Dept. of Informatics, University of Oslo

Analog Correlators for CMOS Signal Processing

Cand. Scient. Thesis
February 1993
John Arne Nesheim
Dept. of Informatics, University of Oslo

Preface

This is the thesis for my Cand. Scient. degree at Dept. of Informatics, University of Oslo.

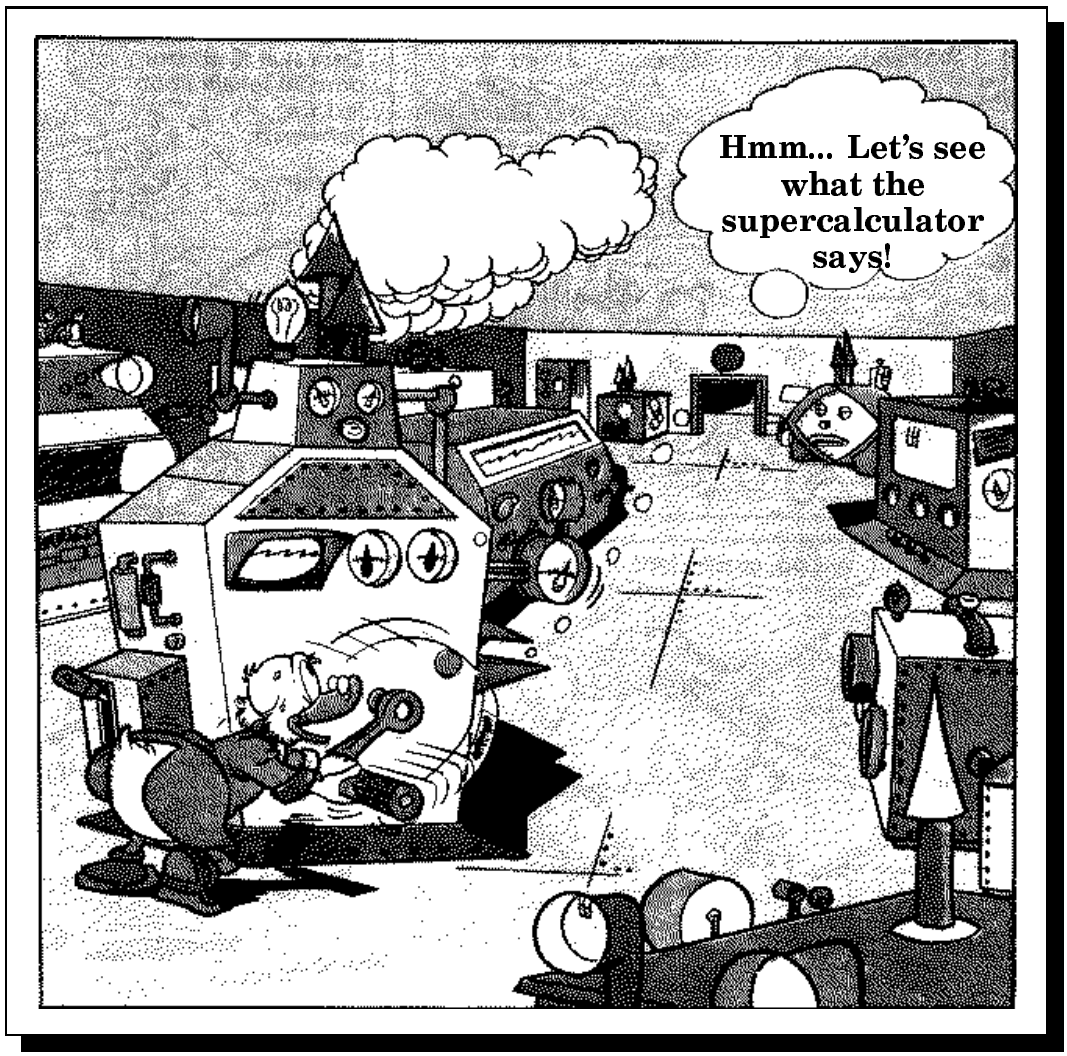
The time I have spent writing this thesis has been interesting and challenging. Since my background is mostly digital a lot of the "Analog way of thinking" and the analog CMOS technique was new to me. The work has varied from designing the layout of the chip to testing the fabricated chip in our laboratory.

I would like to thank my supervisor Tor Sverre Lande and Yngvar Berg for all the help during the writing of this thesis. Morten Salomonsen has also given a lot of hints, both scientific and non-scientific. Finally I thank my parents, they have always encouraged me in my education.

Blindern, February 1993
John Arne Nesheim

Contents

Figures



An analog calculator?

1
Introduction

In this chapter I will discuss the use of the CMOS transistor as a digital switch and as an analog device. I will mention some circuits using this analog technique. Finally I will shortly describe the intentions of this work, and describe the rest of this thesis.

1.1 Digital systems

Many of the computers we use today have a microprocessor constructed with the CMOS (Complementary Metal Oxide Semiconductor) technology. CMOS is a technology building on the principle of a surface field-effect transistor. This was first mentioned in the early thirties by Lilienfield and Heil. Shockley and Pearson made further studies on the subject towards the end of the fourties. In 1960 the first MOSFET-circuit was produced, using a termic oxide silicon structure. Since the end of the sixties we have had a rapid development regarding both size and speed. Several types of semiconductors have been used (Ge, Si and Ga As), and there are also different insulators (SiO_2, SiN_4 and Al_2O_3). Most widely used is the combination Si and SiO_2.

In traditional digital circuits the CMOS transistor has been looked upon as a switch. This is of course a nice way of using the transistor in digital systems, were we have just two values of interest: on/off, 0/1, 0V/5V etc., which is easily realized with switches.

As an example, let us look at a simple inverter. It has one input (V_{in}) and one output (V_{out}). In CMOS we can construct an inverter by using two transistors, the diagram is shown in figure 1.1. The same figure also shows a simulation of this inverter. In this circuit both the p- and n-transistor works as switches. We are usually thinking in terms of voltages in these kinds of circuits, and we can see that both transistors can not be open at the same time. Thus there will not be any current floating through the

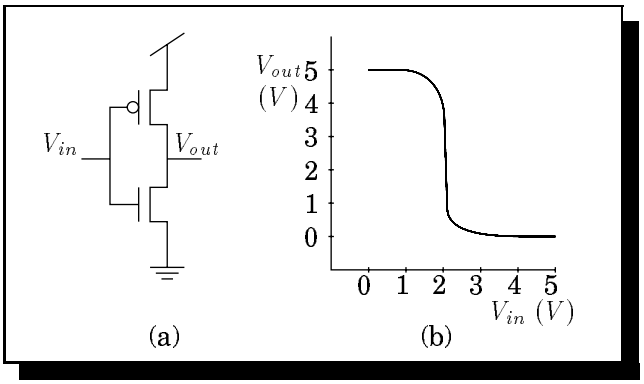


Figure 1.1: Diagram and simulation of an inverter

In (a) we have a transistor diagram of a standard digital inverter, and (b) shows a simulation of the same circuit. Since only one of the transistors is open at a time, there is no current going through the circuit when V_{in} is stable at $0V$ or $5V$. The only time there is any current is when there is a change of state in the circuit, that is, when $V_{in} \approx 2.5V$ (the steep part of the curve in (b)). This is why CMOS circuits has a quite low power dissipation compared to for example a TTL circuit.

transistors as long as the circuit does not change state. Briefly the circuit works as follows: for $V_{in} = 0V$ only the p-transistor is open, thus leading V_{out} to V_{dd} or $5V$. In the opposite case ($V_{in} = 5V$), the n-transistor is the open one, and V_{out} is set to $0V$ (or Ground). We say that a transistor is open as long as the gate-source voltage is higher than or equal to some *threshold-voltage*. Otherwise the transistor is closed.

Unfortunately, very few (or none) of the physical problem we are facing are digital. We are living in an analog world with digital instruments. One clear example showing this is the CD-player. It plays a CD-record (which has the sound digitally stored, after converting from analog to digital during recording), and converts the digital signals to analog signals feeding the amplifier. Thus we are making a double conversion, first we go from analog to digital, then back to analog again. The reason for doing all this is of course the quality of the digitally stored signals. It has almost no noise, giving a sound quality so high that we think it is perfect. Several other examples can be found, all of them underlining the fact that digital systems have helped us solving problems.

1.2 What is analog CMOS?

As mentioned earlier we have learned that the MOS transistor has a threshold-voltage, and as long as the gate voltage is below this voltage the transistor is closed. But the MOS transistor has qualities which are not used in pure digital circuits. It is *not* a switch, on the contrary it lets a small but well-defined current through itself when

operated below the threshold-voltage. (This current is so small that we usually treat it as zero in digital systems.) How we define this current in terms of mathematical equations is one of the subjects treated in the next chapter. So this is what we mean by the concept of analog CMOS; to use the CMOS-transistor in the sub-threshold region, or in *weak inversion*.

This technique can be used to construct circuits which have remarkable power of computation compared to their sizes and complexity. Another advantage with these circuits are the extremely low power dissipation, since the currents flowing are so small.

1.3 Human senses

In the book "Analog VLSI and Neural Systems" Carver Mead [13] gives some examples of how to use the CMOS transistor in weak inversion. I will mention two of these circuits here, the "Silicon retina" and the "Electronic cochlea". These are two circuits trying to imitate human senses. The basic building blocks of these circuits are quite simple, and have some amazing qualities.

The first one, "Silicon retina", is a circuit imitating our the retina in our eye. Without any use of optical equipment, one lets light straight onto the silicon chip and gets a "seeing" chip. The circuit is produced in a standard CMOS process, and use normal transistors and bipolar photo-transistors. The way it works is so close to our retina that it can even be fooled by optical delusions!

The other circuit, "Electronic cochlea" ([11],[22]), is an imitation of a part of our ear, the cochlea. This is the part of the ear where the frequency decoding of sound is done. This is done by some physical arrangements, the circuit does it by using an electronic model, and find the different frequencies in the input signal. This circuit takes the analog signal as a direct input to the chip.

What both circuits have in common is the fact that they have quite small and simple building blocks, require no (or little) preprocessing of the signals to be analyzed, and they both imitate the human senses quite well. At present this seems to be very difficult to manage using digital techniques. For example, digital image processing is known to require very much computational power. In addition to these two circuits I have mentioned here, several other interesting circuits are described in the same book.

1.4 Some projects using analog CMOS

At Dept. of Informatics we are a group working with different circuits using this technique. Some are working with neural nets, trying to realize those by using analog CMOS. Nets like this require several smaller circuits, like multiplicators, UV-programmed weights, transconductance amplifiers. Especially the realization of the weights are of great interest. Several types of nets are treated. There are some works on the use of photo-transistors and processing circuits, analog memory (which we all can use in our projects...), UV-structures, etc. Combinations of analog and digital circuits on one chip are also tried out. This thesis is about one type of signal processing, and others are looking at other solutions.

1.5 About this report

1.5.1 General information

An analog signal can be said to consist of several frequency components. In many applications it is interesting to find what a signal really looks like, that is, what is its main frequency and which of the harmonics are present. (Something like this is happening in the human ear when we recognize different kinds of sounds.) In this work I have tried to construct an analog CMOS-circuit which looks at the frequency contents of an analog signal.

Circuits like this have been made, by first sampling the signal and then process the new signal (FFT). But there might be cases where we do not need the digital system like we did in the case of the CD player. We might just process the analog signal directly, without any conversion, and get a circuit which is smaller, but equally effective. If we can make a single chip to do this complex job, we can say that we use the properties of the CMOS transistor better than a digital circuit does.

An example of the use of this can be analysis (and synthesis) of sound (speech).We can for example imagine the frequency specter of different sounds recorded and stored (with the new analog memory element we now have), and continually compare these with the results from the the circuits and thereby recognize for example words. There is of course a lot of work to be done before we get that far. Another example where we could find it useful to find the frequency specter is in the examination of radiation from a comet. This radiation can tell us a lot about the contents of the comet.

In this work the circuit is not made for use on a particular type of signal, but it will accept a 'normal' analog signal and process it. The signals used in the testing of the circuit is taken from a signal generator.

1.5.2 The rest of the report

This first chapter gives a short introduction to the term analog VLSI, compares it with the digital VLSI technique, and gives some examples of how we can use analog circuits. The rest of this thesis is built up like this:
Chapter 2. Gives a short introduction to the CMOS transistor operation in weak inversion, and some small circuit examples.
Chapter 3. This is a review of the traditional spectral analysis, including Fourier transform, the FFT-algorithm, and some other techniques.
Chapter 4. Here we start the main part of this thesis. The correlator is the main computing element on my chip, and correlator circuits are treated in detail. There are many types of correlators, so simulations and measured results from different circuits are presented.
Chapter 5. Gives a detailed description of the circuit I have designed. I treat all the different circuits on the chip. As in the previous chapter I include both simulations and measured results from the different parts of the circuit.
Chapter 6. A vital part of this work is the testing. This chapter deals with the testing of the chip (not the small circuits) as a system. Testing of this kind is a lot more complicated then the testing in the two previous chapters, so the results are not always so promising.

Chapter 7. This chapter gives some test-results from another chip. A short description of this larger system is also included.
Chapter 8. The circuit produced is far from perfect. Here I describe the problems with the chip, and give suggestions to improvements and changes to get a better version.
Chapter 9. This is a conclusion, a look back at what we have done and what experiences we gained from it.
Appendix A. Gives a complete circuit diagram, and some additional information to the circuits.
Appendix B. Contains some information about the different programs and electronic equipment used during the work with this thesis.

2
Background

In this chapter I will shortly describe the CMOS transistor and its properties in the sub-threshold mode. I briefly describe some basic circuits, taken from the book "Analog VLSI and Neural Systems" by Carver Mead [13].

2.1 The CMOS transistor in weak inversion

2.1.1 The CMOS-transistor

The CMOS process use two types of transistors; the p- and n-type. The n-transistor has electrons as charge carriers, the p-type has holes. A transistor has three connections, the gate which controls the current through the transistor, the source and drain. There are, however, no principle difference between the source and drain on a single transistor, it is symmetric. For a transistor like the one shown in figure 2.1 we will in digital systems call the transistor open when $V_{gs} \geq V_t$. V_t is the threshold voltage, typically $0.8V$ for a normal process. For $V_{gs} < V_t$ the transistor is said to be closed. In this case we have no current going in the transistor, except from some small leakage current, which we usually can neglect in digital circuits.

2.1.2 The sub-threshold region

There is a current flowing through the transistor even when $V_{gs} < V_t$. This current is a well defined function, described by an exponential equation. We wish to see what really happens in this region of operation. If we examine the flow of charges we end up with the following equation, taken from [13]:

$$I = I_0 e^{\frac{-\kappa q V_g}{kT}} (e^{\frac{qV_s}{kT}} - e^{\frac{qV_d}{kT}}) \tag{2.1}$$

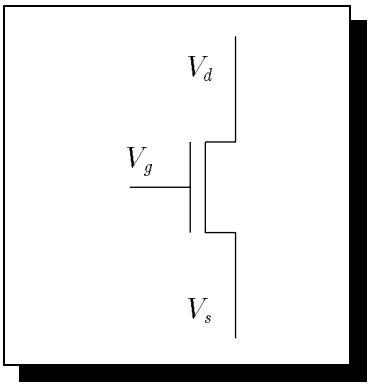


Figure 2.1: A simple n-transistor

V_g is the gate voltage, V_d is the drain voltage and V_s is the source voltage. The equation for the current I is given in equation 2.1.

Here we need an explanation of the different parts of the equation:

κ	:	kappa, compensating for reduced gate efficiency,
I	:	the current through the transistor,
I_0	:	a constant, including the width and length of the transistor,
q	:	electron charge, positive for p- and negative for n-transistors,
k	:	Bolzmanns constant,
T	:	the temperature,
V_g	:	the gate voltage,
V_s	:	the source voltage,
V_d	:	the drain voltage.

This is the basic equation for all further treatment of circuits.

2.2 Some basic circuits

When we operate the transistor in weak inversion there are some circuits that are more common than others. Many circuits used in analog CMOS constructions use the *current mirror* and the *differential pair* in various ways. The reader of this work should be familiar with these basic circuits and how they work, and I will only shortly mention them here.

2.2.1 Current mirror and differential pair

There are several occasions where we wish to make a copy of a current. A circuit doing this is shown in figure 2.2 (a), and a simulation is shown in (b). Thus, if we have a current I_1 as input the circuit will produce an output current I_2 so that $I_2 = I_1$. An important feature of this circuit is that we can make as many copies we want, there is no load on the input.

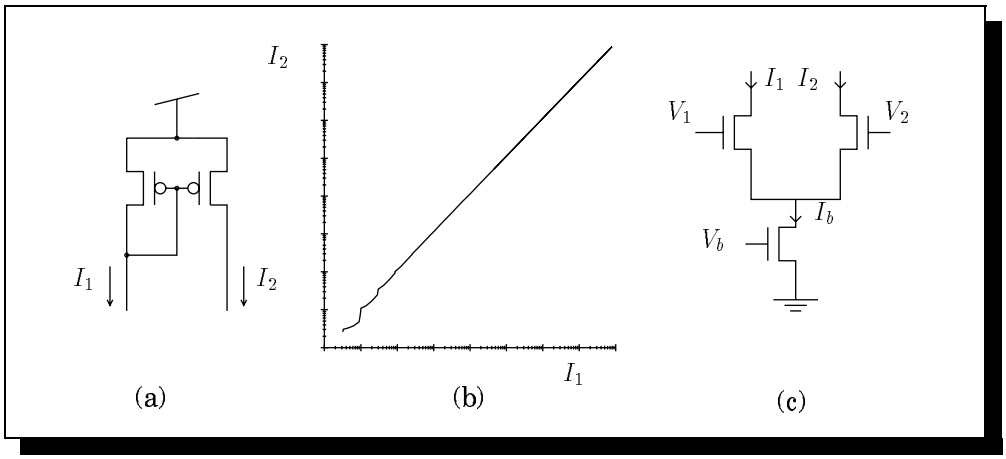


Figure 2.2: A current mirror, simulation and a differential pair

In (a) we have the diagram of a current mirror, and (b) shows a simulation of this circuit. I_2 is drawn as a function of I_1. Both axes have logarithmic scale, going from 10^{-13} to $10^{-6}A$. In (c) is a diagram of the differential pair.

The differential pair is a circuit taking the input as the difference between two values (voltages). Figure 2.2 (c) shows this circuit. V_b control the current I_b, and I_b is split in two parts I_1 and I_2 depending on the values of V_1 and V_2.

2.2.2 The transconductance amplifier

If we connect a current mirror on top of a differential pair as shown in figure 2.3, we get a transconductance amplifier. The equation for I_{out} is then given [13]:

$$I_{out} = I_1 - I_2 = I_b tanh \frac{\kappa(V_1 - V_2)}{2} \tag{2.2}$$

In this equation all voltages are given in units of $\frac{kT}{q}$. This circuit is often used, but we can also expand it to get a wide range amplifier with improved qualities. We will return to this circuit in chapter 5, which describes my circuits in detail.

2.3 Circuit deviations

When we calculate the current or simulate the behavior of a circuit we use an equation describing the current through the transistor, like the one given in equation 2.1. This equation is, however, only a simplified and theoretical description of the real situation. In a real transistor there are several effects that can make the world seem a bit more complicated.

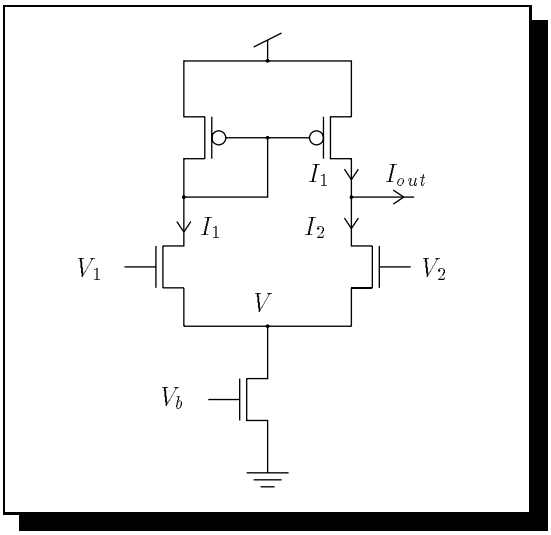


Figure 2.3: A transconductance amplifier

The output from the circuit is in this case a current and the equation for I_{out} is given in 2.2.

2.3.1 Gate efficiency and Body effect

Since the substrate of the chip is not intrinsic (pure), the gate voltage will not be able to control the current 100 percent effective. This is due to the charge stored in the channel between source and drain. In equation 2.1 the parameter κ (kappa) is introduced to compensate for this effect. This κ is multiplied with the gate voltage, thus reducing the gate efficiency. A typical value would be $\kappa = 0.6$. This parameter (κ) will also model the Body effect.

2.3.2 Early effect

The saturation current in a transistor is dependent of the drain-source voltage as well as the gate-source voltage. This is because the depletion region increases with increasing drain voltage. Thus we get a shorter channel and increased current. We may correct by introducing an Early voltage V_0 in the transistor equation. (See [13, Appendix B]).

2.3.3 Transistor mismatch

On a single chip the doping will vary considerably, and this will give different values of the I_0 for different transistors. So, for example a current mirror will generally not give us an exact copy of a current because of the difference in the two transistors.

2.4 Basic circuits on my chip

On the chip I have made, the delay line is a vital part. A delay line is made up of several transconductance amplifiers, I have used the wide range variant, with a larger amplification than the one shown in figure 2.3. This will be described in chapter 5 together with the description of the rest of the chip.

3

Signal processing and spectral analysis

In this chapter I will briefly describe what we understand with the term spectral analysis , and discuss some aspects of this field. There are written several books about this topic, I have included some of it here to give an idea of what my circuit is supposed to do.

3.1 Signal processing

All types of signals, either analog or digital, for which we build circuits or systems to process, carry some information. In general we say that signal processing is the work done to make this information visible and understandable for us. There are numerous examples of this, take for example TV- and radio-transmissions, radar, sonar, the treatment of cosmic radiation etc.

In this work I try to construct a circuit which can tell us something about the frequency spectra of analog signals. This is a vital part of the signal processing field, and several books treat this topic. Digital systems that perform this task have been made, but the use of the CMOS transistor in weak inversion and having one chip doing all the work is a new approach.

3.2 Spectral analysis

3.2.1 A short review

A signal may be represented in two ways. The way we usually like to see signals is in the *time* domain (as on an oscilloscope), but there are also occasions where we want the signal to be represented in the *frequency* domain. The transformation from one of these representations to the other is known as the *Fourier transform*.

For a periodic signal (function) we can represent the function $f(t)$ by an infinite sum (taken from [3]):

$$f(t) = \sum_{-\infty}^{\infty} c_k e^{jk\omega_0 t} \tag{3.1}$$

where:

$$c_k = \frac{\omega_0}{2\pi} \int_{-\pi/\omega_0}^{\pi/\omega_0} f(t) e^{-jk\omega_0 t} dt \tag{3.2}$$

These equations tells us that every function (or signal) can be described by a sum of sine-wave-functions (since $e^{-jk\theta} = cosk\theta + jsink\theta$). Equation 3.1 is the complex Fourier series of $f(t)$.

