempts to generate waveforms similar to actual inputs, and the performance of the circuit is required. It is the test engineer's responsibility to choose test signals that accurately represent operating conditions, so that if the circuit correctly processes the test signal, it will correctly process actual signals as well.

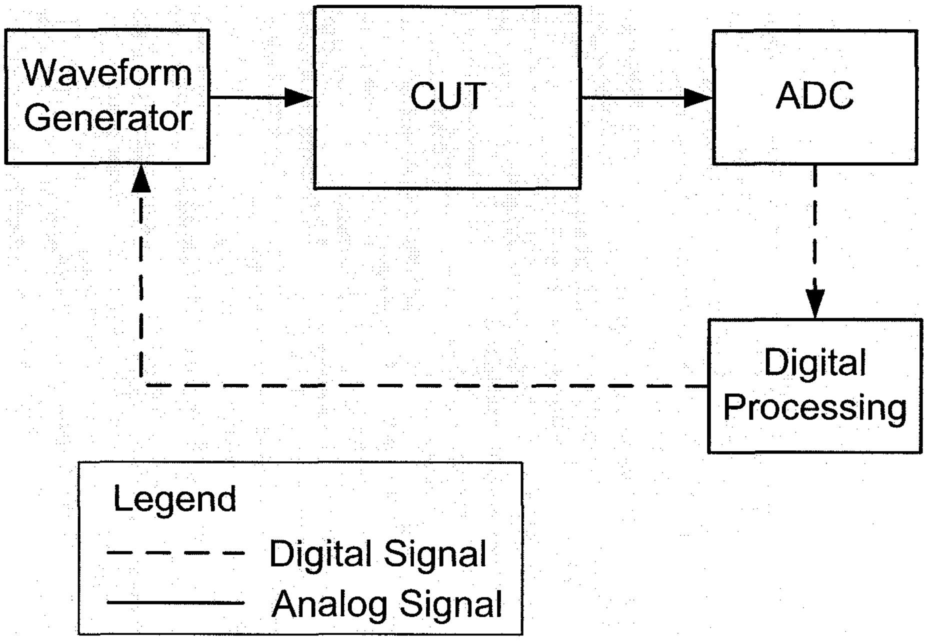


Figure 2.11: Analog Test Apparatus (based on [70])

Reference [70] offers a typical block diagram of an analog test apparatus, as shown in Figure 2.11. Note that the majority of the circuit surrounding the analog circuit under test (CUT) is digital. This is by design: because digital circuits are generally less prone to errors than their analog counterparts, as much of the test apparatus as possible is implemented digitally. This offers significant advantages, particularly that the design is easier to implement (digital design tools are more advanced than those for analog), verification of the test apparatus is easier, and the interface with other circuit components is much easier to implement as well.

2.6. Chapter Summary

This chapter has reviewed the necessary background on OFDM systems, analog decoders, interface circuits, and analog testing. In the next chapter these concepts are applied to propose specific circuits and test circuits for an FFT processor, SHs, and comparators.

Chapter 3

Proposed FFT and Interface Circuits

Chapter 2 introduced the block diagram for a combined FFT-AD system, in Figure 2.7. This chapter proposes applicable circuits and test circuits for the FFT core, SH, comparator, and output register blocks in this diagram. The approximate area requirements for each circuit are also calculated, assuming the (256,121) coded system targeted in this thesis. The circuits are only introduced here; the detailed design is included in Chapter 4.

This chapter is organized as follows. The discussion for each block is contained in one section: the FFT, SH, comparator, and output register cell are introduced in Section 3.1, 3.2, 3.3, and 3.4, respectively. In each section, the circuit is first introduced. Next the test requirements for this circuit are discussed, and the test circuit is introduced. Finally, the area estimate for each block is calculated. Section 3.5 is the chapter summary.

3.1. FFT Core

The system-level FFT core design was completed by N. Sadeghi [5]. He developed initial specifications and performed system-level simulation in Matlab and C, including mismatch analysis. He then developed the schematics in Cadence and ran SPICE simulations on the individual cells. This thesis assumes the existence of a core schematic which, at the schematic level, is characterized and ready for layout. Figure 3.1 shows the butterfly diagram of an 8-bit FFT processor. This diagram was chosen for its size; a 256-bit processor's butterfly diagram is far too large to be printed here. This is similar to the diagram in Chapter 2, except that the number of weighting factors has been reduced from three to one. This is due to Sadeghi's findings stating that an FFT need only generate weighting factors in the first quadrant; the remainder can be generated by using the first quadrant's weighting factors and interchanging wires (explained in detail in [10]). Furthermore, through BER simulation he found that an entire 256-bit FFT can be effectively implemented using only three, not eight as theoretically expected, weighting factors in the first quadrant (again, explained in [10]), while providing sufficient performance within a coded system. The cells, including current mirrors, weighting factors, addition, and wire interchanging are discussed below.

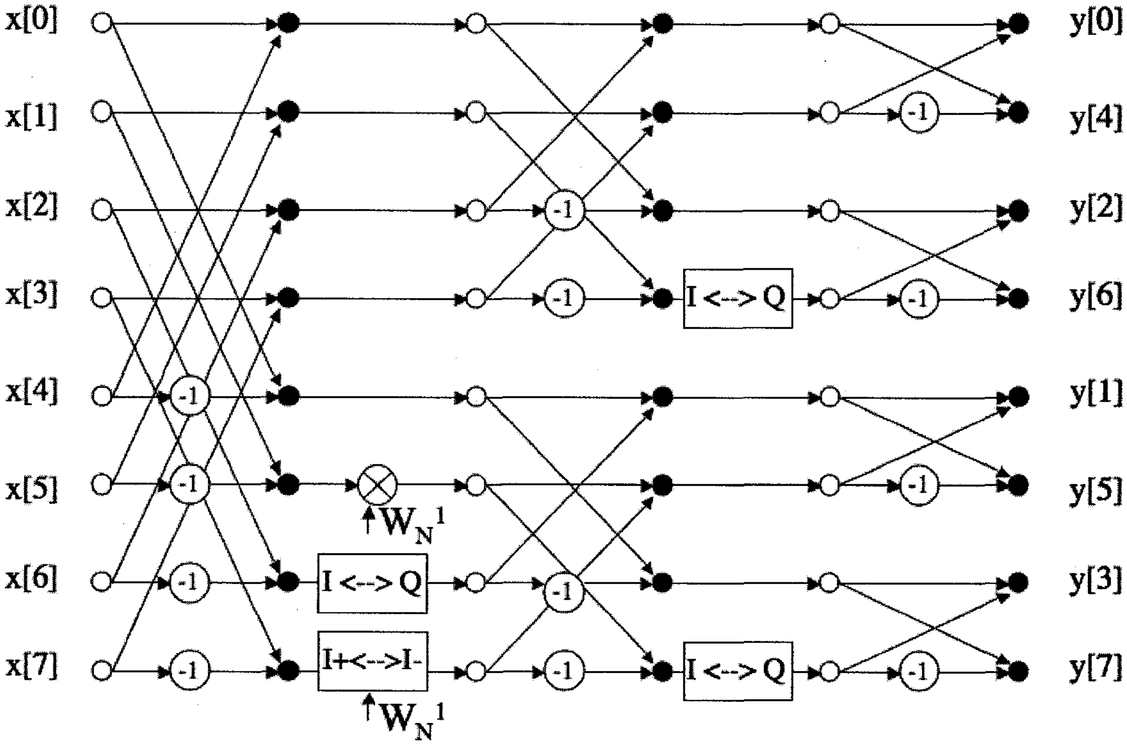


Figure 3.1: Eight-Bit FFT Butterfly Diagram (Modified for Layout)

Two current mirrors were required to create the 'base' current mirror: one PMOS and one NMOS. These were used for straight one-to-one duplication of currents. Recall that a single line in the butterfly diagram corresponds to four wires (each signal is represented by real and imaginary, differential analog values), so every base mirror actually consists of 4 current mirrors of unity ratio and a base W/L ratio.

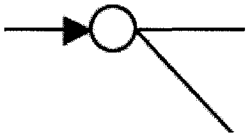


Figure 3.2: Base Current Mirror Representation

For an exact 256-bit FFT calculation 32 weighting factors (WFs) must be used. However, from a circuits perspective, this would create too many cells that, due to manufacturing inaccuracies, may well be indistinguishable from its neighbors in physical implementations. System simulations suggested that a suitable FFT could be created using only 3 weighting factors, with magnitudes of 0.4, 0.7, and 0.9. This can be accomplished with only a minor loss in FFT accuracy [5].

There are three independent weighting factors used in the FFT (thus six cells were required, to allow for both NMOS and PMOS mirrors). Each weighted cell consists of 4 current mirrors of a specific non-unity current ratio. The input transistor in each mirror uses the base W/L ratio, and the mirroring transistor uses

$$\left(\frac{W}{L}\right)_{mirror} = \left(\frac{W}{L}\right)_{base} \times (W_F)_k \tag{3.1}$$

where k is 0 to 2. Figure 3.4 shows the layout of one standard-size current mirror, as laid out by Pat Mercier. Recall that each transistor consists of 10 minimum-width fingers. Along the bottom of the image is ground, to which the source of each transistor is attached. The middle transistor is the input, with drain connected to gate, and the outside two transistors which each have one output pin on its drain. Weighted mirrors are created simply by reducing the number of fingers in the outer two fingers.

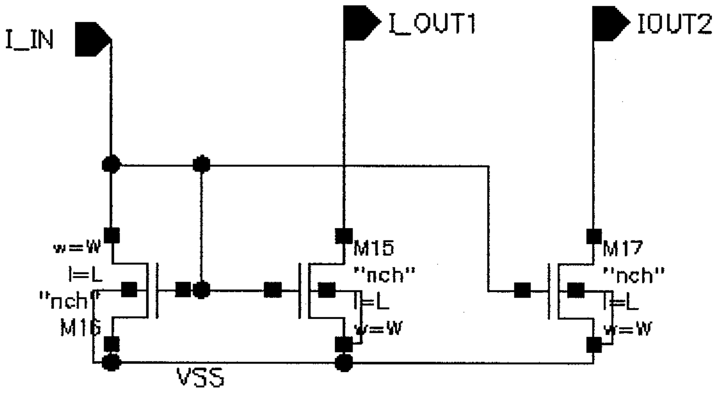


Figure 3.3: Schematic of one Standard-Size NMOS Current Mirror

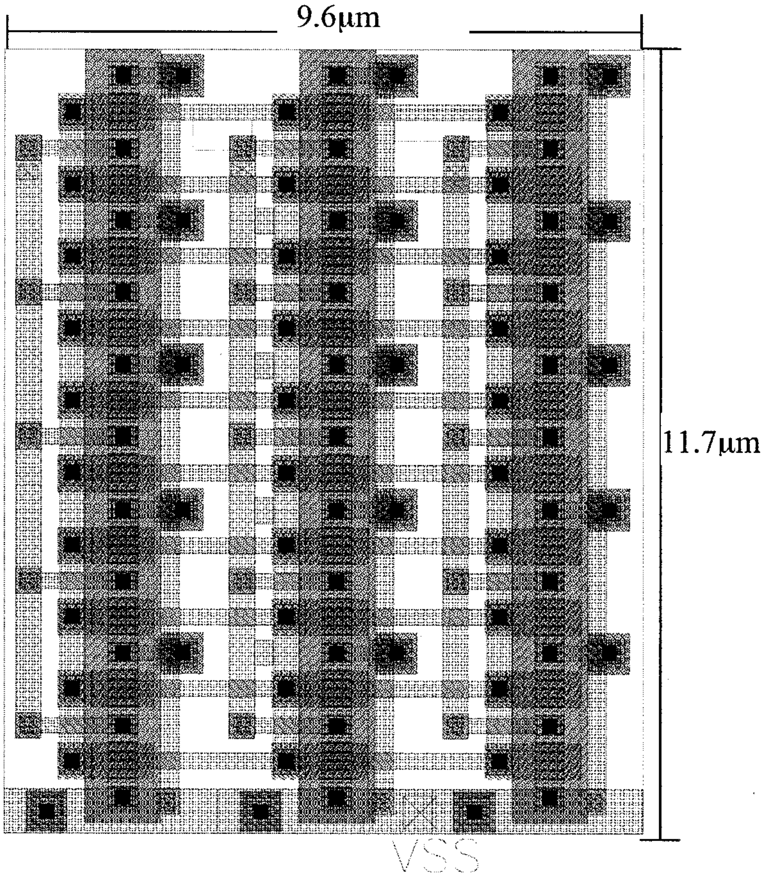


Figure 3.4: Layout of Standard NMOS Current Mirror (180nm 6M1P Process)

The weighted current mirror cells implement the WFs in the first quadrant. The remainder of the WFs, having the same magnitude but a differing phase, must be appropriately compensated. To achieve this, 'negative' (interchanging the positive and negative signals) and 'complex' (interchanging the real and imaginary signals) cells can be used. In the flow graph, they are assigned the symbols shown in Figure 3.5 and 3.6. The negative cell signifies a standard current mirror with the differential lines of each pair interchanged at the output. These cells are not implemented as physical devices; they simply represent re-wiring of the outputs of a standard mirror.

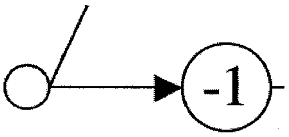


Figure 3.5: Negative Cell Representation

Likewise, the complex cell signifies an interchanging of the real and complex inputs to a current mirror.

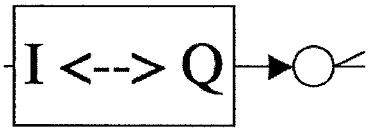


Figure 3.6: Complex Cell Representation

An addition cell, represented by the image in Figure 3.7, is not required. As the signals are current mode, to perform an addition the outputs of two current mirrors may simply be wired together.

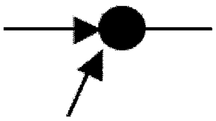


Figure 3.7: Addition Cell Representation

Now we will consider a test circuit for this core. In terms of system characterization, it is advantageous to be able to see internal values as they are passed between the two large blocks within the full analog system; it would be valuable to know if the FFT is functional on its own. For this, the circuit shown in Figure 3.8 is proposed. The proposed system, consisting of current mirrors within the FFT and an analog multiplexer, allows for viewing internal analog values.

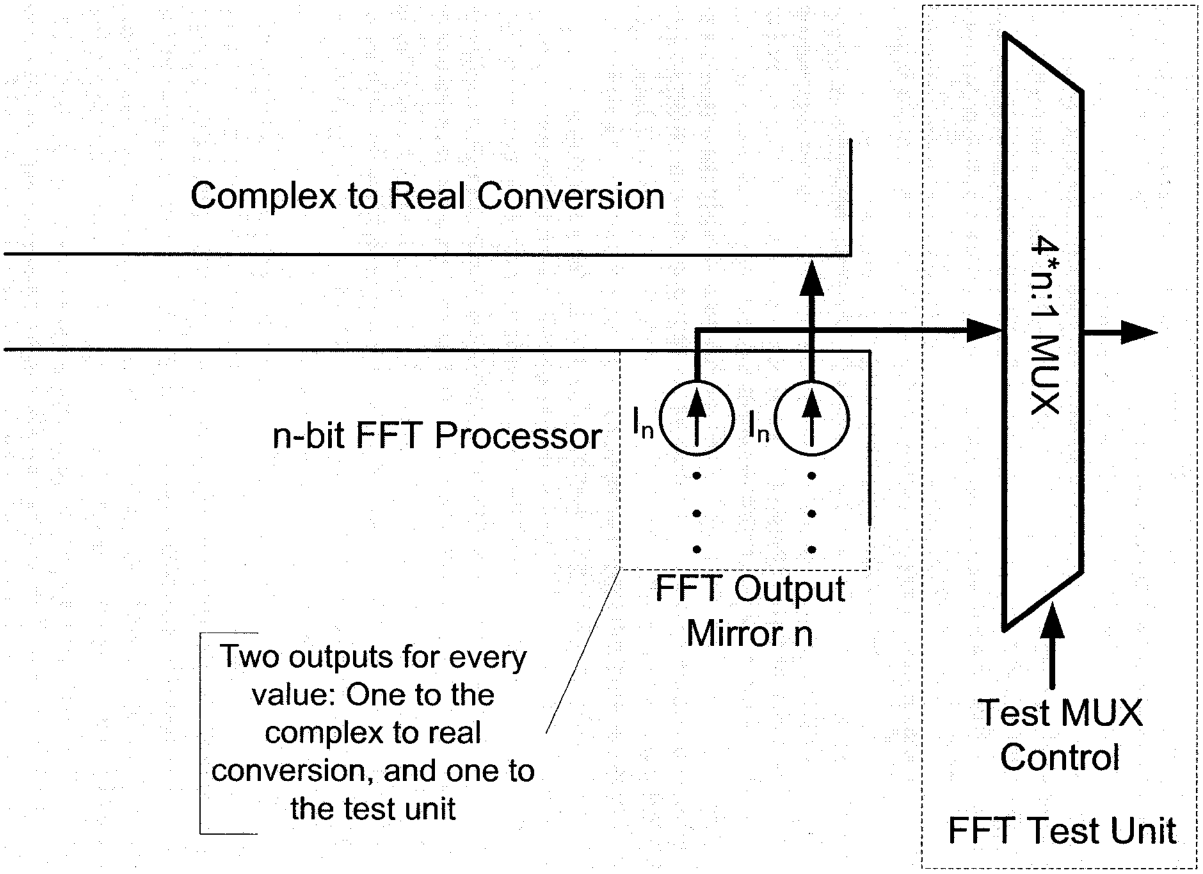

Complex to Real Conversion
n-bit FFT Processor
I_n
I_n
FFT Output Mirror n
Two outputs for every value: One to the complex to real conversion, and one to the test unit
4*n:1 MUX
Test MUX Control
FFT Test Unit

Figure 3.8: FFT Test Circuit

The values transmitted to the output are current-mode. This was chosen for three reasons. First, analog decoders make extensive use of current-mode signaling, and are certainly not restricted to visualizing elements only in the voltage domain. Second, current signals are easily copied: if a current mirror is already in use, a second transistor (as discussed in the next subsection) can easily be added with little impact on the overall circuit dynamics. Finally, currents are more easily transmitted over larger distances. While there may be a noticeable voltage drop in a wire that runs the length of the chip, a current signal will suffer far less degradation.

The output of the FFT is already current mode, so the final current mirror need only be slightly modified to allow an additional output. An example of a modified current mirror can be seen in Figure 3.9. The modifications to a PMOS current mirror would be very similar.

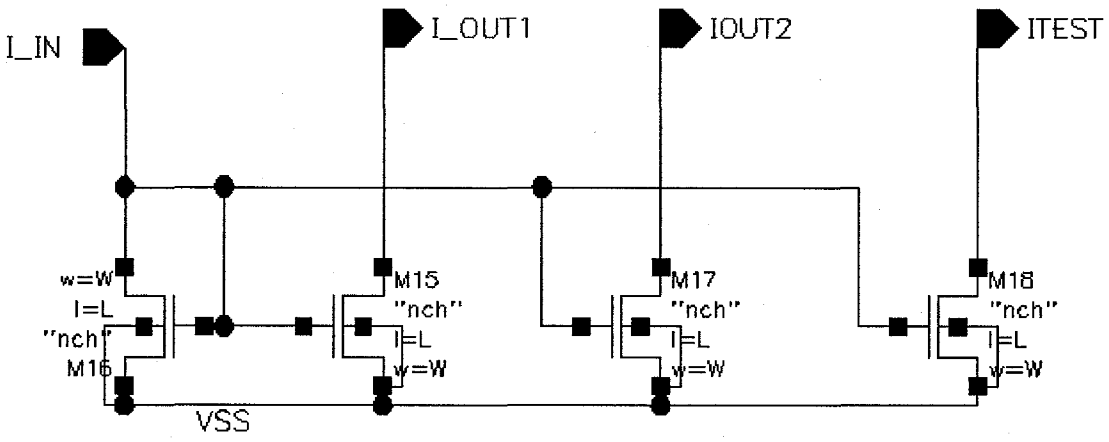


Figure 3.9: Modified Current Mirror

An analog multiplexer is a signal distribution scheme, and so has the same options as for the SH signal distribution, which are the bus and multiplexer methods. However, for the test circuit the MUX would only be incremented once per 'system period' (in this thesis termed t_{SYS}). That is, to view a given output the appropriate control line need be set only once, while the data is being loaded. To view the next line, the control line is incremented and the data is loaded again. Since the control lines do not change during the AD processing time, there is no concern that control line switching will feed through to the MUX output. The actual decision regarding bus or MUX architecture will be left for the next chapter; in this chapter's area estimates, a bus architecture is assumed.

Control logic is required to activate the desired transmission gate(s). The control logic could be generated in two ways:

- Line Decoder: line decode logic generates zeros on every line except the chosen line. This is the more versatile method, but is costly as it requires
$$n_{PINS} = \left| \log_2(n_C + 1) \right. \qquad (3.2)$$
pins.
- Shift Register: In this method, a shift register is loaded with all zeros except for the cell under test. The comparators must be tested sequentially. However, in this method many comparators can be tested at once, simply by loading all (or any pattern of) ones into the shift register. Of course, only one input value can be provided at any one time. This requires
$$n_{PINS} = 2 \qquad (3.3)$$
pins (One for data input and one for register clock).

For its better scalability, fewer pins, and capacity for testing multiple comparators, the shift register option was selected.

Now let us estimate the silicon area required for the block and its test circuit. Laid out as in Figure 3.4, each NMOS mirror is sized approximately 38.5 μm X 11.7 μm, and the PMOS mirror is 38.7 μm X 12.5 μm. A simple count of the number of cells required suggests that a 256-bit FFT core would require 11 980 individual current mirrors. Assuming half of these are NMOS and half are PMOS, and allowing 10% for wiring overhead, the area estimate for this core is:
$$A_{256-FFT} = 1.1 \cdot \left[11980 \cdot \left(\frac{38.5 \cdot 11.7}{2} + \frac{38.7 \cdot 12.7}{2} \right) \mu m^2 \right] \approx 6.2 mm^2 , \qquad (3.4)$$
or a die approximately 2.5 mm on a side. As the (256,121) AD core in [14] had an area of only 2.85 mm^2, the FFT area appears to dominate the system's overall size.

Addition of a test circuit means adding one extra transistor to each output mirror, and adding a multiplexer, consisting of one transmission gate and one shift register cell per bit, of the same size as the FFT output. Assuming each transistor has an area of 8 μm^2, TG has an area of 16 μm^2, and each shift register cell has an area of 50 μm^2 (reasonable values for the available 0.18 μm TSMC process), the total overhead for the FFT test circuit would be

$$\%Overhead = \frac{256*(8\mu m^2 + 16\mu m^2 + 50\mu m^2)}{6.2mm^2} \times 100\% = 0.3\%. \qquad (3.5)$$

This is a low overhead cost.

3.2. Sample and Hold Circuit

The SH circuit, the main circuit used in the input interface, is based on the designs used in the interface for previous ADs in this lab ([4, 14, 24, 57]). The circuit implemented is as in Figure 3.10 below.

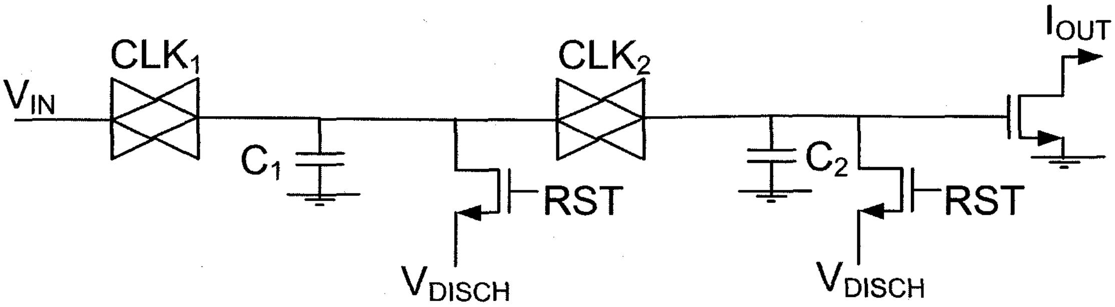


Figure 3.10: Proposed Sample and Hold Circuit

This circuit combines the two SH stages shown in Figure 2.7, allowing them to be designed and laid out as one. In addition, the output signal I$_{OUT}$ is a current, not voltage, signal; this reflects the need for a current-mode signal to be fed into the FFT. In addition, for a 256-bit FFT the first row of current mirrors are PMOS, so the NMOS transistor here neatly supplies a correct input current. As a note, if the first row of the FFT were NMOS circuits (as in a 128-bit FFT), the output transistors of the first mirror could double as the voltage-current converters here. In this case, there would be two output transistors instead of one.

For guidance on which SH parameters require testing, we can look to Table 2.1 in Chapter 2, which listed key design parameters for SH units. They are pedestal error, clock feedthrough, data feedthrough and voltage droop. Each of these parameters is discussed below, and its sensitivity to the output.

In the previous chapter, equation 2.11 described pedestal error. The net result of this error is a change of the voltage on the capacitor, ΔV_C. However, V_C is already directly tied to the output, so to be able to view this error we only need to view the output during the time when pedestal error occurs (the sample time to hold time transition).

Clock and data feedthrough both affect V_C. As with pedestal error, then, we need to be able to view ΔV_C during the time when clock and data feedthrough is most relevant. The sample time to hold time transition; clock feedthough also occurs during the hold time to sample time transition, but is of no interest as V_C is not valid at that time.

As seen in the previous chapter, equation 2.12 described voltage droop. This also results in a ΔV_C, though at a different time from pedestal error. To view this, we must observe V_C during the hold phase.

In summary, the above testability analysis borders on the trivial. The effect of every error can be observed as a change ΔV_C in the capacitor's stored voltage. In addition, since the SH circuits are already fed by the system's primary inputs (PIs), known values can easily be sent to each unit. The only circuit required, then, is to make the voltage V_C visible to a primary output (PO).

The proposed test circuit for SHs is as shown in Figure 3.11. To the 'basic' SH shown in Figure 3.1 has been added a single transistor. As determined above, the only circuit required is one that provides circuit observability to a PO. This can be accomplished by using a single transistor as a current source, which is then wired through a multiplexer to a PO. These currents are multiplexed to an output; assuming a single output is available, then multiplexer must then be sized to the ratio N:1 for N SH units.

