A 9 BIT 10 MHZ A/D MACROCELL

by

Andrew J. Burstein

Memorandum No. UCB/ERL M87/84

20 November 1987

COVER PAGE

A 9 BIT 10 MHZ A/D MACROCELL

by

Andrew J. Burstein

Memorandum No. UCB/ERL M87/84

20 November 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

A 9 BIT 10 MHZ A/D MACROCELL

by

Andrew J. Burstein

Memorandum No. UCB/ERL M87/84

20 November 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Research supported by DARPA Contract N00039-87-C-0182.

Contents

Table of Contents 1

List of Figures 3

1 Introduction 4

2 Architecture 6

3 Circuit and Layout Description 10

3.1 Opamp 10

3.2 Comparator 12

3.3 Digital Control 13

3.4 Error Correction 14

3.5 Clock 14

3.6 Top Level 15

4 Ext2spice: A New Tool 33

4.1 Description 33

4.2 Algorithm 37

4.3 Speed 39

5 Simulations 41

6 A Guide to Testing 49

6.1 A/D Biasing and I/O 49

6.2 Test Section Biasing and I/O 51

6.3 Future Work 52

7 Conclusion 57

Bibliography 58

A File Index 59

List of Figures

2.1 A Single Stage of the Four Stage A/D 8
2.2 Error Correction Architecture 9

3.1 Opamp Core Circuitry 17
3.2 Opamp Section Circuitry 18
3.3 Opamp Section Layout 19
3.4 Comparator Core Circuitry 20
3.5 Comparator Section Circuitry 21
3.6 Comparator Section Layout 22
3.7 Comparator Core Layout 23
3.8 Digital Control Circuitry 24
3.9 Latch and Clocked Gate Circuits 25
3.10 Error Correction Logic 26
3.11 Clock Generator Circuitry 27
3.12 Signal Flow Between Stages 28
3.13 Overall Clocking and Biasing 29
3.14 Timing of Interstage Signals 30
3.15 Floorplan of a Single Stage 31
3.16 Floorplan of the A/D 32

4.1 A Typical (Corrected) Extract Section of Magic Technology File . . 35
4.2 Extract Section of Magic Technology File for Ext2spice 36
4.3 Comparator Core Output from Ext2spice 38

5.1 Resistor Cell as converted by Ext2spice 44
5.2 Modified Resistor Cell 45
5.3 A Transition of Nonoverlapping Clocks 46
5.4 Voltages at *pup+* and *pup-* in Comparator 47
5.5 0.9% Overshoot with 1.5pF Load 48

6.1 Chip Layout 55
6.2 Bias Circuits in A/D 56

Chapter 1

Introduction

This paper describes the layout of a 10MHz nine bit analog to digital converter implemented in a 1.6μm scalable cmos process. The architecture for this converter was developed by Steve Lewis [1] [2]. Previously, it has been implemented in a 3μm process using +/- 5 V supplies, and a 1.25μm process using 0 V to 5 V supplies. Both of the processes used a special capacitor layer. The third implementation, described herein, uses a standard digital cmos process - containing no special capacitor layers - and uses 0 V to 5 V supplies. It is intended to be used as a standard cell in the Lager design system, allowing a video rate A/D to be placed on the same chip with digital image processing circuitry.

This paper contains five emphases. First, it provides an overview of the architecture of the A/D and the algorithms it uses.

Second it will describe the circuits used and their layout on silicon, paying particular attention to the differences between this layout and the previous ones. Changes in the relative size of the opamp circuitry, vis a vis the comparator circuitry, made some drastic changes in floorplanning quite attractive.

Next is a description of a new tool written to improve the process of extracting circuit information from the layout editor, magic, into the form used by spice, the circuit simulation program. Unfortunately, the present system of extraction programs commonly used contains some deficiencies. Most notably, source and gate diffusion regions are approximated by lumped capacitances, instead of utilizing the more accurate area and perimeter of diffusion parameters available in spice. To fill this gap, a new program, ext2spice, was developed. Ext2spice accurately passes on the areas and perimeters of diffusion regions in the layout to spice. It also contains several other features, to make it easier for the designer to organize and understand the extracted version of his circuit.

This is followed by a description of the spice simulations used to develop and confirm the design.

Finally, there is a guide to testing the A/D circuit. It includes detailed information on biasing and I/O.

Chapter 2

Architecture

This nine bit A/D consists of four pipelined stages. Each stage outputs three bits, but one bit from each of the last three stages is redundant, and is only used for error correction. Figure 2.1 shows the architecture for the stages. The input is sampled and held with a gain of four. This voltage goes to a three bit A/D, whose output controls a three bit DAC. The DAC output is subtracted from the held input voltage and sent to the next stage. Since the DAC output should match the input to three bits, the residual should be less than one eighth of the full range of the stage. Therefore, each stage could actually have an input gain of eight and still keep all signal magnitudes within the same limits as in the first stage. By having a gain of only four, half of the range of each stage past the first stage does not appear to be used. This, in fact, is why these stages only produce two bits each.

The full range of a stage would be used if the A/D in the previous stage made a decision error in the least significant bit. In that case, the DAC output is only accurate to two bits, and the residual could be up to one quarter of the full range of the stage. But since the input gain of the each stage is only four, there will be no overflow. The stage could still make a decision based on the residual, and then use its redundant bit to correct the LSB of the previous stage.

A one bit decision error can be caused by input offset voltages in the comparators in the three bit A/D. Since the A/D, with digital error correction, can tolerate a one level decision error, comparator offset voltages are not critical.

Therefore, a simple comparator circuit - without input offset cancellation - is accecptable.

Because of the pipelined architecture, the bits from each stage must be delayed to resynchronize them before being digitally corrected and then output. Figure 2.2 shows how this is done.

While this A/D has the architecture of a nine bit converter, the application to which it will be applied (image processing) only requires eight bits resolution. It simply turned out to be easier to design a nine bit A/D. First of all, it is desirable for each stage in the pipeline to share as much of the same layout as possible, to reduce the amount of design work and to reduce the chances of malfunctions. This is best accomplished by making all of the stages the same size. Using i stages of n bits each yields $N = n + (n-1)^{i-1}$ total bits output. There are only two solutions which yield $N = 8$: $n = 8$, $i = 1$; and $n = 2$, $i = 7$. The first solution is simply a single flash A/D, which would take up too much area with 255 comparators. The second is seven two bit stages. This would turn out to be larger than four stages of three bits because of the need to use a total of seven opamps. Furthermore, using the same architecture as the previous implementations minimized the amount of redesign necessary.

Figure 2.1: A Single Stage of the Four Stage A/D

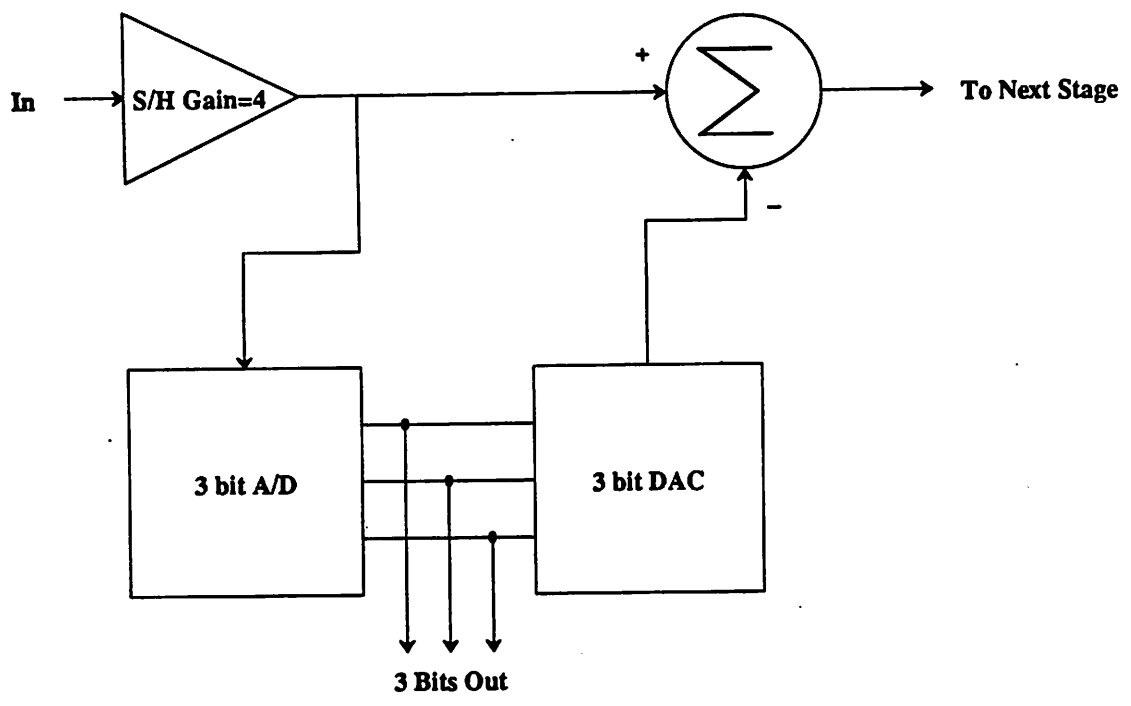
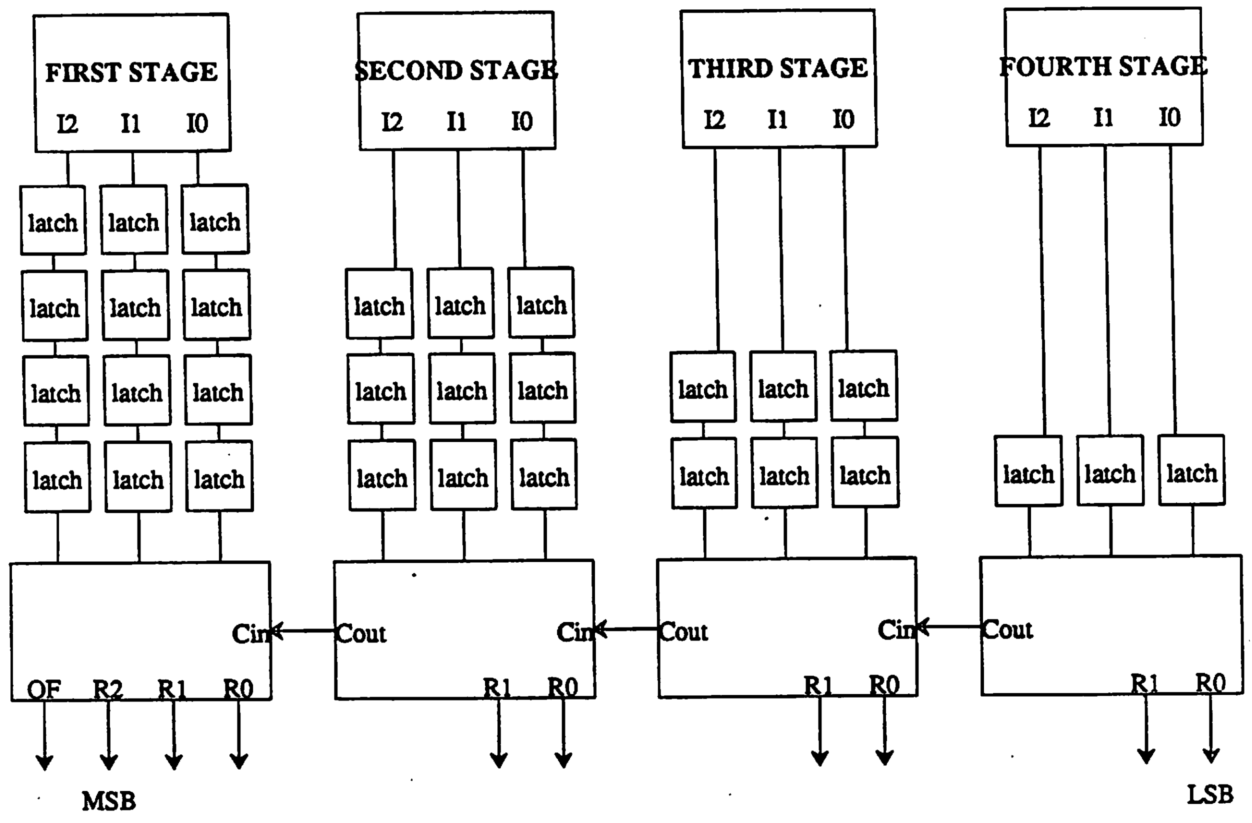


Figure 2.2: Error Correction Architecture



Chapter 3

Circuit and Layout Description

The layout is divided into four major stages, one for each pipelined stage of the A/D. Within each stage resides three sections: opamp, comparators, and digital control. In addition to the four stages is the error correction logic. The clock generator is layed out in a cell independent of the rest of the A/D.

3.1 Opamp

Function

The opamp performs the input sample and hold with a gain of four. In the second, third, and fourth stages the opamp also subtracts the previous stage's DAC output from its S/H output. Thus the sigma function shown in Figure 2.1 actually takes place in the next stage.

Circuitry

The opamp is a fully differential class A/B design, based on a design by Castello and Gray [3]. The layout of the opamp core is based on a layout made by Max Hauser [4]. This layout was used as a starting point in order to save development time. The opamp was modified to increase its speed as determined by spice simulations.

See Figure 3.1 for the schematics of the opamp core, as well as the bias circuitry included in the core. Since the opamp is completely symmetric, only one half of the circuitry is shown. The complete circuit contains two copies of the half circuit, with the inputs cross connected and the outputs cross connected. The major difference between this circuit and the opamp used in the 10 V implementation is the lack of buffers between the input devices and the *pmir* and *nmir* nodes. These buffers are impractical with the lower power supply range.

The complete sample and hold, and subtraction, circuit is shown in Figure 3.2. The feedback capacitors, C, are approximately 0.25pF, and the common mode feedback capacitors are 0.15pF. The circuit is the same for all stages, except for the first stage, which has a different input section. The first stage does not have to subtract a DAC voltage since it has no previous stage. The first stage actually uses the same transistors as the other stages, they are merely connected differently.

The opamp completes all of its operations over the course of one cycle of a two phase nonoverlapping clock. During phiB, the input charges the 4C input capacitors, and other nodes are grounded. The input nodes of the opamp are grounded during a special clock phase, phiBprime. PhiBprime turns on at the same time as phiB, but turns off slightly beforehand. This ensures that the charges injected into the input capacitors are equal. During phiA, the feedback loops are closed and the circuit produces an output. Note that there are two feedback paths: one from the output to the input, the other from the output to the common mode feedback. In the first stage, the input voltages are referenced to each other; in the following stages, to the DAC inputs.

Since the output of one opamp is the input of the following opamp, the clocking of neighboring stages must be opposite. That is, when going between adjacent stages, all **A** phases become **B** phases and *vice versa*.

Layout

The opamp core is layed out in three instances of two cells. In the center is the bias circuitry. On either side is an instance of an opamp half circuit. Thus, all of the differential circuitry is bilaterally symmetric to improve matching.

The rest of the opamp circuitry - the capacitors and switches - also use bilateral symmetry. The switches are located on the side of the opamp section opposite the opamp core, on the bottom of Figure 3.3. In between are the capacitors. The central capacitors are the feedback capacitors; the four on either side are used in parallel as the input capacitors. Thus, the capacitors within each feedback path are common centroid. Each of these capacitors is surrounded by guard rings and all are in a common well. The capacitors have metal1 as one plate, and polysilicon and metal2 as the other plate.

3.2 Comparator

Function

The comparator section performs the analog functions of the three bit A/D and DAC. It compares the output voltage from the sample and hold to voltages on a resistor string, *i.e.* it is a flash A/D. This same resistor string is the source for the D/A voltages.

Circuitry

The comparator circuit, shown in Figure 3.4, is similar to that used by the other implementation with 5 V to 0 V supplies. Transistors M1 - M6 form a differential amplifier, which is in operation while the opamp is producing an output. As soon as the opamp has finished, the signal *dpon* goes low and *latchon* goes high. This turns off the current through M1 - M4 and diverts it through the latch, M9 and M10. When latching, transistors M3 and M4 are turned off, by lowering *cgbias*, to prevent a short circuit path through M3 M1 M2 M4. Since switching M3 and M4 can cause charge injection into the nodes *pup+* and *pup-*, it is important to have the same parasitic capacitances at these two nodes. Therefore, the two comparator outputs are buffered by inverters placed within the comparator cells. Using a simple inverter circuit as a buffer would waste power, since the input would be near 2.5 V before it is latched, causing both devices to conduct. Instead, the clocked buffers M12 - M17 are used. This allows a high degree of matching between the parasitic capacitances of the two critical nodes, with little cost in power or speed. The buffers are the major difference between this and the previous 5 V latch implementation. The 10 V latch had additional input follower devices, which were impractical with a smaller power supply range.

The circuitry of the entire comparator section is detailed in Figure 3.5. During phiB - while the opamp is loading its input capacitors - the comparators are loading their own input capacitors from the resistor string. Like the opamp, the comparators have the grounding switches at their input nodes turn off first. During this phase, the comparator outputs are latched and the DAC outputs are turned on. The output from the opamp is transmitted to the comparator inputs through their input capacitors during phi A.

Layout

The comparator section layout is shown in Figure 3.6. Each comparator consists of a stack of cells containing from top to bottom: comparator core, capacitors, resistor string segment, and switches. These stacks are placed side by side to create an array of seven comparators. Clocks and bias lines run horizontally; signals, vertically. The layout of the comparator core is detailed in Figure 3.7. Note that the signal path is symmetric to improve matching.

3.3 Digital Control

Function

The digital control section in each stage decodes the thermometer code output from the flash converter. Its output controls three functions. First, it produces the signals A - G which control the DAC output (see Figure 3.5). I and J control the criss cross input of the opamp (Figure 3.2). Finally, it creates the three bits which encode the decision into binary. These three bits are sent to the pipeline to the digital error correction.

