NTNU – Trondheim
Norwegian University of
Science and Technology

8-bit 50ksps ULV SAR ADC

Fredrik Hilding Rosenberg

Master of Science in Electronics
Submission date: June 2015
Supervisor: Trond Ytterdal, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Abstract

With the growing market of MCUs and embedded electronic powered by batteries, more energy efficient peripherals are needed. This project presents an Analog to Digital converter using ultra low supply voltage on transistor level. With a charge recycling spilt capacitive DAC using set and down switching method. And a comparator with a dynamic amplifier, resulted in a very energy efficient SAR ADC. The ADC got a samplings rate of 50k Hz and an ENOB of 7.89 bits, while only draining 75nW power from a 0.5V supply. This results in a Walden FOM of 7.39fJ/conv.step.

Table of Contents

List of Tables

List of Figures

Abbreviations

ADC	=	Analog to Digital Converter
CMOS	=	Complementary Metal Oxide Semiconductor
CPU	=	Central Processing Unit
C_U	=	Unit capacitance
DAC	=	Digital to Analog Converter
DNL	=	Differential Non Linearity
E_{GAIN}	=	Gain error
E_{OFF}	=	Offset error
ENOB	=	Effective Number Of Bits
FFT	=	Fast Fourier Transform
FOM	=	Figure Of Merit
F_S	=	Samplings speed
INL	=	Integral Non Linearity
LSB	=	Least Significant Bit
MCU	=	Micro Control Unit
MOS	=	Metal Oxide Semiconductor
MSB	=	Most Significant Bit
RMS	=	Root Mean Square
SAR	=	Successive approximation register
S& H	=	Sample and Hold module
SNR	=	Signal to Noise Ratio
SNDR	=	Signal to Noise and Distortion Ratio
ULV	=	Ultra Low Voltage
V_{DS}	=	Drain-source voltage in a transistor
V_{DD}	=	Supply voltage
V_{GS}	=	Gate source voltage in a transistor
V_{REFP}	=	Positive reference voltage
V_{REFN}	=	Negative reference voltage
V_{SS}	=	Analog ground
V_{TH}	=	Threshold voltage

Chapter 1

Introduction

A micro controller or MCU consists of a processor core, a memory and programmable input/output peripherals. It is simpler than the micro-processor (CPU) but also more integrated. This makes them cheap to use and to develop, and the number of MCU used per person is becoming very high. Many MCUs are used embedded in other machinery such as cars, phones, appliances etc. The global MCU market is huge, resulting in a considerable total value although a single MCU is cheap.

The more advanced MCUs use more energy. So in order to achieve as long battery life time as possible, it becomes more and more important to produce energy efficient MCUs. We want them to do as much as possible with as little energy as possible.

A MCU often have to interact with sensors that produce analog signals. An analog to digital converter (ADC) is necessary to convert the incoming data to a MCU into a form that the processor can recognize. The objective of this master thesis is to develop an energy efficient ADC as a part of a larger idea of a ULV micro controller. The work builds upon a previous master thesis developing some of the digital building needed for a MCU.

For this ADC to work properly with the rest of the MCU some properties are set. This includes a sampling speed of 50k samples per second, supply voltage of 0.5 V and 8 bit accuracy.

This thesis first presents a literature review, followed by the basic theory of MCUs. In chapter 4, the design made in this master thesis is presented, followed by necessary analyses and finally the conclusion of the work.

Chapter 2

Basic Theory

2.1 Transistor design

A transistor is a semiconductor that consists of gate, source and drain that are made of metal. The metal in source and drain is connected to doped silicon while the gate is insulated from the silicon by a thin oxide layer. The silicon between source and drain is the channel; this is doped different from the source. MOS stands for metal oxide semiconductor, and describes how the transistor is made. They are of n-type or p-type and the coupling of both is known as CMOS. Today most circuitry is CMOS technology because the production cost and process yield is better than most other technologies [9].

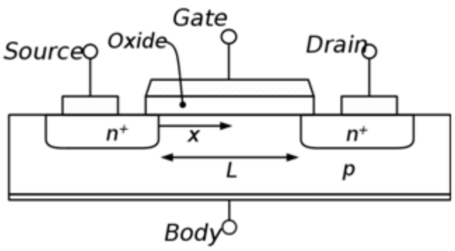


Figure 2.1: Physical view of a NMOS transistor

The source and the drain are physically similar in a transistor, and source is described as the channel with the lowest voltage. Therefore the current will go from drain to source in a NMOS transistor. The length of a transistor is the space between source and drain; this is a design parameter that changes the properties of the transistor. The gate is covered in an insulating oxide to avoid any currents flowing between gate and source. Normally the p- substrate works as an insulator, but when an electric current is applied on the gate an electric field is created, leading free electrons toward the gate and the channel between source and drain is created. This lets current flow from drain to source, and we say that the transistor is active. A higher voltage on gate creates a wider channel, and larger currents are allowed to flow.

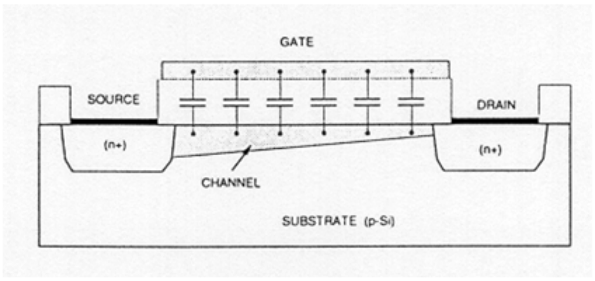


Figure 2.2: Physical view of a NMOS transistors channel

The minimum gate voltage needed to complete a channel between source and drain is called threshold voltage (V_{TH}). Another voltage that is important for the transistors properties is the effective voltage. This is the difference of the threshold and the gate voltage. V-effective tells us how good conductor the transistor is, with a given gate voltage.

$$V_{EFF} = V_{GS} - V_{TH} \tag{2.1}$$

If V_{EFF} is larger than 100mV the transistor is considered to be in strong inversion, and a good conductor. Strong inversion is the most common usage for a transistor, as it can then be considered to be a switch. This is very useful in digital logic and for many different purposes in analog design.

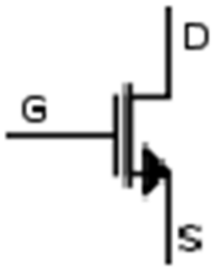


Figure 2.3: Schematic symbol for a NMOS transistor

For the transistor to be considered a conductor there also needs to be a voltage difference between drain and source. If this voltage is lower than V_{EFF}, the transistor is considered to be in the triode region. Giving a linear correlation between I_D current through the transistor and V_{DS}. The transistor can now be described as a resistor. If V_{DS} is larger than V_{EFF} the current is at its maximum given that $V_{GS} > V_{TH}$.

$$C_{GS} = WLC_{OX} \tag{2.2}$$

Gate capacitance exists since the gate is covered with a thin insulating layer between the gate and the channel. This stops all DC currents from flowing from gate to any other node, but creates a small capacitance that can induce an AC current from gate to source when switching [9]. This is normally a very small capacitance. However, some times we do have large transistors and very sensitive nodes. For those time this can cause some problems.

2.1.1 Power consumption in transistor design

Total power consumption in a transistor circuit is given by equation 2.3.

$$P_{total} = P_{dynamic} + P_{leakage} + P_{short-circuit} \tag{2.3}$$

Dynamic power is the energy needed to complete the switching and drive the circuit. This is the power that does something useful and carries the signal further. All other contributions are completely unwanted, and should be reduced as much as possible, they only create noise.

$$P_{dynamic} = V_{DD}^2 F_s C_L \tag{2.4}$$

Dynamic power is given by the switching speed, and the load of the gate and supply voltage. Reducing any of these parameters will reduce the energy drained. This is the major reason for deciding to design a ULV MCU.

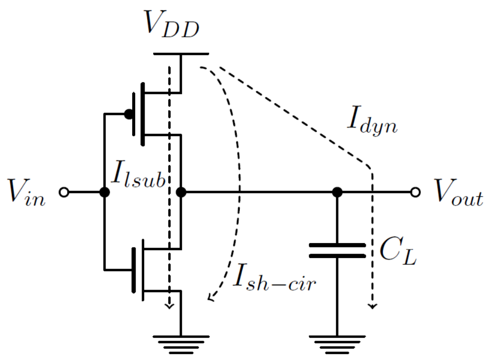


Figure 2.4: Power contributions in an inverter

Leakage power is due to a fault on transistors. There will always be a current from source to drain even if the transistor is off. This effect comes from the fact that our transistors are not ideal, they cannot turn completely off, only have a very high resistance. This affect the time a transistor can be left in hold mode, without changing the output voltage too much.

Figure 2.5 shows the short-circuit power that occurs in every switch. There is a time when both transistors are half on/half off and therefore let a small current through from V_{DD} to V_{SS}. This power is often lumped with dynamic as they happen at the same time.

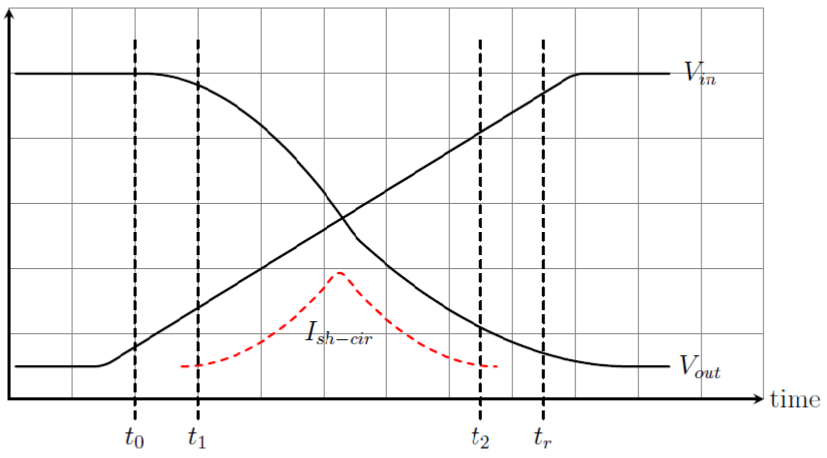


Figure 2.5: Short-circuit power

The red line indicates the current drawn from V_{DD}, and comes as a peak, if the transistors switch fast. This current is reduced by fast settling time of the transistors, and a lower V_{DD}. Settling time is given by the overdrive, the size of V_{EFF}.

2.1.2 PMOS/NMOS

Most common for production of large integrated circuits today is to use Complementary Metal Oxide Semiconductor (CMOS). This technology is called complementary since both NMOS and PMOS are used together as they have some different characteristics. There is a difference on how the doping of the different areas in the transistor is done. First one to be created was the NMOS transistor, where the source and drain consist of a material with extra electrons. While the positive charged substrate got room for extra electrons, NMOS is turned on by a high voltage on the gate. PMOS is the opposite. It is turned on with a low voltage on the gate, the substrate consists of electrons, and the source/drain if positive charged silicon. NMOS is typically three times stronger than PMOS, which means it leads currents better if the size is equal [9].

The definition of source and drain are therefore opposite as well, so all the equations have a small change to them. NMOS is commonly used for pull-down, if source is connected to ground there will not be any voltage drop over the transistor. In the same manner, PMOS's source is often connected to supply, to avoid any voltage drop when the transistor is used to pull up the output voltage.

2.1.3 Body biasing

To ensure better linearity and less process variations it is possible to apply an electrical potential on the substrate. Letting the voltage difference between the substrate and gate be changed. This results in a more linear channel effect, dependent of the voltage rather than only the production. To make this easy the body is often shorted to either gate or source depending on what transistor it is. For NMOS bulk is shorted to source, since this is the lower voltage, and PMOS should be shorted to drain.

2.1.4 Sub threshold, weak inversion

As discussed in chapter 2.1.1, lowering supply voltage is an effective way to reduce power consumption. Sub threshold is when the supply voltage is lower than the threshold voltage of our transistors. Instead of driving the transistors to active region they are left in the triode region [9].

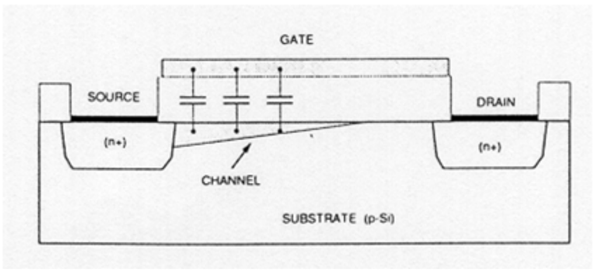


Figure 2.6: Channel with weak inversion

Having a transistor in sub threshold region causes weak inversion and low transconductance due to small bias currents. They are therefore slow on charging capacitive nodes and strongly dependent on transistor-threshold matching when designing. The normal imbalance between NMOS and PMOS is often three timeslarger. Depending on the process used the matching are worse, also either PMOS or NMOS can be the stronger one, not always NMOS as in super threshold. This causes even larger problems with process variations.

2.1.5 Near threshold, moderate inversion

When the drain-source voltage is similar to the threshold voltage the characteristics of the transistor contain both those of super threshold, and sub threshold. A channel is created and works as a good conductor, while the current consumption still is low. This is however a dangerous design approach. Very small process differences can leave some transistors in weak inversion, while some are in strong inversion and therefore creates large errors [4].

2.1.6 Transistor problems in digital logic

With shared nodes several off transistors can leak a small current each to this node, and therefore together create a large enough current for the next logic to assume the wrong voltage level. There is some delay time from t1 to t2 in figure 2.5. This is the time it takes for the output to change enough of a latch. To shorten this time a higher voltages and larger currents are needed.

2.1.7 Stacked transistors

Stacking transistors reduce both on and leakage currents, but does also slow down the circuit, as the currents are reduced. There is also a small voltage drop over each transistor so they may not succeed to give the right output if several are stacked together [9].

