

Copyright © 1999, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**A LOW POWER, 8-bit, 200 MHz  
DIGITAL-TO-ANALOG CONVERTER**

by

Sam Blackman

Memorandum No. UCB/ERL M99/61

7 December 1999

**A LOW POWER, 8-bit, 200 MHz  
DIGITAL-TO-ANALOG CONVERTER**

by

Sam Blackman

Memorandum No. UCB/ERL M99/61

7 December 1999

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

# **A Low Power, 8-bit, 200 MHz Digital-to-Analog Converter**

by

Sam Blackman

Professor Robert Brodersen

December 3, 1999

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering

University of California, Berkeley

94720

## Abstract

The goal of this research was to design a low power 8-bit, 200 MHz digital-to-analog converter suitable for use in an integrated basestation transmitter – and eventually in a transceiver. There were many challenges throughout the design process, including determining the matching requirements of the devices, investigating what percentage of segmentation should be used, and examining the signal-to-noise ratio vs. power tradeoff.

A filter and mixer that directly follow the digital-to-analog converter were fully designed. A prototype of the DAC, filter, and mixer that represented the RF front-end of a basestation transmitter on the same silicon was built in a .25  $\mu\text{m}$  process. To test the chip, a 4-layer PCB board was also designed.

## Acknowledgment

Graduate school here at Berkeley has been a fantastic experience. I have never experienced an environment that has so much intensity – not just academically but socially as well. The university provides the opportunity for a balanced life, a benefit that cannot be underestimated. I'd like to thank a few people who have enriched my years here.

Professor Bob Brodersen has been a terrific advisor and mentor. If I was the orphan type without real parents, he is the father figure I would choose – after Barry Bonds of the San Francisco Giants. Bob manages to combine a driving work ethic in the group with a great sense of humor; he has made the Berkeley Wireless Research Center an absolutely ideal place for brilliant research to occur. I cannot thank him enough.

My peers in the RF group are incredible. I would especially like to single out Johan Vanderhaegen and David Sobel, my cubicle mates. I have never met anyone with as wide a breadth of knowledge as Johan; they must put something special in that Belgian Coke. Johan patiently helped me whenever I struggled with a concept – which was often. Dave helped me out a lot, too, though not quite as patiently. Sayf Alalusi, Brian Limketkai, Chinh Doan, and Dennis Yee are analog artists, Cadence-style. And Andy Klein was full of lucrative financial advice.

Most importantly, my family and friends have been tremendous source of support. As cheesy as it sounds, they keep you going when times are rough. I would not have survived without them – literally, in the case of my parents. Thank you all for everything!

## Table of Contents

	<u>Page</u>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1. Motivation	1
1.2. Thesis Organization	3
<b>Chapter 2. Digital-to-Analog Converter Architectures</b>	<b>5</b>
2.1. Overview	5
2.2. Resistor-Ladder (Voltage Divider) Architecture	5
2.3. Charge Division Architecture	7
2.4. Current Division Architecture	10
2.5. Current-steering Architectures	12
2.6. Segmented Architectures	15
<b>Chapter 3. DAC Circuit Design Techniques</b>	<b>21</b>
3.1. Overview	21
3.2. Analog Aspects	23
3.2.1. Power Consumption	24
3.2.2. Matching Requirements	25
3.2.3. Signal-to-Noise Ratio	27
3.2.4. Speed Requirements	29
3.2.5. Bias Circuitry	30
3.3. Digital Aspects	32
3.3.1. 4-to-1 Multiplexor	32
3.3.2. Binary-to-Thermometer Decoding Logic	33

3.3.3. Current Source Decoder	34
3.4. Layout Issues	35
3.5. Technology Scaling	37
<b>Chapter 4. Experimental Prototype and Testing Setup</b>	<b>40</b>
4.1. Prototype	40
4.2. Test Board Schematic Design	41
4.2.1. Chip-on-board Technology	41
4.2.2. 3.3 volt to 2.5 volt Voltage Translation	42
4.2.3. Ground and Power Planes	42
4.2.4. Reference Voltage and Current Creation	42
4.3. Test Board Layout	43
4.3.1. Before Layout	43
4.3.2. Beginning Layout	44
4.3.3. Placing Components	44
4.3.4. Routing Power and Ground	45
4.3.5. Routing Signal Lines	47
4.3.6. Last Steps	47
<b>Chapter 5. Conclusions</b>	<b>48</b>
5.1. Conclusion	48
5.2. Performance	49
5.3. Scaling into the Future	50
5.4. Summary	51
<b>Appendix A. Basics of Transmitter System Design</b>	<b>52</b>
<b>Appendix B. Binary-to-Thermometer Decoding</b>	<b>56</b>



<b>Appendix C. Schematics and Layouts</b>	<b>57</b>
C.1. Digital Components	57
C.2. Analog Components	58
C.2.1. Current Source Cell	58
C.2.2. Cascode Biasing Circuit	59
C.2.3. Complete DAC Layout	62
<b>References</b>	<b>64</b>

# Chapter 1

## Introduction

---

### 1.1 Motivation

The use of wireless communication devices has proliferated in the past decade. Cellular phones have become a ubiquitous aspect of daily life in America, and even more so in countries with relatively expensive copper-wire based phone systems such as Finland and Sweden (of course, having dominant companies nearby such as Nokia and Ericsson has helped!). Despite their popularity, however, these devices have limited bandwidth and for the most part are limited to voice and text transmissions. A decade ago, those were the only major communication applications and thus these low bandwidth devices were perfectly sufficient. With the explosion of the Internet, however, users are expecting – no, *demanding* – much higher bandwidth wireless links [Figure 1.1]. Images, sound, and video are only a few examples of the applications that can utilize this increased capacity. Eventually, high-speed



**Figure 1.1.** Wireless devices, from low-bandwidth to high-bandwidth: The Palm VII, Nokia 7110 cell phone, National Semiconductor WebPAD, artist's rendering of an InfoPad prototype.

wireless links will enable information appliances of all varieties, with devices like the UC Berkeley InfoPad and the 3Com Palm VII being prime examples of the possibilities.

However, the high-bandwidth aspect is only one part of the envisioned communication system. Wireless devices tend to be mobile, and thus the low power aspect of the design is critical. A fast link does not do one much good if it can only be used for an hour before needing to be wired to the wall for a recharge! Thus it is critical for any transceiver IC designed for wireless communications to be extremely low power. The third aspect, as any engineer knows, is low cost. Currently, manufacturing costs of communication chipsets are relatively high due to the number of discrete components needed. Although not much of a factor for the digital-to-analog converter block, in the mixer, LNA, and filtering blocks replacing off-chip components can drop the manufacturing price by orders of magnitude. Thus integration of the digital-to-analog converter with the rest of the transceiver components is of utmost importance.

The digital-to-analog converter was designed with all these issues in mind. It was

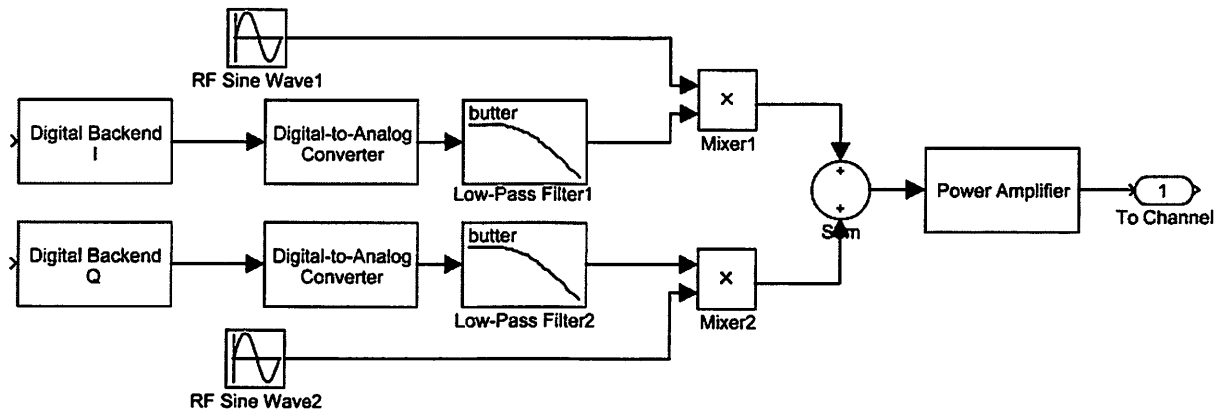


Figure 1.2. Simulink block diagram of a transmitter analog front end.

then integrated into a basestation transmitter for the single-chip radio. A typical transmitter block diagram is shown in Figure 1.2.<sup>1</sup> The digital backend circuitry generates two digital signals, the in-phase and the quadrature bitstreams. These digital bitstreams are converted to an analog signal in the digital-to-analog converter. Next, the two analog signals are then filtered to remove spikes and other spurious non-idealities, mixed up to the RF frequency, summed together, and sent off to the power amplifier for transmission. The specification for our specific system were a 25 MHz bandwidth, with a carrier frequency of 1.96 GHz, and was designed to support up to 15 users with data rates of up to 1.6 Mbit/second.<sup>2</sup> Please see Appendix A for a slightly more detailed description of transmitter operation.

By working closely with the designer of the following stages, it was determined that if the speed of the digital-to-analog converter could be increased, the requirement on the subsequent low-pass filter would be relaxed. Thus the specification on the DAC was raised from 100 MHz to 200 MHz, reducing the filter design from third order to first order. Examples like this illustrate the advantage of performing a system-level design rather than designing each block individually. The final design was built in a .25  $\mu\text{m}$  double-poly CMOS process, and a PCB board was designed to test the chip.

## 1.2 Thesis Organization

The different approaches to digital-to-analog converter design are discussed in Chapter 2. In Chapter 3, a description of the design methodology used for this specific digital-to-analog converter is given, as well as thoughts about how well the design will scale for future technologies. In Chapter 4, measurement and performance results from the prototype built in .25  $\mu\text{m}$  CMOS are described. Finally, in Chapter 5 conclusions from the project are drawn.

## Chapter 2

# Digital-to-Analog Converter Architectures

---

### 2.1 Overview

Digital-to-analog converters can be built using several different topologies. Each topology has various strengths and weaknesses, which can be summarized by several criteria: integral non-linearity, differential non-linearity, monotonicity, chip area (in terms of die size), settling time, and matching requirements. This section briefly examines the different architectures and discusses their pros and cons. Next, the topology chosen for the implementation – a current-steering architecture – will be examined in some detail.

### 2.2 Resistor-Ladder (Voltage Divider) Architecture

The resistor-ladder architecture is by far the simplest D/A implementation. It is

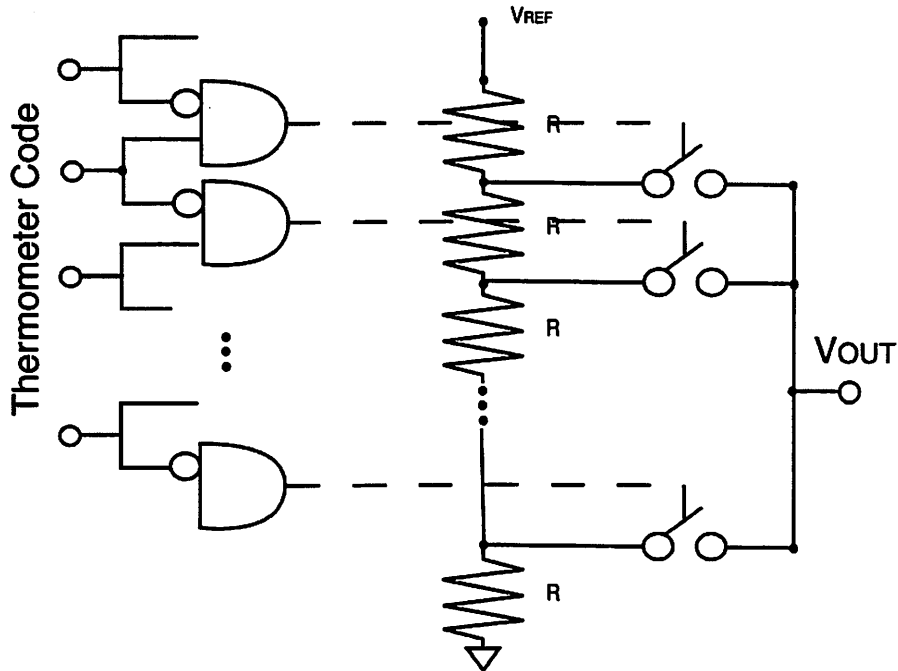


Figure 2.2a. Resistor-ladder DAC with thermometer decoding.

essentially a string of identical resistors in series, with the “top” resistor tied to power and the “bottom” resistor tied to ground [Figure 2.2a]. The nodes in between each resistor have different voltages depending on their proximity to power (i.e. nodes that are higher in the ladder have higher voltages), and by using thermometer or binary decoding on the digital signal one specific node can be selected as the correct analog voltage. The number of resistor elements determines the resolution of the resistor-ladder DAC; an  $m$ -bit DAC requires a ladder with  $2^m$  resistors. With this exponential growth in the number of elements, clearly for higher-bit requirements the size of the DAC grows prohibitively large.

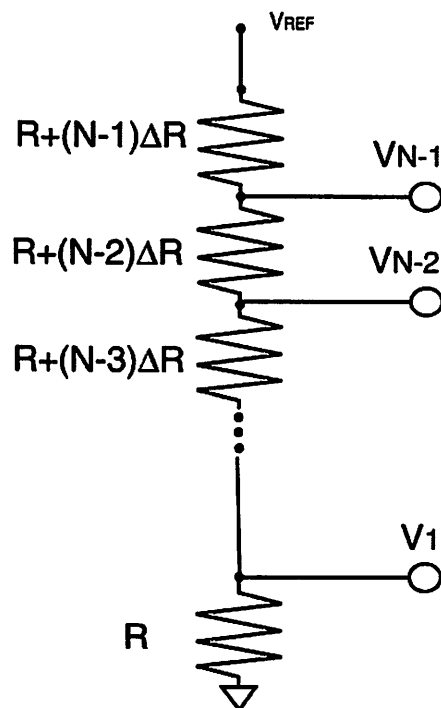


Figure 2.2b. Modeling resistor mismatch with a linear gradient.

Resistor-ladder converters have some benefits besides simplicity of design. They are inherently monotonic as long as the switching elements are designed correctly. The DNL is relatively low compared to other architectures; to calculate explicitly, it is most intuitive to model resistor mismatch by a linear gradient [Figure 2.2b]. Then the DNL can be calculated by subtracting the difference between two elements  $V_j$  and  $V_{j+1}$  from the ideal value of 1 LSB,  $V_{REF}/N$ :

$$\begin{aligned}
 V_j &= \frac{\sum_{k=0}^{j-1} (R + k\Delta R)}{\sum_{k=0}^{N-1} (R + k\Delta R)} * V_{REF} \\
 &= \frac{jR + \frac{j(j-1)}{2} \Delta R}{NR + \frac{N(N-1)}{2} \Delta R} * V_{REF} \\
 DNL_j &= V_{j+1} - V_j - \frac{V_{REF}}{N} \\
 &\approx \left( j - \frac{N-1}{2} \right) \frac{\Delta R V_{REF}}{R * N},
 \end{aligned}$$

given  $R \gg (N-1)\Delta R/2$ . From this derivation, it is clear that the worst-case DNL error occurs at  $j=1$  and  $j=N-1$ .<sup>3</sup> As the maximum DNL error must be less than 1 LSB to prevent the system from losing a bit of resolution.

The maximum INL can be calculated in a similar fashion. INL is defined as the deviation of the actual tap voltage from the ideal tap voltage, and thus for a linear gradient the INL is as follows:

$$INL_j = \frac{j}{N} V_{REF} - \frac{jR + \frac{j(j-1)}{2} \Delta R}{NR + \frac{N(N-1)}{2} \Delta R} V_{REF}$$



Using the same assumption as for the DNL calculation, it becomes apparent that  $INL_j$  reaches a maximum of  $NV_{REF}(\Delta R/8R)$  at  $j=N/2$ .

Of course, assuming a linear gradient is a rather naive model. A more accurate model uses random mismatch to model the resistors. Razavi [3] goes through a detailed derivation of random mismatch, and by using a Gaussian probability distribution function arrives at a maximum INL error of:

$$INL_{\max} = \frac{1}{\sqrt{4N}} \frac{\Delta R}{R} V_{REF}$$

This maximum error again occurs at  $j=N/2$ , though this value is a standard deviation and thus a likelihood measurement rather than a deterministic number [3]. Note that this error is significantly less than the error predicted with a linear gradient, as in the random mismatch model individual errors tend to cancel each other out rather than adding systematically.

Resistor-ladder DACs have fallen out of favor recently for several reasons. The absolute resistor values for our process varied by over 50% from slow to fast processes, which can create significant differences in power consumption from chip to chip. Furthermore, there was no matching information so determining mismatch ratios between resistors on the same chip was difficult. Finally, resistor-ladder DACs tend to be slow due to the large resistance load on the output. This can be avoided in several ways (small resistor values or large output buffers), but each method burns a significant amount of power. Thus, in modern CMOS processes, a resistor-ladder DAC is probably not the best solution.