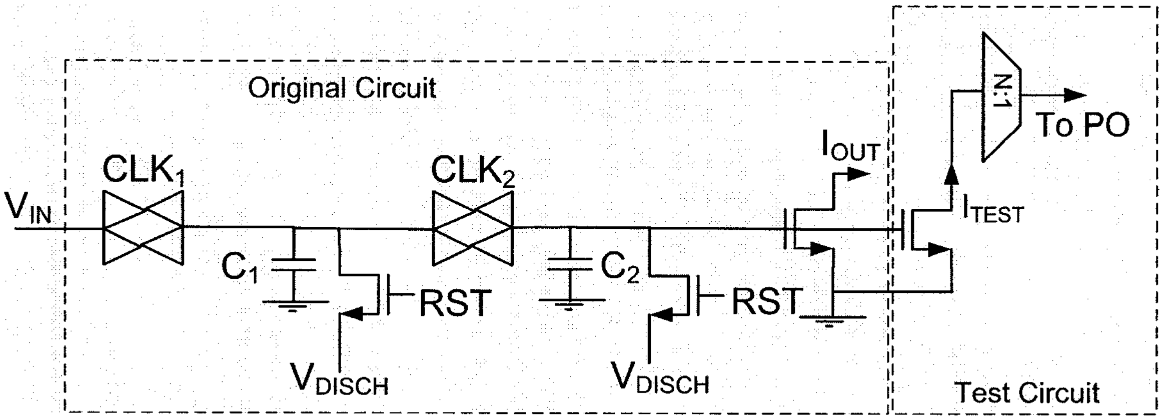


Figure 3.11: Proposed Sample and Hold Test Circuit

The current output can then be read by test equipment; while not shown, an ADC could also be added at an area cost for a digital output. The addition of a transistor increases the capacitance of the storage node. However, this is an uncontrolled value, so care should be taken that the input capacitance of the test transistor is much smaller than C1 as labeled in Figure 3.2.

Now consider the area required for the SH and its test circuit. The total area for the SH itself is dominated by the capacitor size. The TSMC process available to the designers offers metal-insulator-metal (MiM) capacitors which are sized 1fF/μm^2. Using the value of 80 fF used in [14], then each capacitor has an area of 80μm^2. Assuming the same transistor size of 8μm^2 as in the last section, the area of one SH is

$$A_{SH} = 7 \cdot 8\mu m^2 + 2 \cdot 80\mu m^2 = 216\mu m^2. \tag{3.6}$$

The area of the full number of 1024 SHs would be

$$A_{SH} = 1024 \cdot 216\mu m^2 \approx 0.22mm^2. \qquad (3.7)$$

Now to implement the test circuit, one extra transistor is added to each circuit to generate I_{TEST}, and a 1024-bit analog multiplexer is added. Here the same size estimate will be used for the analog multiplexer as in Section 3.2.1. This results in an overhead of

$$Overhead\% = \frac{1024 \cdot \left(8\mu m^2 + 16\mu m^2 + 50\mu m^2\right)}{0.22mm^2} \times 100\% = 34\%. \qquad (3.8)$$

This is larger, but in terms of the overall system size is still a small increase in area. In addition, it could be reduced by changing the N:1 ratio to N:N_P, where N_P is the number of pins used. This would reduce the number of shift registers required, though not the number of TGs. This area improvement comes at the cost of an increased number of pins.

This completes the circuits required for the input interface. In the next two sections we will examine the circuits required for the output interface: the comparator, and the shift register.

3.3. Comparator

This section introduces the comparator circuit used in the output interface and discusses test circuits for the comparators.

First consider the design of a comparator. From Table 2.2, we know that the ideal comparator design must: be clocked (with an input reset/precharge to $V_{DD}/2$), have a differential input, have a low mismatch in the input stage (or, ideally, have switched-capacitance mismatch compensation), have a high gain amplifier at the input, and have no feedback to input stage. As a further restraint, the output of the AD core is current-mode, so the comparator input must be current-mode.

In discussion with Chris Winstead at Utah State University (USU), a 'dynamic comparator' with a winner-take-all (WTA) input stage was chosen. These are shown in Figure 3.12 and 3.13 below. The WTA stage performs a nonlinear amplification and a current-voltage signal transformation. The dynamic comparator stage uses non-overlapping clocks to achieve a mismatch-compensated voltage gain.

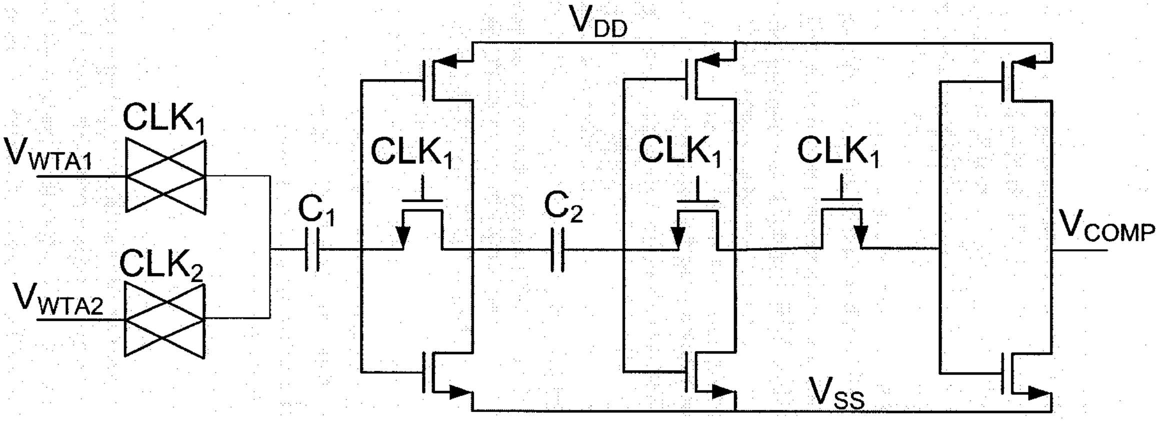


Figure 3.12: Dynamic Comparator

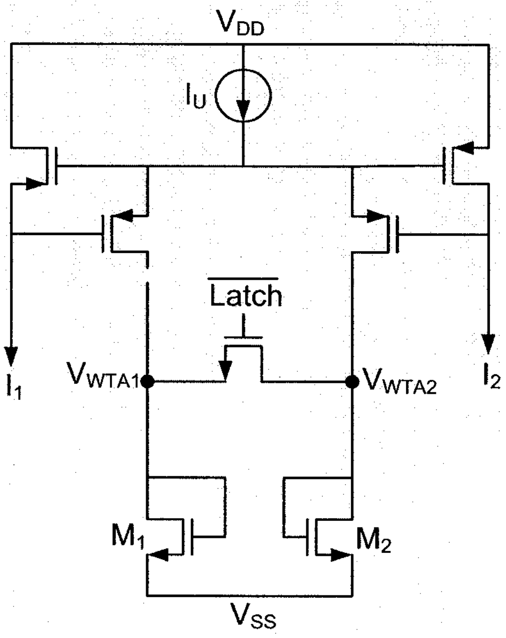


Figure 3.13: Input WTA stage

Now to consider testability of the comparator, we look at the parameters that require monitoring in comparators. They are, as listed in Table 2.2, speed, power consumption, input offset, resolution, hysteresis, and common-mode rejection.

Comparator outputs are digital, so a slower-than-design speed manifests itself as excessively slow pull-up or pull-down times. The output of the comparators under consideration is clocked, so if they are processing slower than the acceptable design speed, the fault will manifest itself through functional testing.

Resolution is the comparator's accuracy – the range at which it can reliably differentiate between the input signals. Because it is comparing outputs from a subthreshold circuit (the input signals are on the order of 10 nA), accuracy is of utmost importance. In addition, the comparator's resolution can be affected by transistor mismatch, which is a real possibility. Resolution, then, should be considered.

Consider the comparator's input threshold V_{TH} on a number line, as in Figure 3.14.

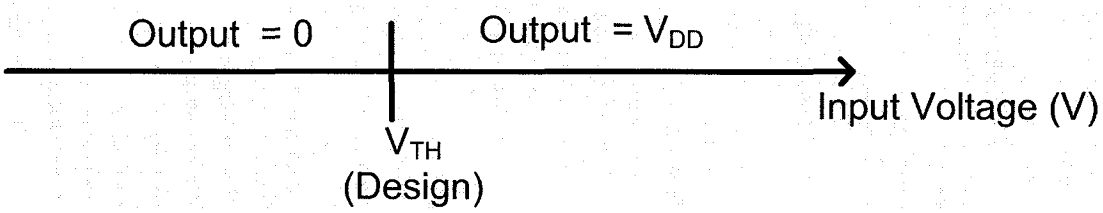


Figure 3.14: Graphical Representation of Comparator Input Threshold Voltage

Resolution refers to the 'blurring' of this line, as in Figure 3.15. Offset, conversely, represents an actual movement in the threshold voltage, as in Figure 3.16. The maximum error in V_{TH}, then, is a combination of these two, as in Figure 3.17.

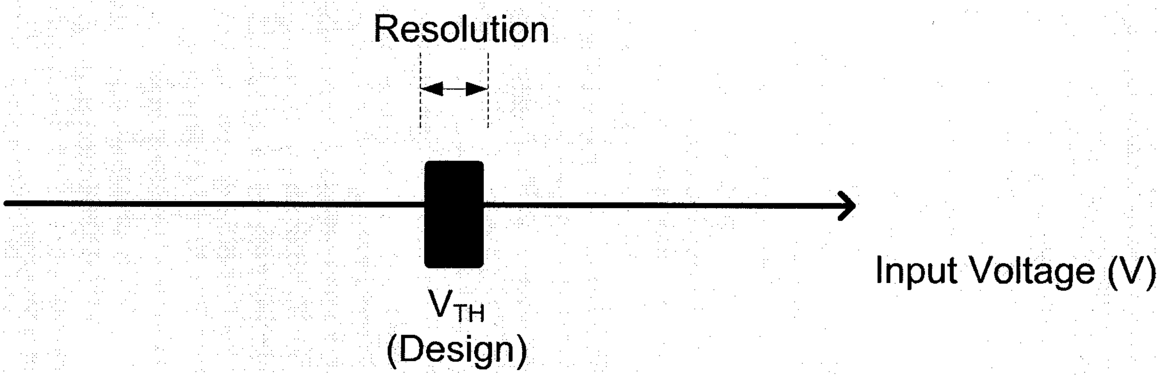


Figure 3.15: Graphical Representation of Input Threshold Voltage Resolution

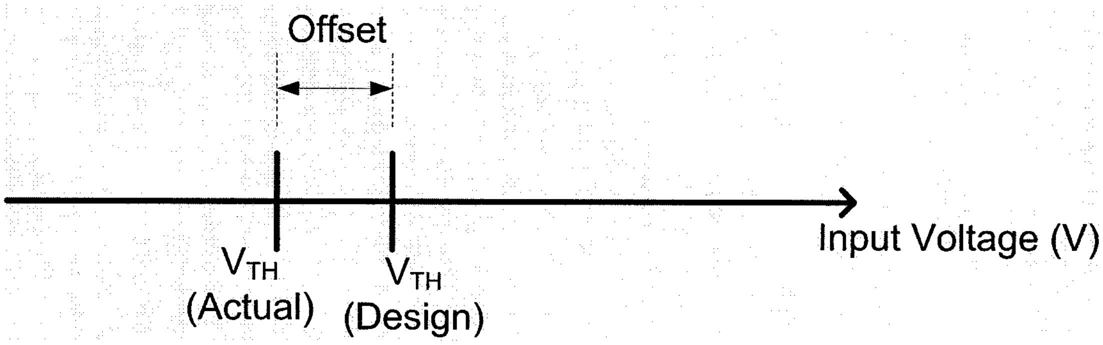


Figure 3.16: Graphical Representation of Input Threshold Voltage Offset

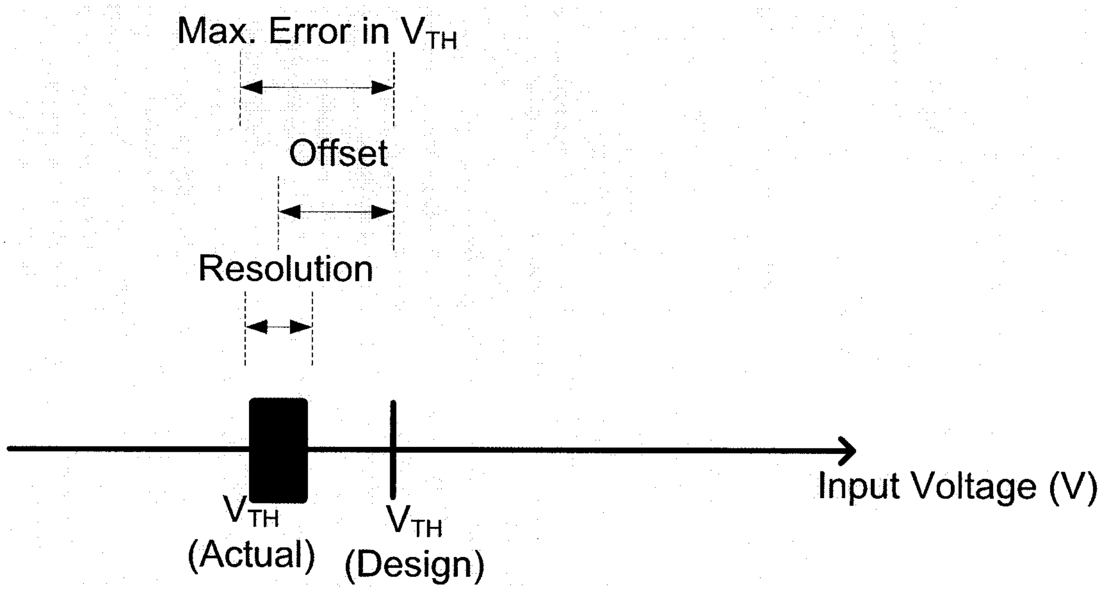


Figure 3.17: Graphical Representation of Offset and Resolution

The key question is whether these values must be considered separately or if they should be considered in the same test. Both of these are properties of the comparator input stage; both change the range of inputs for which the comparator reliably produces correct outputs. More importantly, from a functional standpoint it is irrelevant whether a comparator produces poor results because it has a large offset or a poor resolution; it is only important the comparator is producing poor results. Given this, it is possible to check the maximum allowable variation of offset and resolution at the same time, as in Figure 3.18.

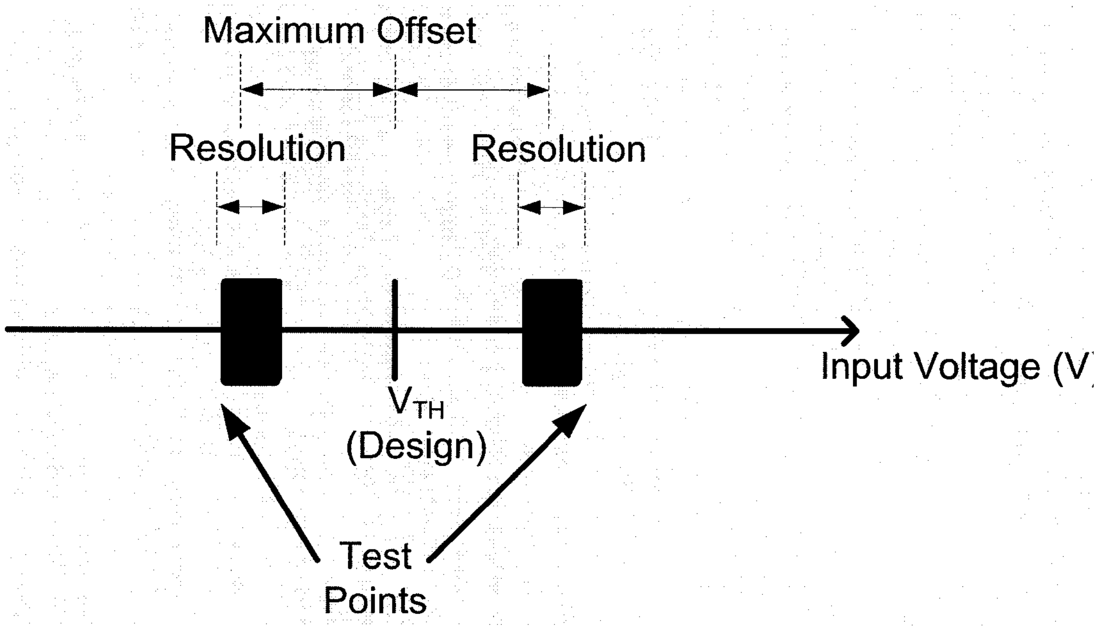


Figure 3.18: Test Points for Input Offset and Resolution

Thus only two points need to be tested, to ensure proper results. This requires controlling both inputs and observing the output of the comparator.

Effectively, testing the power consumption refers to I_{DDQ} testing, which may be a reliable manufacturing test. This is effective for physical-level models, such as testing for the correct operation of specific transistors. Here we will assume the circuit to be correctly manufactured, and only be concerned with parametric error due to mismatch. For these errors, I_{DDQ} testing will not be effective.

Hysteresis is typically a concern in comparator testing, but as this comparator is reset between reads, hysteresis is not a concern.

The common mode rejection ratio (CMRR) is the measure of a change in resolution over the range of input values. In order to check the CMRR, then, the resolution test above must be re-run several times, each with a different common mode across the input range. If the comparator has the same resolution over these tests, then it passes the test.

In summary, with the exception of power supply testing, all tests can be performed by setting known inputs and viewing the outputs. This led to the selection of the proposed test circuit below.

If we take a basic comparator circuit in an AD output interface to look like in Figure 3.19 ('SR' is a shift register cell), the proposed testing circuit is as in Figure 3.20.

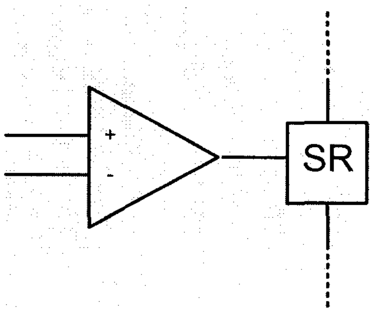


Figure 3.19: Comparator Circuit

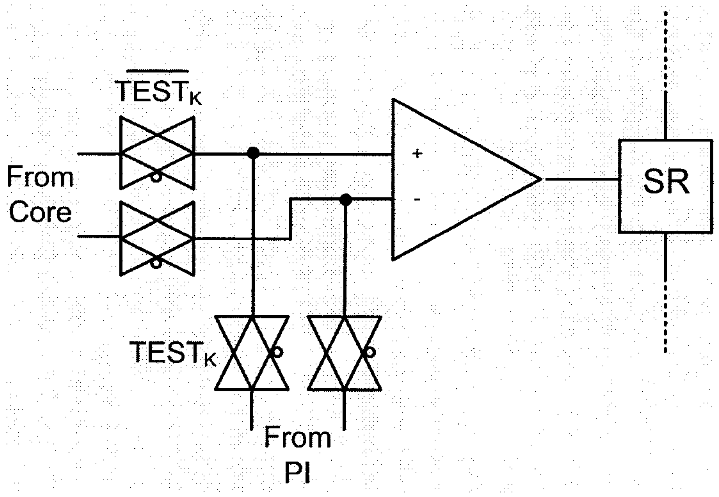


Figure 3.20: Test Modifications to Comparator Circuit

The input controllability comes from two input pins that have been added. The output observability comes from the shift register cell. For example, a test to see if the delay of the comparator's 0 to 1 switching time is less than 100ns could be run in this manner:

1. Set TEST_K to 1 to activate the test for this comparator
2. A given input is set on the inputs to generate a 0 on the output
3. A second input is set to generate a 1 on the output
4. A timer counts 100ns
5. The shift register is clocked out and the appropriate bit is read. If the output is 1, the comparator passes; if not, it fails.

As can be seen in the figure, this requires the addition of four TGs per comparator, two external pins to provide the test inputs, and a test control line for each comparator (denoted TEST_K, where K ranges from 0 to (n_C)-1 and n_C is the total number of comparators on -chip. One input pattern, normally all-zeros, must be reserved for normal operation, where no TEST lines are 1). To generate the control signals, the two options as presented in Section 3.1 are again available: the line decoder option and the shift register option. Again, and for the same reasoning as in Section 3.1, the shift register option was selected.

Now we move to the comparator area estimate. The comparator itself has 21 transistors, assuming a single transistor to generate the current supply I_U. Winstead suggested values of approximately 200fF for C_1, and 300fF for C_2; these values, along with the standard 8 μm^2 transistor size used in the area estimates in Section 3.1 and 3.2, will be used for the area estimates. The estimate area for one comparator cell is

$$A_{COMPARATOR} = 21 \cdot 8\mu m^2 + 200\mu m^2 + 300\mu m^2 = 668\mu m^2 . \qquad (3.9)$$

All 121 comparators in the system would then take

$$A_{COMPARATOR} = 121 \cdot 668\mu m^2 \approx 81000\mu m^2 \qquad (3.10)$$

The test circuit requires adding 8 transistors per comparator. As a result, the test overhead would be

$$Overhead\% = \frac{\left(4 \cdot 8\mu m^2 + 50\mu m^2\right)}{668\mu m^2} \times 100\% = 12.2\%. \qquad (3.11)$$

This is a typical overhead for test circuits.

3.4. Output Register

As this system has 121 comparators, there must also be 121 shift register cells. These values are never stored in a working circuit; they are taken in from the comparator then shifted out immediately. For this reason, a dynamic register such as that shown in Figure 3.21 was selected. This circuit is directly available in textbooks such as [63]; the only modification is to add a multiplexer at the input so the register can be switched from a parallel load (when all 121 comparators are driving their respective cell; V_{COMP} is the comparator output voltage in Figure 3.12) to shift mode, when the values are being shifted out (the output Q_{OUT} is tied to D_{IN} of the next cell).

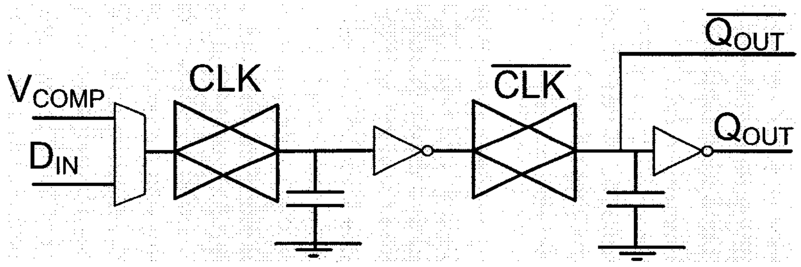


Figure 3.21: Proposed Output Register Circuit

The testability analysis for this block is quite brief. While dynamic comparators are considered, from an ad hoc point of view, to be difficult to test, this register could quite easily be made scannable. The only requirement would be the addition of a single input pin for a digital input to the first register element. Aside from the one pin, there would be no further area requirements.

The output register area estimate is also brief. The register itself has 22 transistors, most of which are in the two-bit MUX. Using the standard $8\mu m^2$ used in earlier estimates, the area for one register cell is

$$A_{COMPARATOR} = 22 \cdot 8\mu m^2 = 176\mu m^2 . \quad (3.12)$$

All 121 register cells in the system would then require

$$A_{REGISTER} = 121 * 176\mu m^2 \approx 21000\mu m^2 . \quad (3.13)$$

The test circuit requires no additional transistors, and pins have not been considered in the overhead calculations. As such, the test overhead is zero for this circuit.

3.5. Chapter Summary

This section offered test circuits for the sample and holds at the system input, the comparators and shift register at the system output, and the interface between the FFT and AD subsystems. The next chapter will focus on the detailed design of the interface and test circuits for a 256-bit FFT-AD system (the FFT and AD cores are designed in [5] and [14] respectively).

Chapter 4

(256,121) FFT-Analog Decoder System Interface

This chapter presents a design methodology for the interface for a (256, 121) AD-FFT system. The system diagram is shown in Figure 4.1. This figure is an elaboration on Figure 2.7, substituting specific values for variables and adding input elements that were previously implied. The AD core, which is being re-used from [14], is a Block Turbo Code (BTC) decoder. As can be seen in Figure 4.1, the output of the AD core requires 121 comparators, and the input requires 1024 discrete input SHs. This chapter demonstrates the detailed design calculations and decisions involved in designing this interface.

Section 4.1 covers the design of the input interface and its associated test circuit. Because both the SH test circuit and the FFT test circuit require an analog multiplexer, these are combined to make one large multiplexer, large enough for each of the 1024 FFT signals and 1024 SH outputs; this design is in Section 4.2. Section 4.3 discusses the output interface; Section 4.4 contains a brief note on power supplies for this system, and Section 4.5 summarizes the chapter.

4.1. Input Interface

The input interface consists of the signal generation, signal distribution, and sample and hold blocks shown in Figure 4.1. While this system would in reality receive values from a communications receiver, for the purposes of testing the signals will be generated by a digital to analog converters (DAC). The first decision then is whether to integrate the DAC on-chip with the FFT, or on its own. The design of a DAC itself is not a part of this thesis.

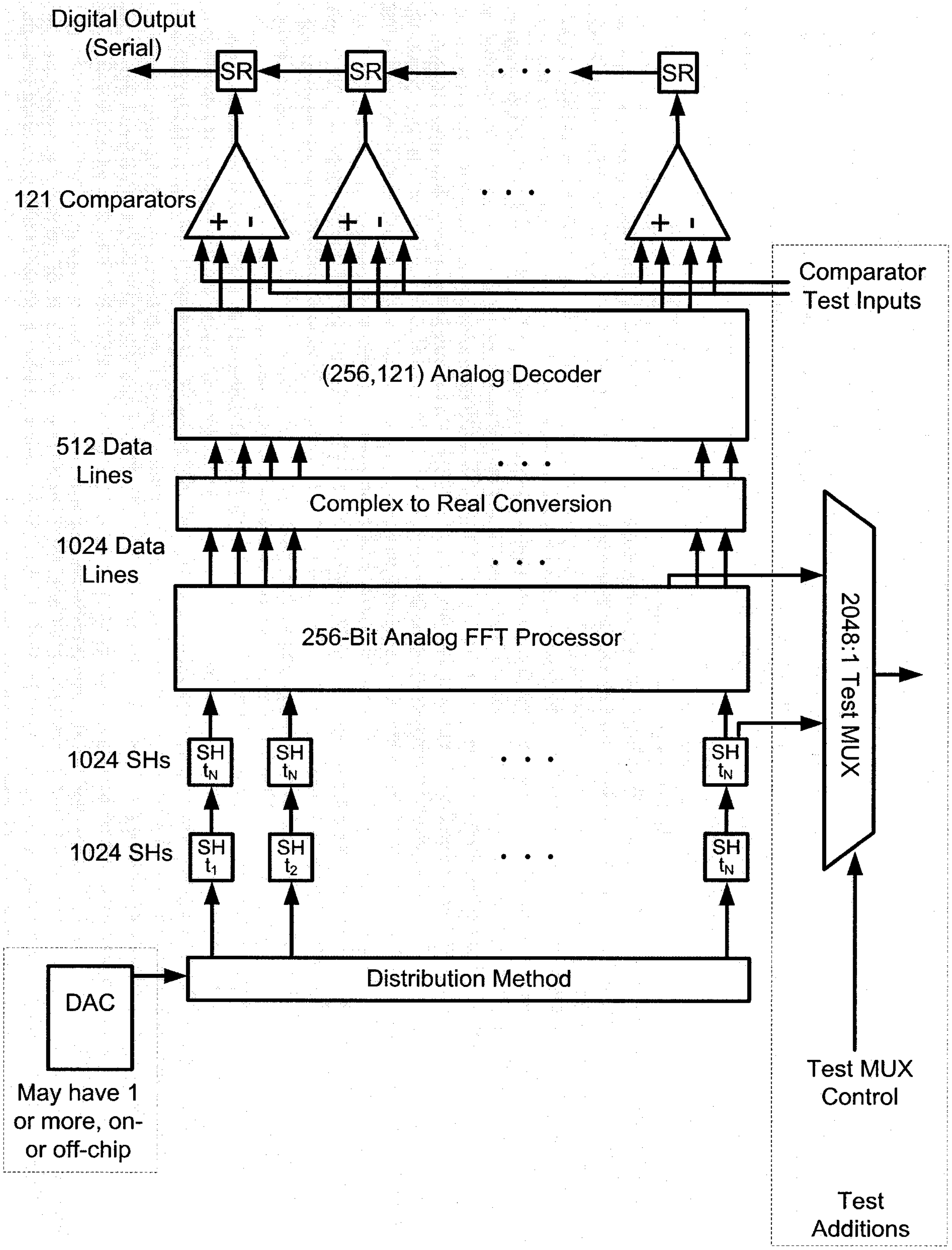


Figure 4.1: (256, 121) FFT-AD System Block Diagram

In the previous AD-only designs [4, 14, 24] in this lab, the DAC was off-chip and the signals were loaded one at a time. This necessitated a very slow operating speed, due to the high capacitance on the input pin; the off-chip DAC needed to charge the PCB trace, the input pin, the bonding wire, the bonding pad, the on-chip wiring, and the capacitor itself. Without detailed analysis it is still easy to see that this excess capacitance slows the overall speed of the circuit. This was acceptable for previous circuits, as the number of values to be loaded was small (two of the above references required loading only sixteen values per cycle). However, this FFT requires considerably more values, so the time available per value is considerably less. For this reason the DAC will be integrated on-chip for this design.