2.2 Analog to Digital Converters (ADC)

Data converts are important for communication between the world and the electronics. Analog to digital converters are used with sensors to detect the electronics environment, or with an antenna for wireless communication. Important parameters for an ADC are speed, given in number of samples per second, and accuracy, given in number of bits. Different applications need different specifications and communications often needs to be fast and accurate to achieve high data transfer rates. This does often result in a large and power intensive module and are therefore best suited for larger devices where power consumption is less important. On the opposite end we got devices like a digital thermometer. A sampling rate of one sample per second is enough as the temperature changes far slower, and an accuracy of 0.1 degree over the range -40 to +60 °Cshould be fine for most consumers. This equals an accuracy of 10 bits, with is considered as medium accuracy. For biomedical devices the accuracy can often be lower, depending on what is to be measured.

Table 2.1: Different design topologies

Low-to-Medium speed	**Medium speed**	**High speed**
High Accuracy	**Medium Accuracy**	**Low-to-Medium Accuracy**
Integrating,	Successive Approximation	Flash
Oversampling	Algorithmic	Folding
		Pipelined

There are different ways to implement an ADC depending on its use in table 2.1 most common design approaches are listed. Due to power consumption and area use there are today three main technologies. For very high speeds and low accuracy flash technology is still used. This is far faster than the others as it only uses one cycle to complete the conversion. The other two technologies use oversampling and are mostly designed as a delta-sigma or SAR. Both utilize N cycles for their conversion and are therefore slower than flash. Delta-sigma got more complex logic, and require a very good amplifier to work, and are therefore considered to be a better option when accuracy of around 15 bits or higher is needed. SAR is very power efficient for medium accuracy, but the area and power consumption is doubled for every additional bit, making it less efficient for high accuracy conversions [9] [17].

2.2.1 SAR-ADC

Successive-approximation converters utilize a binary search to figure out the right value. It starts by asking if the input signal is higher or lower than half of the maximum input range. The answer of this question gives the most significant bit. If positive input is larger than than $\frac{1}{2}V_{REF}$ we store a one in the register, and increase the output voltage from the DAC by $\frac{1}{4}$ so see if the signals is larger or smaller than $\frac{3}{4}$ of the input range. This will then give us the second most significant bit. This sequence continues until an accepted accuracy has been acquired, and the search space is divided in two every time [9].

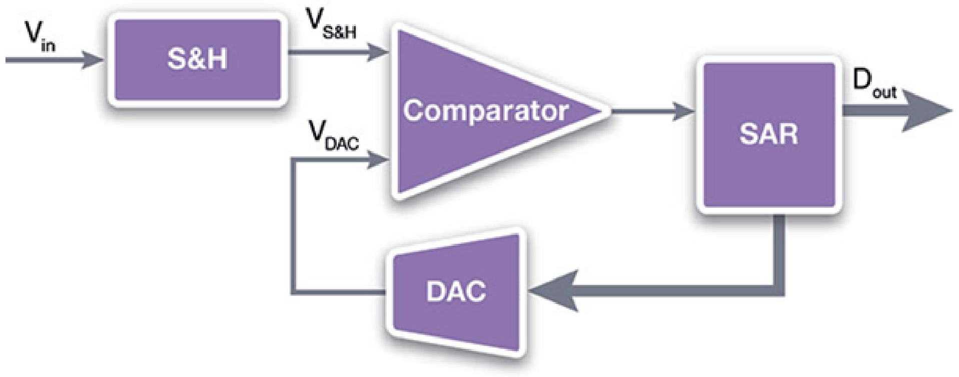


Figure 2.7: Modules in a SAR-ADC

Doing this with a circuit requires several different components. First we need to sample the input signal onto a capacitor in the sample and hold module. This is to make sure the input voltage remains the same for the whole conversion. The comparator does then compare the voltage kept by the Sample and Hold module to the output signal of our DAC, and amplifies the difference. For the successive approximation registers(SAR) digital logic to be able to read the result of the comparison this signal should be close to supply or ground voltage and result in logic 1 or 0. The SAR-module contains a register and control logic for the whole circuit. To describe an analog value in ones and zeroes, every bit is connected to a voltage level. The most significant bit (MSB) is associated with the largest voltage level, and should be half of the voltage range. We use the term MSB instead of first or last bit, as there is no rules on how these should be communicated. Typical they are sent as a bus, and then there is no first or last, just several simultaneously. The bit associated with the smallest voltage difference measured are called least significant bit (LSB). This is the smallest voltage difference possible to describe with this number of bits [24].

$$V_{LSB} = \frac{V_{refp} - V_{refn}}{2^N} \tag{2.5}$$

It is very little analog circuitry in a SAR ADC. There is basically only the comparator, some switchers and capacitances. This makes it very suitable for low voltages and low power circumstances [17].

2.3 Digital to Analog Converter

In the same way as an ADC is used by electronics to sense the real word, a DAC can be used to interact with the world around it and be used to transmit wireless signals. A DAC is also needed to make the SAR ADC work. Designing a DAC is often simpler than designing an ADC, since we get a discreet value and there will not be any data loss. DACs as ADCs differ in settling time and accuracy. For the DAC in a SAR the most important is to give out the right voltage and linearity, and to keep the settling time low enough. The DAC needs to settle before the comparator compares the values, and this can be problematic if the capacitances are too high. Linearity is important since its hard to compensate for it later, while it is possible to remove offset digitally.

2.3.1 Charge redistribution DAC

One of the most popular ways to implement the DAC is to use a capacitive array; this was first done by James McCreary in 1975 [22]. While earlier designs used resistor string for voltage scaling, it is today most common to use capacitors. Capacitors got better matching and are less dependent of the resistance in the switches. MOS transistors got a resistance while on, but does not have any voltage drop. This is good as cap arrays are only voltage dependent, and do not depend as much on the currents. Having a capacitive array lets us sample the input signal directly onto this array, removing the need for a separate Sample and Hold module. It is most common to use a binary scaled DAC, where the MSB capacitor is 50$\%$ of total capacitance, MSB-1 is 25$\%$ and then continues down the same way. To achieve the right total capacitance we need a dummy capacitor of the same size as the LSB capacitor. Improved matching of these arrays are often based on unit capacitances, letting C_{LSB} be equal to one unit capacitance (Cu), and then double the size for each bit added accuracy. This result in a total array capacitance of $2N * Cu$.

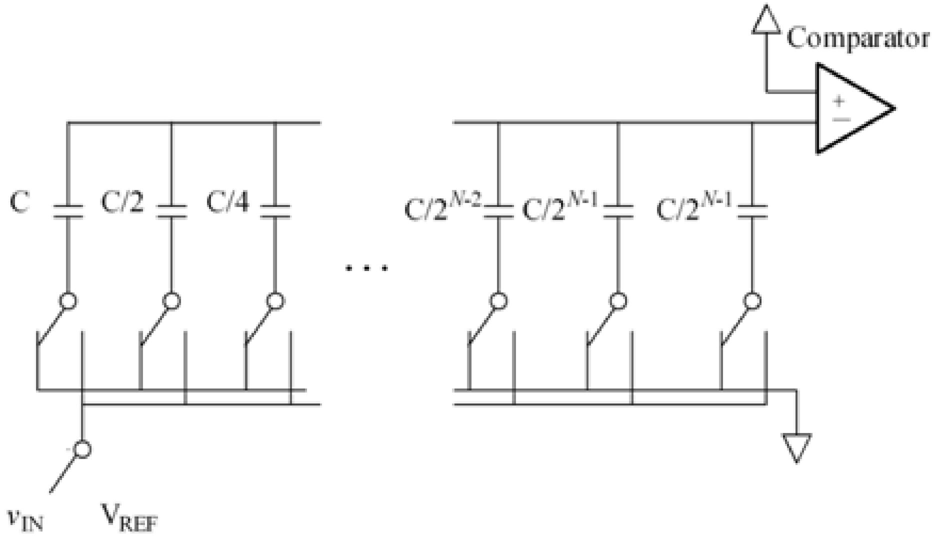


Figure 2.8: Charge recycling DAC

First are all capacitors charged to V_{IN}, and then the bottom of all capacitors are connected to V_{REF}. When the bit cycling is ready to start, the MSB capacitor is first connected to ground reducing the input voltage of the comparator by $\frac{1}{2}$ V-ref. This new value is compared to ground, and the result is saved. If $V_{IN} - V_{REF} > 0$ the value is one, and the MSB capacitor stays connected to ground. Otherwise this capacitor is reconnected to V_{REF}, and the input and the comparator are back to start. In the next cycle MSB-1 is connected to ground. The MSB capacitor stays connected to V_{REF} if the result from the first comparison were 1, and is connected to ground if it was 0. This will reduce the voltage by and a new comparison can be made. After N cycles all bit should have been determined and saved in the register.

There are some problems using a capacitive DAC. Most of them are connected to the fact that total capacitance are doubled for each additional bit accuracy. For an 8-bit ADC there is need for 256 unit capacitances, and for a 12-bit this is increased to 4096. For these sizes there are a lot of problems adding up. The first is the area, since even if each unit capacitance is small; adding so many will require a large area. The second problem is the power consumption, since every capacitor needs to be fully charged, and a large capacitance will increase the energy consumption accordingly.

2.3.2 Arbitrary and capacitor scaling

A way to reduce the large increase in total capacitance of a high accuracy DAC is to use another scaling method than binary [26]. Have instead scaled all capacitors arbitrary, giving them values 224, 128, 72, 40, 23, 12, 6, 4, 2, 1. By doing this scaling they have reduced the total capacitance by 50%, but this does comes with a cost of increased INL.

2.3.3 C2C capacitor scaling

C2C is an even more capacity saving way to implement the DAC. Total capacitance is given by $A = Cu(3N - 1)$ were N is the number of bits.[3] When all bits are using the same capacitor in series it gives a mismatch in this capacitor with a very large impact of the complete design. Small parasitic capacitances add up quick, and impede the linearity. This makes them only suitable for low accuracy (4-6 bits) applications.

2.3.4 Hybrid resistor-capacitor DAC

The first DAC created had a resistor string, and this made it possible with different voltages at different nodes. A hybrid DAC replaces some of the smallest capacitors with a unit capacitance and changes the reference voltage instead. For each bit where a capacitor is replaced by a new reference voltage the total capacitance is almost reduced by 50%. Using different references requires the implementation of an extra module. A reference generator does consume quite a lot of power as it is hard to turn off when not in use, and therefore continue to drain current. For a low power device this continued draw current may be a large disadvantage. The relative power consumption is dependent on the sampling speed. When sampling on a higher speed the power consumption of the DAC is increased much, and the power saving from less total capacitance overweighs the always on voltage generator.

2.3.5 Split capacitances

It is possible to split the array into two smaller arrays, usually divided into two sub arrays, MSB and LSB array. This will reduce the total capacitance since the MSB capacitor of each array is going to be a lot smaller. These arrays are connected with a bridge capacitor, and therefore use the fact that two capacitors in series are a lot smaller than either one of them. With the right size of the bridge capacitor the total capacitance of the full LSB array and the bridge should be equal to the smallest capacitor of the MSB-array [6].

$$C_{Bridge} = \frac{2^L}{2^L - 1} \tag{2.6}$$

Of course there are some problems with this approach as with all others. This bridge capacitor is very important to get right; a small mismatch here has a major impact on the mismatch higher up. The parasitic around the bridge does also highly affect INL errors. Especially hard is it when the bridge capacitor is just a little bit larger than the unit capacitance. For an 8-bit ADC it should be $8/7 * Cu$ and this might be hard to succeed designing.[17] [6]

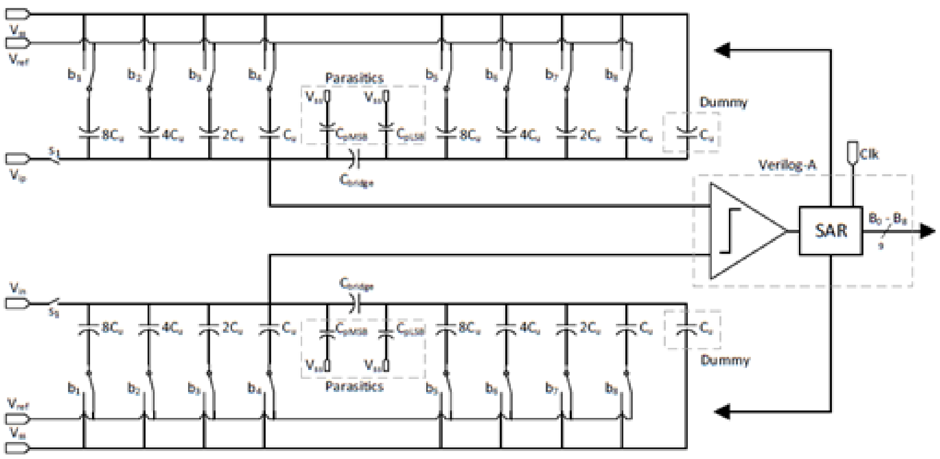
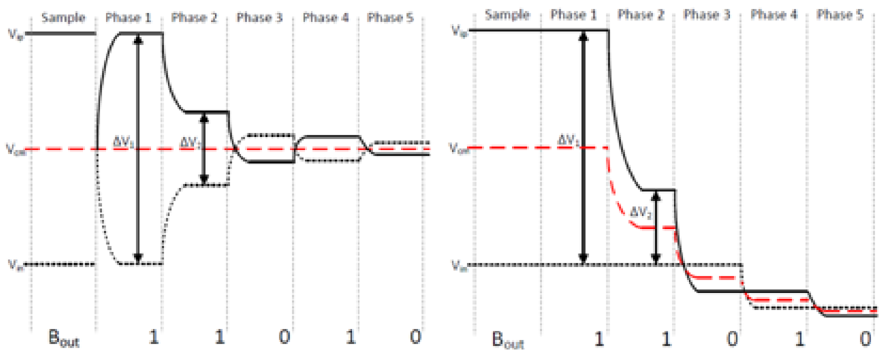


Figure 2.9: DAC with split cap

2.3.6 Charge recycling

One way to reduce the power consumption is to reuse the charge stored on the MSB capacitor after the first comparisons. By adding some extra switching we can reuse the charge to power the MSB-1 capacitor, and in that way reduce the power consumption. By using this method we can reduce the power drained from V_{REF} by 24%, but at a cost of N-1 more switched [12].