### 2.3 Charge Division Architecture

A second main architecture for digital-to-analog converters is charge division. An example of a charge division circuit is shown in Figure 2.3a. Capacitors  $C_1$  to  $C_N$  are all identically sized, and their bottom plates can switch between  $V_{REF}$  and ground – allowing each capacitor to inject the amount of charge  $Q = C \cdot V = C \cdot V_{REF}$  onto the output node. Each switch is controlled by the digital thermometer code, and thus the number of nodes turned on determines the charge on the output node. The circuit operates in two stages. In stage one, the switch  $S_p$  turns on and all the capacitor bottom nodes are connected to ground. This discharges the array of capacitors. Next, in stage two,  $S_p$  turns off, and the digital thermometer code is applied to individual switches. This applies  $V_{REF}$  to all the capacitors up to the height of the thermometer code (and thus the value of the digital signal), and the voltage on the output node is  $j \cdot V_{REF} / N$  (where  $j$  is the “height” of the thermometer code) [3]. While this is an unweighted unit-element example of a charge sharing DAC, as with other topologies it can also be implemented in a binary version by using different capacitor sizes.

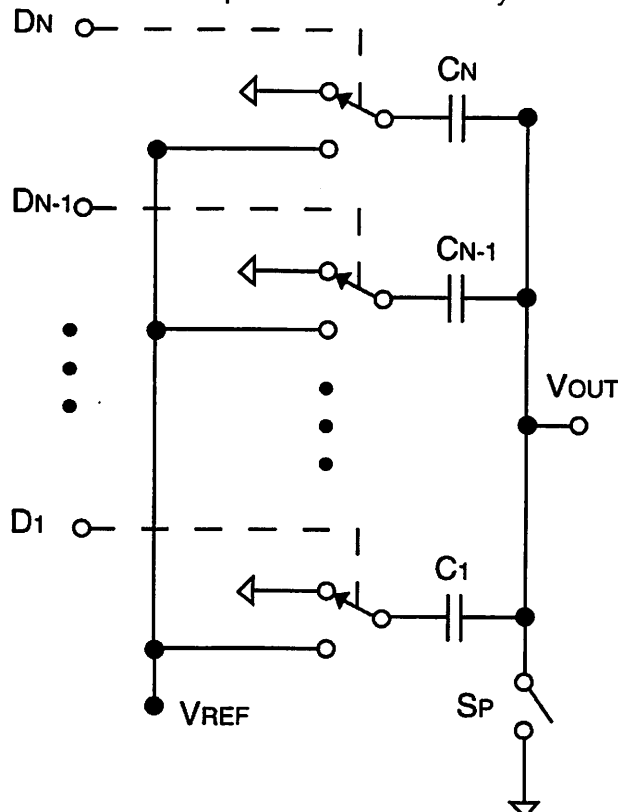


Figure 2.3a. Typical charge division DAC topology.

Clearly, the timing and control logic for charge sharing digital-to-analog converters is significantly more complex than that of resistor-ladder topologies. They also suffer from the same mismatch problems (both linear gradient and random mismatches occur) as resistor-ladder DACs, and derivations similar to those used in the previous section can be used to calculate INL and DNL. Another significant problem with charge-division architectures is that building decently sized capacitors in CMOS takes a substantial amount of chip area. Finally, several additional nonlinearities arise, including capacitor voltage dependence and the nonlinearity of the junction capacitance connected to the output node.<sup>4</sup> Implementing this topology would have used the most area and still been relatively inaccurate, so it was ruled out as a possible contender.

## 2.4 Current Division Architecture

A third type of the reference division architectures is current division. As with the previous two schemes, it operates by dividing a reference current among several transistors, and then selecting among the various current outputs. A binary implementation is shown below in Figure 2.4a. The four devices on the leftmost side draw  $4/7$  of the reference current, while the center and right sides draw  $2/7$  and  $1/7$  of the reference, respectively. The digital input then turns on some of the nodes, which pass their current ( $1I$ ,  $2I$ , or  $4I$ ) to the output node. The selected currents add, giving the analog current representation of the digital input. Sometimes a voltage output is desired rather than a current one, and to convert the output to a voltage simply requires attaching a resistor between the output node and power. Alternatively, a transimpedance amplifier may be used to provide the current to voltage conversion [3].

There are two major drawbacks with current division topologies. First, the stack of current dividing transistors positioned above  $I_{REF}$  reduces the available output voltage range, and thus can be impractical in low voltage circuits. This is dangerous: if the output voltage decreases enough, the current dividers may go out of saturation. Secondly, since each device divides the reference current,  $I_{REF}$  must be  $N$  (the total number of input levels) times greater than each of the output currents. This can require a huge device to provide the current source. Obviously, this is mainly a problem in a circuit with uniform current division, and can be avoided by using a binary implementation. Binary implementations, however, suffer from non-monotonicity and thus are avoided unless space is a primary concern. To avoid these drawbacks, designers generally implement current steering digital-to-analog converters rather than current dividing topologies. These current steering topologies will now be examined.

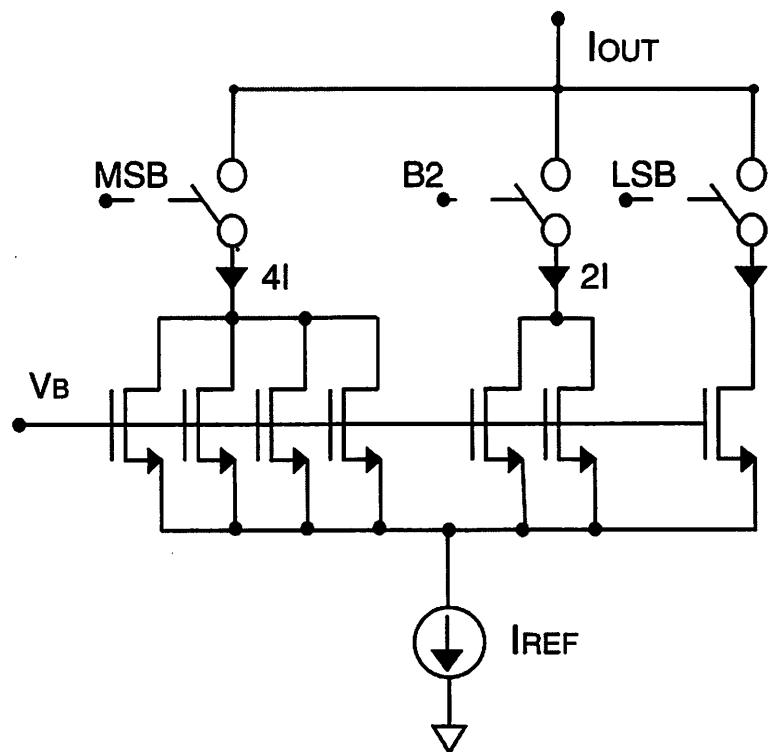
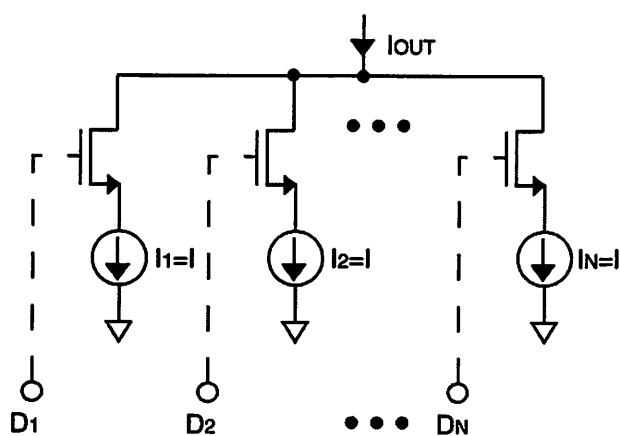


Figure 2.4a. Basic current division DAC circuit, using binary weighting.

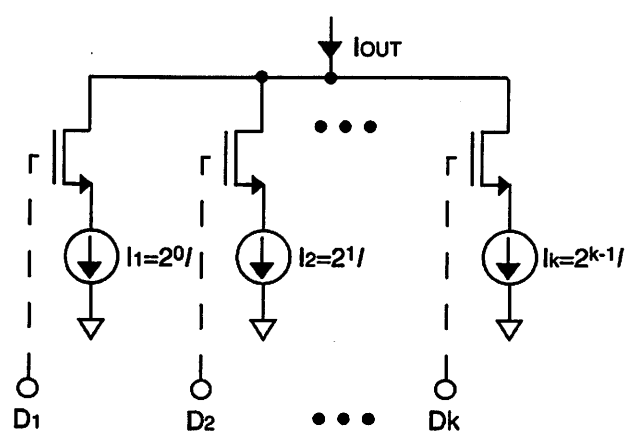
## 2.5 Current-steering Architectures

Current-steering architectures differ from current dividing architectures in that they replicate a reference current source rather than divide it. As shown in Figures 2.5a and 2.5b, the reference source is simply replicated in each branch of the DAC, and each branch current is switched on or off based on the input code. For the binary version, the reference current is multiplied by a power of two, creating larger currents to represent higher magnitude digital signals. In the unit-element version, each current branch produces an equal amount of current, and thus  $2^N$  current source elements are needed. When the digital input increases by 1 LSB, one additional current source is turned on. Although unit-element arrays have the drawback of much larger area, they also have the benefit of guaranteed monotonicity. At the midcode, when the digital input increments in either direction, the analog output is varied by one switch turning on or off. Binary arrays contain no such



Thermometer Code

Figure 2.5a. Unit-element current-steering DAC.



Binary Code

Figure 2.5b. Binary current-steering DAC.

guarantee, as during midcode transitions where the MSB source is varying one direction and the LSB sources are varying another direction (i.e. binary 011 to 100 transition corresponding to a decimal 3 to 4 transition), the output can change by substantially more than 1 LSB – causing a potentially substantial glitch [Figure 2.5c]. [3,5] Thus a binary array is not a good choice for applications when monotonicity is an important criterion.

A second advantage of unit-element DACs over binary-weighted types is matching. The matching requirement is significantly reduced, as 50% matching of the unit current source results in a DNL of under 0.5 LSB [Figure 2.5d].<sup>5</sup> For the binary-weighted version, the severity of the matching problem is determined by the weight of the bit [Figure 2.5e].

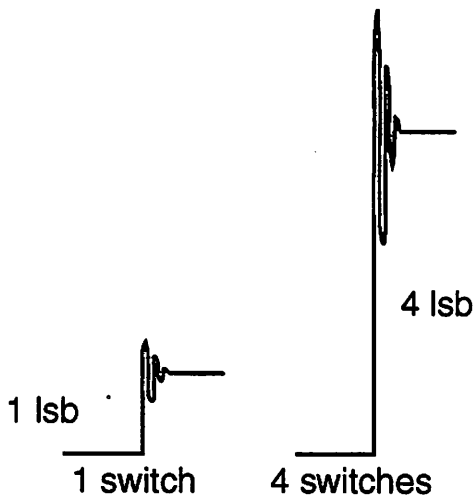


Figure 2.5f. Glitch advantage for thermometer-coded DAC.

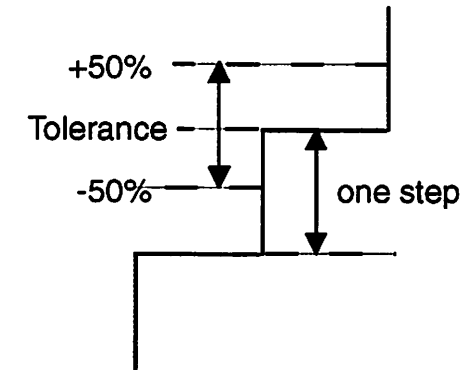


Figure 2.5d. Matching tolerance for thermometer coding.

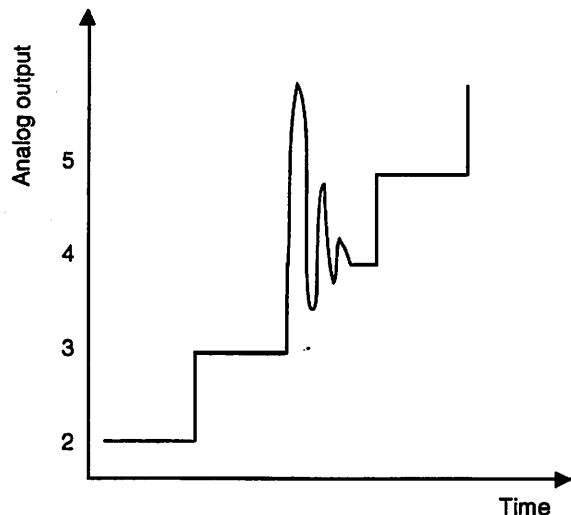


Figure 2.5c. Glitch disadvantage for binary-weighted DAC.

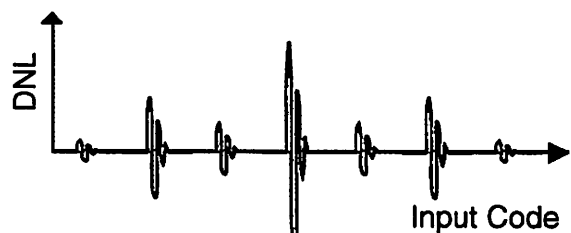


Figure 2.5e. Matching problems hurt DNL performance for binary-weighted DAC.

The matching of the MSB current source transistor must be extremely accurate, and is the DNL limiter for the entire digital-to-analog converter.

A third benefit of thermometer coding is that glitches do not contribute to nonlinearity. This is seen in Figure 2.5f, which illustrates that the size of the glitch in thermometer-coded DACs is proportional to the number of switches that switch at a given sample time. When the step is small (i.e. one LSB changes), the glitch is extremely small, and when the step is larger (i.e. four LSB's change), the glitch is larger. However, since the number of switches that change is proportional to the size of the signal step, the magnitude of the glitch is directly proportional to the amplitude of the signal step. Thus the glitch does not cause any nonlinearity in the converter's analog output signal.

For a more quantitative explanation, consider the argument described in Lin and Bult's paper, and illustrated in Figure 2.5g. [5] It shows a sine wave input signal, where  $x(n)$  is the digital input at sample  $n$  and  $y(n)$  is the analog output at sample  $n$ . For the purposes of this proof, the analog signal  $y(n)$  is broken into two parts, the current sources  $a$  that were on during the previous sample and the new current source  $b$  that are in the process of switching. Therefore the value of the analog

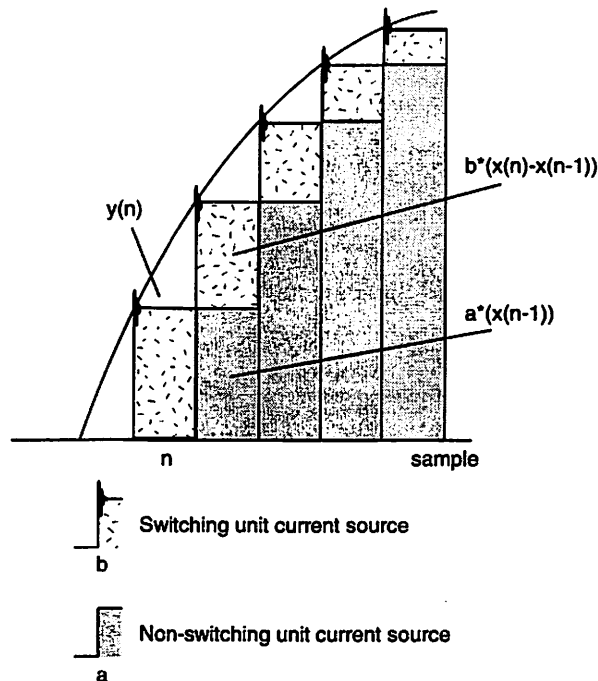


Figure 2.5g. Linear output function for thermometer-coded DAC.

signal due to the non-switching elements  $a$  is  $a * x(n-1)$ , with " $x(n-1)$ " representing the value of the digital input signal in the previous clock cycle. The switching current sources are labeled  $b$  in the diagram and for a given sample contribute  $b * [x(n) - x(n-1)]$  to the analog output signal, where " $[x(n) - x(n-1)]$ " is the difference between the current and previous values of the digital input signal. Given these two parts, the total value of the analog output is equal to:

$$y(n) = a * x(n-1) + b * [x(n) - x(n-1)]$$

As " $a$ " and " $b$ " are constant and independent of  $x(n)$ , the output signal  $y(n)$  is only linearly dependent on  $x(n)$ , and thus distortion caused by the glitches is effectively zero.

From the previous discussion it is clear that unit-element digital-to-analog converters have several advantages over binary-weighted converters. However, as mentioned previously, they have one large drawback: they take a tremendous amount of area.  $2^N$  current source elements are necessary to generate the correct output signal, and therefore for resolutions above 8 bits using an entirely unit-element array becomes less attractive. When the DAC specification calls for high resolution as well as good linearity and monotonicity, a combination of the binary-weighted and unit-element arrays can be used to create a segmented analog-to-digital converter.