Circuitry

See Figure 3.8. The clocked gates operate normally while the comparator outputs are latched. During the other clock phase, their outputs are high. This is helpful because the signals to the DAC must turn off all of the DAC switches during this phase. (The clocked gate circuits are detailed in Figure 3.9). Also, the values of **I** and **J** must be opposite at all times, or the inputs to the next stage will be shorted together.

3.4 Error Correction

Figure 2.2 shows the pipelining used to synchronize the digital outputs of each stage. If one stage finds that its decision is outside the expected range, it passes on a carry bit to the next higher stage's error correction logic. If no decision errors were made, all carry bits will be zero. The previous two implementations of the A/D did their error correction off chip, so the logic circuits of Figure 3.10 were newly designed for this layout.

3.5 Clock

A new clock generator cell was created to supply the two phase non-overlapping clock plus the prime clock phases, which turn off in advance. As seen in Figure 3.11, an inverter delay line creates the delay between phases. For example, phi1prime turns off two inverter delays before phi1 turns off, and 9 inverter delays before phi2 turns on.

3.6 Top Level

The signal flow between stages is shown in Figure 3.12. Each stage sends opamp, DAC, and criss cross signals to the following stage. Each stage also outputs three bits. The clock and bias signals going to the stages are illustrated in 3.13. Figure 3.14 shows the timing of these signals. In these diagrams, phiA and phiB are relative to individual stages, while phi1 and phi2 are global.

The previous designs arranged the four stages in a horizontal line, with opamp and comparator sections side by side within each stage. Clock and bias lines ran horizontally across the layout. This scheme caused some waste of space since the comparators had to be stretched to match the pitch of the opamp. This problem would be even more severe in the current design because of the large size of the opamp capacitors. A much better fit resulted from placing the comparator section on top of the opamp section as shown in Figure 3.15. In addition, this arrangement meant that no clock lines would have to run through the opamp section, a situation which is quite beneficial in reducing noise. On top of the comparator section is the digital control.

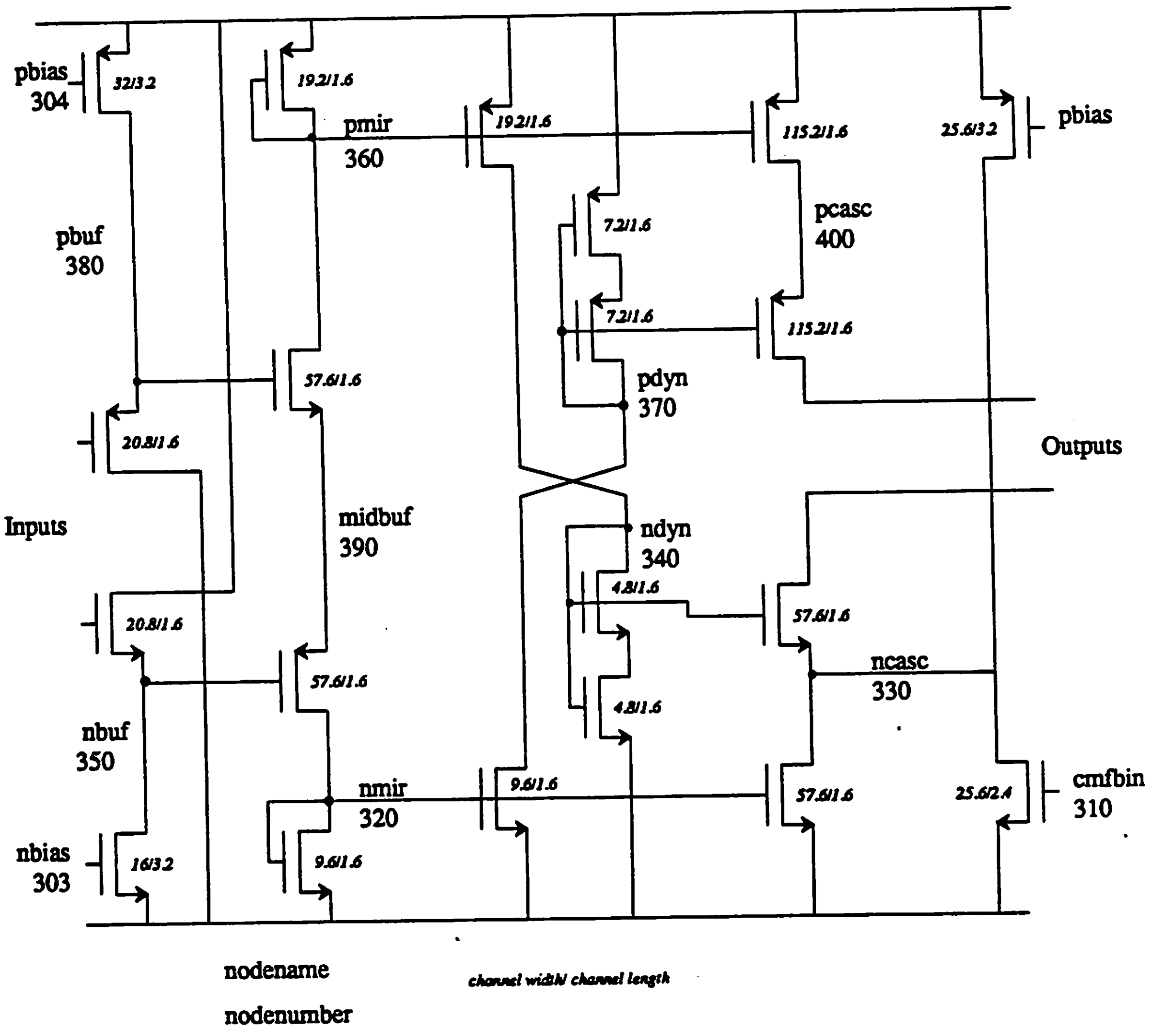
The placement of the stages is also novel. Instead of four in row, the stages are placed in a square, each stage in its own quadrant. See Figure 3.16. The first and third stage are placed side by side on top, as are the second and fourth stages, on the bottom. Since adjacent stages use opposite clocking, ie. phi1 and phi2 are reversed in stage 2 as compared to stages 1 and 3, this floorplan places the in-phase stages side by side. Thus they can share the same horizontal clock lines. Another benefit of this placement is a more nearly square aspect ratio. The output pipelining latches are on the top and bottom, and the error correction logic is at the bottom of the layout. The bias generator cell is on the left side of the first stage.

Clock lines enter from either the top center or bottom center. The digital outputs come out of the bottom. All of the analog interfaces are grouped in the upper left.

The total size of the A/D cell, including error correction, but not including clock generator or I/O pads, is $1.7mm$ by $1.3mm$, or $2.2mm^2$. This compares favorably with the $1.25\mu m$ layout which occupied $2.1mm^2$, and did not include error correction.

Figure 3.1: Opamp Core Circuitry

OPAMP HALF CIRCUIT



OPAMP BIAS

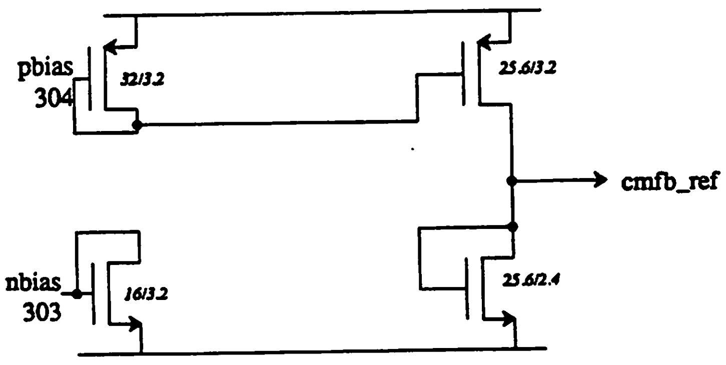
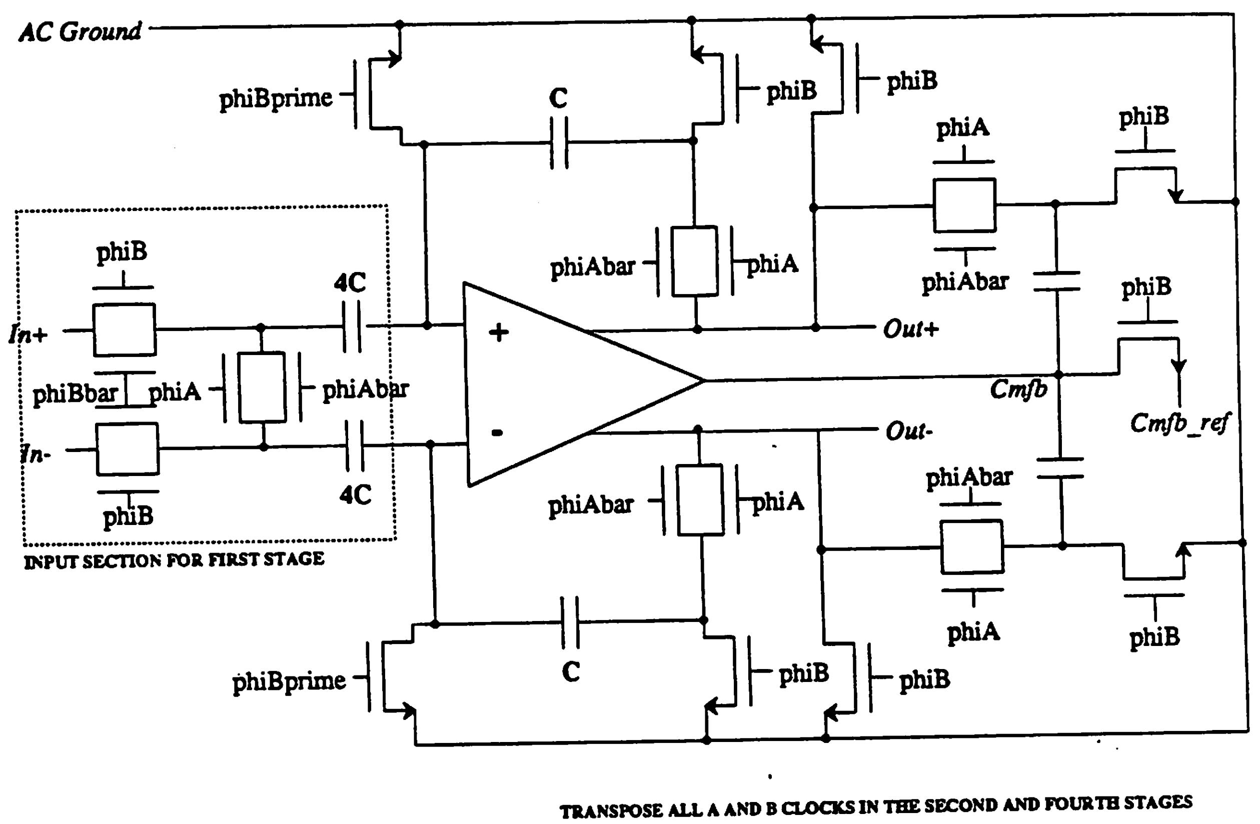


Figure 3.2: Opamp Section Circuitry



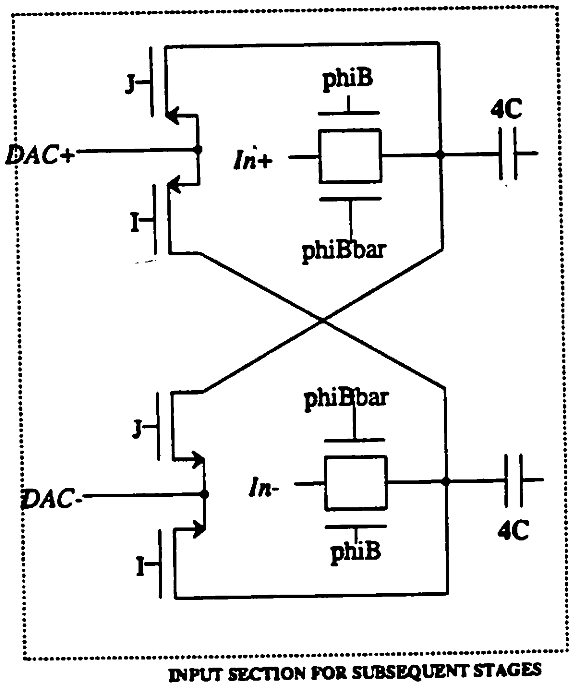
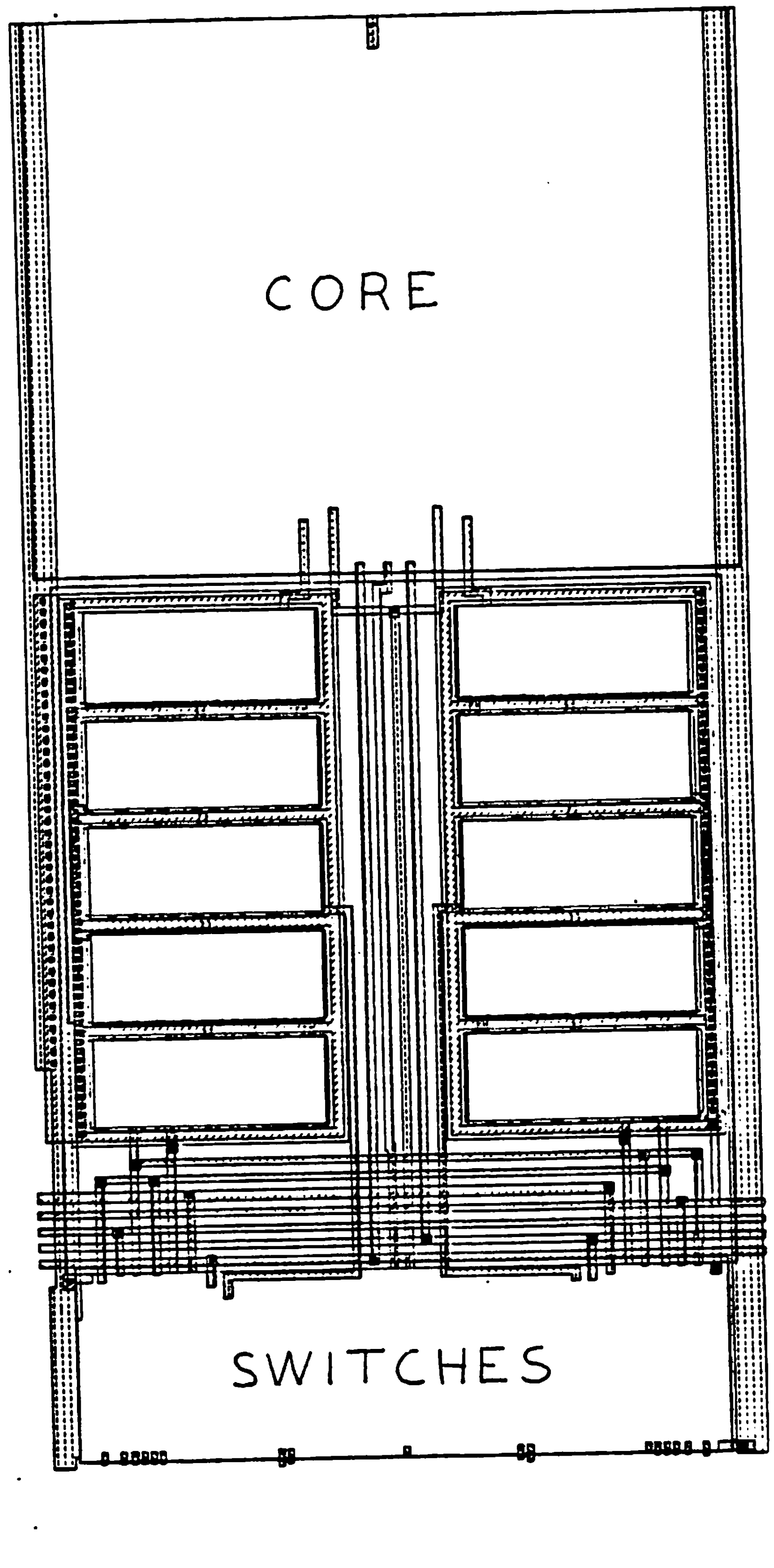