2.3.7 Set and down



(A) Conventional switching (B) Monotonic switching

Figure 2.10: Set-and-down switching method

For the comparing we can use set-and-down switching in the DAC, this is a very energy efficient logarithm, due to removal of MSB capacitors the total capacitance is reduced by 50 %. First all capacitors are connected to V_{IN} and V_{REFP}, and the input signal is sampled onto both arrays. When we have the first comparison, comparing input direct lets us use one less cycle. The MSB-1 switch of the array with the highest voltage is set to ground, lowering the voltage of that input node by V_{REFP}. Now a new comparison is done, and the highest voltage switched down. There is no charging of the capacitors after the first cycle, which is why it is so energy efficient. A problem with this is that the common mode voltage is changing between V_{REFP} and V_{REFN}. This can cause a problem for the linearity of the comparator, as the offset is affected by the changing common mode voltage [19].

2.3.8 Switch back

To avoid the problems with the change of the common mode voltage in a set-and-down DAC, Huang come up with a new method of switching [16]. This method requires just the same circuitry as set-and-down, but switch a little bit different. First in the sampling phase all capacitors are connected to V_{SS}, except the MSB. When the bit cycling start, and MSB is found the DAC-array with the highest voltage will switch the MSB capacitor down to V_{SS}. Now the bit cycling continues in the usual way, but in contrast to set-and-down the lowest voltage is switched up to V_{REF}. This leaves the common mode voltage to be closer to V_{REF} [17] [20].

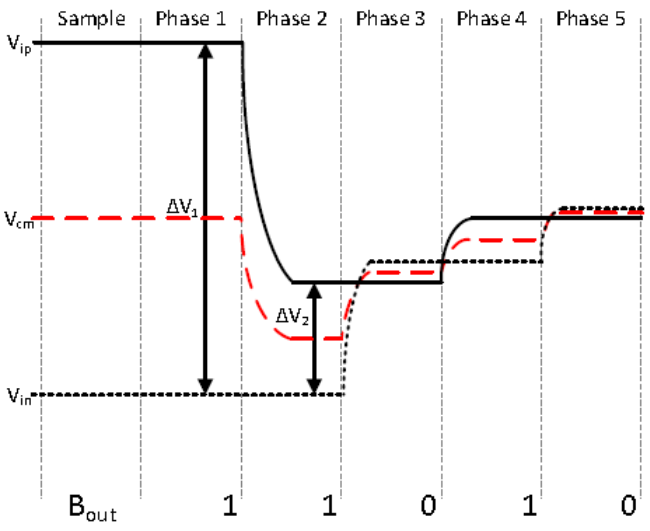


Figure 2.11: Switch back method

2.4 Comparator theory

To be able to use digital logic we need to change all signals to be either a logic one, or a zero. This is still different voltages, but for the logic to be fast enough it should be as high as possible to ensure switching speed is kept. The task of the comparator in this chip is to amplify the difference between the inputs, onto a level were digital logic can understand it [23].

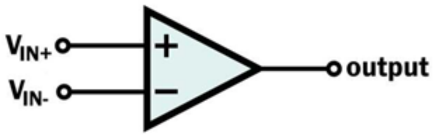


Figure 2.12: Comparator

A comparator amplifies the difference between the inputs and for single ended inputs the negative node is connected to ground instead. When the input difference is amplified the comparator locks it to either V_{DD} or V_{SS}. A comparator works as a connector between analog and digital circuits.

2.4.1 Amplifier

The first part of making a comparator work is to amplify the differential input signals to speed up and reduce noise in later stages. Having a large signal early increases the SNR and therefore reduces the chance of errors. Resistive divider is a way to see difference of the input currents with the inputs connected to the gate of the input transistors. The input voltage determines how large resistance the transistor keeps up, and therefore controls the current flow. The drain of each transistor is connected to a node with a stable connection to ground, either in form of a transistor in off mode, or one biased to keep a stable resistance. The larger current will therefore charge up the node faster [5] [11].

When using a resistive divider, a larger current through the input transistors results in faster settling and better noise performance. Since the comparator only is needed for 50% of the time, it is possible to turn of the amplifier while waiting for the digital logic and the DAC to settle. This is called a dynamic amplifier, as it changes due to the need of it [27].

2.4.2 Latch

To stabilize the output we are using a latch, locking the output to V_{DD} or V_{SS} to symbolize analog one or zero. The latch is made to be decisive, and therefore changes its input as well as the output. The inputs fastest increasing is defined as high and therefore increased further up to V_{DD}. While the lower is then considered to be low and input is forced to V_{SS}. This latch needs to be reset for every comparison. This is done by either lowering the input to zero or, add a lock signal, that turns it off, and in that manner also reduce its power consumption [25] [18].

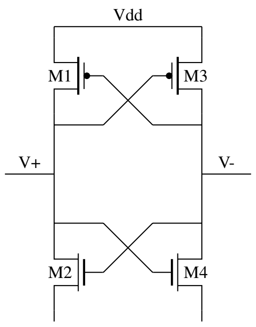


Figure 2.13: Basic Latch

2.5 Noise

All unwanted parts of a signal are considered as noise. It's normal to look at each single contribution as uncorrelated with the others. This is not totally correct, but lets the calculation become a lot easier because we can just add all contributions together.

2.5.1 Thermal Noise

Thermal noise is random electron movement inside the conductors caused by heat. This is an imperfection in materials and only extensive cooling can reduce the effect. Since this is expensive to do, it is not used for microcontrollers. These thermal charges are disappearing inside capacitors or large transistors, these absorb the charge but large capacitances forces us to increase the signals and therefore achieve higher SNR.

$$\sigma_{thermal} = \frac{kT}{C} \tag{2.7}$$

Thermal noise is typical a larger problem for ADCs with high accuracy and high speed. Small ADCs got other more dominating problems with increased capacitor size well over the limits of the thermal noise contribution [9].

2.5.2 Quantification noise

Quantifications noise exists in all ADCs and it will always be LSB. When we measure an analog value and convert it to a digital it may be as much as a half bit. This is showed in figure 2.14 In a) we can see how the digital output follows a continuing input line. Even with a prefect digitalization we got a difference between in and out signals. This difference is illustrated in b) as $V_{IN} - V_{OUT}$. We can see how the difference varies from zero to V_{LSB} [28] .

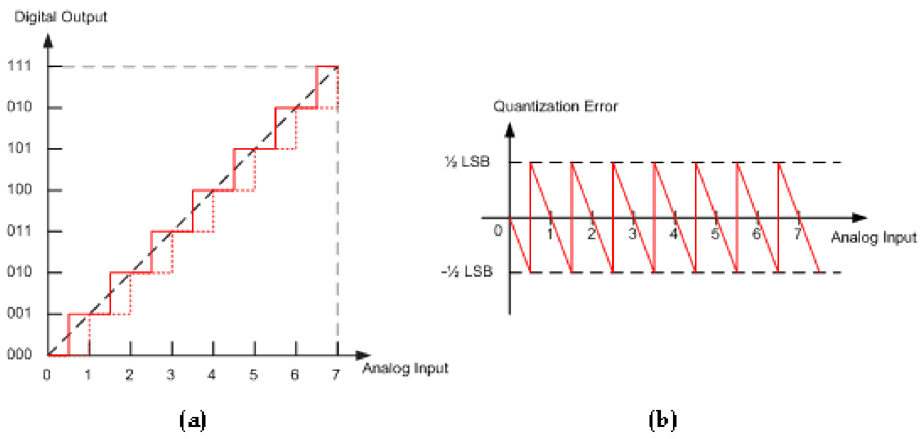


Figure 2.14: Quantization error, a) code fault, b) max error

If we only consider these two noise sources first, we can calculate the minimum capacitance needed to achieve thermal noise less than LSB. For most SARs this is far lower than the actual capacitance used.

2.5.3 Flicker noise

When a transistor switches from on to off there will be a charge injection. This is a large problem whenever there is a clock signal. The problem is larger for large transistors as C_{DS} relate to transistor size. 2.1.1 [9] Digital logic contains a lot of transistors switching every clock period. A possible solution to this problem is therefore to have two different power supplies, so that the analog circuit avoids the noise contribution of the digital logic.

2.5.4 Offset

Offset in an unwanted DC voltage on the input nodes of the comparator. It is important to notice this voltage and see if it is large enough to interfere with any of the results when an input signal is added.

$$E_{OFF} = \frac{V_{OUT}}{V_{LSB}}|_{0..0} \tag{2.8}$$

Mismatch in a comparator is a large contributor to an offset voltage in the same comparator [16]. In [24] this contribution is given by this equation.

$$V_{OSM} = \Delta V_{th} + \frac{1}{2}(\frac{\Delta W}{W} - \frac{\Delta L}{L})(V_{gs} - V_{th}) + \frac{\Delta Q}{C_l} \tag{2.9}$$

Some offset noise is dependent on the common mode voltage of the DAC. We can have a different offset voltage when the inputs are close to V_{DD} or V_{SS}. This is only a problem when the input voltages are close and therefore hard to differ.

2.5.5 Gain error

We want our system to be linear, meaning a change in input results in the same change on output. Gain is the gradient on the output curve, and therefore a gain error is the wrong ending point, similarly to the offset is wrong at the starting point.

$$E_{GAIN} = \frac{V_{OUT}}{N^2 V_{LSB}}|_{1..1} \tag{2.10}$$

For an ADC this is a problem for the DAC more than the comparator. Capacitor matching is therefore the main contribution to remove this fault.

2.5.6 INL

Integral Nonlinearity error is one of the most important faults we can have in an ADC when offset and gain errors are removed. Typical conversions are in the middle of the input range, and therefore it is here we want the best accuracy. INL happens when we do not have the same step change for all values. They may be caused by several factors as mismatch, slow circuit or comparator noise.

$$INL = \frac{V_{OUT-i} - V_{LSB} * i}{V_{LSB}}|_{i=0->2^N} \tag{2.11}$$

INL given in the equation above is the difference between output and input. If this relationship is non-linear the output will not follow our input signal as good as it does in 2.14. There is no way to compensate for INL digitally, therefore it is important to avoid INL from the beginning. In equation 2.9 we can see that the overdrive, $V_{GS} > V_{TH}$, is affected by mismatch. This is a major concern since this will change linearity of the ADC [15]. Yung-Hui Chung has proposed a way [8] to reduce the INL error in a DAC due to mismatch. By splitting up MSB capacitor into a new array and changing what capacitors you use the INL over time is improved if we look at the statistics.

2.5.7 DNL

Differential Nonlinearity error is the error in step size for each digital word. For an ideal converter this is 0, while for most converters design it may vary between 0 and 1LSB. We often refer to DNL as the maximum magnitude of DNL error [9].

2.5.8 Missing codes

Missing codes is when steps are so uneven that it is impossible to achieve some digital values. An AD converter is guaranteed to not have any missing codes if DNL error is less than 1LSB, or if maximum INL error 0.5LSB [9].

2.5.9 Distortion

Using differential input, there are not any distortion from our input signals. But inside our circuit there will be production errors and these will cause the differential pair to behave differently. This difference is mainly caused by mismatch and can contribute to wrong results in a conversion.

2.5.10 Kickback

To avoid mismatch at the most critical point the input transistors may become huge and therefore their parasitic capacitors may affect other parts of the ADC design. Problems occur when charge from clock switching inside the comparator is transferred out of the input terminals and onto the capacitive DAC array [9]. Compensation may be needed when Cu is low, since these input transistors are so large. These transistors create some large parasitic, which can send charge back when any gates on these input transistors change fast. Three different ways to cancel most of the kickback are presented by Figuerredo and Vital [10] By adding one shorted transistor on each input a similar capacitance with an opposite charge will be created.

2.5.11 Meta stability

The comparator needs to succeed some timing. In a SAR it needs to give results to digital logic and let the logic read it within a clock cycle. To be sure, this comparison should be completed in 1/2 clock cycle. If we are done comparing within 1/2 cycle we can use another half cycle to read the value. Meta stability occurs when the amplifier in our comparator does not have enough gain or a transistor in the latch is too slow on settling to a final value. It is most problematic when we got a small differential input, since $Vgs - latch = gm(Vinp > Vinn)$ [14] [13].

2.5.12 Hysteresis

Hysteresis is when the output not only depends on this input but also the last conversion. This is a typical problem with fast circuits, where there is some overlap between signals, so the output does not have time to fully reset. As for this ADC it should not be a problem but we should always make sure that it works [9].

2.6 Control logic

The control logic works like a finite state machine, first there is a initial state, waiting for a start covert signal. When this arrive we continue to sampling state. The input signal is loaded onto the arrays. When this is done, the next state is cycling state. For N cycles each bit is determined. After N clock cycles the ready flag is set high, and the comparator is ready for a new conversion. While the output us bused out.

To use ULV on digital logic we need to be sure to have enough difference between a logic '1' and '0'. A small difference makes it recipient of noise and other errors. Therefore it is often created to not change unless the the voltage change is significant. The difference size does impact the speed for the next logic to settle.

2.7 Process variations

Producing an integrated circuit is a long process with a lot of variables. Even though 130nm is considered an old process it is far from perfected in productions manners. Models given by the manufactures have these variations included and they are possible to simulate. This helps us to design robust enough to get a satisfying yield from productions.