## 2.6. Segmented Architectures

This section is a quick discussion of segmented DACs. For the converter built in this project, it was not necessary to segment the design as the specification called for a relatively low resolution of 8 bits, and furthermore there was not any constraint on chip area. In fact, even if size had been important, the overhead incurred by using segmentation for such a low



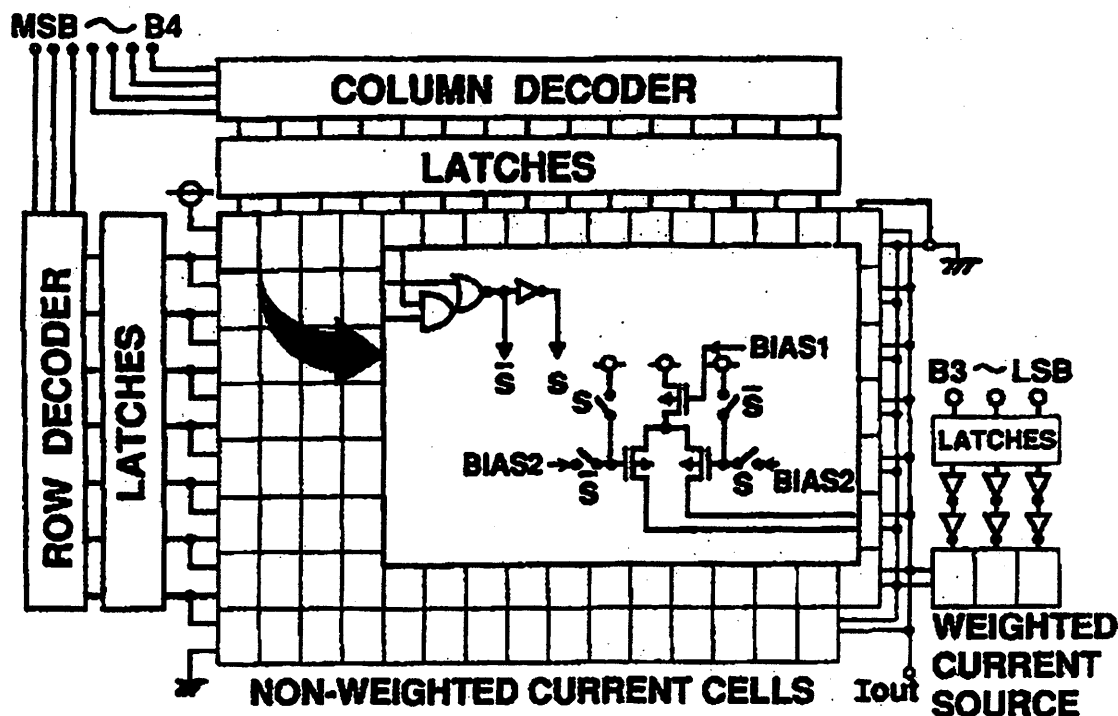


Figure 2.6a. An example of an 8-bit segmented architecture.

resolution would probably have resulted in more area usage than an entirely thermometer-coded design. With these caveats in mind, it is important to remember that future designs may require higher resolutions and thus the use of a segmented design may be desirable.

Segmented designs combine thermometer-coding and binary-weighted coding in a single DAC. Thermometer-coded converters have significantly better DNL, while binary-weighted converters use less area. To take advantage of this, thermometer coding is used for the MSBs of the converter, while binary weighting is used for the LSBs. A typical architecture is shown in Figure 2.6a.<sup>6</sup> This is an 8-bit DAC, with the current from the 5 MSBs created by the thermometer-coded array and the current from the 3 LSBs generated by the binary-weighted array. To make the array rectangular (as opposed to a long, two-dimensional line of 127 current sources), two binary-to-thermometer decoders are implemented, creating a matrix of current sources. The outputs from this array feed into the

node  $I_{OUT}$ , which is also sourcing current from the binary-weighted current sources. This output current then represents the analog output, which can be converted to a voltage if required by a resistor or transimpedance amplifier.

The obvious question that then arises is how much to segment the converter (i.e. how many bits should be represented by a unit-element array and how many by binary-weighted elements). This issue was analyzed in [5], via a MATLAB simulation. 1024 current sources were randomly generated, each with a mean of 1 LSB and a standard deviation  $\sigma$  of 0.2 LSB's. For the thermometer-coded architecture, 1024 individual cells were created, each with one current source, while for the binary-weighted architecture, 10 cells were created. The first cell representing the MSB had 512 current sources, the next MSB had 256, etc. Next, simulations were run to determine the INL and DNL for each topology. The results are shown in Table 2.6a. Note that the INL results are equal for the two topologies, while the DNL performance of the binary-weighted design is dramatically worse than the thermometer-coded. The author then estimated the area required to achieve a given linearity

Requirement	Binary Weighted	Thermometer Coded
INL	$16\sigma$	$16\sigma$
DNL	$32\sigma$	$\sigma$
Area (maintain INL=0.5-lsb)	$256 * A_{unit}$	$256 * A_{unit}$
Area (maintain INL=1-lsb)	$64 * A_{unit}$	$64 * A_{unit}$
Area (maintain DNL=0.5-lsb)	$1024 * A_{unit}$	$A_{unit}$

Table 2.6a. Area requirements to achieve specific INL/DNL as determined by Matlab simulation.

performance using a rough approximation provided by Pelgrom et al.<sup>7</sup>

$$\text{Area} \propto \frac{1}{\sigma^2}$$

These results are also in Table 2.6a. Notice that to achieve a given INL the analog area is the same for the two topologies, while to achieve a given DNL specification the analog area for the binary-weighted device is 1024 times as large as the thermometer-coded area.

Taking their cue from BWRC design methodologies, Lin and Bult used these Matlab results to create an area versus segmentation plot. This plot allows the derivation of the optimal degree of segmentation. This is illustrated in Figure 2.6b. On the far left of the chart, where the segmentation is 0% (i.e. completely binary-weighted implementation), the total area is the same as derived in the table, or  $2^{10} \cdot A_{\text{unit}}$ . This point is dominated by the size required to meet the DNL specification. As segmentation increases, the normalized area required decreases, as the thermometer-coded array uses much less area than the savings generated by reducing resolution necessary in the binary-weighted section. However, at a normalized area of  $2^6 \cdot A_{\text{unit}}$ , the INL requirement becomes the limiting specification, which

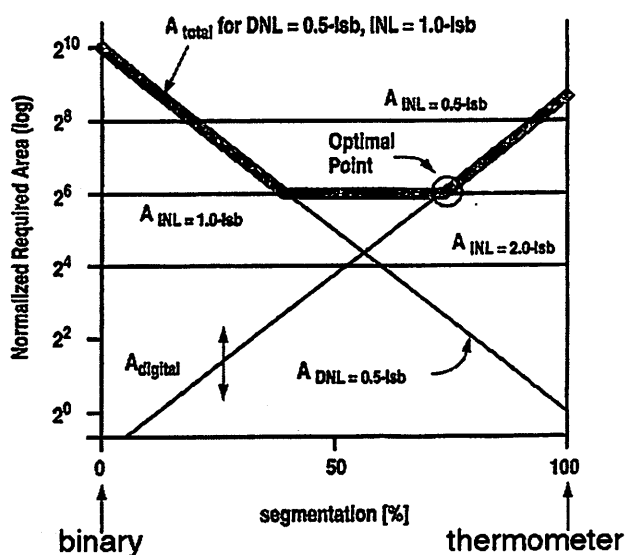


Figure 2.6b. Area required versus percentage of segmentation.

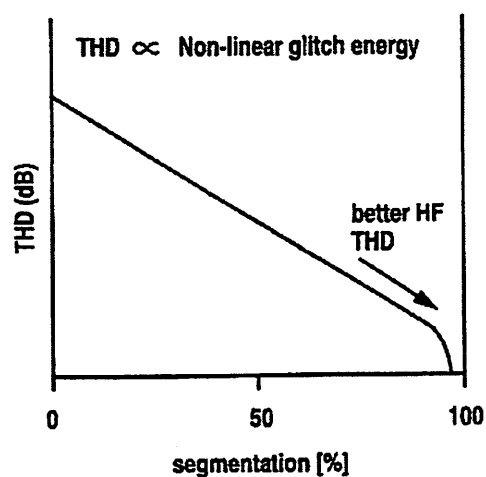


Figure 2.6c. Total harmonic distortion as a function of segmentation.

is equivalent for both the binary-weighted section and the unit-element array. Thus the area remains constant for certain degrees of segmentation, as shown by the straight line in the center of Figure 2.6b. Eventually, however, the decoding logic demanded by the thermometer-coded section begins to dominate, with an area of  $2^{M \times} A_{\text{decode}}$ , where  $A_{\text{decode}}$  is the area of the digital decoding logic in each current source cell. As the number of current sources exponentially increases with a higher percentage of segmentation, this digital logic area is the dominant factor in the size of the converter.

Given this plot, it would seem that any point on the horizontal line would be an optimal level of segmentation. However, this is not correct. As mentioned in Section 2.5, the glitch performance of thermometer coding is significantly better than that of binary-weighted sections. Thus the more bits that are contained in the LSB section, the higher the glitch frequency. In fact, each additional binary-weighted bit contributes a factor of two to the total harmonic distortion (THD) figure for the circuit, as shown in Figure 2.6c. [5] Therefore, the optimal degree of segmentation occurs at the rightmost part of the horizontal line in the plot, which is still a minimum area (limited by INL requirements), meets the DNL specification, and has the smallest THD.

With this information, designing an optimized segmented digital-to-analog converter becomes relatively simple. After laying out a single current source, the respective areas of the digital and analog parts can be calculated. Next, matching accuracy for the specific design and fabrication process should be determined to calculate the acceptable standard deviation for individual current sources. Finally, plotting this data on a graph similar to 2.6b produces the optimal degree of segmentation. At this point, the high-level decision (i.e. what degree of

segmentation to use) has been completed and the design can be implemented.

## Chapter 3

# DAC Circuit Design Techniques

---

### 3.1 Overview

Despite the array of digital-to-analog converter topologies available, it was relatively simple to pick which topology to use for this project. Process tolerances and area considerations rule out resistor-ladder and charge division architectures, and there are uniformity issues with current division topologies. This leaves the obvious choice of current-steering architectures, which was in fact chosen for the transmitter implementation. This section discusses design decisions made once the architecture choice was finalized, with emphasis placed on low power and technology scaling issues.

Figure 3.1 shows the 100%-segmented (i.e. entirely unit-element) DAC architecture chosen for this design. The first block the digital input signal encounters is a four-to-one multiplexor, which converts a 32-bit wide 50 MHz digital input signal into an 8-bit 200 MHz signal. This block is strictly for testing purposes; after examining waveforms generated by the Hewlett-Packard logic analyzer, it was ascertained that the signal produced at 200 MHz was more or less a sine wave. Compounding the poor signal generator quality at extremely high clock rates is the difficulty in getting a signal that fast onto the board. To allay these fears, the multiplexor was added to reduce the input signal's clock rate to 50 MHz. Of course, once the digital backend circuits are complete, they will generate the 200 MHz signal

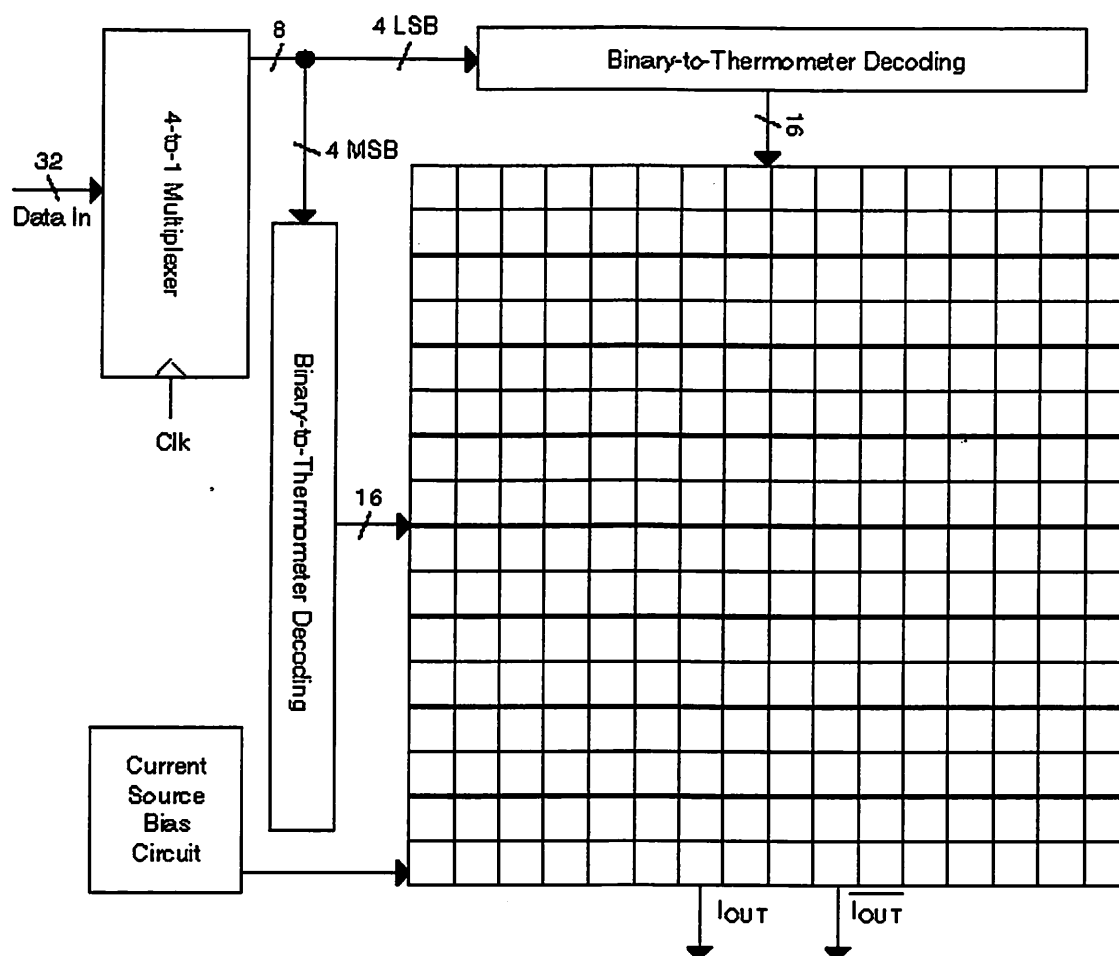


Figure 3.1 Block diagram of current-switched segmented DAC architecture used.

on-chip and thus the multiplexor block can be removed.

After the multiplexor, the 8-bit digital signal undergoes binary-to-thermometer decoding. The four most significant bits drive the column decoder, while the four least significant bits drive the row decoder. After decoding, there are 16 row and 16 column bits, which act as control signals for the current source blocks. The number of current sources active at a given time is determined by the value of the input signal; for a digital input of 00000000, there will be zero current in  $I_{OUT}$  and  $255 \cdot I_{REF}$  flowing in node  $\overline{I_{OUT}}$ . For an input of 11111111 the situation is reversed, and turning on the correct number of current sources can represent all digital values between these two extremes.

The current source blocks are seen in the 16 by 16 matrix, and the circuitry included in each is shown in Figure 3.2. Each current source contains digital decoding circuit that determines whether to turn on  $I_{OUT}$  or  $\overline{I_{OUT}}$  based on the row and column signals. The analog part of the cell consists of a cascoded current source and a differential switch; the current source is biased with a standard high-swing biasing scheme. PMOS devices were

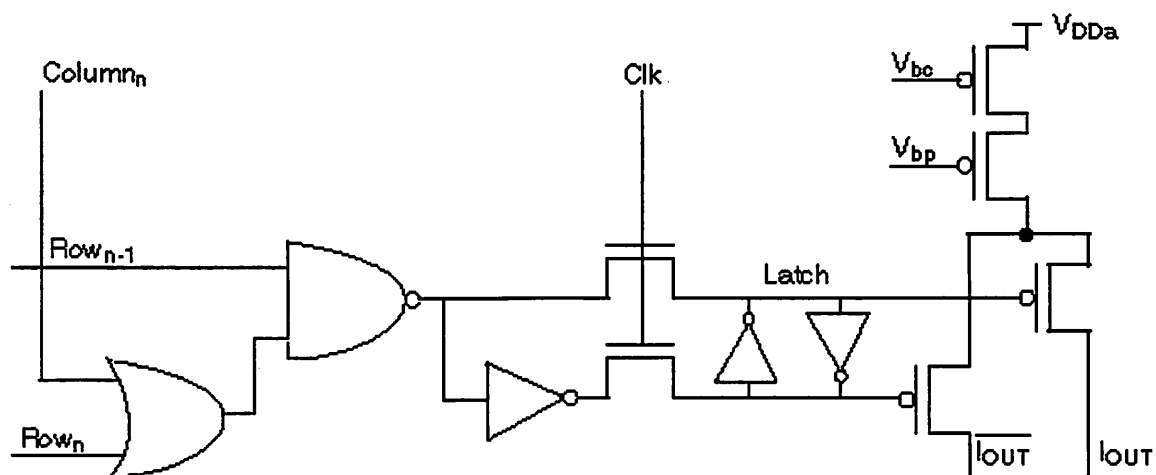


Figure 3.2. Current source cell.



used in the current source because they reduce crosstalk. Previous designers had tried both NMOS and PMOS for the current sources and found that the crosstalk from digital portions of the chip is much less in the P-type devices. This makes logical sense as the PMOS devices are built in an n-well and therefore is shielded from the substrate.<sup>8</sup>

The intimate details of the transistor sizing, matching requirements, and power reductions considered for the design will be discussed next.

## 3.2 Analog Aspects

Having decided on the 100% segmented topology, the majority of time spent on the design dealt with the analog aspects of the current source cell. Decisions were made on the amount of current each source should generate and how small the transistors could be and still meet matching requirements. Each area will be explored in detail.

### 3.2.1 Power Consumption

The dominant source of power consumption in the digital-to-analog converter is the analog portion of the current source matrix. Since the design is differential, each element is sourcing the reference current regardless of the output value. Thus a rough estimate of total power consumption (excluding power consumed by the digital blocks, which is almost negligible due to the dynamic nature of the CMOS standard cells) is easy: simply multiply the number of current sources by the reference current by the supply voltage:

$$P_{TOT} = 2^N * I_{REF} * V_{DDa},$$

where  $N$  is the resolution of the DAC (obviously this only holds for a 100%-segmented current steering topology). From this derivation it is clear that the power consumed varies

directly with the reference current chosen, and thus the lowest current possible should be used to minimize power consumption.

### 3.2.2 Matching requirements

From the previous discussion, it is apparent that ideally the reference current should be as low as possible. Unfortunately, there are several constraints that provide a minimum value for the reference current, including matching requirements, signal-to-noise ratio, and speed considerations. Matching requirements will be analyzed first.