There need not be only one DAC; in fact with only one DAC the signal generation will likely be the system's critical path. Adding more DACs, and dividing the total number of SHs between them, the following calculations were used in the choice of number of DACs to include on-chip. Assuming a (256, 121) code, and given a target bitrate (BR), the number of 256-bit coded frames that must be processed in one second is

$$t_{SYS} = \left(\frac{256}{121} \times BR \right)^{-1}. \tag{4.1}$$

In this time, here called the system period, 1024 samples must be loaded into the SH registers. The time available for each sample is then

$$t_{LOAD} = t_{SYS} \times \frac{M}{1024}, \tag{4.2}$$

where M is the number of DACs used. The system requires the data to be held for the total processing time, t_{SYS}. For the worst case, which is very first SH to be loaded, the SH must also hold the data for the time required to load all the data in the bank of SHs. This is as long as t_{SYS} if only one DAC is used, but when multiple DACs are used this would be

$$t_{STOR} = \frac{2t_{SYS}}{M}. \tag{4.3}$$

The number of DACs was chosen using calculations shown in the MATLAB script *NumOfDACs.m* in Appendix A. This generated the graph of DAC clock frequency, equal to 1/ t_{LOAD}, versus M, the number of DACs used, shown in Figure 4.2.

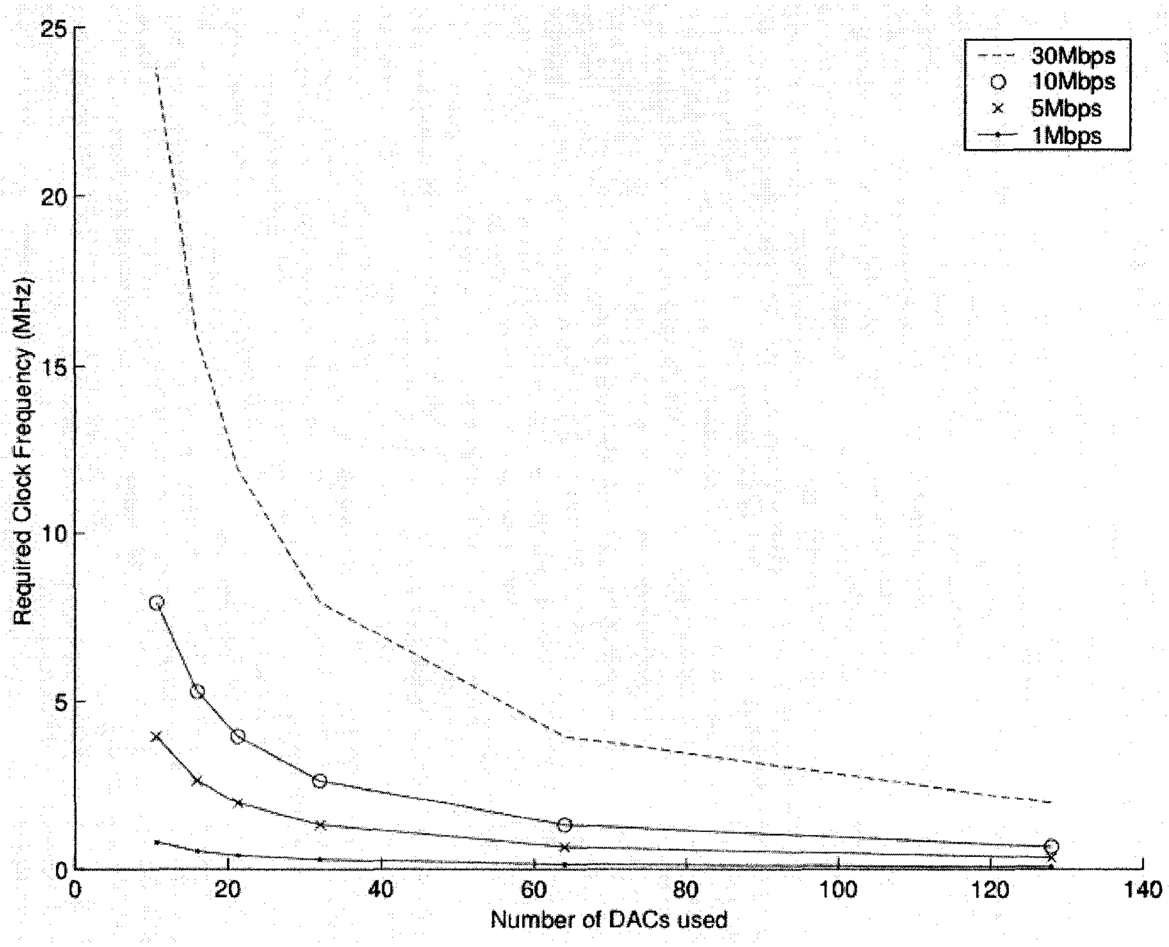


Figure 4.2: Design Clock Frequency for Given Number of DACs and Bitrates

Regardless of target bitrate, this figure suggests that the required clock speed would dramatically rise after 32 DACs on-board; 32 DACs are thus chosen for this application. The 1024 SHs would then be spread 32 per DAC.

Next, the output from the DACs must somehow be directed to the appropriate SH at the appropriate time. The choices were shown in Chapter 2: bus distribution in Figure 2.10 and MUX distribution in Figure 2.11. This distribution network's operation can be modeled with a basic RC circuit with a time constant τ, where τ is the Elmore delay of the distribution network [63]. R is the on-resistance of the distribution network and C is a combination of the capacitor's and the network's capacitance. In this system, the charging of the capacitor voltage can be modeled

$$V_C = V_{DAC}(1 - e^{-\frac{t_{LOAD}}{\tau}}) \tag{4.4}$$

This implies that the capacitor voltage V_C will not reach V_{DAC} for a very long time; however, if we instead search for the time where V_C reaches 99% of V_{DAC}, we find that

$$0.99V_{DAC} = V_{DAC}(1 - e^{-\frac{t_{LOAD}}{\tau}}) \tag{4.5}$$

$$\therefore \frac{t_{LOAD}}{\tau} = 4.6. \tag{4.6}$$

To achieve this we set the target sampling time to be greater than 5τ. Using a safety factor of 2, we then select the minimum clock period to be

$$t_{LOAD} = 10\tau. \tag{4.7}$$

Then we must calculate τ for both distribution options. In both cases, we assume that the transistors in the transmission gates are in the Triode region. R is the on-resistance of the circuit, which can be determined by R_P in parallel with R_N where each transistor's resistance is

$$R_{DS} = \frac{1}{\mu C_{OX}}(\frac{W}{L})(V_{GS} - V_T). \tag{4.8}$$

C is determined by two factors: the capacitance of the input capacitor itself, and the capacitance of the input line. For the bus method, the capacitance of the input line is determined by

$$C_{TOTAL} = C_{WIRE} + N_{SH} \cdot C_{TG}, \tag{4.9}$$

where N is the number of SHs in the line, C_{TG} is the input capacitance of the transmission gate, and C_{WIRE} is the metal-to-base capacitance of the wire itself. For the multiplexer method, ignoring the capacitance of the short wire, the capacitance of one node is

$$C_{TOTAL} \approx C_{ON} + C_{GD-NMOS} + C_{SD-PMOS}, \tag{4.10}$$

where C_{ON} is the capacitance of the 'on' transmission gate, plus the input capacitance of the 'off' TG. With these, τ can finally be calculated. The Elmore delay formula, for N different capacitors connected via a resistive network to a voltage source, is

$$\tau_{TOTAL} = \sum_{k=1}^{N} C_k R_k \tag{4.11}$$

where C_k represents each capacitor, and R_k represents the resistance, including the source output resistance, from the source to that capacitor [63]. For the bus option, this amounts to

$$\tau_{BUS} = C_{WIRE} R_{DAC} + C_{SH} \left(R_{DAC} + R_{SH}\right). \tag{4.12}$$

Likewise, for the MUX method the time constant is

$$\tau_{MUX} = C_{NODE} R_{DAC} + \sum_{n=1}^{8} n \cdot C_{NODE} R_{TG} + C_{SH} \left(R_{DAC} + R_{SH} + 8 \cdot R_{TG}\right) \tag{4.13}$$

The delay for both cases was calculated using the Matlab script *SHoptimize.m* in Appendix A; values for the physical parameters were taken from MOSIS extracted models [71].

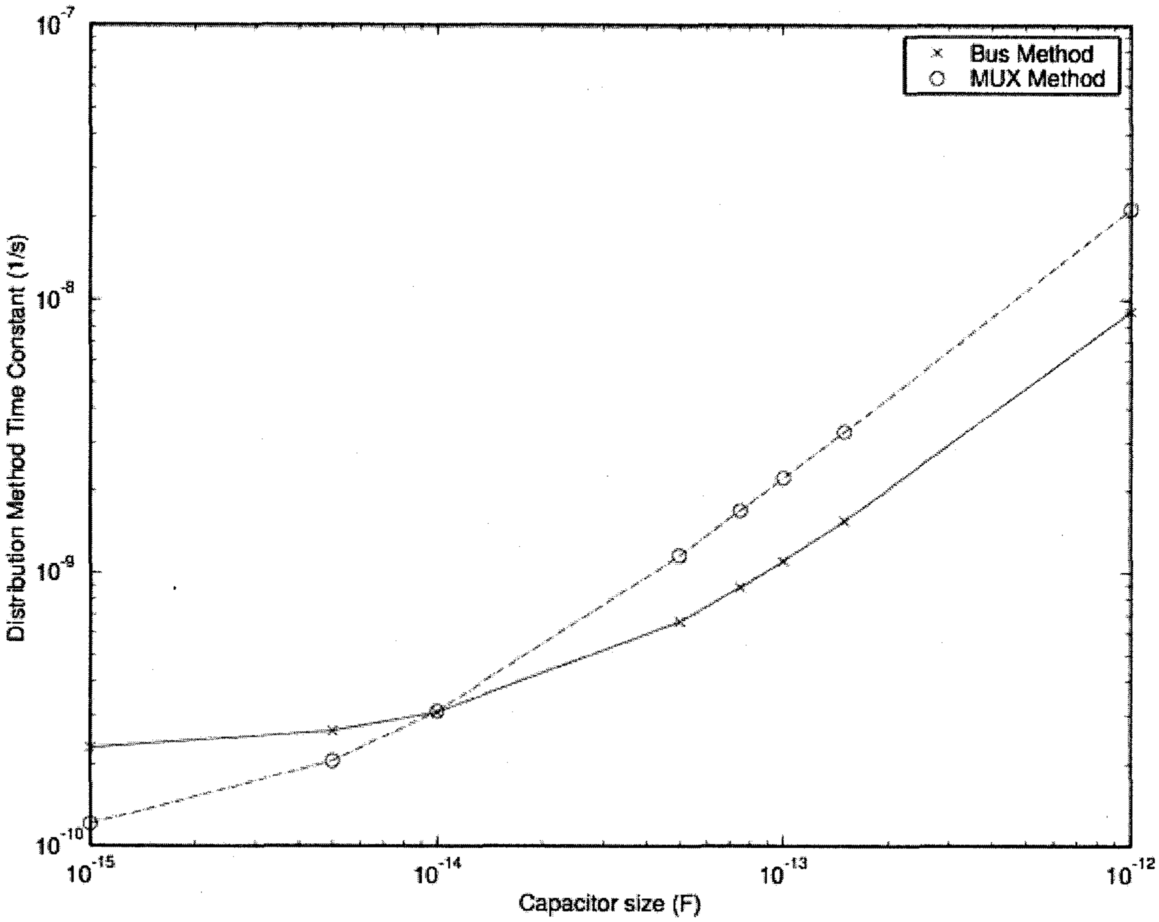


Figure 4.3: Time Constant of Distribution Methods by SH Capacitor Size

Consider Figure 4.3, which was generated by this script. This suggests that there are two broad regions for consideration. In the first, in the approximate range of 1 to 10fF, the Elmore delay analysis is necessary, and in general the MUX method seems to be the option with the smaller time constant. In the second region, in the general range of greater than 50fF, the time constant is dominated by the size of the capacitor being charged; this is because the SH capacitance, C_{SH}, is an order of magnitude larger than the parasitics represented by C_{WIRE} and C_{NODE}. in this range the time constants for the bus and MUX methods are

$$\tau_{BUS} \approx C_{SH}\left(R_{DAC} + R_{SH}\right) \tag{4.14}$$

and

$$\tau_{MUX} \approx C_{SH}\left(R_{DAC} + R_{SH} + 8 \cdot R_{TG}\right) \tag{4.15}$$

respectively. In this region, it is clear that the bus method is the faster option. As an additional consideration, for this system's 1024 values the bus method requires 1024 transmission gates; the multiplexer method requires 2047, more than twice as many. For these reasons the bus distribution method was selected for this system.

A critic may point out that data and clock feedthrough were ignored during this design decision. This is because data and clock feedthrough are not well characterized and it is not clear what effect it will have on the overall circuit. Given that, the designer believes it is more valuable to pay attention to that which can be measured over that which may or may not have an effect on the circuit's performance. It is hoped that the SH test circuits will allow insight into this effect, so that future designs can take these into account.

The SH circuit was introduced in Chapter 3; Here we detail the device sizes chosen for this system. The minimum speed is set by the voltage droop, which is the loss of stored voltage over time. This is in turn the result of leakage current, which is dominated by subthreshold leakage through the switch capacitors. Substituting (4.2) in (2.12), we get

$$\Delta V = 64 t_{LOAD} \frac{I_{LEAK}}{C}. \tag{4.16}$$

with I_{LEAK} and C as controllable variables. Clearly, maximizing C and minimizing I_{LEAK} will achieve the lowest voltage droop. These two targets are discussed below.

Current leakage (I_{LEAK}) is thought to be a major cause of imperfections in previous designs [14]. This occurs primarily through the discharge transistor: the controllable variables for this device are the width and length, and the voltage level of the source connection, V_{DISCH} in Figure 3.10. As a result, the discharge transistor was selected to be a 'long-channel' device, so that

$$\frac{W_{DISCH}}{L_{DISCH}} = \frac{0.28\mu m}{0.6\mu m}. \tag{4.17}$$

Looking at equation (2.13), describing subthreshold current, we can see that leakage current is dependent on V_{DS}. As a result, the discharge line was disconnected from ground, and instead connected to its own line, V_{DISCH}. Making V_{DISCH} nonzero reduces the differential value between V_C and V_{DISCH}, in turn reducing subthreshold current. Since V_{DISCH} is an external connection, a precise value does not need to be selected here.

The capacitor used in previous designs was 80fF; these ADs suffered from poor performance; the manufactured devices' BER was considerably higher than expected [14]. Unfortunately, because the systems had no test circuits it is difficult to know if any one circuit is performing correctly. The sample and holds are suspected, though not proven, to be a part of the systems' performance issues. As a result, the designers set out to nearly double this value to 150fF per capacitor. Early calculations, such as the DAC to SH calculations in Appendix A, use this value. However, these capacitors were implemented using the mixed-signal metal-insulator-metal capacitors available in our TSMC process, which have a capacitance of 1 fF per um^2. This requires the total space for all 2048 capacitors to be:

$$A_C = 1 \left. fF \middle/ \mu m^2 \right. \bullet 150 fF \bullet 1024 \bullet 2 = 307200 \mu m^2, \tag{4.18}$$

or a square more than 0.5mm on a side. Recalling that this does not even consider other area such as wiring and transmission gates, this is too large for any practical implementation. As such, the design focus shifted to physical size.

In physical implementation, it would be ideal if all the SHs were lined up in a row; this reduces the noise from running data lines next to the capacitors. Considering the minimum width of the capacitors as specified by the CMC design rules, wiring overhead, and spacing between capacitors, each SH must be at least 6.0μm wide. The 1024 SH units then, must be at least:

$$A_C = 6.0\mu m \bullet 1024 = 6.14mm. \tag{4.19}$$

The area estimate for the FFT core (see equation 3.4) was a square 2.5mm on a side. The possible solutions to this problem are:

1. wrapping the SHs around two sides of the FFT core (in the shape of a capital L), or
2. staggering the SH units so they use less space.

As the SHs must also have a DAC immediately behind them, option (1) was rejected because it requires interface area on two sides instead of one. In addition, the core could be resized to a rectangular shape, so it is the same width as the input interface. Attempting to meet this minimum width of the capacitor, and with some iteration with the extraction tool, a length of 15.7μm was chosen. This only calculates to 60 fF; however, extracted values and added parasitics resulted in a value of 110 fF.

Finally, the transmission gate sizes needs to be selected. By the guidance of Table 2.1, the TG must have identically sized PMOS and NMOS devices. In addition, because their resistance is in the distribution path, they should have as small a resistance as possible; this translates to minimum length and a greater than minimum width. The following device sizes were chosen:

$$\frac{W_{TG}}{L_{TG}} = \frac{1.5\mu m}{0.18\mu m} \tag{4.20}$$

This completes the design of the input interface. The next step is to design the test apparatus for the input interface. Recall from Chapter 3 that the only external circuit required is an analog multiplexer; this is designed in the next section.

4.2. Input Interface and FFT Test Circuit

As both the SH and FFT test circuits require an analog multiplexer, they were combined to create one large multiplexer, with 2048 inputs. In Chapter 3 a brief overview of the analog multiplexer was given; there shift registers were chosen to generate the required control signals. The multiplexer design is discussed here.

Thorough coverage of analog multiplexer design is available in [47]. It mentions two major concerns: settling time and crosstalk. These parameters are analogous to the distribution scheme parameters described earlier; they will be given only brief coverage here.

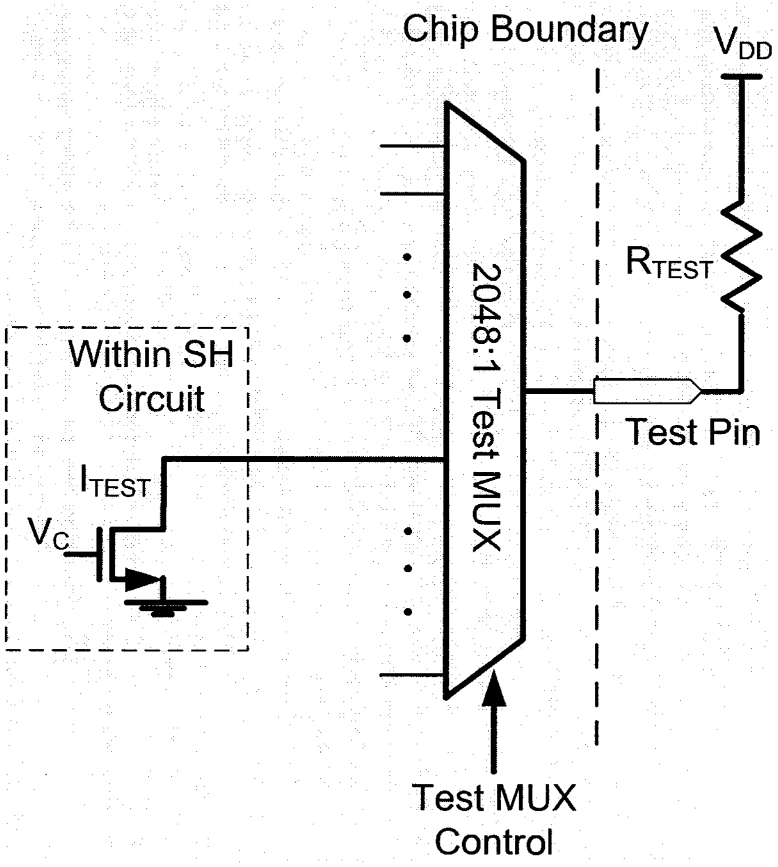


Figure 4.4: Test Multiplexer External Connections

t_S, the total settling time, is the time required for the output to reach 99% of its final value. Like t_{LOAD} above, it is determined by the time constant τ of this system. This system is as in Figure 4.4. The dominant capacitance is the output pin, and the dominant resistance is the external test resistor. Ignoring the transmission gate itself, the time constant is

$$\tau_{MUX} \approx R_{TEST} \cdot C_{PIN} \tag{4.21}$$

Conveniently this test resistor is a passive element, so can be easily characterized. However, the pin capacitance is currently unknown.

Data feedthrough, as defined in a digital textbook such as [63], is thought of as being crosstalk due to inter-wire capacitance. However, most crosstalk concerns revolve around high-frequency signals; they do not apply in this situation. Because the signals are current-mode, the larger concern is of leakage current through the nominally off transmission gates. If there is some leakage current I_{LEAK} through each TG, the total output current would be

$$I_{OUT} = I_N + \sum_{i \neq N} I_{LEAK} . \tag{4.22}$$

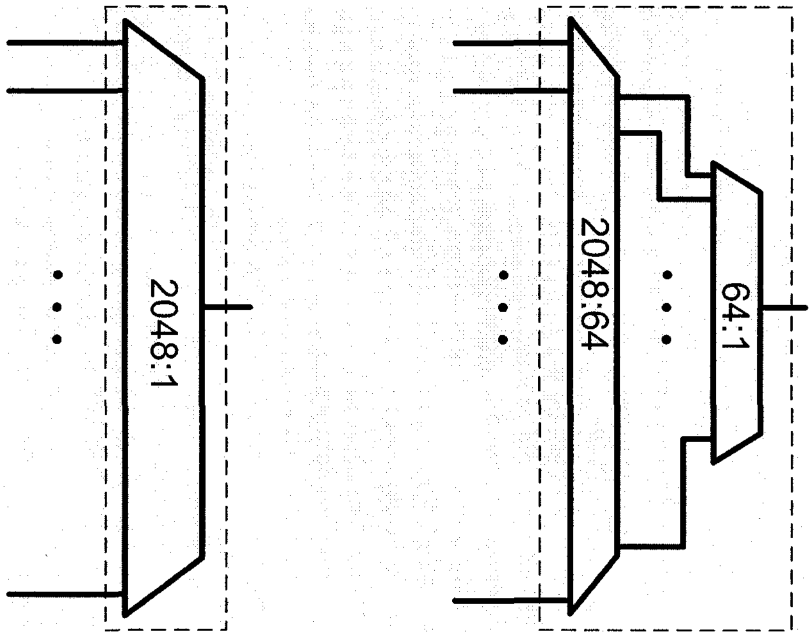


Figure 4.5: Two-Stage Multiplexer Architecture

As there are 2048 other TGs in this application, the difference can be significant. At this point, we can now visit the question of using a MUX or a bus architecture. The MUX architecture uses multiple stages of transmission gate; this 'transistor stacking' will reduce the overall leakage current. However, a full MUX implementation is costly from an area perspective. For this reason, a modified MUX architecture, consisting of two stages of MUX is implemented, as in Figure 4.5.

This completes the full input interface and test sections. In the next section we will discuss the output interface.

4.3. Output Interface

The output interface consists of the comparators and shift register cells shown in Figure 4.1. The comparator itself was selected and designed by C. Winstead of USU; the specific design of this unit was not performed by the author of this thesis so are not contained here.

The schematic for the output register was introduced in Chapter 3. The non-inverted output is not necessary for this application. To minimize area requirements, all devices can be minimally sized. The test circuit requires a row of transmission gates implemented in the circuit. For design simplicity, the same transmission gates are sized as those in equation (4.18).

This completes all system design work for the interfaces. We will end this chapter with a short note on system voltage supply.

4.4. Supply Voltage

It is common practice to provide different voltage supplies for the digital and analog portions of the chip. A total of six independent power domains pins are suggested for this system. These are digital registers, DACs, SHs, FFT, AD core, and comparators. This allows for reduced power supply noise, and characterization of each of the circuits' power consumption. All supply voltages should be 1.8V.

4.5. Chapter Summary

A design methodology for the system shown in Figure 4.1 was demonstrated. Device sizes and system-level values for the components introduced in Chapter 3 were calculated. In the next chapter, we will see a portion of this system implemented and fabricated.

Chapter 5

64-Bit FFT Test Chip Implementation

Chapter 4 presented the detailed design for interface and test circuits for a full system. In this chapter, the physical layout of a test chip, containing a 64-bit FFT, is presented. This was implemented to demonstrate the FFT as a stand-alone system. The block diagram of the chip is shown in Figure 5.1. In the center of the chip is a 64-bit FFT core, implemented using the circuits introduced in the 'FFT Core' section of Chapter 3. The input interface is the same as that designed in Chapter 4 (but scaled down for a 64-bit FFT, instead of the 256-bit FFT used in Chapter 4). The analog values are supplied using DACs, which are included on-chip. The output is the FFT test circuit introduced in Chapter 3.

Sections 5.1, 5.2, and 5.3 each describe a section of the FFT system: the FFT core, the input interface, and the FFT test circuit respectively. Section 5.4 describes some additional components included on-chip for characterization using bench equipment, and then Section 5.5 describes the actual fabrication process. Section 5.6 outlines the test methods for the FFT system, and then Section 5.7 summarizes the chapter.

5.1. FFT Core Implementation

The core was implemented using the standard cells introduced in Chapter 4; the schematic is available in [5]. The place-and-route (PAR) tools had difficulty fully processing the full sixty-four bit FFT; it was slow and frequently crashed. To reduce the design complexity, a 16-bit FFT unit was first laid out, then used as a block in the larger system. Recall that an FFT processor is constructed recursively: a 64-bit FFT contains, among other processing elements, two 32-bit FFTs, and a 32-bit FFT contains (among other processing elements) two 16-bit FFTs. As a result there are a total of four 16-bit FFTs in the core.