Process variations are divided into global and local variations. For global variations all circuits on the same wafer got the same variation. This might be that all capacitors are larger than designed, or all transistors are a bit wider.

Local process variations are hard to deal with and cause mismatch between transistors and capacitors.

2.8 Capacitor types

We got three different capacitor designs to choose from and all have advantages and disadvantages. Important for this design is the lowest possible capacitance and size while matching and linearity should be high.

2.8.1 Polly capacitor

Polly capacitors are created by stacking two polly-silicon layers with insulator in between. This is highly adoptable, cheap to create and gives a high unit value. They are therefore a good choice when a large capacitor is needed, and space is a problem. They do however have a problem with linearity as the process variations are decisive. These are not recommended to use with flash. [1]

2.8.2 MIM Capacitor

By using two top metal layers with an insulator between, we can create an effective capacitor. All characteristics are good for this design. The unit capacitance is a bit lower than for the polly capacitor. MIM capacitors are easy to create but they are more expensive because they utilize two more layers. These layers will increase total cost of the chip by around 20 %.

2.8.3 MOM Capacitor

Mom is based on coupling capacitance between fingers running in parallel. Metal layers are stacked to increase capacity density. Width and spacing are kept at a minimum to achieve maximum capacity density. We cannot use the top layer for this, so in a 4LM process there are 3 layers stacked, and in a 5LM we can have either 3 or 4 layers. The use of four layers will increase the capacitance and improve the mismatch. We can change length of each finger, and number of fingers to increase the size. Layouts are however fixed, and we got three different sizes to choose from. For this design the smallest should be considered, as mismatch and power consumption is the most important. There is no need for larger capacitances. MOMS are free capacitors (no extra masks) with good linearity and high density. Traditional unit cell layout is showing poor mismatch performance due to high sensitivity to process variations. Common centroid layout is therefore needed. [1]

2.8.4 Capacitor Calibration

There are ways to compensate for the offset of the input nodes of the comparator. Mohammadi and Sadeghipour use an offset trimming circuit [23]. After production this chip is tested for offset voltage, and then calibrated by turning on shorted transistors. Another possibility is to use predefined trim capacitors in the DAC, and use enough of them to cancel out offset, as done in [7].

2.9 Simulations

It is important to check that the conversion works for all inputs, also when they change fast. This is called dynamic range. We have to see if a fast changing input will decrease our accuracy. [9]. All sampling is therefore done with input signals higher than expected in normal usage.

2.9.1 Coherent Sampling

To ensure the design is working properly, we need to simulate in a way which we take samples at the right time. By using Coherent sampling we remove the need for windowing, and increase the Fast Fourier Transforms (FFT) resolution [2].

$$\frac{F_{in}}{F_{sample}} = \frac{N_{window}}{N_{record}} \tag{2.12}$$

We do not want these fractions to have a perfect answer, so therefore they should be based on a prime number. We do this by choosing Nwindow to be a prime number in the right value range and then evaluate Fin from equation 2.13.

$$F_{in} = \frac{N_{window}}{N_{record}} * F_{sample} \tag{2.13}$$

If we are unable to perform coherent sampling, windowed mode needs to be used instead. Windows makes us less dependent on slide lobes, because they differ in how fast they fall off. Typical values are 20 or 26 dB per decade [2].

2.9.2 Temperature Simulations

Our models are not a perfect description on how a printed transistor behaves. We must therefore make sure which work fine, even when the behavior is changed. To do this we simulate with different models. There are some production differences that can be avoided if the design is cone correctly. We can simulate for problems when all PMOS or NMOS are either faster or slower than normal. This is done by exchanging the normal transistor model for a slower or faster one. Manufactures typical creates these for customers to make sure their circuits will work before going to production.

2.9.3 Monte Carlo

To be sure the yield is high enough we need to simulate with production errors. Atmel got design models developed with the manufacture, simulating typical process errors. We want to test as many different options as possible and are therefore running Monte Carlo simulations. Monte Carlo is all random variation and requires very many runs to be accurate. It is therefore advised to use Latin Hypercube instead as this makes sure that more different values for each parameter are used.

2.9.4 In linearity Error

We can look for INL by using a sweeping signal over the input pins. Testing from lowest to the highest input, this will result in all possible output codes. It is important to be sure that we don't have any missing codes. We can find INL-error by looking at the difference between output and input signals. This will look like a saw tooth line for each code, and have peaks around $\frac{1}{2}$ LSB. This graph should look like figure 2.14. This is the quantization error, and if any of the saw teeth are higher than the others we got INL-errors.

2.10 Measurements

2.10.1 Settling time of Comparator, DAC and SAR logic

To make sure the settling time of the comparator is fast enough it should be simulated with transient analyze on the worst possible condition. These are low supply voltage, slow-slow transistors and low temperature. If the time from start to finish of one comparison still is good, we can conclude with it to be a success. If the settling time is to slow, higher current should be considered.

2.10.2 Power consumption

To measure the power consumption it is possible to simulate the current drawn from supply, and multiply the average current drawn over some time with the supply voltage. To be sure we get an accurate result, the ADC should be converting the whole time, and preferable convert different input values. We use average and not RMS as some other part of the circuitry can use the current we deliver back.

2.10.3 SNDR

Signal to noise and distortion ratio is a way to measure the strength of the output signal. We want the signal to be a lot higher than the noise created by the module. For an ADC there is a maximum SNDR, for an 8-bit ADC this is 49.9dB. This means that the signal is 100 times larger than the noise level [9].

$$SNDR = 6.02 * N + 1.76 \tag{2.14}$$

2.10.4 Input impedance of the comparator

This capacitance is important as it tells us how large input currents we need just to create a different voltage on the input nodes. A large capacitance reduces the noise, but forces us to use larger signals.

$$C = \frac{I}{\frac{dV}{dt}} \tag{2.15}$$

2.10.5 ENOB

It is most common to give the accuracy in effective number of bits. This is calculated from SNDR, and describes how many of our bits that are useful [29].

$$ENOB = \frac{SNDR - 1.76}{6.02} \tag{2.16}$$

2.10.6 Figure of Merit

The best way to compare different SAR ADCs is to look at Walden's figure of merit [20]. This is a way to compare ADCs with different specifications to see how good they are. Power is given in watt, ENOB is effective number of bits, and Fs is the sampling frequency. These three parameters are trade-offs for an ADC, if you increase one you typical reduce the other two as well [29].

$$FOM = \frac{Power}{2^{ENOB} + Fs} \tag{2.17}$$

Chapter 3

Design

3.1 Design process

The design process used is the normal process of Atmel. This includes three different reviews for each verification level. Each of these design levels needs to be approved by several other employees with different background to verify the work done. This is to make sure everything is designed according to other participants of the same project and senior engineers have options to comment on possible problems and solutions. Other designer are therefore included in the process to decide on what features that are needed and how they should interact with the rest of the design.

3.1.1 Specifications Review

Some project starts with a Feature Review to decide what the module should include, and be used for. In this project this part was done earlier, so the work started by preparing for specifications review. This review is a presentation of the basics functionality and how this is planned to be implement. It makes sure that time is not wasted on the wrong things. since it is no point creating something good that never is going to be used.

Module interface

Typically a system is built up of different modules that can be designed separately and then put together at a later point. In the specification review all modules with functionality and interface is described. Knowing how all modules interact with each other helps when the design decision needs to be done later. This is also the time to find different dependencies, if the rest of the project team needs to provide anything for your module to work.

Timing diagram

Functionality is easier to test when it's known how it should behave. A timing diagram for the whole system helps when designing and setting up timing specifications. It's easier to decide where time can be spared, and were it is needed. Often a lot of modules depend on each other. Knowing if something can be done earlier is therefore important, and locating the critical timing path that decides how much time is needed a conversions.

Block diagram

By drawing all the modules as blocks, we can build a simple system. Each block is replaced by a Verilog model, letting us test some basic functionality and look for functionality issues. All modules should try to indicate a real module as good as possible, without being too complicated. Some timing delays and internal resistance is therefore added.

Test benches

Everything needs to be simulated and tested to ensure good results. Having the tests early gives a smoother design process and faster response about problems.

Electrical characteristics (level 1)

Estimate of power consumption and area usage, needs to be given. This is to give other participants a chance to say early if this is good enough, and let them work accordingly. Expected results are given for each module and the full system. This is the description on how everything is expected to perform. All results are based on equations, prior knowledge and experience. The resuls of the specifications review is presented in table 3.2.

3.1.2 Design Review

When the specifications are given it is time to start designing each module. The verilog modules should be replaced by transistor design based on models created for more accurate simulations. The new values may differ from the prior review and a new review is therefore needed to comment on the new results and see if anything in the project has changed. Some problems can be easier to solve other places in the project.

Transistor level modules

All modules need to be implemented on transistor level and verified. Now all design decisions are made to achieve the results needed. This is one of the more time consuming parts. The specifications from prior review must be accounted for, but some changes might be done when unknown problems shows up.

Electrical characteristics (level 2)

The new models needs to be tested by simulation tools. These results give a better indication of how a finished product will behave.

3.2 Complete design

The task is to design the major functionality of an ADC. This design consists of a DAC, a comparator and some control logic. Each part is designed as it own module, and then connected together into one module called ADC-core.

3.2.1 Interface

The ADC core is the black box other designer will have to deal with. It is therefore imprtant to describe all pins and the functionality in a simple way. ADC core is implemented as one module with the interface given in figure 3.1 and table 3.1.

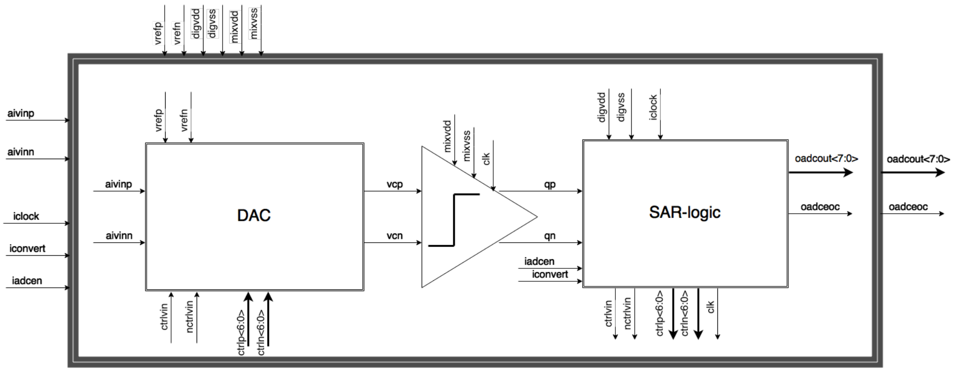


Figure 3.1: Module interface in a SAR-ADC

We have two different power supplies to account for that digital logic got a lot of flicker noise. This results in splitting the power into two. Digital logic is not as sensible to noise, so increased flicker noise is acceptable. The same goes for ground signal, this is also divided in two. The signal named ”iadcen” is used to enable and power up the ADC-core, if low this should be in sleep mode. Setting this high lets the clock signal trough to the comparator. To start each conversion signal named ”iconvert” needs to be high for at least 1/2 clock cycle.

Table 3.1: Interface

Pin name	**Type**	**Domain**	**Description**
Mix12Vdd	power	Mix12vdd	0.5V power supply for analog circuitry
Mix12vss	power	Mix12vdd	Ground for analog circuitry
Dig12vdd	power	Dig12vdd	0.5 V power supply for digital logic
Dig12vss	power	Dig12vdd	Ground for digital logic
Vrefp	Analog input	Mix12vdd	Positive reference voltage
Vrefn	Analog input	Mix12vdd	Negative input voltage
iclock	Digital input	Dig12vdd	Clock signal
aivinp	Analog input	Mix12vdd	Positive input node
aivinn	Analog input	Mix12vdd	Negative input node
iadcen	Digital input	Dig12vdd	Enable signal for ADC core
iconvert	Digital input	Dig12vdd	Start convert signal
oadceoc	Digital output	Dig12vdd	End of conversion flag
oadcout$< 7 : 0 >$	Digital output	Dig12vdd	Resulting bits

When the conversion is done, the signal ”oadceoc” is set high to notify the MCU that the values are now readable. The values are read from $oadcout < 7 : 0 >$, witch is a 8bit parallel bus, and the output is kept until next conversion is complete.

3.2.2 Dependencies

Voltage reference

Our DAC needs two different reference voltages to be able to give out the right values. In the assignment the input range is given to be between 0 V and 0.5 V. This is the same values that should be in the DAC and it is the same values as V_{DD} and V_{SS} for the circuit. We can however not use the supply voltage as V_{refp} and ground as V_{refn}. These nodes are too noisy to get the accuracy needed. It is important that these signals are stable even when high currents are pulled for the supply.

Clock signal

We need a clock signal that is 9 times faster than the conversion speed. This means 450 kHz if we want to sample at 50ksps. A lower clock speed can be applied, but this will change the conversion rate accordingly.

3.3 Specification

First the assignment, and then specifications review and a lot of calculations have resulted in the specification given in table 3.2.