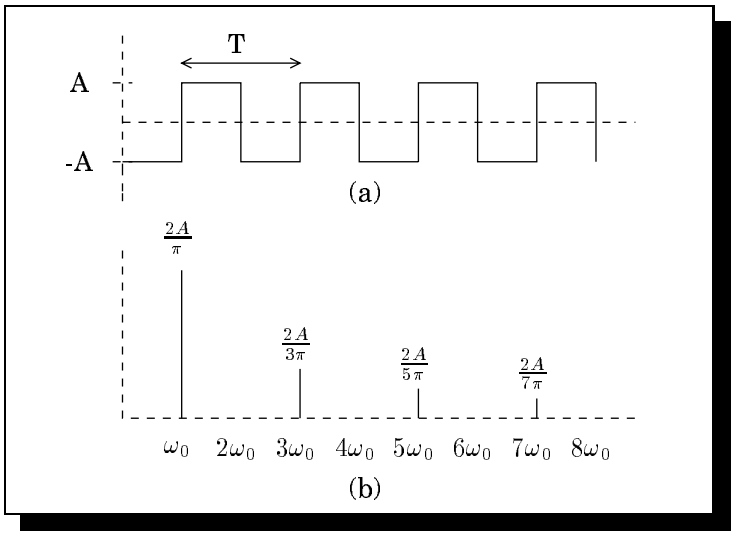


Figure 3.1: The amplitude spectrum of a square wave

In (a) the input is a square wave, with period T and amplitude A. Part (b) shows the frequency (amplitude) spectrum of this function. We have that $c_k = \frac{-j2A}{\pi k}$, so the amplitude spectrum is a plot of the values of $|c_k| = \frac{2A}{\pi k}$.

In equation 3.1, ω_0 is the fundamental frequency (in rad/sec). Then $k\omega_0$ denotes the kth harmonic. So, the expansion of a function in a Fourier series is a harmonic analysis of

$f(t)$. It will result in a transformation from the time domain to the discrete frequency domain. As an example we can look at the square wave input function, shown in figure 3.1 (a). Using equations 3.1 and 3.2 we find that $c_k = 0$ for k even, and $c_k = \frac{-j2A}{\pi k}$ for k odd. Thus we get the amplitude spectrum as in figure 3.1 (b), where $\omega_0 = \frac{2\pi}{T}$.

One can expand this theory to non-periodic functions (signals) as well. If the period of the signal goes towards infinity, that is $\omega_0 \rightarrow 0$, the signal will become non-periodic. We then get the Fourier transform pair:

$$f(t) \longleftrightarrow F(j\omega) \tag{3.3}$$

where:

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega}dt \tag{3.4}$$

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(j\omega)e^{j\omega}d\omega \tag{3.5}$$

The frequency domain is no longer discrete, we have a continuous function for the spectrum of $f(t)$ ($|F(j\omega)|$).

By applying these equations we can get information about the frequencies in almost any signal. This technique can also be used for digital signals, with sampled values as input. A chip for performing the FFT on a signal has been made earlier as a Cand. Scient. thesis [18].

3.2.2 The FFT-algorithm

If we want to do a digital spectrum analysis, the most common used technique is the FFT algorithm. To understand how it works we must take a closer look at the DFT operation, the discrete Fourier transformation. Since this is an operation on signals represented digitally, the signal is described by a sequence of sampled values. The general Fourier transform will still work, but as we no longer have a continuous signal we can not find the c_k's by integrating, thus we must use a summation instead. Since we no longer work with continuous signals, but sequences of sampled values I use other names for the coefficients.

We have a sequence of samples $f(nT)$, this is the sampled original signal with N samples with time interval T. We can express this sequence as a sum of sinusodial sequences with different frequencies, each having N samples. For N odd we have these equations:

$$\{f(nT)\} = \{F(0)\} + \sum_{k=1}^{\frac{N-1}{2}} 2|F(k)|cos[2\pi\frac{k}{NT}nT + \Phi(k)] \tag{3.6}$$

where $F(k)$ is the Fourier coefficients:

$$F(k) = \frac{1}{N}\sum_{n=0}^{N-1} f(nT)e^{-j\frac{2\pi}{NT}knT} \tag{3.7}$$

In equation 3.6 we also get the phase of each frequency component, $\Phi(k)$, as a separate part. For N even some differences in the equations will appear (because of the properties of the Fourier coefficients), see for example [23] for details.

So, for a sampled signal with sampling interval T and N samples, we get the following

from a DFT:

- Lowest frequency : $\frac{1}{NT}$.
- Frequency resolution : $\frac{1}{NT}$.
- Highest frequency : $\frac{N-1}{N}\frac{1}{2T}$, N odd, $\frac{1}{2T}$, N even.

If we apply this method on a non-periodic signal we must look at a part of the signal by using a *window* function of some kind. This technique can also be used to get better results for some signals.

The DFT works properly, but have a large number of operations to reach the result. In fact we must perform $2N^2$ multiplications. This leads to the use of FFT algorithms. There are several such algorithms, all of them computing the DFT in a more efficient way than the DFT itself.

As an example we can take the Cooley-Tukeys algorithm (from [23]). Very briefly this algorithm splits the given sequence in two parts, and treat each of these parts as a new sequence. This splitting is repeated until we have N sequences with one sample. To compute the DFT for these one-sample sequences is easy. For this algorithm to work, the number of samples can not be chosen free. It must be a power of two, $N = 2^P$. We split the original sequence into one sequence with all the odd-numbered items, and one with all the even-numbered.We then get the following relations:

$$F(k) = \frac{1}{2}[F(k)_{even} + F(k)_{odd}e^{-j\frac{2\pi}{N}k}] \tag{3.8}$$

$$F(k+\frac{N}{2}) = \frac{1}{2}[F(k)_{even} - F(k)_{odd}e^{-j\frac{2\pi}{N}k}] \tag{3.9}$$

Using these formulas we can write recursive programs to compute the FFT sequence. With this new approach the number of multiplications is reduced from $2N^2$ to $2Nlog_2N$, that is , the ratio between DFT and FFT is $\frac{N}{log_2N}$ in number of multiplications. For N=8192 for example, the ratio will be about 630.

3.3 Correlation

The correlation function can be used to find similarities between signals. All equations in this section are taken from [3]. The function is defined by:

$$\rho_{ff}(\tau) = \int_{-\infty}^{\infty} f(t)f(t+\tau)dt \tag{3.10}$$

where $f(t)$ is a finite energy signal, and τ is a time shifting parameter. This function gives the auto-correlation of $f(t)$. In the same way we define the cross-correlation of two signals $f(t)$ and $g(t)$:

$$\rho_{fg}(\tau) = \int_{-\infty}^{\infty} f(t)g(t+\tau)dt \tag{3.11}$$

If we want to find out if there are any similarities between two signals we use the cross-correlation function with τ as a scanning parameter. ρ_{fg} will reach a maximum for $\tau = x$ when the two signals have the largest similarity (unless $\rho_{fg} = 0$ for all τ, then $f(t)$ and $g(t)$ are uncorrelated). This can be useful when we analyze random signals.

It can be shown that the correlation functions and the energy spectrum of the signals form a Fourier transform pair:

$$\mathcal{F}[\rho_{ff}] = \mathcal{E}(\omega) \tag{3.12}$$

and:

$$\mathcal{F}[\rho_{fg}] = \mathcal{E}(\omega) \tag{3.13}$$

where $\mathcal{E}(\omega) = |F(\omega)|^2$.
Since $|F(\omega)|$ will show us the spectrum of the function $f(t)$, the Fourier transform of a correlation function will give us information about the spectrum of $f(t)$ as well.

The FFT and DFT algorithms can be used to calculate the correlation functions for sampled signals. These are known as Fast Correlation algorithms.

4

Correlation and correlators

In this chapter I will first talk about correlation in general, then describe some different types of correlation circuits (correlators). Simulations, measured results and theoretical calculations are included. The correlator is the basic calculating element on the chip I have designed.

4.1 What is correlation?

The word correlation can have different meanings, depending on the context. In an encyclopedia we find the following general definition:

correlation: If we measure the value of two variables, it can be of interest to find if the two variables have a similar behavior. That is, we have positive correlation when an increase in one variable corresponds with an increase in the other. There are for example positive correlation between human body-height and -weight. Negative correlation occurs when an increase in one variable corresponds with a decrease in the other. We have negative correlation between age and speed of adult humans [2].

4.1.1 What do we need ?

In chapter 3, I mentioned the correlation function used in signal processing. But in this work we will use transistor circuits to perform the processing on the signals. These circuits will usually not do exactly the same operations as ideal mathematical models. On the other hand we can make powerful circuits with only a few transistors, while

more accurate models will require large circuits. The question is then if we can use the power we get from small and simple circuits, but still manage to handle quite difficult problems. In this work this computing is done by a correlator. In our case this is a circuit that tells us if two signals are in phase (e.g. have similar behavior). This is why we call it a correlator, it performs a computation like the one described in the definition above.

4.2 Correlators

4.2.1 Circuit solutions

A correlator is a circuit which give us information about how two signals corresponds with each other. These circuits will not give us the same output as a "perfect" correlator in the mathematical sense will do, but the circuits will give us an output current that changes with the correlation between the inputs. We shall see that we can build both small and powerful circuits containing only a few transistors.

There are several variants of the correlator. I will start with the simplest form, then take the version I use on my chip, and finally describe some other alternatives.

4.2.2 About the layout

A factor that is not considered in the following is how to scale the different transistors in the circuits. In general there is a $\frac{W}{L} = 1$ on all transistors[1]. This may not be the optimal solution, but will be good enough in this prototype.

Since the chip had more than enough area for the layout of my circuits, I have not tried to make the circuits as small as possible. Generally they are drawn quite straight forward, without any optimization.

4.3 The asymmetric simple correlator

4.3.1 Description

The simplest form of a correlator is four transistors connected as shown in figure 4.1. This correlator has n-transistors. The correlators I use have p-transistors, but this does not influence the function of the correlator. All the equations in this section assume that we operate in weak inversion. I have no simple correlator on my chip, so the measured results are taken from another chip.

Let us take a look at the simple correlator, what we want is to express I_{out} as a function of I_1 and I_2. We start with the basic equation for a current through a transistor, (given in equation 2.1):

$$I = I_0 e^{\frac{-\kappa q V_g}{kT}} (e^{\frac{qV_s}{kT}} - e^{\frac{qV_d}{kT}}) \tag{4.1}$$

[1] Width to length ratio.

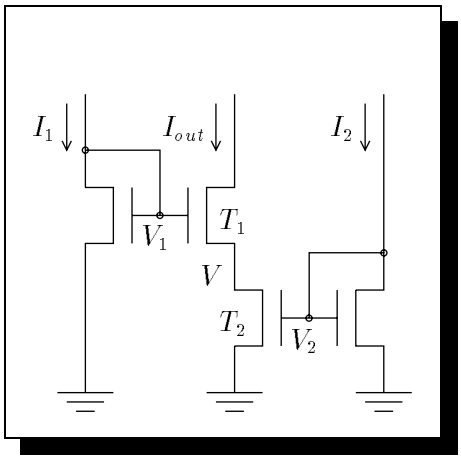


Figure 4.1: Asymmetric simple correlator

This is the simplest correlator circuit, containing only four transistors, but performs a remarkable powerful computation. The equation for I_{out} is given in 4.10.

Since we have n-transistors the charge (q) is negative, and assuming all voltages in units of $\frac{kt}{q}$, we can reduce equation 4.1 to:

$$I = I_0 e^{\kappa V_g}(e^{-V_s} - e^{-V_d}) \tag{4.2}$$

The two input transistors are operating in saturation, since they both are connected as diodes, that is, $V_s = 0$ and $V_d = V_g$. The equations for I_1 and I_2 as functions of the voltages V_1 and V_2 is then given:

$$I_1 = I_0 e^{\kappa V_1} \tag{4.3}$$

$$I_2 = I_0 e^{\kappa V_2} \tag{4.4}$$

We can now find two expressions for I_{out}, one for each of the transistors T_1 and T_2. If we assume the drain voltage of T_1 to be high, we know that T_1 is saturated. The equation for I_{out} is then given:

$$I_{out} = I_0 e^{\kappa V_1 - V} \tag{4.5}$$

We solve for e^{-V}:

$$e^{-V} = \frac{I_{out}}{I_0 e^{\kappa V_1}} \tag{4.6}$$

We can not assume that the T_2 transistor is saturated, since the value of V can be very low. So the equation for the current through T_2 then becomes:

$$I_{out} = I_0 e^{\kappa V_2}(1 - e^{-V}) \tag{4.7}$$

Combining equations 4.6 and 4.7 gives:

$$I_{out} = I_0 e^{\kappa V_2}(1 - \frac{I_{out}}{I_0 e^{\kappa V_1}}) \tag{4.8}$$

Here we recognize the expressions for I_1 and I_2 given in equations 4.3 and 4.4. We insert these and solve for I_{out}:

$$I_{out} = I_2(1 - \frac{I_{out}}{I_1}) \tag{4.9}$$

$$I_{out} = \frac{I_1 I_2}{I_1 + I_2} \tag{4.10}$$

As we can see we have an equation for our correlator, and it will give us information about how the two input currents corresponds with each other. We also notice that the circuit will not *normalize* the input signals, that is, the output current will depend on the input signals amplitude as well as phase difference.

4.3.2 Simulations and measured results

There are many interesting measurements on the correlator. I have divided them in two groups: one for ac- and one for dc-characteristics.

DC-characteristics

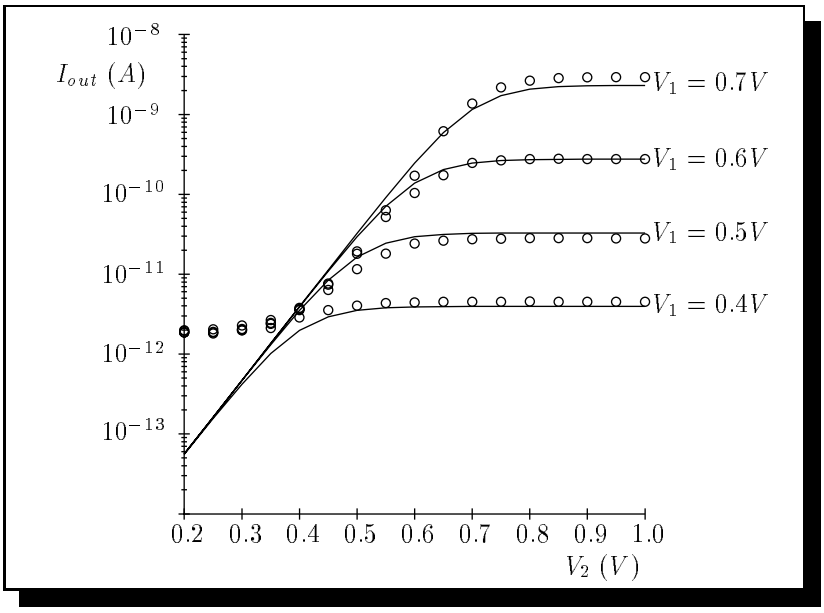


Figure 4.2: I_{out} as a function of the input voltages on a simple correlator

The solid lines is computed from the equations 4.10, 4.3 and 4.4, with $I_0 = 8 \cdot 10^{-16}$, $\kappa = 0.545$ and $\frac{q}{kT} = 39V^{-1}$. The circles are measured results. As we can see the output current begins to saturate when V_2 becomes larger than V_1.

If both inputs of the correlator are large, we will also have a large output current. As shown in figure 4.2 the output current is plotted as a function of V_1 and V_2, with V_2 varying and V_1 set at fixed values. As we would expect the output current saturates when the value of V_2 gets higher than V_1. This is of course due to the fact that V_1 is limiting the current through the T_1 transistor. Thus the lowest input current will always set the limit of the output current.

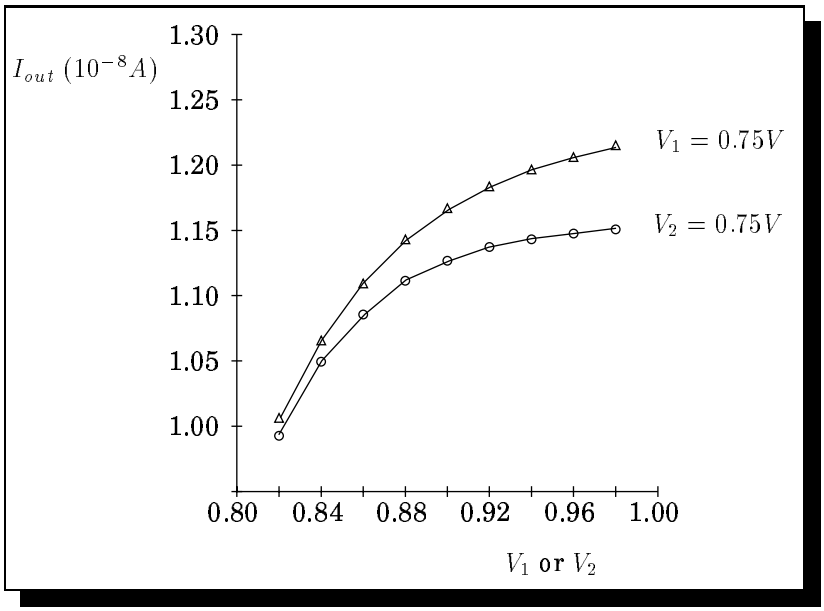


Figure 4.3: Interchanging V_1 and V_2 on an asymmetric correlator

When we look at the equation for I_{out} 4.10 there seems to be no difference between the V_1 and V_2 (or I_1 and I_2) inputs.This figure shows that the output current is different depending on which of the inputs that limits the current. The triangle styled curve is measured with V_1 at a fixed value at $0.75V$. V_2 is then swept from $0.8V$ to $1.0V$, this is as we can see the region where the output current starts to saturate. The circle styled curve is measured in the same way, but with $V_2 = 0.75V$ and V_1 swept. The measured results in this figure are not from the same chip as the results in figure 4.2, so the values of I_{out} are not directly comparable. (The value of I_0 can vary a lot from one transistor to another.)

In equation 4.10 the input currents I_1 and I_2 are symmetrical. According to this equation there should be no change in the output current if we interchange the values of V_1 and V_2. But if we look closer at the circuit this is not the case, that is why this correlator is called *asymmetrical.* We illustrate this by measuring the value of I_{out} for a fixed value of V_1 and varying V_2, and then interchange the two inputs. This is done in figure 4.3. We clearly see that they do not give the same output. How can we explain this

asymmetrical behavior? Let us look back at the correlator circuit in figure 4.1. The output current is limited by the lowest value of the two inputs V_1 and V_2. If $V_1 < V_2$ the T_1 transistor gives the limit for I_{out}. The V_{ds} for T_1 will be large (as we assumed in the derivation of the final equation), in fact it will be more than $4V$. This is because the voltage V will be pulled down to only a few mV above Gnd. So the gate-source voltage of the limiting transistor will be nearly the same as the gate voltage. On the other hand, if $V_1 > V_2$, T_2 limits the current. In this case the voltage V will increase. For normal subthreshold operation it will still be under 0.4V however. The gate-source voltage is the same as the gate voltage on T_2, that is, only a few mV larger than in the previous case. The drain-source voltage will be considerably smaller than in the other case. As we know , not only the gate-source voltage decides the current through a transistor, but the higher the drain-source voltage the higher the output current will be. In this circuit the large difference in drain-source voltages has greater influence on the output current than the small difference in the gate-source voltages. Thus, if the top transistor has the lowest gate voltage, we get the highest output current.

AC-characteristics

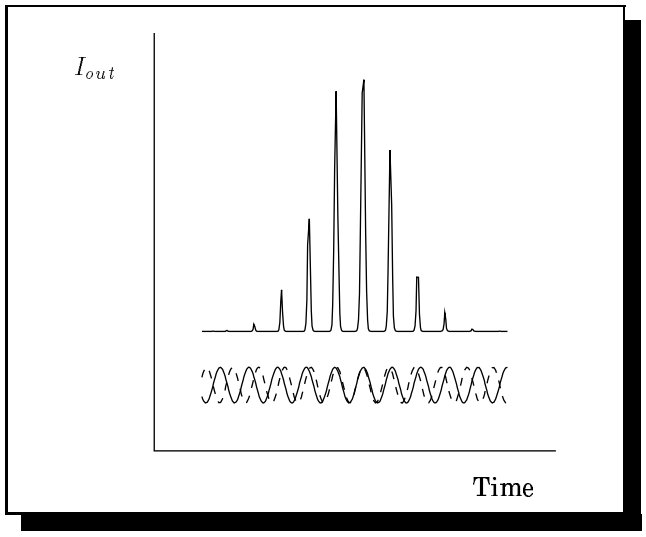


Figure 4.4: Simulated response on a sine wave input

The inputs are both sine waves, with a dc-level of $0.6V$ *and the amplitude (peak to peak) is* $0.6V$*. The frequency for the* V_1 *input is 10kHz and for* V_2 *it is 11kHz.* V_1 *and* V_2 *are plotted at the bottom of the figure, and* I_{out} *is the top graph. The x-axis has units of time. We clearly see that the amplitude of* I_{out} *increases as the two inputs phase difference decrease.*

As mentioned earlier I have no simple correlator on my chip. Unfortunately the chips which contain this correlator do not work as they should any longer. So this ac-part will present simulated results, and since it is a quite small circuit the results from the simulator should be quite close to the values we would have got if we measured. I have simulated the previous measurements (dc) and the simulated and measured results matches well.

What we want to see is how the output current of the correlator vary when we connect two sine waves to the input nodes. The curves in figure 4.4 shows a simulation of this. The correlator gives the highest output when the two signals are in phase. It would in some cases be of interest to see how the correlator works when we increase the frequency of the input signals. In many applications this is a problem, but on my chip we will typically operate with frequencies within the audio range (<20kHz).

I have no units on the y-axis, but typical values will be in the nano-ampere area. Exact values will depend on the input signals amplitude and dc-level. In a simulation of this kind however, the interesting part is to observe the *change* of the output. The similar simulations of other correlators are presented in the same way.

4.4 The symmetric correlator

We have shown that the correlator in the previous section has a weakness: it is not symmetrical. However, it is not difficult to modify the simple correlator, so that this problem is eliminated. We then get the *symmetric correlator*.

4.4.1 Description

If we make two correlators, and let the inputs control the upper transistor on one correlator and the lower on the other, we get symmetric operation. All we have to do is to connect the outputs, thus summing the two currents into one I_{out}. If we examine this circuit, shown in figure 4.5 (a), we get the same result as we did with the asymmetric version, the only difference is the magnitude of I_{out}. Since we have two correlators, the output current will be twice as large as in the asymmetric correlator.