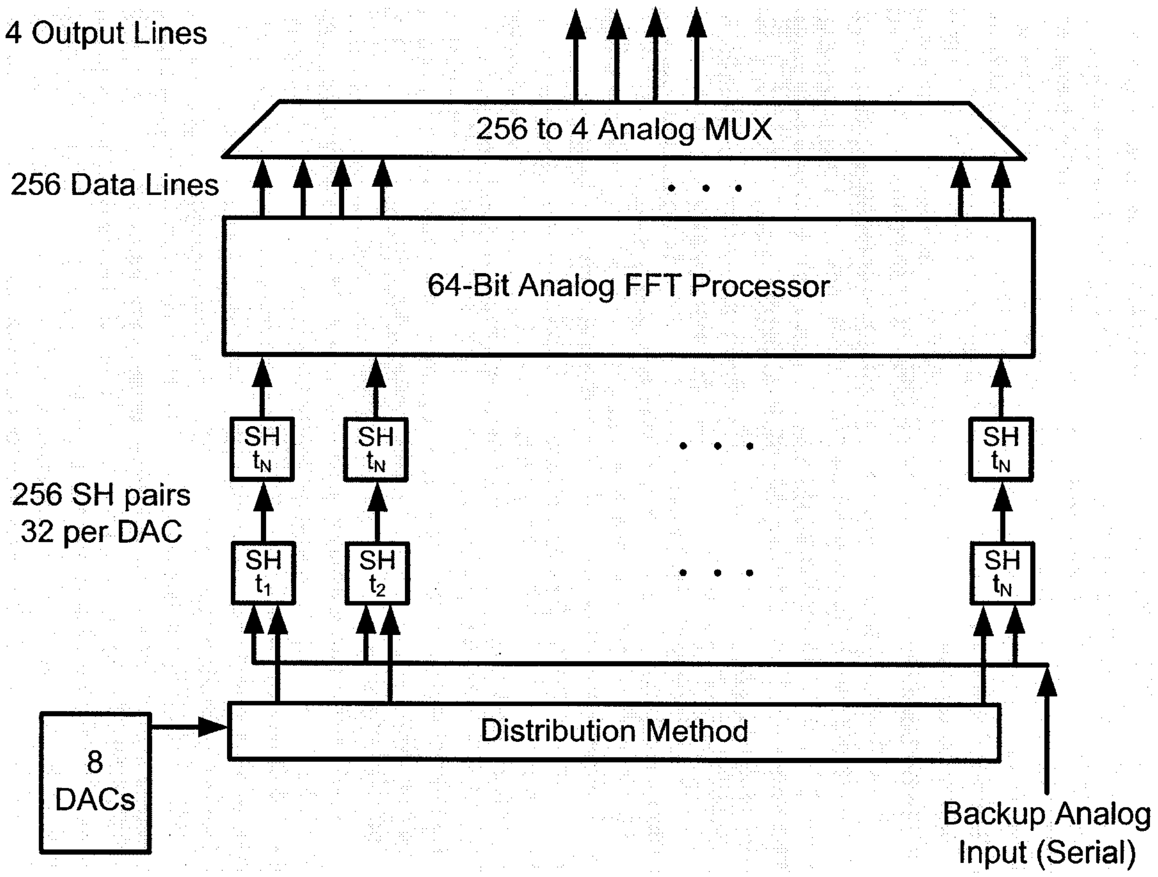


Figure 5.1: Test Chip Block Diagram

The layout of the 16-bit FFT is shown in Figure C.3. As explained in Chapter 4, the basic cells are NMOS and PMOS current mirrors. These are separated by large power supply stripes; these alternate between V_{DD} and V_{SS}. The total core layout can be seen in Figure C.2. Its size as implemented was 792μm X 977μm, or just less than 0.8mm^2. The original area estimate, using the same assumptions as in Chapter 3, for a 64-bit FFT was

$$A_{64-BIT} = 1.1 * \left[528 \left(\frac{38.5 * 11.7}{2} + \frac{38.7 * 12.7}{2} \right) \right] \approx 1.1 mm^2 . \tag{5.1}$$

This is reasonably close to reality.

To design the power wiring, an estimate of the FFT core's power consumption was required. To do so, consider that an FFT consists of columns of current mirrors, each of which duplicates its respective input current. For this analysis, ignore the weighting factor cells: given this, each FFT processor then consists of a single column of current mirrors, and then two FFT processors of half as many bits. Starting from the smallest FFT processor, a two-bit FFT consists of only one pair of current mirrors: in the two-bit case the input currents are simply doubled and then added. Since the adding does not change the current magnitude, it is ignored; the result is that the total current consumption is the currents at the input plus twice the input currents at the output. This can be represented as

$$I_{2-BIT} = \sum_{n=1}^{4*2} I_n + 2\sum_{n=1}^{4*2} I = 3\sum_{n=1}^{4*2} I_n . \tag{5.2}$$

where the summation represents the sum of all the input currents, at four currents per bit.

A four-bit FFT doubles its input currents, and then feeds them into a pair of two-bit FFTs. For a worst-case scenario, the individual input currents I_n will be considered to be identical, at the maximum possible input magnitude. Since the currents being fed into each two-bit FFT have already been summed, they are each double the input current. Thus the total amount of current used by a four-bit FFT is equal to

$$I_{4-BIT} = \sum_{n=1}^{4*4} I_n + 2\sum_{n=1}^{4*4} I + 2\left(2I_{2-BIT}\right) \tag{5.3}$$

$$= 7\sum_{n=1}^{4*2} I_n . \tag{5.4}$$

This makes use of the assumption of identical values of I_n: as a result

$$2\sum_{n=1}^{2*2} I_n = \sum_{n=1}^{4*2} I_n . \tag{5.5}$$

This pattern continues: with the doubling of the number of bits processed, the total current draw grows according to the description

$$I_{N_{BIT}} = 2N_{BIT} \cdot \sum_{n=1}^{4N_{BIT}} I_n , \tag{5.6}$$

where N_{BIT} is the total number of bits in the FFT. This allows us an estimate of the overall draw of the 64-bit FFT. Assuming input values of 10nA,

$$I_{64} = 2 \cdot 64 \cdot \sum_{n=1}^{4*64} 10^{-6} \tag{5.7}$$

$$= 1.3mA. \tag{5.8}$$

The expected power consumption can then be quickly calculated, assuming a supply voltage of 1.8V.

$$P_{64} = 1.8 \cdot I_{64} = 2.4mW. \tag{5.9}$$

This is certainly an upper bound; both the weighting factors and variances in I_n will reduce the overall power consumption.

As this is at best a loose estimate, the supply line wires were conservatively chosen to be 12 μm wide, routed on the Metal 1 layer, which outside the FFT cells was dedicated to power and ground routing. The sheet resistance of this layer according to MOSIS parameters [71] is 0.08 ohms/square. Using the wire resistance equation given in [63]

$$R = R_{SQ} \frac{L}{W} \Omega, \tag{5.10}$$

this translates to a power line resistivity of

$$\rho_{VDD} = 6.67 \Omega/_{mm}. \tag{5.11}$$

Then the resistance for a single power strip in the FFT core, approximately 950 μm in length, is 6 Ω. Post-layout extraction confirmed this value. Approximating 300 μm in distribution wiring from the power pin to the core, equal to 2 Ω, the total expected resistance from pad to the furthest point is approximately 8 Ω. The maximum possible voltage drop is then

$$V_{DD_{DROP}} = I_{64} \cdot 8\Omega = 10mV. \tag{5.12}$$

The remainder of the wiring was completed using Cadence's IC Craftsman PAR tool. Due to the low speed of the core's operation, resistive parasitics in the signal wiring is of no concern (according to the authors of [72], resistive parasitics need only be accounted for in circuits operating above 1GHz). An analysis of the effect of parasitic capacitance on the FFT's performance is to be included in [10].

5.2. Input Interface Implementation

The full input interface is shown in Figure B.1 (schematic) and C.6 (layout). It consists of eight five-bit DACs, each of which drives 32 SH units. Because of the SH capacitor sizes, they could not all be arranged in a single line; as a result they are staggered as shown in Figure C.6.

Several transmission gates were required for the layout of this chip. A single transmission gate was designed and used in all cases, with some minor modifications for use in the output multiplexer. The sizing was as chosen for TGs in Chapter 4. The schematic for this gate is in Figure B.10, and the layout image is Figure C.15. The sizes used are shown in Table 5.1.

Table 5.1: Transmission Gate Sizing

Cell	W/L Ratio
Transmission Gate PMOS	1500nm/180nm
Transmission Gate NMOS	1500nm/180nm

All shift registers in the chip were implemented using the same basic cell (the schematic of a single bit register is shown in Figure B.9, and the layout of a five bit shift register is shown in Figure C.11). The cells were chosen to have the sizes as listed in Table 5.1 and 5.2. The TGs were reused from the analog multiplexer, and the PMOS and NMOS were sized identically so the cell would form a rectangular shape.

Table 5.2: Shift Register Device Sizing

Cell	W/L Ratio
Inverter PMOS	500nm/180nm
Inverter NMOS	500nm/180nm

The same basic cell was used in all 'register' instances throughout the chip. This includes the input shift register (only the positive output was used in this case), and the input and output MUX control shift registers.

The DAC layout is in Figure C.10. Layout was completed using Tanner Tools at Utah State University (USU). I released the layer-encoding map (cmosp18.strmMapTable) to the USU group, who exported their design as a GDSII file. This was then imported into Cadence using the same layer map file. Because USU is not party to CMC's license agreements, it was laid out using slightly different DRC rules from the remainder of the system (USU used TSMC rules available through MOSIS). Because the issuer of both rules was TSMC, we had hoped the rules would be quite similar. However, upon importing the file we found that, despite being DRC-clean according to the MOSIS rules, over 1200 DRC errors were present using the CMC rules. We first inquired with TSMC (through CMC) whether we could intermix the DRC files, submitting the design as-is, given that the DAC was in fact DRC-clean with the MOSIS rules. The answer was negative. Given this the designer was left to eliminate the errors by hand (as releasing the DRC rules to USU would violate the terms of the CMC NDA). Once this was completed the U of A team sent the USU team an extracted SPICE netlist to ensure that the DAC was still operational. USU confirmed, and the DAC was integrated with the remainder of the circuit.

One DAC-SH unit is shown in Figure B.3, B.4, B.5, and B.6 (schematics) and C.8 (layout). The input of one DAC unit is a five-bit shift register; once this is loaded, pulsing the DAC clock causes the value to be placed on the DAC output. Once the DAC clock is driven low, the value placed on the clock is

$$V_{LOW} = V_{REF} - V_{HIGH}. \tag{5.13}$$

Since differential values are loaded onto sequential SHs, this reduces the amount of values that must be fed into the DACs.

The DAC – SH distribution is a single line with one TG tap for each SH (The Bus method of distribution as introduced in Chapter 2 and discussed in Chapter 4). There is also a pair of backup gates that allow for analog input. The control line for this unit is connected to a pair of transmission gates; if this signal is high, they allow the DAC voltage to drive the signal line, and if the signal is low, they allow the input pins to drive the signal line (these are shown in Figures B.6 and C.9). There are four inputs, so they must be demultiplexed to the eight DAC units (this is shown in Figure B.2 and C.7).

5.3. FFT Test Circuit Implementation

As introduced in Chapter 3, the FFT test circuit consists of an analog multiplexer. As there was only one output from the FFT, mirroring, as discussed in Chapter 3, was not required. The implementation of this unit is discussed in this section.

The basic unit for the MUX is the four-bit transmission gate, whose schematic is shown in Appendix B.13 and layout shown in C.18. It is itself based on the standard transmission gate, whose schematic and layout are shown in Figures B.10 and C.15 respectively. These are linked together to create an analog multiplexer with a ratio 32:4 (see B.12 and C.17). If these are nested, they create a multiplexer with ratio 256:4, which is what is required for this chip (see B.11 and C.16 for the schematic and layout respectively).

This smallest block has only one control signal (and its inverse), which runs vertically over the four transmission gates. In order to avoid antenna errors, the control line was run on Metal 3, then at each transmission gate the connection goes up through Metal 4 before being connected to the gates through a stack of vias leading from Metal 4 to Poly.

The layout of these control signals was the area-limiting factor. Because we want similar rise and fall times, it was important to layout all the signals on the same metal layer. The total metal layout was chosen as in Table 5.3.

Table 5.3: Analog Multiplexer Metal Use

Layer	Use
Metal 1	Power/Ground
Metal 2	Input/Output
Metal 3	Control Signals
Metal 4	Control Signal Connections

It should be noted that actual rise/fall time of the control signals is not a key design measure, because the output multiplexer is set only once each FFT process cycle.

5.4. Test Components

To allow for characterization of their properties, four individual components were placed on-chip. These consisted of one NMOS current mirror, one PMOS current mirror, one sample and hold element (the layout image is Figure C.4) and one section of the dynamic comparator discussed in Chapter 4 (the layout image is Figure C.5). These are all included in Figure 5.2. They are also powered independently and are accessible directly from pins, so can be fully characterized using lab equipment. Finally, two extra pins were tied together, to measure the capacitance of the analog pin used. To find the pin's capacitance, measure the total capacitance of the path between the two pins and divide by two.

This completes the outline of all the components contained on-chip. The next section describes the actual fabrication of the chip, and physical modifications required to fit the chip to the manufacturer's specifications.

5.5. Chip Fabrication

This chip was submitted to Taiwan Semiconductor Manufacturing Company (TSMC) through CMC Microsystems on August 9, 2006. The process is a 0.18μm six-metal, one-poly (6M1P) process with mixed-metal options (deep N-well, MiM-capacitors, and thick-Metal6 for inductors). The CMC Systems run number is 0603CF. The chip was implemented on a 2mm x 2mm die; the floor plan is shown in Figure 5.2, and the full chip layout image is Figure C.1. The minimum bonding pitch available is 70 μm; to reduce crosstalk, this chip used a bonding pitch of 80 μm.

As is typical in silicon designs, the final step before submission was to ensure a minimum density for each layer. For the Poly and Metal 1-6 layers, this was easily completed using the Diva Layer Fill script made available by CMC. However, there was also a minimum layer density for the 'CTM' layer, the mixed-signal layer required for the capacitors. This layer has specific requirements, in particular:

- CTM must be under laid with Metal 5, and
- No layer can be under the CTM/Metal 5 combination.

Because of these rules, the standard layer fill script did not effectively fill the CTM layer. As a result, dummy capacitors with both CTM and Metal 5 tied to ground were placed in empty locations in the chip to fill CTM to the minimum density. Once this was complete, the remainder of the chip was filled with the other seven layers.

The chip will return in a 120CQFP package. As the cavity size in this package is 8.13 mm square, and this chip is 2 mm square, there was some question as to whether or not the bonding service will be able to complete this order. Regardless of package, when it does return the FFT system will need to be tested, according to the test plan in the next section.

5.6. Test Plan

The first stage of characterization involves characterizing the test units. In particular, we can measure the actual facts of data feedthrough and clock feedthrough on the capacitor voltage. Likewise, the comparator can be tested for functionality. In addition, the through current mirrors on-chip can be characterized using lab equipment.

Next to be tested are the multiplexer registers. On both the output multiplexer and the SH units, there are shift registers to control the data flow. These are fed by their respective ‘START’ pins, and the outputs are visible through the ‘TEST_OUT’ pin. This will assure the functionality of the shift registers themselves; nothing more.

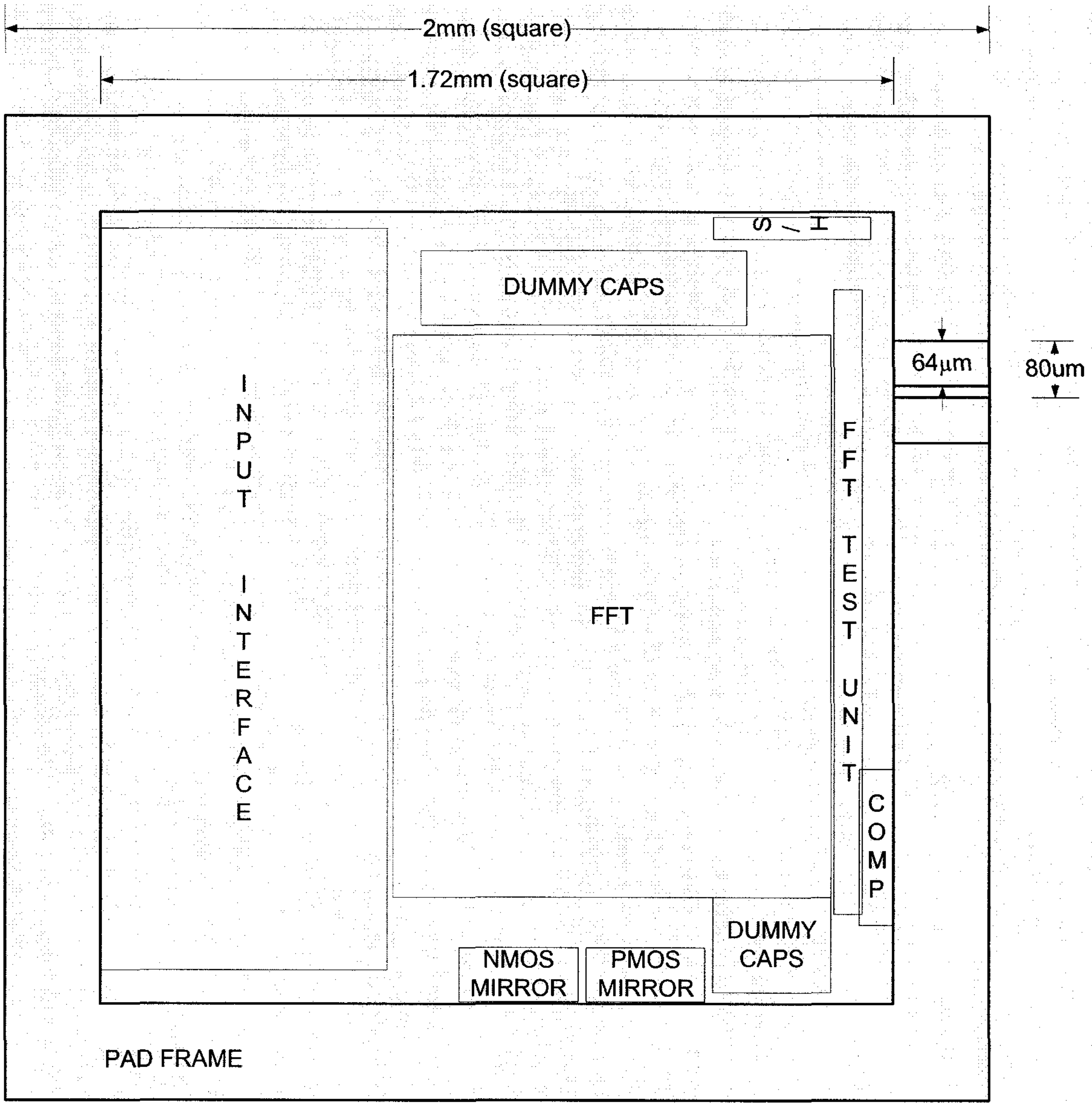


Figure 5.2: Chip Floorplan (As Implemented)

From this step, the functional testing of the FFT can begin. The testing is run as follows:

1. BKUP_CTL is set to one (1.8V) to allow the DACs access to the SHs.
2. The one is placed at the input of the START lines.
3. The MUX clock signals are pulsed through to place the one in the first register.
4. Five bits are fed serially into the input lines, using the IP_IN and IP_CLK pin.
5. DAC_CLK is set high to advance the values into the DAC.

6. The IP_MUX clock signal is pulsed to advance the control to the next SH.
7. Steps three-five are repeated 31 more times.
8. The first four outputs can then be viewed. Only four outputs can be viewed during any one cycle. To view all outputs for the given input pattern, the same pattern must be loaded 64 times in total.

If no outputs are visible, this may be because the DACs are faulty. In that case,

1. BKUP_CTL is set to zero to allow the external pins access to the SHs.
2. IP_MUX_CTL is set to zero to allow access to the first four DAC units.
3. The one is placed at the input of the START lines.
4. The MUX_CLK is pulsed to advance the shift register
5. The MUX clock signals are pulsed through place the one in the first register.
6. Four analog values are placed at the analog in pins.
7. Steps 4-5 are repeated 15 more times.
8. IP_MUX_CTL is set to one to allow access to the second four DAC units.
9. Steps 4-5 are repeated 15 more times.

In this way, the entire FFT system can be characterized.

When testing, one must consider what makes a system a 'success'. In general, if the system is able to produce an output whereby one can calculate what the input was, that could be, at least from a VLSI standpoint, considered to be a 'successful' circuit. The much more important question is whether the output corresponds to a Fast-Fourier Transform of the input data. It is, of course, unlikely that the output will be a perfect FFT of the input data. However, like in all analog testing, if the data is within some range it can still receive a passing grade.

In terms of characterizing the system, a system of linear equations can be developed to determine the actual weighting factors within the system. Simple comparisons of suitable numbers of input and output vectors will allow the tester to get an accurate picture of mismatch within the weighted current mirrors.

Recall in Chapter 2 it was mentioned that Dai et al. [40] considered AD device mismatch to be a form of channel noise, and demonstrated a means to refer it to the input of the decoder. The general idea upon which this approach is founded is that the very purpose of a channel coding system is to remove errors, so there already exists the means to combat these parametric deviations from the ideal. These deviations need only be characterized as another source of noise in the communications system shown in Figures 2.2 and 2.5. If such a characterization can be made, then the system should be considered a 'success'. Whether the actual magnitude of this noise is acceptable would depend on its relative value compared to the magnitude of noise introduced by the other steps in the chain.

5.7. Chapter Summary

The layout and physical considerations of implementing a 64-bit FFT system were covered in this chapter. The system consists of a 64-bit FFT core, 256 SH units, and a 256:4 output multiplexer, each as introduced in Chapter 3. In addition, one SH unit, one comparator section, and two current mirrors were included on-chip for characterization. This was fabricated through CMC Microsystems run 0603CF.

Chapter 6

Conclusions and Future Work

A design methodology for the interface of analog decoders is demonstrated. This extends the body of research on analog decoders by focusing on their interface with the remainder of a communication system. This thesis introduces three circuits for analog decoders: an analog FFT processor, a sample and hold, and a comparator, that are appropriate for this application. It also introduces a test circuit for each of these circuits, designed to give designers the ability to characterize their designs. Furthermore, a design methodology was demonstrated for a full FFT-AD system; this methodology was partially used in the implementation of a 64-bit FFT processor chip.

Figure 5.2 demonstrates the largest concern facing analog decoders. The input interface constitutes almost a quarter of the total die area excluding bond pads. This area is dominated by the MiM capacitors; it would be much more space-efficient if these capacitors could be removed outright. At the least, an investigation into more area-efficient implementations of capacitors would be a worthwhile endeavor. Even before pursuing this, a good project would be to gain an understanding of the system's requirements. How much voltage droop can occur before the data is irretrievably corrupted? The answers to these are unlikely to be single numbers, but rather there likely exist relationships that need to be understood. Perhaps a future researcher will find that the current emphasis on voltage droop is altogether unnecessary, and find that the SH capacitance values can be much smaller than they are now.

The output interface poses design issues of its own. Comparator mismatch is problematic, and including compensation circuits would give the output interface as many size problems as the input interface. Some work, such as in [68], has attempted to compensate for this mismatch in other innovative ways. It is not clear if this method has any application in ADs, but the general concept may be applicable. For example, rather than implement one comparator for each output bit, a designer may choose to implement, say, 16, and use a windowing technique, perhaps using SHs, to evaluate the bits. This allows larger, more complex, comparators to be implemented without increasing the output interface's area requirements. The designer could then also implement a BIST technique where 18 comparators are actually fabricated; they are then tested, and the two with the worst offset are rejected and not used. Since the data is already being shuffled for the windowing technique, adding this capability should not be much harder.

In this FFT implementation each succeeding column doubles the total magnitude of the currents. This is problematic, as in large FFTs the transistors at the tail end will be processing currents orders of magnitude larger than the transistors near the input. This is certainly undesirable, from both a characterization and a power consumption point of view. The simple solution is to halve the current mirror weighting – rather than mirror the current, each mirror would reproduce 50% of the input. Weighted current mirrors would produce 50% of the weighted value. This should keep the total current values constant; some analysis would be required to ensure that mismatch concerns will not be magnified in this configuration. More fundamentally, it should be asked whether the FFT is most easily implemented, or if it would be easier to implement the original DFT equation in analog. The FFT is commonly used, as it is easily implemented in digital logic, but there is no precedent proving that the FFT is easier to implement in analog than the DFT.

One difficulty faced in this thesis is that there is no easy way to characterize an analog system on the order of even a (256,121) system. There exist modeling techniques such as that in [73] for RF design; these could hopefully find application in ADs.

More broadly, analog decoders represent the application of information theoretic concepts. Applications of information theoretic concepts to other problems include applications to analog circuit design [68] [37] and even genomic analysis [74]. This suggests that there is a wide range of applications for these concepts. A further push to integrate these concepts with other design methods would be invaluable.

Bibliography

[1] R. Togneri and C. deSilva, *Fundamentals of Information Theory and Coding Design.* Champan & Hall, 2002, 384 pp.

[2] C. Schlegel and L. Perez, *Trellis and Turbo Coding.* IEEE Press, 2004, 380 pp.

[3] F. Lustenberger, M. Helfenstein, G. S. Moschytz, H. A. Loeliger and F. Tarkoy, "All–analog decoder for a binary (18,9,5) tail–biting trellis code," *European Solid-State Circuits Conference,* 1999, pp. 362-365.

[4] M. Yiu, V. C. Gaudet, C. Schlegel and C. Winstead, "Digital built-in self-test of CMOS analog iterative decoders," *IEEE International Symposium on Circuits and Systems,* 2005, pp. 2204-2207.

[5] N. Sadeghi, "Analog Current Mode Fast-Fourier Transform Systems," MSc. Thesis, Unpublished.

[6] G. Durgin, *Space-Time Wireless Channels.* Pearson Education, 2003, 345 pp.

[7] B. Lathi, "Appendix A: Orthogonality of some signal sets," *Modern Digital and Analog Communication Systems*, 3rd ed., Oxford University Press, 1998, pp. 764-765.

[8] J. Cooley and J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation,* vol. 19, pp. 297--301, 1965.

[9] N. Sadeghi, "$(16,11)^2$ Block Turbo Decoder Decoder with FFT (Design Review Presentation)," March 2006.

[10] N. Sadeghi, H. Nik, C. Schlegel and V. C. Gaudet, "Analog FFT interface for ultra-low power analog receiver architectures," *Analog Decoding Workshop,* Turin, Italy, 2006.

[11] R. Wiegerink, *Analysis and Synthesis of MOS Translinear Circuits.* Boston: Kluwer Academic Publishers, 1993, 156 pp.