Figure 3.3: Opamp Section Layout



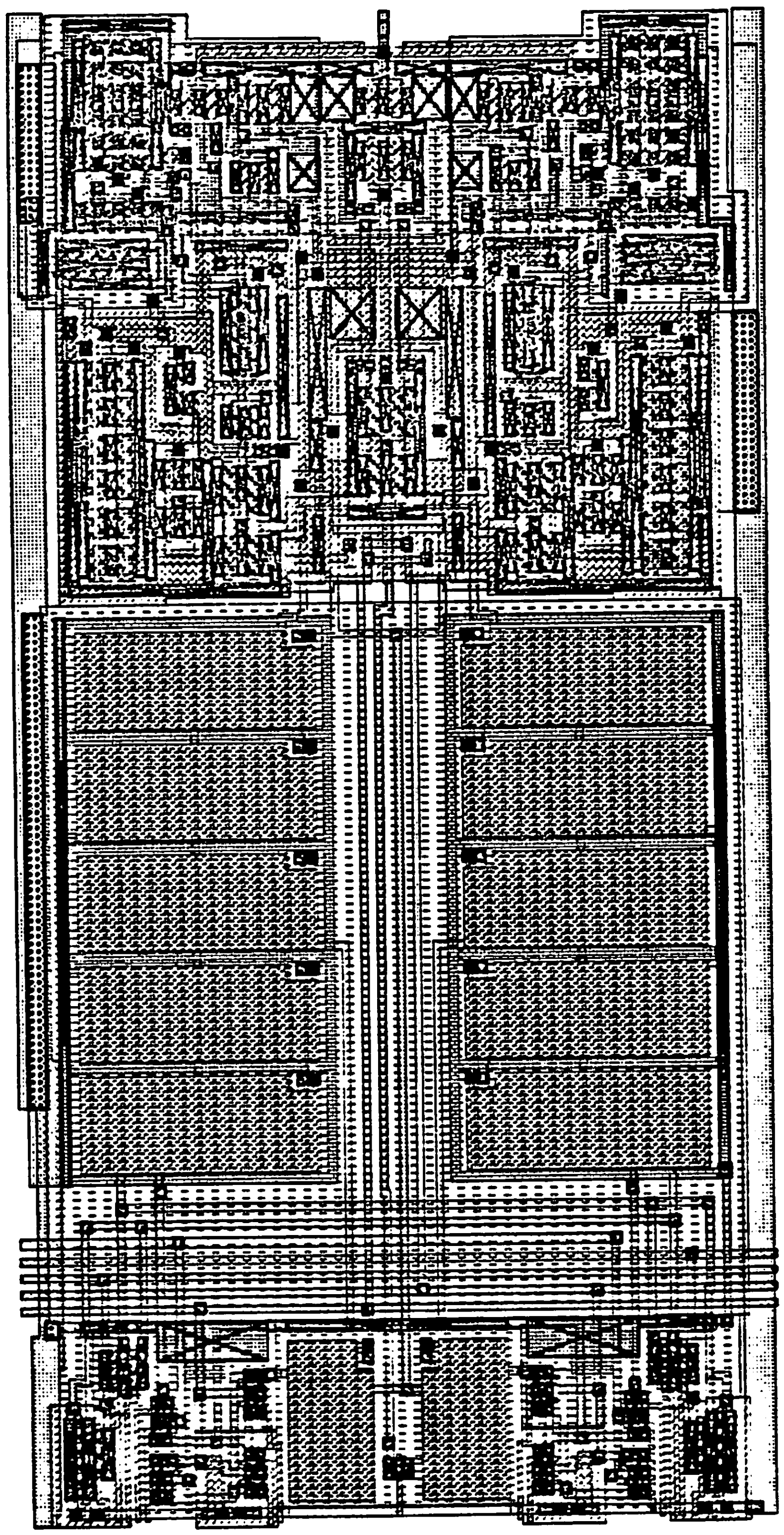
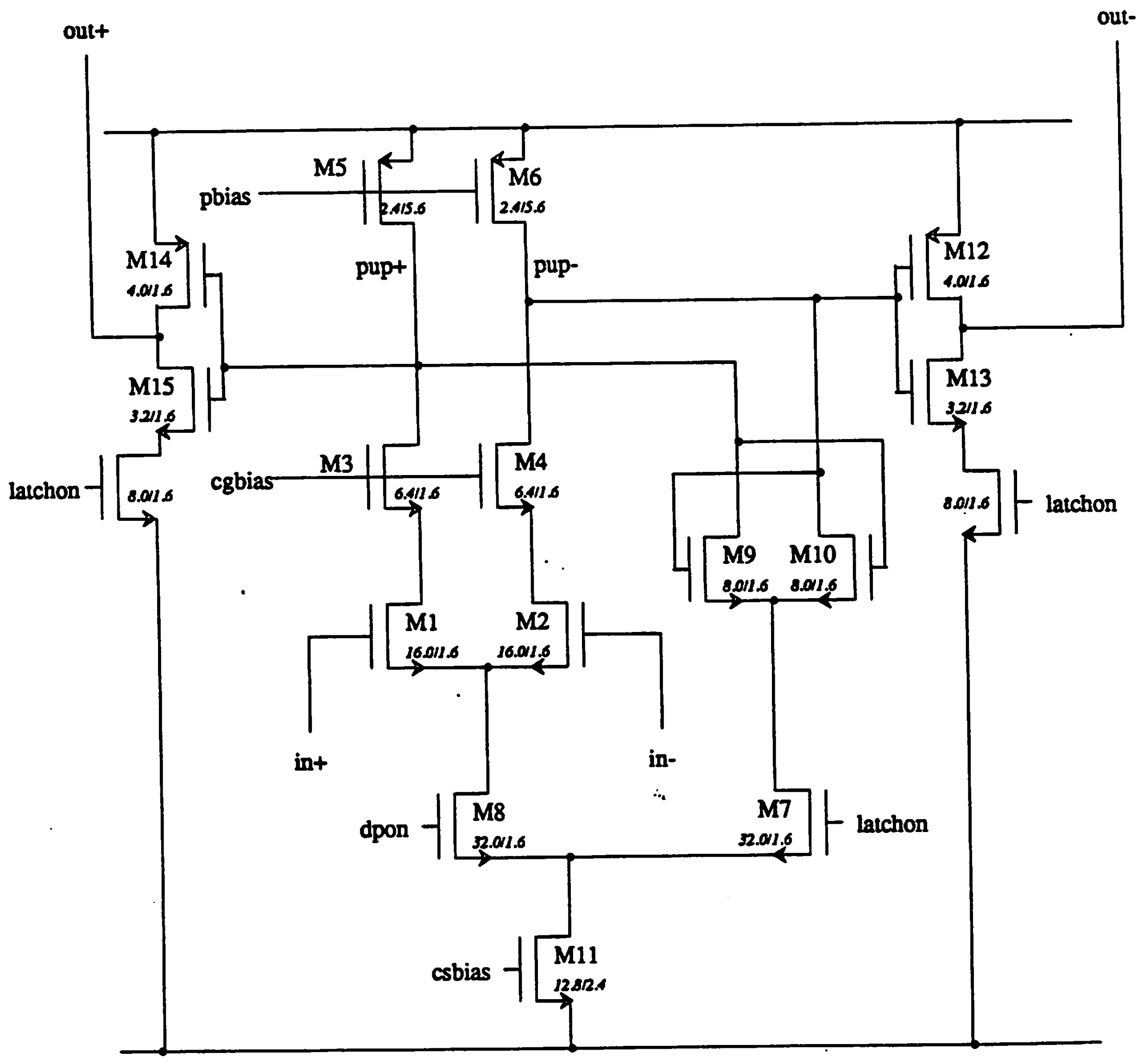


Figure 3.4: Comparator Core Circuitry



channel length/ channel width

Figure 3.5: Comparator Section Circuitry

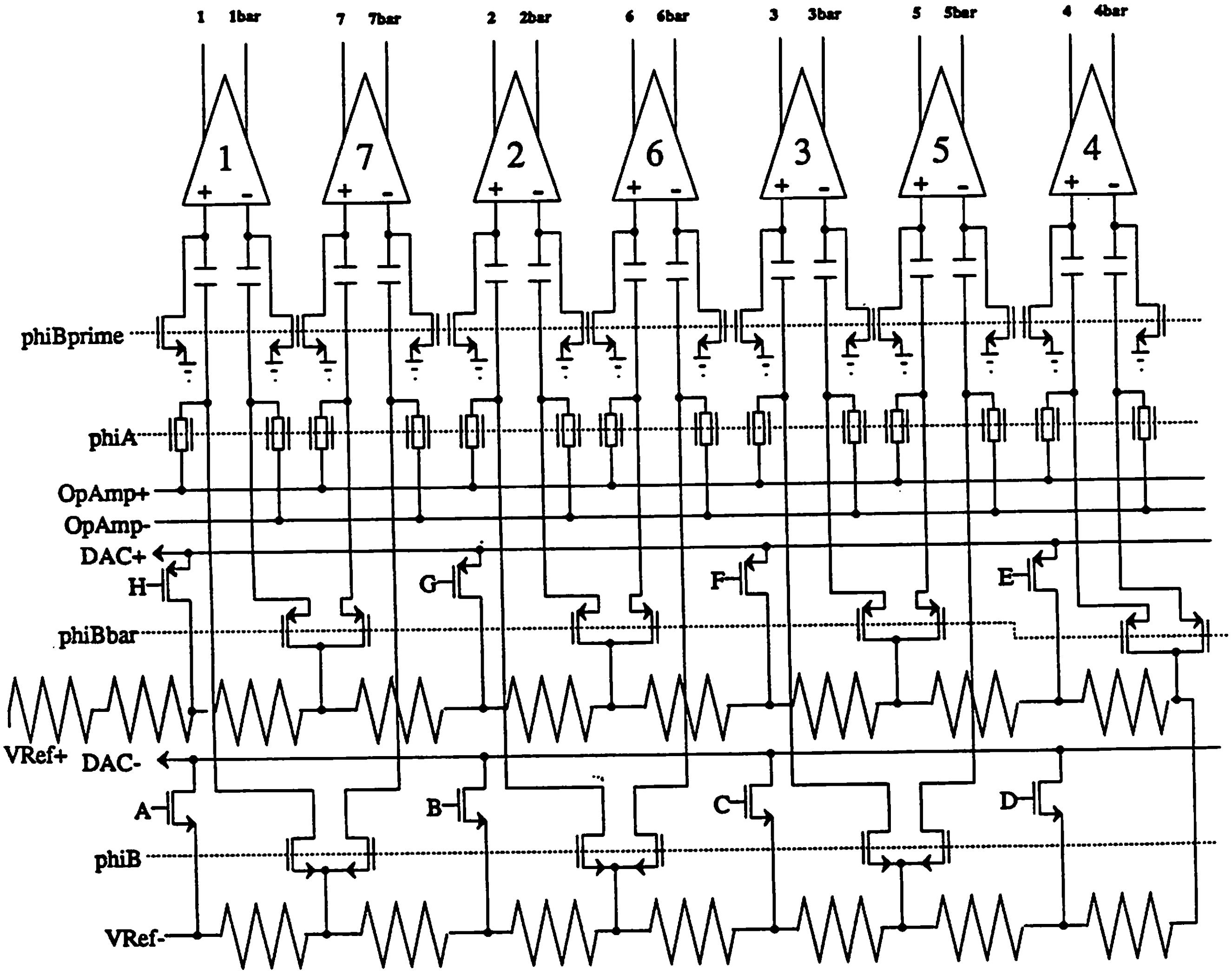


Figure 3.6: Comparator Section Layout

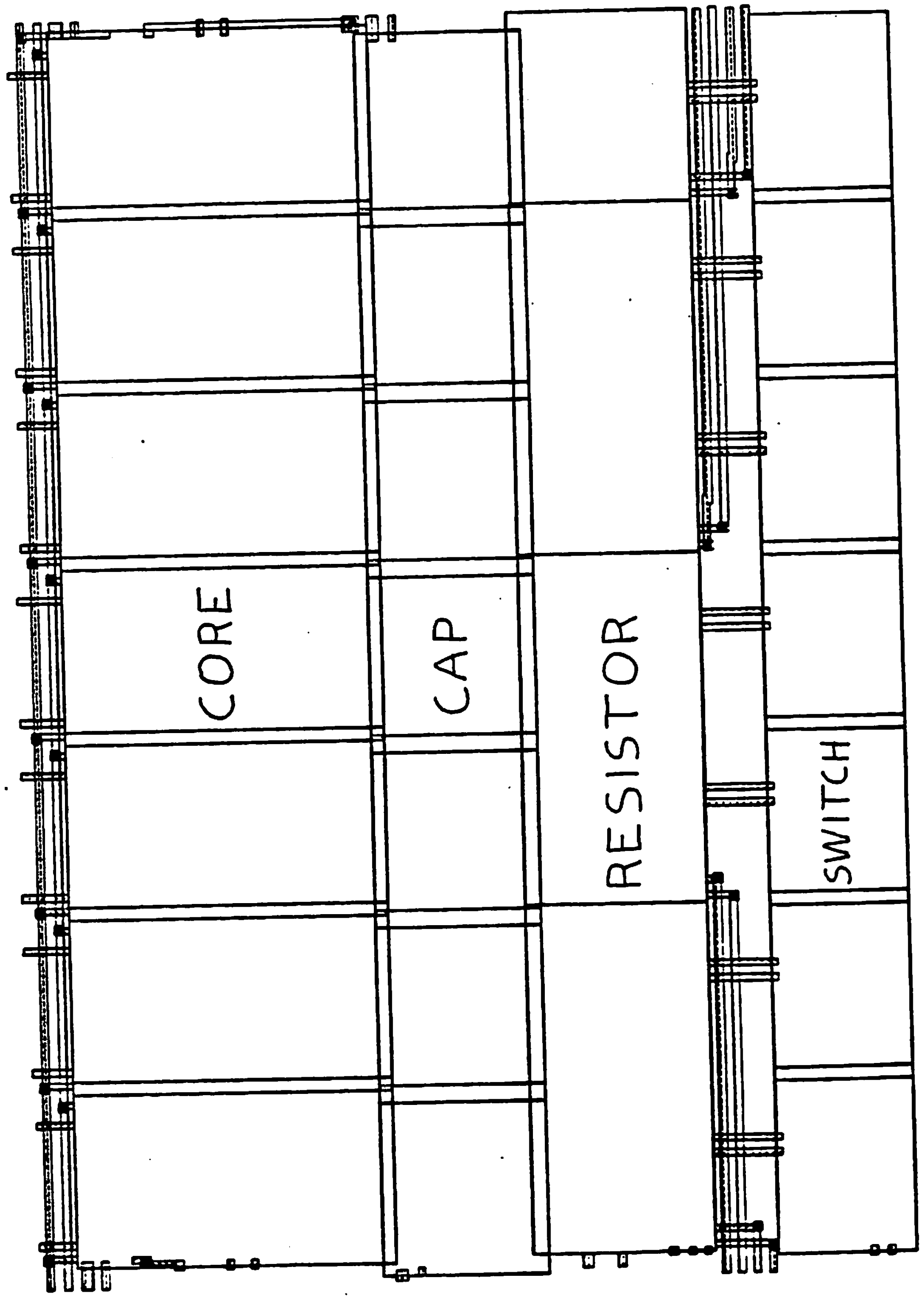


Figure 3.7: Comparator Core Layout

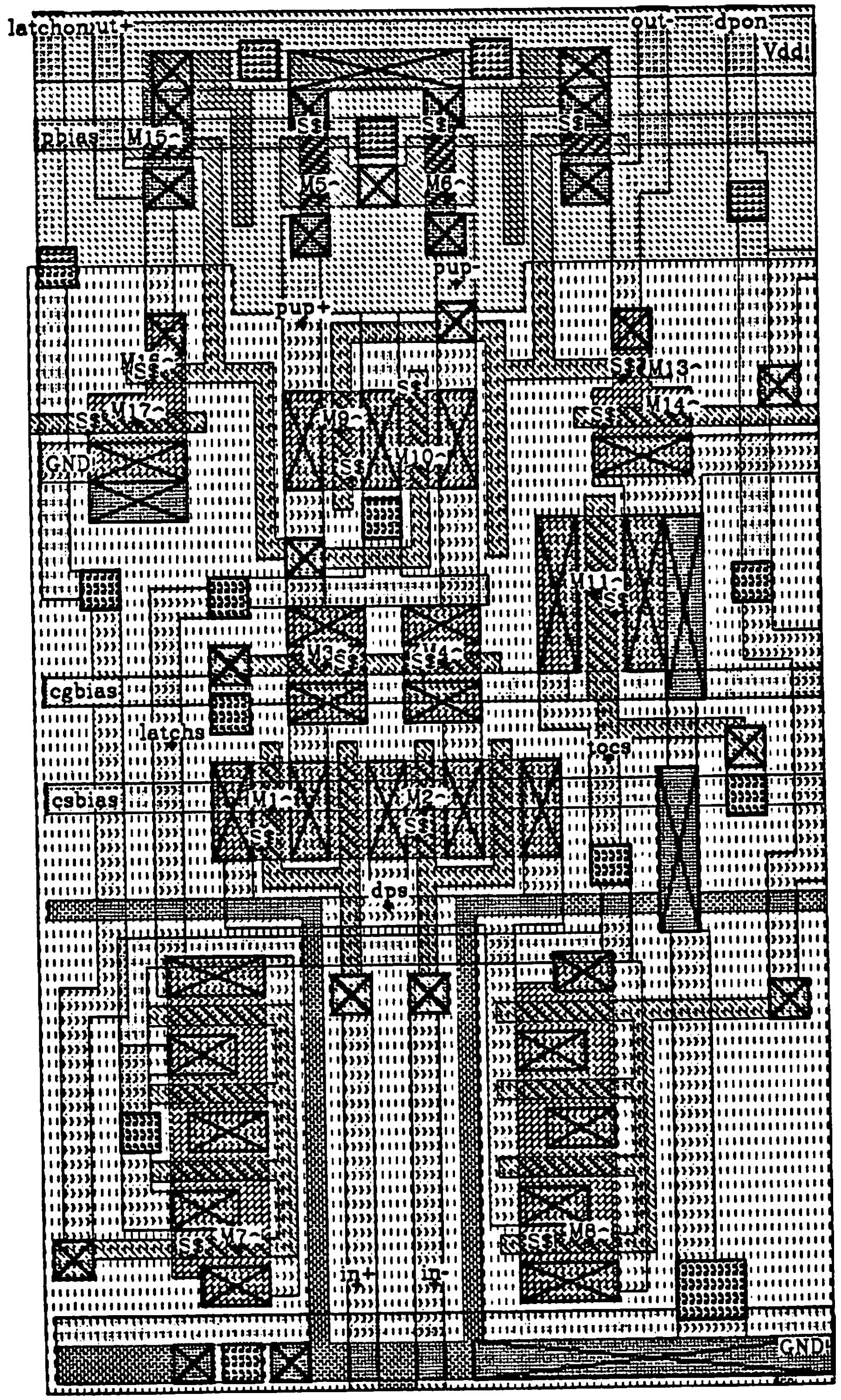
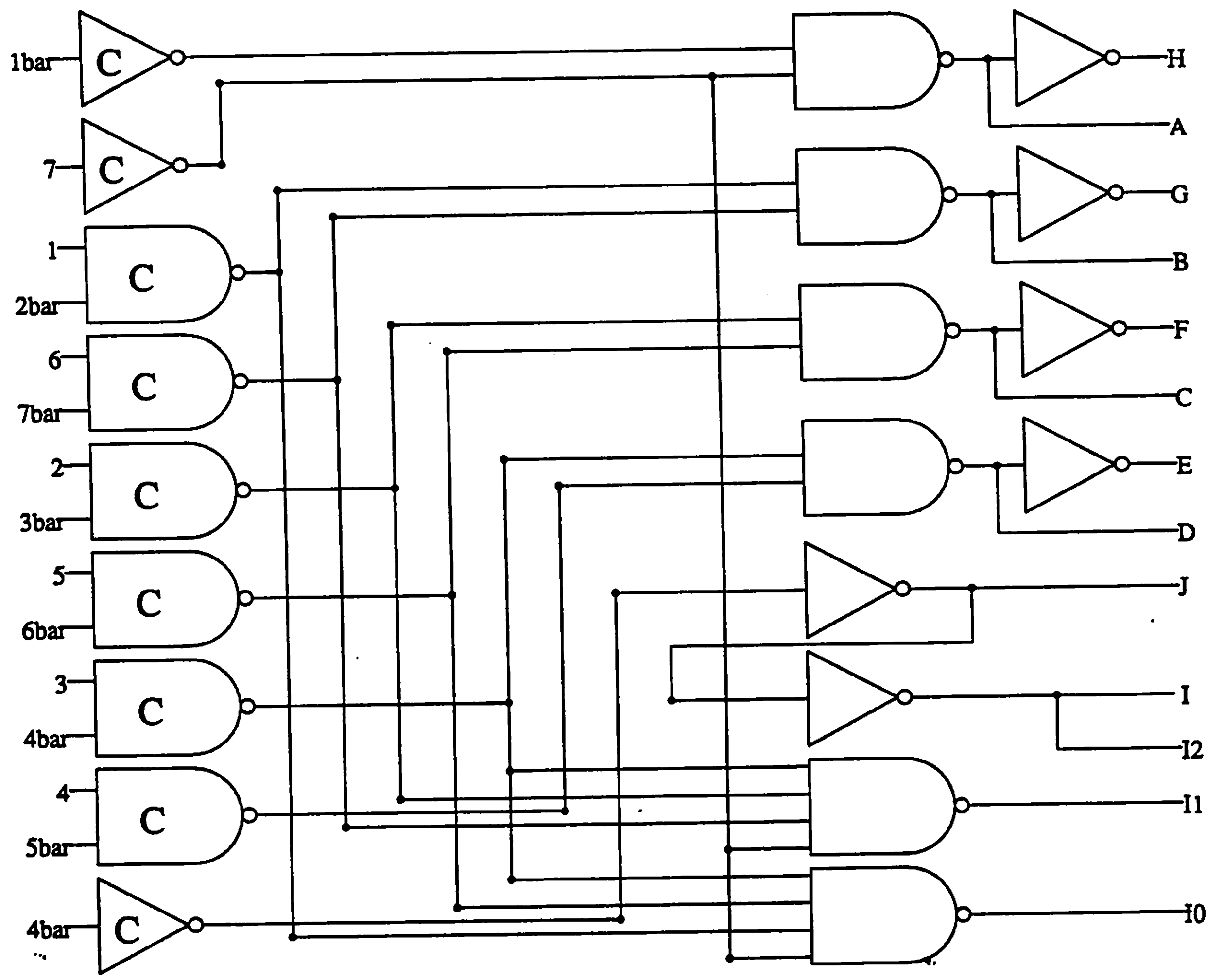