As defined by Pelgrom, mismatch "is the process that causes time-independent random variations in physical quantities of identically designed devices." [7] Essentially, this means that each current source in the matrix generates a current that varies slightly from the desired current,  $I_{REF}$ . Therefore the current sources have to be designed in such a way that random variations do not degrade the performance of the circuit below its specifications.

Pelgrom's paper has become the de facto standard for analysis of transistor matching, and thus his formula for the standard deviation of saturation current for two identically sized devices was used for the design. This formula is:

$$\frac{\sigma^2(I_d)}{I_d^2} = \frac{4\sigma^2(V_{T0})}{(V_{GS} - V_{T0})^2} + \frac{\sigma^2(\beta)}{\beta^2},$$

where  $\sigma^2(V_{T0}) = \frac{A_{VT0}^2}{WL} + S_{VT0}^2 D^2$  and  $\frac{\sigma^2(\beta)}{\beta^2} = \frac{A_\beta^2}{WL} + S_\beta^2 D^2$ . Most of these variables are

process-dependent constants. Using these results, an equation for the minimum size device

that still provides a reasonable current standard deviation can be determined. Bastos conveniently provides this in his paper<sup>9</sup>:

$$(WL)_{\min} = \frac{1}{2} \left[ A_{\beta}^2 + \frac{4A_{VT}^2}{(V_{GS} - V_T)^2} \right] / \left( \frac{\sigma_I}{I} \right)^2$$

$A_{\beta}$ ,  $A_{VT}$ ,  $V_{GS}$ , and  $V_T$  are process parameters, while  $I$  is the current generated by a given source and  $\sigma_I$  is the relative standard deviation of one current source.

The constraint on  $I$  is that we want to minimize current to minimize power consumption. However, the constraint on  $\sigma_I$  is not as clear. Examine the following argument. Each current source is controlled by bits of the digital input word, and must have an error less than 1/2 LSB. This requirement is simple to meet for the few sources controlled by the digital LSB's. However, for those controlled by the MSB, this requirement becomes relatively difficult:

$$\frac{\Delta I_{MSB}}{I_{MSB}} \leq \frac{1/2LSB}{2^{N-1}LSB} = 0.39\%$$

$N$  is the DAC accuracy, and thus in our design  $N = 8$ . Now, our MSB current source is essentially built by wiring up in parallel  $2^{N-1}$  current sources, and therefore the INL can be approximated by adding the variances of the 128 current sources [8]:

$$INL \approx \sqrt{2^{N-1}} \left( \frac{\sigma_I}{I} \right) LSB$$

Using standard probability theory to assume that each current source has a value that follows a normal distribution, the design yield becomes an important design characteristic. A coarse estimate can be produced with by calculating the  $p(INL \leq 1/2LSB)$ , but Bastos performed

a Monte Carlo simulation to come up with a more accurate figure of 0.3% for  $\left(\frac{\sigma_I}{I}\right)$ . This resulted in a yield estimation of 99%, as shown in Figure 3.3. Thus, given Pelgrom's equation for  $(WL)_{\min}$ , Bastos' value of 0.3% for  $\left(\frac{\sigma_I}{I}\right)$ , and the process parameters for the fabrication facility being used, the minimum area of the current source needed to fulfill matching requirements can be determined. For our process (ST Microelectronics .25  $\mu\text{m}$  CMOS) and relatively low accuracy requirements, this turned out to be miniscule: the minimum device area was only  $2 \mu\text{m}^2$ .

### 3.2.3 Signal-to-Noise Ratio

To maintain an accuracy of at least 8 bits of resolution, it was imperative that the signal-to-noise ratio of the DAC remained above 48 dB. To provide a sufficient safety margin, the design aimed for a signal-to-noise ratio of 60 dB. The signal power was easily calculated: it was simply a function of the chosen output reference current.

$$P_{sig} = \frac{(I_{ref} * 2^N)^2}{2}$$

In the previous equation,  $N$  is the number of bit of accuracy (and in this design, the number of current sources turning on).

The noise power is dominated by thermal noise in the cascoded current sources. The current source transistor itself is the main noise contributor in a given current cell and thus we approximate the thermal noise power using just this device. Referring to Gray and Meyer,

the power of the noise is given by:

$$P_{noise} = 2 * 2^N * \frac{2}{3} * 4kT * g_m \Delta f^{10}$$

The various factors are as follows. The leading coefficient factor of two is used to account for the noise added to each current source by the biasing circuit.  $2^N$  is again the number of sources actively contributing to the noise figure.  $\Delta f$  is the frequency of the baseband signal (12.5 MHz for our radio, but set at 30 MHz to provide a comfortable safety margin), while  $g_m$

is estimated as  $\frac{2 * I_{ref}}{V_{dsat}}$ . The  $\frac{2}{3} * 4kT$  factors are all constants.

Once the signal and noise powers are derived, the signal-to-noise ratio is just that: a ratio:

$$SNR = 10 * \log \left( \frac{P_{sig}}{P_{noise}} \right)$$

With a 10  $\mu$ amp reference current source, this value turned out to be remarkably high, on the order of 84 dB. In fact, using these rough estimations, the reference current source could be reduced to 100 nanoamps and still provide the desired SNR of 60 dB!

At first, simulation did not bear out these promising estimations, as something in the design was causing a tremendous amount of noise. Hspice simulations showed a SNR of around 20 dB. The reader can imagine the ensuing terror. However, consultation with the esteemed Mr. Vanderhaegen provided the solution to the problem: the noise generated by the current source biasing circuit was adding coherently into every current source since there was no capacitor on the bias node. Adding a large bypass capacitor to this line resolved the

problem and Hspice kindly reported an SNR for the entire DAC of 62 dB. This was quite sufficient given the 48 dB requirement provided by the system designer.

### 3.2.4 Speed Requirements

This was a relatively high-speed digital-to-analog converter, and thus the output current needed to charge the capacitor load of the subsequent filter stage was significant. For this stage, working closely with the designer of the filter stage proved profitable; he estimated the output capacitance the DAC would have to drive at around 5 picofarads. To determine the amount of current necessary to charge this capacitor fast enough, the standard formula for charging a cap was used:

$$I = C * \frac{\Delta V}{\Delta t}$$

Assuming a worst-case 2 V (peak-to-peak) swing and a 200 Mhz clock, the current needed to drive the output load is:

$$I = 5pF * \frac{2V}{\frac{1}{200e6Hz}} = 2mA$$

This worst case occurs when the DAC transitions from a digital input value of zero to an input value of 255 (i.e. all the current sources steer output current through the positive terminal). From the formula, this transition requires at least 2 mA of current to drive the load in 5 ns. This drove the design choice of 10  $\mu$ A per current source, which results in 2.56 mA of output current – sufficient to drive the output load fast enough.

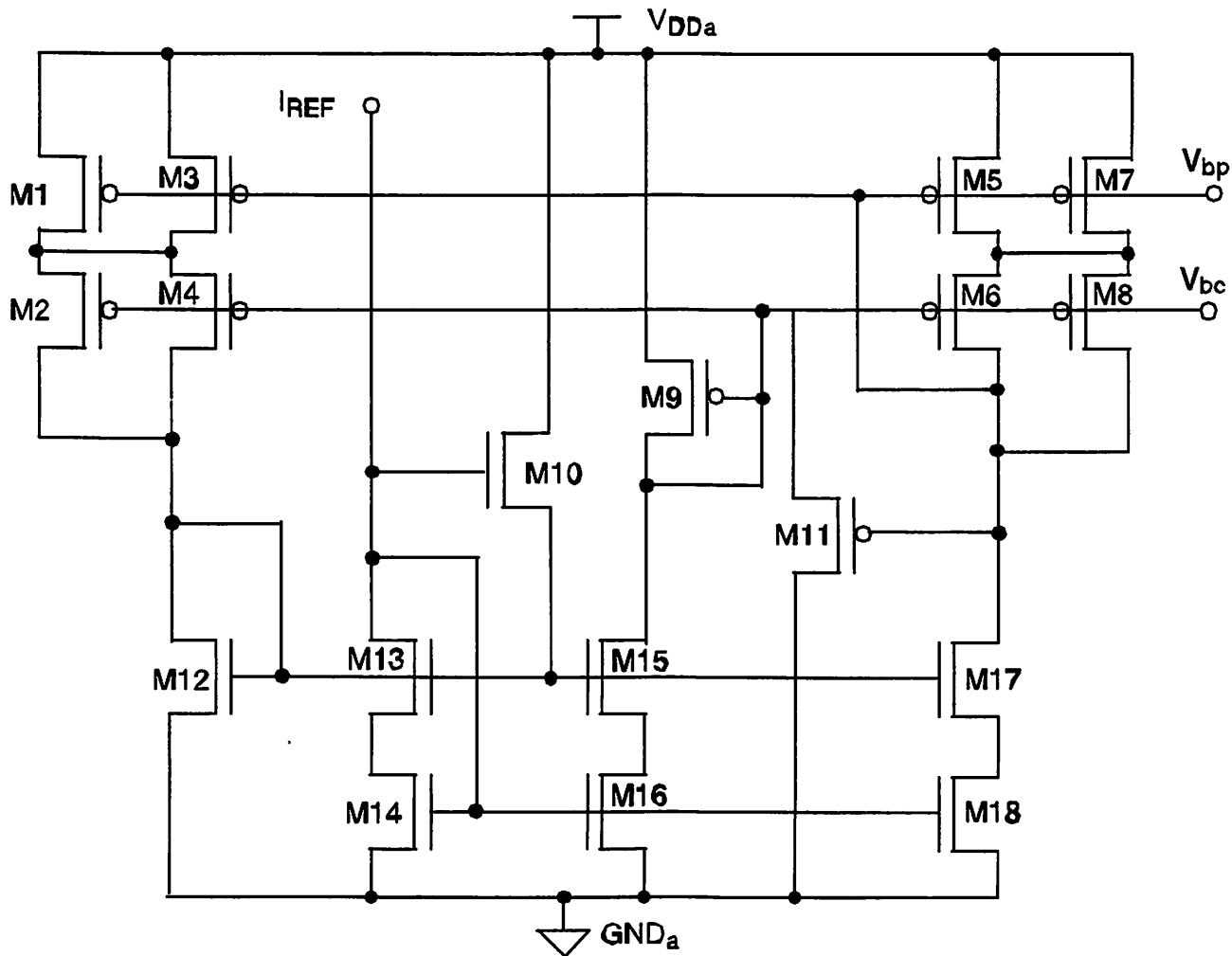


Figure 3.3. The high-swing cascode biasing circuit utilized in the DAC.

### 3.2.5 Bias Circuitry

The current source transistor and its cascoded neighbor need to be biased to produce the desired output current. For this purpose, a standard high-swing bias circuit was used, as shown in Figure 3.3. The circuit uses a reference current from off-chip. Figure 3.3 does not contain all the transistor details; please refer to Appendix B, which contains actual schematics. The circuit works as follows. By making the  $W/L$  ratio of M9 smaller than the device it is biasing (in this case M6/M8), the voltage on the gate of M6/M8 is decreased (by one threshold voltage).<sup>11</sup> This provides more headroom by raising the voltage on the source

of M6/M8 from  $V_{dda} - \Delta V - V_T$  to  $V_{dda} - \Delta V$ . Calculations show that the ratio between M9 and the device it is biasing should be  $\frac{1}{4}$ , but simulations showed that a lower ratio (approximately  $\frac{1}{6}$ ) provided better operation.

To improve matching with the transistors they are biasing, M5/M7 as well as M6/M8 are the same sizes as the devices they setting the gate voltage of, but placed in parallel to achieve the desired width. The same reasoning was used for transistors M12 and M9; six shorter devices were placed in series rather than using one long device.

The last aspect of the circuit that requires explanation involves devices M10 and M11. These are minimum sized devices that function as "startup circuits." Depending on initial voltages presented to the chip, the biasing circuit may initially be in cutoff mode rather than the desired saturation. The following illustration will take use M10 as an example. If the circuit is in cutoff, the voltage on the gate of M10 will still be above threshold and thus it will begin to conductor current. This will yank the voltage on the source of M10 towards  $V_{Dda}$ , lifting the gate voltages of M12, M13, M15, and M17 above cutoff and into the desired operating point. The circuit will now be functioning correctly, and M10 will no longer be a factor.

### 3.3 Digital Aspects

Clearly, the design time for a digital-to-analog converter is dominated by the analog components. However, given its digital interface to the world, standard digital cells play a role as well. In this section the digital components of the DAC will be briefly discussed. All the digital blocks were implemented with standard cells, with the exception of the decoder



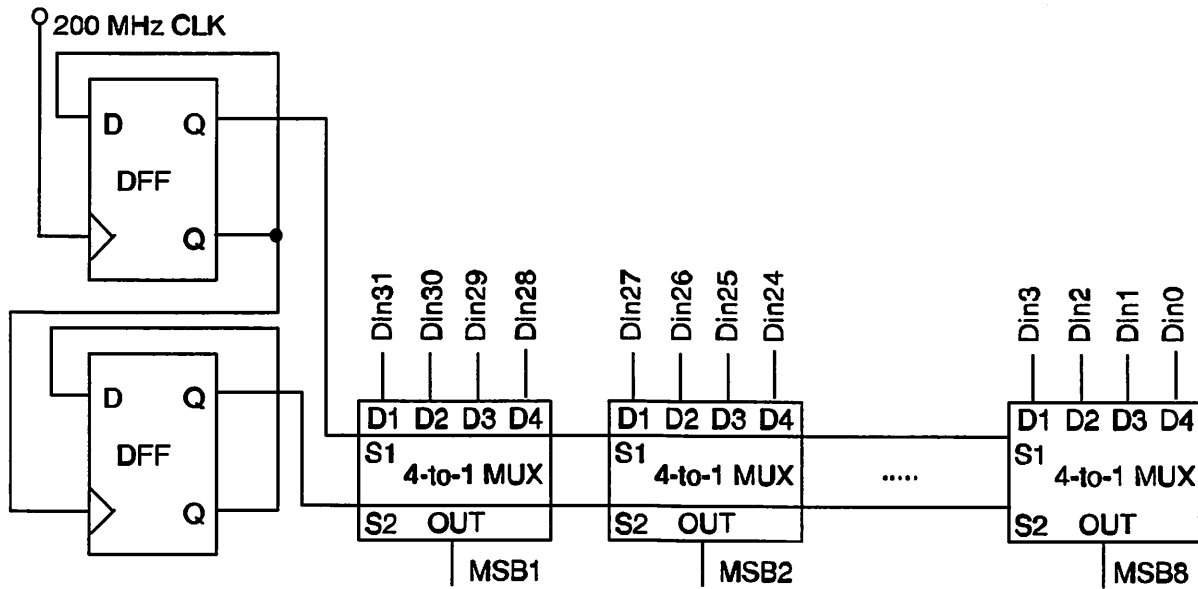


Figure 3.4. 32-to-8 bit multiplexer setup. This allows 32-bit, 50 MHz input signals to be seen as 8-bit, 200 MHz signals to the DAC.

blocks in the current sources. These were full custom to minimize the area – an important consideration given that there are 256 of them!

### 3.3.1 4-to-1 Multiplexor

The 4-to-1 multiplexor is not an integral component to the DAC; in fact, when the digital backend circuitry is completed and placed on the same chip as the analog components, the mux will not be necessary at all. However, our pattern generating system here at the BWRC only produces clean signals up to about 50 MHz. Therefore to ensure that the DAC is testable at its specified speed of 200 MHz, a 4-to-1 mux was added, as shown in Figure 3.4.

The operation is simple. There are eight 4-to-1 multiplexors, one for each digital input bit. The two flip-flops are wired as a two-bit counter, and the  $Q_{BAR}$  outputs of the

flip-flops are used as the select signal inputs to the eight muxes. The digital inputs change every 20 ns, while during each clock cycle the multiplexors choose one of the four input signals. This results in each 8-bit input being fed to the DAC at 5 ns intervals, resulting in the desired 200 MHz input signal.

### 3.3.2 Binary-to-Thermometer Decoding Logic

Once the 200 MHz 8-bit digital input stream is established, it must be converted from binary encoding to thermometer encoding. To control the 256 current sources, the eight bits are divided into the 4 rows and 4 columns, each of which is fed to a binary-to-thermometer decoder block. The logic operation that each decoder performs is shown below in Table 3.1 (note that the table shown is only for a 3-bit decoder; the reader can easily extrapolate the table to understand how a 4-bit decoder would work).

The decoder is implemented in the same fashion as the multiplexor, with standard

<u>A</u>	<u>B</u>	<u>C</u>	<u>T1</u>	<u>T2</u>	<u>T3</u>	<u>T4</u>	<u>T5</u>	<u>T6</u>	<u>T7</u>
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
1	0	0	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1

Table 3.1. Binary-to-thermometer decoding for a 3-bit binary coding.

cell blocks connected manually together. Please see Appendix B for a schematic of the completed decoder.