Table 3.2: Specifications

	Min	**Typical**	**Max**	**Uint**
Input capacitance		7.5		pF
Integral non-linearity		0.5	± 0.5	LSB
SNR		49.0		dB
SFDR		54.2		dB
ENOB	7.5	7.8		bits
Power consumption		125		nW

3.4 Noise conditions

Our targeted ENOB is 7.8 bit. From this we can calculate the SNDR of the full system. In equation 3.1 we can see that to achieve 7.8 bits in accuracy we need at last a signal to noise ratio of 48.7 dB. The worst case scenario has an ENOB of 7.5 bit, witch equals a SNDR of 46.7 dB. We should therefore have a targeted SNDR over 49dB, and a minimum of 46.7 dB.

$$SNDR = ENOB * 6.02 + 1.76 = 48.7dB \tag{3.1}$$

For further calculations we assume that all noise sources are independent, and therefore possible to add together. There are some noise contributions that are hard to deal with, this is quantification noise. The only way to make it smaller is to increase the number of bits, or signal power.

$$\sigma = \frac{V_{LSB}}{\sqrt{12}} = 563\mu V_{RMS} \rightarrow 52.9dBSNR \tag{3.2}$$

As in all electronics temperature causes electrons to move, and therefore create noise. This noise is fairly small, and almost irrelevant for this design.

$$\sigma_{Thermalnoisesample} = \frac{k_B * T}{C_{tot}} = 23.5\mu V_{RMS} \rightarrow 80.4dBSNR \tag{3.3}$$

Noise from the comparator and reference buffer is assumed to be the most critical sources. We will therefore try to give these components as much slack as we can in their specification. Backwards engineering of the target accuracy gives us the equation for targeted noise in these modules.

$$\sqrt{2 * x^2 + 563^2 + 25^2} = 887\mu V_{RMS} \tag{3.4}$$

$$x = 484\mu V_{RMS} \tag{3.5}$$

Table 3.3: Noise Table

Noise source	**Target** $[\mu V_{RMS}]$	**SNR contribution**	**Comment**
Quantization noise	564	52.9 dB	
Sampling cap $\frac{kT}{C}$	25	80.0dB	
Comparator noise	484	54.3 dB	
Reference Noise	484	54.3 dB	
Total	887	49 dB	

INL errors are important to avoid as we don't want any wrong codes. Maximum INL is therefore set to be a half LSB, as this is the closest we are able to get, because of quantization noise.

$$SNDR \cong 20 * log(\frac{2^8}{INL_{MAX}}) = 54.185dB \tag{3.6}$$

3.5 Comparator

The comparator was the first design in verilogA for basic simulations. This module was made to make sure the design worked in total, and give an indication on how everything was going to behave. The characteristics like settling time should be implemented to make sure all modules are timed coherent. We avoid adding noise, and rather keep it like an ideal module, letting everting be perfect inside the major bounds of physics. The full code of this module is given in the appendix. On the second design stage a new module were made on the transistor level. Everything is simulated with cadence to assure the right behaviour.

3.5.1 Specifications

In table 3.4 the necessary specifications for the comparator are given. This is the results for worst case scenarios, most of the time the results should be a lot better.

Table 3.4: Comparator specifications

	Value	**Comment**
Noise contribution	484 μV_{RMS}	54.3 dB
Settling time	550 ns	$\frac{1}{4}$ clock cycle
Input referred offset	950 μV_{RMS}	0.5 LSB

3.5.2 Dependencies

We need a clock signal and a bias current from the circuitry around. It makes sense to have these outside the core as there probably is needed more of these signal in other places in a micro controller. To clock the comparator a clock signal of maximum 450k Hz is needed.

If a lower frequency is applied everything should still work fine, the sampling speed is just slowed down. The power consumption will not be optimal for a lower frequency. Everything here is optimized around 50k sample/sec.

3.5.3 Design decisions

First of all are some parameters that were tried to keep as good as possible. Low bias current for instance, it was set low from the beginning, and the raised true out the design to achieve the requirements. For starts the idea was to keep everything as simple as possible, more transistor basically means more things that can change the behavior in an unwanted manner. More transistors even if they are small, increase the power consumption and the area used. The principle of KISS (Keep It Stupid Simple) is therefore applied in the early design stages. Adjustments had to be made to overcome some problems, so a few more transistors were added along the way.

PMOS input transistors

There are several different ways to design amplifiers, all with different characteristics. In this design we wanted to use the set-and-down switching method for the DAC. Due to this, PMOS were choosed as it is working better when the gate voltage is close to ground. For each iteration the set-and-down switching method is getting closer and closer to ground. Even if the input voltage should be close to V_{DD} this will be a very uncritical decision for the comparator since the difference between positive and negative input should be large every time this happens.

Dynamic range

As discussed earlier in chapter 2.6 the output of the comparator needs to be translated to a digital code. This means that it should either be a 0 (V_{SS}) or a 1 (V_{DD}). It is therefore important that the latch forces the output close enough to these values. This is done by making sure that there is not a voltage drop over any of the transistors. To do this all transistors used to increase the output voltage are of the type PMOS, while all used to reduce it are NMOS.

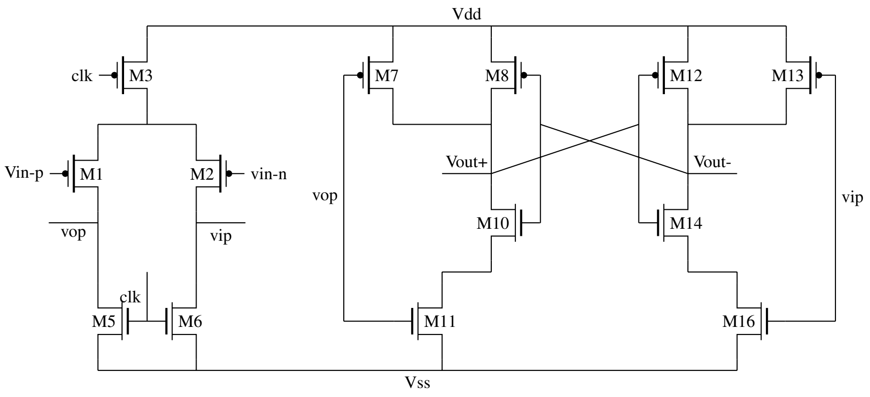


Figure 3.2: Comparator design

There is no need to clock the latch as long as the amplifier is clocked. Instead the proposed design got a more complicated latch, using the amplifier for control. This is possible when the amplifier is accurate, and well designed. All transistors in the latch are small and most work as switches. The critical part in this design is the two input transistors. If these are accurate with low noise, then the rest will have a far larger signal to work with.

Settling time

There is no need to clock the latch as long as the amplifier is clocked. Instead the proposed design got a more complicated latch, using the amplifier for control. This is possible when the amplifier is accurate, and well designed. All transistors in the latch are small and most work as switches. If these are accurate with low noise, then the rest will have a far larger signal to work with.

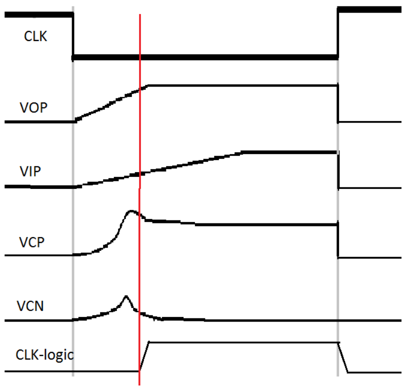


Figure 3.3: Power contributions in an inverter

With a conversion rate of 50k samples/sec the comparator needs to use a clock signal of at least 450 kHz. To give the DAC enough time to settle in a critical moment it needs a $\frac{1}{2}$ clock cycle, while the logic should get a $\frac{1}{4}$. This leaves the comparator with $\frac{1}{4}$ clock cycles, or 0.55 μs.

Mismatch

To reduce the effect of process variations the input transistors were made larger than normal. These are the most important transistors, as their threshold voltage is essential for the behavior of the comparator. Increasing their size also increase total area and power consumption. All other transistors are therefore design with minimum length, and width just enough to conduct the currents needed. Short and wide transistors are often leaky, but simulations show that this is still only a small contribution of the total amount used.

3.5.4 Known drawbacks with this design

The design process is full of tradeoffs and there are therefore bound to be some parts of the design that can be problematic if the specifications are changed. The most important is discussed below.

Power consumption

Energy efficiency is the major concern, so all contributions should be located to see if there is worth doing something to reduce it further. In the amplifier there are three different contributions to the current drawn from supply. First of is the leakage power, that occures due to wide transistors. This can be fixed by increasing the length of these transistors, mainly M3, M5 and M6. This will however force us to increase the width and so the area and the short-circuit currents are increased. The other power contributions are hard to differ in simulations. Short-circuit current will occur, but there are not very much we can do to avoid it. Increasing gate voltage, and have an active bulk bias should help if the problems shows to be too large. Otherwise it is only dependent on supply voltage and switching speed, both unchangeable without a complete redesign.

Last is the dynamic power, in this design it is only used to charge up the capacitive nodes vip and vop. Both are dependent on the transistor size only, and these transistors are close to the smallest size possible within the specifications.

Inside the latch the dynamic power should be low, as the load is only a XOR gate. But there are a lot of transistors changing slowly, so the shot-circuit power is the most dominating. There are bound to be some leakage currents here as well, but dealing with it will most likely increase the total power consumption.

Kickback

The large input transistors create large internal capacitances in these transistors. Switching with the clock signal this result in large amount of charges forced back to the DAC. There are possible ways to reduce the kickback, some are described in chapter 2.5.10. This have however not been done here as the kickback is applied on both of the capacitive arrays, the differential difference is the same between the two.

3.6 DAC

We started off with a verilog module to simulate the control system. This was created like aa adder, adding up all control signals that are high. This sum is then divided by 256, the maximum value, and the multiplied with $(V_{REFP} - V_{REFN})$.

From equation 2.4 we can see that V_{dd}, F_s and C_L is deciding the power consumption. Vdd and Fs are given in the assignment, so there is no room for design improvement here. The focus is therefore to reduce the capacitive load as much as possible.

3.6.1 Interface

Table 3.5: Interface of the DAC

Pin name	**Type**	**Domain**	**Description**
Vrefp	Analog input	Mix12vdd	Positive reference voltage
Vrefn	Analog input	Mix12vdd	Negative input voltage
aivinp	Analog input	Mix12vdd	Positive input node
aivinn	Analog input	Mix12vdd	Negative input node
ctrl_ vin	Digital input	Dig12vdd	Control signal for input switch
ctrlp $< 0:6 >$	Digital input	Dig12vdd	Control signal for positive arrays inverters
ctrln $< 0:6 >$	Digital input	Dig12vdd	Control signal for negative arrays inverters
vcp	Analog output	Mix12vdd	Positive output
vcn	Analog output	Mix12vdd	Negative output

3.6.2 Specification

There are fewer minimum values that have to be competed by the DAC, but more as good as possible. Most important in the functionality, the DAC needs to settling to $\frac{1}{2}V_{LSB}$ within 1.1 µs. Additionally there are some factors that needs to be good enough, but really should not be a problem. This the leakage in the inverters, it need to be so small that the capacitors don't change value noticeable. Accuracy is determined by the leakage current, and the matching of capacitors. Layout and DAC design are the major contributes for this.

As long as these demands are accounted for the task is to reduce the power consumption and area as much as possible. This is mainly done by reducing total capacitance.

3.6.3 Design decisions

Charge redistribution DAC

If charge redistributions DAC is used, we do not need a sample and hold circuit. This is therefore a major power saver. The only reason not to use a charge redistribution DAC is if we not are going to use a capacitive DAC. This is possible, but all other design topologies are larger and use more power.

Matching

The modules used for simulations have been tested extensively by the producer to give us clues on how they are going to performed before type out. This also includes some basic models used for brief calculations. All capacitors have an error chance given as σ. We will therefore have to design the capacitor so that in as many times as possible the errors becomes so small that they are not critical. We can calculate the mismatch in the total array as in equation 3.7

$$\sigma(\sum_{i=1}^{M} C_i) = \sqrt{M} * \sigma(C_0) \tag{3.7}$$

Where M is total number of unit capacitances, 256. Less than LSB fault is wanted, so these calculations are done with 3 sigma accumulated fault.

$$3 * \sigma(C_0 * \sqrt{256} \leq 0.5 \rightarrow \sigma(C_0 \leq 1.04\%) \tag{3.8}$$

To achieve less than $\frac{1}{2}$ LSB fault the matching of all capacitors needs to be better than 1.04 %. We got three different capacitor types to choose from, and all got their own advantages and disadvantages. Important for this design is the lowest possible capacitance, matching, linearity and size.

Capacitor type

Polly: Polly is created to have as high unit value as possible, but it has not been optimized for linearity. It's created by stacking two poly layers, and will not work with flash.

$$\frac{3\sigma(C_{polly})}{A} > 1.04\% \tag{3.9}$$

$$A > \frac{3\sigma(C_{polly})}{1.04\%} = 0.81\mu m^2 \tag{3.10}$$

$$C = A * 2.55\frac{fF}{\mu m^2} = 25.01fF \tag{3.11}$$

This is a fairly large capacitor and this will yield a total capacitance of 6.5pF. This is far more than what should be needed in the DAC. We want as low total capacitance as possible.

MIM: By using two top metal layers with an insulator between, we can create an effective capacitor. All characteristics are good for this design, but it is more expensive because it utilizes two more layers. We can get away with this if the digital logic or other circuitry is using these layers as well.

$$\frac{3\sigma(C_{mim}}{A} > 1.04\% \tag{3.12}$$

$$A > \frac{3\sigma(C_{mim}}{1.04\%} = 5.07\mu m^2 \tag{3.13}$$

$$C = A * 1.6\frac{fF}{\mu m^2} = 8.12fF \tag{3.14}$$

MOM CAP Mom is based on coupling capacitance between fingers running in parallel. Metal layers are stacked to increase capacity density. Width and spacing are kept minimum to achieve maximum capacity density. We cannot use the top layer for this, so in a 4LM process there are 3 layers stacked, and in a 5LM we can have either 3 or 4. Four layers used will increase the capacitance, and also reduce the matching.

We can change length of each finger, and the number of fingers to increase the size. Layouts are however fixed, and we got three different sizes to choose from. For this design the smallest should be considered, as mismatch and power consumption is the most important. There is no need for larger capacitances.

MOM is free capacitors (no extra masks) with good linearity and high density. But tradition unit cell layout is showing poor mismatch performance due to high sensitivity to process variations. Common centroid layout is therefore needed.