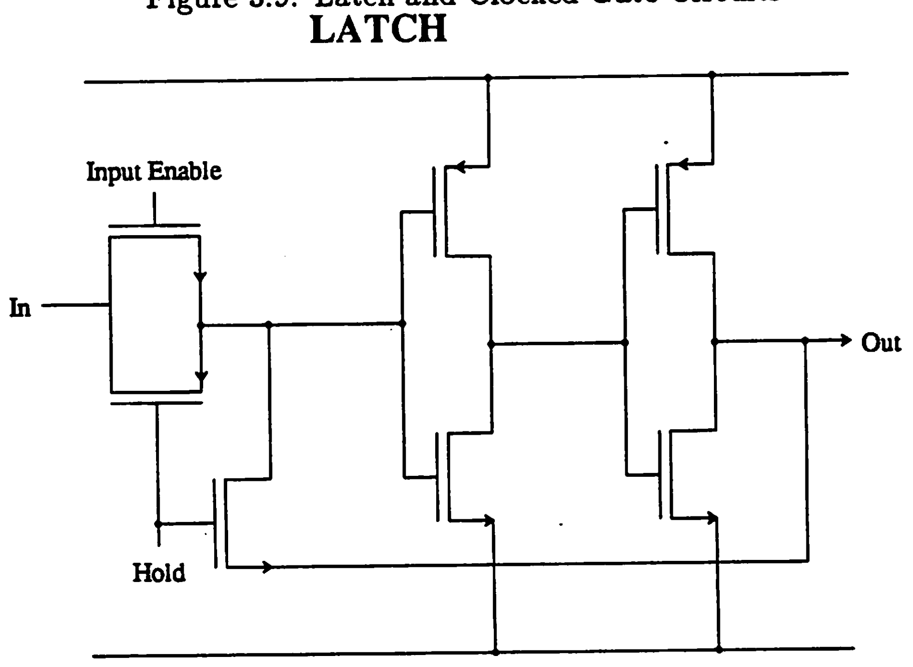
Figure 3.8: Digital Control Circuitry

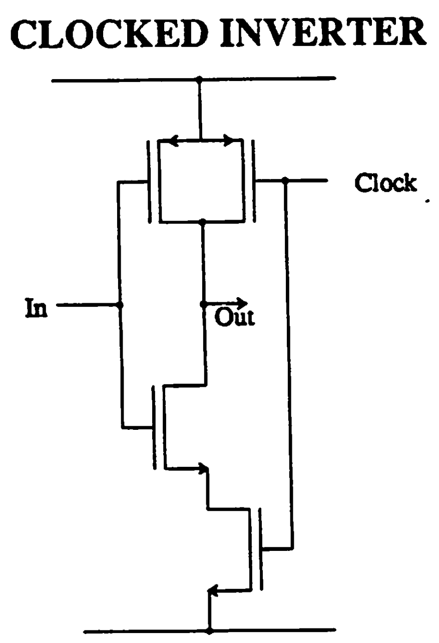


C CLOCKED GATE

A - G: DAC CONTROLS
I J: CRISS CROSS CONTROLS
I0 -I2: 3 BIT A/D VALUE

Figure 3.9: Latch and Clocked Gate Circuits





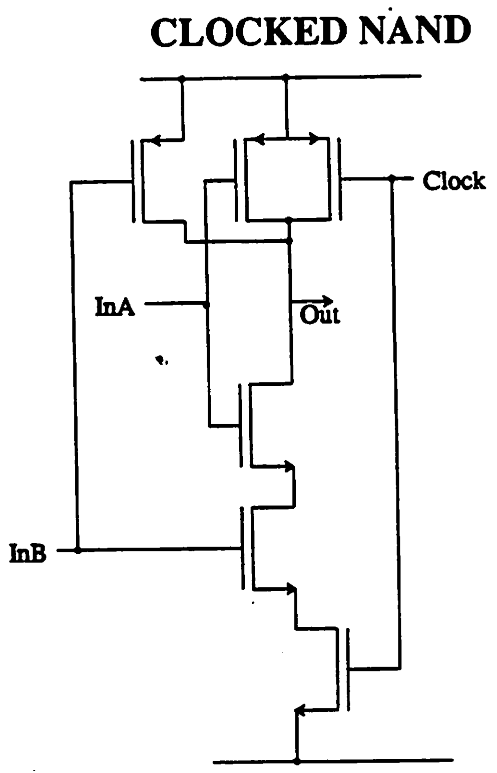


Figure 3.10: Error Correction Logic

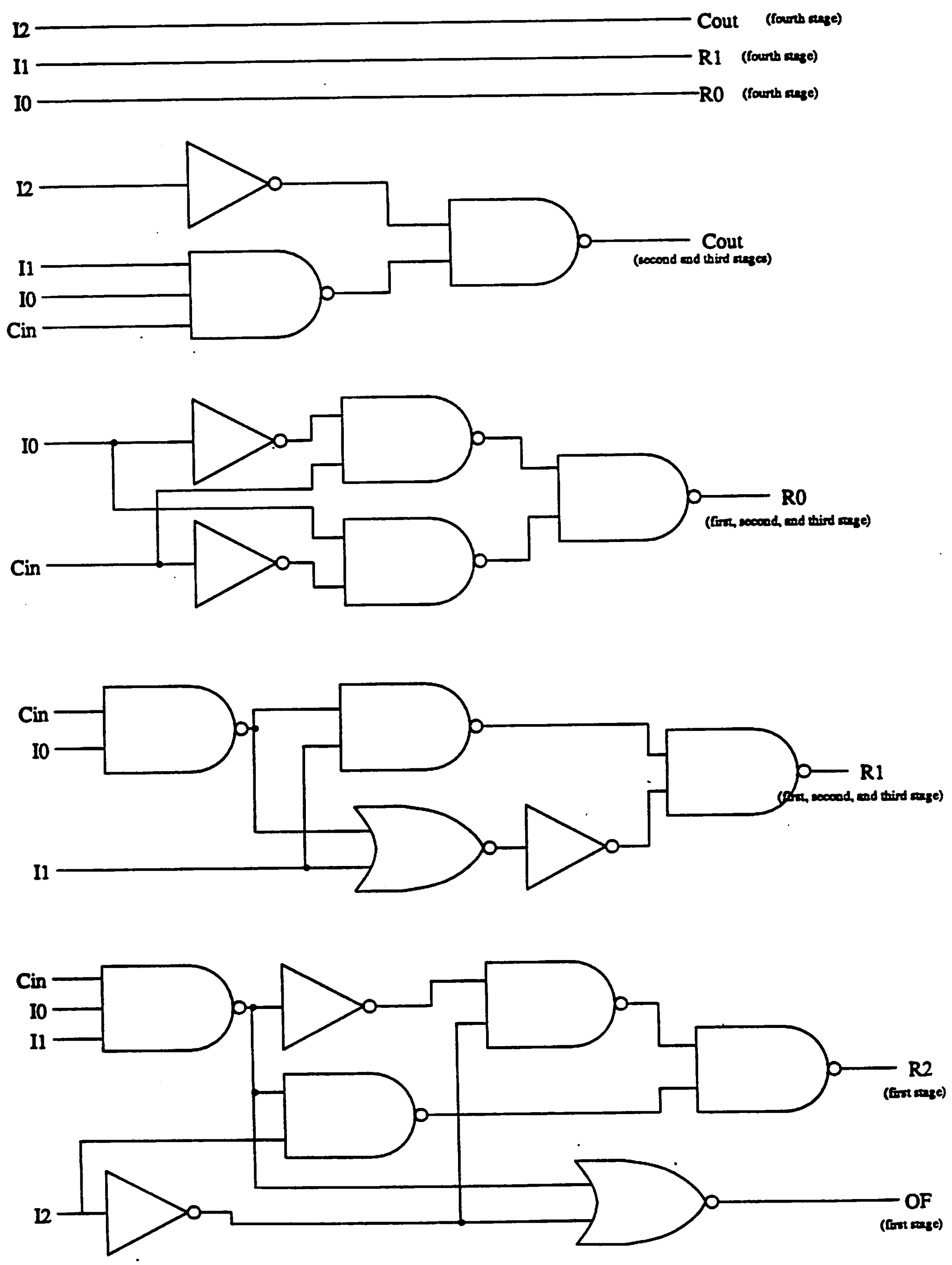