The correlator in figure 4.5 (a) is a simplified version of the one I have used on my chip. In figure 4.5 (b) is the diagram of the version I have used. This one is a basic symmetric correlator with some extra transistors added. First, we have an n-transistor for each input, this let us control the size of the input currents. By changing the gate voltage on this transistor (V_{dc}), we can move the dc-level of the output voltage up and down. Secondly, we have an input stage (the two transistors in the circles), and capacitive coupling from the inputs. As we see, we can have two different types of input stages. The left version is a normal diode connected n-transistor, the second one is a more unusual p-version. I will describe the two solutions in the section about ac-characteristics.

4.4.2 Simulations and measured results

As seen in the figures the symmetric correlator has p-transistors, unlike the asymmetrical one which uses n-transistors. This will not affect the *behavior* of the circuit,

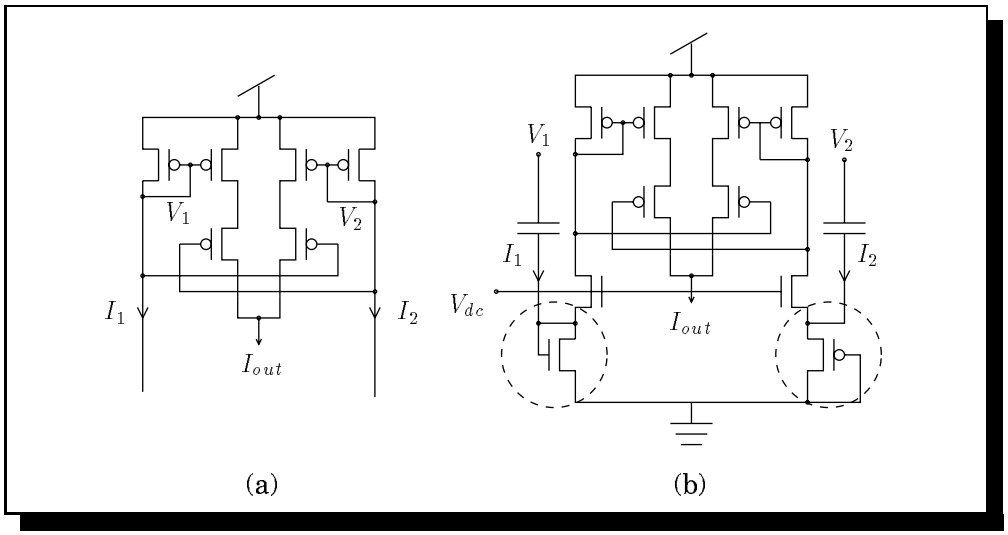


Figure 4.5: Symmetric correlator

In (a) we have the basic symmetric correlator, containing two asymmetric correlators, and in (b) we have the version used on my chip. The circles shows the two different input stages, explained in the text.

but there is a difference in gain between the two types, resulting in different currents. Because of the capacitors on the inputs, the dc-characterization is made by simulating the circuit. The ac-part however, contains measured results.

DC-characteristics

A simulation of the symmetrical correlator is shown in figure 4.6. As we can see the behavior of the circuit is similar to the asymmetric one, but as expected I_{out} is doubled. In fact, the output current from the symmetric correlator is twice the *average* value of the two output currents from the asymmetric correlator with its inputs interchanged. This equals the sum of the two asymmetrical correlator currents. In the figure the value of V_1 is fixed at 4.35V. Both correlators are simulated with p-type correlator transistors. As expected, interchanging the inputs on the symmetric correlator has no effect. The ac-characterization is the important part, since we always will have ac-signals as inputs to the correlators on the chip.

AC-characteristics

Before I present the measured results from this circuit, I will describe the two different input stages. The first solution, the n-type diode connection, is straight-forward. The

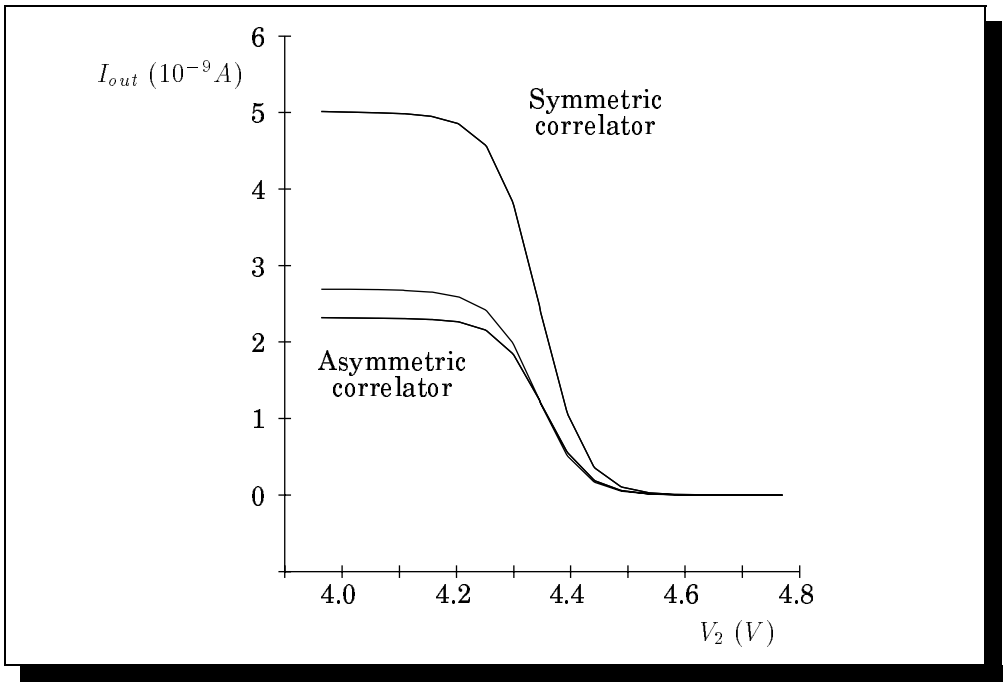


Figure 4.6: DC-simulation of the symmetric correlator

The top curve is I_{out} from the symmetric correlator, and the two others are outputs from the asymmetric correlator, with the inputs interchanged. All curves show the output current as a function of V_2 with V_1 fixed at $4.35V$. Compared with the results in figure 4.2, this figure shows the same behavior. Since this is a p-transistor version the voltages are different from the one in figure 4.2, and with only one value for V_1 we do not need logarithmic scale on the y-axis.

input voltage will lead to a current going through the transistor. The disadvantage is the extra gate-capacitance, resulting in an extra damping of the input signals. Smaller voltage-amplitudes will give a small output current. In the p-transistor variant, we have no extra gate-capacitance. But, we need a larger voltage on the input node to get the p-transistor to lead a current to ground. This is not a problem, in general we can choose the level of the input signals. Since we do not have the gate-capacitance the input voltage will have a larger amplitude than in the other solution, this leading to a larger current. For the n-transistor solution the typical value of this gate voltage will be $V_{dc} = 1.0V$, and for the p-transistor version $V_{dc} = 2.8V$. The correlator on my chip has the p-transistor solution. With this solution the currents feeding the correlator will be an inverted version of the inputs (the correlator responds to the lower half-period of the input signals). In the rest of this work I call these current the (*normal*) inputs, since these are the ones I use in my correlator.

The inputs to the correlators are connected via two capacitors. On the chip these are

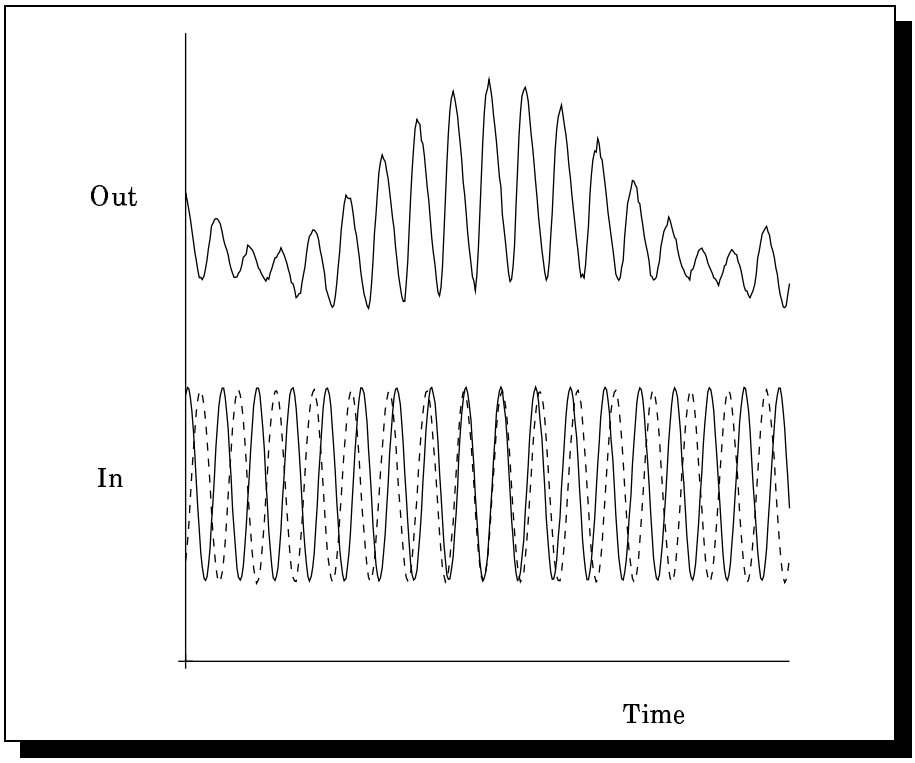


Figure 4.7: Measured symmetric correlator response on sine wave inputs

The top curve shows the output voltage, and the two at the bottom show the inputs. We notice that the behavior is similar to the one in figure 4.4.

produced by two poly-layers above each other. In the simulations in the rest of this work I have used values of 1pF for the capacitors.

Even if this correlator is symmetric, only one half period of the inputs will influence on the output current. This can clearly be seen in figure 4.7. This is because the input currents will increase with one half period and decrease in the other. If we want I_{out} to respond on both half periods we must build a new correlator. The curves in figure 4.7 are measured voltages from the oscilloscope. The output voltage is the voltage across an n-diode to Gnd. Inputs are two sine waves, with frequency 10kHz and 11kHz, with an amplitude (peak to peak) of 3V and a dc-level on 2.5V. To get the correlator to respond correctly, the V_{dc} is set to 2.75V. The dc-level of the output is about 650mV, and the maximum amplitude only about 40mV. To get the oscilloscope to trigg correctly we can not use any of the input sine waves as the trigger source. This made it very difficult to get this measured result look as nice as it is in the figure. What we need is to use the output voltage as a trigger source, and adjust the trigger level to exactly the top of the largest amplitude. Because we measure the voltage over a diode we get a logarithmic compression of the current. We see this if we compare figure 4.4

with figure 4.7. The simulated curve in figure 4.4 has a larger difference between its minimum and maximum than the measured curve in figure 4.7.

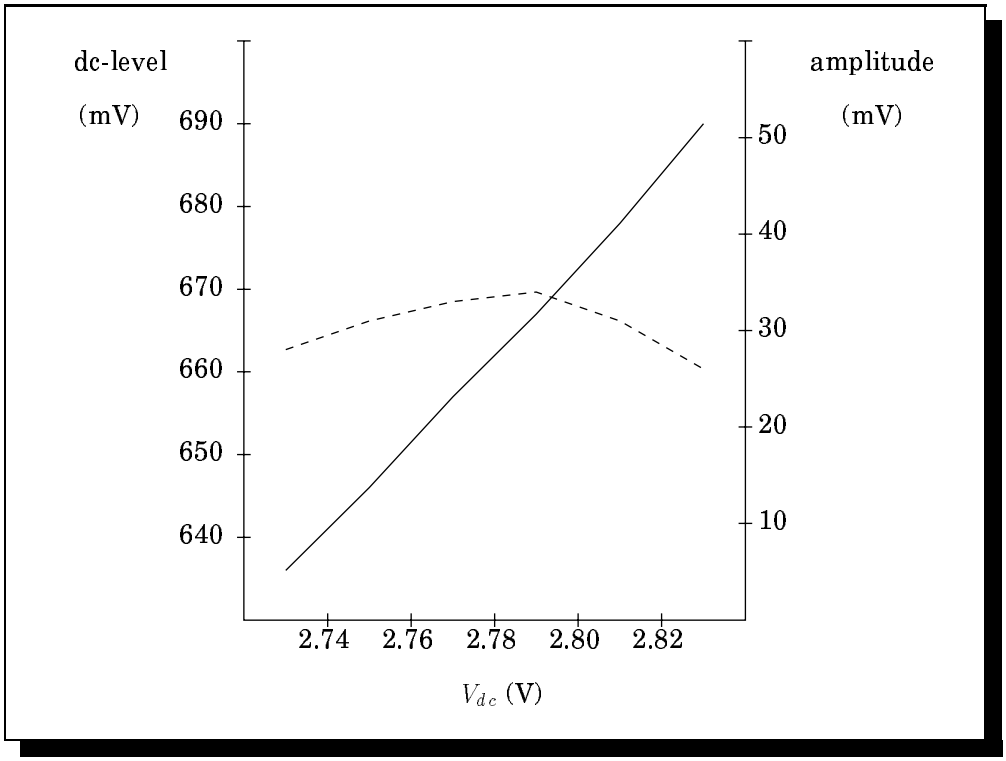


Figure 4.8: Changing the dc-level of the output

The voltage on the gate (V_{dc}) has values from 2.73V to 2.83V. The inputs to the correlator is two 10kHz sine waves with peak to peak amplitudes of 3V and dc-level 2.5V. The left axis shows the dc-level values and corresponds with the drawn line. The dashed line shows the amplitudes of the output, and the values are shown on the right axis.

As mentioned earlier we have added a transistor to regulate the dc-level of the output voltage. By varying the V_{dc} input and then measure the output voltage we get the result shown in figure 4.8. As we see, the dc-level will increase when we increase V_{dc}. Only a small range for the variation of V_{dc}, about 2.7V to 2.8V, are possible however. If we go outside this area the amplitude of the output current will decrease, and this is not what we want. If the voltage becomes smaller the current is to small to produce larger output current. On the other hand, if the voltage increase to much the input node will not be able to alternate as much as it should, this also leads to a smaller output current. Depending on the input signal amplitude, the output amplitude will also change, this will only move the dashed line up or down.

4.5 Other correlators

In addition to the two correlators I have discussed in detail, one can imagine a number of other variants. I have chosen two circuits, the improved symmetric correlator and the correlator/anticorrelator, and will examine these now. However, these correlators are treated quite briefly compared to the previous ones. Both correlators have the input-stage with n-transistors (a normal diode), and the simulations presented have $V_{dc} = 1.0V$.

4.5.1 The improved symmetric correlator

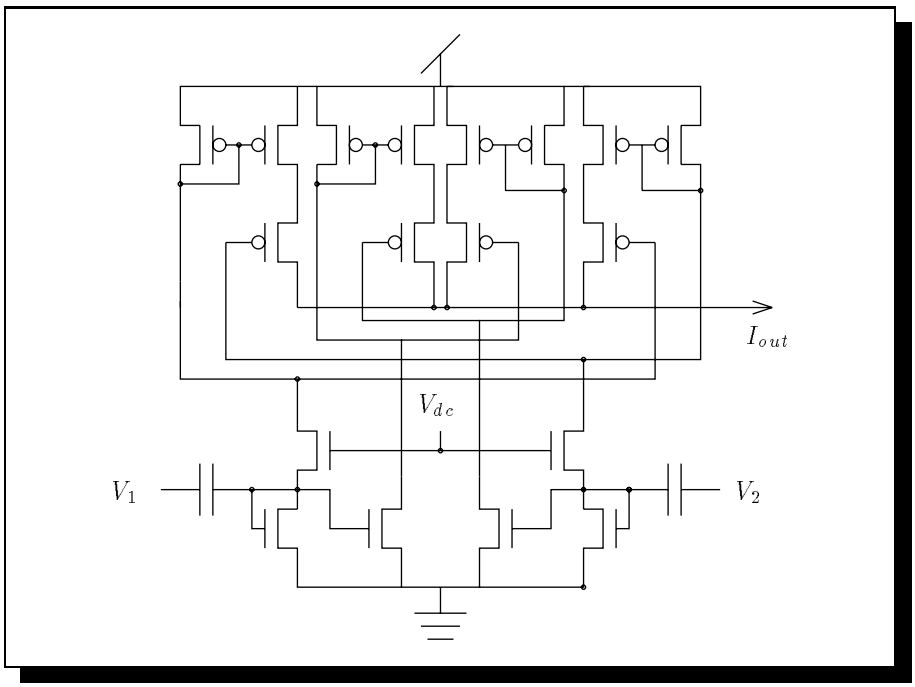


Figure 4.9: Improved symmetric correlator

This correlator will have an output current which responds on both half-periods of the input signals. The output will be the sum of four currents from different correlators. The extra n-transistors make currents that are inverted compared to the currents used in the symmetrical correlator. In this way we achieve full-wave operation.

It is possible to improve the symmetrical correlator. The correlator described in the previous section has one weakness, the output current is varying only with one half-period of the input signals. To avoid this it is necessary to double the numbers of correlation transistors (from 4 to 8). In some cases we might need a full-wave operation,

but in other cases the simple asymmetrical version will be sufficient. For example, if we want to integrate the output current from the correlator, full-wave operation will be a better solution.

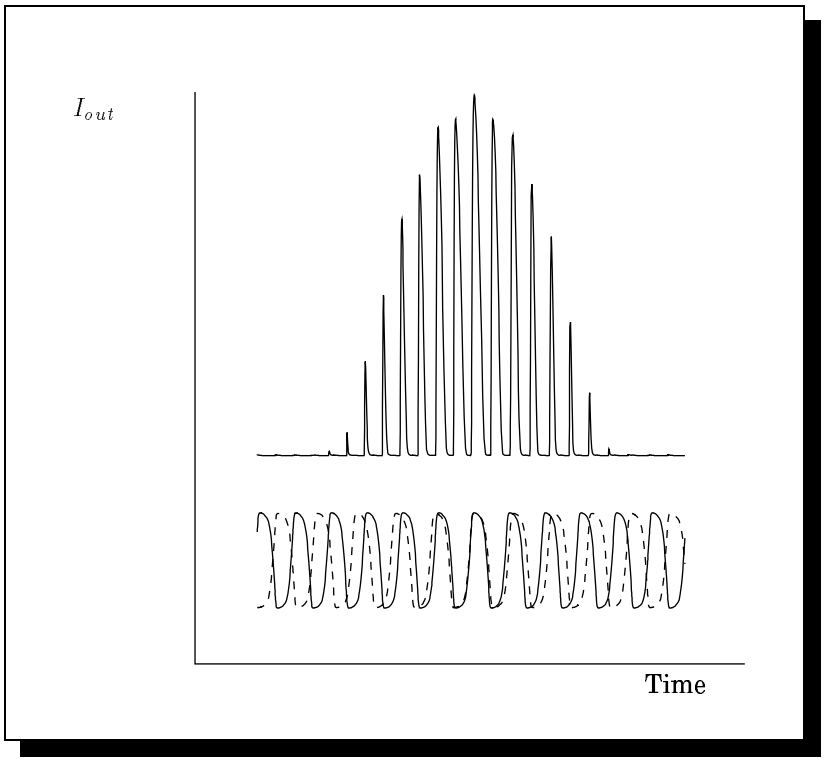


Figure 4.10: Simulation of the improved symmetric correlator

The top curve is I_{out}, the bottom curves are the inputs. We now have correlation on both half periods of the inputs. (Compare with figure 4.4.) As we see the input curves are not exact sine-waves, the curves actually shows the signals on the gate of the input transistor, after the capacitive coupling, see figure 4.9. As earlier the inputs to the capacitors are two sine waves, 10kHz and 11kHz, peak to peak amplitude 3V and dc-level 2.5V.

The improved version is shown in figure 4.9. The operation is similar to the other versions, but an extra transistor make currents that responds to the upper half-period of the input signal. I call these currents inverted currents, since they are inverted compared to the normal input currents described earlier. (Compared to the input signal however, the names of these currents should be interchanged.) We can now build a new symmetric correlator with these two new currents as inputs, and in the end we can add all four output currents into one new I_{out}. A simulation of this circuit, similar to the simulation in figure 4.4, is shown in figure 4.10. We can clearly see the differ-

ence in the result.

Another interesting observation in this simulation is the fact that the output current is different for the two half-periods. We can see this by looking at the peaks near the maximum at the middle of figure 4.10. The two peaks on each side of the maximum peak is only a little larger than the next peaks. In figure 4.12 we see this even clearer. Here the peaks next to the middle peak are smaller than the peaks one step further out. The reason for this is the V_{dc}-voltage. This voltage will have influence on the current for the low half-period. On my chip this will have no effect since I use only the low half-period.

4.5.2 Correlator/anti-correlator

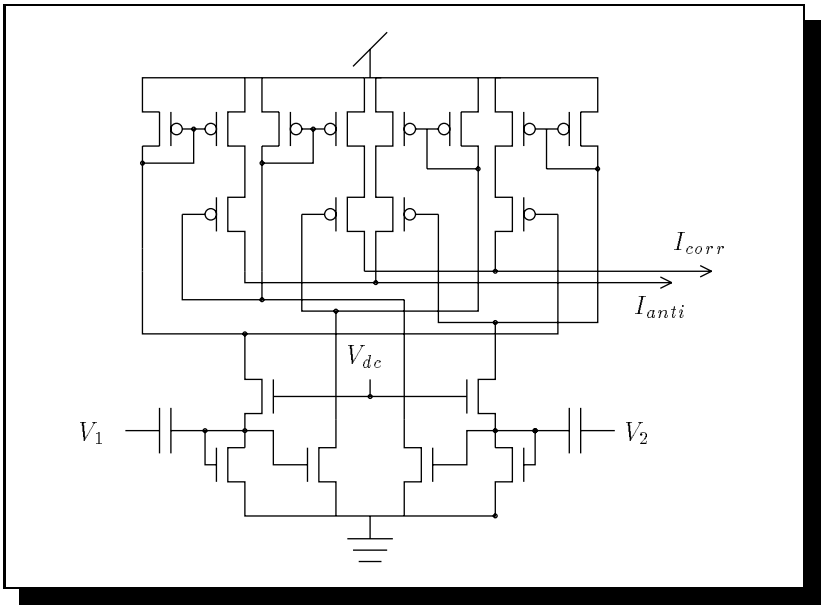


Figure 4.11: A correlator and anti-correlator

This circuit has two output currents. The first (I_{corr}) is the normal output current like the ones from previous sections. The second (I_{anti}) is the opposite current, when the two signals have a opposite phases this current will be large.

Another task that we might want is a circuit to perform the opposite of a correlation. We call this (as expected) *anti-correlation*, and this is also done by a small change of the original circuit. Figure 4.11 shows a diagram of a circuit with both the correlation- and anti-correlation-current as outputs. To achieve this we simply use the same currents as in the improved symmetric correlator, but combine one inverted current with one normal current. Then we can make one, two, or four stages, depending on how accurate

we want the output current to be.