[12] B. Gilbert, "Current-mode circuits from a translinear viewpoint: A tutorial," *Analogue IC Design: The Current-Mode Approach* C. Toumazou, F. J. Lidgey and D. G. Haigh, Eds. London: Peter Peregrinus, 1990, pp. 12-92.

[13] B. Gilbert, "A precise four-quadrant multiplier with subnanosecond response," *IEEE Journal of Solid-State Circuits,* vol. 3, pp. 365-373, 1968.

[14] C. Winstead, "Analog Iterative Error Control Decoders," PhD. Thesis, University of Alberta, 2004, 251 pp.

[15] M. H. Shakibi, D. A. Johns and K. W. Martin, "A 200 MHz 3.3 V BiCMOS class-IV partial-response analog viterbi decoder," *IEEE Custom Integrated Circuits Conference,* 1995, pp. 567-570.

[16] R. C. Davis and H. A. Loeliger, "A nonalgorithmic maximum likelihood decoder for trellis codes," *IEEE Transactions on Information Theory,* vol. 39, pp. 1450-1453, 1993.

[17] H. A. Loeliger, F. Lustenberger, M. Helfenstein and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Transactions on Information Theory,* vol. 47, pp. 837-843, 2001.

[18] J. Hagenauer, "Der analogs decoder," Germany German Patent 19-725-275.3, Filed June 1997.

[19] J. Hagenauer and M. Winklhofer, "The analog decoder," *IEEE International Symposium on Information Theory,* 1998, pp. 145.

[20] H. A. Loeliger, F. Lustenberger, M. Helfenstein and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Transactions on Information Theory,* 1998, pp. 146.

[21] M. Moerz, T. Gabara, R. Yan and J. Hagenauer, "An analog 0.25 μm BiCMOS tailbiting MAP decoder," *IEEE International Solid-State Circuits Conference,* 2000, pp. 356-357.

[22] C. Winstead, J. Dai, S. Yu, C. Myers, R. Harrison and C. Schlegel, "CMOS analog MAP decoder for an (8,4) Hamming code," *IEEE Journal of Solid-State Circuits,* vol. 39, pp. 122-131, 2004.

[23] C. Winstead, V. C. Gaudet and C. Schlegel, "Analog iterative decoding of error control codes," *IEEE Canadian Conference on Electrical and Computer Engineering,* 2003, pp. 1539-1542.

[24] C. Winstead, N. Nguyen, V. C. Gaudet and C. Schlegel, "Low-voltage CMOS circuits for analog iterative decoders," *IEEE Transactions on Circuits and Systems I,* vol. 53, pp. 829-841, 2006.

[25] D. Vogrig, A. Gerosa, A. Neviani, A. G. Amat, G. Montorsi and S. Benedetto, "A 0.35-μm CMOS analog turbo decoder for the 40-bit rate 1/3 UMTS channel code," *IEEE Journal of Solid-State Circuits,* vol. 40, pp. 753-762, 2005.

[26] V. C. Gaudet and P. G. Gulak, "A 13.3-Mb/s 0.35-μm CMOS analog turbo decoder IC with a configurable interleaver," *IEEE Journal of Solid-State Circuits,* vol. 38, pp. 2010-2015, 2003.

[27] V. C. Gaudet, R. J. Gaudet and P. G. Gulak, "Programmable interleaver design for analog iterative decoders," *IEEE Transactions on Circuits and Systems II,* vol. 49, pp. 457-464, 2002.

[28] S. Hemati, A. H. Banihashemi and C. Plett, "A 0.18-µm CMOS analog min-sum iterative decoder for a (32,8) LDPC code," *IEEE Journal of Solid-State Circuits,* vol. 41, pp. 2531-2540, 2006.

[29] UC Berkeley, Dept. of EECS, "SPICE (Simulation Package with Integrated Circuit Emphasis)," 1975.

[30] S. Hemati and A. H. Banihashemi, "Comparison between continuous-time asynchronous and discrete-time synchronous iterative decoding," *IEEE Global Telecommunications Conference,* 2004, pp. 356-360.

[31] S. Hemati and A. H. Banihashemi, "On the dynamics of continuous-time analog iterative decoding," *IEEE International Symposium on Information Theory,* 2004, pp. 262.

[32] V. S. S. A. Devarakonda and C. Winstead, "Accuracy of dynamical models for analog iterative error control decoders," *IEEE Midwest Symposium on Circuits and Systems,* 2005, vol. 2, pp. 1506-1509.

[33] H. Kahn, "Random sampling (Monte Carlo) techniques in neutron attenuation problems—I," *Nucleonics*, pp. 27-37, May 1950.

[34] C. Winstead and C. Schlegel, "Importance sampling for SPICE-level verification of analog decoders," *IEEE International Symposium on Information Theory,* 2003, pp. 103.

[35] M. Ferrari and S. Bellini, "Importance sampling simulation of concatenated block codes," *IEEE International Conference on Communications,* vol. 147, pp. 245-251, 2000.

[36] P. J. Smith, M. Shafi and H. Gao, "Quick simulation: a review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Communications,* vol. 15, pp. 597-613, 1997.

[37] C. Winstead and C. Schlegel, "Density evolution analysis of device mismatch in analog decoders," *IEEE International Symposium on Information Theory,* 2004, pp. 293.

[38] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits,* vol. 24, pp. 1433-1439, 1989.

[39] F. Lustenberger and H. A. Loeliger, "On mismatch errors in analog-VLSI error correcting decoders," *IEEE International Symposium on Circuits and Systems,* 2001, vol. 4, pp. 198-201.

[40] J. Dai, "Design Methodology for Analog VLSI Implementations of Error Control Decoders," PhD. Thesis, University of Utah, 2001, 207 pp.

[41] M. Helfenstein, F. Lustenberger, H. A. Loeliger, F. Tarkoy and G. S. Moschytz, "High-speed interfaces for analog, iterative VLSI decoders," *IEEE International Symposium on Circuits and Systems,* 1999, vol.2, pp. 428-431.

[42] A. Sedra and K. Smith, *Microelectronic Circuits.* ,4th ed.New York: Oxford University Press, 1998, 1024 pp.

[43] Z. Lao, A. Thiede, H. Lienhart, M. Schlechtweg, W. Bronner, J. Hornung, A. Hulsmann and T. Jakobus, "5 Gsample/s track-hold and 3 Gsample/s quasi-sample-hold ICs," *IEEE International Solid-State Circuits Conference,* 1998, pp. 328-329.

[44] C. Toumazou, J. B. Hughes and N. C. Bassersby, *Switch-Currents: An Analogue Technique for Digital Technology.* London: Peter Peregrinus, 1993, 595 pp.

[45] K. Leclavattananon and C. Toumazou, "Switched-voltage: an adaptation of switched-currents for voltage-mode design," *Electronics Letters,* vol. 34, pp. 503-504, 1998.

[46] H. Matsumoto, K. Murao and K. Ohno, "A switched-voltage high-accuracy sample/hold circuit," *IEEE Midwest Symposium on Circuits and Systems,* 2004, pp. I-105-8.

[47] R. Pallas-Areny and J. G. Webster, *Analog Signal Processing.* New York: John Wiley & Sons, Inc., 1999, 586 pp.

[48] M. Nayebi and B. A. Wooley, "A 10-bit video BiCMOS track-and-hold amplifier," *IEEE Journal of Solid-State Circuits,* vol. 24, pp. 1507-1516, 1989.

[49] B. Razavi, *Design of Analog CMOS Integrated Circuits.* Boston: McGraw-Hill, 2000, 684 pp.

[50] C. Winstead, A. Dai, S. Yu, R. Harrison, C. Myers and C. Schlegel, "Analog decoding of product codes," *International Symposium on Information Theory,* 2002, pp. 230.

[51] J. M. Martins and V. F. Dias, "Analysis of clock feedthrough effects in switched-current cells," *Design of Circuits and Integrated Systems,* 1997, pp. 253-258.

[52] A. T. K. Tang and C. Toumazou, "High performance CMOS current comparator," *Electronics Letters,* vol. 30, pp. 5-6, 1994.

[53] C. Toumazou, F. J. Lidgey and D. G. Haigh, *Analogue IC Design: The Current-Mode Approach.* London: Peter Peregrinus, 1990, 646 pp.

[54] H. Traff, "Novel approach to high speed CMOS current comparators," *Electronics Letters,* vol. 28, pp. 310-312, 1992.

[55] W. J. Marble, "A Low-Power Precision Dynamic Comparator in Submicron CMOS," MSc. Thesis. August 1999, 60 pp.

[56] F. Chen, S. Ramaswamy and B. Bakkaloglu, "A 1.5V 1mA 80dB passive sigma delta ADC in 0.13 μm digital CMOS process," *IEEE International Solid-State Circuits Conference*, 2003, pp. 54-477.

[57] S. Yu, "Design And Test Of Error Control Decoders In Analog CMOS," PhD. Thesis, University of Utah, 2003, 124 pp.

[58] L. Samid, P. Volz and Y. Manoli, "A dynamic analysis of a latched CMOS comparator," *IEEE International Symposium on Circuits and Systems,* 2004, pp. 181-184.

[59] C. Fayomi, G. Roberts and M. Sawan, "Low power/low voltage high speed CMOS differential track and latch comparator with rail-to-rail input," *IEEE International Symposium on Circuits and Systems,* 2000, pp. 653-656.

[60] K. Moolpho, J. Ngarmnil and S. Sitjongsataporn, "A high speed low input current low voltage CMOS current comparator," *IEEE International Symposium on Circuits and Systems,* 2003, pp. I-433-436.

[61] L. Ravezzi, D. Stoppa and G. F. Dalla Betta, "Simple high-speed CMOS current comparator," *Electronics Letters,* vol. 33, pp. 1829-1830, 1997.

[62] G. Palmisano and G. Palumbo, "High performance CMOS current comparator design," *IEEE Transactions on Circuits and Systems II,* vol. 43, pp. 785-790, 1996.

[63] J. Rabaey, A. Chandrakasan and B. Nikolic, *Digital Integrated Circuits.* 2nd ed. Prentice-Hall, 2002, 759 pp.

[64] P. Uthaichana and E. Leelarasmee, "Low power CMOS dynamic latch comparators," *Conference on Convergent Technologies for Asia-Pacific Region,* 2003, pp. 605-608.

[65] C. Palmisano and G. Palumbo, "Offset compensation technique for CMOS current comparators," *Electronics Letters,* vol. 30, pp. 852-854, 1994.

[66] A. Worapishet, J. B. Hughes and C. Toumazou, "An improved CMOS offset-compensated current comparator for high speed applications," *IEEE International Symposium on Circuits and Systems,* 1998, pp. 535-538 vol.1.

[67] C. Petrie, T. Sun and M. Miller, "A high-gain offset-compensated differential amplifier," *IEEE International Symposium on Circuits and Systems*, 2004, pp. I-489-92.

[68] M. Frey and H. A. Loeliger, "On flash A/D-converters with low-precision comparators," *IEEE International Symposium on Circuits and Systems,* 2006, pp. 3926-3929.

[69] L. T. Wang, C. W. Wu and X. Wen, *VLSI Test Principles and Architectures: Design for Testability.* San Francisco: Morgan Kaufman, 2006, 808 pp.

[70] M. Hafed, A. Abaskharoun and G. Roberts, "A 4-GHz Effective Sample Rate Integrated Test Core for Analog and Mixed-Signal Circuits," *IEEE Journal of Solid-State Circuits,* vol. 37, pp. 499-514, 2002.

[71] The MOSIS Service, "Wafer Electrical Test Data and SPICE Model Parameters: TSMC CL018/CR018/CM018 (0.18 μm): http://www.mosis.com/Technical/Testdata/tsmc-018-prm.html", 2006.

[72] A. Agarwal, H. Sampath, V. Yelamanchili and R. Vemuri, "Accurate estimation of parasitic capacitances in analog circuits," *IEEE Design, Automation, and Test in Europe,* 2004, pp. 1364-1365.

[73] J. Xu, M. C. E. Yagoub, R. Ding and Q. J. Zhang, "Neural-based dynamic modeling of nonlinear microwave circuits," *IEEE Transactions on Microwave Theory and Techniques,* vol. 50, pp. 2769-2780, 2002.

[74] J. Hagenauer, Z. Dawy, B. Gobel, P. Hanus and J. Mueller, "Genomic analysis using methods from information theory," *IEEE Information Theory Workshop,* 2004, pp. 55-59.

[75] IEEE Std. 802.11A-1999, IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band.

[76] IEEE Std. 802.11G-2003, Telecommunications and Information Exchange Between Systems - Local And Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[77] IEEE Std. 802.16E-2005 IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands.

Appendix A

Matlab Scripts Used in Interface Design

```
%NumOfDACs.m
%Calculates number of DACs required for a given throughput

BitsPerFrame = 121; %Assuming a (256,121) code
BitsPerDAC=[8 16 32 48 64 96]; %Sweeping across these possibilities
BitsPerSecond=[3e7 1e7 5e6 1e6]; %Likewise, considering these speeds
NumOfDACs = zeros(4,6);
ClkFreq = zeros(4,6);

for i = 1:4 %One iteration for each line on the graph
  bps = BitsPerSecond(i); %Set the rate for this iteration
  FrameSpeed = bps./BitsPerFrame;
  TimePerFrame = 1./FrameSpeed;
  NumOfDACs=1024./BitsPerDAC;
  ClkPeriod = TimePerFrame./BitsPerDAC;
  ClkFreq(i,:) = 1./(ClkPeriod*1e6);  %in MHz
end

hold on;
plot(NumOfDACs, ClkFreq(1,:), '--');
plot(NumOfDACs, ClkFreq(2,:), ':');
plot(NumOfDACs, ClkFreq(3,:), '-');
plot(NumOfDACs, ClkFreq(4,:), '.-');
xlabel('Number of DACs used');
ylabel('Required Clock Frequency (MHz)');
legend ('30Mbps','10Mbps','5Mbps','1Mbps');
```

```
%SH_optimize.m
%Calculates the maximum clock frequency for two distribution methods

%----------------------------------------------------------------------
%Process Constants from MOSIS Spice Parameters
%----------------------------------------------------------------------
mu0_N=270.3885488/1e4;  %m^2/Vs
mu0_P=113.5996838/1e4;
Cgdo_N=9.05e-10;  %in F/m - multiply by W to get the answer
Cgdo_P=6.59e-10;
Cgso_N=9.05e-10;  %in F/m - multiply by W to get the answer
Cgso_P=6.59e-10;
Cj0_N=1.002472e-3;
Cj0_P=1.154588e-3;     %in F/m^2 - muliply by area to get answer
Cjsw_N=2.529156e-10;
Cjsw_P=2.429929e-10;  %in F/m
Vt_N = 0.52;
Vt_P = 0.51;  %Use Absolute Value
K_N = 172.3e-6;
K_P = 65.8e-6;
Cox_N=(K_N*2)/(mu0_N);       %k = (UnCox)/2; solve for Cox
Cox_P=(K_P*2)/(mu0_P);       %Cox uses cm^2
R_sq = 0.08; %Ohms/Square

%----------------------------------------------------------------------
%Design Variables
%----------------------------------------------------------------------
%Transmission gate sizing
W_N=0.6e-06;
L_N=0.18e-06;  %m
W_P= W_N;  %Sizing them the same to minimize pedestal error
L_P=0.18e-06;  %m

BitsPerDAC=32;
NumOfDACs=1024./BitsPerDAC;
C_cap = [1e-15 10e-15 50e-15 100e-15 150e-15]; %F
Vin = 1.8; % assume the worst case

%----------------------------------------------------------------------
%Capacitance Calculations
%----------------------------------------------------------------------
C_db_N = Cj0_N*(W_N*L_N)+Cjsw_N*(2*L_N+W_N);
C_sb_N = C_db_N;
C_dg_N_OFF = Cgdo_N*W_N;  %Just C_gd, for the 2nd transistor
```

```
C_g_N = W_N*L_N*Cox_N + Cgso_N*W_N + Cgdo_N*W_N;
%account for both C_gd and C_gs

C_db_P = Cj0_P*(W_P*L_P)+Cjsw_P*(2*L_P+W_P);
C_sb_P = C_db_P;
C_sg_P_OFF = Cgso_P*W_P;  %Just C_gs, for the 2nd transistor
C_g_P = W_P*L_P*Cox_P + Cgso_P*W_P + Cgdo_P*W_P;

% Take the wire to be on Metal 2; Metal 1 is for VDD
W_wire = 0.28e-6;
L_wire = 5e-6*BitsPerDAC; %assuming 5 microns per SH

C_wire = W_wire*L_wire*19e-18+L_wire*59e-18;
C_others = BitsPerDAC.*(C_dg_N_OFF + C_sg_P_OFF);

%-------------------------------------------------------------------------
%Resistance Calculations
%-------------------------------------------------------------------------

VGS_N = 1.8 - Vin;   %gate at VDD when TG is on
VGS_P = Vin - 0;     %grounded gate when TG is on

Veff_N = (VGS_N-Vt_N);
Veff_P = (VGS_P-Vt_P);

if Veff_N <= 0
   Rds_N = 1e15; %'infinity' - take a subthreshold transistor to be off.
else
   Rds_N = 1./(mu0_N*Cox_N*(W_N/L_N)*Veff_N);
end

if Veff_P <= 0
   Rds_P = 1e15; %'infinity' - take a subthreshold transistor to be off.
else
   Rds_P = 1./(mu0_P*Cox_P*(W_P/L_P)*Veff_P);
end

R_trans = 1./(1./Rds_N + 1./Rds_P);
R_out_DAC = R_trans; %assumption
%-------------------------------------------------------------------------
%Aggregate Calculations
%-------------------------------------------------------------------------

%BUS METHOD
C1_BUS = C_wire + C_others;
```

```
C_trans = C_db_N + C_sb_N + C_g_N + C_db_P + C_sb_P + C_g_P;
C2_BUS = C_cap + C_dg_N_OFF + C_sg_P_OFF;
tau_BUS = R_out_DAC.*C1_BUS+(R_out_DAC+R_trans).*C2_BUS;  %lumped
model - Rabaey pg. 154

%-------------------------------------------------------------------------
%MUX METHOD
C_Node = C_dg_N_OFF + C_sg_P_OFF;
R1 = R_out_DAC + R_trans;
R2 = R_out_DAC + 2*R_trans;
R3 = R_out_DAC + 3*R_trans;
R4 = R_out_DAC + 4*R_trans;
R5 = R_out_DAC + 5*R_trans;
R6 = R_out_DAC + 6*R_trans;
R7 = R_out_DAC + 7*R_trans;
R8 = R_out_DAC + 8*R_trans;

tau_MUX =
R_out_DAC*C_Node+R1*C_Node+R2*C_Node+R3*C_Node+R4*C_Node+R5*C_
Node+R6*C_Node+R7*C_Node+R8*C_cap;

loglog(C_cap,tau_BUS, 'x')
hold on;
loglog(C_cap,tau_MUX, 'o')
loglog(C_cap,tau_BUS, '-')
loglog(C_cap,tau_MUX, '-.')


legend('Bus Method', 'MUX Method')
xlabel('Capacitor size (F)')
ylabel('Distribution Method Time Constant (1/s)')
```