### 3.3.3 Current Source Decoder

Once the input is translated into thermometer code, the last remaining digital aspect is the decoding logic contained in each current source cell. The goal of this logic is very simple: the number of current sources that turn on (i.e. the positive output terminal enabled) should be equal to the value of the thermometer input code. The best way to describe the logic that implements this functionality is as follows. If the previous row is high and either

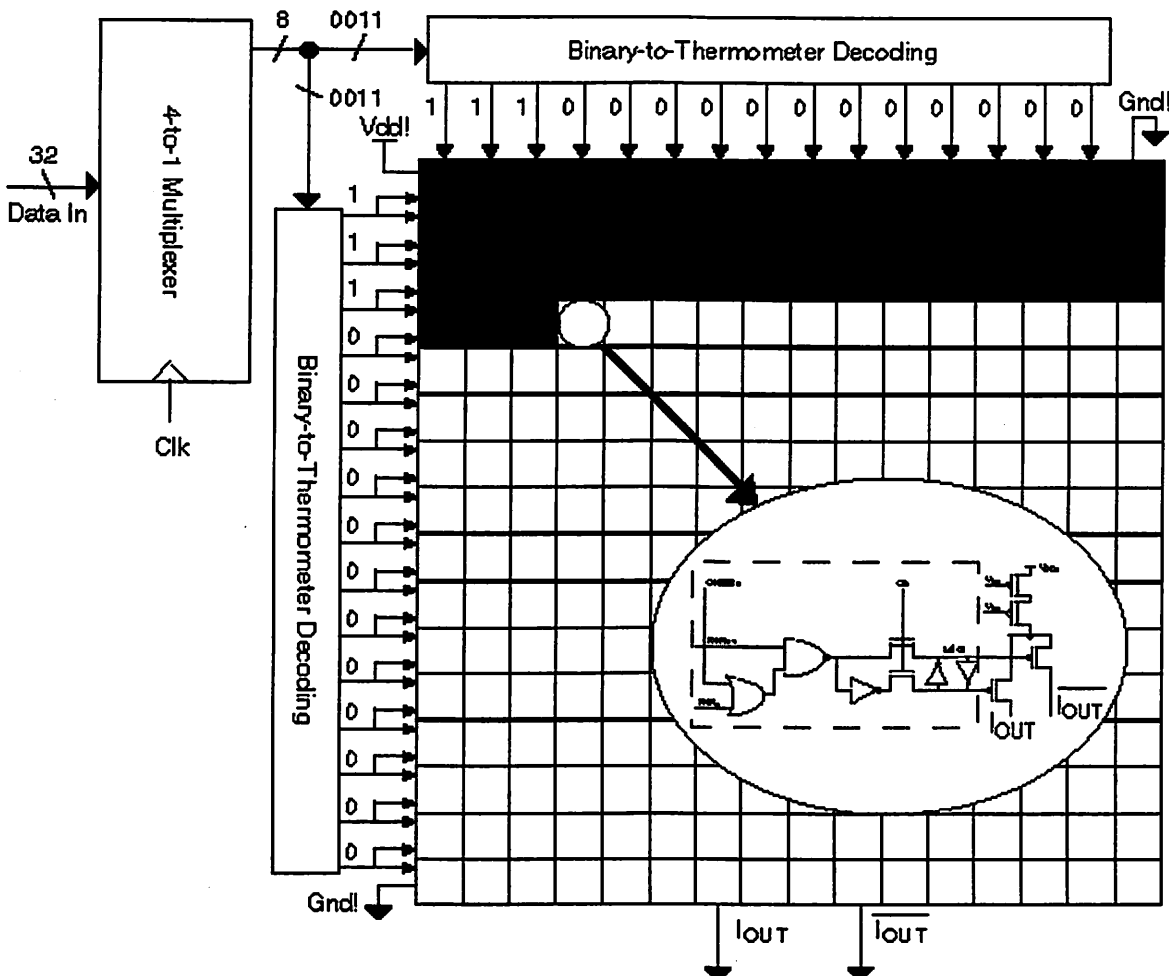


Figure 3.5. An example illustrating the implementation and functionality of the current source decoding logic. The filled-in squares represent the current sources that are funneling current through the positive output terminal.

the current column strobe or the current row strobe is high, then the current source should source current through the positive output terminal. Rewritten in logical fashion,

$$OUT = ROW_{N,1} \text{ AND } (ROW_N \text{ OR } COL_N)$$

After decoding the three input signals, the cell latches the result with a simple inverter pair. The two nodes formed by the latch control the analog switches, thereby passing the current through either  $OUT$  or  $OUT_{bar}$ .

Figure 3.5 provides an example of the decoder logic functionality for a given binary input. The eight input bits are separated into the four MSBs and LSBs and passed to the thermometer-decoding row and column controller blocks. These blocks translate the binary input into thermometer code and pass the values into the current source array. Each current source cell then decodes the three inputs and latches the output, using the value stored to control which terminal is enabled.

### 3.4 Layout Issues

Much of the layout for the DAC was straightforward. All the digital blocks other than those used in the current sources were laid out using standard cells from the ST cell library. Place and route for these cells was performed by hand, however, due to the low number of cells needed and the difficulty in using the automatic place and route tools (the learning curve would have been a few days, and there was a very short time till tapeout). These blocks, which included the row and column decoders, were also laid out with the intent of having equal spacing between the outputs, allowing for easy connection to the current source array. The last thing this designer wanted to do was route 48 digital signals to

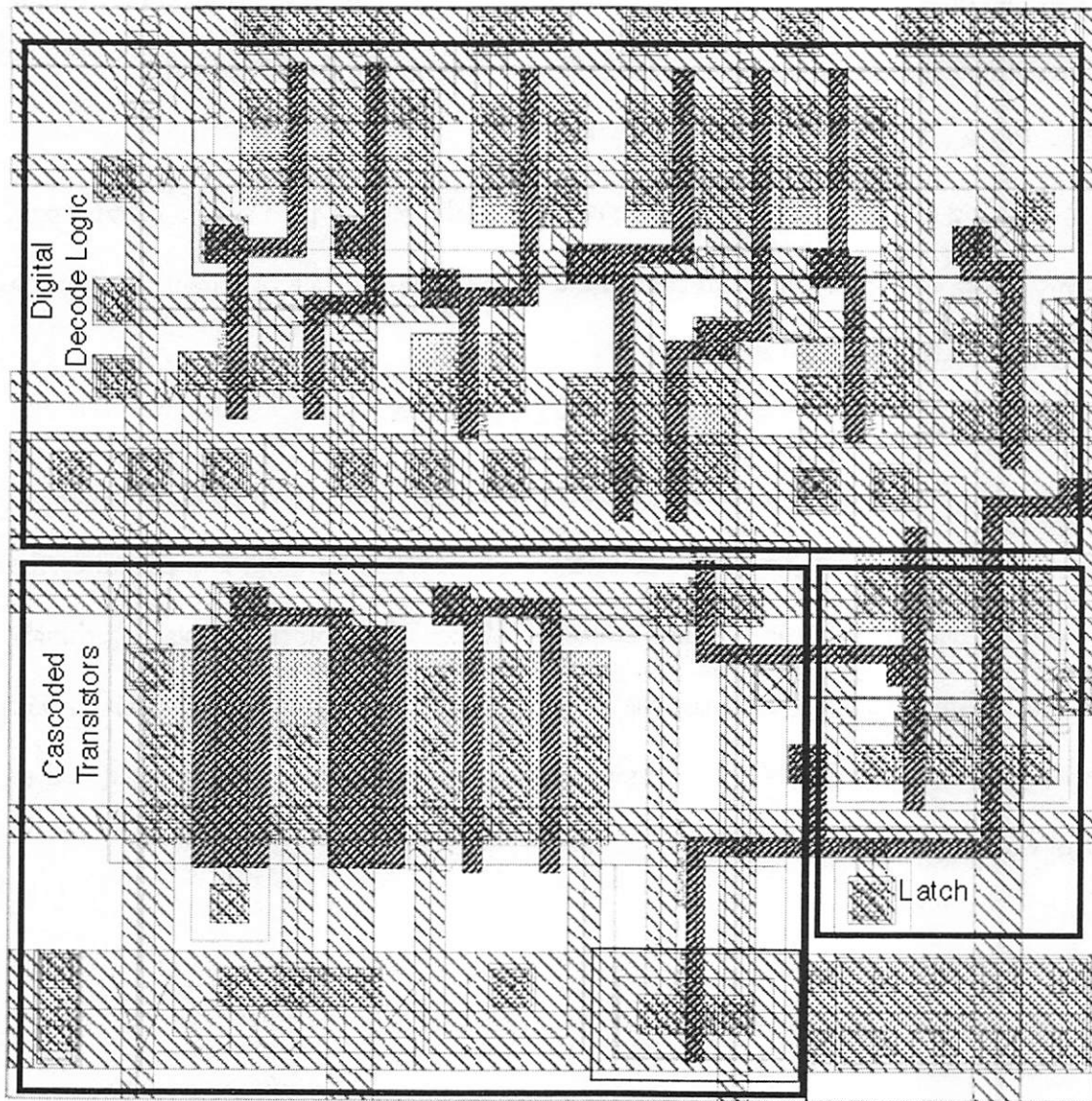


Figure 3.6. Current source cell layout.

the current source array by hand – the possibility of a mistaken connection was simply too large!

On the other hand, the layout of the current source cell was extremely time intensive (Figure 3.6).<sup>12</sup> There were two main goals for this block: a square form factor, so the current sources could be tiled easily, and a small area, as this block would be repeated 256 times! All the input and outputs pins were arranged so that simply abutting the cells together could

form the current source array. To minimize the area, a full custom design was performed, including the digital portions of the current source. After an initial design pass, the cell was 19  $\mu\text{m}$  by 19  $\mu\text{m}$ ; after several revisions, the size was reduced to 14  $\mu\text{m}$  by 14  $\mu\text{m}$  – a savings of nearly 50%, quite significant for a cell that will be duplicated  $2^N$  times! However, the transmitter chip turned out to be pin-limited, and thus the area savings was not relevant for this prototype. In the future, when the transmitter is integrated with the digital back end and receiver front end, this area savings could be useful.

### 3.5 Technology Scaling

One of the major frustrations of analog circuits is that they do not scale with technology as nicely as digital circuits do. Of course, analog designers do not mind this fact as it provides them with excellent job security! Regardless, one of the goals of this design was to make a DAC that was scalable into the future. By going with a relatively simple design that only utilized unit-weighted current sources, this goal was achieved. To achieve the same performance on a new process, the only element in the design that would need to be redone is the analog portion of the current source cell. Everything else should scale cleanly: the current source digital decoding logic, the row and column decoders, etc.

There are several reasons why the analog devices in the current sources – just two transistors! – require a full redesign. First, as channel lengths decrease, the current drive of the transistor decreases. Therefore to maintain the same speed (assuming the DAC is driving a similar load), the current source cannot be arbitrarily scaled down. A second reason involves matching requirements. As devices shrink, the matching accuracy between

individual current sources becomes poorer. This hurts INL performance and decreases the SNR, and provides a finite limit for how much the designer can shrink the transistor current source. Finally, the current source cell is the critical component of the design in terms of area – there are  $2^N$  of them in the converter! Therefore a tight custom layout is critical for minimizing area consumed by the current source array, and changing the relative size of any transistors in the cell result in the need for a complete redesign. However, once the current source is completed, simply tiling the individual sources and connecting the row and column decoders can create the final array. Simple!

Given the relative ease of redesigning the unit-element DAC to scale with technology, it would seem to make sense to create an automatic layout generator for the device. This was attempted several years ago at Berkeley, with rather unappealing results. A

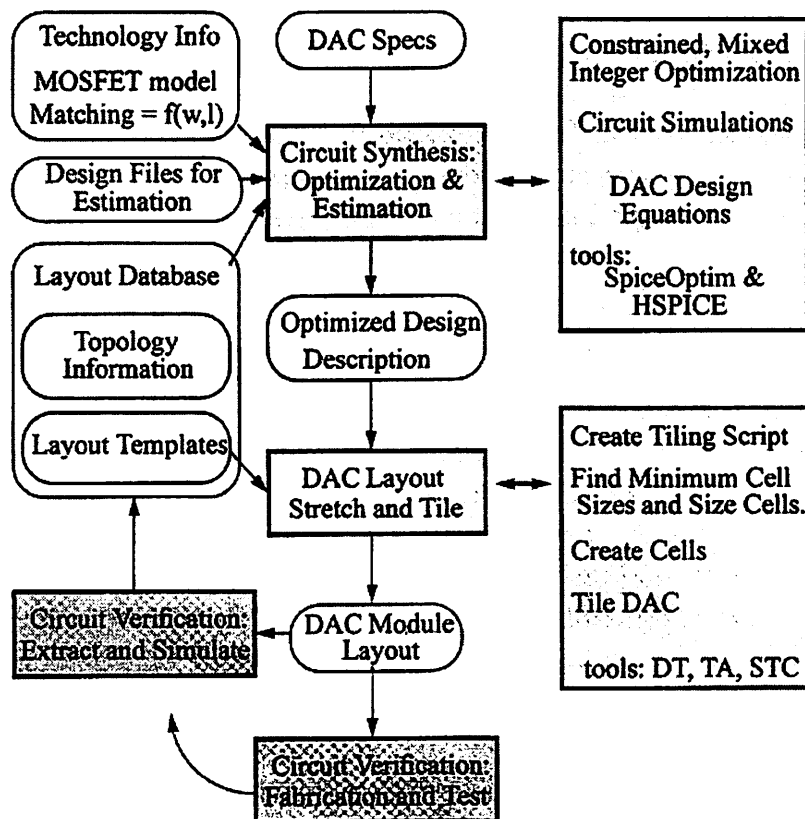


Figure 3.7. Design methodology for automatically generating a digital-to-analog converter.

student working with Professors Gray and Sangiovanni-Vincentelli came up with the module generation methodology shown in Figure 3.7.<sup>13</sup> The generator takes information about the fabrication process, topology information, DAC design equations, and MOS models and performs circuit synthesis. This design is then automatically laid out and simulated, and the results are fed back into the synthesizer. This cycle is repeated many times until the design is fully optimized. While the final DAC synthesized did perform admirably well, according to the author "design development time for synthesis from start to finish is several months – comparable to custom design." [12] The DAC designed in this paper was built in approximately six weeks, suggesting that automatic module generation does not provide a significant design time decrease.

It should be noted that this excellent scaling result only results when the desired accuracy is eight bits or less. For a higher degree of accuracy, it is no longer feasible to use an array of only unit-element weighted current sources, as the number of cells grows exponentially. This requires a DAC composed of both unit-element and binary-weighted current sources, and the optimal level of segmentation must be chosen. This adds substantially more parameters to the high-level design of the DAC and thus does not lend itself to scalable designs. Assuming the simple 100% segmented design, the approximate amount of area required for an  $n$ -bit DAC can be calculated:

$$Area_{DAC} = 2^n A_{CS} + 0.2\sqrt{2^n} A_{CS}$$

$A_{CS}$  represents the area of one current source. The total area is then simply the area from the current source matrix added to the space taken by the decoding circuits and biasing circuitry.



## Chapter 4

# Experiment Prototype and Testing Setup

---

### 4.1 Prototype

A prototype of the digital-to-analog converter was designed and built in an ST Microelectronics 0.25  $\mu\text{m}$  double-poly process. In the test chip, two separate instances of the DAC are included, a standalone one for testing purposes as well as an integrated base station transmitter. The entire test chip is 3 mm by 5 mm, which is significantly more die area than the circuits require; as can be seen in Figure 4.1, the circuits only consume about 15% of the die. The DAC itself is only 250  $\mu\text{m}$  by 300  $\mu\text{m}$ . Note that the die area size requirement is dominated by the pin count, with 32 digital input pins necessary to get the digital input data stream aboard the chip. To meet fabrication requirements, dummy layers were added to the areas that did not contain any circuitry.

The DAC consists of several main components, which can be clearly identified in the

layout in Figure 4.2. On the left is a bypass capacitor block to reduce noise on the current source biasing line. Beneath it is the high-swing biasing circuit. The rectangular block on the top of the DAC is the multiplexor, which feeds into the row and column decoders. These are located to the right of the current source array and directly above the current source array. The final component is the 16 by 16 grid of current source cells, which are the dominant consumers of area.

## 4.2 Test Board Schematic Design

Designing the board to test the chip proved significantly more difficult than expected. There were several hurdles to overcome, including affixing the test chip to the board using chip-on-board technology; translating 3.3 volt input signals from the HP pattern generator down to the 2.5 volt signals required for the chip; separating the various power and ground planes on the board so noisy digital signals would not couple in and hurt the analog components; and figuring out how to generate the necessary reference voltages and currents. The solutions to each of these will be discussed in detail next.

### 4.2.1 Chip-on-board Technology

First, instead of using a conventional package for the chip, it was mounted directly to the printed circuit board using chip-on-board technology. Using this mounting method drastically reduced the bondwire and lead inductance as it allowed the die to be placed much closer to the pads.<sup>14</sup> However, it required a more complex implementation on the PC board as the chip structure had to be created and added, rather than just affixing a simple connector package.

### 4.2.2 3.3 volt to 2.5 volt Voltage Translation

Unfortunately, the high-end HP pattern generator at the BWRC only generates 3.3-volt logic signals – regardless of the pod purchased to use with it. This presented a difficulty as the chip required 2.5-volt signals, and most of the pads did not have overdrive protection. A previous group had run into this difficulty and attempted to resolve it using a resistor-divider network. While this provided the correct voltage at the input of the chip, it also slowed the inputs significantly (down from the high MHz range to the kHz, in fact!). For our testing purposes, this was unacceptable. HP was consulted, and emphatically denied the existence of any 2.5-volt pod solution. At this point some serious web surfing was engaged in an attempt to find a buffer chip fast enough to achieve the necessary 200 MHz voltage translation. Eventually a device was found from Texas Instruments, the SN74LVCC3245A octal bus transceiver that performed the translation at high speeds, and this was used to lower all digital input signals from 3.3-volts to 2.5 volts.

### 4.2.3 Ground and Power Planes

The test chip had a remarkable number of  $V_{DD}$  and GND lines for one device. Besides the standard digital power and ground, there was a baseband digital power, baseband analog power, and RF analog power for a total of four required independent power sources! To ensure that no noise coupling occurred between the various power lines, a multilayer board was built with no more than two well-insulated power planes per layer.