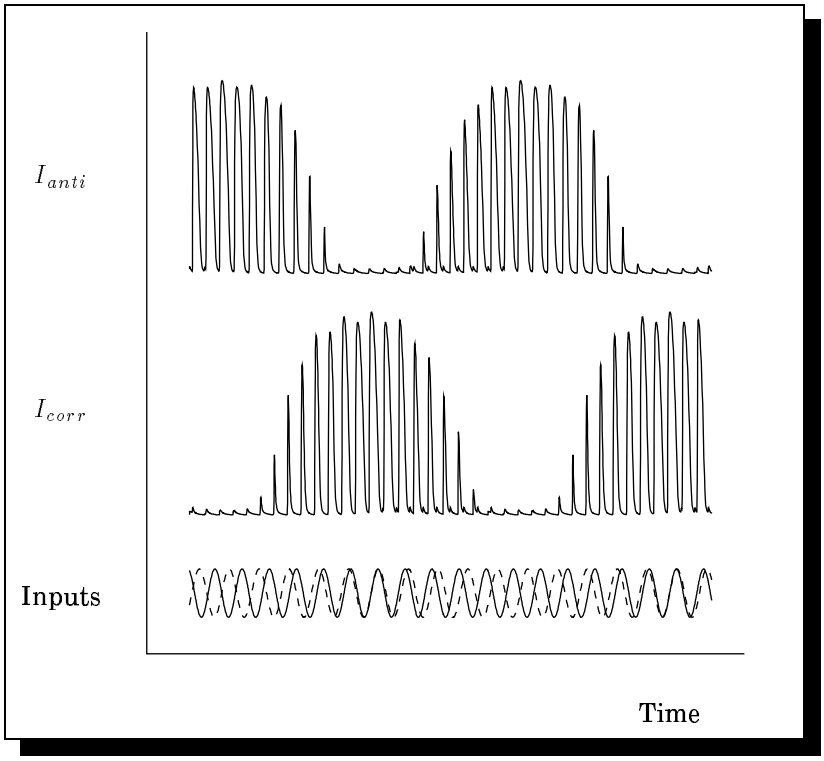


Figure 4.12: Simulation of the correlator/anticorrelator circuit

The two curves at the bottom are the inputs. The top curve is the anti-correlation output, and the middle curve is the correlation output. We can see that the two currents I_{corr} and I_{anti} are the opposite of each other. Inputs are 10kHz and 11kHz sine waves, with peak to peak amplitude 3V and dc-level 2.5V.

In this correlator we have combined as follows: two normal currents and two inverted currents as inputs to two correlators. These are added to form I_{corr}. To form I_{anti} we also have two (anti-)correlators, combining one normal and one inverted current each. In figure 4.12 we clearly see how the two output currents behave. By combining the current in this way we get correlation (and anti-correlation) on both half-periods of the input signals. We notice the influence of the voltage V_{dc}, the peaks corresponding to the low half-period is smaller than the peak corresponding to the upper half-period. We see this best in the curve for I_{corr} since one of the internal correlators has both its inputs from the normal current. Thus the V_{dc} voltage reduces the currents corresponding to the low half-period (the normal current).

4.6 A major drawback

There is one major problem with these correlators. As we know, the output current is defined by $I_{out} = \frac{I_1 I_2}{I_1 + I_2}$. When we put an ac-signal on the inputs, the output current will be largest when the two input signals have the same phase. But, the magnitude of I_{out} will of course also depend on the sizes of the input signals. We can say that the circuits do not *normalize*, or they do not have the ability to *scale* the inputs so that only the phase difference decide the output. As long as we can control the size of both inputs it is not a problem. However, on my chip one of the input signals will be a delayed version of the other input. And when we delay a signal it will also be damped, thus the output current from the correlator will decrease. This problem, and others, will be discussed in chapter 8.

4.7 The use of correlators on my chip

The correlator is the main computational element on my chip. It is a powerful circuit, with very few transistors. I have used the symmetric version from figure 4.5 (b). The detailed description of the other parts of the chip, and how the correlator cooperate with these, is the subject of the next chapter.

Correlator circuits are used in many other systems. In for example [1], [5], [9] and [12] the correlator circuit is used.

5

The development of a chip : PASUC

This chapter describes all the other circuits needed to make the chip work. First I will give a short and general description of the chip, then I take a close look at all the circuits in detail. Simulated and measured results are included.

5.1 What is PASUC?

Everything must have a name, and this chip is of course no exception. I call my chip PASUC, which is short for:

Processing
Analog
Signals
Using
Correlators

As the name tells us, we want to process analog signals, and our main computing element is the correlator treated in the previous chapter. We wanted to make this chip to see how the correlator could be used to give us information about the frequency spectrum of an analog signal. This has been done digitally, by using an FFT algorithm. It is then necessary to do a sampling of the original signal and then use digital techniques to process it. In our circuit the signal should be a direct input to the chip, without any pre-processing.

We know that the correlator gives information about how two signals corresponds with

each other. This circuit uses auto-correlation, that is, we see how the input signal correlate with a delayed version of itself. This correlation will vary when the frequency of the input signal is changed, and this is the main idea behind the circuit.

5.2 The PASUC chip

5.2.1 A sketch of the circuit

A sketch of the circuit is shown in figure 5.1. It has 24 stages connected in a row. Every stage consists of the following elements:

- a delay element,
- a correlator,
- a local-winner-take-all,
- a select circuit,
- a shift-register.

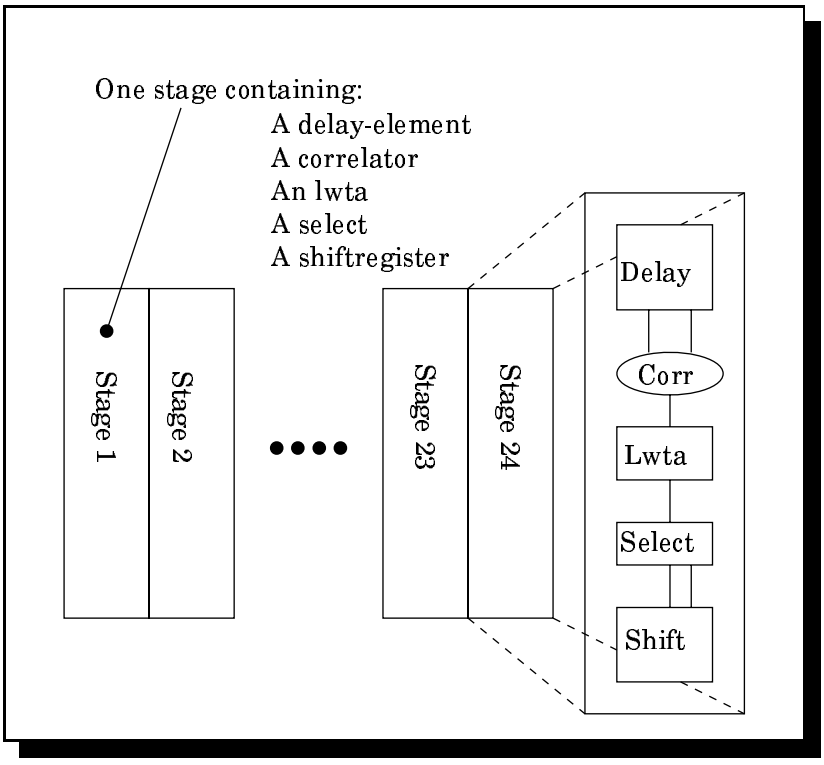


Figure 5.1: A sketch of the circuit

The correlators perform an auto-correlation on the input signal, this is the key to the function of the chip. The other circuits are there to get the correct inputs to the correlator and to make the outputs available to us.

Briefly we can describe how the circuit works as follows: the delay-element (a follower with capacitive load), makes delayed versions of the input signal. The further out in the line we get, the larger the delay will be. These delayed versions is taken as one of the inputs to the correlators, the other inputs being the original input-signal. Thus we do an auto-correlation. The outputs of these correlators will then change as we have different frequencies on the input signal. To make this change more visible, we take the correlation outputs as inputs to a winner-take-all circuit. This serves as a "filter", in a local area the smaller signals will be canceled out by the larger ones. A schematic view of this operation is shown in figure 5.2.

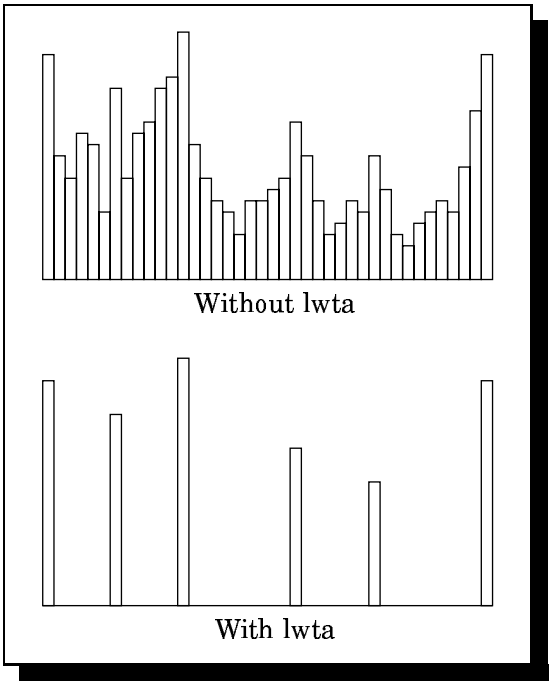


Figure 5.2: The function of the lwta

This is a schematic diagram, it shows that winners can be of different sizes, a winner one place can be a looser another place. The important thing is that the winners have local maximum values.

With this configuration we get 24 outputs that we want to observe. These outputs are time-multiplexed, we look at one output for a short time, and then switch to the next output etc. To choose one single output at a time we need two things: a shift-register to tell us which element we wish to observe, and a circuit to route this specified output to a global output pin. We call this circuit a select-circuit. We now let the shift-register shift a single bit from left to right, and start from left again when it has reached the end. The element with the shift-register bit set is the one we observe at the output pin.

5.2.2 Some general information

One of the things we must consider when making a chip like this is the sizes of the transistors. As with the correlators, the transistors in the rest of the circuit generally has a width to length ratio of 1, otherwise the ratio will be noted.

This was just a very short and general description of how the chip is supposed to work. Even if this is not a very complex circuit, there are a lot of things that must be considered when we construct the circuit. Many of the circuits on the chip are internal circuits, it is not possible to measure the outputs from them directly. A separate test-copy of each circuit-element so that it would be possible to make measurements on each of them is often convenient to have. The only separate test-circuit I have on my chip is the correlator. Therefore I will present both simulated and measured results. The detailed description and analyses of the different circuits is the main topic of the rest of this chapter.

5.3 The delay element

5.3.1 Description

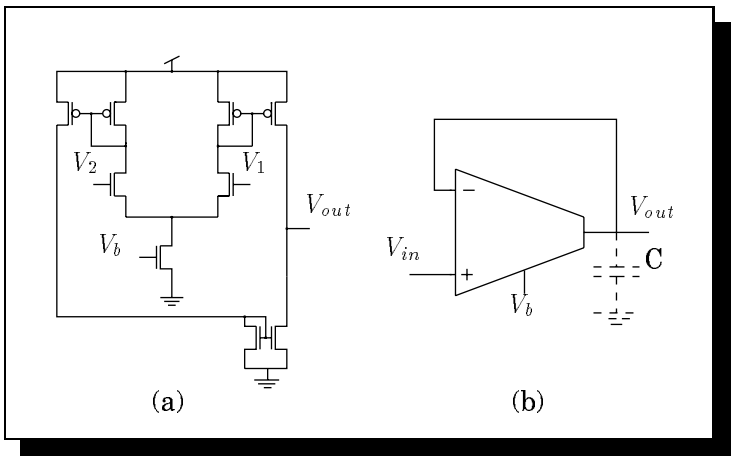


Figure 5.3: The delay element

In (a) we have the wide-range transconductance amplifier, and in (b) the follower/delay element. In the symbol for the wide-range amplifier in (b) V_2 is the – marked terminal and V_1 the + marked.

The task of the delay element is as the name implies to give us delayed copies of the input signal. To build a follower we can use a simple transconductance amplifier like the one shown in figure 2.3 (page 10). The amplification (gain) of an amplifier like this is typically about 200. By modifying this circuit a little we can get an amplifica-

tion factor of about 2000. This circuit, shown in figure 5.3 (a) is called a wide-range transconductance amplifier. Two current mirrors have been added, and we reduce the lower limit for V_{out} compared to the simple transconductance amplifier limit (the V_{min} problem). When connected as a follower however, we do not have this V_{min} problem anyway, so the main reason for choosing the wide-range version is higher gain. An effect we have to live with is the offset in the amplifier. But, as we shall see later, this is not the main problem with the followers on my chip.

To build a follower we simply connect the output voltage to the V_2 input, like it is done in the circuit in figure 5.3 (b). We first consider the amplifier connected as a follower (without the dashed capacitor connected). We know that the output voltage is the difference in input voltages multiplied with the voltage gain A, so we have the following equation for V_{out} (from [13]):

$$V_{out} = A(V_{in} - V_{out}) \tag{5.1}$$

From equation 5.1 we then get the transfer function of the circuit:

$$\frac{V_{out}}{V_{in}} = \frac{A}{1+A} = 1 - \frac{1}{1+A} \tag{5.2}$$

Considering the fact that we use a wide-range amplifier with a gain $A \approx 2000$, we observe from equation 5.2 that V_{out} is very close to a copy of V_{in}. Hence we now have a *voltage follower*.

The follower gives us a copy of the input voltage, but we want the output to be a delayed version of the input. If we add a capacitor between Gnd and the output of the amplifier (the dashed part of the circuit in figure 5.3 (b)), we get a new circuit to examine. The output current (I_{out}) from the transconductance amplifier is given in equation 2.2, and we have the following equation for the charging of the capacitor (given in [13]):

$$C\frac{dV_{out}}{dt} = I_{out} \tag{5.3}$$

If we assume that we stay in the linear area (that is, we have small steps in the input changes), the tanh function can be approximated by its argument, so from equation 5.3 we then get:

$$C\frac{dV_{out}}{dt} = G(V_{in} - V_{out}) \tag{5.4}$$

In s-notation this becomes:

$$\frac{V_{out}}{V_{in}} = \frac{1}{\tau s + 1} \quad , \quad \tau = \frac{C}{G} \tag{5.5}$$

This is the solution of a first order differential equation, the same solution appears for the RC-integrator. (A simple low pass filter.) There are however big advantages with the transconductance version. If we make a delay line, consisting of several delay elements in series, the stages will in fact be isolated from each other. Thus, every stage has its own current supply, and the rise time will be held at reasonable values even for long lines. This is not the case for an RC-line, where one current must charge all the capacitors, thus giving large rise times after only a few stages.

An equation that gives us the relationship between the output voltage and the previous

input voltages is given in [13]:

$$V_{out}(t) = \int_0^{\infty} V_{in}(t-\Delta)e^{-\frac{\Delta}{\tau}}d\Delta \tag{5.6}$$

Here t is the time of the observation, and Δ is the time prior to the observation. This means that $V_{out}(t)$ is depending on the previous values of V_{in}, the older they get the less influence. We have an exponential decrease of the influence.

We now have a circuit to give us delayed versions of an input signal. How large this delay is will depend on two factors:

- The size of the capacitor. This is decided before we make the layout of the circuit, the capacitor on this chip is made by two overlapping poly layers.
- The bias of the transconductance amplifier, that is , the amount of current it can supply. In the equations this is the G. We can change the value of G by adjusting the V_b voltage. The output current also depends on the sizes of the transistors in the amplifier.

My amplifier has a layout similar to the one in [13][1].

5.3.2 Delaylines

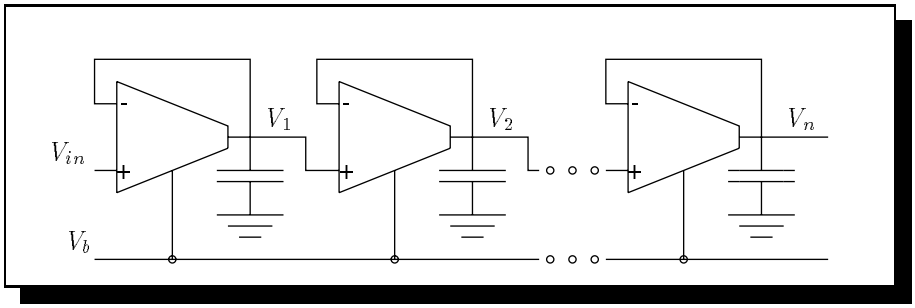


Figure 5.4: An n-stage delayline with transconductance amplifiers

On my chip n=24, and the capacitors are not grounded but connected as inputs to the correlators. This does not affect the behavior of the circuit. All capacitors have the same size.

As I mentioned earlier it is possible to connect many followers in cascade and get what we call a delay line. In figure 5.4 such a delay line is shown. On my chip I have a delayline with 24 stages. In the figure the bias voltage is equal for every stage. This makes the delay equal for all stages, since the capacitors are equally sized. We can get different delays on the different stages by making a *tipped polyline*. That is, we make V_b as a polyline, with both ends as input pins. Setting different voltages on these pins will (due to the resistive poly layer) give us different V_b values on every stage. This is a *voltage divider*, the poly is serving as several resistors in series.

[1] Plate 7 (b).

5.3.3 Simulations and measured results

Since I do not have a single delay element on my chip I must simulate the behavior of this circuit. It will be shown however, that the circuit does really work as it should, when I test the delay line. One can also treat the line as one element, by setting the bias voltage so high that the elements act very close to a follower.

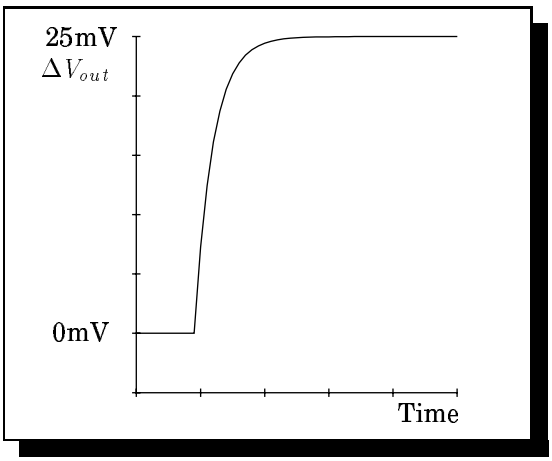


Figure 5.5: Simulated delay element response to a small input change

The curve shows the response to a input change (ramp) of 25mV. The x-axis has unit time, and the values will depend on the bias-voltage on the transconductance amplifier and the size of the capacitor. The input voltage prior to the ramp is 2.5V. We clearly see that the output follows the input, but due to the charging of the capacitor, it will take some time to reach the final value.

In figure 5.5 I have simulated one single delay element. The response to a 25mV ramp is a typical delay element (or low-pass filter) response, V_{out} follows an exponential function (like $y = K(1 - e^{-\frac{t}{\tau}})$). For larger input changes (steps) the transconductance amplifier no longer stay linear (the tanh-function can not be approximated by its argument). The response will then be more like a straight line, and the output for a sine wave input will become a triangle shaped curve. In that case the circuit is limited by its slew-rate, that is, the amplifier can not supply the amount of current needed to charge the capacitor fast enough so that the output can follow the input reasonably good. On this chip we will concentrate the testing on smooth signals, thus we will stay in the linear region.

The 24-stage delay line on the chip may be viewed as a single delay element. With the bias voltage at 1.3V, and V_{in} as a sine wave signal, the response is as shown in figure 5.6. With a bias voltage at 1.3V we are outside the weak inversion region. But the transconductance amplifier will work as a follower even with bias voltages over the threshold-voltage. With the delay as small as in this case, the output voltage would

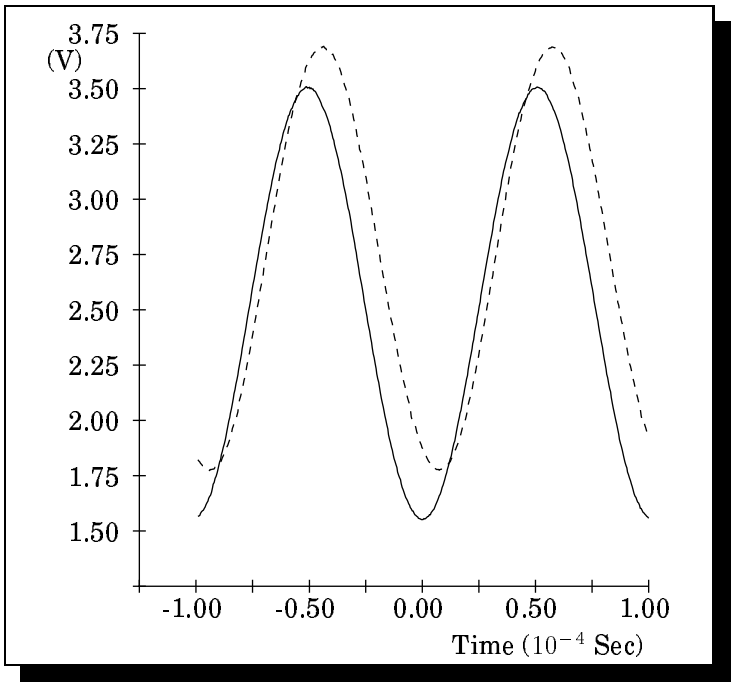


Figure 5.6: Measured delay line response

The solid curve shows the input signal, f=10kHz. The output after 24 stages of the delay line is shown with the dashed curve. Notice the offset, in this case about +200mV.

be a very good delayed copy of the input. But we do have one deviation, the offset voltage. As seen in the figure, V_{out} is raised by about 200mV. The offset problem is a well known effect, which can not be eliminated. We can reduce it however, by using a Benson diode[2]. For large delay lines this can be of great importance. On my line however, the offset problem seem to more or less vanish due to the *damping problem*, since a 24-stage line is a quite small line.

By decreasing the bias voltage, the current charging the capacitor will decrease as well. Thus we get larger delay, and when this delay is large enough the delayed signal will be damped significantly (see chapter 8). This might be a problem for the circuit. To get the correlators to distinguish between frequencies we must have some delay, but then the inputs to the correlators will be weaker towards the end of the line. This again leads to the problem of finding the correct winners from the output currents. More about this in section 5.4.

[2]The Benson diode can be either a n- or p-type, placed on one of the sides of the amplifier between the current mirrors. The chip described in chapter 7 use a simple amplifier with a Benson diode, and a transistor diagram of a simple amplifier with Benson diode is shown in figure A.5.