The smallest one possible to create with this PDK and design rules is 5x5um with a capacitance of 29.6 fF. This is far larger than the mismatch condition gives if we are able to set them in perfect centroid.

$$\frac{A * \sigma(C_{MOM})}{5x5\mu m^2} = 0.138\% \tag{3.15}$$

Decision

Polly capacitors are worse in every way for this design, but they are useful when large capacitors are needed. The best results would be to use MIM cap, as it is possible to design these smaller than MOM cap. Linearity is good in both cases, so this really ends up as a price to power consumption trade off. If the digital logic don't use any top layer this properly be too expensive to utilize.

Using set and down switching results in a DAC that is only charged one time per conversion. This is calculated with the lowest possible MOM capacitor. Using MIM-cap instead would reduce the power consumption of the DAC by more than 50 %, and still keep the matching well enough. In total this should give 10 % power reduction and 40 % area reduction, but also an increased cost for production of the whole chip.

$$I = \frac{V_{refp} * C_{tot} * F_s}{2} = 94.7nA \tag{3.16}$$

After discussion MIM cap is too expensive to use here, since cost is an important factor for consumer electronics.

Both Polly and MOM are therefore our last options. With both types some other scheme is needed to reduce the capacitor size. This is important for both area and power consumption. Looking through different options split cap looks most promising since it does not have any increase in INL. This design approach is very dependent on matching, but if MOM cap is used the matching is still far better than the specifications require.

Set and down

Using the set-and-down switching method lets us remove the MSB capacitors, and therefore reduce the total capacitance by 50 %. After choosing to use fairly large MOM capacitors this is very important in order to keep it at a reasonable level.

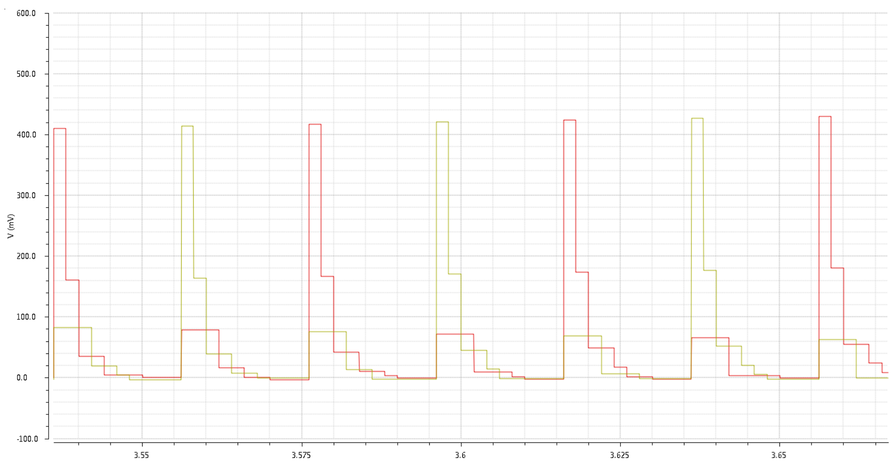


Figure 3.4: Set-and-down DAC schematic

In figure 3.5 we can see that we now only use 256 unit capacitances. When each capacitor is 29.6 fF this results in a total capacitance of 7.6 pF. This is still large, so further improvement needs to be done.

Split Cap

To reduce the capacitances further we can use the fact that capacitors in series are experienced as smaller. We can therefore split the array into two sub arrays, and then connect the LSB array in series with a bridge capacitor. This way the binary scaling is reset, and total capacitance is reduced by additional 80 %. With this approach total capacitance will be 1.36 pF.

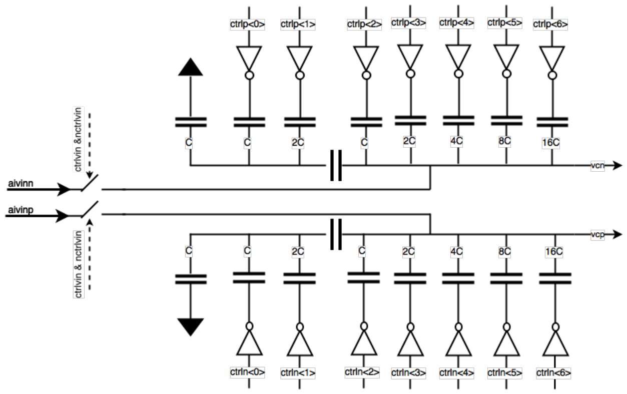


Figure 3.5: Set-and-down DAC schematic, with split cap

3.6.4 Level shifter (inverter)

Bye using inverters, these switches should be smaller and simpler. They may have more leakage, but total power consumption should be better anyway.

$$R_{on-max} = \frac{t_{max}}{C_{MSB}} = \frac{1.1\mu s}{16 * 29.6fF} = 237k\Omega \tag{3.17}$$

This should conduct enough current to charge $16 * Cu$ within $\frac{1}{2}$ clock cycle. From simulations on transistor level this does not look like a problem. This will however be limited by the reference buffer since we will not be able to pull very high currents over a short time then. We want this to be low leakage to avoid change if value inside the DAC. There is a trade off between settling time, and leakage. This is possible to solve with a more complex design, but this should not be needed. This leakage is from V_{REFP} to V_{REFN}, and will not disturb the DAC, but may increase the overall power consumption, especially when the ADC is off. We use inverters created as a master thesis last year, simulations indicate that these behave in the right manner. Overall for the DAC it is more important that the values are keep stable as close to V_{REFP} and V_{REFN} as possible by the inverters.

3.6.5 Input switch

To load the input onto both capacitive arrays we need a switch with low on resistance. For the worst case, one array needs to be charged from V_{REFN} to Vin-max. This should equal 0.5V over 32*Cu. Settling have to be done within 1.1us, with these demands we can calculate Ron max from eq 3.18.

$$R_{on-max} = \frac{t_{max}}{C_{array}} = \frac{1.1\mu s}{32 * 29.6fF} = 118k\Omega \tag{3.18}$$

3.6.6 Know flaws with this design

The set and down switching method has one large problem. The common mode voltage on the inputs will change between V_{REFP} and V_{REFN}. This is a large gap and might increase the dynamic offset, and therefore increase the INL. If it gets too bad the switchback method should be implemented instead, but this will again increase the power consumption. Instead of switching down every cycle, we can switch down the first, and then switch the opposite node up thereafter. This keeps the common mode voltage around $\frac{1}{2}$ V_{REFP} at all time. The bridge capacitor in a cap array is very sensitive to process variations. A small variation here got considerably larger impact on the accuracy than on any other capacitor. Especially parasitic are scary, because they will be as they just add error to the whole LSB array. All switches used in this design is built up from transistors, therefore they will leak a small current at all time. If this current is too large the potential over the capacitors might change, and therefore change the result.

3.7 Digital logic

The digital logic is created as a finite state machine (FSM). It uses the start convert signal as a global reset, this is to ensure that if it the conversion for some reason takes to much time it's possible to reset without powering down.

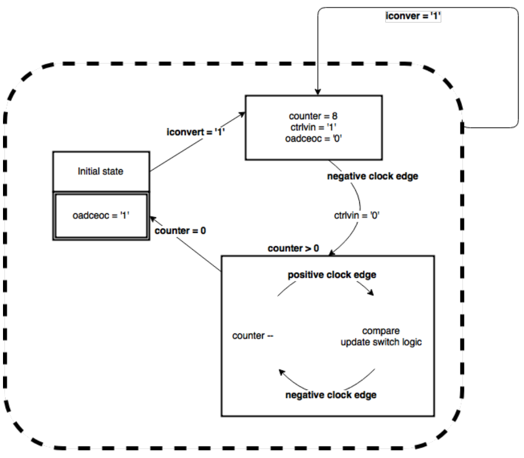


Figure 3.6: Flow diagram of logic statemachine

Every conversion follow the same procedure, independent of what the input signal is.

1. 'iconvert' goes high, this starts the clock counter, and open the input switch. This switch is otherwise closed to save power as the capacitor can drain if they are connected to V_{REFP}. The output flag 'oadceoc' is set low, as we now start on next conversion.

- When 'iconvert' is low again, and we get to the first clock cycle we move to the next state.

2. Second state is were the bit cycling is done. On every positive clock flank comparator values are read, and the control signals for switches updated. On negative clock flank counter is reduced by one.

- This continues until $'counter == 0'$

3. The FSM has now reach it's final state. The conversion is done, so the output flag 'oadceoc' is set high. Now the logic waits for a new 'iconvert' signal to restart.

3.8 Functionality and timing diagram

To easier be able to see faults in simulations a timing diagram were set up. This shows the basic functionality, with the most important signals. In figure 3.7 we see how some of the major signals are changing trough two different conversions. We can see that the clock signal first starts after 'iadcen' is set high. further we need 8 clock cycles after the input have been sampled onto the capacitive arrays. The logic clock is opposite of the input clock, and triggered only when the comparator is in use. 'ctrlvin' is set high by 'iconvert' and then kept high until one negative flank of logic clock after the start signal have been set low.

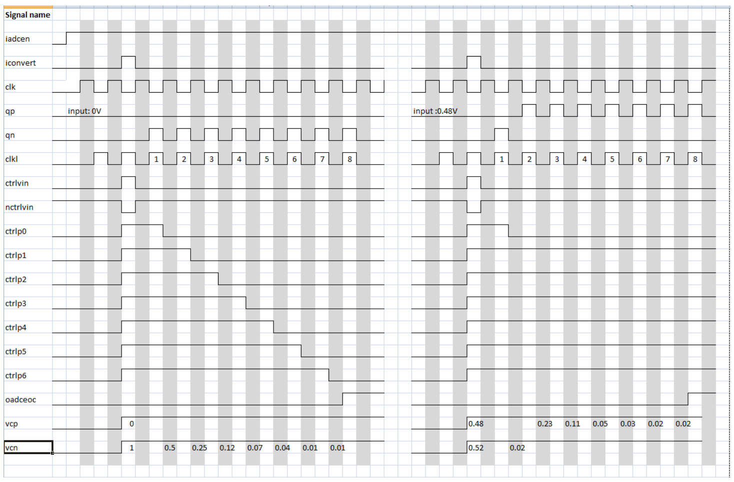


Figure 3.7: Flow diagram of logic state machine

Figure 3.7 shows two different conversions. In the left one we have maximum input difference, and therefore the output of the comparator should be the same the whole way. We can see that signal 'vcn' gets lower and lower, since we are using set-and-down switching. It gets closer and closer to 'vcp', but never reaches it. In the right figure we have two inputs close to each other. The highest one switched down in the first comparison, this gives it a so low value that the rest of the cycles 'vcp' is switched down. The result of the conversion is '10000000'.

3.9 Test benches

To approve this design, two main test benches have been used. Both utilizes a verilog module to translate the output bus back to one signal.

3.9.1 INL test

By having an input increasing slowly from $V_{REFN} to V_{REFP}$ letting the ADC do one conversion for each step. This means that the time from low to high should be $256/50k = 5.12ms$. An even rise in input voltage will result in an output following positive input with a short delay. If we now look at the difference between the input and the output we can see the INL error. Since the output is a digital signal with steps the difference should look like a sawtooth, were each teeth got a size of $\frac{1}{2}V_{LSB}$.

3.9.2 Full test

For full testing we look at the ADC-core as a black box, typical like other users would do. We apply all the nesseseary signals on different pins, and then see how the result behave. It is tested for all limits, and is then expected to work for all parameters in between. A full test runs for 21 ms with $\frac{509}{1024} * 50k$ as input signal. This is Nyquist rate for the input frequency.

Chapter 4

Analysis

Simulations in Cadence is used indicate the behaviour of the circuit. Design model given from the factory is used to achieve as good estimates as possible. Test benches were created in the specification review, and have been used for the whole design. In this chapter the end results are listed and commented. There have been many more simulations, but this is the set up that gives the overall most satisfying results.

Table 4.1: Typical parameters

Temperature	27 °C
Input signal	25k Hz
Mixvdd	0.5 V
Digvdd	0.5 V
Sampling speed	50k Samp/sec
Clock signal	450k Hz

By using the parameter stated in table 4.1 we ran one transient analysis for 21ms and an INL test. The transient analyse were to calculate noise, and see how the ADC behaved in time and frequency domain. Using FFT, noise and accuracy were calculated and inserted into table 4.2. The over all power consumption is the rms current drained from supplies. Noise conditions were one seed ans maximum noise frequency of 100M Hz.

Table 4.2: Results with normal values

ENOB	7.892	bit
Sampling speed	50	k samp/sec
INL	0.5	bit
Input range	480	μV
Power consumption	74.57	nW

All characteristics is as expected, except the power consumption that is lower. This is without implemented digital logic.

4.1 Power consumption

Power consumption is simulated taking rms of the current drawn from supply for each module. All simulations are done with an input signal varying for full range over a long time, to give accurate results. The average consumption is simulated, this is to ensure nothing is wrong when its going to be used. Experienced power consumption can be different, if for example the input is a DC current around $\frac{1}{2}$ 90 % of the time. There will also be small differences on how fast the comparator settles depending on the input.

Table 4.3: Simulated power consumption

Comparator	65.81	nA
DAC	83.33	nA
Total	149.14	nA

The power consumption is far better than first estimated. However there is still possibilities for improvement. Basic calculations would indicate that it is possible for that DAC to only use 44nA in the capacitors, there is still some power consumption the all switches, but this should be possible to reduce even further.

$$I = \frac{V_{refp} * C_{tot} * F_s}{2} = 45.2nA \tag{4.1}$$

To find leakage currents, the input signal 'iadcen' were turned low, this stops all clocks, and therefore stops all movement in the core. The comparator uses 3nA when of, this is a little high, and probably comes from wide and short transistors in both the amplifier and the latch. For the DAC leakage currents is equivalent to leakage in switches. This is currently 2nA, with is also a bit high, and should be looked at.