Appendix B

Interface Schematics

Input Interface

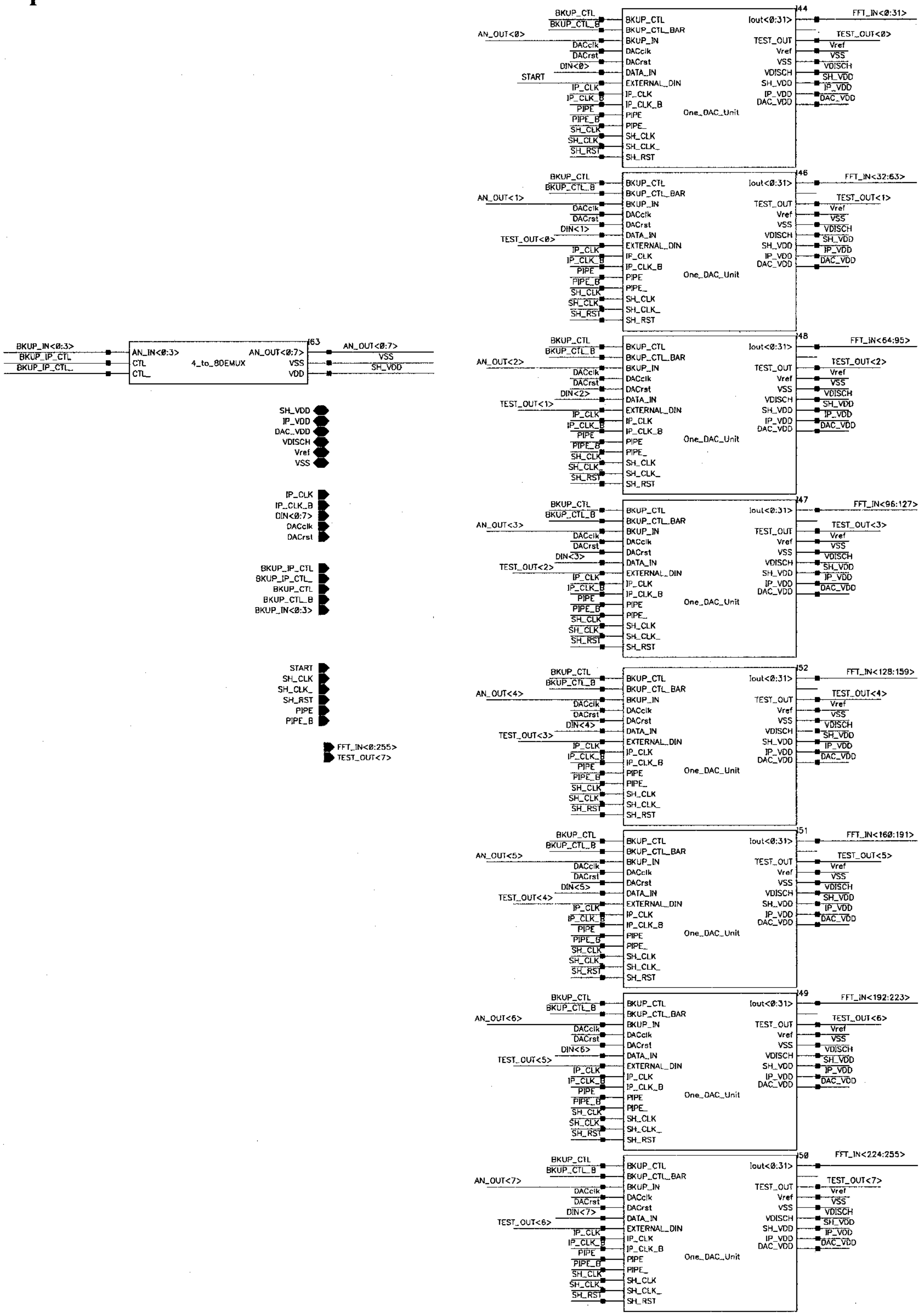


Figure B.1: Complete Input Interface

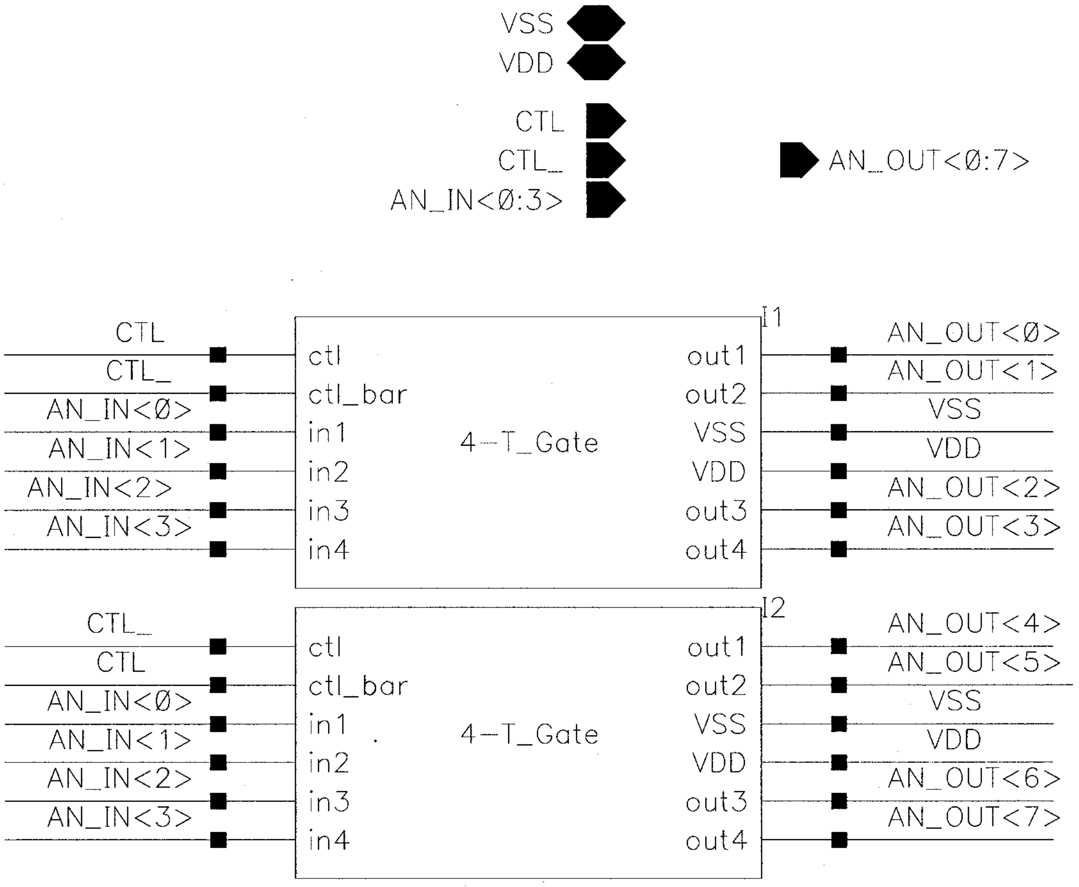


Figure B.2: Backup Demultiplexer

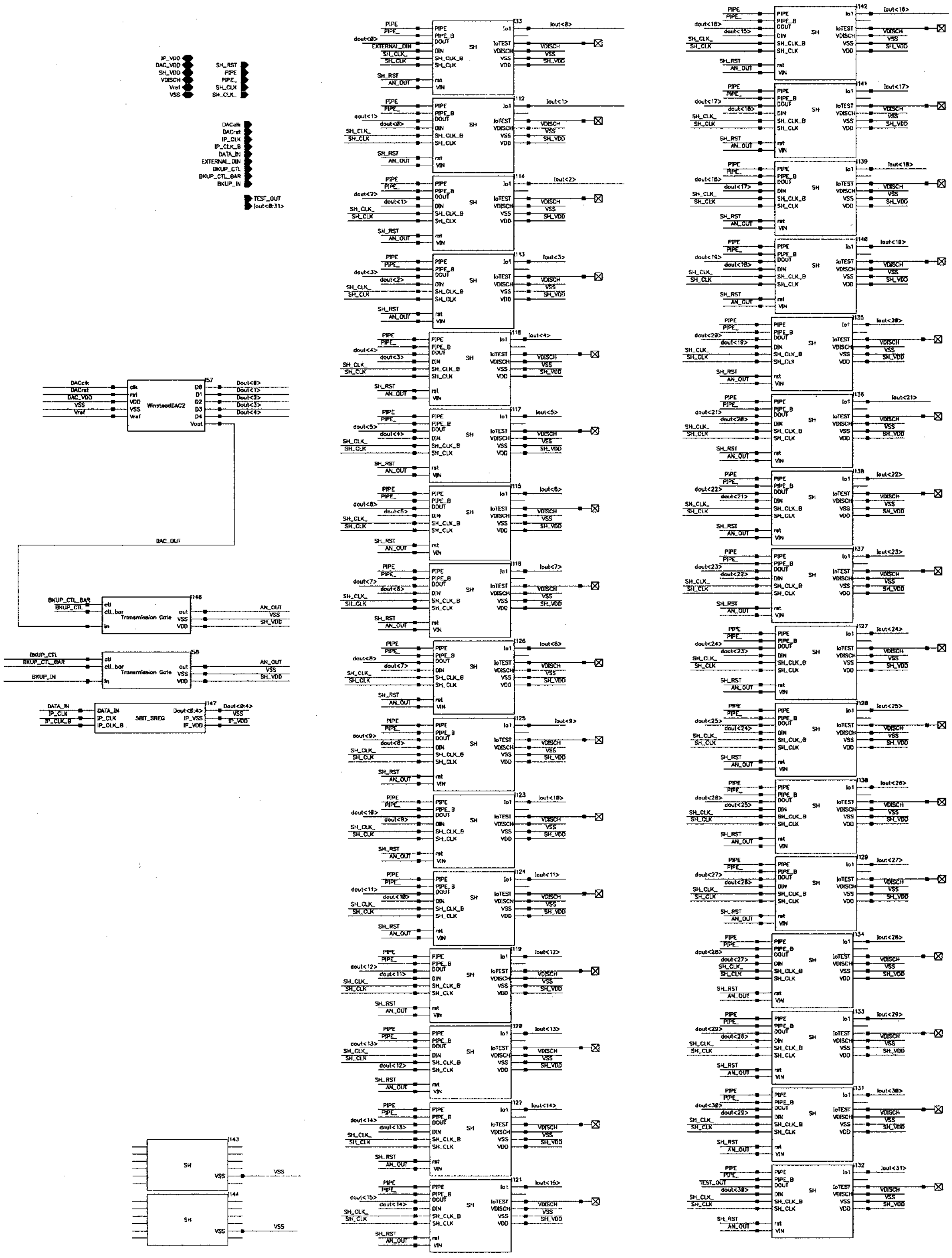


Figure B.3: One DAC Unit

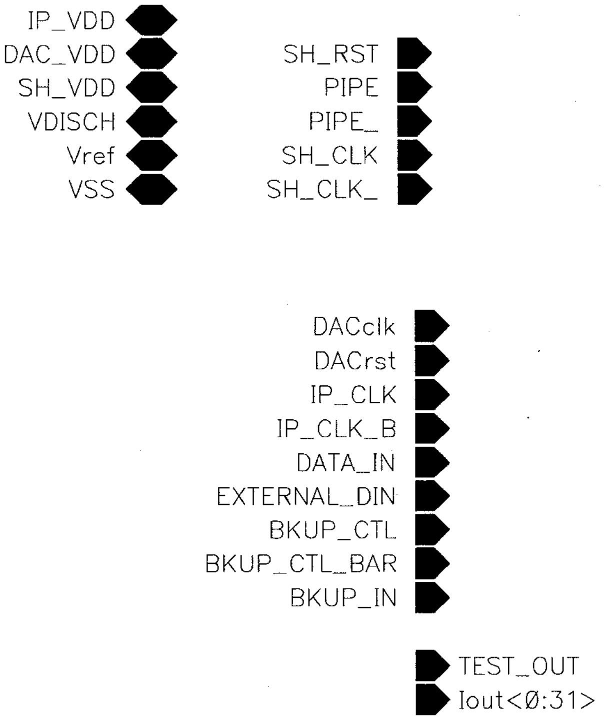


Figure B.4: One DAC Unit (Closeup on Pins)

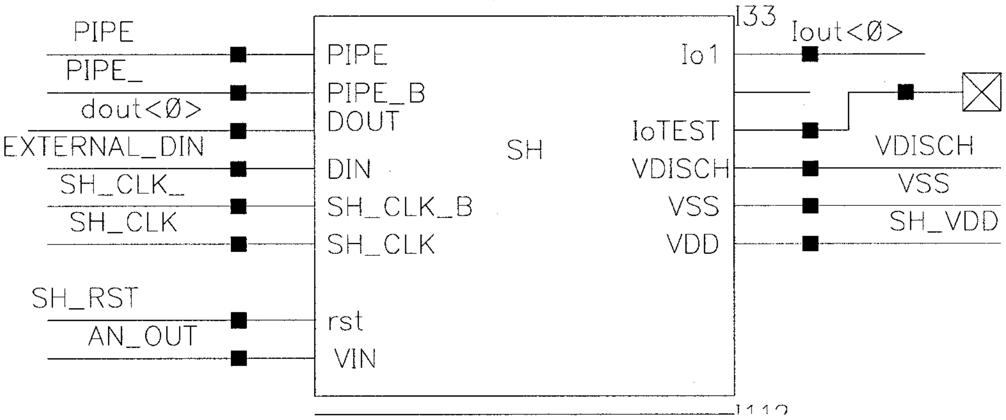


Figure B.5: One DAC Unit (Closeup on One S/H Unit)

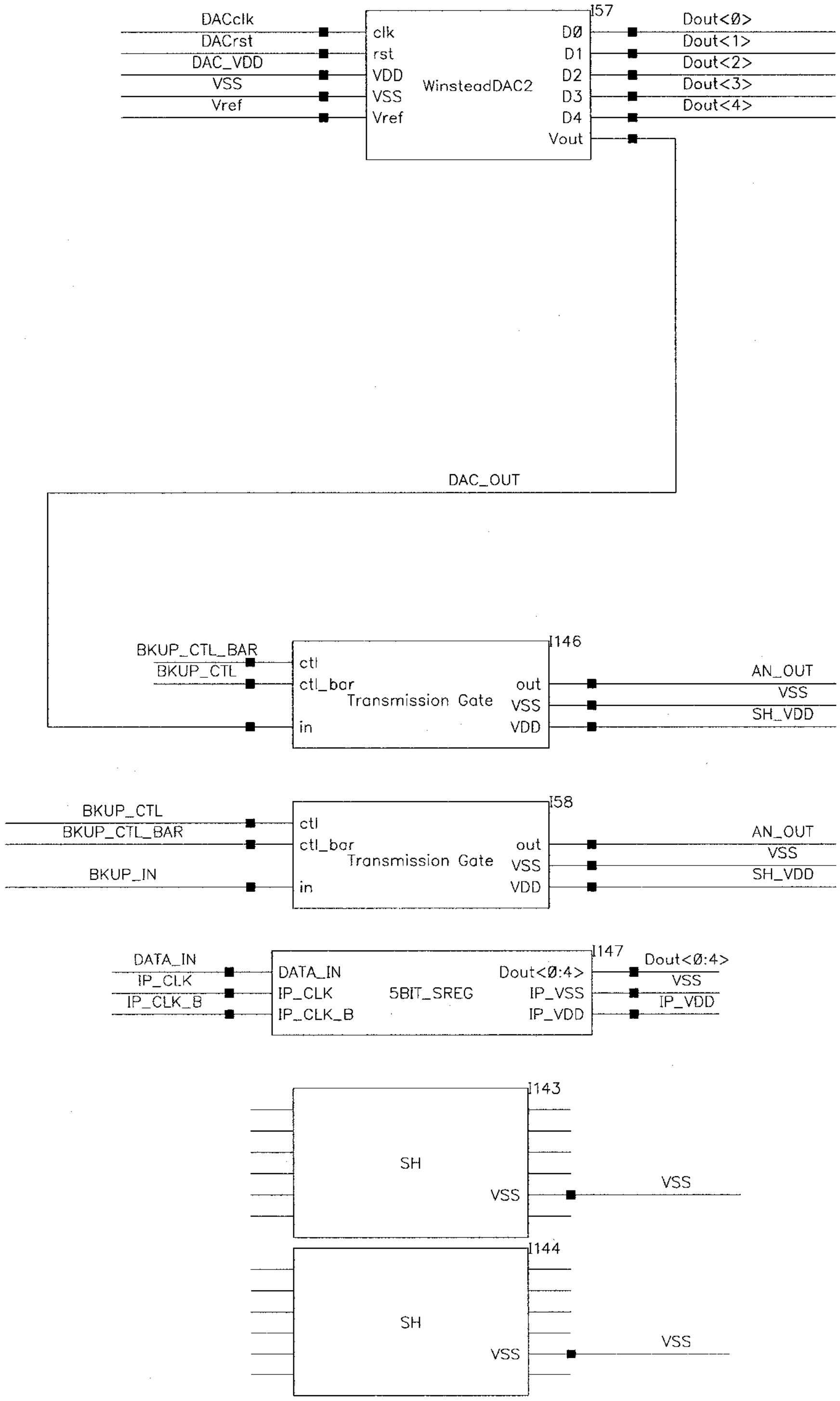


Figure B.6: One DAC Unit (Closeup on DAC, Register, and Dummy SH Units)

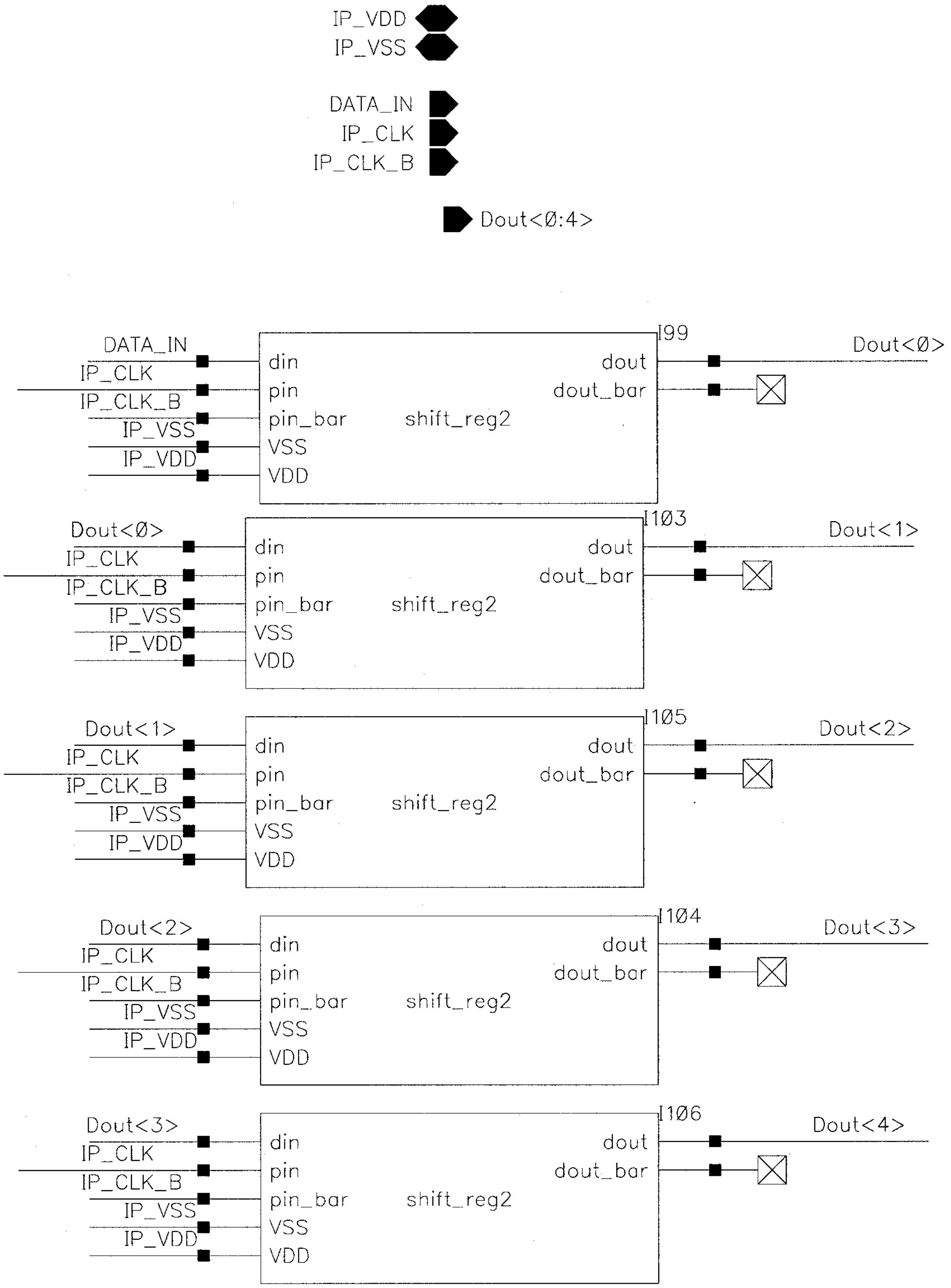


Figure B.7: 5-Bit Shift Register

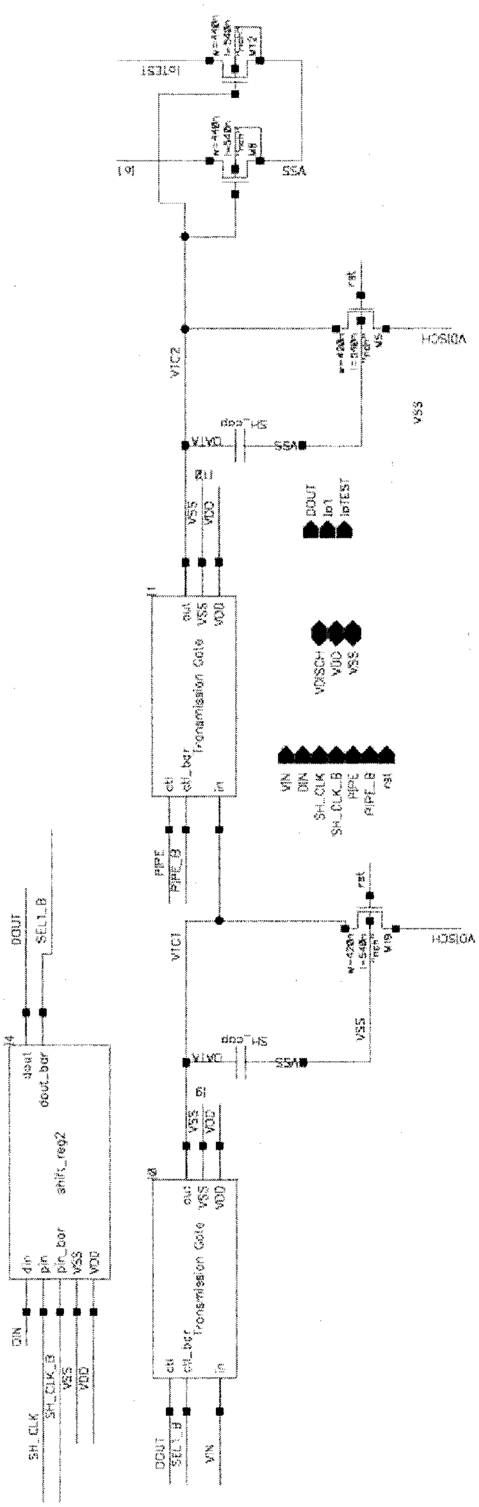


Figure B.8: Sample and Hold Unit

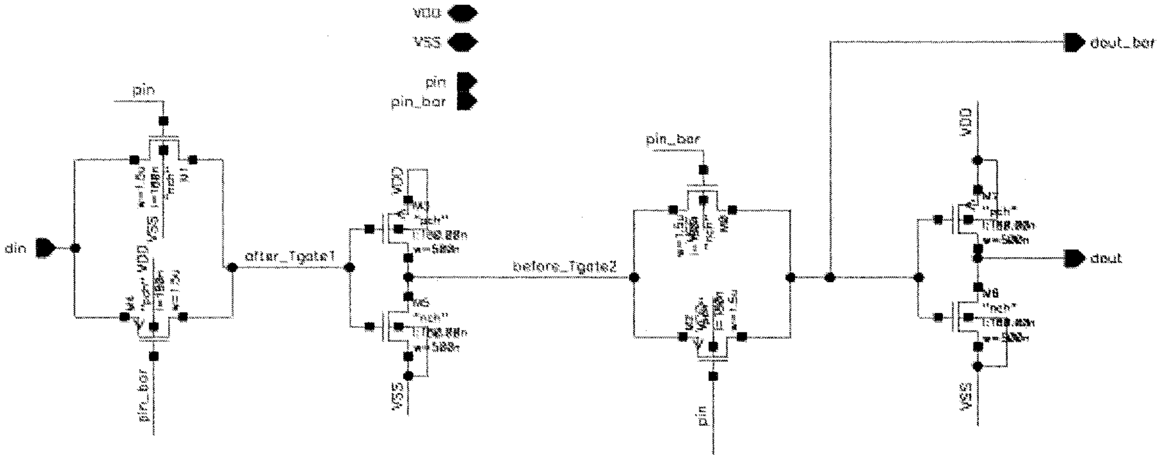


Figure B.9: Shift Register Cell

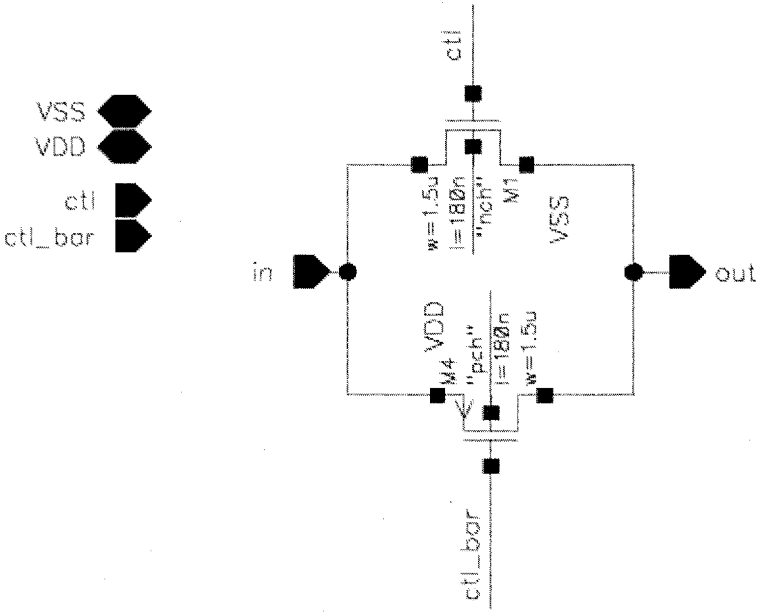


Figure B.10– Basic Transmission Gate

Output Interface

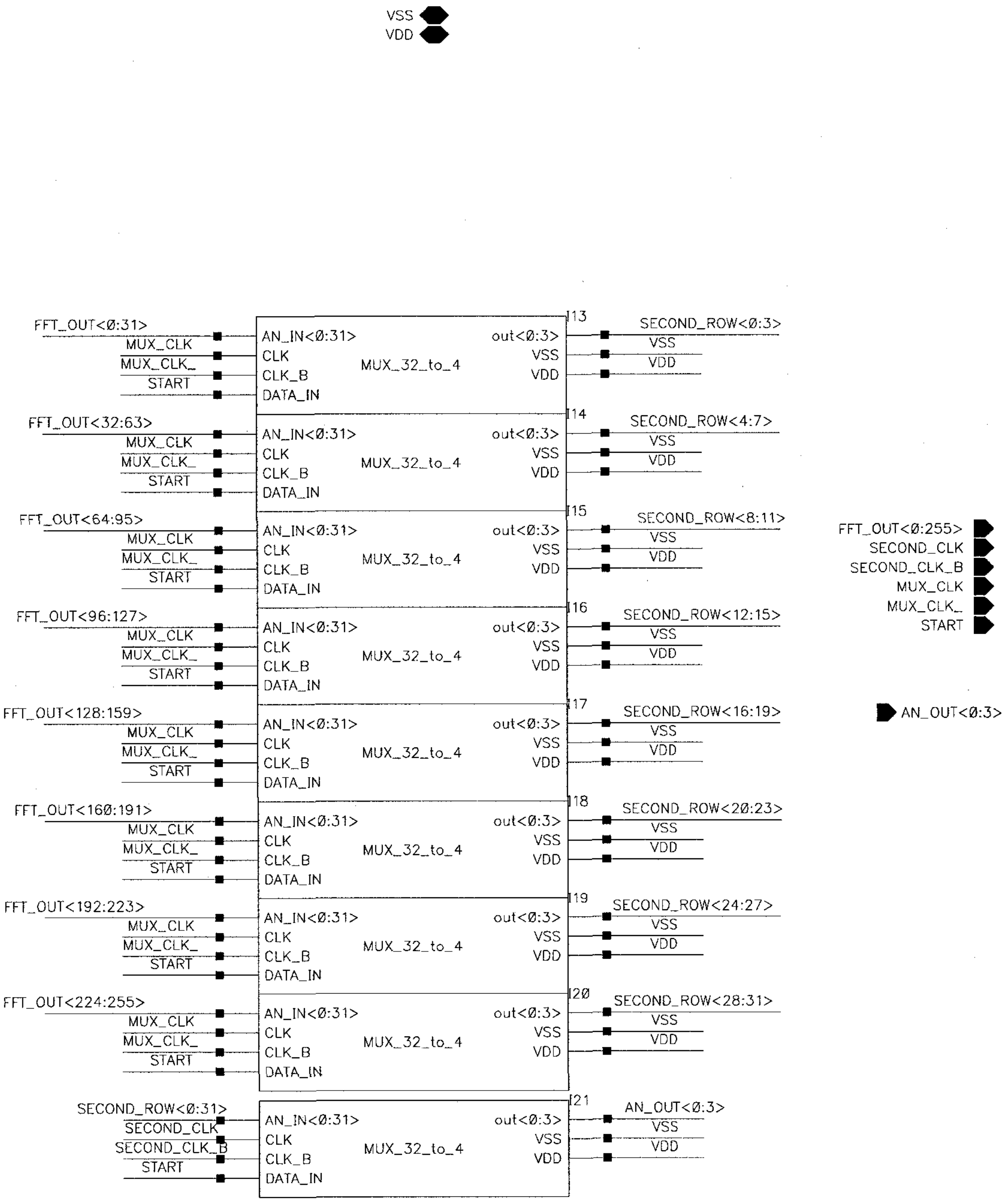


Figure B.11: Output Multiplexer (256 to Four-bit)

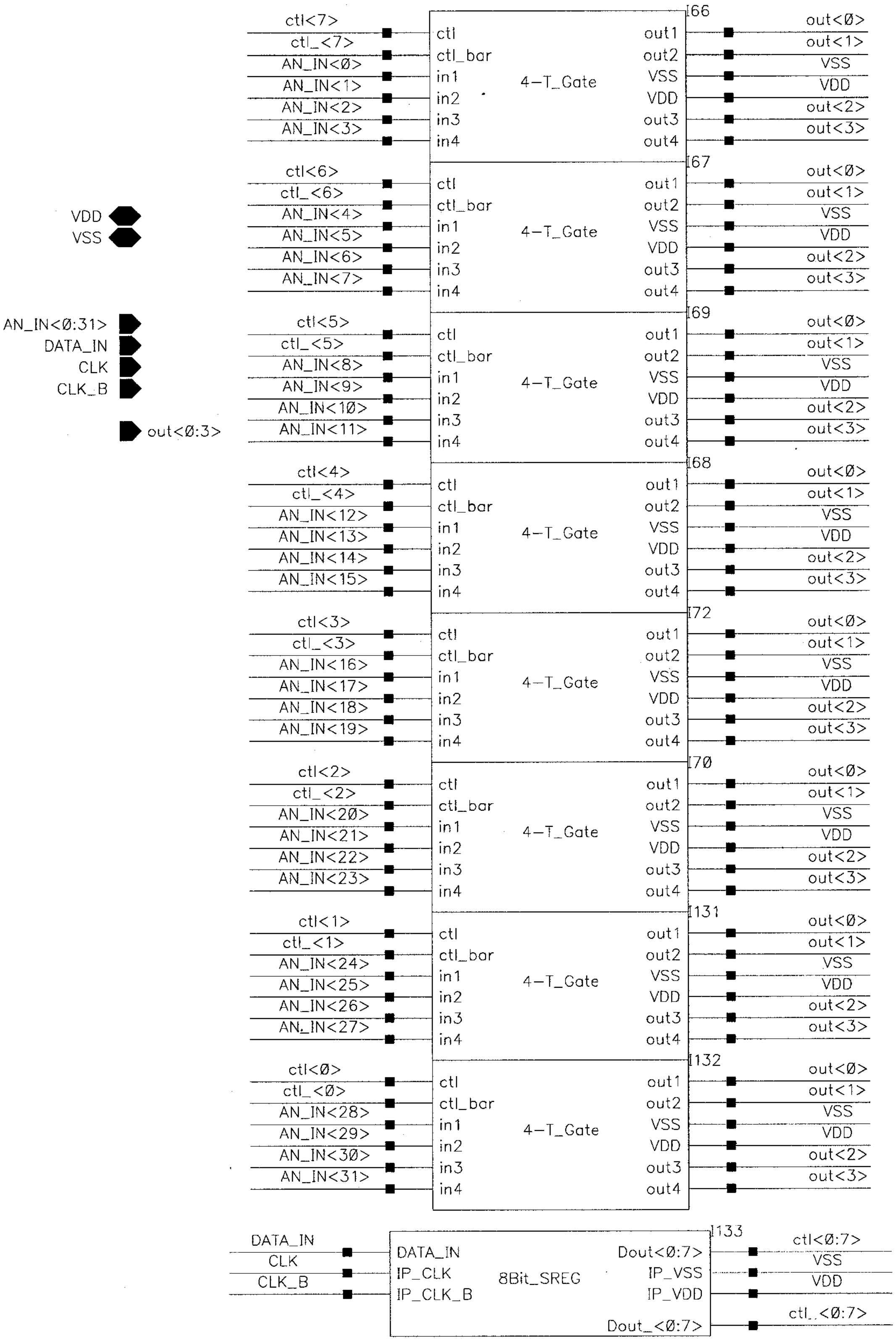


Figure B.12: Multiplexer (32 to Four-bit)