### 4.2.4 Reference Voltage and Current Creation

There were no less than ten independent voltage and current references sources required for the chip. These included bias currents for the DAC, mixer, buffer, and low pass

filter circuits, as well as all the power and ground voltages. Options considered for implementing these sources ranged from stacking multiple power supply boxes in the lab to adding chips on the board to generate each source. The latter won without much difficulty, with voltage regulator (LM317EMPCT) and current source (LM334M) devices from National Semiconductor being utilized to create the necessary voltages and currents.

## 4.3 Test Board Layout

Designing schematics for RF PCB boards is relatively simple. Doing the actual layout is a process rick with pitfalls. In this section, various techniques that were used during the layout of the test board will be discussed. Much of this knowledge is courtesy of Robert Frye, a visiting researcher from Lucent Technologies.

### 4.3.1 Before Layout

Good preparation before you begin layout will save hours of grief. Before you do anything else, order all the necessary parts. This avoids last minute discoveries that a part is on backorder for six weeks, or that there are only four parts available in the United States! Next, decide which board technology is necessary. For this project, a standard 4-layer board was chosen, with FR4 used as the dielectric (FR4 has a dielectric constant of about 4.5). It is tough to know at the beginning how many layers to use, but the deciding factor should definitely be how many signals will be routed. There are generally not a large number of signals on an analog/RF board; four layers should be sufficient. On a digital board, however, consider using more than four layers. At this point it is a good idea to call the board manufacturer and find out specifics of their process. Find out the smallest via drill pitch available, the different dielectric thicknesses, the various board thicknesses, etc. These are all

parameters that are instrumental in determining design decisions.

### 4.3.2 Beginning Layout

When ready to begin actual layout, it is critical to read the tool's "Getting Started" documentation. Generally there are several steps that must be performed before actually putting down components; these include defining the board outline, placing mounting holes, setting the layers correctly, and defining part footprints and padstacks. At this point make sure that there is a footprint that corresponds correctly to every part used in your schematic.

### 4.3.3 Placing Components

Generally PCB tools allow the user to directly import schematics into layout, automatically associating the correct footprint with each part. Some tools also have an autoplacer, which attempts to locate each component in a useful area. This functionality was included in the OrCAD package we used, but since there were not that many components and the quality of placement was far from guaranteed the parts were placed by hand. Some key ideas in this process: first, make sure that you separate any digital components from their analog counterparts. These parts will probably have separate ground and power planes (via split plane technology), and routing the split planes gets difficult fast if the parts are intermingled. Keeping the noisy digital supplies away from the analog parts is also a bonus.

Next, take a careful look at which components the pins on the chip under test connect to. It sounds obvious, but if a supply connects to the top of the chip, make sure to put that voltage regulator on the top of the board! Similarly, any differential signals running

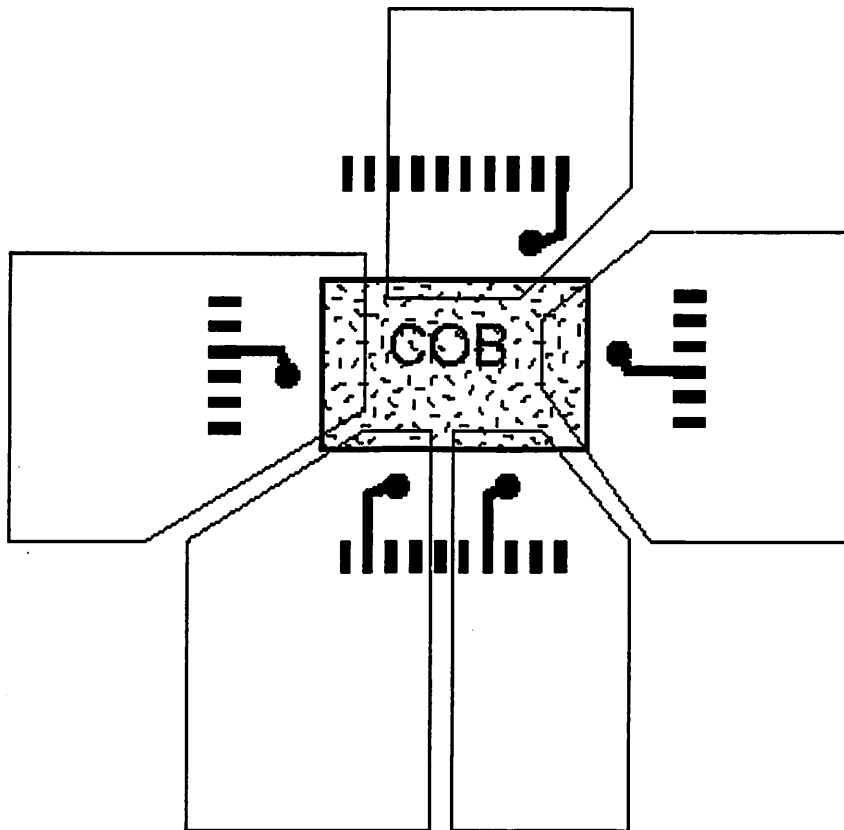


Figure 5.1. Splitting the power plane and connecting to the planes with a via between the COB and its bond pads.

to a balun should be the same length to avoid any phase shift between the two lines. To ensure this is the case, placement of the components in the signal path should be evenly spaced. Finally, all bypass capacitors and terminating resistors should be placed as close to the chip as possible. Proper placement makes routing a breeze, while poor placement can make it a nightmare.

#### 4.3.4 Routing Power and Ground

RF components are very sensitive to noise from power supplies on other areas of the board, so it is important to different power supplies separated. For our chip, we had no less than four power supplies: digital power, analog baseband power, analog digital power, and analog RF power. This was just on-chip power; on the board there were a total of ten supply

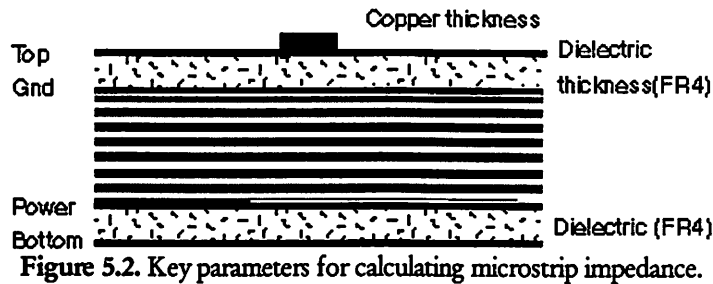


Figure 5.2. Key parameters for calculating microstrip impedance.

voltages and four reference voltages. As many of these supplies were then split up for measurement purposes (i.e. it was necessary to be able to just turn on standalone components or just turn on the entire chain), there were eventually 17 different voltages that needed to be routed! This is a difficult task, especially when the chip being routed to has 4 mil pads and an 8 mil pitch. To throw another bone in the pile, long thin power line traces have a substantial inductance at RF frequencies. Thus the dilemma: trying to route seventeen separate voltages to the chip without using copper lines.

The solution is to split up the power plane layer into many different planes, and then run these planes underneath the chip. The key to the success of this plan is the size of the vias available. This board was originally designed with 20 mil drill hole vias, which are extremely large relative to the size of the chip. However, after consulting our vendor we realized that 8-mil vias were available. With this technology it was possible to place vias between the chip-on-board package and its bond pads. Thus a trace could be run directly from the bond pad to the via, and the via then connected directly to the correct power plane. This system is illustrated in Figure 4.1. This removes two headaches: the problematic inductance from long power lines and the routing difficulty of multiple power supplies. One key note: make sure that the board tool is correctly connecting signals to the plane layers. This should be evident by the appearance of thermal reliefs on the layer the via connects to.

### 4.3.5 Routing Signal Lines

After the trauma of splitting the power layers, routing the signal lines is quite relaxing. The only traces that are extremely critical are the matched impedance lines, generally specified at 50 ohms. To ensure that these are the correct impedance, there are three parameters needed: the dielectric constant (for FR4, this is about 4.5), the thickness in mils of the dielectric, and the thickness of the copper trace. Figure 5.2 illustrates the necessary dimensions. For our board, ½ ounce copper was used, which corresponds to approximately .6 mils. The dielectric thickness was 6 mils. Using a program called AppCad (as well as confirming with the board manufacturer), to generate the desired 50 ohm impedance 10 mil microstrip lines were necessary with 20 mil spacing.

### 4.3.6. Last Steps

When the board is complete, it is useful to perform a few double-checks. Highlight each net to see how cleanly it is wired – often, there is significant more twisting and turning than necessary. Next, turn on only one layer at a time. This can provide a very revealing illustration of inefficiently routed lines, as well as any crossed connections. When satisfied, print the various output files the board manufacturer desires, write a readme file, and send it off!



## Chapter 5

### Conclusions

---

#### 5.1 Conclusion

The goals of this research were severalfold. First, different DAC architectures were analyzed to determine the optimal topology for the given performance specifications. Second, the exact implementation of the chosen architecture was investigated in an effort to use the minimum amount of power while adhering to operational guidelines. After a complete analysis, the block was simulated for functionality verification. Once confirmation of correct operation was achieved, the chip was laid out and integrated with the rest of the transmitter. Currently, the device is being fabricated and a board is being designed to test the transmitter when it comes back from the fab. All testing should be completed by early December, 1999.

## 5.2 Performance

Simulation results from Spice demonstrate that the DAC achieves all required performance specifications in terms of signal-to-noise ratio, speed, and accuracy while consuming only 2.5 mW of power and .075 mm<sup>2</sup>. Signal-to-noise ratio was approximately 60 dB, significantly above the requirement of 48 dB (calculated by 8 bits of accuracy at approximately 6.02 dB/bit). The system achieved the 200 MHz performance benchmark in testing while driving the load capacitance of the following mixer. Finally, there should be no problems with accuracy as the current source transistors were sized in such a way that matching would not be a problem.

However, simulations did show that there was significant glitch energy when the input was fully switched (i.e. the digital input word went from 0 to 255 or vice-versa). For a nanosecond or so after the transition there would be a relatively large current spike. Some of this error was due to simulation effects, such as having an ideal voltage source attached to the output node to represent the following stages. However, even when the model was improved to include a resistor in series with the source, the spike still appeared. One plausible explanation for it is that as the voltages on the gates of differential switches are changing, both switches are on for a finite period of time. This potentially allows more current to flow from power to ground as both sides of the differential pair are conducting. The latch is also designed to keep at least one device on at all times to keep the cascoded current source from going into triode or cutoff. While this serves an important purpose, in future designs the trade-off between glitch energy and linearity should be examined carefully. Functionally, this glitch did not cause a problem as the low-pass filter that follows the DAC in the transmitter chain filtered out the high-frequency content in the analog signal.

However, for a design that is focused on low-power performance, this is a non-negligible energy-wasting factor that should be eliminated.

### 5.3 Scaling into the Future

As long as the accuracy requirement of the DAC remains at 8 bits, this design should scale happily into the sunset. However, rumblings from the system designer of the transmitter are that the next version will require a 10-bit DAC. If this turns out to be the case, some significant redesign could be required. Simply scaling the array up to 1024 current sources will probably require too much power and area, and a more segmented design (i.e. some binary-weighted current sources) will have to be implemented. Nevertheless, there could be some significant design reuse if the unit-element array of this current DAC is used in the next-generation design. Improved control logic and the binary-weighted sources would have to be concatenated with the unit element DAC.

Aside from fundamental changes in the accuracy requirements of the converter, it should scale quite effectively with technology – up to a point. Currently, the main design constraint on the size and power consumption is the amount of current needed to drive the following stage, and is simply a function of the load capacitance on the output. As devices get smaller, however, this load capacitance will shrink extremely quickly, to the point where it could become negligible and not a concern for speed requirements. Instead, the noise and matching requirements and will be the limiting factor in how small the DAC can be and what kind of power it will require. In an analogous fashion, a similar transformation is happening now with digital design: while load capacitance from large gate fanout used to be

---

the dominant speed limiter, with shrinking device sizes the interconnect capacitance is now the main factor. Regardless, in future designs this emerging trend should be examined carefully.

## 5.4 Summary

This project involved designing an integrated CMOS digital-to-analog converter for use in wireless radio transmitter. The performance specifications were 8-bits, 200 MHz while driving at least a 10 picofarad load in the following stage. Several possible topologies were analyzed, and the chosen architecture was optimized for adequate performance with minimum power consumption. After full-custom layout implementation (including full integration with the rest of the transmitter chain), the chip is currently being fabricated in a double-poly, 6-metal layer 0.25  $\mu\text{m}$  process at ST Microelectronics. A test board is being designed to validate the performance when it arrives back from the fab.

## Appendix A: Basics of Transmitter System Design

A quick overview of the wireless system design used in the BWRC WCDMA spread-spectrum system may be useful to certain readers (it certainly was to me!). In the digital backend, the data is modulated with QPSK encoding. This creates a 4 point constellation, as seen in figure A.1. Each point in the constellation becomes one symbol, and thus our system has 2 bits per symbol. This relatively low number of bits per symbol is due to the inherently noisy channels that wireless system encounter; other systems use more complex constellations (such as 64-QAM in cable modems), but these modulation schemes are too susceptible to interference from neighboring channels for use in wireless systems.<sup>15</sup>

The two bits that make up each symbol are divided into two channels, generally labeled in-phase and quadrature. Thus each symbol is divided into two data streams (I and Q), each running at 800 kHz. Performing the simple math, this gives each user a data rate of 1600 bits per second. At this point – still in the digital backend – the bitstream is multiplied by a spreading sequence (Figure A.2). The spreading sequence is running at 12.5 MHz, which is known as the chip rate of the system. Each user has a unique spreading code, and this allows the receiver to decode the incoming signal and receive the correct data. Using a spreading sequence is the main idea behind all CDMA and has several nice properties. First,

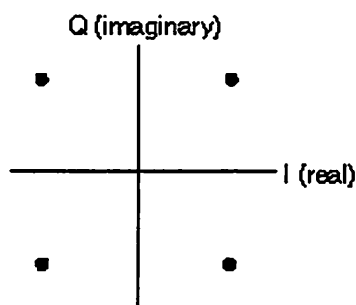


Figure A.1. 4 Point Constellation.

the power spectral density of the signal is spread out across a much wider frequency range, making it more resistant to unexpected noise sources in the channel. This is illustrated in Figure A.3. Second, the spreading sequences tend to be

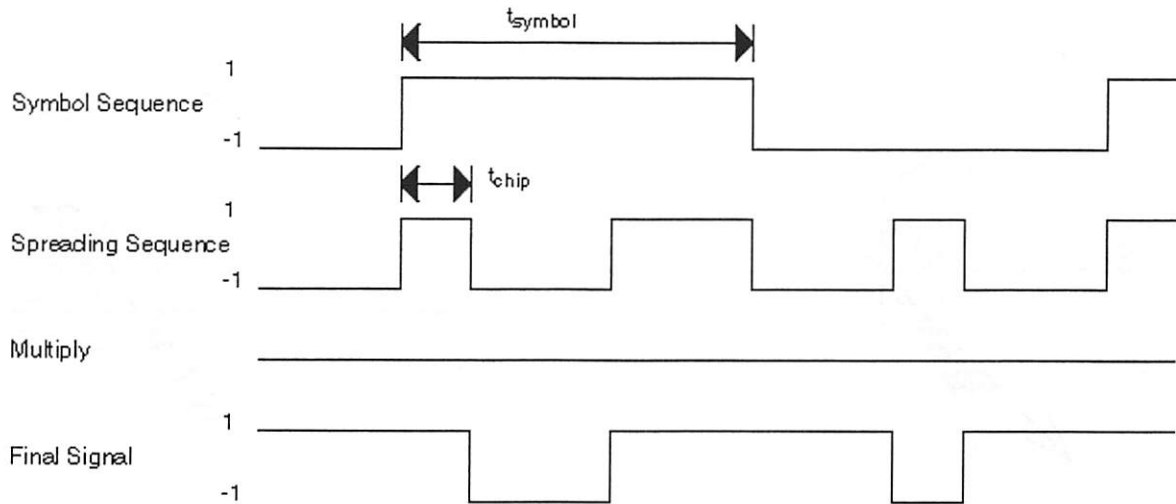


Figure A.2. Spreading of initial symbol sequence.

orthogonal to other spreading sequences, which also provides significant noise reduction. At this point, the digital processing is essentially complete, and the backend outputs two digital bitstreams at 12.5 MHz apiece.

At this point, the analog frontend comes into play. The two digital bitstreams first enter the D/A converters and are converted into an analog representation in the current domain. System simulations with Matlab determined the specifications of the converter; for this WCDMA designed here, 8 bits were sufficient. This analog signal – now running at 200



Figure A.3. Power spectral density plot of symbol and chip symbols.

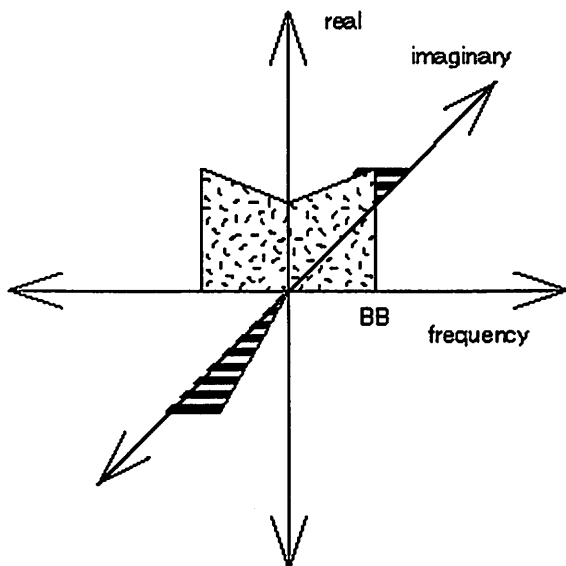


Figure A.4a. FFT of baseband signal.