Figure 3.11: Clock Generator Circuitry

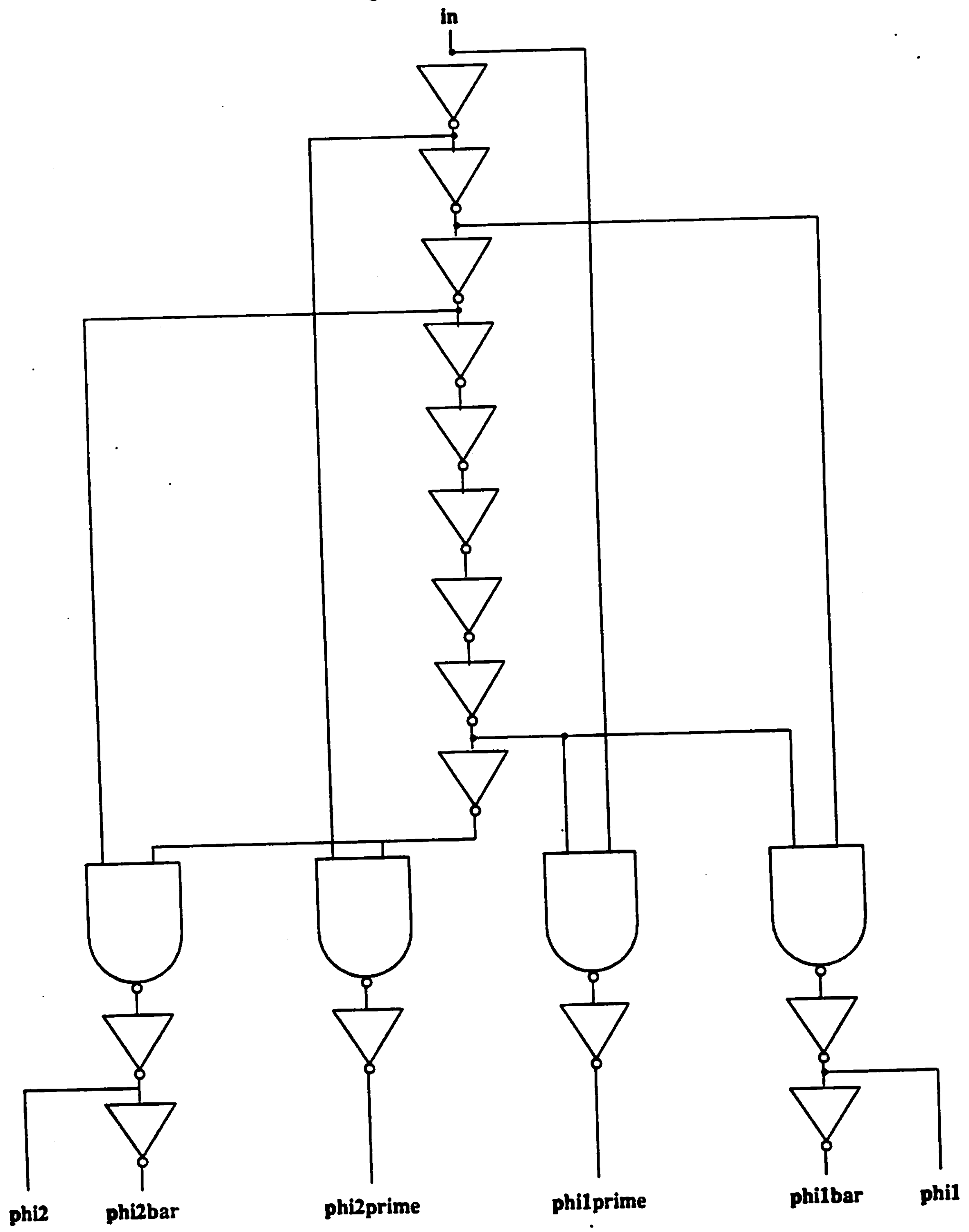


Figure 3.12: Signal Flow Between Stages

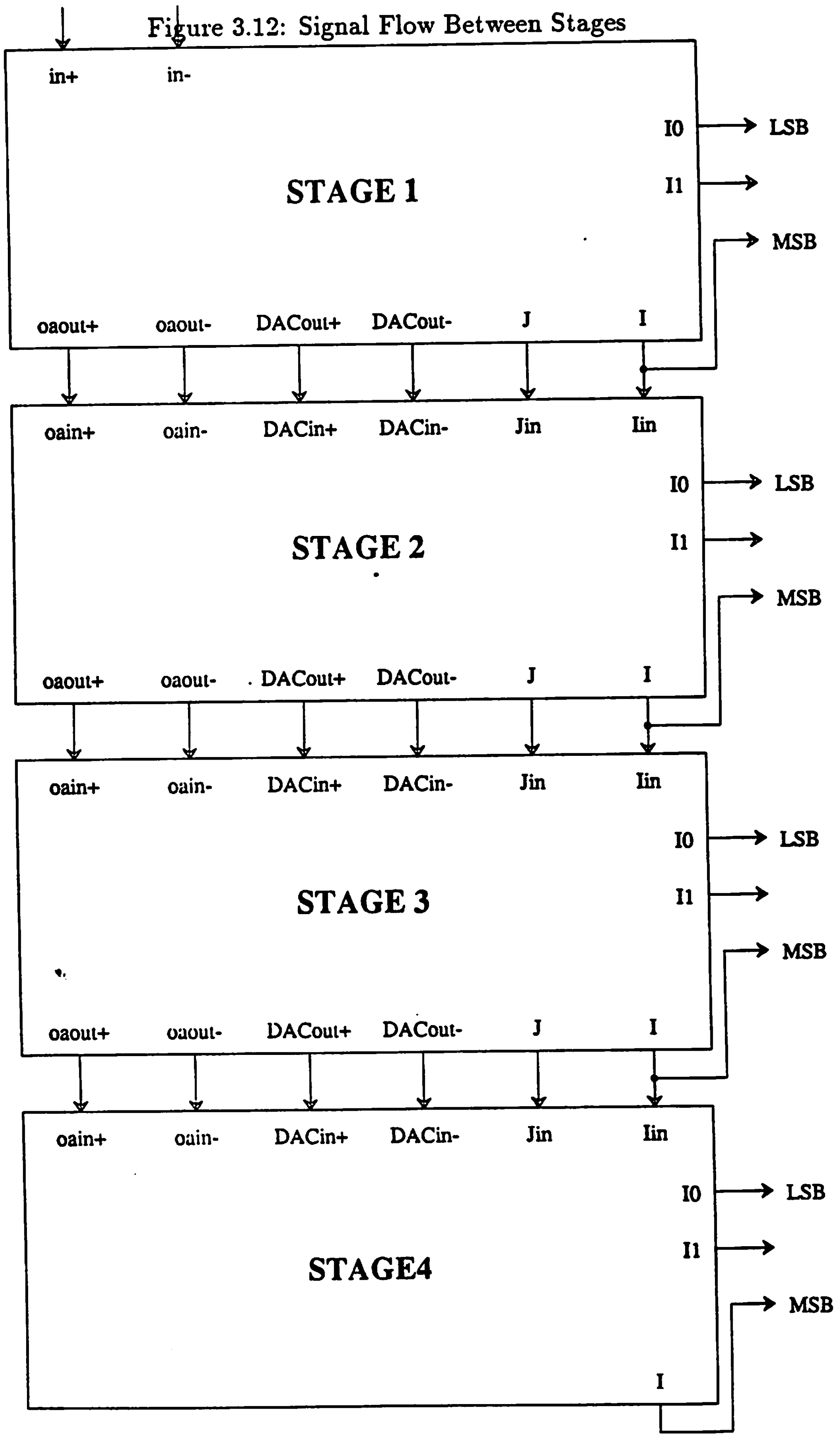


Figure 3.13: Overall Clocking and Biasing

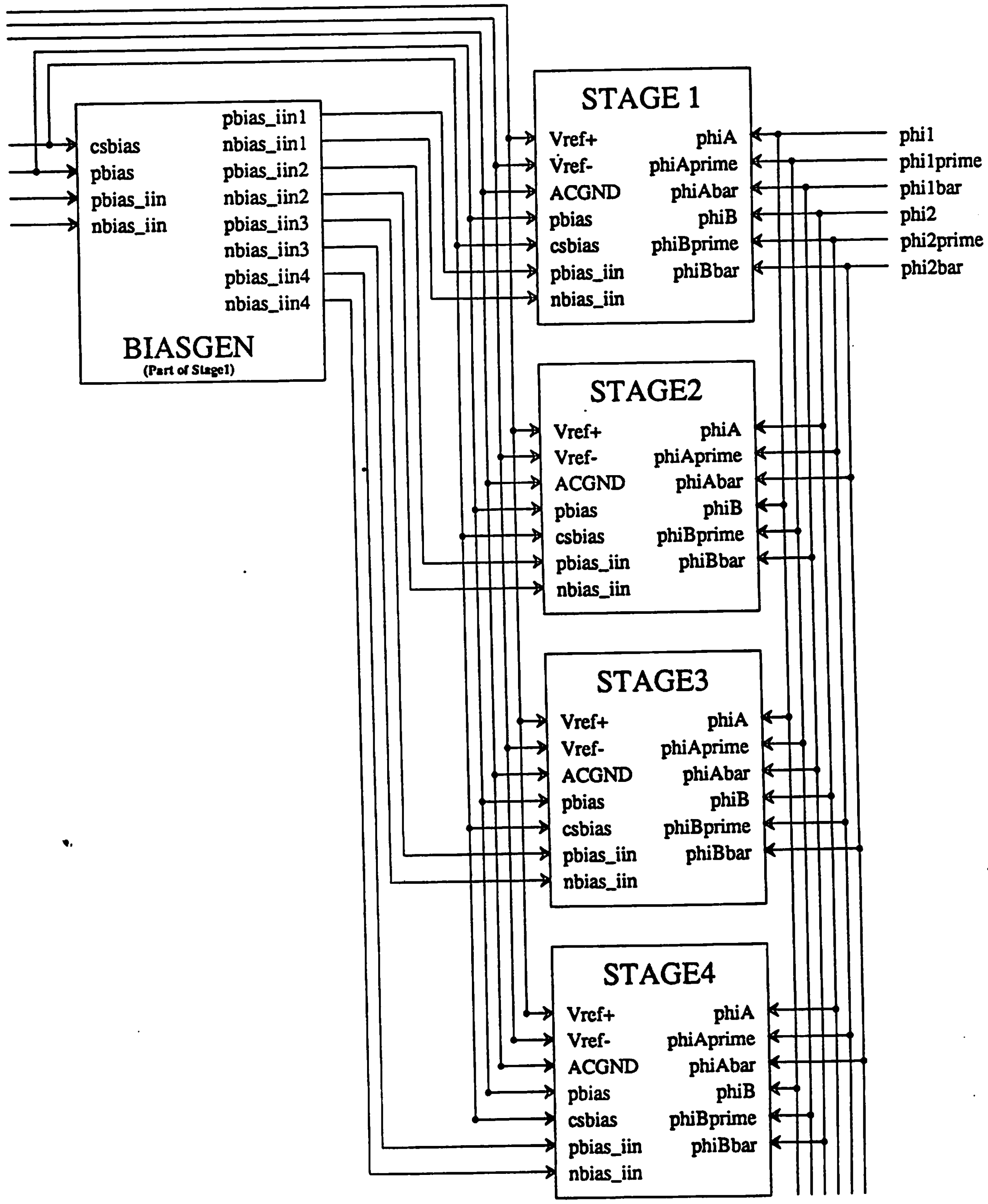


Figure 3.14: Timing of Interstage Signals

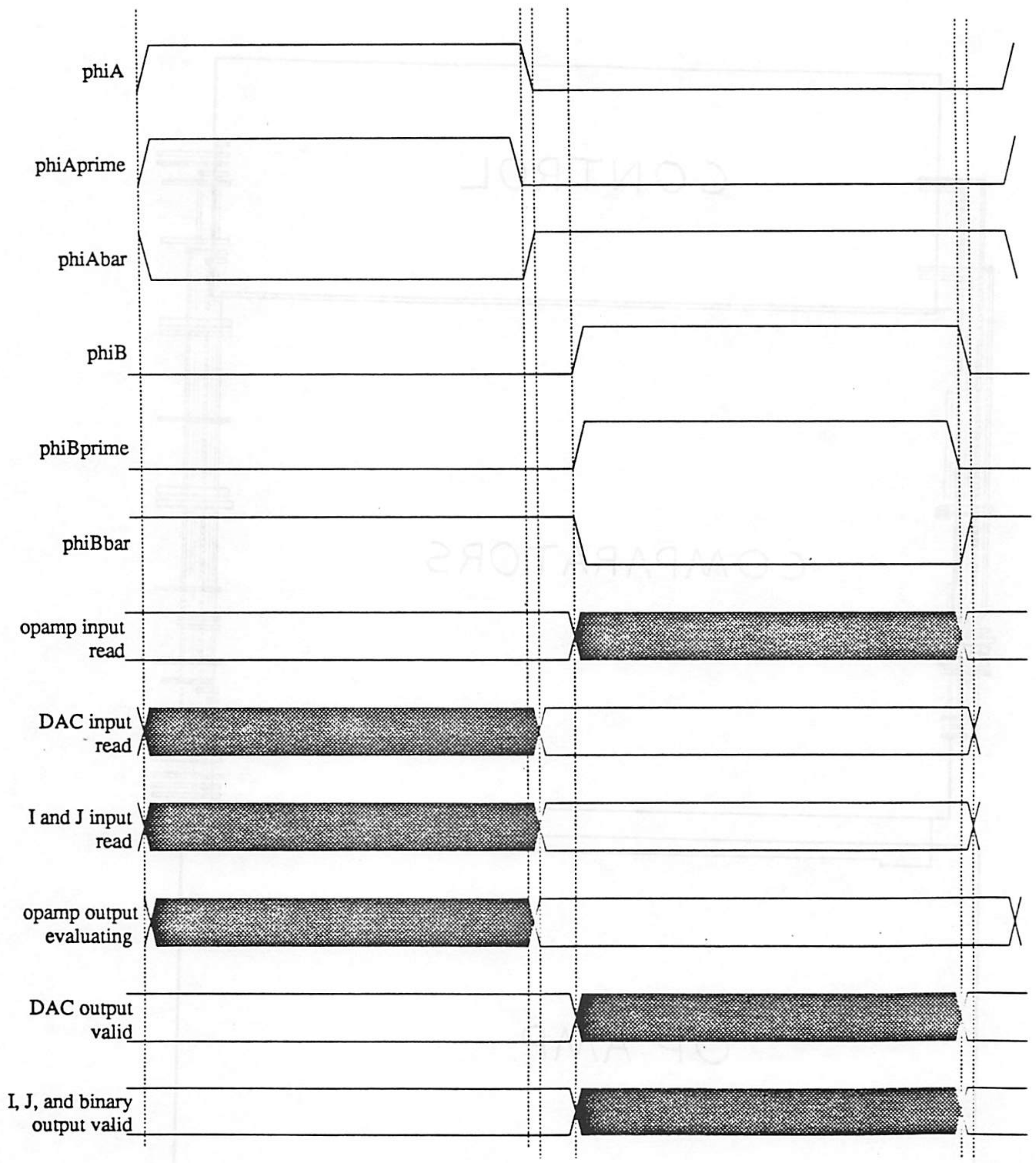
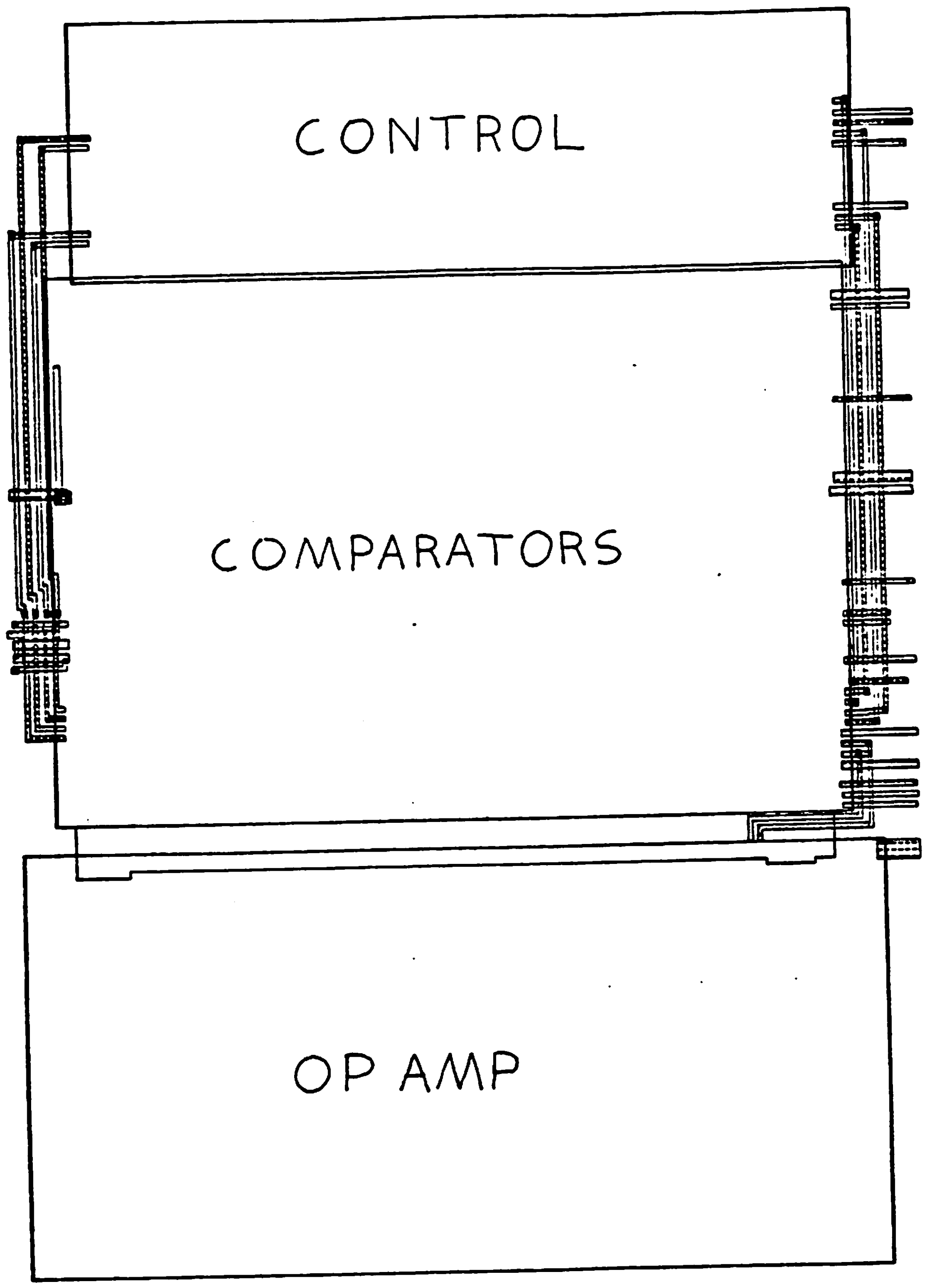