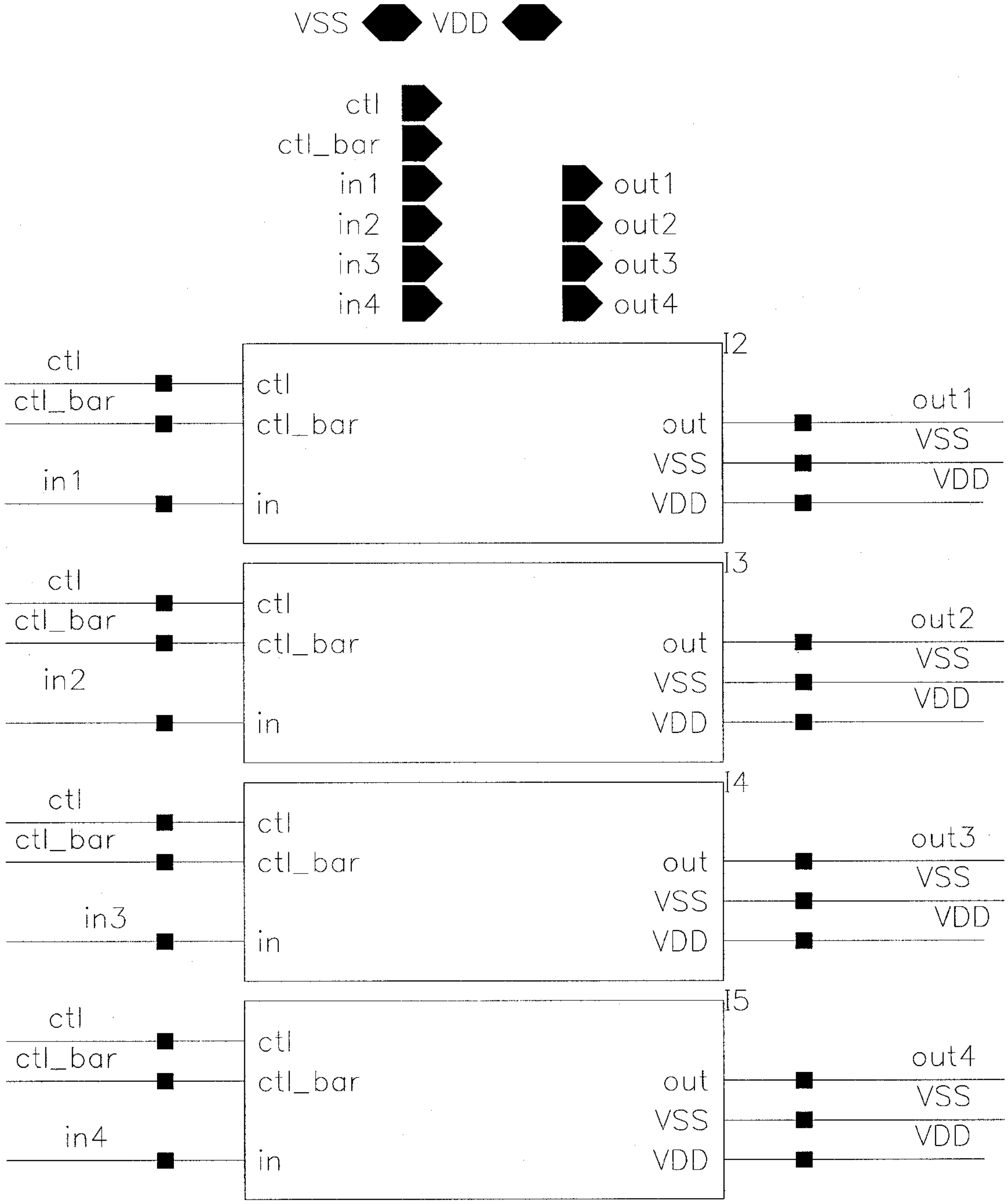


Figure B.13: Four-Bit Transmission Gate

Appendix C

Layout Images

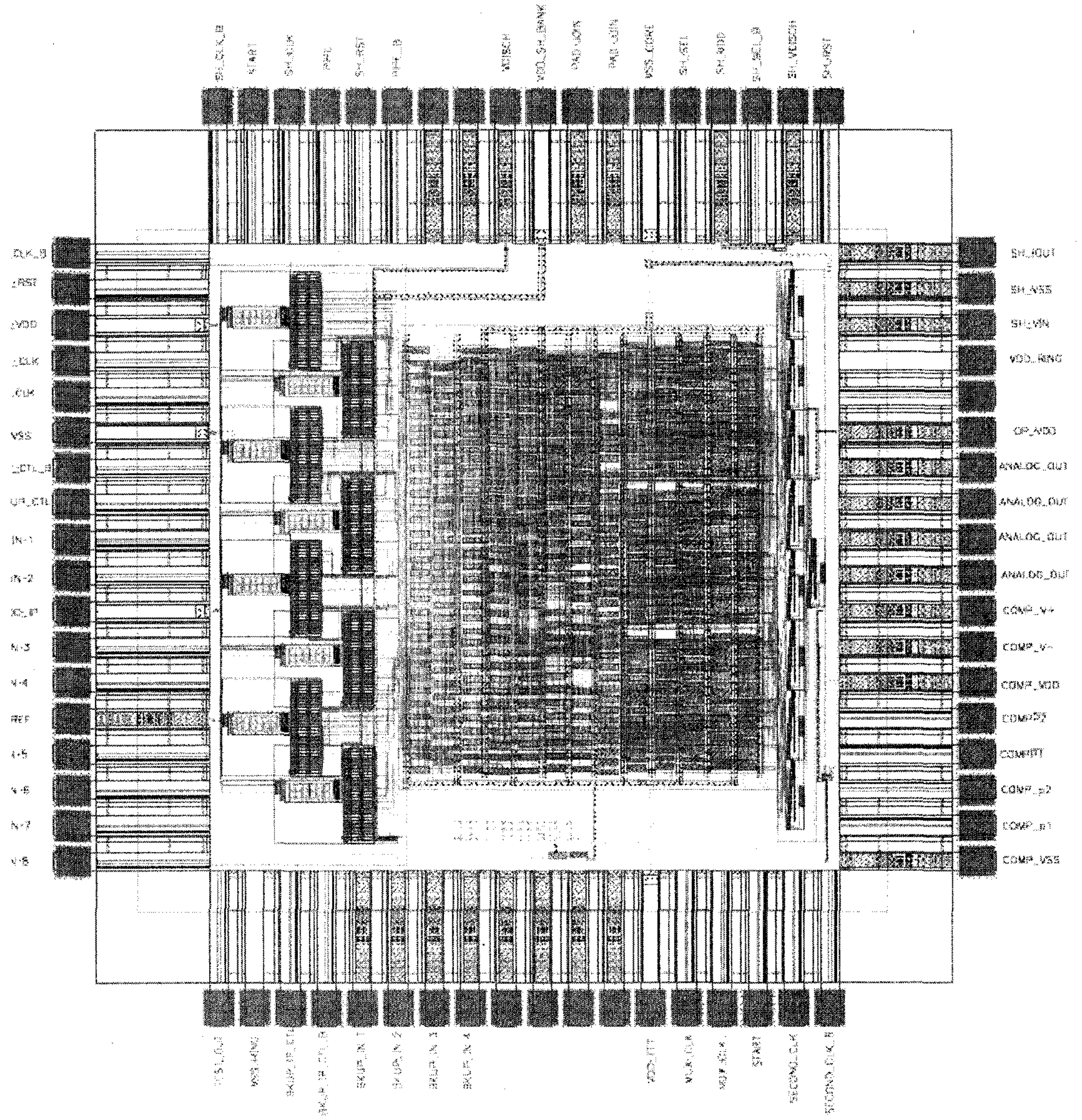


Figure C.1: Full Chip Layout

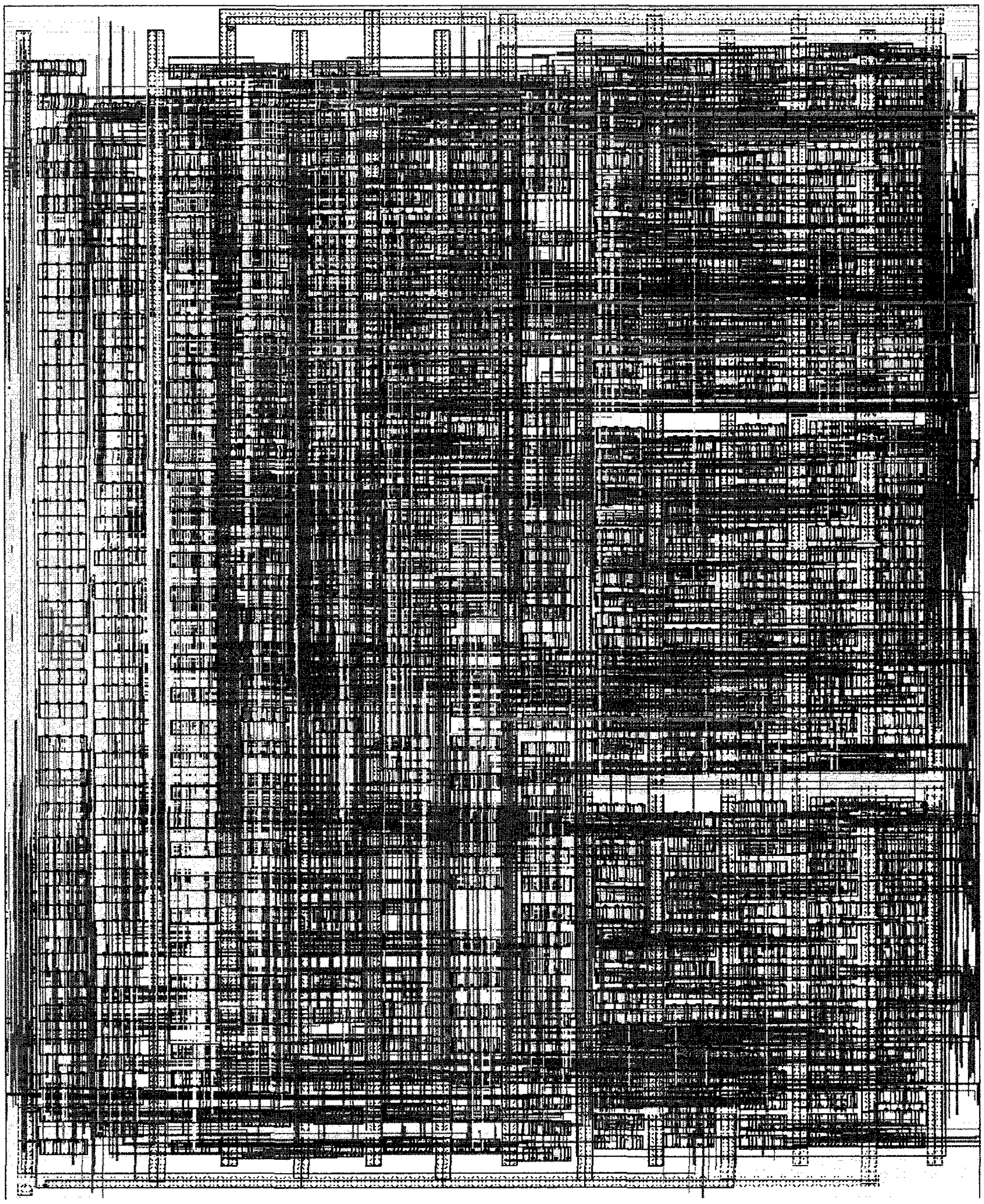


Figure C.2 – Sixty-Four Bit FFT Unit

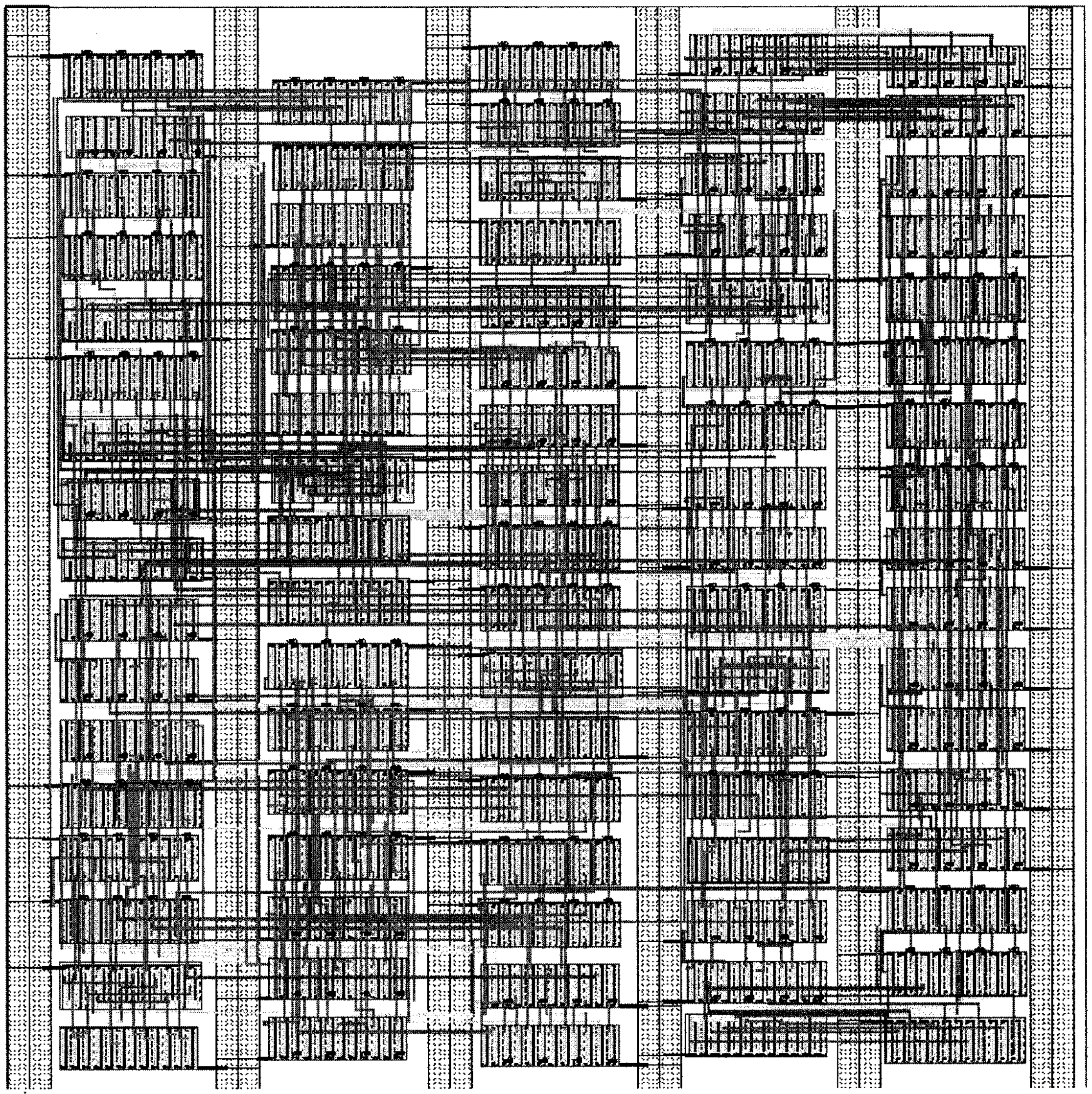


Figure C.3 – Sixteen Bit FFT Unit

Characterization Units

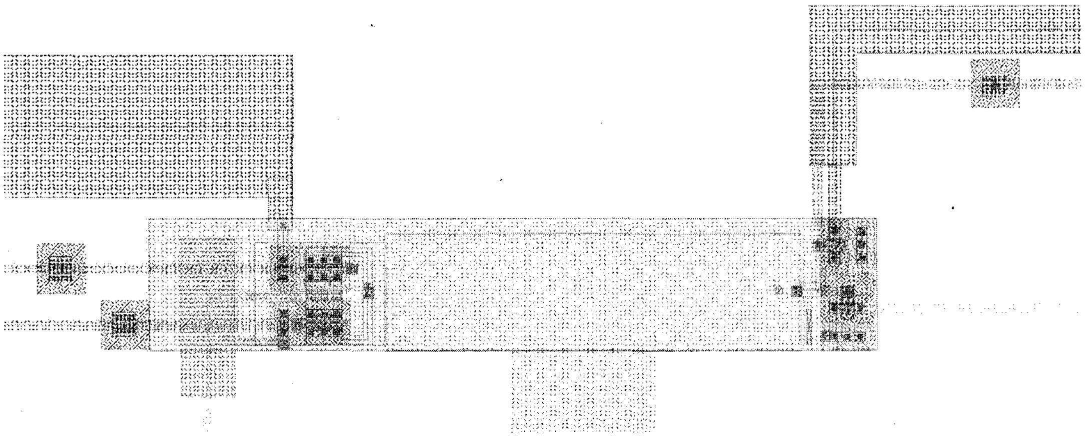


Figure C.4: Sample and Hold Unit

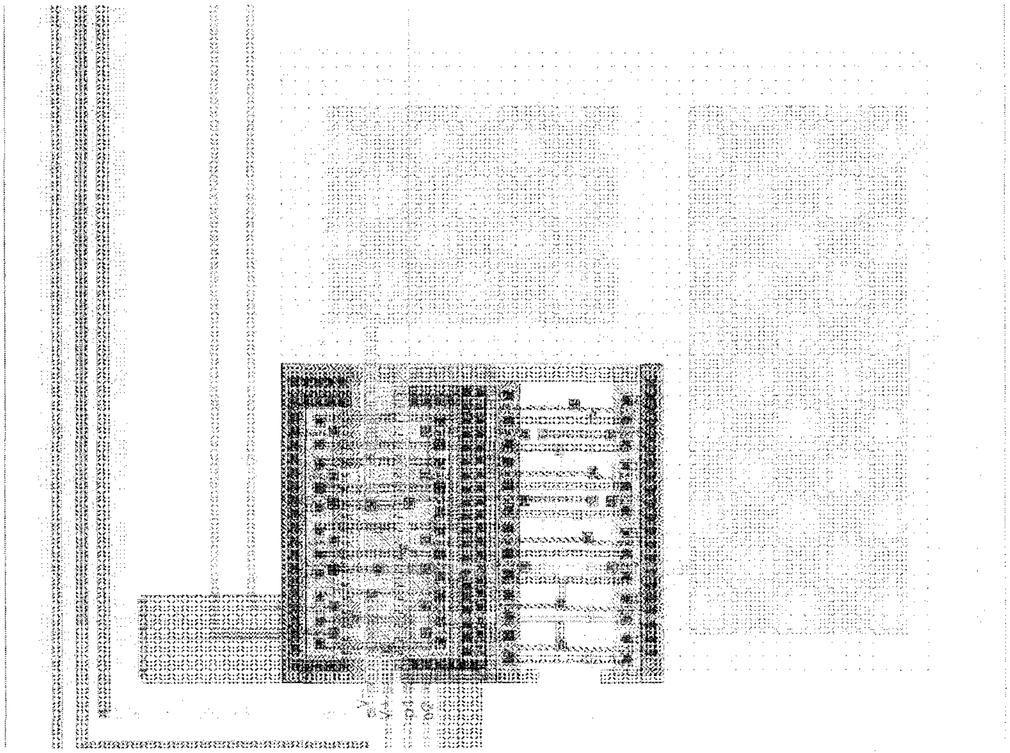


Figure C.5: One Stage of a Dynamic Comparator

Input Interface

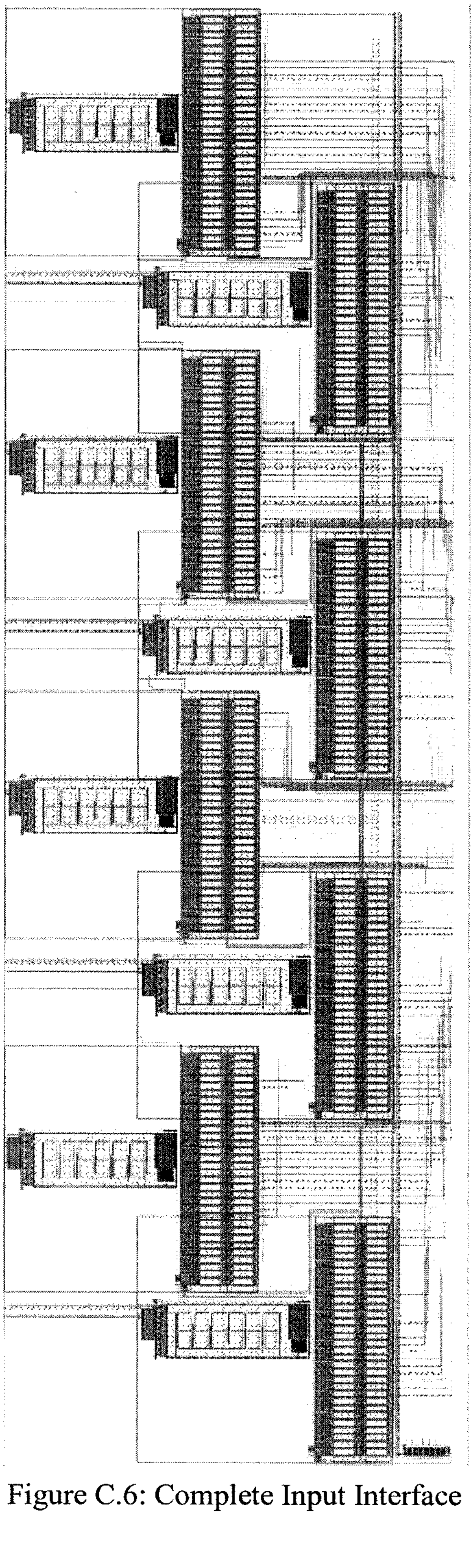


Figure C.6: Complete Input Interface

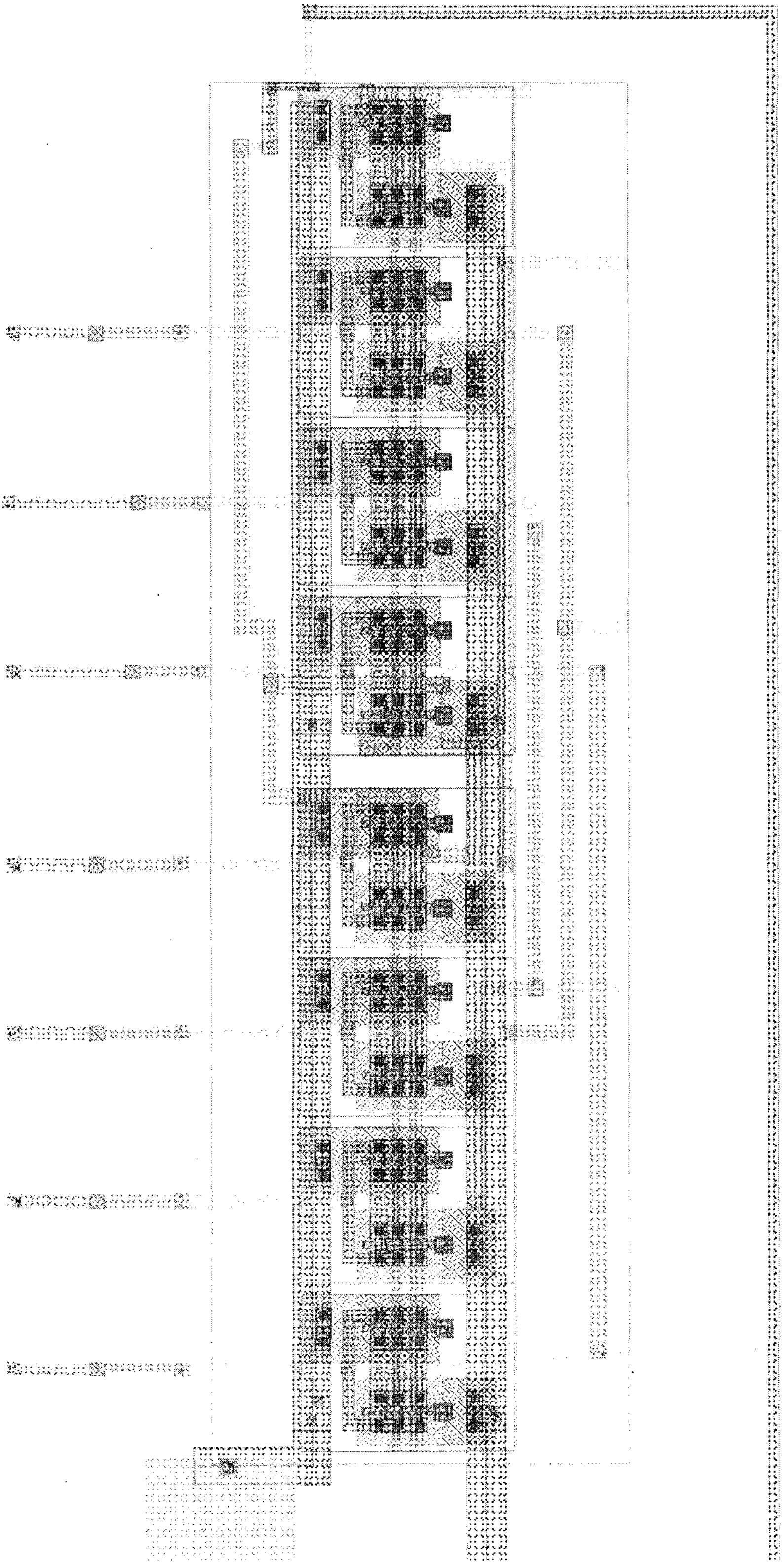


Figure C.7: Backup Demultplexer

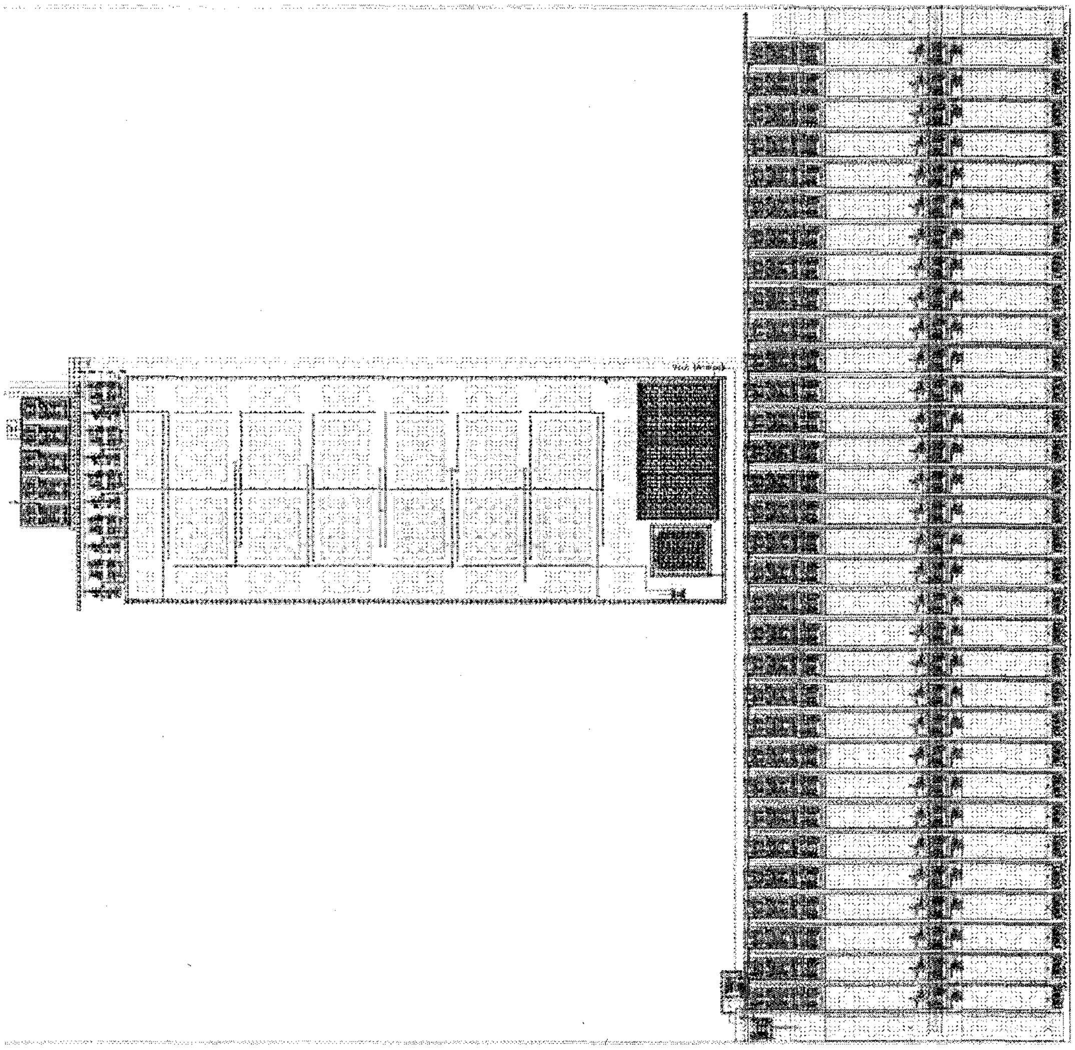


Figure C.8: One DAC Unit

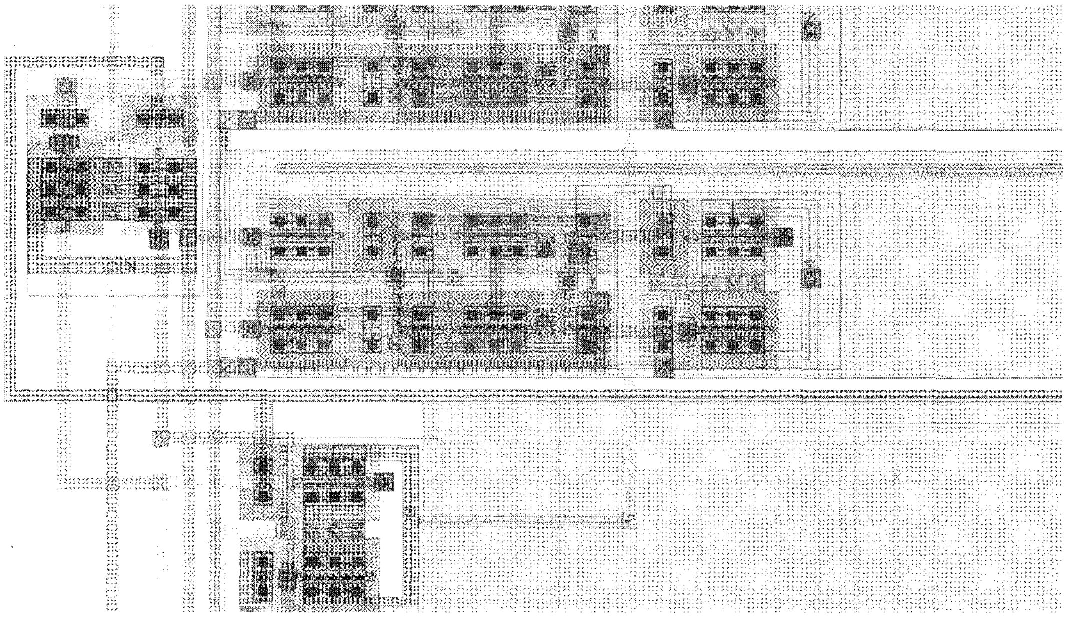