These result would probably be worse when the layout is complete, due to added parasitics.

4.2 INL test

An INL testbench were used, resulting in a perfect sawtooth graph seen in figure 4.1. This only shows a part of the input values, all look the same.

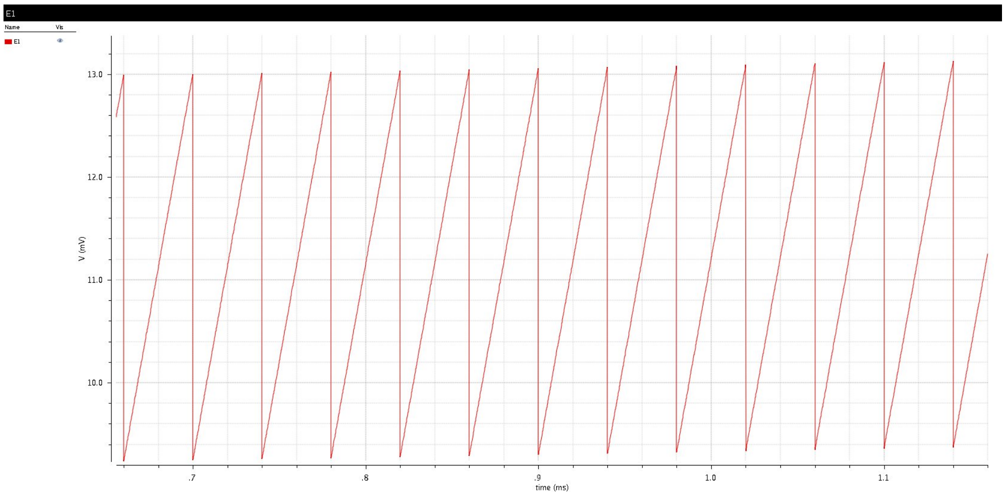


Figure 4.1: Showing the INL error

4.3 Corners

This micro controller is supposed to be used by consumers, and needs therefore to complete tests showing all expected variations it can be postponed for.

Table 4.4: Corners

Corner	**ENOB**	**SNR**	**SFDR**
Temperature -40°C	7.915	49.41	64.05
Temperature +85°C	7.804	48.74	57.18
Transistor fast	7.879	49.19	60.80
Transistor slow	7.843	48.98	57.76

Everything works as supposed for all different corners. There are some small changes to the performance, but noting critical.

4.4 Matching

Matching simulation is done by using the models sent from productions facilities. For the results in table 4.5 Latin Hypercube in Mote Carlo simulation, to indicate as many different possibilities as possible. This was done with 50 runs, more should of course be better, but this simulation is time taking 50 runs should be enough.

Table 4.5: Matching results

	Min	**Max**	**Mean**	**Median**	**Standard deviation**
ENOB	7.385	7.984	7.825	7.856	122.2m
SNR	46.22	49.83	48.87	49.05	735.7m
SFDR	56.37	67.1	62.14	62.20	2.694

We can see from the results in table 4.5 mean and median is as specified. There are a little problem with with the worst case scenario. If we look at the standard deviation and the mean value, calculations will show that less than 0.5% of all productions will fall out of the specified region.

4.5 FOM

The logic here is not implemented on gate level and therefore the power consumption is not assured for. Estimates indicate that it should be around 27nA. If we add this current flow to our equation for FOM we can compare it to other designs. With this estimate out energy consumptions goes up to 88.5nW. This equals to a FOM of 7.43 fJ/conversion step.

Table 4.6: Compared to other work, with estimated power consumption in logic

-	**This design**	**Dai Zhang** [31]	**Jixuan Xiang** [30]	**Li Lun** [21]
Technology	130nm	130nm	65nm	180nm
Samplings rate	50k	1k	400Ms	40k
ENOB	7.825	9.12	7.83	10.75
SNR	49.05	56.7	48.6	
SFDR	62.14	67.6	60.7	
Power consump.	88.5nW*	53nW	0.766mW	10 μ W
FOM	7.43jF/conv*	94.5fJ/conv.	7.9fJ/conv.	322 fJ/conv

Chapter 5

Conclusion

An ADC according to the specifications given have been designed on transistor level. The characteristics is satisfying all over. As always there is possibilities for improvements, here mostly concerning the power consumption in the DAC. All capacitors are fairly large and there should be possibilities to scale them down. Overall we can see in table 4.6 that this is a good ADC concerning the energy efficiency. It is expected that the power consumption will rise when the layout is done due to added parasitics.

Bibliography

[1] *Analog Characterization - 130nm common platform.*

[2] *Tutorial 1040, coherent sampling vs. window sampling.*

[3] H. Balasubramaniam, W. Galjan, W.H. Krautschneider, and H. Neubauer. 12-bit hybrid c2c dac based sar adc with floating voltage shield. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1–5, Nov 2009.

[4] L. Chang and W. Haensch. Near-threshold operation for power-efficient computing? it depends. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 1155–1159, June 2012.

[5] Jung-Sheng Chen and Ming-Dou Ker. The impact of gate-oxide breakdown on common-source amplifiers with diode-connected active load in low-voltage cmos processes. *Electron Devices, IEEE Transactions on*, 54(11):2860–2870, Nov 2007.

[6] Yanfei Chen, Xiaolei Zhu, Hirotaka Tamura, M. Kibune, Y. Tomita, T. Hamada, M. Yoshioka, K. Ishikawa, T. Takayama, J. Ogawa, S. Tsukamoto, and T. Kuroda. Split capacitor dac mismatch calibration in successive approximation adc. In *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, pages 279–282, Sept 2009.

[7] Yanfei Chen, Xiaolei Zhu, Hirotaka Tamura, M. Kibune, Y. Tomita, T. Hamada, M. Yoshioka, K. Ishikawa, T. Takayama, J. Ogawa, S. Tsukamoto, and T. Kuroda. Split capacitor dac mismatch calibration in successive approximation adc. In *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, pages 279–282, Sept 2009.

[8] Yung-Hui Chung, Meng-Hsuan Wu, and Hung-Sung Li. A 12-bit 8.47-fj/conversion-step capacitor-swapping sar adc in 110-nm cmos. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 62(1):10–18, Jan 2015.

[9] Ken Martin David A Jones. *Analog integrated circuit design.* Wiley, 1997.

[10] P.M. Figueiredo and J.C. Vital. Low kickback noise techniques for cmos latched comparators. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 1, pages I–537–40 Vol.1, May 2004.

[11] I.M. Filanovsky, J.K. Jarvenhaara, and N.T. Tchamov. On design of low-voltage cmos current amplifiers. In *Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on*, pages 563–566, Aug 2014.

[12] B.P. Ginsburg and A.P. Chandrakasan. An energy-efficient charge recycling approach for a sar converter with capacitive dac. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 184–187 Vol. 1, May 2005.

[13] S. Guhados, P.J. Hurst, and S.H. Lewis. A pipelined adc with metastability error rate $< 10^{-15}$ errors/sample. *Solid-State Circuits, IEEE Journal of*, 47(9):2119–2128, Sept 2012.

[14] S. Guhados, P.J. Hurst, and S.H. Lewis. A pipelined adc with metastability error rate 10 errors/sample. *Solid-State Circuits, IEEE Journal of*, 47(9):2119–2128, Sept 2012.

[15] Weibo Hu, Yen-Ting Liu, Tam Nguyen, D.Y.C. Lie, and B.P. Ginsburg. An 8-bit single-ended ultra-low-power sar adc with a novel dac switching method and a counter-based digital control circuitry. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(7):1726–1739, July 2013.

[16] Guan-Ying Huang, Soon-Jyh Chang, Chun-Cheng Liu, and Ying-Zu Lin. 10-bit 30-ms/s sar adc using a switchback switching method. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(3):584–588, March 2013.

[17] Simon Josephsen. A comparison of low power dacs for a 9-bit 50 ms/s sar-adc. Master's thesis, NTNU, 2012.

[18] V. Katyal, R.L. Geiger, and D.J. Chen. A new high precision low offset dynamic comparator for high resolution high speed adcs. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 5–8, Dec 2006.

[19] Chun-Cheng Liu, Soon-Jyh Chang, Guan-Ying Huang, and Ying-Zu Lin. A 10-bit 50-ms/s sar adc with a monotonic capacitor switching procedure. *Solid-State Circuits, IEEE Journal of*, 45(4):731–740, April 2010.

[20] Tsung-Che Lu, Lan-Da Van, Chi-Sheng Lin, and Chun-Ming Huang. A 0.5v 1ks/s 2.5nw 8.52-enob 6.8fj/conversion-step sar adc for biomedical applications. In *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, pages 1–4, Sept 2011.

[21] Li Lun, Liu Dongsheng, Lei Weila, Hu Yu, Zou Xuecheng, and Li Dawei. A low power charge-redistribution sar adc with a monotonic switching procedure. In *Solid-State and Integrated Circuit Technology (ICSICT), 2014 12th IEEE International Conference on*, pages 1–3, Oct 2014.

[22] J.L. McCreary and P.R. Gray. All-mos charge redistribution analog-to-digital conversion techniques. i. *Solid-State Circuits, IEEE Journal of*, 10(6):371–379, Dec 1975.

[23] M. Mohammadi and K.D. Sadeghipour. A 0.5v 200mhz offset trimmable latch comparator in standard 0.18um cmos process. In *Electrical Engineering (ICEE), 2013 21st Iranian Conference on*, pages 1–4, May 2013.

[24] Jrgen Moe Sandvik. En variabel bit lengde 9-bit 50ms/s sar adc. Master's thesis, NTNU, 2012.

[25] C.J. Solis and G.O. Ducoudray. High resolution low power 0.6 cmos 40mhz dynamic latch comparator. In *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, pages 1045–1048, Aug 2010.

[26] Hung-Yen Tai, Hung-Wei Chen, and Hsin-Shu Chen. A 3.2fj/c.-s. 0.35v 10b 100ks/s sar adc in 90nm cmos. In *VLSI Circuits (VLSIC), 2012 Symposium on*, pages 92–93, June 2012.

[27] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E.A.M. Klumperink, and B. Nauta. A 10-bit charge-redistribution adc consuming 1.9 $\mu wat 1 ms/s$. *Solid-State Circuits, IEEE Journal of*, 45(5):1007–1015, May 2010.

[28] N. Verma and A.P. Chandrakasan. An ultra low energy 12-bit rate-resolution scalable sar adc for wireless sensor nodes. *Solid-State Circuits, IEEE Journal of*, 42(6):1196–1205, June 2007.

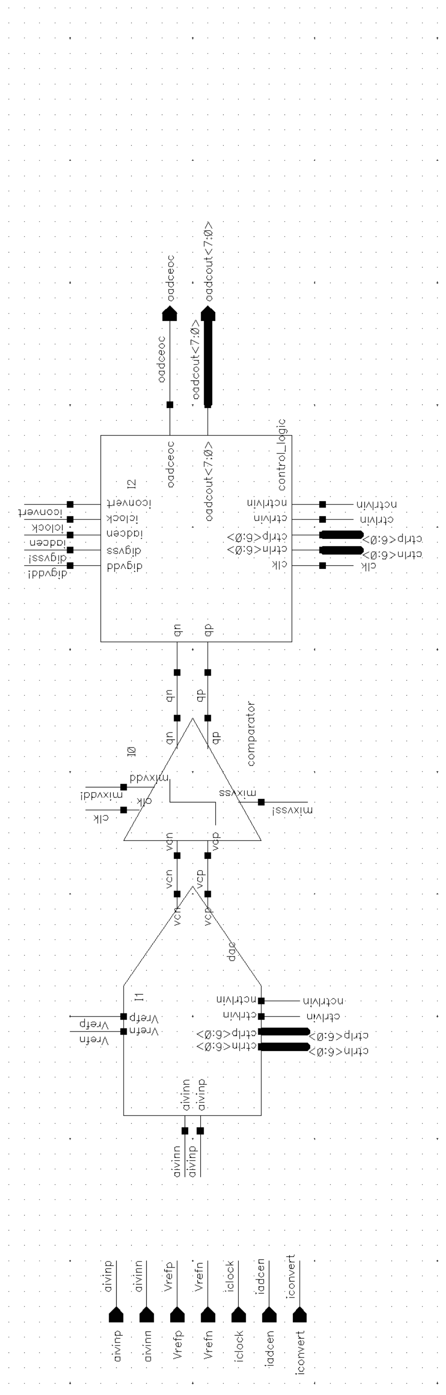
[29] Robert H. Walden. Analog-to-digital converter survey and analysis. In *IEEE Journal onSelected Areas in Communication*, pages 17(4):539–550, April 1999.

[30] Jixuan Xiang, Jian Mei, Hao Chang, and Fan Ye. A 7.9-fj/conversion-step 8-b 400-ms/s 2-b-per-cycle sar adc with a preset capacitive dac. In *ASIC (ASICON), 2013 IEEE 10th International Conference on*, pages 1–4, Oct 2013.

[31] Dai Zhang, A. Bhide, and A. Alvandpour. A 53-nw 9.12-enob 1-ks/s sar adc in 0.13nm cmos for medical implant devices. In *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pages 467–470, Sept 2011.