Figure 3.15: Floorplan of a Single Stage



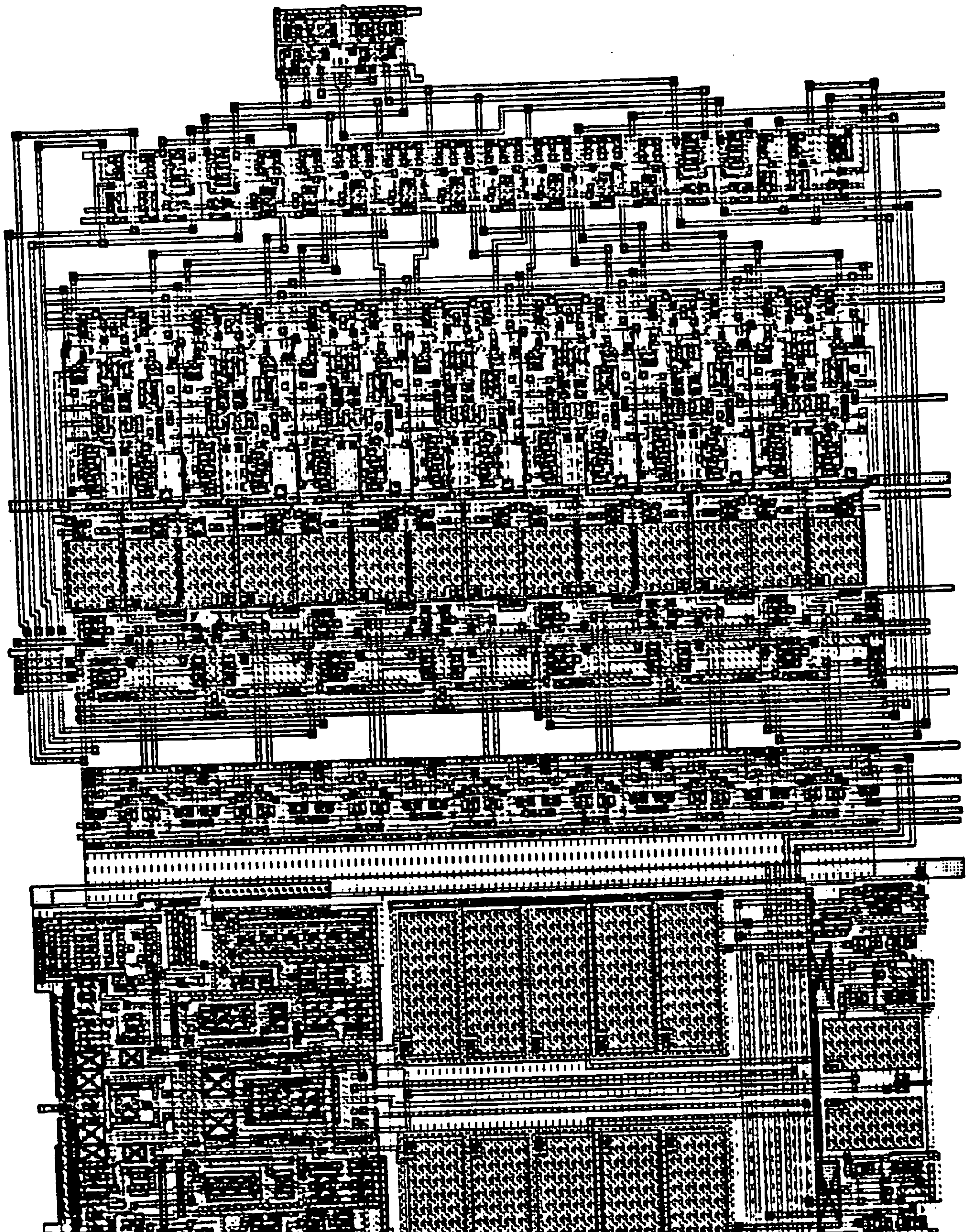
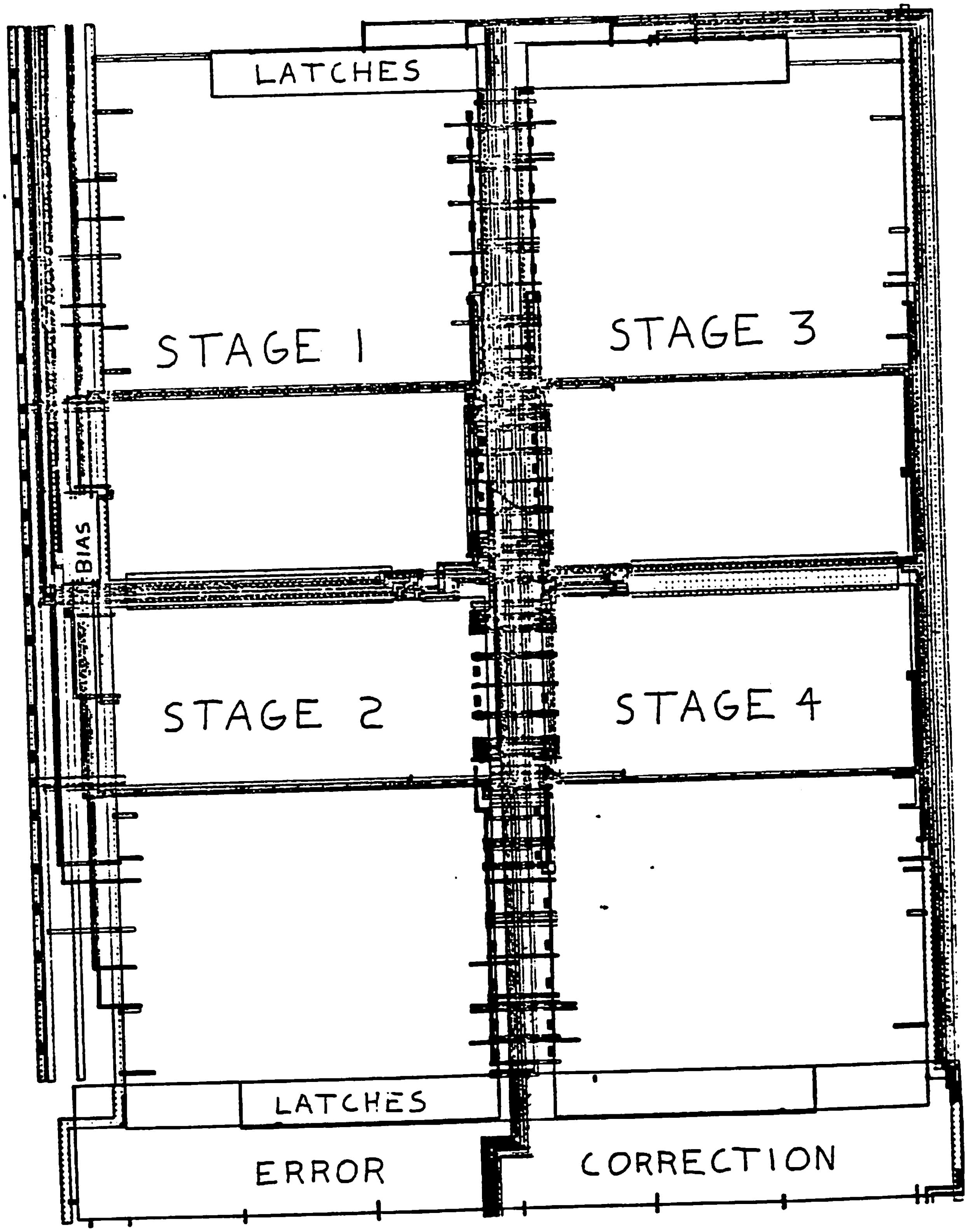
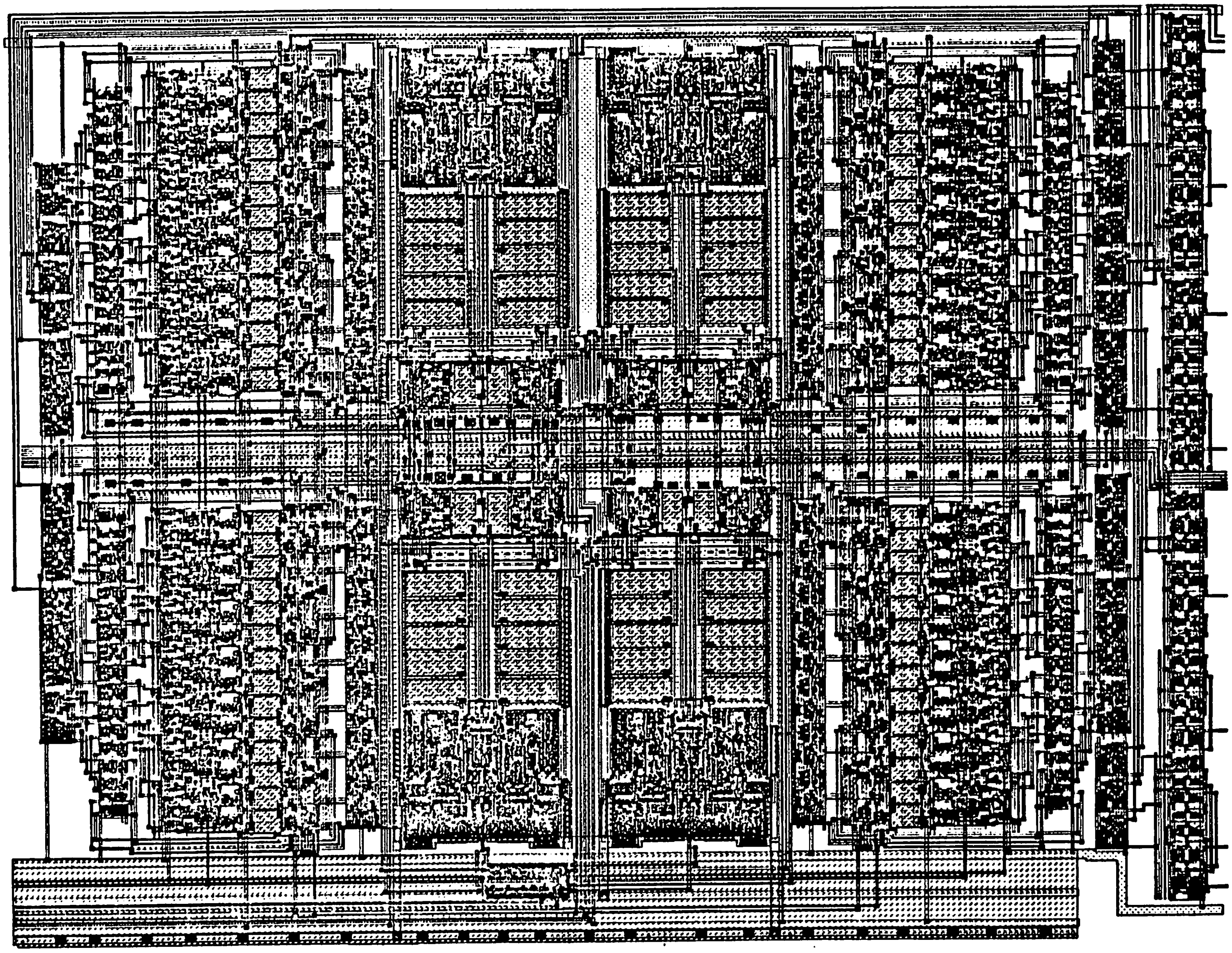


Figure 3.16: Floorplan of the A/D





Chapter 4

Ext2spice: A New Tool

4.1 Description

Heretofore, extracting a circuit from the magic layout editor into spice format has been a three stage process. Within magic, the layout is extracted. This produces a *.ext* file for each cell in the layout. The next step is to run ext2sim on the top level *.ext* file. Ext2sim combines the various *.ext* files into a single *.sim* file, essentially a flattened circuit containing transistors and capacitors. Finally, sim2spice converts the *.sim* file into a format acceptable to spice.

Unfortunately, some valuable information is lost along the way. The magic extraction does not explicitly output the areas and perimeters of diffusion regions, a major source of parasitic capacitance. Instead, it approximates these parasitic capacitances as lumped capacitances. If given the sizes of the diffusion regions, spice could form its own, more accurate, model of these voltage dependent parasitic capacitances.

The spice extractor actually does output areas and perimeters of various regions, ostensibly to calculate resistance. This is fortunate: no modification of the magic extractor is neccessary; all of the necessary information is already present in the *.ext* files. A new program written in C, ext2spice, was developed to read this resistance information and translate it into areas and perimeters of source and drain diffusion regions in the spice circuit. Ext2spice converts directly from ext format to spice format, so it is not necessary to use sim2spice. Nor is it necessary to use ext2sim, unless the user needs the sim format file for other simulator programs.

Ext2spice does require two minor modifications of the extract section of the magic technology file. (The magic technology file contains all of the process dependent information used by the magic layout editor and extractor.) The nominal parasitic capacitance of the diffusion regions should be set to zero to suppress the creation of the lumped capacitances from these regions. Such capacitors would be redundant with the capacitors formed by spice. It is also necessary to modify some of the resistance specifications, to aid ext2spice in determining which resistance classes correspond to the diffusion regions. Changing the resistance values will not affect the spice circuit, which does not contain resistors. If the user is concerned with the effect of these changes on other simulation tools, it is a simple matter to create a special extraction style used only for ext2spice.

Figure 4.1 shows what a typical technology file extract style might look like. (This style is similar to the style for a $3\mu m$ layout in the currently used scmos.tech21 technology file. The code in Figure 4.1, however, contains corrections to mistakes in the scmos.tech21 file.) Figure 4.2 shows a typical extract style intended to be used for ext2spice.

In addition to improving the accuracy of the spice circuit, ext2spice contains features which make the circuit better organized and easier to debug. Rather than flattening the whole circuit, ext2spice preserves the layout's cell hierarchy, transforming each cell into a spice subcircuit. This makes it very easy to locate transistors in even a very complicated layout. The user can also assign identification numbers to transistors, and specify source and drain nodes, so the spice circuit can conform to the transistor numbering scheme of a schematic. Finally, ext2spice can merge parallel transistors into larger transistors. This is quite helpful if the layout contains transistors with "fingered" gates, in which case each "finger" of the transistor is extracted as a separate transistor. Ext2spice will recombine these into a single transistor.

Ext2spice also implements two features similar to features found in sim2spice.

```
style lambda=1.5

    lambda      150

    step        100

    resist      poly,pcontact,pfet,nfet          30000
    resist      pdiff,ppd                        80000
    resist      pdc,ppc                          85000
    resist      ndiff,nnd                        40000
    resist      ndc,nnc                          45000
    resist      m2contact                          100
    resist      metal1                             105
    resist      metal2,pad                          50
    resist      pwell                          5000000
    resist      nwell                          5000000

    areacap     poly,pc/active                      43
    areacap     metal1                              22
    areacap     metal2,pad                          14
    areacap     ndiff,ndc/active,nnd,nnc/active    330
    areacap     pdiff,pdc/active,ppd,ppc/active    390

    perimc      ndiff,ndc/active,nncont/active  space      260
    perimc      pdiff,pdc/active,ppcont/active  space      350

    overlap     metal1  pdiff,ndiff,psd,nsd      36
    overlap     metal2  pdiff,ndiff,psd,nsd      10        metal1
    overlap     metal1  poly                     39
    overlap     metal2  poly                     17
metal1
    overlap     metal2  metal1                   28

    fet pfet    pdiff,pdc,ppcont      2      pfet   Vdd!         0
0
    fet nfet    ndiff,ndc,nncont      2      nfet   GND!         0
0
```

Figure 4.1: A Typical (Corrected) Extract Section of Magic Technology File

```
style ext2spice=1.5

    lambda      150

    step        100

    resist      ndiff,ndc/active                    40000
    resist      pdiff,pdc/active                    80000

    areacap     poly,pc/active                         43
    areacap     metal1                                 22
    areacap     metal2,pad                             14


    overlap     metal1  pdiff,ndiff,psd,nsd         36
    overlap     metal2  pdiff,ndiff,psd,nsd         10        metal1
    overlap     metal1  poly                        39
    overlap     metal2  poly                        17
metal1
    overlap     metal2  metal1                      28

    fet pfet    pdiff,pdc,ppcont      2     pfet  Vdd!        0
0
    fet nfet    ndiff,ndc,nncont      2     nfet  GND!        0
0
```

Figure 4.2: Extract Section of Magic Technology File for Ext2spice

It can provide a listing of all node names and numbers, in this case on a subcircuit by subcircuit basis. Also, the user can assign node numbers to specific node names. This feature is even more helpful in a hierarchical spice circuit. The spice description of a cell will remain the same whether the cell is tested by itself, or as a small part of a large circuit. Since cells are not flattened, they can retain their own node numbering schemes without interfering with the circuit at large.

Figure 4.3 shows how ext2spice handled the comparator circuit of Figures 3.4 and 3.7. Notice that the parallel transistors are merged, and the transistors are numbered and oriented as labeled. This subcircuit was taken from the spice file for the entire A/D.

4.2 Algorithm

Ext2spice receives all of its information from the *.ext* files. It reads each *.ext* file twice. During the first pass, it creates a list of all of the cells used in the circuit. Each cell contains a list of all of the instances of other cells (subcells) it contains. If the cell needs to make a connection (merge, in ext) to a node in an instance of one of its subcells, it makes that node a terminal of the subcell. The list of cells is sorted so that a cell will always be listed before all of its subcells.

After reading all of the *.ext* files once, the program contains a complete list of cells and their subcells, but only some of the terminals in each cell have been identified. Some of the terminals in the cells actually are nodes belonging to their subcells. For example, suppose cell A contains instance B which contains instance C, which contains instance D. Suppose further that cell A needs to connect to a node in D. As stated above, the node will be made a terminal of cell B. But it also needs to be made a terminal of cells C and D. Therefore, ext2spice goes through the cells again, pushing the terminals down through the cell hierarchy as far as they need to go.

Now, ext2spice reads the *.ext* files for a second time. It goes through the cells in reverse order, so that a cell is always read before any of its parent cells.

Figure 4.3: Comparator Core Output from Ext2spice

```
*****  Subcircuit from file ./compcinv.ext
.SUBCKT compcinv 0 1 3 500 502 531 530 501 4 506 5 6 307 308 505
M1 7 307 535 0 N W=16.0U L=1.6U AD=45.7P PD=17.1U AS=44.8P PS=15.2U
M2 8 308 535 0 N W=16.0U L=1.6U AD=45.7P PD=17.1U AS=44.8P PS=15.2U
M3 532 500 7 0 N W=6.4U L=1.6U AD=25.6P PD=13.5U AS=18.3P PS=6.9U
M4 533 500 8 0 N W=6.4U L=1.6U AD=25.6P PD=13.5U AS=18.3P PS=6.9U
M5 532 501 1 1 P W=2.4U L=5.6U AD=12.2P PD=12.0U AS=10.6P PS=9.0U
M6 533 501 1 1 P W=2.4U L=5.6U AD=12.2P PD=12.0U AS=10.6P PS=9.0U
M7 536 506 534 0 N W=32.0U L=1.6U AD=92.8P PD=34.0U AS=90.0P
PS=34.1U
M8 535 505 536 0 N W=32.0U L=1.6U AD=89.6P PD=30.4U AS=92.8P
PS=34.0U
M9 532 533 534 0 N W=8.0U L=1.6U AD=32.0P PD=16.9U AS=22.5P PS=8.5U
M10 533 532 534 0 N W=8.0U L=1.6U AD=32.0P PD=16.9U AS=22.5P PS=8.5U
M11 536 502 0 0 N W=12.8U L=2.4U AD=37.1P PD=13.6U AS=51.2P PS=23.5U
M12 531 533 1 1 P W=4.0U L=1.6U AD=16.0P PD=12.0U AS=17.6P PS=15.0U
M13 531 533 9 0 N W=3.2U L=1.6U AD=12.8P PD=11.2U AS=4.4P PS=2.7U
M14 9 5 0 0 N W=8.0U L=1.6U AD=11.0P PD=6.9U AS=32.0P PS=14.7U
M15 1 532 530 1 P W=4.0U L=1.6U AD=17.6P PD=15.0U AS=16.0P PS=12.0U
M16 530 532 10 0 N W=3.2U L=1.6U AD=12.8P PD=11.2U AS=4.4P PS=2.7U
M17 10 4 0 0 N W=8.0U L=1.6U AD=11.0P PD=6.9U AS=32.0P PS=14.7U
C1 506 0 23.0F
C2 505 0 20.0F
C3 532 0 10.0F
C4 533 0 10.0F
C5 502 0 13.0F
C6 0 0 27.0F
C7 1 0 16.0F
C8 536 0 22.0F
C9 501 0 13.0F
C10 534 0 18.0F
.ENDS
```

All nodes, transistors, and capacitors are read from the cell's file, and their values are stored. The program also merges together nodes as specified by ext. When the cell is fully analyzed, it is written to the *.spice* file as a subcircuit. With one exception, all information concerning nodes, transistors, and capacitors within this cell can now be erased from memory. This, incidentally, helps the speed and memory performance of ext2spice: at any given time, it never has to store all of the nodes in a layout, only the nodes within a cell.

The one additional piece of information that must be passed between cells concerns the terminals. A cell must know whether any of its nodes are short circuited within any of its subcells, so that it can give them the same node number. (The circuit would still be topologically correct if they had different numbers, but spice would not accept it.) Therefore, each cell saves information to tell its parents which, if any, of its terminals, are merged together. When it is the parent's turn to be read, ext2spice merges together its nodes that are attached to its subcircuits' terminals, if these terminal had been merged together within the subcircuit. Because of the order in which the cells are read, all of the subcells are guaranteed to be processed before the parent cell. Thus, any merger information that is relevant outside of a particular cell is popped up through its terminals.

After ext2spice rereads, processes, and outputs the top level cell, it is finished.

4.3 Speed

Table 4.1 shows how ext2spice's treatment of cells greatly enhances its speed performance for large circuits. The entire A/D has between four to five times as many nodes, transistors, and capacitors as each individual stage, but only takes twice as much cpu time to convert.