5.3.4 Offset in the delayline

One could write a lot about offset problems in transconductance amplifiers. The amplifiers I use have offset as well, but extra long transistors are used at the output to reduce the Early effect. But as we see in figure 5.6 we have a considerably offset. I have tested this on several chips. The different chips have different offsets (as expected), but the offset on one chip is nearly constant for voltages between 0.5V and 4.5V (typical +200mV). This is not what we expected. Because of the extra current mirror in the negative current, a negative offset seems natural. This offset should also depend on the input voltage. Simulations in *analog* gives the expected results. On the PASUC chip we did not have the necessary nodes available as outputs to investigate this in detail. But as I have mentioned, we do not have an offset problem here because of the damping of the signals. Therefore I have not treated this issue in detail.

5.4 The correlator

5.4.1 Connections

Since the correlator is the most important element on my chip I have discussed different types and their function in chapter 4. In this section I will describe how the correlator works together with the other elements on the chip.

- The correlator has two inputs. The first one is common for every correlator, the second one is the output from each stage of the delayline. Both inputs have capacitive coupling, thus we can only work with ac-signals. The capacitors on the second input is the same as we use to get a delayed signal, described in the previous section.
- The output of the correlator is the sum of two currents (as we use the symmetric correlator, shown in figure 4.5), and this current is taken as an input to a local-winner-take-all circuit. That is, we do not want to observe all the currents, only the ones with the largest values.

Due to the fact that the correlators I_{out} depends on the amplitude of the inputs, we have a problem when we shall find the local winners described above.

The equation for the output current from the correlator is derived assuming that the transistors operate in weak inversion. As input to the delayline we use larger signals however, and we must ask if this leads to strong inversion operation of the correlator transistors. The answer is that because of the capacitive coupling to the correlator inputs the voltages will be considerably damped, and the output currents will be very small, so we still operate in weak inversion.

5.4.2 Auto-correlation

By auto-correlation we mean the correlation between a signal and a delayed version of itself. If we try to measure the output of a correlator as a current we will have problems finding the amplitude. Since we use an oscilloscope we must convert the current into a voltage, this can easily be done by a diode-connected n-transistor to ground on the output from the correlator. We can measure this voltage over this diode on the scope. It is now interesting to see how this voltage vary when we change the delay with the

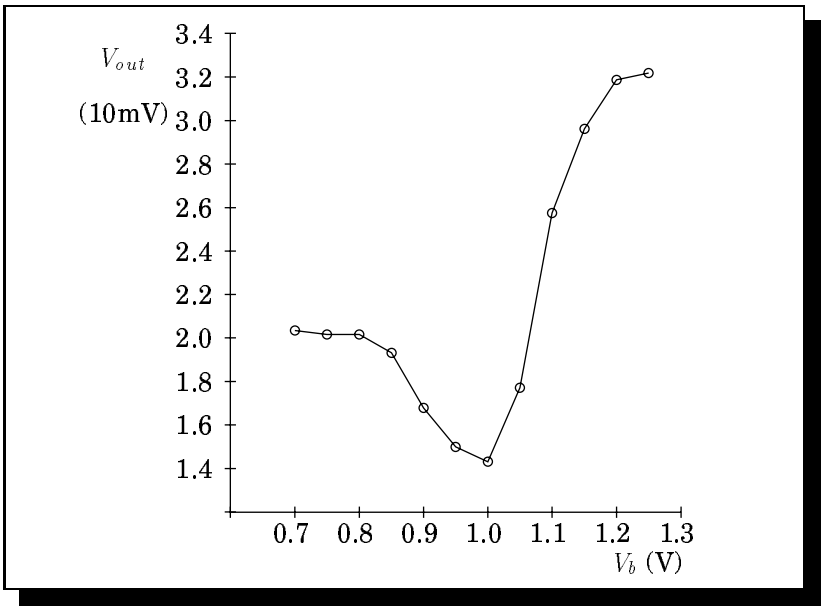


Figure 5.7: The auto-correlation with varying delay

The curve shows the measured voltage over a diode connected to the output of the correlator. The circles are measured points, lines are drawn between them. One input to the correlator is a 10kHz sine wave signal with peak to peak amplitude 3V and dc -level 2.5V, the other a delayed version. The size of the delay depends on the bias voltage (V_b), here it is varied from 0.7V to 1.25V. For lower values the signal will be to much damped, for higher voltages the signal will be a nearly perfect follower (For larger V_b values the output will stay at about 32mV).

bias control of the followers. In figure 5.7 this is done. As expected the voltage will be highest when we have no delay (the right end of the curve). The output reaches a minimum when the delayed signal is delayed by one half period, occurring when V_b is about 1V. When the delay is greater than this the output will again increase, this is actually a correlation with the *previous* period of the input signal. We observe that the correlation output is smaller when the result comes from a correlation with the previous period. This is because of the damping of the signal when the delay increases.

So the behavior of the correlator is like expected, but still there is a problem with these measured results. The size of the output signal is very small, and we should expect greater values. Simulating the same circuit gives us considerably larger output voltages. The reason for this small output change is probably the sizes of the capacitors.

5.4.3 Correlation and frequency

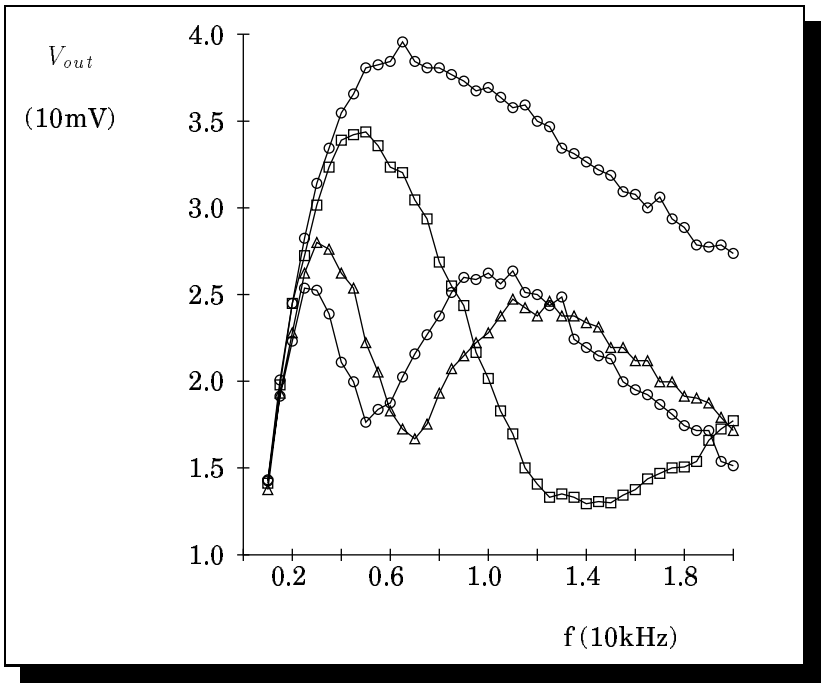


Figure 5.8: The frequency response of the auto-correlator

As input to the circuit we use a sine wave with the frequency swept from 1kHz to 20kHz. Peak to peak amplitude is 3V and the dc-level is 2.5V. The four curves shows the output with four different delays (V_b), on the circuit this would correspond to different stages along the line. We see that they have one or two local maximum output values. The top circle-styled curve has $V_b = 1.5V$, the box-styled curve has $V_b = 1.04V$, the triangle styled curve has $V_b = 0.97V$ and the bottom circle-styled curve has $V_b = 0.95V$.

We have seen that the auto-correlation will give us an output that changes with the delay of the signal. Since the delay will vary from near zero at the beginning of the line to a considerably delay at the end, it is obvious that the correlators will give us different output currents. What then if we try to change the *frequency* of the input signal, and have a constant delay? Although if we change the frequency and not the delay, the ratio between the delay and the period of the signal will change. Thus, one should expect that the output from the (auto-)correlator has a maximum at some frequency, depending on the delay. And this is exactly what we get. A look at the curves in figure 5.8 shows this.

This figure also shows the most severe problem with this circuit. The maximum values

are not equal. That is, even if one stage has a local maximum, one of the neighbors might have a greater output value without being a maximum. This makes it impossible to be sure that we find the correct winners all the time. The reason for this is as mentioned earlier the damping of the signal in the delay line as the delay increases. There are two ways to eliminate this problem:

- Make a delay line with no damping, that is, perfect copies are made. Unfortunately very few things in this world are perfect, so this solution is not a realistic one.
- Get the correlator to *scale* the input signals, so that we get a form of normalized signals on the output. How this could be done is not a straightforward task, I will discuss improved circuit operations in chapter 8.

Even if this is a problem we can certainly see that the correlators have the properties we need. The output vary with both delay and frequency, and the correlator turns out to be quite powerful compared to its size.

5.5 The Winner Take All circuit (WTA)

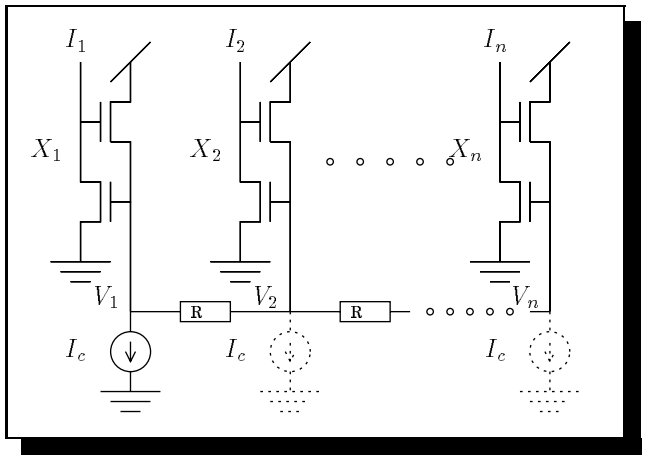


Figure 5.9: The winner take all circuit

The figure shows both the WTA and LWTA circuit. With R=0 and the dotted lines not connected we have a WTA, and with R set to some fixed value and the dotted lines connected we have the LWTA circuit.

As mentioned in the general description of the circuit we want to do a selection among the output currents from the correlators. What we want is only the largest currents, and neglect the others. To do this require a circuit called the *Local Winner Take All*, from now on called the *LWTA*. First we will examine a variant of this circuit, the *Global* Winner Take All, the word Global is often omitted, so it is called the *WTA* circuit. This

circuit is described in [10].

In figure 5.9 the circuit for this operation is shown. The input currents (I_1, I_2, $\cdots$, I_n), are outputs from the correlators. Consider the case where R=0 and the dotted current sinks not connected. This is the WTA circuit. We now look at the case where we only have two input currents. For $I_1 = I_2$ the voltages X_1 and X_2 will be equal as well. These are the output voltages of the circuit. If one of the inputs, say I_1, is larger than the other we get a new situation. Let $I_1 = I_2 + \delta$. For small values of δ the voltage X_2 will decrease linearly with δ. Larger values of δ will lead to $X_2 \approx 0$, thus X_1 as the larger input will be the winner. The output current will approximately be the current through the transistor controlled by the X_1 voltage at its gate. The operation will be similar for circuits with more than two inputs.

The WTA gives us one winner, the output voltage X_k is a logarithmic compression of the input I_k that is the largest input (for $k = 1 \cdots n$). But we want local winners, that is, input currents larger than their neighbors should be winners. What we want is illustrated in figure 5.2. To get this behavior we must have one current sink for each input. So we must include the dotted sinks in the circuit. In addition the voltages V_1, V_2, $\cdots$, V_n must no longer be forced to one common V. This is done by setting the resistors (R) to some fixed value. With this new circuit the different inputs will influence on their neighbors, thus we no longer have only one winner. How many winners we have will depend on the size of the resistors R. With $R = \infty$ all inputs are winners. The next section describes how these resistors are made.

5.6 Resistors in CMOS

The layers in a process for making a chip have different resistance. One way of making a resistor is to have a resistive line of for example poly, but to get large values one must have very long lines, and this will of course occupy to much of the chip area. Because of this we need another way of making resistors, by using transistors and controlling the current through them. This circuit is described in [13].

5.6.1 Hres part

In figure 5.10 (a) the Hres circuit is shown. Two transistors are connected in series, since we want the resistor to be bidirectional. The boxes labeled B are the bias circuits, they set the gate voltage of the transistors. Lets assume all voltages in units of $\frac{kT}{q}$, and that $\kappa V_{g1} - V_1 = \kappa V_{g2} - V_2 = V_q$. Thus the current through the transistors Q1 and Q2 is:

$$I = I_0 e^{V_q}(e^{V_1 - V_n} - 1) = I_0 e^{V_q}(1 - e^{V_2 - V_n}) \tag{5.7}$$

We solve this for e^{V_n}:

$$2e^{V_n} = e^{V_1} + e^{V_2} \tag{5.8}$$

Further we have:

$$\frac{I}{I_{sat}} = \frac{e^{V_1} - e^{V_2}}{e^{V_1} + e^{V_2}} \tag{5.9}$$

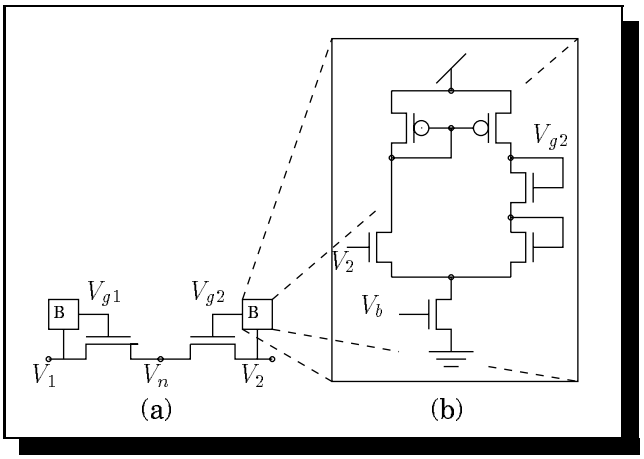


Figure 5.10: A CMOS resistor

Part (a) shows the Hres circuit, and part (b) the bias circuit. The bias circuit sets the gate voltage of the transistor to the desired value above the V_1- and V_2-nodes.

Here $I_{sat} = I_0 e^{V_q}$. We then get the following expression for the current through the transistors:

$$I = I_{sat} tanh \frac{V_1 - V_2}{2} \tag{5.10}$$

Using Ohms law we can find the value of the resistor:

$$R = \frac{2kT/q}{I_{sat}} \tag{5.11}$$

Thus we have a circuit giving us a resistor with two transistors, working well if the bias circuit gives us gate voltages so that we operate in the sub-threshold region.

5.6.2 Bias part

Since the V_g voltages must depend on the voltages V_1 and V_2, we can not set these gate voltages by for instance a separate input pin. Figure 5.10 (b) shows a circuit to set the voltage about one diode offset over the V_1 or V_2 voltage. This is done by a simple transconductance amplifier, connected as a follower, with an extra n-diode connected transistor between the output and the current mirror. By changing the value of V_b we can change the value of V_g, this again will change the value of R.

5.7 The select circuit

Since the number of correlators is considerably large, at least so large that we can not use one output-pin for each correlator, we must find some way of observing all the outputs, but only use one pin. To do this we must use some kind of multiplexer suited

for analog signals. The circuit is shown in figure 5.11 (a). We assume that we have a signal Q and its complement $\overline{Q}$ from a digital shiftregister to control the multiplexer, or select circuit. The operation is quite simple (for once we think digital): If $Q = 0$ the leftmost and rightmost transistors will be open, this will route the voltage X_i to V_{out}. For $Q = 1$ the two middle transistors will be open, and there is a connection between V_{ref} and X_i. What we do then is to let the shiftregister generate a single zero, walking through the register. Thus, only one select circuit will route X_i to V_{out}, all the others set the X_i to a fixed potential V_{ref}. This is done to avoid to much variation on the X_i nodes, so a faster operation is possible.

A simulation of this circuit is shown in figure 5.11 (b). We see that the X_i voltage is routed to V_{out} when $Q = 0$. This simulation does not show how the V_{ref} works, but as we see in figure 5.11 (a), there is no difference in the way V_{out} and V_{ref} is connected to X_i. When one of the connections is active, the other must be passive and vice versa. So V_{ref} will be routed to X_i when $Q = 1$.

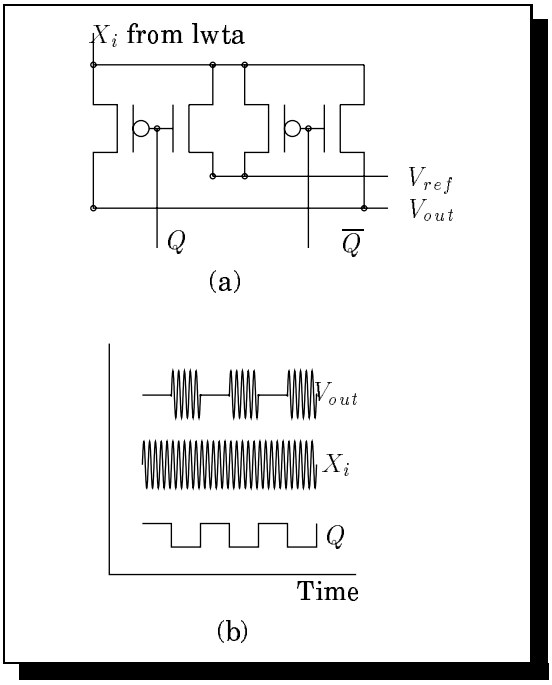


Figure 5.11: The select circuit and a simulation

In (a) we see the select circuit. The X_i is the output voltage from the LWTA, Q and $\overline{Q}$ is output from the shiftregister described in the next section. V_{out} is the output line, this is the one connected to a pin. V_{ref} is a input to the chip, to set the X_i's we do not want to observe. Part (b) shows a simulation of the circuit. The sine wave signal is the voltage on node X_i. As we see the circuit will route the X_i voltage to V_{out} when $Q = 0$.

5.8 The shift-register

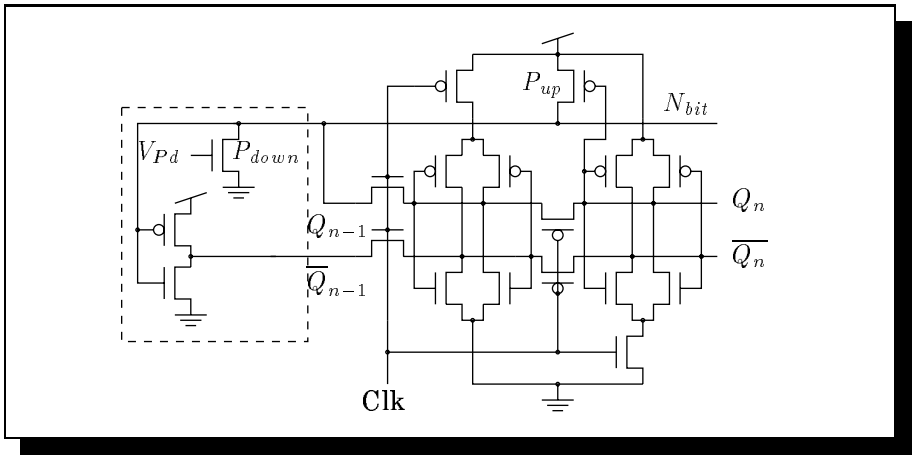


Figure 5.12: A single bit in the shiftregister

The circuit inside the box is in front of the first bit of the register only. The inverter generates the $\overline{Q}$, and the P_{down} transistor generates the next bit N_{bit} together with the P_{up} transistors in each bit of the register. To get this register working, the transistor in the internal four inverters must have a $\frac{W}{L}$ ratio at least four times larger than the clocked pass transistors. The Q and $\overline{Q}$ is used to route one of the output voltages to the V_{out} line. The selected output has $Q = 0$.

As mentioned in the previous section we need a shiftregister to generate Q (and $\overline{Q}$) used in the select circuit. Digital registers of this type are well known, and there are several ways to implement them. It is common to use two clocks, but the one I use have only one clock to control it. This makes the the shiftregister very easy to control. A single bit of this register is shown in figure 5.12. It is described in [14]. In front of the register we need some extra transistors, first we need an inverter to get $\overline{Q}$, and second a pull down transistor to generate the next input bit together with the pull up transistors in every bit of the register. The register works as follows: When the clock starts to run, either the P_{down} or P_{up} transistor will set the line N_{bit} to 0 or 1. The new value depends on what the state of the bits are. It does not really matter, after only a short time we will be in normal operation mode anyhow. If one bit of the register is 0, the P_{up} transistor will generate the next input bit, $N_{bit} = 1$. To get this operation correct we need to control the gate of the P_{down} transistor (V_{Pd}) so that it does not pull down as much as P_{up} pulls up. When the last bit of the register shifts out a 0, there will be no P_{up} transistors on, so P_{down} will generate a new input bit, $N_{bit} = 0$. Thus we have a self-starting shiftregister running as long as the clock pulses are present.

In figure 5.13 is a measured test of this register. With $V_{Pd} = 0.9V$ it generates new

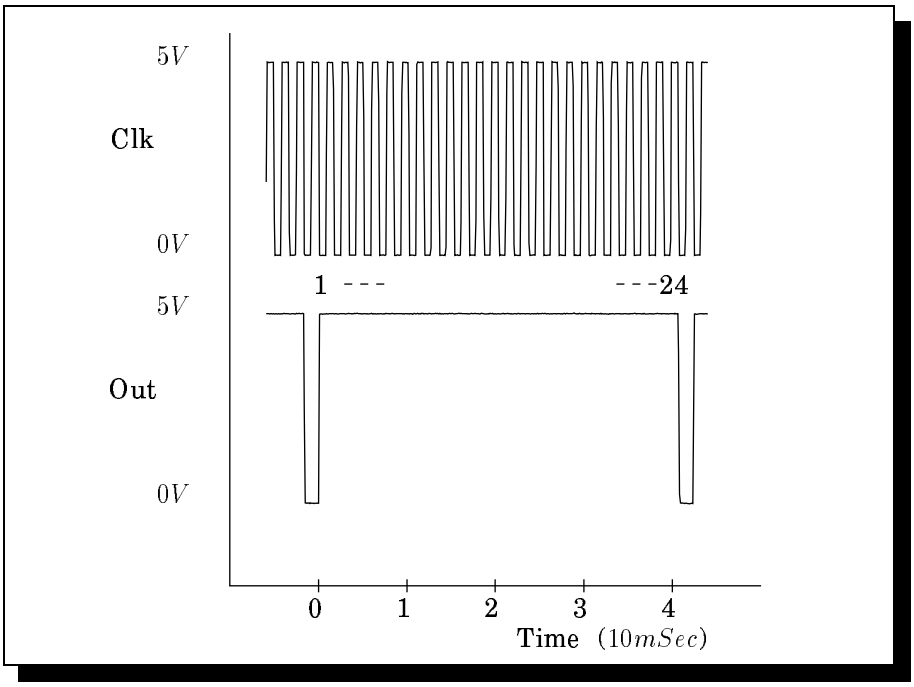


Figure 5.13: Measured output from the shiftregister

The output from the shiftregister is the last bit shifted out. This is an output for testing only. We see that every 25th period we get a zero shifted out.

input bits correctly. The frequency of the clock pulses is 600Hz. When the clock starts the register will behave like expected. At the top of the figure is the clock pulses, alternating between 0V and 5V. The output is the generated next bit (N_{bit}), and stays at 5V most of the time, but as we see there is a pulse (0V) for every 25th clock period.