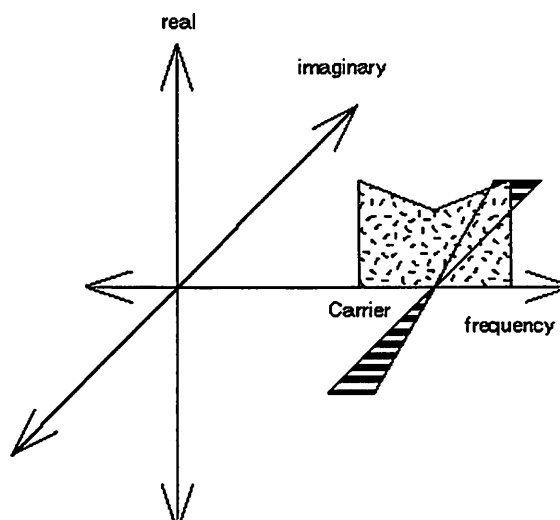


Figure A.4b. Mixed with cosine.

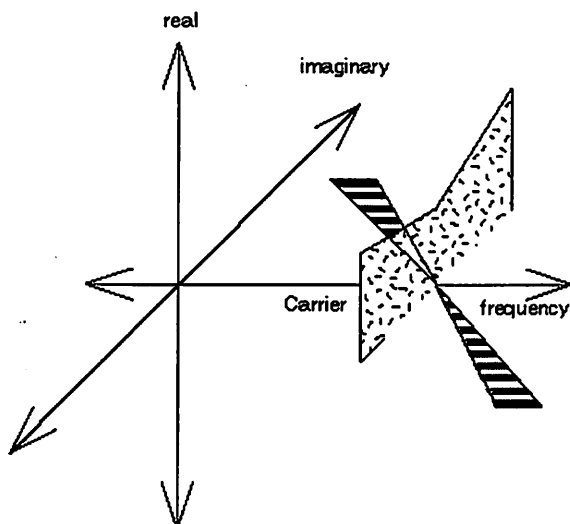


Figure A.4c. Mixed with sin.

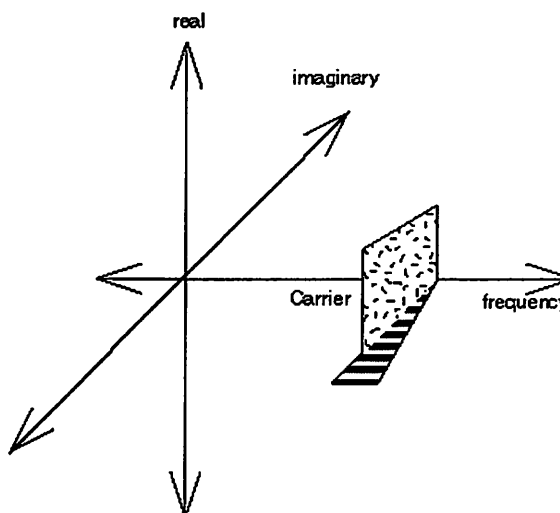


Figure A.4d. I &amp; Q channels mixed with sin and cosine and then added.

MHz – is then filtered by a single-pole RC circuit. A simple single-pole filter is sufficient due to the speed of the DAC.

The filtered output is then mixed up to the carrier frequency, 2.0 GHz for this specific implementation. The I channel is mixed with a sin wave, while the Q channel is mixed with a cosine wave. This results in a non-symmetric frequency spectrum centered around the carrier frequency, allowing use of the full bandwidth allocated to the system. To

---

explain this phenomenon, examine Figure A.4. In A.4a, the fourier transform of random I and Q signals is shown, at the baseband level. When the signals are upconverted with cosine, they are both reproduced at the carrier frequency, as in A.4b. When they are upconverted with sin, they are again reproduced at the carrier frequency, but shifted 90 degrees (A.4c). As the signals are both symmetric, if one of these options is chosen, half the bandwidth is being wasted with a redundant signal. If, however, the I is mixed with cosine and Q is mixed with sin you get an asymmetric output as shown in A.4d. The entire used spectrum contains unique information.

After mixing, the two channels are added and this single signal is amplified. After its long journey, the data is finally free to leave the antenna and head into the wild blue yonder (assuming, of course, that "wild blue yonder" refers to about 10 feet or less of BWRC lab space). And that, in three pages, is how a wideband CDMA spread spectrum transmitter works!



## Appendix B: Binary to Thermometer Decoding

Here is a schematic of the circuit implemented to perform the binary-to-thermometer decoding. The logic used was inspired by a 3-bit decoder implementation by Miki et al.<sup>16</sup> Although fully functional, the design has several levels of gates that could potentially be reduced beyond the current number. Furthermore, extending the logic to a larger number of bits – say 5 – would be very difficult and possibly require a complete redesign. Razavi mentions that it is possible to create a “modular form applicable to higher word lengths,” but does not give any details. [3] This will be something to explore in the next revision of the DAC! However, it may not be necessary because any expansion in the number of bits would probably be handled by an increase in the amount of segmentation (i.e. some binary bits would be added rather than 256 more current sources).

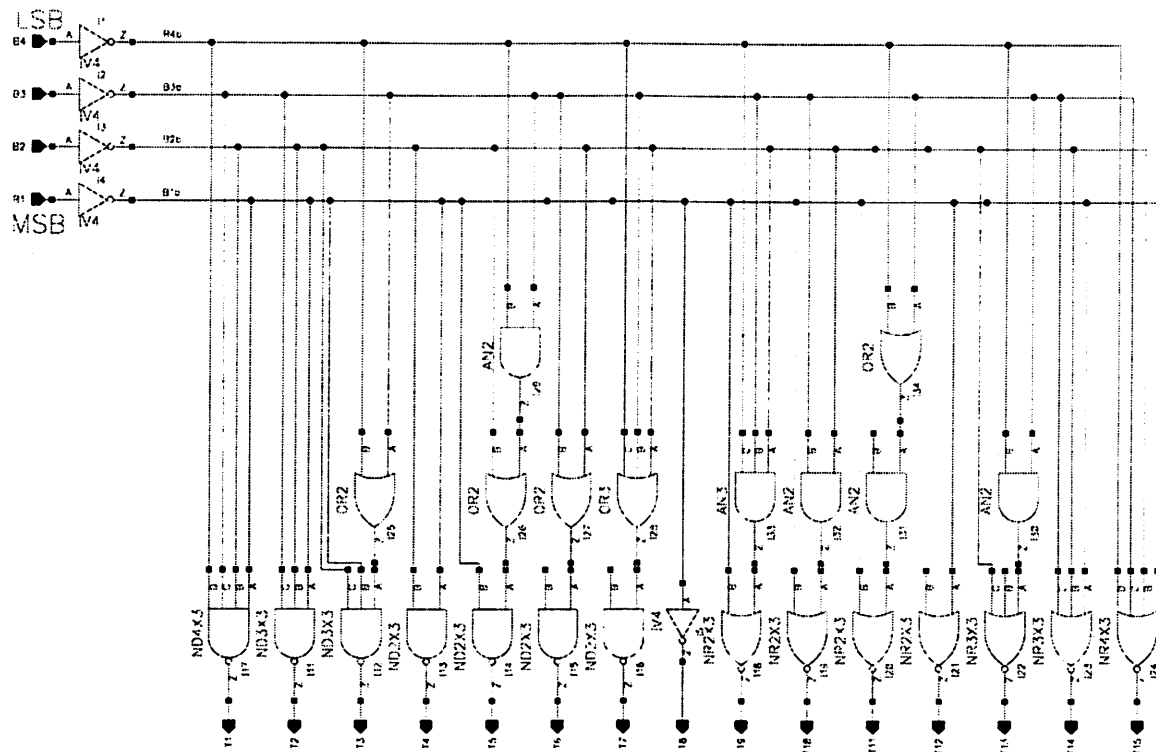


Figure B. Binary-to-thermometer decoding circuit used.

## Appendix C: Schematics and Layouts

This section provides illustrations of the various blocks created for the design. As most of the design decisions have been explained earlier in the paper, the important aspects of each block will be highlighted but not described in great detail.

### C.1. Digital Components

There are two entirely digital implementations in the design, the multiplexor and the binary-to-thermometer decoder. The binary-to-thermometer decoder was described in Appendix B and will not be described again. The multiplexor is built from standard cell components, so the layout will not be shown (see Figure C.1 for the schematic). It simply consists of a chain of eight 4-to-1 muxes. The 200 MHz clock signal is fed into a pair of flip-flops, which act as 2 bit counter, counting from zero to three every four clocks. This cycles through each individual multiplexor input at an effective clock rate of 200 MHz – even though the inputs to the muxes are only changing at 50 MHz. Simple but devastatingly effective. Of course, this still requires the clock to come on-board at 200 MHz. The HP pattern generator can provide a 200 Mhz signal, but not a perfectly square wave. With a little luck the signal it generates will be sufficient for the clock signal.

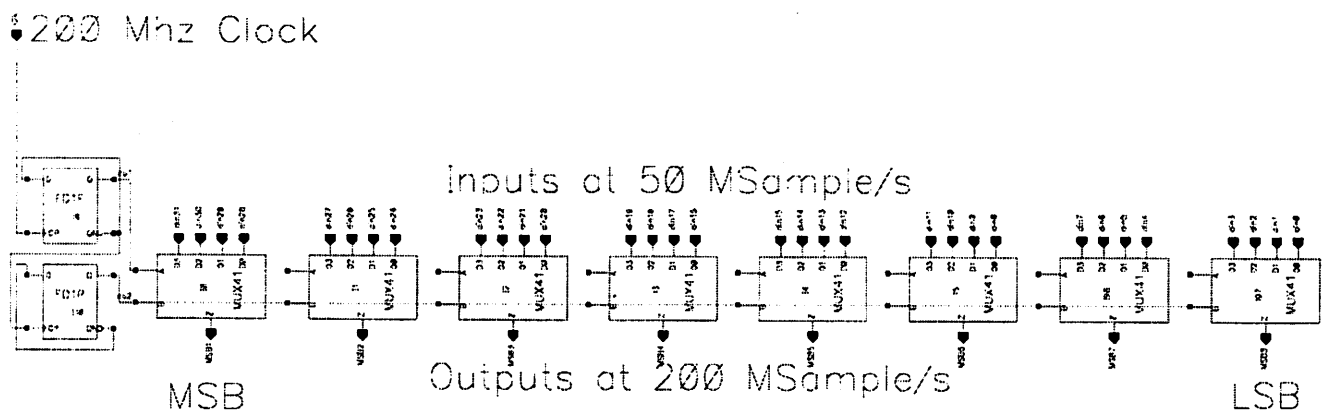


Figure C.1. 32-to-8 bit multiplexor schematic.

## C.2. Analog Components

### C.2.1. Current Source Cell

The current source cell is the critical part of the design due to the fact that it is repeated  $2^N$  times in a 100% segmented architecture. As mentioned earlier, the important design parameter is the current generated by each current source, selected to be 10 mA for this design. The layout is shown in Figure C.2. When first design, standard cells were used

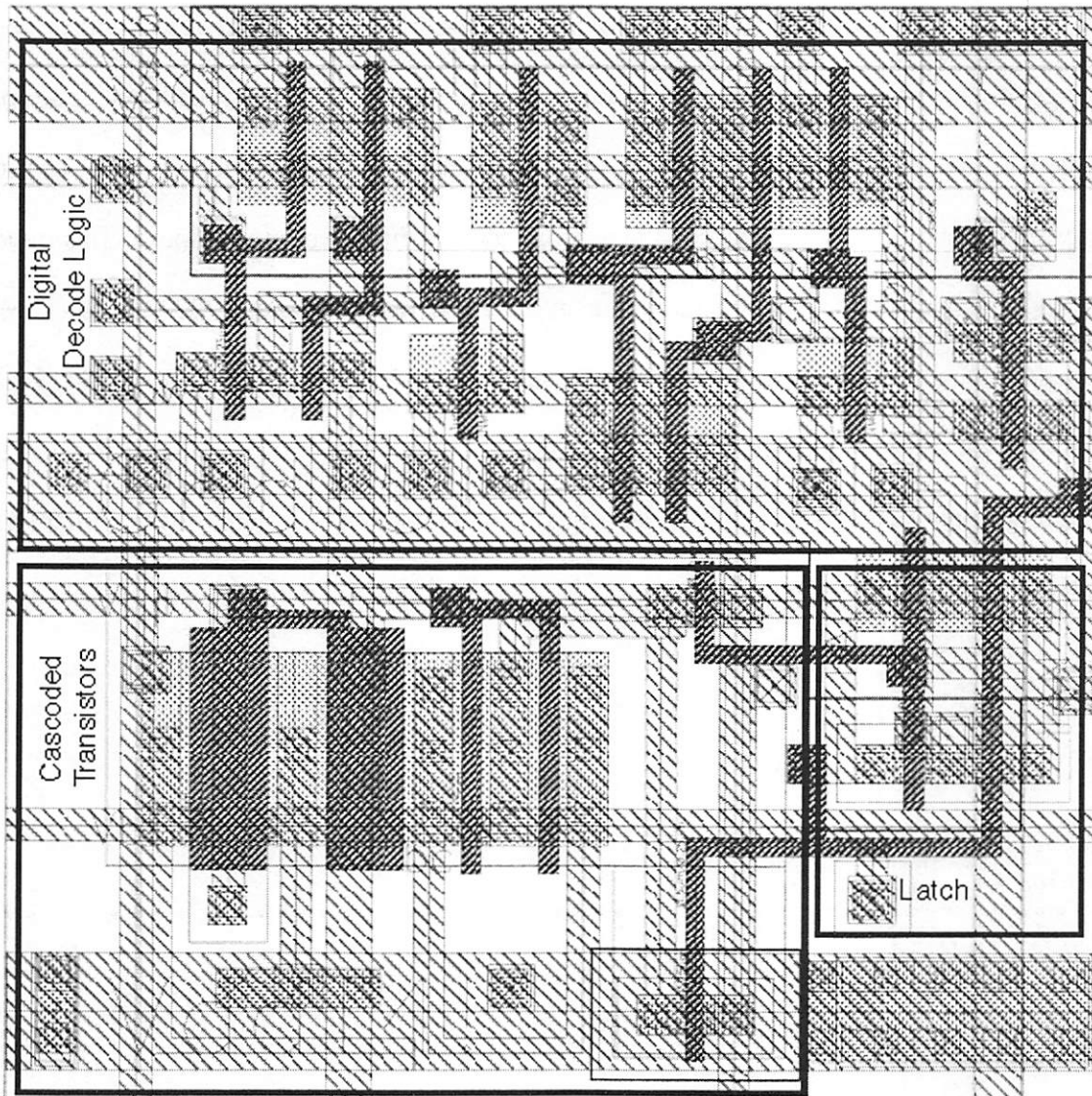


Figure C.2. Current source cell layout.

for the digital decode logic, but this plan was scrapped when it was apparent that this would be quite wasteful in terms of area. Thus the cell is a full custom design. There are several interesting aspects. First, the goal was to keep the block perfectly square to minimize area consumption. Second, global lines – such as Col, Row<sub>N</sub>, Row<sub>N-1</sub>, the cascode bias voltage, Clk, etc. – had to enter the cell and exit the cell at the same horizontal or vertical position so that the cells could be abutted together in array fashion. This saved annoying design time when connecting up the 256 cells! Finally, antenna protection diodes were placed wherever there was room to increase manufacturability – though this isn't a tremendously important issue for a test chip! The schematic for the current source is shown on the next page in Figure C.3.

### C.2.2. Cascode Biasing Circuit

The cascode biasing circuit used is a standard high-swing biasing circuit, as anyone who has ever suffered through Gray and Meyer is fully aware! The main design challenge is in getting the circuit to function effectively at 2.5 volts – this does not leave much headroom, risking dropping the current source out of saturation. Essentially, the circuit takes a standard cascode biasing circuit and adds a voltage-level-shifting device in series with the gate of the cascaded transistor (in this circuit, the cascode transistor is M38/M54; the device has been fingered to provide better matching to the current sources it is biasing). The voltage-level-shifting device is M41-M46 (again, representing one fingered transistor), which acts as a source-follower circuit. The resulting voltage on the gate of the cascode transistor is  $V_T + 2\Delta V$  rather than the standard  $2(V_T + \Delta V)$ , increasing the output swing by a very beneficial  $V_T$ . The schematic and layout are shown in Figures C.4 and C.5, respectively; as there is only one bias circuit in the design, no time was spent to minimize the layout area.