Figure C.9: One DAC Unit (Closeup on Signal Control Demultiplexer)

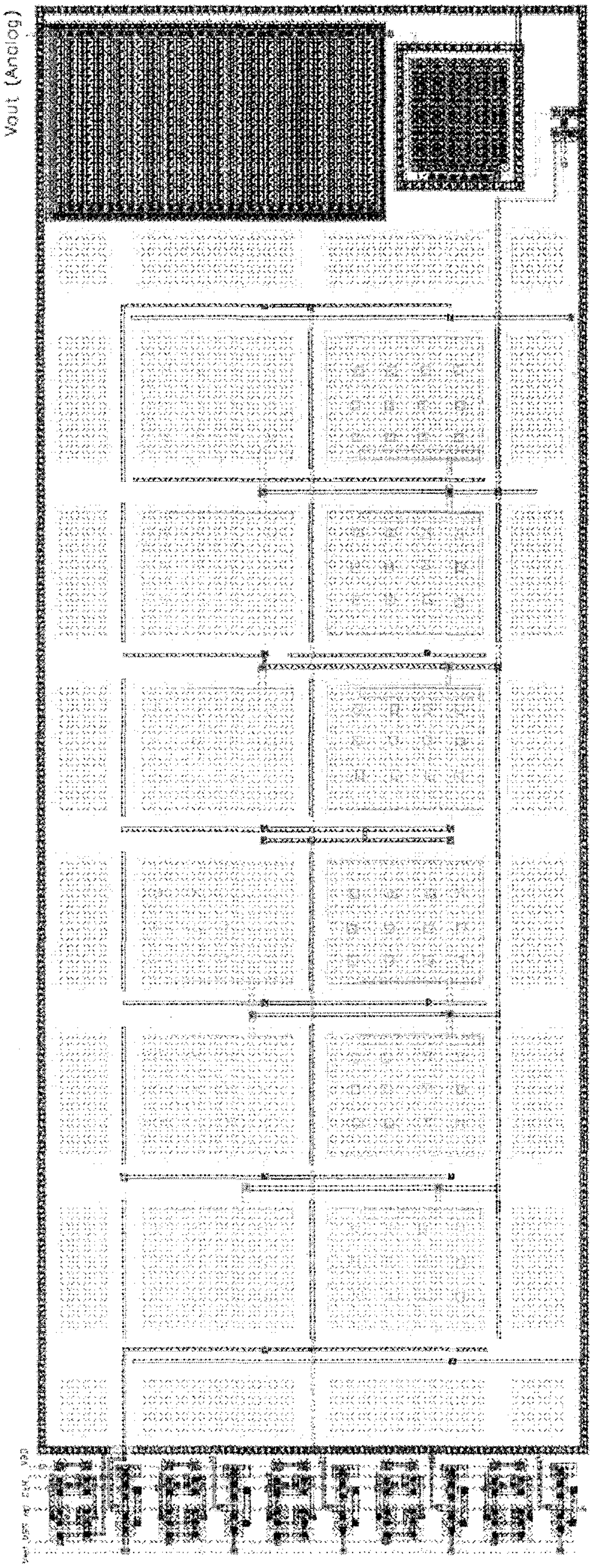


Figure C.10: DAC

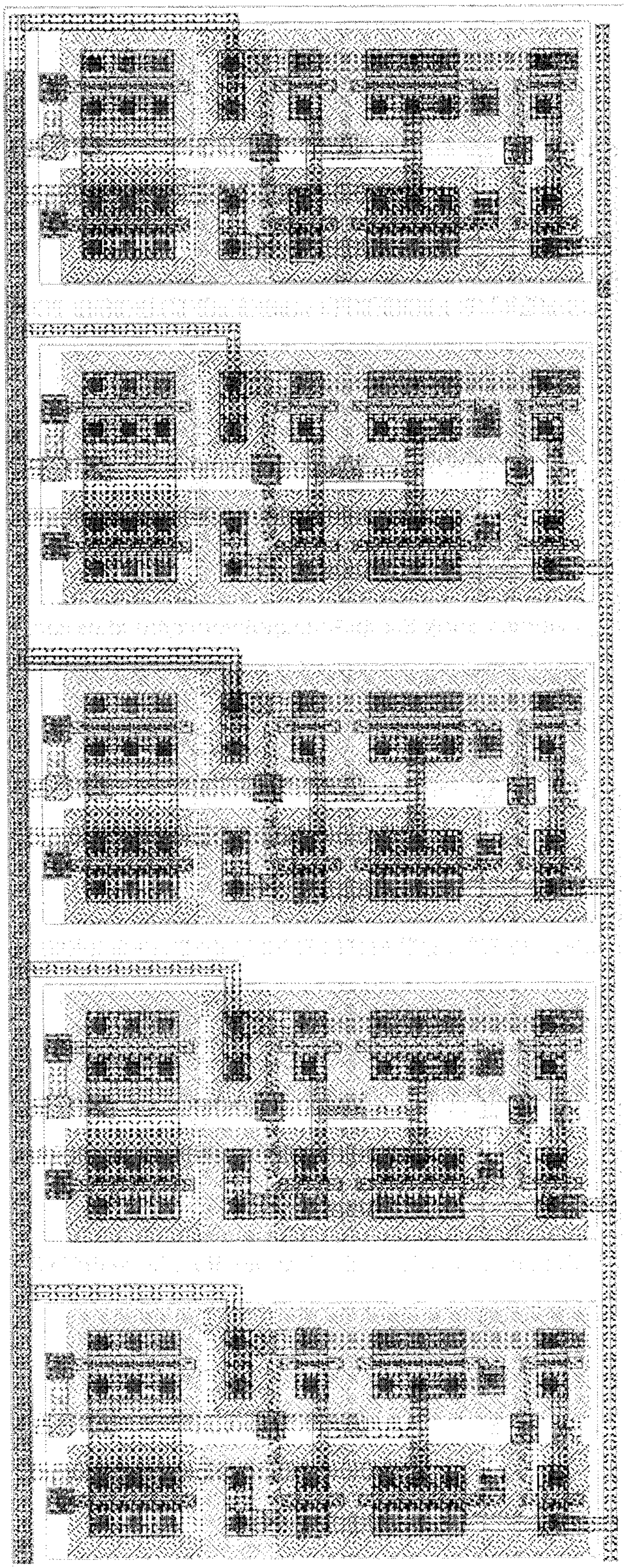


Figure C.11: 5-Bit Shift Register

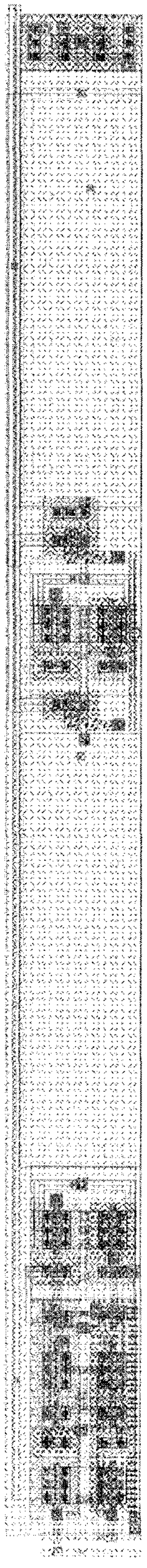


Figure C.12: Sample and Hold Unit

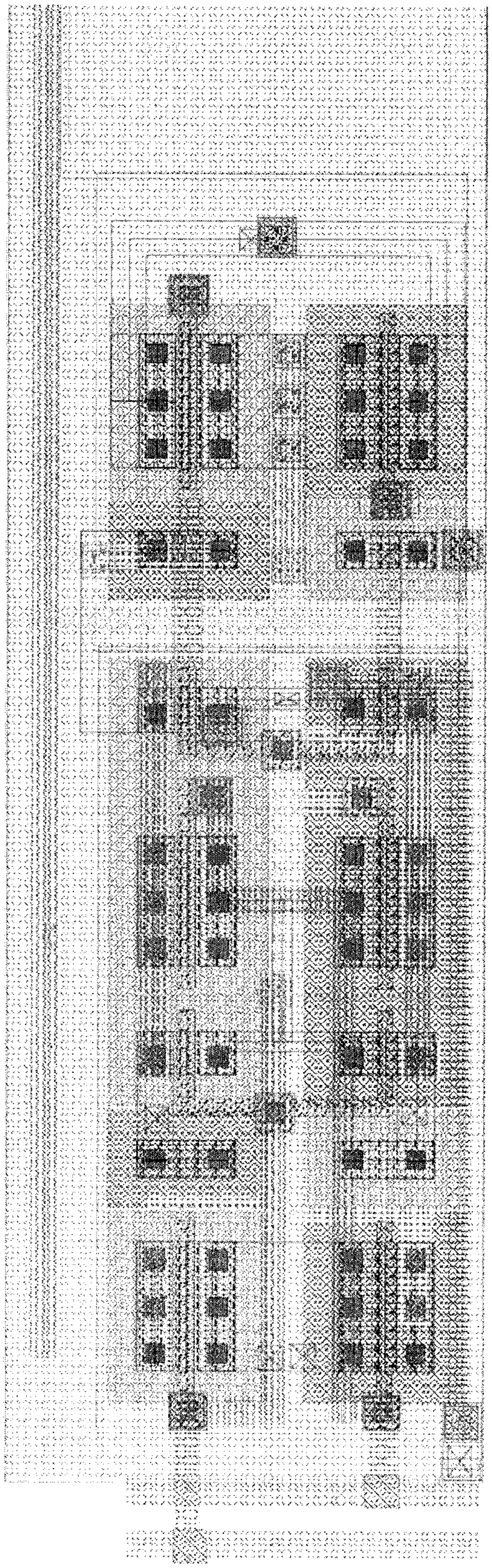


Figure C.13: Sample and Hold Unit (Closeup on Input Control)

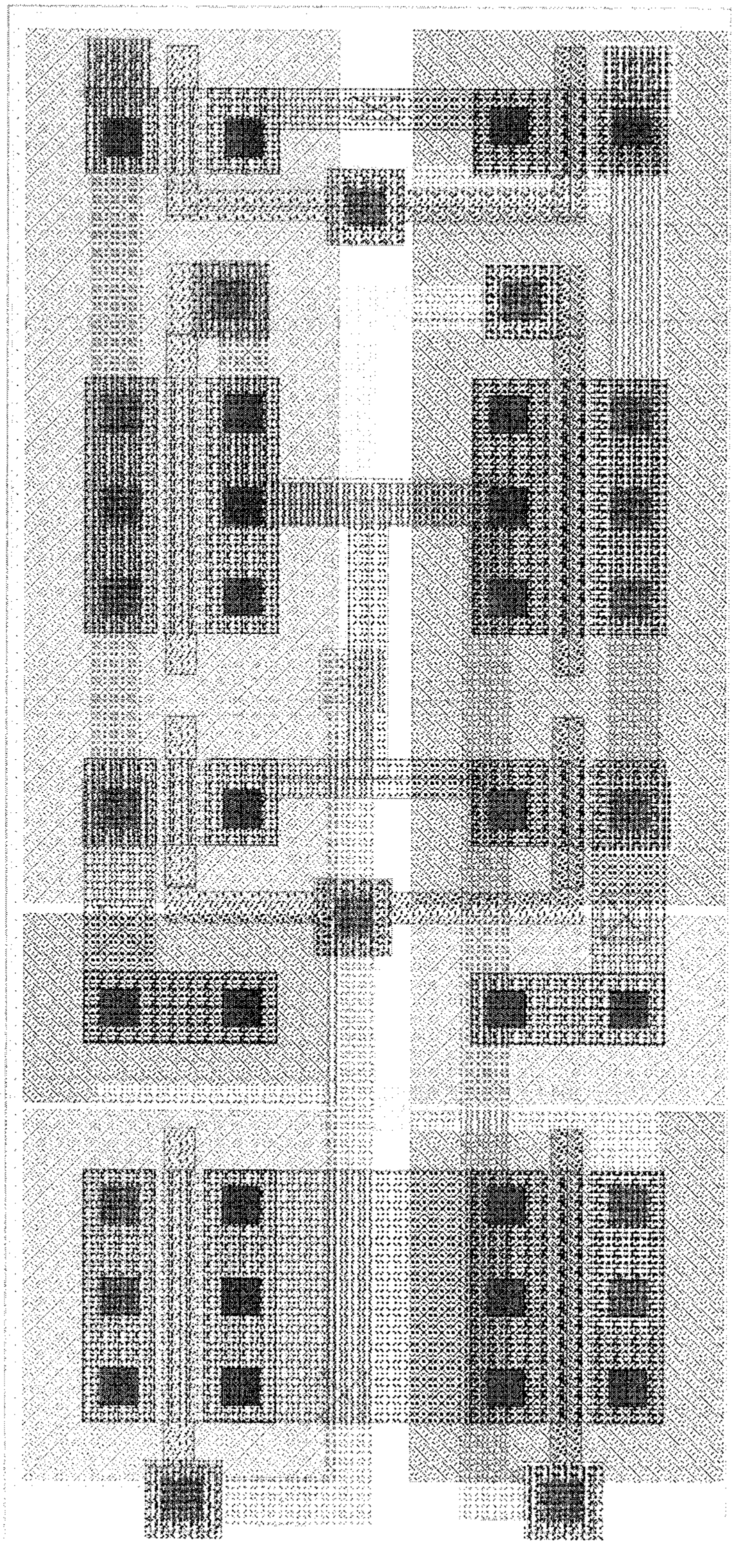


Figure C.14: Shift Register Cell

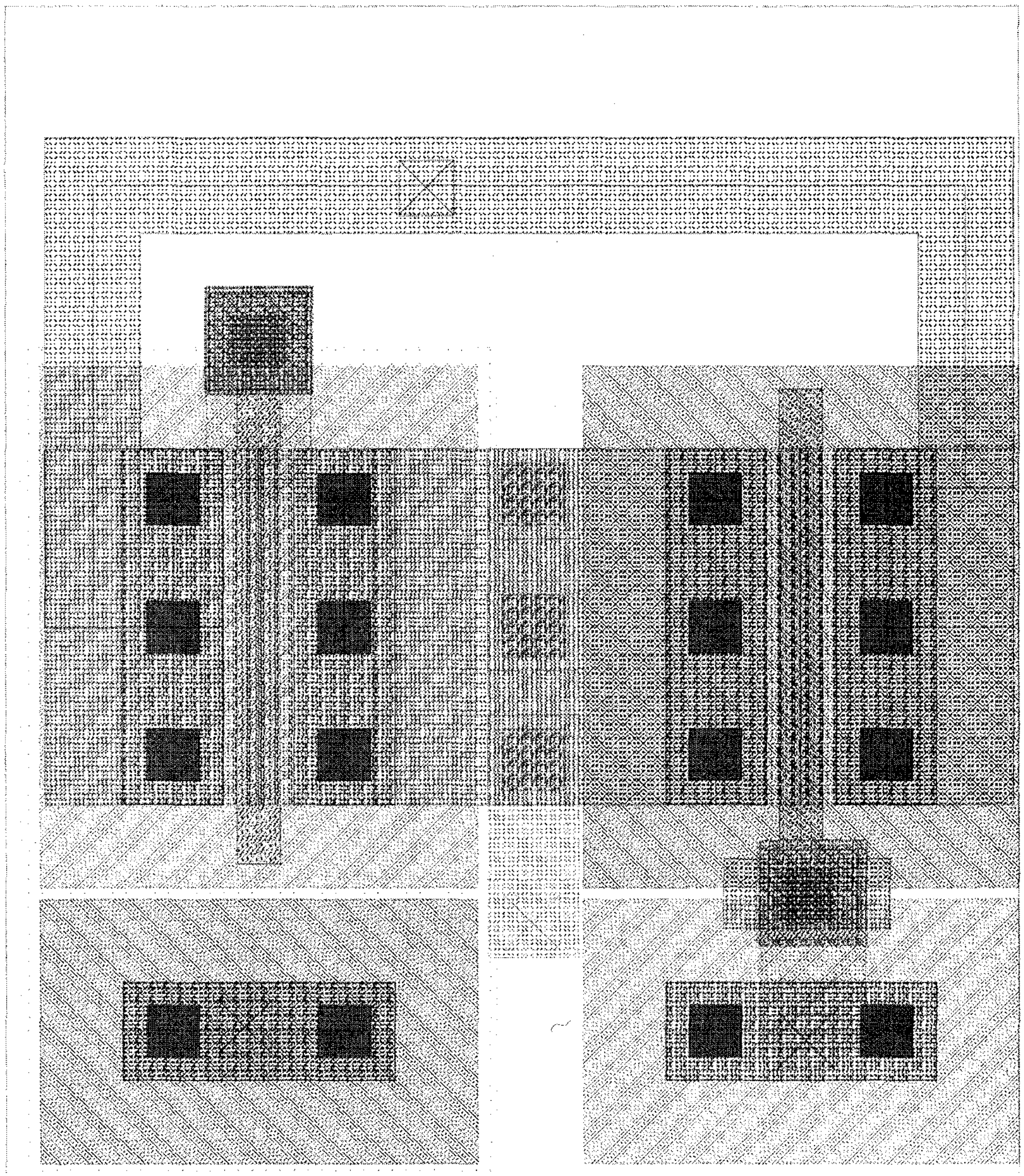


Figure C.15– Basic Transmission Gate

Output Interface

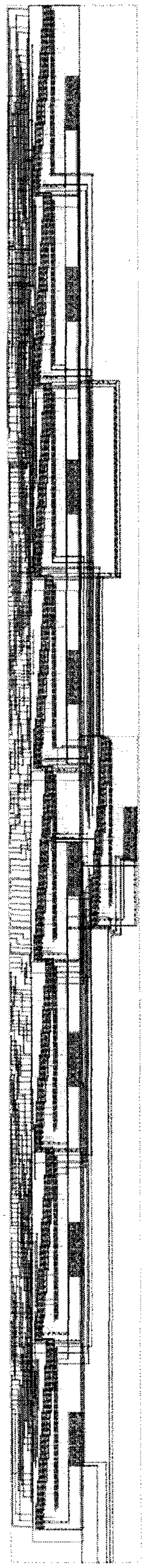


Figure C.16: Output Multiplexer (256 to 4)

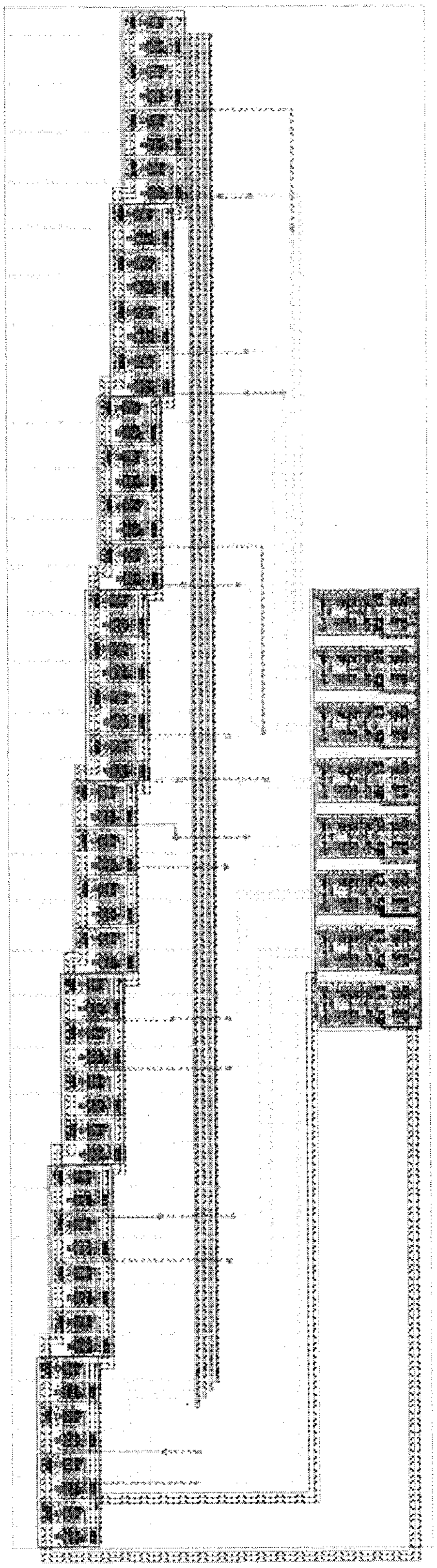


Figure C.17: 32 to 4 Multiplexer

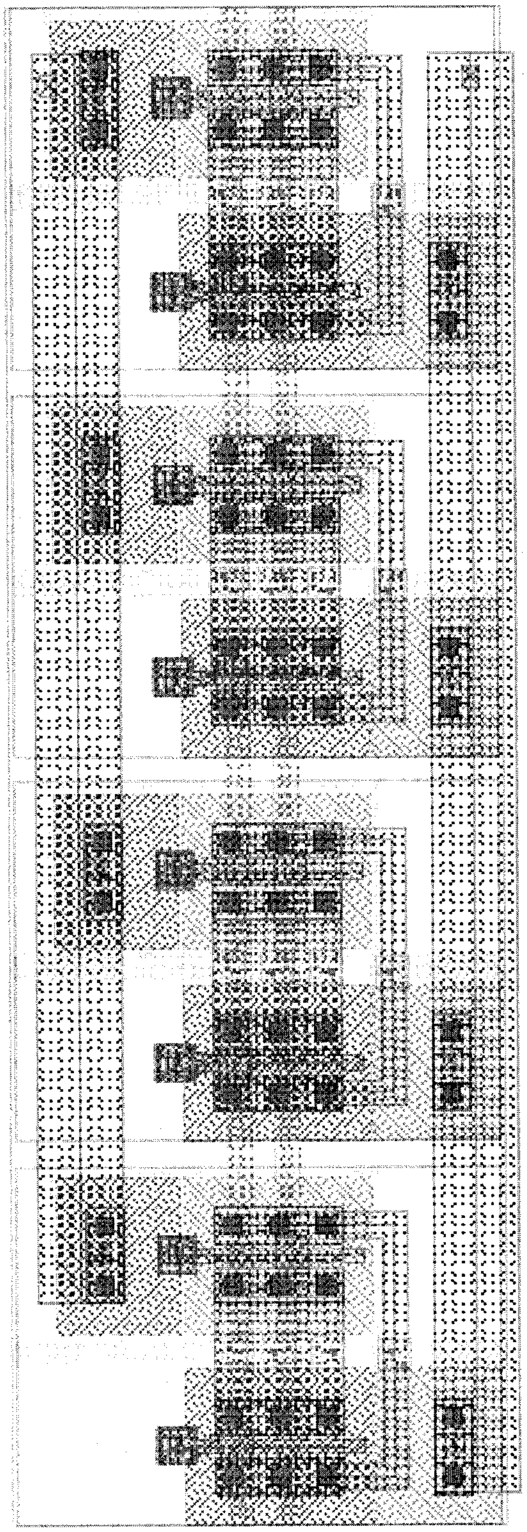


Figure C.18: Four-Bit Transmission Gate