6
Testing the chip

After the work with the design was finished, the chip was fabricated. This was done in USA, and we got 12 chips for testing. This testing is an important part of the work, although it may be difficult to test larger systems. In this chapter I will present results of my testing.

6.1 Testing in general

To make the testing easier we can take interesting nodes directly out on separate pads, this is also done in some occasions on my chip. To do this we must use some extra pins, which we certainly do not have on every chip.

The equipment required to do these kind of measurements is a story on its own. Since we operate the transistors in weak inversion, the currents we want to measure can be extremely small. Thus we need very accurate equipment. On the laboratory we are well equipped, and this makes a lot of the work much easier.

6.2 Before testing

A lot of the testing is described in the previous chapters (4 and 5). In addition many simulations of the circuits were presented. Those where tests and simulations on *single* elements of the chip. It is important to see that each part of the circuit works as it should before we try to test the system.

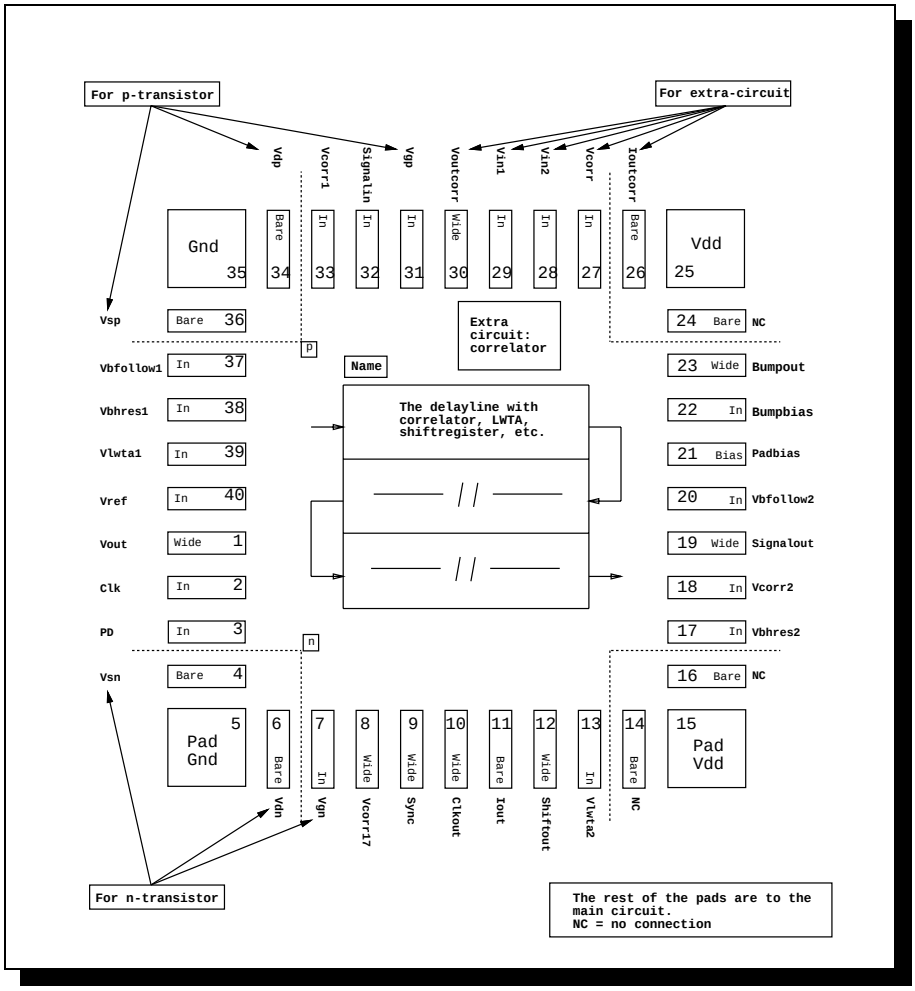


Figure 6.1: Inputs and outputs of the chip

There are as we see different types of pads, for both input and output. A short description of each pin is given in figure 6.2.

Pin no.	Name	Type	Description
1	Vout	Wide	The selected output voltage
2	Clk	In	The clock-pulses to the shiftregister
3	PD	In	Gate-voltage of the pulldown transistor
4	Vsn	Bare	Source voltage of the extra n-transistor
5	PadGnd	Corner	Ground for pads
6	Vdn	Bare	Drain voltage of the extra n-transistor
7	Vgn	In	Gate voltage of the extra n-transistor
8	Vcorr17	Wide	Output voltage of correlator no. 17
9	Sync	Wide	The next shiftregister input bit (N_{bit})
10	Clkout	Wide	The clock at the end of the register
11	Iout	Bare	The selected output current
12	Shiftout	Wide	The bit shifted out of the register
13	Vlwta2	In	2nd. LWTA bias voltage
14	NC	Bare	No connection
15	PadVdd	Corner	Power supply for pads (+5V)
16	NC	Bare	No connection
17	Vbhres2	In	2nd. Hres bias voltage
18	Vcorr2	In	2nd. correlator dc-level bias voltage
19	Signalout	Wide	Signal at the end of the delay line
20	Vbfollow2	In	2nd. follower bias voltage
21	Padbias	Bias	Bias voltage to the pad followers
22	Bumpbias	In	To bump circuit: Out of function
23	Bumpout	Wide	To bump circuit: Out of function
24	NC	Bare	No connection
25	Vdd	Corner	Power supply (+5V)
26	Ioutcorr	Bare	Output current from extra correlator
27	Vcorr	In	Extra correlator dc-level voltage
28	Vin2	In	2nd. input to extra correlator
29	Vin1	In	1st. input to extra correlator
30	Voutcorr	Wide	Output voltage from extra correlator
31	Vgp	In	Gate voltage of the extra p-transistor
32	Signalin	In	Input signal to the circuit
33	Vcorr1	In	1st. correlator dc-level bias voltage
34	Vdp	Bare	Drain voltage of the extra p-transistor
35	Gnd	Corner	Ground
36	Vsp	Bare	Source voltage to the extra p-transistor
37	Vbfollow1	In	1st. follower bias voltage
38	Vbhres1	In	1st. Hres bias voltage
39	Vlwta1	In	1st. LWTA bias voltage
40	Vref	In	Reference voltage to set the unselected output

Figure 6.2: Description of the pins

This figure shows all inputs and outputs to the PASUC chip. Every pin no. has a name, type and short description.

The chip has 40 pins, and a schematic layout of the chip and the inputs/outputs is shown in figure 6.1.The four corners take four pins to V_{dd} and Gnd, and the pads on each side of the corners are Bare pads. So we have 28 pins we can use with all types of pads, and 8 Bare pad pins. On my chip there were some free pins when all the planned inputs and outputs where connected. Some of these free pins are connected to interesting internal nodes, but some are left unused. In addition to the correlator, the extra circuit also contains a *bump circuit*. Pins no. 22 and 23 are connected to this circuit. However, due to a layout error this circuit does not work, so they are not used. The bump circuit is described in [5].

There are two types of input pads on my chip:
- **In** The normal input pad, with a 200Ω resistor in poly.
- **Bias** This is the input to bias all the follower output pads.

Similar, there are one type of output pad:
- **Wide** A wide range follower, for measuring voltages.

In addition a pad for both input and output are used:
- **Bare** This is a direct connection in metal. We can either set a voltage or measure a current.

In figure 6.2 is a description of each pin of my circuit. In addition the corners are special pads for Gnd and V_{dd} for the circuit and the pads. PadGnd and Gnd are connected together, and so are PadV_{dd} and V_{dd}.

Pads like Vbfollow1 and Vbfollow2 is connected to the same node, but this node is a tipped polyline. This enables us to set the delay different for each stage of the line. The following input voltages have tipped polylines : LWTA bias, follower bias, hres bias, correlator dc-level bias. In the tests made on PASUC, the polylines have not been tipped (the same voltage is connected to both ends of the line).

6.3 Testing the chip

To test the function of PASUC as a system is a much harder task than testing the smaller circuits one by one. There are several things we must have in mind when we do a test of the system. Here are some aspects we should consider:
- What is the frequency range of the input signal?
- How should the different bias voltages be set to get a good operation?
- What frequency should the clock have?

On this chip we will try to use input signals in the audio area (<20kHz). It is possible however, that we could have used higher frequencies. But for our purposes this will not be necessary, The main task is to check the function of the chip. Also, we will use the oscilloscope to measure the outputs, since we need the amplitudes. We have a possibility of observing the output currents as well, but our equipment is best fitted to use the first solution. And earlier measurements show us that the voltages will vary enough to get reasonable good results.

The other parameters, setting the bias voltages and the clock frequency, are more a matter of trying different values until we eventually get good results. We can find some voltages by analyzing the circuits and look at the other tests, but small changes in the input voltages might have great influence on the chip performance. In addition

we have the damping problem, that is, the LWTA might find an incorrect winner. All in all this leads up to the fact that the chip is difficult to test, and the tuning of the bias voltages is very important.

6.3.1 Measured results

As we know the correlator gives an output voltage with the same frequency as the input signal. The *amplitude* of this output signal carries the most important information. But the LWTA will not look at this amplitude to select a winner, so any possible results must more or less be read out manually on the scope by observing each output via the select circuit controlled by the shiftregister. This means that we unfortunately not will get any kind of an bar-diagram (FFT-look) as we hoped. In fact, good measured results that clearly differentiate between different frequencies are probably impossible to get.

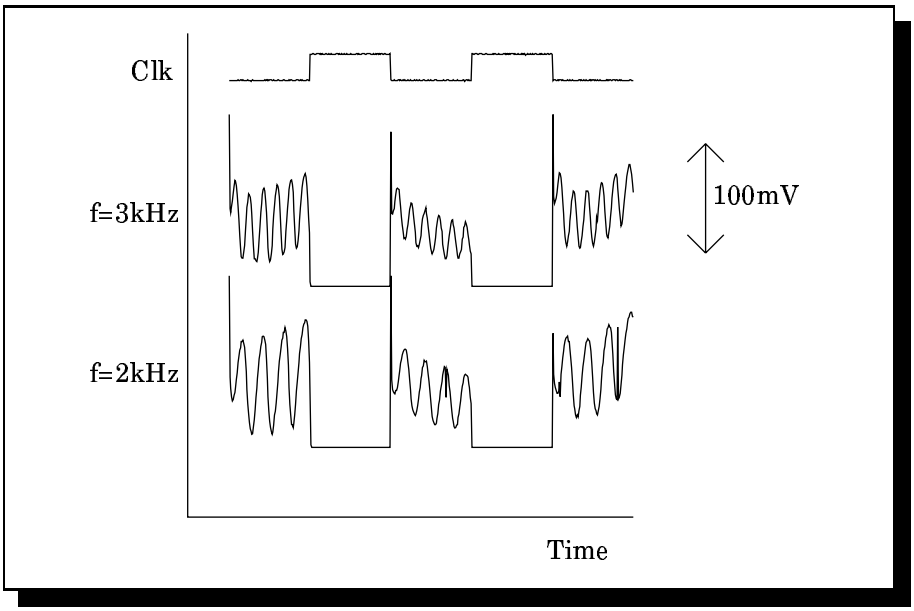


Figure 6.3: Correlator output voltages

The square wave is the clock for the shiftregister. The two ac-curves are the correlator outputs after stage 8, 9 and 10 with input frequencies 2kHz and 3kHz, amplitudes (peak to peak) 3V and dc-level 2.5V. We see hat they have different amplitudes for the different stages and frequencies. The two curves have the same y axis units, indicated with the 100mV arrow. The dc-level of both curves is about 530mV.

We can make a test that indicates that the idea behind the circuit is correct. If we let the clock-frequency be low compared to the input signal frequency we can observe the ac-components in each clock period. In figure 6.3 I have measured the voltage on the

V_{out} line. The two ac-curves are the response for two input sine waves with frequencies 2kHz and 3kHz. From this figure we can see that the correlation outputs from different stages are shifted out to the V_{out} line. In this example I have looked at the signal after stage 8, 9 and 10, from left to right. The clock signal has a frequency of 250Hz. The following bias voltages are used:

- V_{dc}: 2.8V
- V_{PD}: 0.9V
- V_{LWTA}: 1.6V
- V_{bhres}: 0.35V
- V_{ref}: 2.05V

The control voltages to the LWTA circuit will work only to set the different currents from the correlator, thus giving us the opportunity to get the curves in the last figure. The LWTA function seems not to work as we wanted, but as I have already mentioned two reason for this, it is not unexpected. The V_{ref} voltage will influence on both the ac- and dc-level of the output, since this voltage is routed through to the correlator outputs not selected.

The measured curves in figure 6.3 shows the output from only three different stages with two different input frequencies. During the testing I looked at the outputs from all the stages, and the three stages presented in the figure were selected because of the clear difference in the amplitudes of the outputs. The other stages showed similar behavior, but with smaller changes from one stage to the next. We also see that the dc-levels are different in the different stages, but the LWTA will still not work. The LWTA is important if we want nice curves, and the output from the correlator should probably be integrated.

7
Another system : FREDD

Although the main part of this work is the design and testing of the PASUC chip, I have done some related testing on another system. Results from these tests and a description of the chip is presented in this chapter. The chip is designed by Tor Sverre Lande.

7.1 An overview

The PASUC chip is quite small, the delayline has only 24 elements. This second system has a delayline with 240 elements. It is called FREDD, which is short for:

FREquency
Dependent
Delay

Many of the voltages in the line are direct outputs (for example the delayed signal after 56 stages). One stage contains three delay elements as described in chapter 5. Totally we have 80 elements on this chip. In addition to the delay elements each stage contains a correlator, a WTA and a feedback to the delay elements. The idea of the circuit is to control the delay of each element through the feedback, so that we for example have one period in the delayline, independent of the frequency. How this is done is the subject the next section.

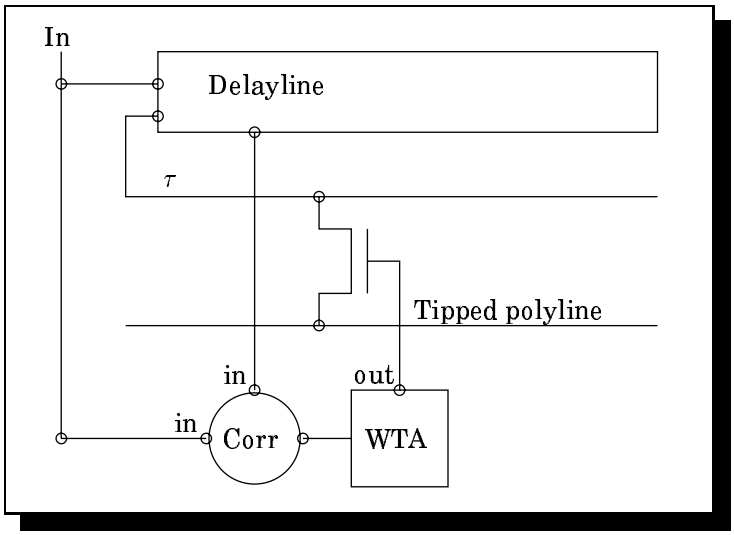


Figure 7.1: A schematic diagram of FREDD

Several correlators are connected to the delayline and the outputs from the WTA-circuit controls the transistor that sets τ to the delayline.

7.2 The chip

A description and explanation of the chip is necessary, but FREDD is not treated in detail as PASUC. There are two main reasons for this: (1) FREDD is not designed by me, and (2) many of the circuits are very similar to the ones in PASUC. Figures showing the circuits of FREDD are not included in this chapter, but can be found in appendix A (figures A.4 and A.5). A schematic diagram of one stage is however drawn in figure 7.1.

7.2.1 The delayline

The line is built up in nearly the same way as on the PASUC chip. The transconductance amplifiers on the PASUC line is the wide range type, but on FREDD the normal type is used. In addition a *Benson diode* is used to improve the performance. See figure A.5 for a transistor diagram. Figure A.4 shows a single stage. This is actually a third order section, but we will now use it as a normal delayline, thus turning the feedback amplifier off by setting $Q = 0V$. The outputs from the line are taken after every third amplifier. We set the delay by changing the value of τ. The τ-line is connected to a pad that we either can observe or set. If we want a normal delayline operation we set this value as done previously on the PASUC chip. In the tests described here however, τ will be set via some feedback transistors controlled by a WTA circuit. The output from each stage of the line is connected (capacitively) to a correlator.

7.2.2 The correlator and WTA

Correlators of different kinds are treated in chapter 4. Here we have used the correlator/anticorrelator circuit from figure 4.11. But only the correlator current is used. The inputs to the correlator is the original signal and delayed versions, thus we perform an auto-correlation. The level of the output current can be biased with the *spont* voltage, or V_{dc} as it is called in chapter 4. The output current is connected to a Winner Take All circuit. (See figure 5.9 for a diagram of a WTA circuit). In addition a capacitor is connected between Gnd and the input of the WTA to integrate the signal. An extra transistor is added to control the WTA function. An n-transistor between the common winner-line and Gnd can be used to adjust the current through the circuit. The gate is controlled by the voltage *WTAbias*.

7.2.3 The feedback

A WTA produces one winning voltage, the others will be nearly $0V$. This voltage is taken as a gate-voltage for the feedback transistors. The τ-line is connected via these feedback transistors to a tipped polyline. To stabilize the τ-line we integrate the signal by adding a capacitor to Gnd. By setting the ends of this polyline to different voltages, τ will be set to different values depending on where the winner is located. This is again decided by the the WTA-circuit. So, depending on where we have the largest output current from the correlator, the delay will change. As we know, the correlator current depends on both the delay and the frequency of the input signal, and by setting fixed values at the ends of the polyline we can get the delay to change as the frequency of the input signal changes. In this way we can have a nearly constant number of taps in a fixed part of a period, independent of the frequency of the input signal.

7.3 How does FREDD work?

The idea behind the chip is quite simple: let the frequency of the input signal decide the voltage that controls the delay. By using this we can get the operation we want: the delayline should hold a fixed part of the period of the input signal, independent of its frequency.

The key to this operation is the feedback transistor. This transistor controls the τ-line, and thereby the delay. To explain how we get the correct transistor to win, we must look back to the operation of the correlator. We know that the output current is not only depended of the phase delay between the two inputs, but also of the amplitudes. As on the PASUC chip, the signal is damped as it gets delayed. On this chip the Correlators early in the line will always win, even if one of the inputs is a little delayed compared with the other. This is a result of the damping of the signal, so even if the signal is delayed exactly one period some place along the, line the early correlator will win. When we increase the frequency we need a larger τ to reduce the delay.

When the circuit was designed, the idea was that when the frequency of the input signal increased, the winning correlation between the input signal and the signal delayed exactly *one* period should move towards the beginning of the line. This is however not what happens. The measured results show that the winning correlator moves away from the beginning of the line. With the start of the polyline at a lower potential than

the end of the line, the circuit work quite well. But the problem is to find *where* we correlate to find the winners. Measured results indicate that the winners are located early in the line, thus we do not have correlation between the original signal and the previous period. Unfortunately we do not have enough nodes available as outputs to investigate this problem as detailed as we want.

7.4 Testing of the chip

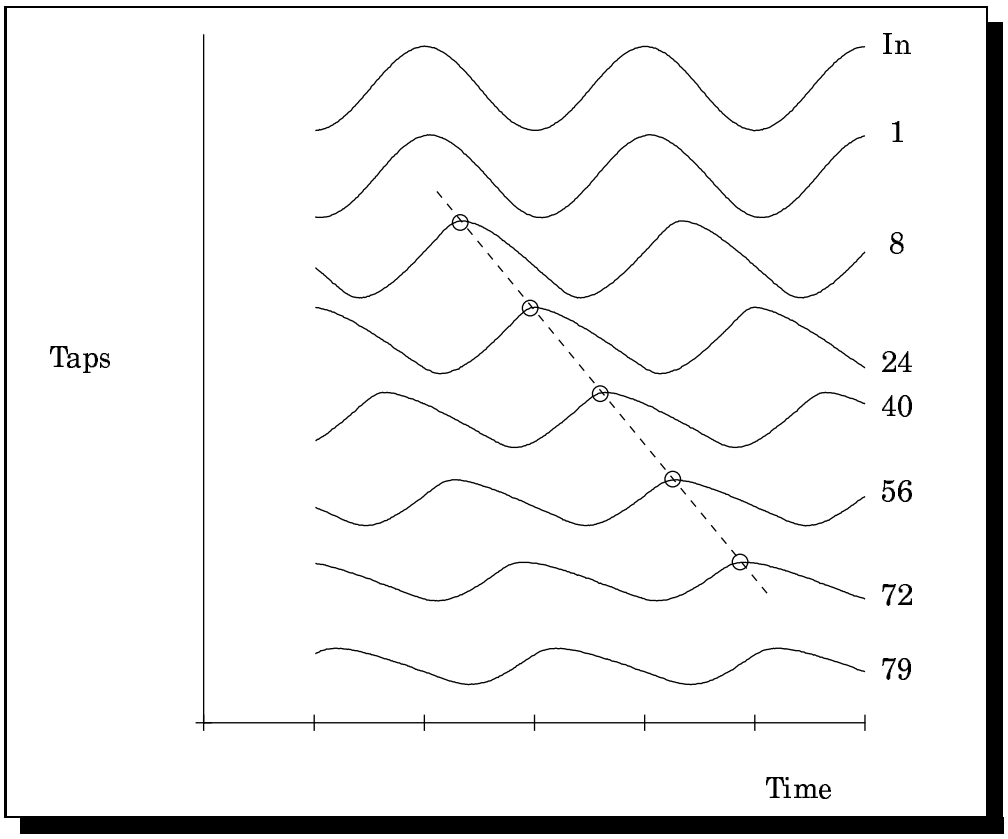


Figure 7.2: Measured taps in the delayline

The first curve shows the original input signal. The other curves are taps in the line. The number refers to the stage number, that is, if we multiply it by three we get the number of amplifiers before the tap. As in the PASUC-line, the signal becomes damped for large delays.

The delay line in FREDD is longer than the one in PASUC. The line has several taps to output pads, thus giving us the opportunity to observe the contents of the line. In figure 7.2 we can see the contents of the line for a fixed delay per element. We observe

that the delay between equally spaced taps is constant (the dashed line).