Execution Time On a Sun 3/160	
Section Converted	CPU Time in Seconds
Entire A/D	35
One Stage	15
Comparator Section	5
Opamp Section	4

Table 4.1: Ext2spice Execution Time

Chapter 5

Simulations

During the creation of this A/D, simulations were used for two purposes: circuit design and connectivity confirmation.

Digital

The two digital parts of the A/D, the control section for each stage and the digital error correction, were first tested using esim, a switch level logic simulator. The circuits were extracted from magic and converted to *.sim* format with ext2sim.

Clock

Figure 5.3 shows a portion of the spice simulation of the clock generator with a 2.0pF load on each output. The prime phase turns off approximately 1ns before the regular phase. There is about a 3ns period between phases to ensure they are nonoverlapping. The master clock changed at 50ns.

Comparator

The two major analog circuits - the comparator and the opamp - were designed with the aid of spice. The comparator layout was completely original, but the device sizes were easily derived by scaling the previous implementations. Here, the real task was creating a compact and balanced layout, not designing the circuit. Therefore, it did not take many iterations between spice and circuit modification to create a working comparator. Figure 5.4 shows the simulated voltages at *pup+* and *pup-* (see Figure 3.4) for inputs of +-0.01 V and -+0.01 V.

OpAmp

The opamp is the most critical part of the A/D. The speed of the A/D is limited by the settling time of the opamps. In order to achieve eight bits of accuracy, the opamp must settle to nine bits accuracy in less than one half of a clock period, 50ns for a 10MHz clock. The opamp must also have sufficent gain, and of course must remain stable.

The design process of the opamp core was quite different than in the design of the comparator. The starting point was a completed layout (which had easily expandable transistors). But it took much experimentation to create an opamp with satisfactory specifications.

The speed of the opamp is determined by its transconductance and the capacitance it has to drive. Since the opamp was simulated using the extracted circuit, all of the capacitances from the feedback capacitors, switches, and local wiring were already included in the circuit. In the A/D, the opamps are also loaded by the comparators and the following stage's opamp. The input capacitors of the opamps are 1.0pF and their parasitics are 0.5pF. The capacitance into the comparator stage was tested in spice, by inputting a constant current. The capacitance was estimated at 1.3pF using $C = I(\frac{dV}{dt})^{-1}$. This sums to 2.8pF, but the simulations were conducted using 3.5pF to be more conservative. Using a 10MHz nonoverlapping clock, the opamp was considered to have 42ns to complete its settling.

The stability of the opamp was tested in the time domain, *i.e.* with the transient analysis in spice. Since the load also serves as the compensation capacitor, a smaller load would decrease stability. Figure 5.5 shows the output with only a 1.5pF load. The overshoot was approximately 0.9%.

Gain and Settling as a Function of Bias Current		
Bias Current (uA)	Gain	% of Final Value at 42ns
40	1443	99.30
50	1131	99.70
60	929	99.84
70	911	99.85

Table 5.1: Opamp Simulation Results With 3.5pF Load

A/D

In order to confirm the design and layout of the entire A/D, it is desirable to test the largest assembly of its components as is possible. Unfortunately, spice simulations are extremely cpu intensive, especially for large circuits. The highest level of spice simulation were tests of the four individual pipelined stages. Thus, approximately one quarter of the circuitry was tested at a time. However, the manner in which these parts were extracted helped to confirm the connectivity of the whole layout, not just each stage. The entire layout was extracted and then converted to spice using ext2spice. Since the spice circuit was organized into subcircuits, it was a simple matter to remove the subcircuits representing three out of four stages and the error correction (as well as some parasitic capacitors which would be left dangling), leaving one stage to be simulated. The stages were only removed temporarily by making spice treat them as comments.

Of course, the load on the tested stage from the other three stages was no longer present; 2.2 pF capacitors had to be added to the opamp output to emulate the loading of the following stage. This is a conservative estimate since the input capacitance of the opamps is 1.6pF. In order to increase the chances of having the spice3 program successfully complete these simulations, some of its accuracy tolerances were relaxed. This did not have a major impact on the value of the simulations, since the most critical parameters - such as opamp settling - were learned from the previous, smaller simulations. The main purpose of the stage level simulations was to check the three bit decision outputs, the criss cross control

```
*****  Subcircuit from file
/usr3/rwb/burstein/magic/comp/resd.ext
.SUBCKT resd 0 1 3 4
C1 3 0 5.0F
C2 3 0 3.0F
.ENDS
```

Figure 5.1: Resistor Cell as converted by Ext2spice

outputs, the DAC outputs, and the opamp outputs (in coarse detail).

An important point here is that each stage was tested individually, but from within the same circuit. Thus, while testing the second stage, its inputs were put into the same nodes from which the outputs of the first stage were measured. The inputs to the latches and error correction logic were the same nodes as the digital outputs of the four D/A stages. Using spice with ext2spice fullfilled two functions: testing functionality within each section, and confirming connectivity between sections.

As an additional illustration of ext2spice's benefits, consider the way it greatly simplified the task of including the resistor string in the spice circuit. The resistors are made of polysilicon. However, magic thinks that polysilicon is a wiring layer, not a resistor layer. Therefore, it extracts the entire resistor string as one single node. To prevent this calamity, the resistors were created with a narrow break in them, separating their two terminals into separate nodes in magic. This break was easy to repair when simulation was completed since all of the resistors in the A/D are instances of the same cell. The break was also made to deliberately violate design rules, so that it would not pass through a final design rule check unnoticed!

Figure 5.1 shows what ext2spice did to the resistor cell. All of the resistors were correctly placed, but they were effectively infinite resistors (between subcircuit nodes 3 and 4) with some parasitic capacitances on the terminals. Simply adding one resistor to the subcircuit, as shown in Figure 5.2, created the sixty four resistors in the four resistor strings.

```
*****  Subcircuit from file
/usr3/rwb/burstein/magic/comp/resd.ext
.SUBCKT resd 0 1 3 4
R1 3 4 500
C1 3 0 5.0F
C2 3 0 3.0F
.ENDS
```

Figure 5.2: Modified Resistor Cell

Figure 5.3: A Transition of Nonoverlapping Clocks

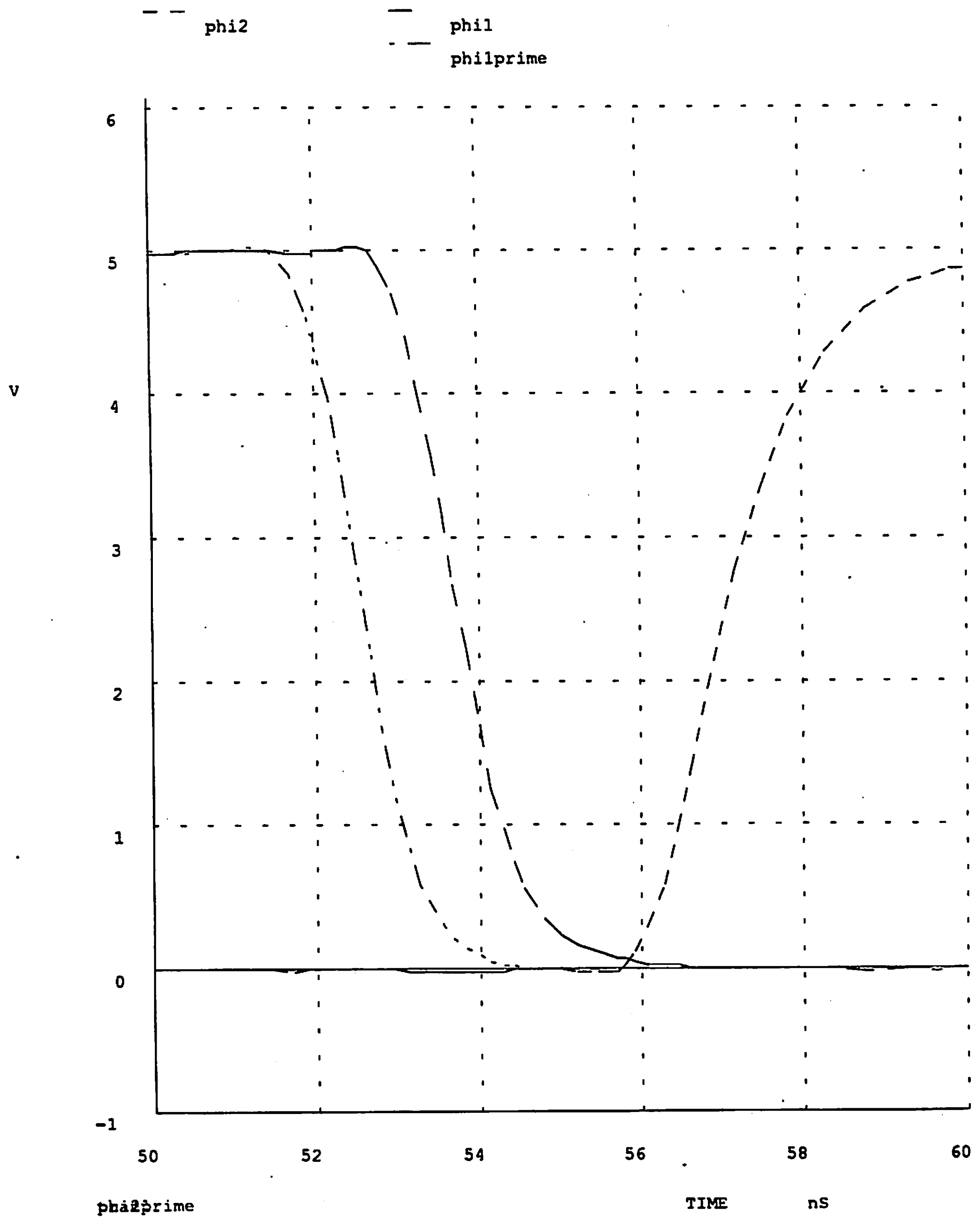


Figure 5.4: Voltages at *pup+* and *pup-* in Comparator

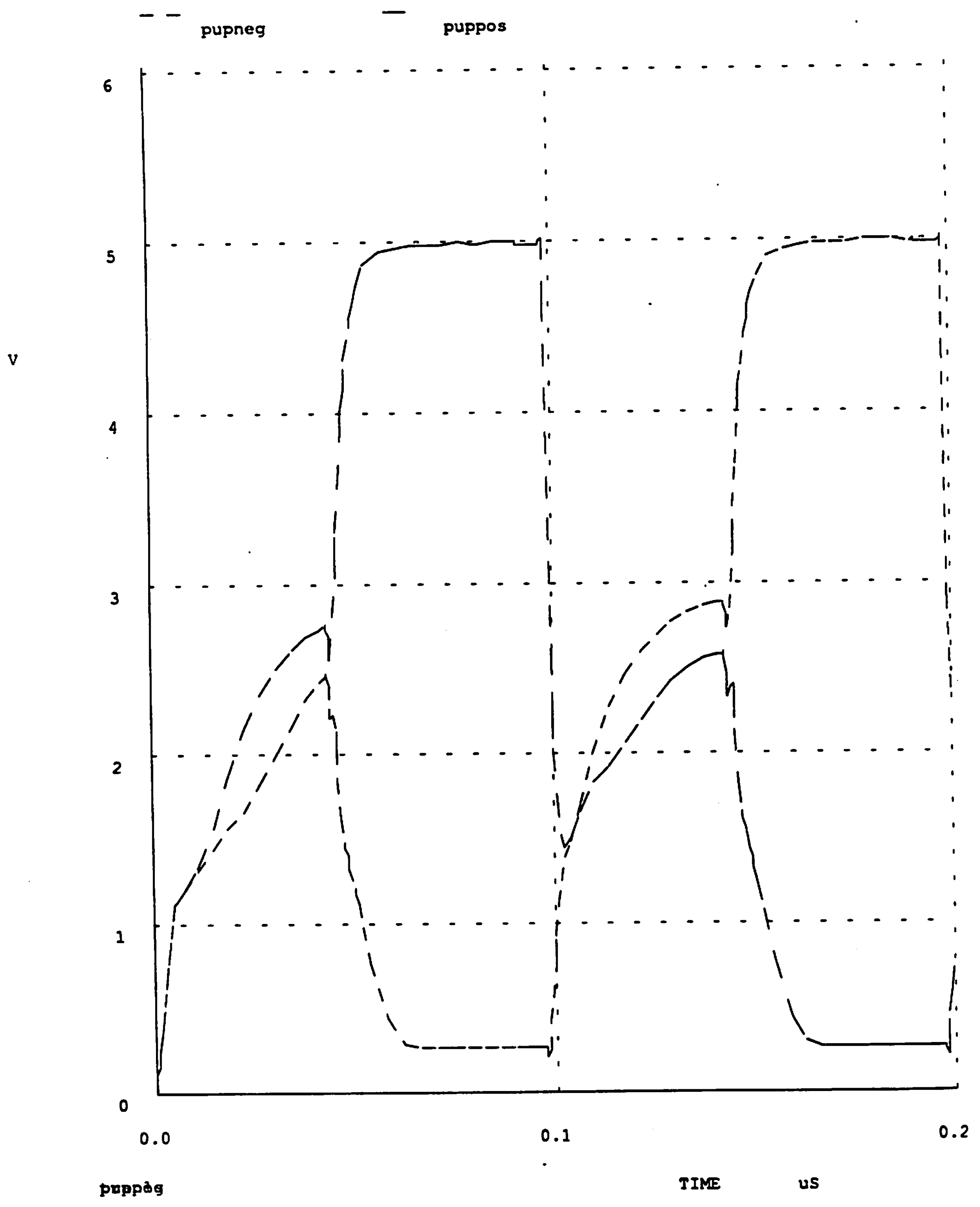
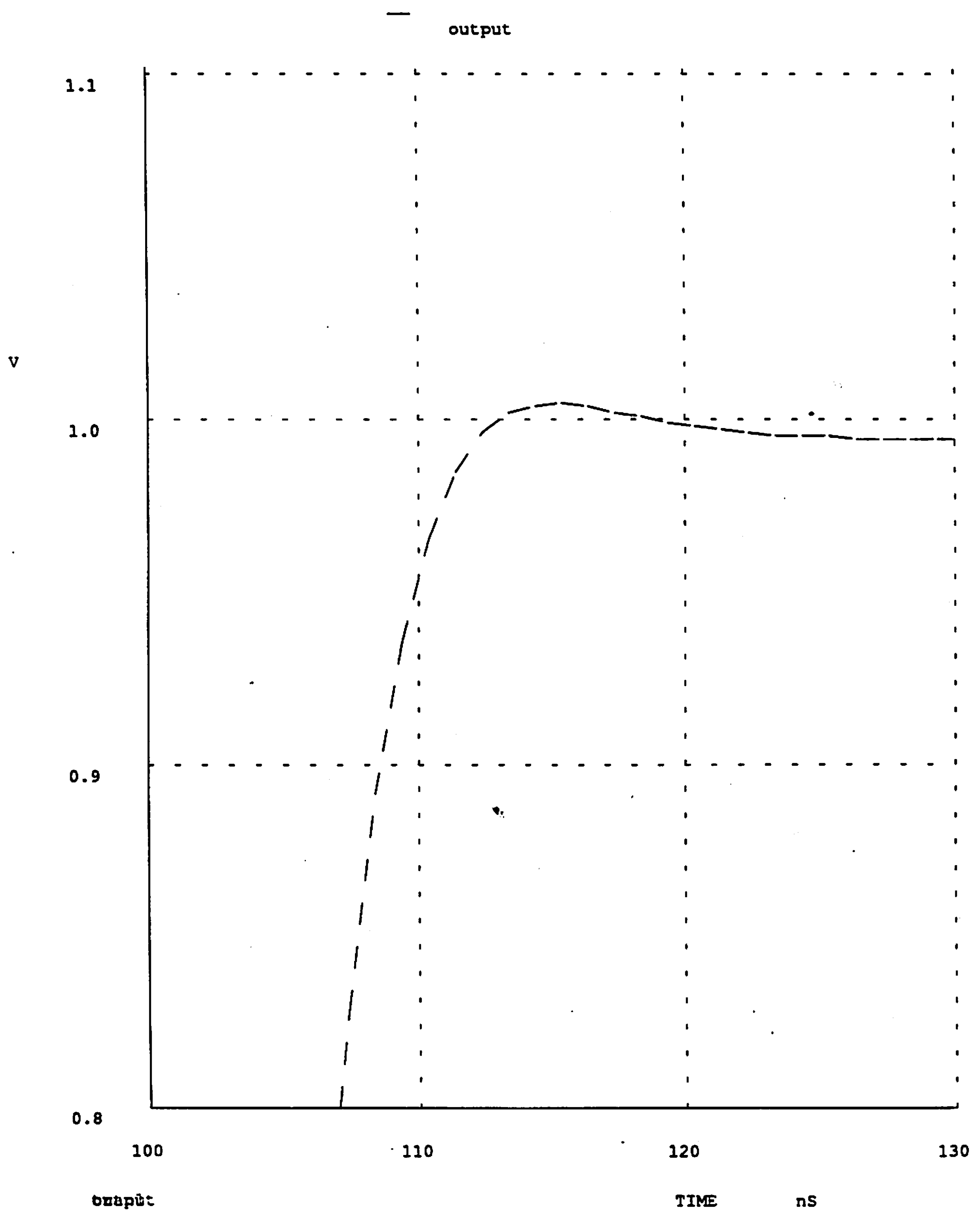


Figure 5.5: 0.9% Overshoot with 1.5pF Load



Chapter 6

A Guide to Testing

The A/D has been placed on a chip, shown in Figure 6.1, to be fabricated for testing purposes. In addition to the A/D, the chip contains duplicates of the first stage and of a single comparator. These circuits have many of their internal nodes connected to I/O pins so that their voltages can be easily measured. These test circuits will not be able to function at high speeds because of the additional parasitic capacitance from the pads and pins. They should, however, function properly at slow speeds. The tester should be able to to determine whether the opamp, DAC, and comparators work at low speeds. This will be helpful for locating any circuit faults, and will make it easier to determine correct bias currents.