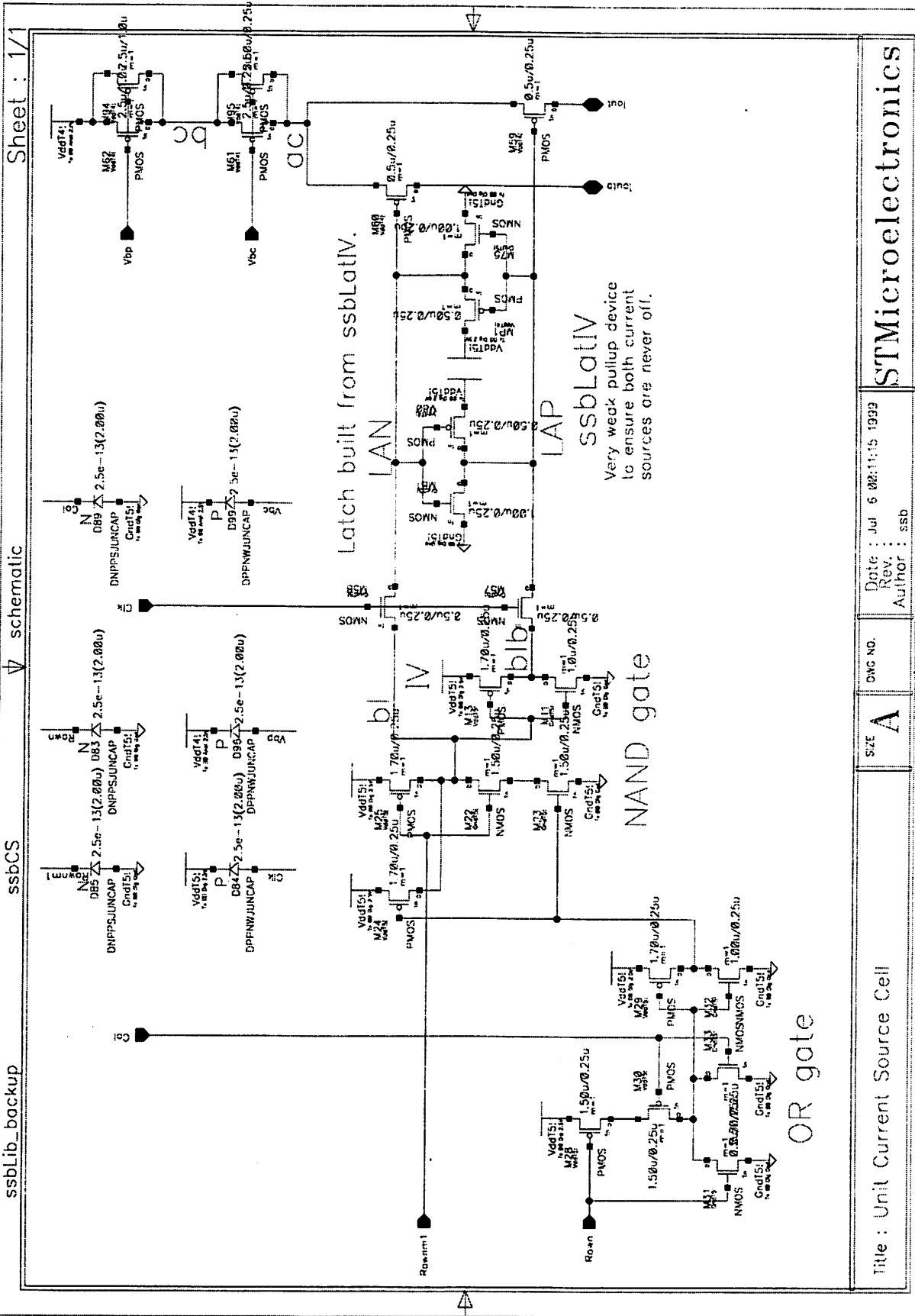


Figure C.3. Current source cell schematic.

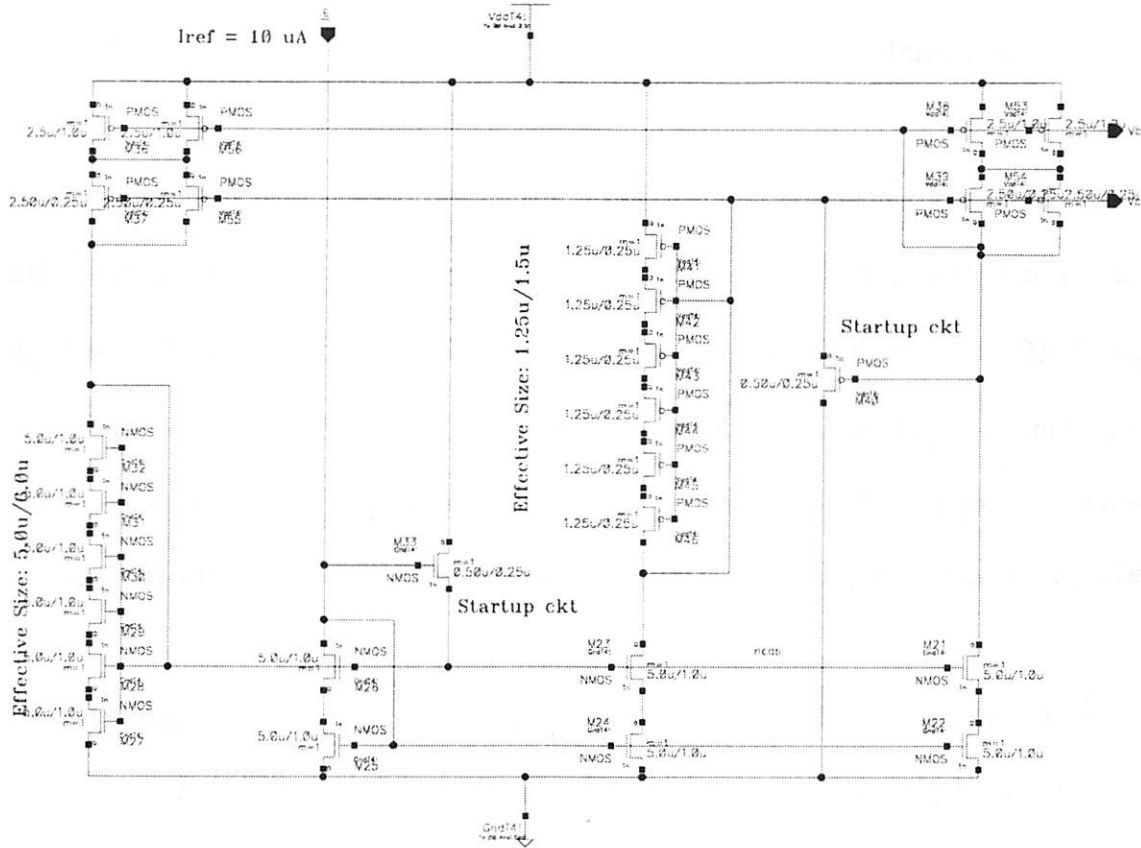


Figure C.4. Cascode biasing circuit schematic.

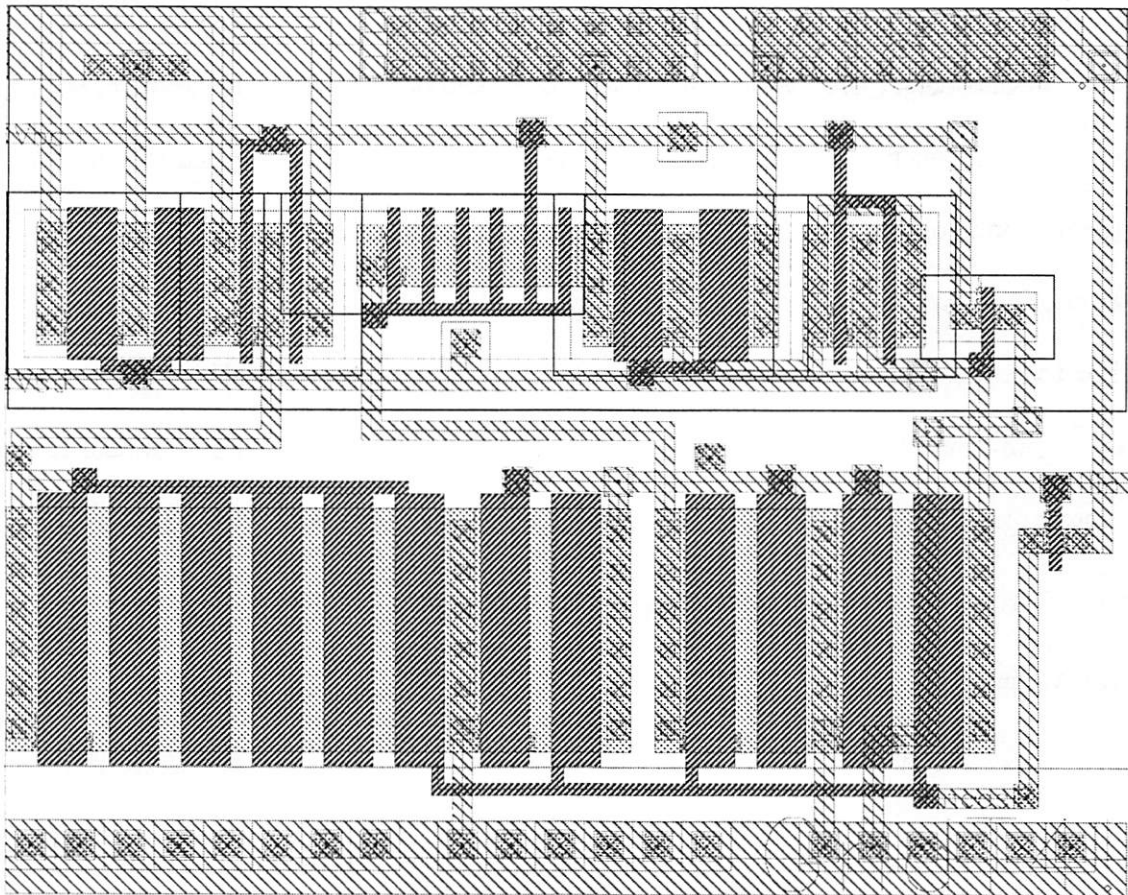


Figure C.5. Cascode biasing circuit layout.

### C.2.3. Complete DAC Layout

Figure C.6 contains the finished layout. Isn't she a beauty? That all depends on how well it works, of course! 0.25  $\mu\text{m}$  technology results in very impressive area numbers; the entire DAC is only 250 by 300  $\mu\text{m}$ , resulting in an area of only 0.075 square millimeters! This resulted in a tremendous amount of unused area on our 2 mm by 5 mm testchip. Of course, the area is necessary to have enough pins to get all the digital input signals in. Once the back-end digital design is complete, the entire transmitter will probably fit in a 4 mm<sup>2</sup> package.

The floorplan of the complete digital-to-analog converter is as follows. The vast majority of the layout area is dominated by the current source array, which is the 16 by 16 grid. On the far left of the design is the 32-to-8 bit multiplexor. The outputs of the multiplexor feed row and column decoders, implemented as two identical binary-to-thermometer decoders. Here the benefits of spending extra design time to make the current source cell square again come into play, as very little excess wiring was needed to connect both the row and column decoders to the current source array. At the bottom right corner is the cascode bias circuit, and the gigantic rectangle on the bottom of the design is a bypass capacitor to reduce the noise on the current source voltage biasing line. The capacitor is probably significantly larger than necessary, but as the design needed to be rectangular for easier integration into the full-chip floorplan, all the extra space was used for this bypass capacitor. At the top right are the two output lines carrying  $I_{OUT}$  and  $\overline{I_{OUT}}$ . Quite a bit of work for two little signals!

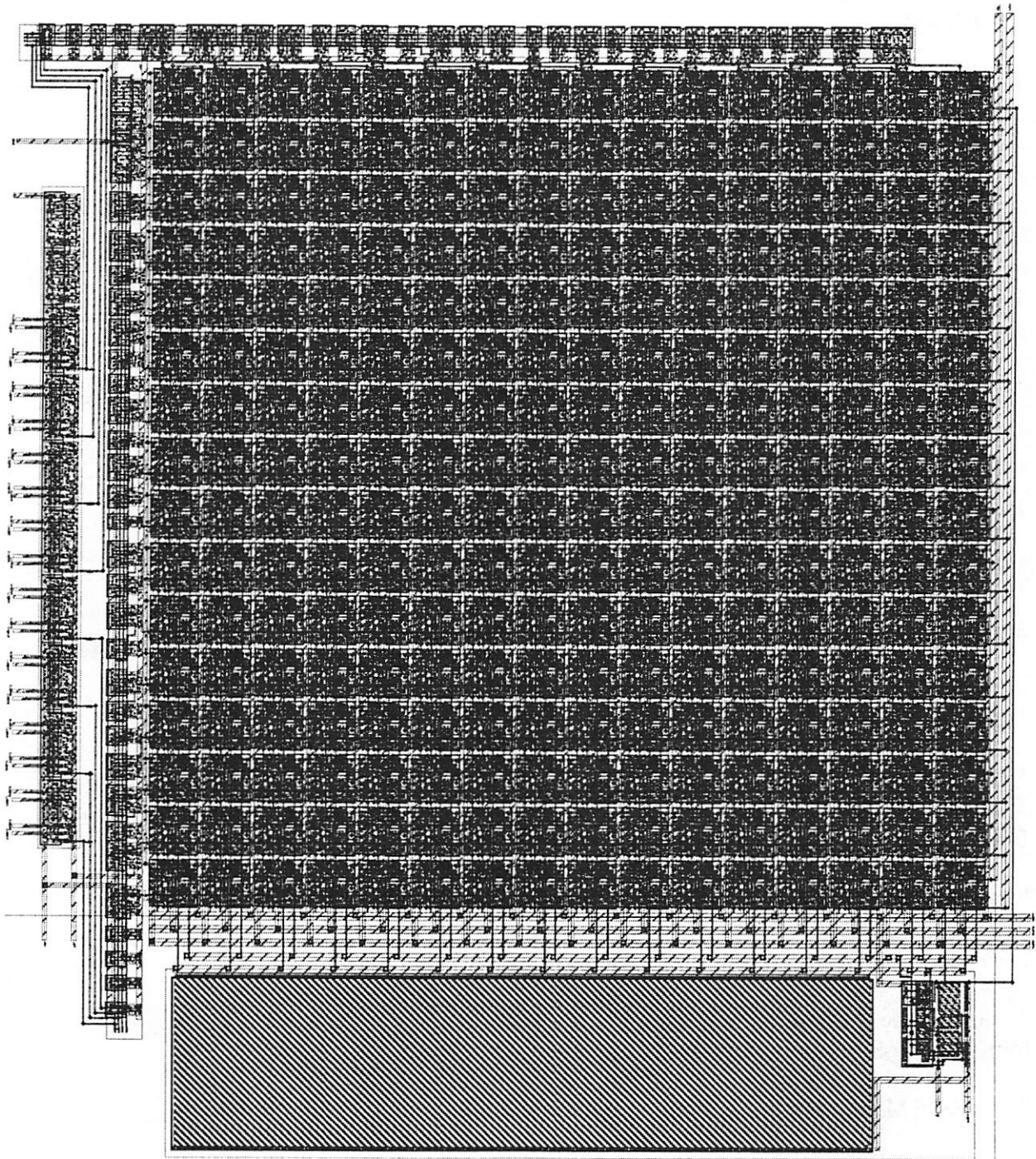


Figure C.6. The complete DAC layout.



## References

- <sup>1</sup> Johan Vanderhaegen designed the basestation system.
- <sup>2</sup> Craig Teuscher, Dennis Yee, Ning Zhang, Robert Brodersen, "Design of a Wideband Spread Spectrum Radio Using Adaptive Multiuser Detection." ISCAS, Monterrey, CA, 1996.
- <sup>3</sup> Behzad Razavi, "Principles of Data Conversion System Design." Copyright 1995, IEEE Press, Piscataway, NJ. Much of the derivations presented in the architecture overviews are derived from Razavi's book.
- <sup>4</sup> J.L. McCreary, "Matching Properties, and Voltage and Temperature Dependence of MOS Capacitors," *IEEE Journal of Solid-State Circuits*, vol. SC-16, pp. 608-616, December 1981.
- <sup>5</sup> Chi-Hung Lin and Klaas Bult, "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm<sup>2</sup>," *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 12, December 1998.
- <sup>6</sup> Yasuyuki Nakamura, Takahiro Miki, Atsushi Maeda, Harufusa Kondoh, and Nobuharu Yazawa, "A 10-b 70-MS/s CMOS D/A Converter," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 4, April 1991.
- <sup>7</sup> 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, No. 5, October 1989.
- <sup>8</sup> Klaas Bult told me this information in an email. Ian O'Donnell had raised the point in a design review, and I did not know the answer so I emailed Mr. Bult. He is Chief Scientist at Broadcom Corporation and a professor at UCLA.
- <sup>9</sup> Jose Bastos, Augusto Marques, Michel Steyaert, and Willy Sansen, "A 12-Bit Intrinsic Accuracy High-Speed CMOS DAC," *IEEE Journal of Solid State Circuits*, Vol. 33, No. 12, December 1998.
- <sup>10</sup> Paul R Gray and Robert G. Meyer, "Analysis and Design of Analog Integrated Circuits," John Wiley & Sons, 1993. Page 727.
- <sup>11</sup> Gray and Meyer, *Ibid*, page 283.
- <sup>12</sup> Adam Eldredge was an excellent source of knowledge and inspiration for the current source layout.
- <sup>13</sup> Robert R. Neff, Paul R. Gray, and Alberto Sangiovanni-Vincentelli, "A Module Generator for High-Speed CMOS Current Output Digital/Analog Converters," *IEEE Journal of Solid State Circuits*, Vol.31, No.3, (IEEE 1995 Custom Integrated Circuits Conference, Santa Clara, CA, USA, 1-4 May 1995.) IEEE, March 1996. p.448-51.

---

<sup>14</sup> George Chien, "High Speed, Low Power, Low Voltage Pipelined Analog-to-Digital Converter," Memorandum No. UCB/ERL M96/27 (UCB Masters Thesis), May 15, 1996, page 33.

<sup>15</sup> From CNET's glossary at

<http://www.cnet.com/Resources/Info/Glossary/Terms/cdma.html>.

<sup>16</sup> Takahiro Miki, Yasuyuki Nakamura, et al, "An 80-Mhz 8-bit CMOS D/A Converter," *IEEE Journal of Solid State Circuits*, Vol. 21, No. 6, December 1986.