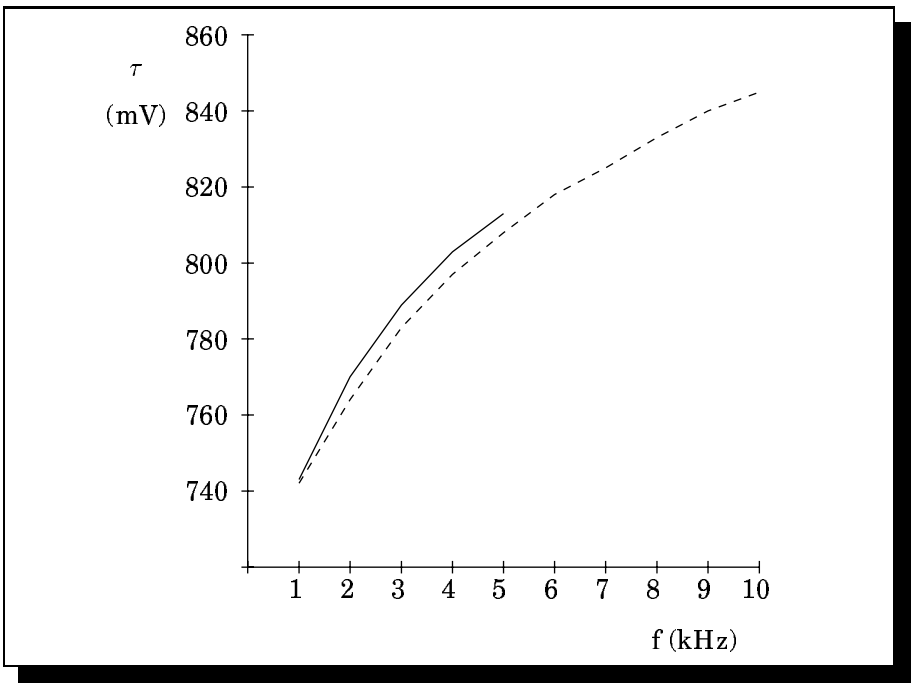


Figure 7.3: Measured variation of τ

The input frequency is first swept from 1kHz to 5kHz (solid curve), then from 1kHz to 10kHz (dashed curve). τ is a measure of the size of the delay in each element.

More interesting is it to see how the τ line voltage changes when the input signal changes frequency. This might seem to be straightforward to measure, but unfortunately it is not. There are several voltages that must be set correctly to get the correct operation. Figure 7.3 shows this behavior. For the 1kHz to 5kHz sweep the following voltages are used:

- Start of polyline: 0.7V
- End of polyline : 1.25V
- WTAbias : 0.85V
- Spont : 1.5V

For the 1kHz to 10kHz sweep the voltages are changed to:

- Start of polyline: 0.6V
- End of polyline : 2.0V
- WTAbias : 0.86V
- Spont : 1.6

For both curves the input signal is a sine wave with dc-level 2.5V and an amplitude (peak to peak) of 4V. The bias voltage to the followers in the outputpads is set to 1.3V. What we observe is that if we want the circuit to operate for a larger frequency range,

it is difficult to maintain the correct steepness of the τ-curve. This means that the tipped polyline have to small voltage differences between the different taps to the τ-line.

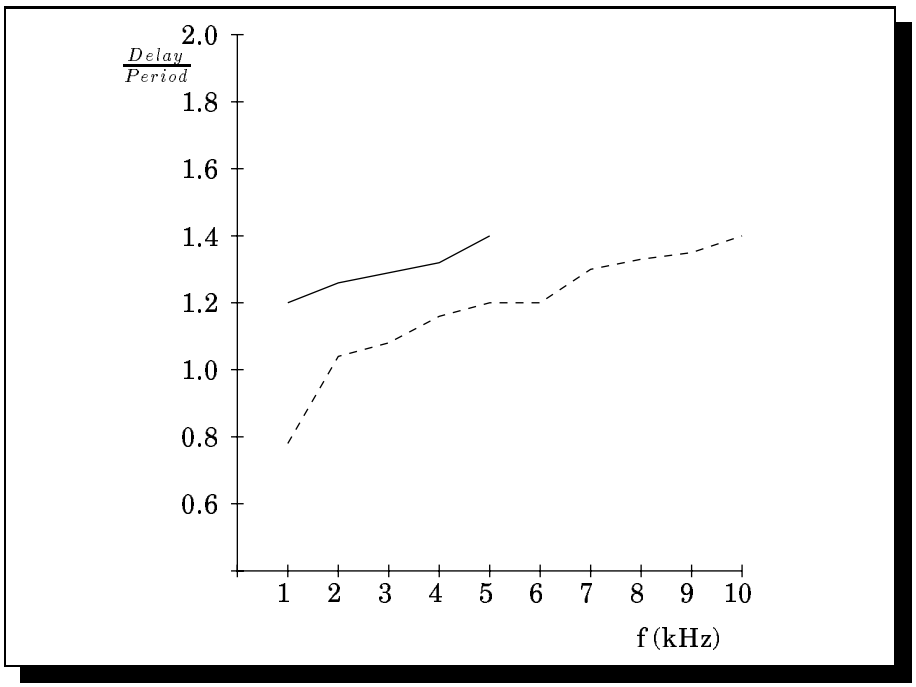


Figure 7.4: Measured $\frac{Delay}{Period}$

In this figure we have plotted the ratio $\frac{Delay}{Period}$ against the input frequency. What we want is the ratio to be constant, but as we see this is not the case. If we want exactly one period in the delay line the ratio should be equal to 1. The solid line shows the result for the 1kHz to 5kHz sweep, and the dashed line the 1kHz to 10kHz sweep.

By setting the ends of the tipped polyline to different values we can decide how τ should change. This chip will work for only a quite small frequency range, so we have tested it for a few quite low frequencies only. As mentioned we want to have a fixed number of taps per period of the input signal. This number should not depend on the frequency of the input signal. What we need then is to adjust the delay, a low frequency means a large delay, and a high frequency means a small delay. The voltage τ sets the delay, an increasing τ implies a decreasing delay. With the feedback system described earlier we can get this adjustment of τ. Figure 7.4 shows a plot of the ratio $\frac{Delay}{Period}$. The bias voltages are the same as for figure 7.3. The 1kHz to 5kHz sweep gives a ratio from about 1.2 to 1.4, the value increases with increasing frequency. This implies that the changes in the τ-voltage are to small. The curve for the 1kHz to 10kHz sweep shows the same behavior. This figure also illustrates that only small changes in bias voltages can have great influence on the performance. In figure 7.3 we see that the two curves

have a small difference up to 5kHz (at 1kHz they are almost equal), but in figure 7.4 the curves clearly shows different ratios.

8

Problems and improvements

The chip which is constructed in this work has not all the best solutions. A number of things could have been done in better ways. In this chapter I suggest some improvements and possible ways to expand the chip for better performance.

8.1 Problems with the PASUC chip

The chip described in this thesis is a prototype. We wanted to see how the correlator-circuit could be used to process analog signals. But as I have briefly mentioned previously in the text, there are some problems that we discovered during testing of the chip. I will now describe these problems, and later suggest some alternative solutions that could be better.

The chip was designed to find different frequency components in an analog signal. The test results from chapter 6 show that the circuit has some problems with the LWTA. Because of this I have used pure sine waves as inputs, and the function of the chip is reduced to find the main frequency of the input signal.

8.1.1 The unscaleable correlator

The correlator circuit is both simple and powerful. We recall the function (from equation 4.10) for the output current for the circuit given in figure 4.1:

$$I_{out} = \frac{I_1 I_2}{I_1 + I_2} \tag{8.1}$$

As the simulations and measured results shows, this circuit will give us information about two signals being in phase or not. However, the amplitude of the output is indeed dependent of the amplitudes of the input signals as well. This is also clearly seen from equation 8.1. This is what I mean with the term unscaleable. So this behavior is not at all surprising. If we want the output of the circuits to give us information about the phase of the input signals only, we have to make sure that both inputs to the correlators are scaled equally. The two inputs to one single correlator do not have to be equal, but the corresponding inputs throughout the 24 stages in the circuit must be the same as on the first one. This is however not so straightforward to do as we might believe, we do not have a correlator that normalize the inputs. There exists circuits that perform a kind of division, so further investigations should be done to find a better correlator.

8.1.2 The damping problem

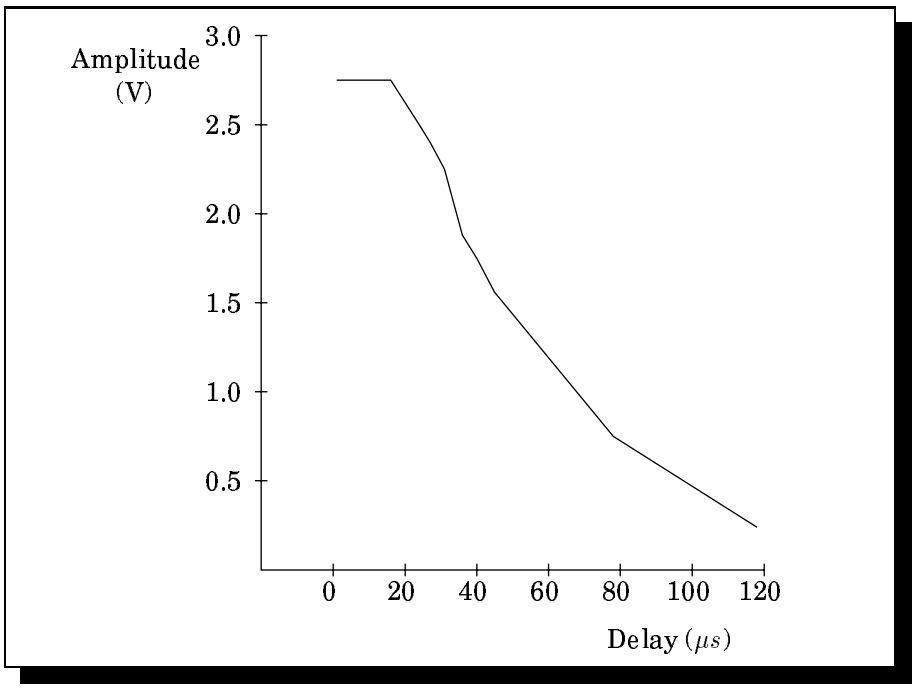


Figure 8.1: Damping through the delay line

The figure shows the amplitude of the signal plotted against the delay at the end of the delayline. The input signal is a 10kHz sine wave with amplitude 3V (peak to peak), and dc-level 2.5V. When the signal is one period delayed (that is 100μs), we see that the amplitude is only 0.5V.

The delayline makes delayed version of the original input signal. The further away from the start of the line, the larger are the delay. In a perfect world this delayed

versions would be exact copies of the input, but this is not the case here. If we want the delay to be considerably large, the delayed versions become damped compared to the original signal. In figure 8.1 we clearly see this for an increasing delay. We observe that when the delay increases, the amplitude of the output signal decrease. We also know that the output signal not only get smaller amplitudes, but also will be misshaped compared to the input sine wave. (See figure 7.2.)

8.1.3 Who is the winner?

Because of the damping and the unscaleable correlator the output currents from the correlator will depend on where we are in the delayline. In figure 5.8 we see that larger delays make smaller signals. So when a winner should be an output referring to two signals in phase, this will certainly not always be the case. In addition the correlator produces ac-voltages, and the outputs with the largest amplitudes are of interest. The measured curves in figure 6.3 are inputs to the LWTA circuit. The correct winners should be the ones having the largest amplitude, but the LWTA will only look at the dc-level. So the LWTA part of the circuit is not working as intended.

8.2 Improving and expanding the chip

The problems described in the previous section are closely related to each other. For example if there were no damping, we would not need the normalized correlator.

To avoid the damping problem we could use an alternative signal representation. One technique is to use pulse-coding of the signals. This method is used in [9]. On my chip this would probably be possible as well, but the amount of information per timestep would be reduced. If we use pulses as inputs to the correlators, the result would be more like a digital signal, since the pulse-coding only use two voltages: 0V and 5V. Thus we could not get a result like the diagram in figure 5.2, which is what we wanted.

On my chip I use a wide-range amplifier. This amplifier has larger gain than the simple version, this is why I chose the wide-range solution. Measurements (from FREDD) show that a normal amplifier with a Benson diode can be used to give delaylines that have good characteristics, so this last solution is probably the best one. (See figures 7.2 and A.5). The correlator I use is the symmetrical version. This is probably not the best solution. The asymmetrical correlator would be good, since all the correlators are connected to the rest of the circuit in the same way. But the output from the correlator is used as input to the LWTA. The LWTA should have dc-voltages as inputs. Thus we should integrate the output from the correlator, and this leads to a new selection. We want a correlator that responds to both half-periods of the input signal. Thus the improved symmetric correlator might have been the best solution, together with an integrating capacitor between the input to the LWTA and Gnd. (FREDD has a solution like this).

My delayline consists of 1st. order low-pass filter sections. These sections can not amplify the input signal. To do this we need higher order sections, 2nd. order and 3rd. order. These types of circuits have the ability to amplify the frequencies below the cutoff-frequency, thus we can maintain the signal value longer than with 1st order

sections.

All the discussion in this thesis has treated processing of one signal only. It could be interesting however, for example to compare two signals and measure the correlation between these two signals (crosscorrelation). The solution is not to let two external signals be connected to correlators. This will not work because we have no information about the time dependence between the two signals. What we perhaps could do is to make two delaylines, each computing the auto-correlation of one signal, and then correlate these results. Before this can be done further testing and improvement on my chip must be done.

8.3 FREDD – problems and possible improvements

The second system I have tested shows better test results than the ones from PASUC. But FREDD is not perfect either, so some comments on this circuit are necessary as well.

8.3.1 Some suggestions

The feedback is the clue to FREDD's operation. If we want exactly one period in the delayline the curves in figure 7.4 must be constant (at one). But this would not be possible, since the system would not be stable. This is because the signal would have the same winners for all frequencies (since exactly one period should be in the line), thus the value of τ could not be changed. If we want the line to hold exactly one period we must change the circuit, and I give a suggestion in 8.3.2. Thus the following suggestions will not make a perfect circuit, but perhaps improve the performance.

The contents of the delayline is decided by the value of τ. Since τ is set by transistors connected to a tipped polyline, the resistance of poly is important. If we want to improve the results, we must have larger potential difference between the taps that control τ. This should be possible (at least for a limited frequency range), by giving the polyline some extra length between the taps. This can not get the circuit to hold exactly one period in the delayline, but might make the circuit work a little better. The measured results so far indicates that the winners are located in the first part of the line. Therefore we might not need correlators/WTA-circuits connected to all of the delayline. This would save area and free extra space to make longer polylines.

As the measured results shows, the circuit performance will be best over a small frequency range. For frequency ranges extending 10kHz the performance will drop. To get the circuit to work it is necessary to adjust the bias-voltages very precisely, and for optimal performance the bias voltages for one frequency range is not the same as for another. In addition to the voltages I have controlled, there are one extra input to FREDD. As mentioned the delayline is built by using 3rd. order sections, and this implies a feedback amplifier controlled by a voltage Q. All the measured results presented have Q=0V, that is, the feedback turned off. Tests that include Q should however be made. The right value of Q might reduce the damping problem (by amplifying the signal a little). In this way we could perhaps get winners from a larger part of the line.

8.3.2 A new version

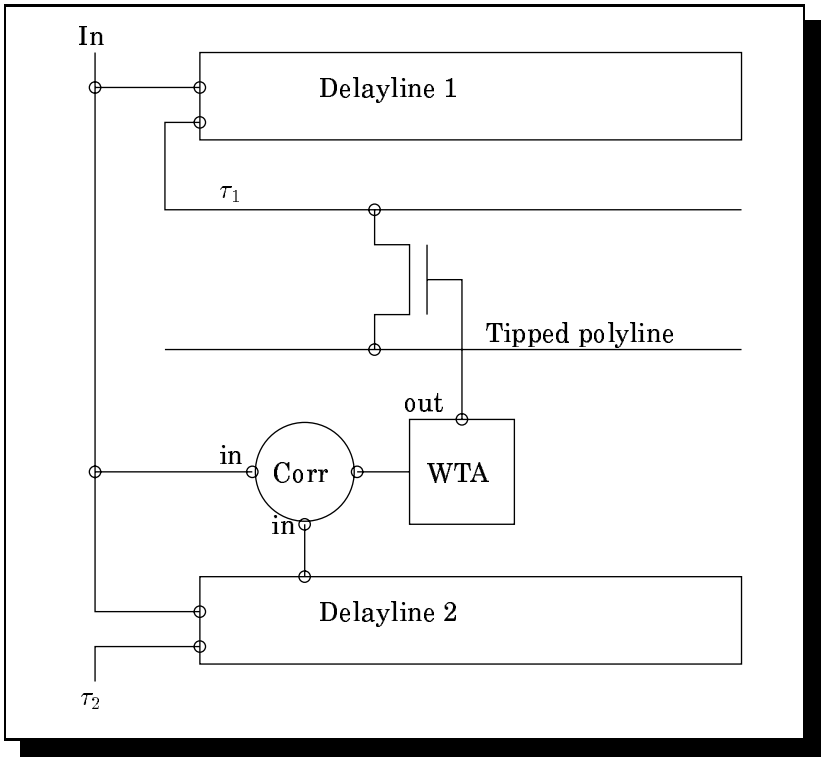


Figure 8.2: A new version of FREDD

This circuit has two delaylines, one for holding the signal and one for computing the correct value of τ. We no longer have the feedback transistor to control τ.

As I mentioned, FREDD will not be able to hold exactly one period of the signal within the line due to the way it is designed. As a possible solution to this problem I suggest a new version, shown in figure 8.2. The idea is quite simple: Instead of having one delayline to hold the signal and compute τ, we use two delaylines. One delayline holds the signal and another line computes the value of τ. In this way we have no feedback-transistors to control the delay, thus it should be easier to get the correct values for τ. This circuit is not designed or simulated yet, so I have no results to present. It is only a suggestion of a principle that might good performance.

8.4 Future work

I feel that a lot of the issues mentioned in this chapter should have been investigated, simulated and tested more. This is particularly the case for the discussion

about FREDD. But as we all know we do not have an unlimited amount of time, so further work will have to be done later.

9
Conclusion

The work with the development, construction and testing of an analog CMOS circuit has given me a lot of experiences. Many things could probably have been done in other and better ways, and it is easy to forget obvious things in front of the screen (For instance the pad bias....). In this chapter I will give a summary of this work.

9.1 The chip

I have tried to design an analog continuous time signal processing chip. The chip tries to find the different frequency components of the input signal. There are different ways to do this, and in digital systems the use of various FFT algorithms are common. On this chip however, we take a different approach. Instead of using large digital circuits that *can* be made, we start with analog circuits that are quite simple, and use their built-in properties. The correlator is a good example of this. It does not give us the traditional correlation function, but we use it anyway. It is very simple, only four transistors are required to compute the output current. This way of building circuits gives us both advantages and disadvantages. We are in a way stuck with some given functions, and these may not be "perfect". What we try to do is to build the system in such a way that it behaves correctly even if the circuits are not optimal. On the other hand, when we use these small circuits we can have a lot of them, so parallel computations are possible. The computing part of my circuit is not controlled by external circuitry or clocked. I have a clock input to the circuit, but this is used only to multiplex all the outputs into one single output pad.

In addition to the design and test of PASUC, testing of a larger system is included as a part of this thesis. The FREDD chip is related to PASUC, many of the internal

circuits are the same. But as we know it performs a totally different task. This shows how powerful correlator-based circuits can be.

9.2 Experiences

The solutions we made on this chip did not always prove to be the best. In spite of this, we got a lot of measured results from the chip, and came up with new ideas of how things could have been done. In all of the phases of this work, from designing the chip to testing it, I have learned that we never should say that this is the ultimate solution. It seems to be a corollary to Murphy's law : The day after you sent the chip for processing, you come up with a better idea.... The measured results presented in this thesis are all taken from the second version of this chip. The first chip that was sent for processing did not work at all, due to a layout error.

The analog CMOS technique makes it possible to design circuits with large computational power compared to the size of the circuits. The small currents flowing in the circuits lead to small power dissipation. The digital systems are and will be useful, but there might be cases where the analog systems can be better solutions. The chips described in this thesis show that even small systems/circuits can perform quite complex jobs. Finally a couple of numbers that I think tell us a lot: Analog computational systems can have up to 100 times the computational power per area unit compared to digital systems, and as little as 1/10000 of the power dissipation.

Bibliography

[1] Andreou, A.G., Boahen, K.A., Pouliquen, P.O., Pavasović, A., Jenkins, R.E. & Strohbehn, K.
Current Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems.
IEEE Transactions on Neural Networks, Vol. 2, No. 2, March 1991.

[2] *Aschehoug og Gyldendals Store Norske leksikon, 2. utg.*
Aschehoug og Gyldendal, 1988.

[3] Baher, H.
Analog & Digital Signal Processing.
John Wiley & Sons, 1990.

[4] Broch, J.T.
Principles of Analog and Digital Frequency Analysis.
Tapir, 1981.

[5] Delbrück, T.
'Bump' Circuits for Computing Similarity and Dissimilarity of Analog Voltages.
International Joint Conference on Neural Networks, pp.475-479, Seattle, July 1991.

[6] Disney, W.
Donald Pocket no. 1-136.
Hjemmet – Pocketbokforlaget.

[7] Kernighan, B.W. & Ritchie, D.M.
The C Programming Language.
Prentice Hall, 1988.

[8] Kuo, F.F.
Network Analysis & Synthesis, 2nd ed.
John Wiley & Sons, 1965.

[9] Lazzaro, J. & Mead, C.
Silicon Models of Auditory Localization.
Neural Computation, Vol. 1, pp.47-57, 1989.

[10] Lazzaro, J., Ryckebush, S., Mahowald, M.A. & Mead, C.
Winner-Take-All Networks of O(n) Complexity.
Caltech Computer Science Department Technical Report Caltech-CS-TR-21-88.

[11] Lyon, R.F. & Mead, C.
An Analog Electronic Cochlea.
IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 36, No. 7, July 1988.

[12] Mahowald, M.A. & Delbrück, T.
Cooperative Stereo Matching Using Static and Dynamic Features.
From *Analog VLSI Implementation of Neural Systems*, edited by Mead, C. & Ismail, M.
Kluwer Academic Publishers, 1989.

[13] Mead, C.
Analog VLSI and Neural Systems.
Addison-Wesley, 1989.

[14] Mead, C. & Delbrück, T.
Scanners for visualizing activity of analog VLSI circuitry.
Analog Integrated Circuits and Signal Processing, Vol. 1, No. 2, Oct., 1991.

[15] Mead, C. & Mahowald, M.A.
A Silicon Model of Early Visual Processing.
Neural Networks, Vol. 1, 1988.

[16] Mead, C., Arreguit, X. & Lazzaro, J.
Analog VLSI Model of Bineural Hearing.
IEEE Transactions on Neural Networks, Vol. 2, No. 2, March 1991.

[17] Millman, J. & Grabel, A.
Microelectronics, 2nd. Ed.
McGraw Hill, 1988.

[18] Nundal, A.
VLSI Implementasjon av Fast Fourier Transform.
Cand. Scient. thesis, Dept. of Informatics, UiO, 1986.

[19] Russel, G. & Sayers, I.L.
Advanced Simulation and Test Methodologies for VLSI Design.
Van Nostrand Reinhold (International), 1989.