Also on the chip are two clock generators, one for the A/D and one for the test circuits. The A/D and test circuitry have been isolated from each other as much as possible. The test circuits draw power from the same wires as the A/D's digital section, and both clock generators share the same input. Otherwise, all bias lines, inputs, and outputs are separate for the A/D and the test circuits.

6.1 A/D Biasing and I/O

The A/D requires four bias currents and three reference voltages. Figure 6.2 shows the on chip bias circuitry. The cell containing these transistors is on the left side of the A/D layout (see Figure 3.16). Two of the bias currents are for the opamps. The currents do not go directly to the opamps: they go to current mirrors which cause currents of equal magnitude to go to the four opamps. (See Figure 3.1 for the biasing within each opamp.) Thus, only one pair of bias currents needs to be supplied to the A/D from off chip, instead of four pairs of currents. It is important to realize that the current mirrors change the direction of the off chip bias currents. Ordinarily, one biases an nfet by putting a positive current into its gate and drain node. In this case, the opamp nfet's are biased by the pfet's of the current mirror, which are biased by a negative current from off chip. Likewise, the opamps' pfet's are biased by nfet's in the current mirror which receive a positive current from off chip.

The two comparator bias currents feed simple diode connected devices. The *pbias* biases the pullup transistors, and the *csbias* feeds the current sources.

During spice simulations, *pbiasiin* and *nbiasiin* were set to $60\mu A$, and *csbias* set at $70\mu A$. *Pbias* was $50\mu A$. These would be reasonable values to use during testing. Increasing *csbias* will increase the speed of the comparators at the cost of common mode input range and power. When *csbias* is set, *pbais* should be adjusted to keep the nodes *pup+* and *pup-* close to the midpoint between the power supplies. (These voltages can be measured directly since these two nodes on the test comparator are connected to I/O pins.) According to the spice simulations, the voltage on *pbias* should be close to 0.0 V. Care should be taken not to let the *pbias* current create a negative voltage on chip. It may be practical to simply tie the *pbias* input to the 0 V supply.

Increasing the opamp bias currents will increase speed and power, while reducing common mode range and opamp gain. It is not vital that both opamp bias currents be equal in magnitude, but if the common mode input range is centered midway between power supplies (ie. *ACGND* is equal to 2.5 V) it would be reasonable to set them equal.

The three reference voltages are the positive and negative references for the resistor strings, and *ACGND*. *ACGND* is the "ground" supply for analog signals. Since the power supplies are +5 V and 0 V, *ACGND* should be 2.5 V. The resistor string voltages should be centered on *ACGND*. Their recommended values are 3.3

V and 1.7 V, ie. +0.8 V and -0.8 V with respect to *ACGND*. These values can be altered to meet the actual output range of the opamps.

The other two analog inputs to the A/D are, of course, the differential voltages to be converted into digital. Since the input sample and hold has a gain of four, their maximum differential voltage without overflow should be +/-0.2 V, assuming the reference voltages are +/-0.8 V.

The A/D has ten digital outputs: 9 bits plus an overflow signal. Because of the piplining, there is a two clock period delay between inputting a voltage and producing a digital output. Because the input to the error correction circuits is latched, the output is valid during both phases of the clock period.

6.2 Test Section Biasing and I/O

Like the A/D, the test circuits require four bias currents. The two comparator currents *pbiastest* and *csbiastest*, are just like their counterparts in the A/D. The two opamp bias currents, *pbiasiintest* and *nbiasiintest*, however, go directly to the bias circuitry in the opamp core shown in Figure 3.1. Thus, **these currents are in the opposite direction from the A/D opamp bias currents.** There is no bias current mirror in the the test section. It doesn't need one: there is only one opamp. The bias cell in the test section only contains the comparator bias transistors shown in Figure 6.2.

The reference voltages *VRef+test*, *VRef-test*, and *ACGNDtest* are all analagous to their counterparts in the A/D. The inputs *in+test* and *in-test* go to both the input of the opamp and the input of the comparator core.

The test section is attached to six other analog pins of special note. Two *oaoutnegtest* and *oaoutpostest* are the outputs of the opamp. The large capacitive load of the pads and pins will greatly curtail the speed of this opamp circuit. However, the opamp should function normally at slow speeds. The outputs of the DAC of the test stage are also attached to pads at nodes *DAC+test* and DAC-test. Finally, the nodes *pup+* and *pup-* in the test comparator are brought off chip. These nodes can be used to determine the proper bias currents for the comparators, and

Listing of Pads on North Side from West to East		
NODE	TYPE	COMMENT
Vdd	Vdd	Analog
GND	GND	Analog
ACGND	unbuffered	for A/D
in+	unbuffered	for A/D
in-	unbuffered	for A/D
Vref+	unbuffered	for A/D resistor string
Vref-	unbuffered	for A/D resistor string
nbias iin	unbuffered	for opamp, check direction!
pbias iin	unbuffered	for opamp, check direction!
csbias	unbuffered	for comparator
pbias	unbuffered	for comparator
DAC-test	unbuffered	from test stage

Table 6.1: Pads on North Side of Chip

to confirm that the comparators make decisions properly. Like the opamp in the test section, the speed of the comparator will be drastically reduced by parasitics.

The test circuits generate nine digital outputs. Three are the bits generated by the test stage. The other six are the clock phases from the test section's clock generator. Since this cell is also new, it would be useful to confirm that it functions, and to characterize the delay between phases.

6.3 Future Work

The immediate task is to examine the test chip being fabricated. It may be possible to enhance the speed of the A/D by reducing the size of the opamp capacitors. These capacitors were designed conservatively large, since their values are not known with a high degree of certainty. They are not made from special capacitor layers, so their values may vary from run to run, or might be different than specified by the extract section of the magic technology file. If the capacitors are too small, charge injection will bring the signals outside of the common mode range of the opamps. If the capacitor sizes do turn out to be consistent and large,

Listing of Pads on West Side from North to South		
NODE	TYPE	COMMENT
pbiastest	unbuffered	for test comparator and stage
csbiastest	unbuffered	for test comparator and stage
Vref-test	unbuffered	for test stage
Vref+test	unbuffered	for test stage
pbias iintest	unbuffered	normal direction!
nbias iintest	unbuffered	normal direction!
ACGNDtest	unbuffered	
in-test	unbuffered	for test comparator and stage
in+test	unbuffered	for test comparator and stage
oaout negtest	unbuffered	
oaout postest	unbuffered	
pup+test	unbuffered	internal node of test comparator
DAC+test	unbuffered	DAC- on opposite corner of chip

Table 6.2: Pads on West Side of Chip

Listing of Pads on South Side from West to East		
NODE	TYPE	COMMENT
pup-test	unbuffered	internal node of test comparator
phi2primetest	out	test clock
phi2test	out	test clock
phi2bartest	out	test clock
phi1primetest	out	test clock
phi1test	out	test clock
phi1bartest	out	test clock
Vdd	Vdd	digital
GND	GND	digital
I2test	out	test stage MSB
I0test	out	test stage LSB
I1test	out	test stage Middle Bit

Table 6.3: Pads on South Side of Chip

Listing of Pads on East Side from North to South		
NODE	TYPE	COMMENT
bit0	out	A/D output
bit1	out	A/D output
bit2	out	A/D output
bit3	out	A/D output
bit4	out	A/D output
Vdd	Vdd	digital
GND	GND	digital
bit5	out	A/D output
bit6	out	A/D output
bit7	out	A/D output
bit8	out	A/D output
OF	out	A/D overflow output
clock	in	master clock for A/D and test

Table 6.4: Pads on East Side of Chip

it should be possible to reduce their size, decreasing the load on the opamps, and thereby decreasing settling time.

Another interesting project would be to use individual stages of the A/D as tiles for an automatic layout generator to create a smaller A/D. For example, the first stage by itself is a three bit A/D. Two additional cells, *stage1* and *stagenot1*, have been designed for this purpose. *Stage1* should be used as the first stage, since it contains the non-subtracting opamp, and the bias generator. If fewer than four total stages are used, the extra bias lines should be tied to Vdd or GND. Subsequent stages should be instances of *notstage1*. These two cells are similar to the stages within the nine bit A/D, but have been wired and labeled to be easily used as macrocells. Within the A/D, the interstage wiring is more compact, but not as convenient to manipulate. Figures 3.12, 3.13, and 3.14 provide the terminals present on these cells, as well as their connectivity and timing.

Figure 6.1: Chip Layout

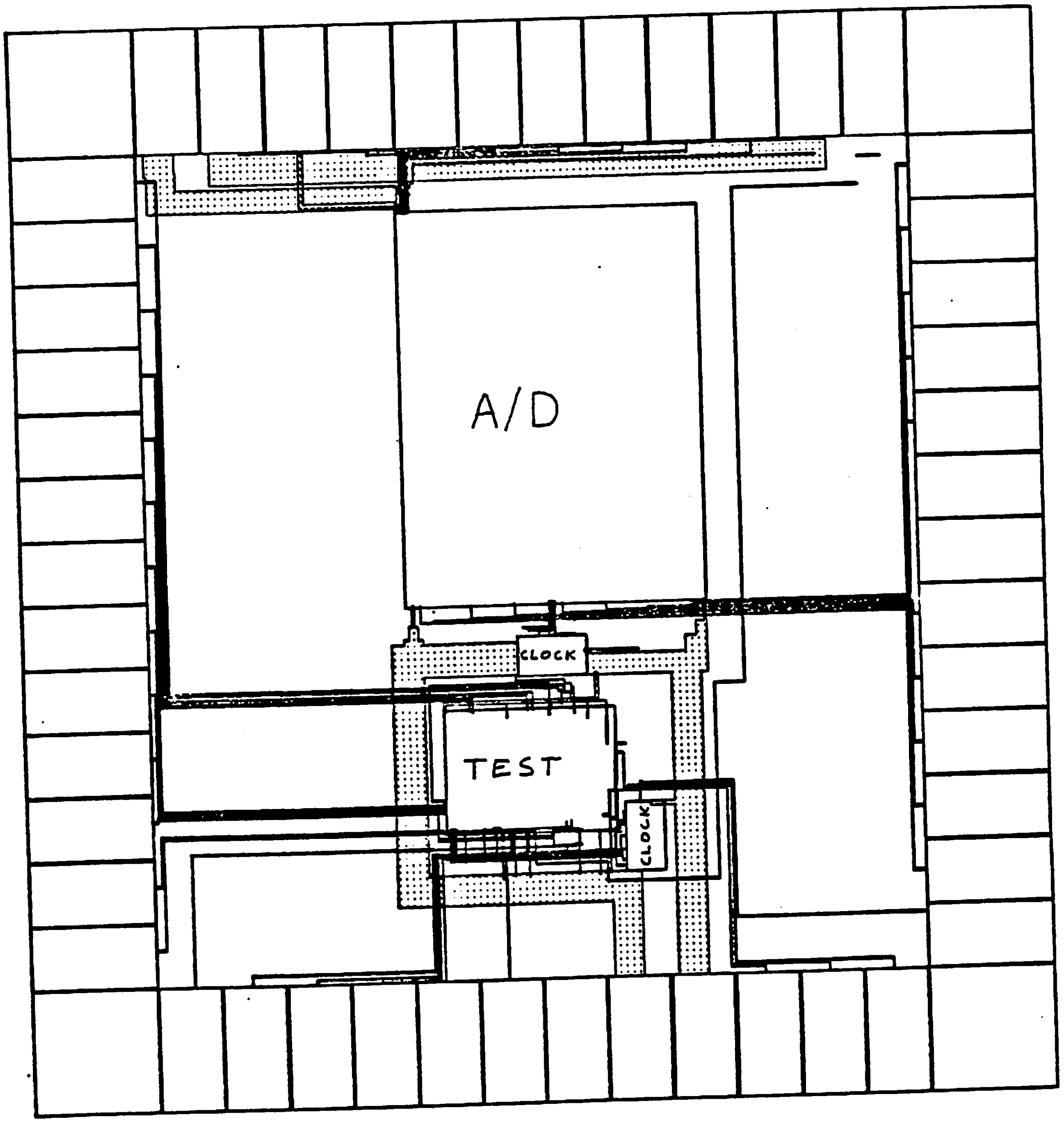
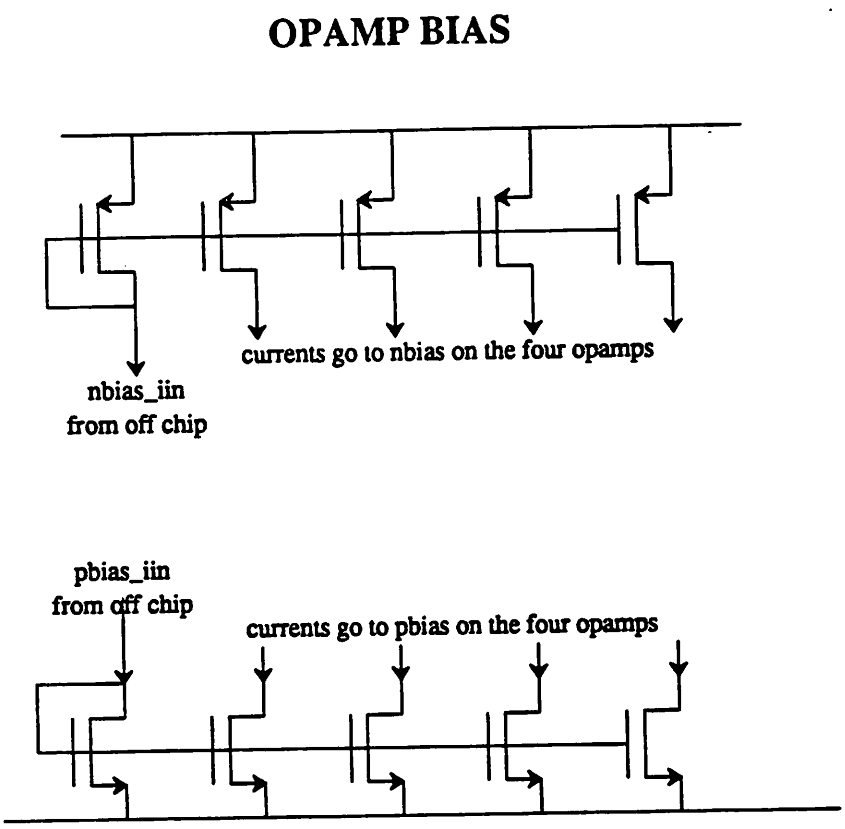
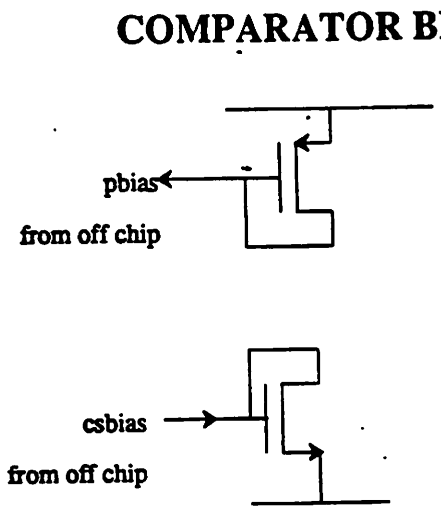


Figure 6.2: Bias Circuits in A/D


OPAMP BIAS
currents go to nbias on the four opamps
nbias_iin
from off chip
pbias_iin
from off chip
currents go to pbias on the four opamps



Chapter 7

Conclusion

This paper describes two accomplishments. The first is a program, ext2spice. Ext2spice allows circuit designer to extract and simulate circuits with greater accuracy and ease. It includes more detailed circuit information and provides a more structured output than do other commonly used programs.

The second is the layout of a nine bit A/D. This layout has been simulated with the aid of ext2spice. It is small enough to be easily included on a chip with other cells.

Bibliography

[1] S. H. Lewis, P.R. Gray, "A Pipelined 5 MHz 9b ADC",1987 IEEE International Solid-State Circuits Conferences, New York, NY, pp. 210-211, February 1987.

[2] S. H. Lewis, PhD Thesis, Not Yet Published.

[3] R. Castello, P. R. Gray, "A High-Performance Micropower Switched-Capacitor Filter", IEEE Journal of Solid State Circuits, pp. 1122-1132, December 1985.

[4] Hauser, Max, Unpublished Work.

Appendix A

File Index

All files related to this report are stored within the brodersuns file systems. All directory locations are given relative to the author's home directory, currently */usr3/rwb/burstein*.

Magic

All magic cells are found in the *magic* directory, or one of its subdirectories. The full list of paths to these directories can be found in *.magic* or *.magic.eros* in the home directory. The top level cell for the test chip is *magic/chip.mag*; for the A/D macrocell, *magic/a2d2.mag*. The clock generator is *magic/delayclock.mag*. The two self standing stages are *magic/stage1.mag* and *stagenot1.mag*.

Ext2spice

Source code for ext2spice is in *ext2spice/ext2spice.c ext2spice/ext2spice.h* and *ext2spice/cells.c*. Executable code is in *ext2spice/ext2spice* and additional documentation in *ext2spice/ext2spice.doc*.

A magic technology file is located in *magic/tech/scmos.tech21*. This contains extract styles for using ext2spice with $3\mu m$ and $1.6\mu m$ layouts, as well as corrected versions of the old $3\mu m$ and $1.6\mu m$ extract styles.

Spice

All spice files are in the *magic/spice* directory. For each simulation, there is an input *.spice* file and a nutmeg rawfile, appended *.out*.

The ext2spice definitions file used to make all of the spice files is *magic/a2d.defs*.