[20] Sze, S. M.
Physics of Semiconductor Devices, 2nd. Ed.
Murray Hill, 1981.

[21] Van Valkenburg, M.E. & Kinariwala, B.
Linear Circuits.
Prentice Hall, 1982.

[22] Watts, L., Kerns, D.A., Lyon, R.F. & Mead, C.
Improved Implementation of the Silicon Cochlea.
IEEE Journal of Solid-State Circuits, Vol. 27, No. 5, May 1992.

[23] Wenstøp, Ø.S.
Digital signalbehandling – en innføring.
Universitetsforlaget, 1986.

[24] Weste, N. & Eshragian, K.
Principles of CMOS VLSI Design.
Addison Wesley, 1985.

A
Complete circuit diagrams

The circuit diagrams presented earlier in this thesis are showing one single element at a time. In this appendix I will give complete circuit diagrams for PASUC, and some diagrams of the FREDD chip.

A.1 PASUC : The elements in one stage

The chip contains 24 stages connected together. In figure A.1 and figure A.2 we see how one stage is constructed. Crossing lines in the figures are connected if there is a dot at the crossing point.

A.1.1 Comments to the upper part of a stage

At the top we have the wide-range transconductance amplifier connected as a delay-element. The delay is set with the size of the capacitor C_2 and the $V_{bfollow}$ voltage. V_{out} from one stage is V_{in} to the next stage in the line. In the first stage the global input signal (an input pin) is connected to V_{in}.
At the bottom of the figure the correlator is drawn. The global signal input and the delayed output have capacitive coupling via C_1 and C_2 to V_1 and V_2. V_{dccorr} sets the dc-level for the correlator output. I_{out} is connected to the LWTA, thus producing an output voltage from the correlator.

A.1.2 Comments to the lower part of a stage

From the correlator we get the I_{in} to the LWTA. To the right of the LWTA circuit is the *hres* and *bias* circuits. The two hres transistors gets their gate-voltages from the bias circuits on this and the next stage. V_{bhres} sets the resistance for the hres.

The output voltage after the LWTA operation is connected to the select circuit in the middle of the figure. This circuit will route a voltage through to the output of the chip. The correct voltage V is selected with the shift register at the bottom. The circuit inside the dotted box is in the first stage only. It produces the right inputs to the shiftregister. V_{Pd} will generate a zero when there are only "ones" (5V) in the register, otherwise a "one" will be generated by the pull up transistors. The register is clocked with Clk, and shifts a zero through itself to select all the LWTA output voltages in turn.

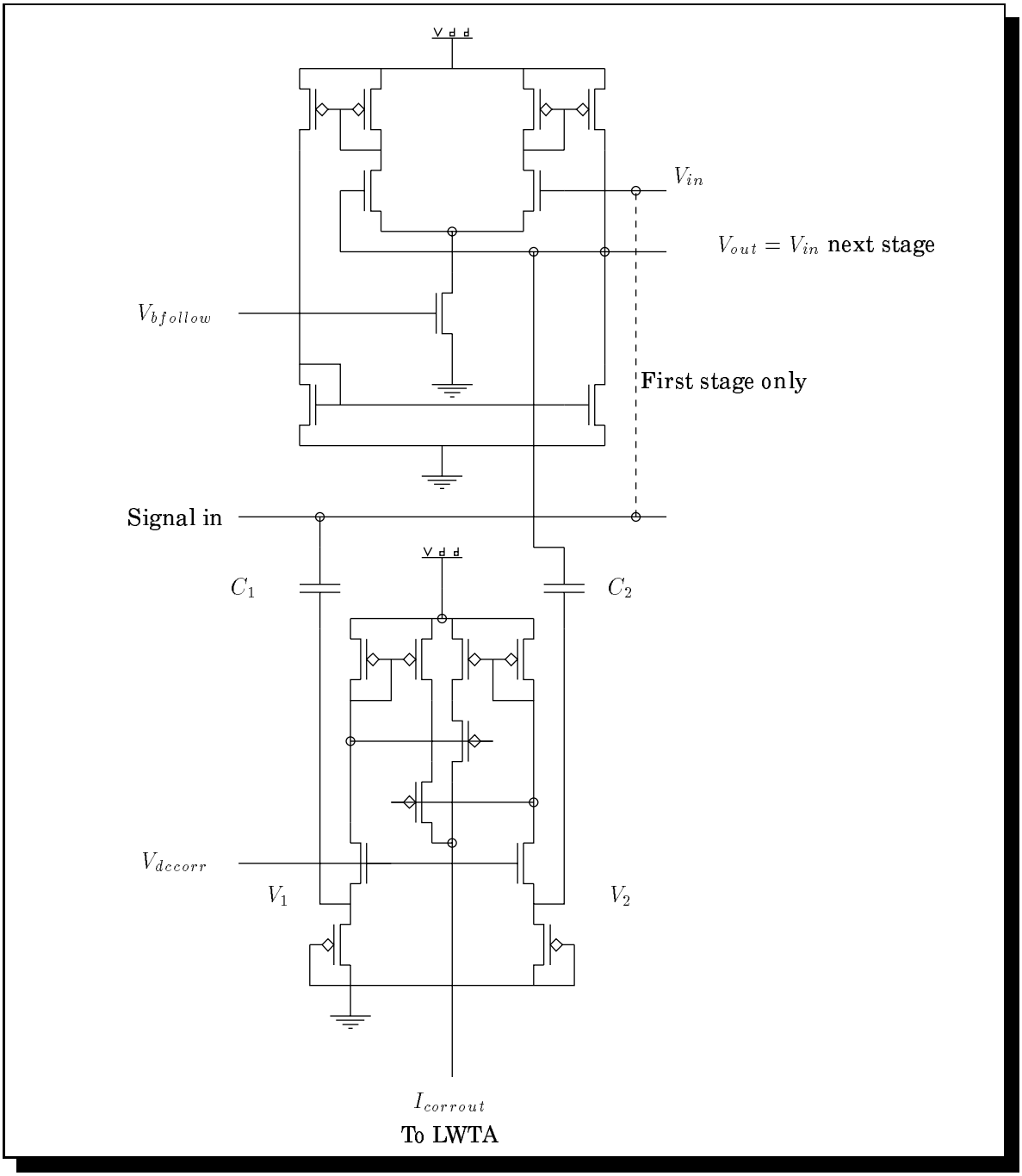


Figure A.1: PASUC : A single stage, upper part

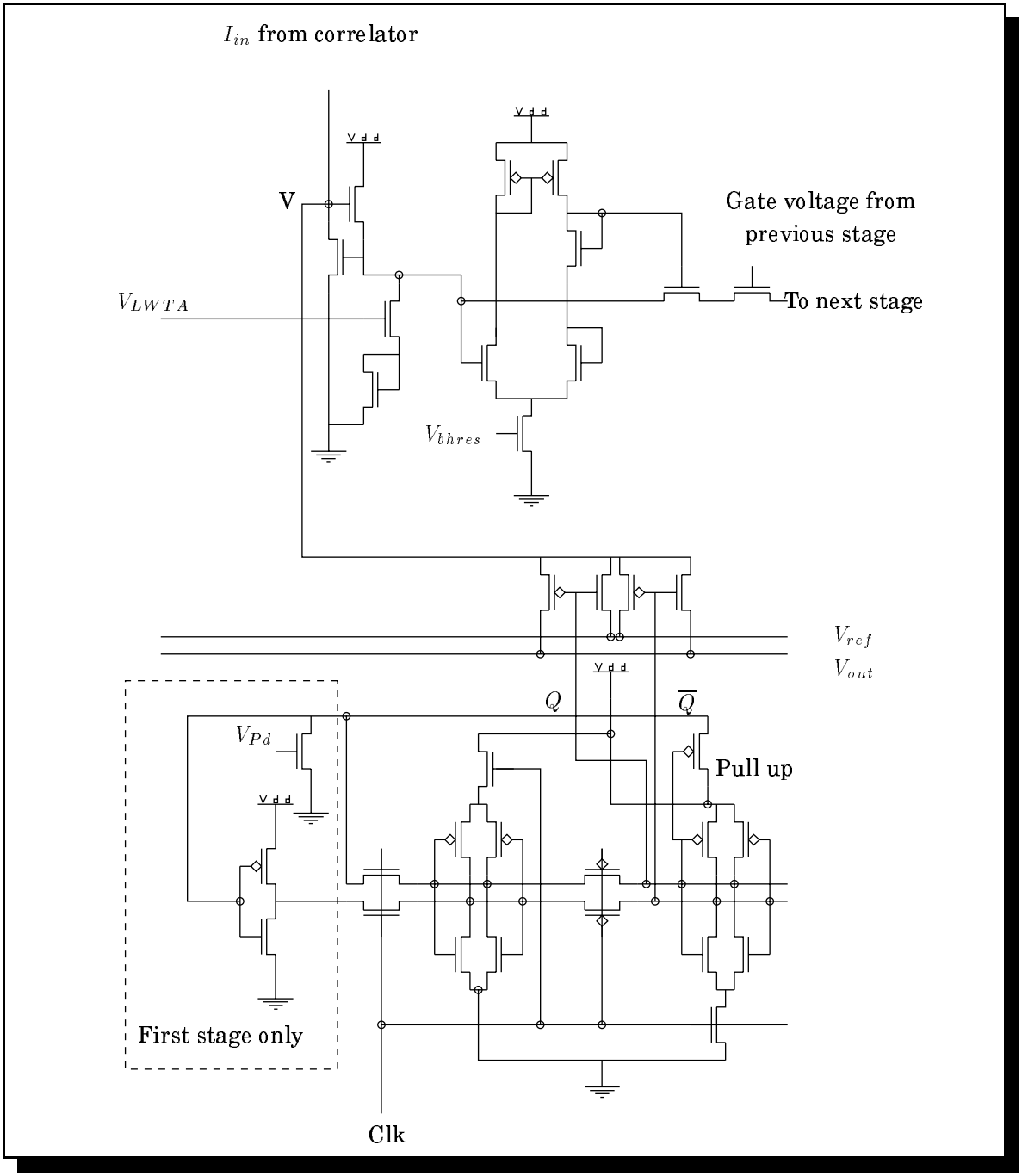


Figure A.2: PASUC : A single stage, lower part

A.2 FREDD : The chip

I have not included any figures of the circuits in FREDD in chapter 7. In this section a diagram showing the inputs/outputs and a transistor diagram of one stage is drawn.

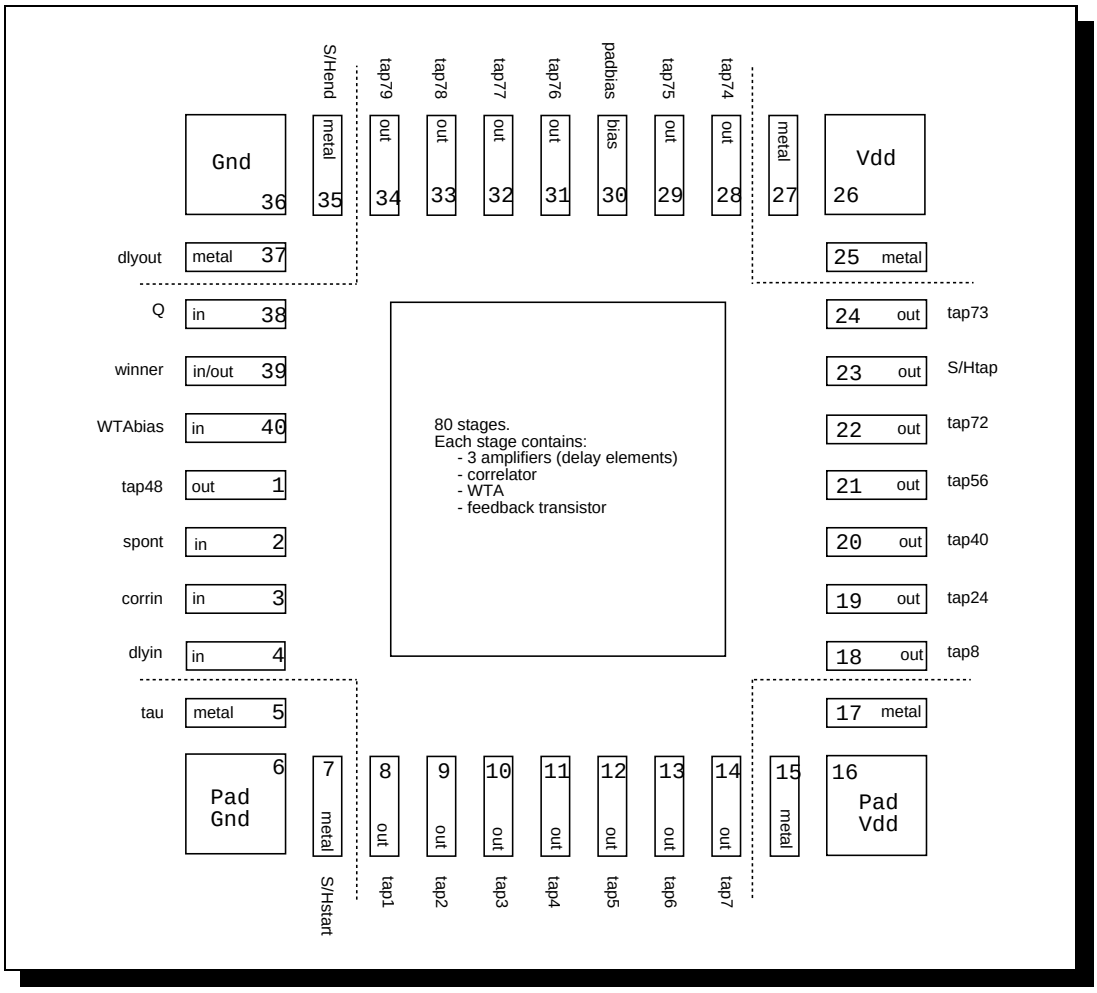


Figure A.3: FREDD : Inputs and Outputs

A.2.1 Comments to the Inputs/Outputs

The following is a short description of the inputs and outputs to the chip.
Inputs:
Q : The feedback control in the 3rd. order section. When Q=0 the circuit works as a normal delay line.
winner : Input or output. We can observe the winning voltage or bias this node in the

circuit.
WTAbias : Controls the WTA operation.
spont : Sets the dc level of the output current from the correlator.
corrin : The first input to the correlator.
dlyin : Input to the delayline. Should be externally connected to corrin.
τ : Input or output. We either observe the τ computed in the circuit (from the tipped polyline), or we can set the value to get accurate delayline characteristics.
S/Hstart and **S/Hend** : The start and end of the tipped polyline. S/H is short for Sample and Hold, the circuit will open only one feedback transistor at a time, and select a value for τ depending on the frequency of the signal.
padbias : Bias voltage for the followerpads.
Outputs:
tapnn : The delay line voltage after stage nn.
dlyout : The voltage after the last amplifier in the delayline. Because of a layout error this pads is not connected. This does not cause any problems since we can observe tap79 which is very close to the end of the line.
The pads used for both inputs and outputs are described under inputs.

A.2.2 Comments to a single stage

The corrin is the first input to all the correlators. The other input is a tap from the delay line. On the first stage (or externally as we have done here) dlyin and corrin should be connected together. The following are global lines connected to all stages:

- winner
- spont
- WTAbias
- Q
- τ

The voltage on the output of the third amplifier is connected to the input of the first amplifier on the next stage. We have a correlator/anticorrelator, but only one output current is used (corr). The other (anti) is not connected. We can get a third order low-pass filter by using the feedback amplifier. In all the tests presented this thesis Q is set to zero, thus we have a normal delayline.

Figure A.5 shows a simple transconductance amplifier with the Benson diode included. This is the amplifier used in the delayline on FREDD, with $V_b = \tau$.

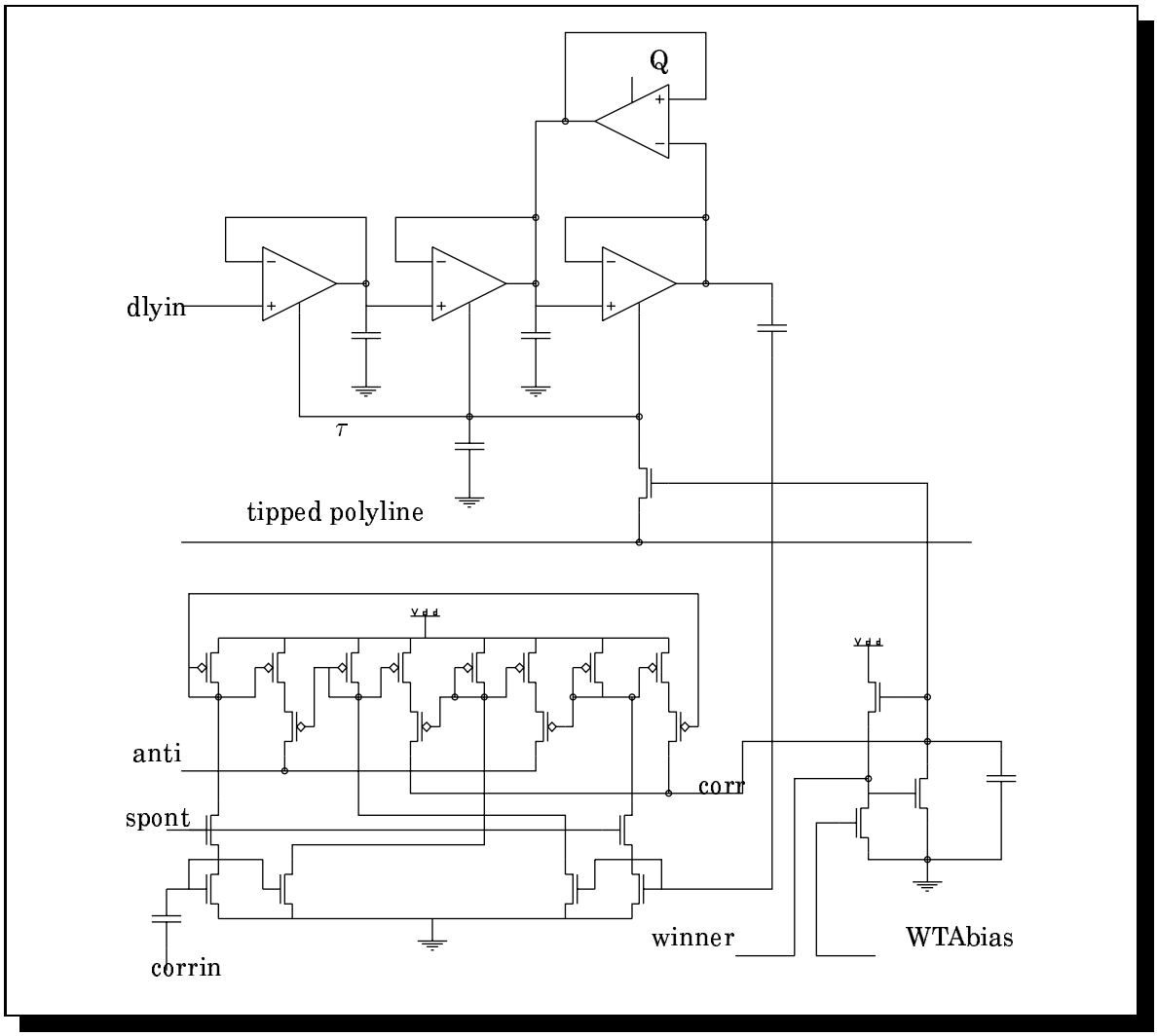


Figure A.4: A single stage on FREDD

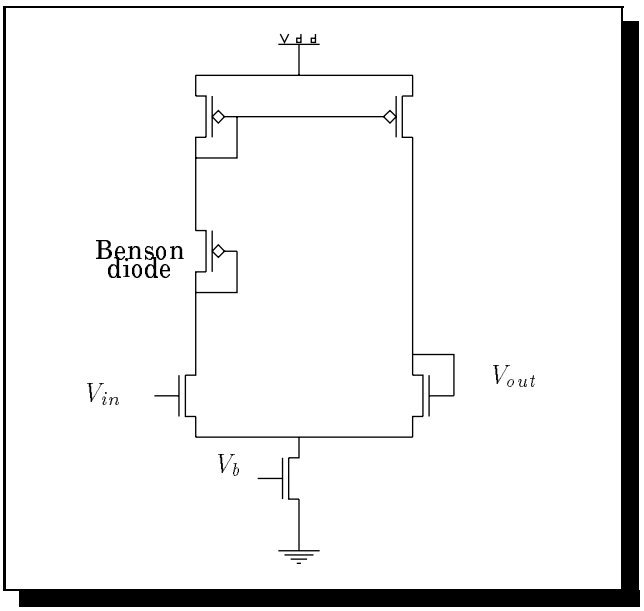


Figure A.5: A simple transconductance amplifier with Benson diode

B
Tools and instruments

A lot of the work in this thesis required different tools and equipment. During the first period the need was mainly software tools such as layout and simulation programs. Later I needed a lot of instruments for the testing of the chips.

B.1 Tools

To make the layout of this chip I used the program *xwol*. It is a program that is easy to use, but not of the most advanced layout programs. It is however easy to get a netlist which one can compare with netlists from the simulator *analog*. Doing this makes it easier to find errors in the layout, and I have compared all the circuits with transistor diagrams from analog. No errors where found. This is done on single elements, like the follower or correlator. To put these elements (cells) together I have used the *wolcomp* system. This system makes it possible to place several cells together, to rotate or mirror cells, etc. Thus it is quite easy to compose new cells that are several small circuits (for example to make one stage as a cell). Then we can make an array of 8 cells to form a 8-stage line, and then put together 3 of these lines to make the final 24-stage line. The routing is the last thing we do, I have done this manually. The circuit has also been checked for layout errors according to the technology used by *xwol*. Simulation is done by the simulator *analog*, which seems to give good results compared to measured data (at least where I have had the possibility to compare). The program *xview* makes it easy to make nice PostScript figures of the measured results (taken via the GPIB bus). The drawing program *until* is used to edit the figures, e.g. remove uninteresting text, and add other text (in LaTeX-code if we want).

I have written this thesis using the *emacs* editor and the LaTeX processing system. I

am sure that spelling errors occurs, but I have tried to correct the text both manually and with the program *ispell*.

B.2 Instruments

The testing of the chips are done in our VLSI-lab. Here is a list of the instruments used:

- *Keithley 230* Programmable voltage source.
- *Keithley 617* Programmable electrometer.
- *HP54503A* Digital 500MHz oscilloscope.
- *Fluke 8050A* Digital multimeter.
- *HP8116A* Function generator.
- *HP3324A* Function generator.
- *Mascot 719* Voltage supplier.
- *Oltronix B603D* Voltage supplier.

In addition I have used a microscope for a check of the chip placement in the 40 pin package. Also a nice testbox for the chip is made here. Inputs and outputs to the chip are easily available, so are the possibility to set up to ten voltages as inputs. The HP- and Keithley-instruments are connected to a GPIB bus and a SUN 3/60.