Appendix

5.0.1 ADC-core



oadceoc
oadcout<7:0>
control_logic
I2
iconvert
iclock
iadcen
digvss!
digvdd!
nctrlvin
ctrlvin
ctrlp<6:0>
ctrln<6:0>
clk
qn
qp
comparator
I0
mixvdd!
mixvss!
vcn
vcp
dac
I1
Vrefp
Vrefn
aivinn
aivinp

5.0.2 Input switch

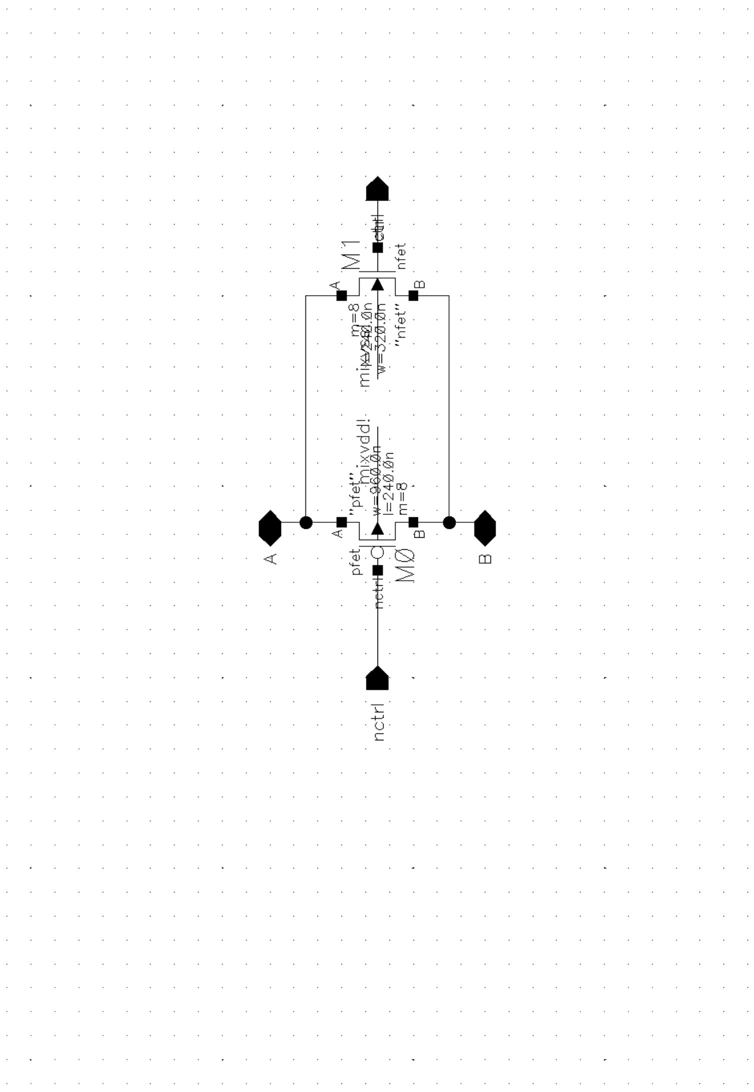


Figure 5.2: Input switch

5.0.3 Inverter

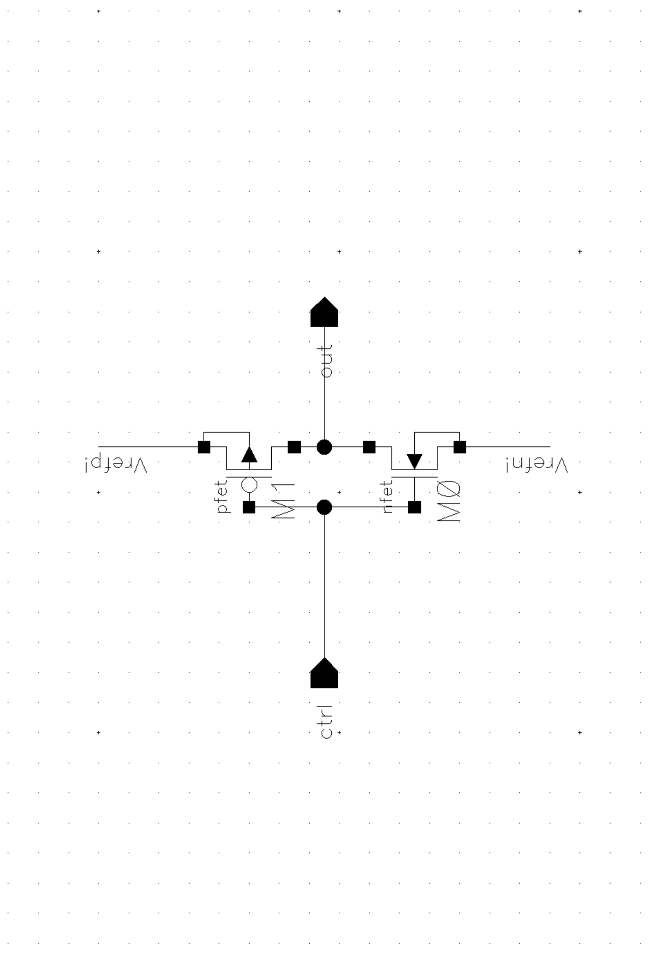


Figure 5.3: Inverter

5.0.4 Full test bench

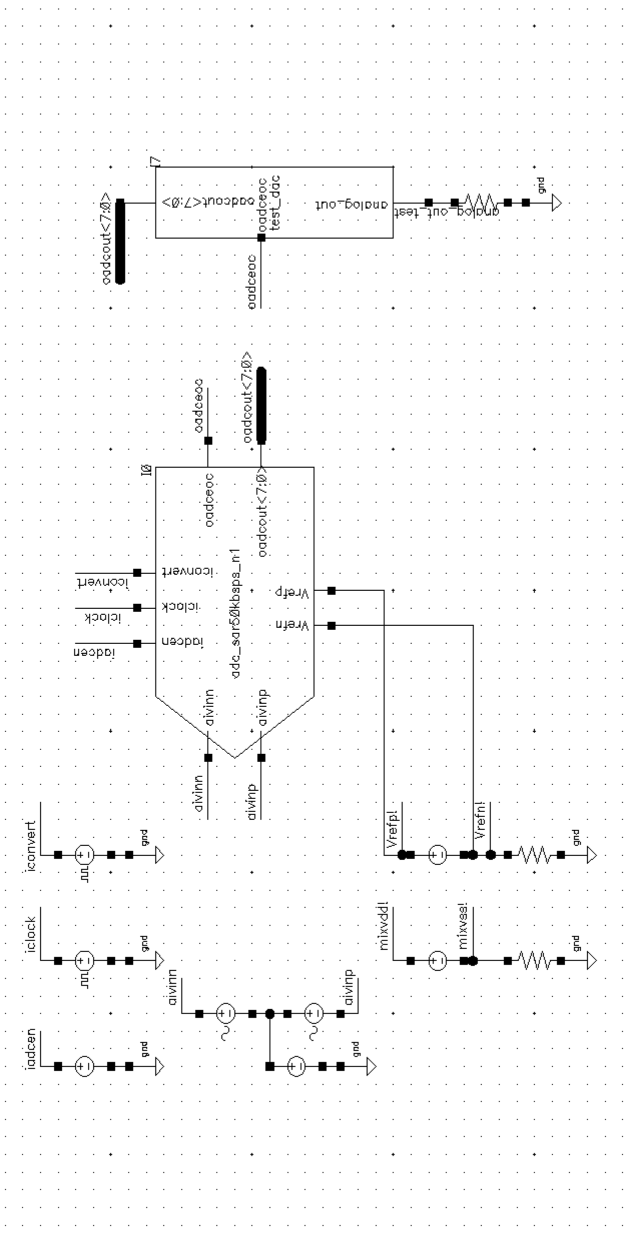


Figure 5.4: Main Test bench

5.0.5 DAC verilog code, for early simulations

```
// VerilogA for adc_sar_project, DAC_verilog, veriloga

`include "constants.vams"
`include "disciplines.vams"

module DAC_verilog (aivinp, aivinn, vrefp, vrefn, ctrlvin, nctrlvin, ctr

 //signals
  input          aivinp;
  input          aivinn;
  input          vrefp;
  input          vrefn;
  input          ctrlvin;
  input          nctrlvin;
  input [6:0]    ctrlp;
  input [6:0]    nctrlp;
  input [6:0]    ctrln;
  input [6:0]    nctrln;
  output         vcn;
  output         vcp;

 // voltage
  voltage aivinp;
  voltage aivinn;
  voltage vrefp;
  voltage vrefn;
  voltage ctrlvin;
  voltage nctrlvin;
  voltage [6:0] ctrlp;
  voltage [6:0] nctrlp;
  voltage [6:0] ctrln;
  voltage [6:0] nctrln;
  voltage vcn;
  voltage vcp;

  real resp;
  real resn;

  parameter real tr = 10p from [0:inf);
  parameter real td = 10p from [0:inf);
  parameter real tt = 10p from [0:inf);
  parameter real vth = 0.25; //threshould for hig/low
  parameter integer  dir_pos = 1 from [-1:1] exclude 0;
```

```
integer resultp;
integer resultn;
real r_resultp;
real r_resultn;
integer ctrl;
real i_vin_p;
real i_vin_n;
real i_res_p;
real i_res_n;
genvar i;


analog begin

   ctrl=(V(ctrlvin)>0.25) ? 1:0;

  @(cross(ctrl - 0.25, dir_pos)) begin
     i_vin_p=V(aivinp);
     i_vin_n=V(aivinn);
   end

     resultp = 0;
     resultn = 0;

     for(i=0;i<7;i=i+1)begin
       if(V(ctrlp[i])>vth)begin
          resultp = resultp +pow(2,i);
       end
       if(V(ctrln[i])>vth)begin
          resultn = resultn +pow(2,i);
       end

     end

     r_resultp = resultp;
     r_resultn = resultn;
     i_res_p=((r_resultp/256)-i_vin_p);
     i_res_n=(((r_resultn)/256)-i_vin_n);
     resp = i_res_p;
     resn = i_res_n;
    V(vcp)<+ transition (resp, tr, td, tt);
    V(vcn)<+ transition (resn, tr, td, tt);
  end // UNMATCHED !!


endmodule
```

5.0.6 Comparator verilog, for early simulations

```
// VerilogA for adc_sar_project, comp_verilog, veriloga

`include "constants.vams"
`include "disciplines.vams"

module comp_verilog(vcp, vcn, iibiascomp, mixvdd, mixvss, clk, qn, qp);

  input vcp;
  input vcn;
  input iibiascomp;
  input mixvdd;
  input mixvss;
  input clk;
  output qn;
  output qp;

  voltage vcp;
  voltage vcn;
  voltage iibiascomp;
  voltage mixvdd;
  voltage mixvss;
  voltage clk;
  voltage qn;
  voltage qp;

  parameter real vhigh = 0.5;
  parameter real vlow = 0;
  parameter real tr = 50n; // risetime
  parameter real tf = 50n; // falltime
  parameter real td = 0.625u; // dealy
  parameter real offset = 0.0001;
  integer CLK_p;
  integer CLK_n;
  parameter integer dir_pos = 1 from [-1:1] exclude 0;
  parameter integer dir_neg = -1 from [-1:1] exclude 0;
  real input_pos, input_neg;
  real output_pos, output_neg;


analog begin

  input_pos = V(vcp);
  input_neg = V(vcn)+offset;

  CLK_p = (V(clk)>0.25) ? 1:0;
  CLK_n = (V(clk)<0.25) ? 0:1;

  @(cross(CLK_p-0.25, dir_pos)) begin
    output_pos = vlow;
    output_neg = vlow;
  end //

  @(cross(CLK_p-0.25, dir_neg)) begin
    if(input_pos > input_neg) begin
      output_pos = vhigh;
      output_neg = vlow;
    end
    else begin
      output_pos = vlow;
      output_neg = vhigh;
    end
  end

  V(qp) <+ transition(output_pos, td, tr, tf);
  V(qn) <+ transition(output_neg, td, tr, tf);

end // UNMATCHED !!



endmodule
```

5.0.7 Digital Logic

```
// VerilogA for nwy_adc_sar8b50ksps_n1, control_logic, veriloga

`include "constants.vams"
`include "disciplines.vams"

module control_logic(qp, qn, iconvert, iclock, iadcen, ctrlp, ctrln, ctr
  //signals
  input qp;
  input qn;
  input iconvert;
  input iclock;
  input iadcen;
  input digvdd;
  input digvss;
  output [6:0] ctrlp;
  output [6:0] ctrln;
  output ctrlvin;
  output [7:0] oadcout;
  output oadceoc;
  output clk;

  //voltages
  voltage qp;
  voltage qn;
  voltage iconvert;
  voltage iclock;
  voltage iadcen;
  voltage digvdd;
  voltage digvss;
  voltage [6:0] ctrlp;
  voltage [6:0] ctrln;
  voltage ctrlvin;
  voltage [7:0] oadcout;
  voltage oadceoc;
  voltage clk;


 //internal paramters
  parameter real vth = 0.25;
  parameter real td = 10n from [0:inf);
  parameter real tt = 10n from [0:inf);
  parameter real vdd = 0.5;
  parameter real vss = 0;
  parameter integer dir = +1 from [-1:1] exclude 0;
  parameter integer dir_neg = -1 from [-1:1] exclude 0;
  integer counter;
  genvar i;

// internal signals
  integer i_ctrlvin;
integer i_ctrlp;
integer i_ctrln;
integer i_oadceoc;
integer i_convert;
integer i_dout;
integer i_adcen;
integer i_qp;
integer i_qn;
integer CLKL;
integer result;
integer i_clk;
integer out_clk;

analog begin

  @(initial_step)begin
    counter = 0;
    i_ctrlp = 127;
    i_ctrln = 127;
    i_ctrlvin = 0;
    i_oadceoc = 0;
    result = 0;
  end // @inital_step

  i_convert = V(iconvert) > vth ? 1:0;
  @(cross(i_convert-vth, dir))begin
    // When iconvert goes high
    counter = 8;
    i_ctrlvin = 1;
    i_ctrlp = 127;
    i_ctrln = 127;
    i_oadceoc = 0;
    result = 511;
  end//positive iconvert flank



  i_qp = (V(qp)>vth ? 1:0);
  i_qn = (V(qn)>vth ? 1:0);
  CLKL = (i_qp^i_qn); // qp XOR qn

  @(cross(CLKL-vth, dir_neg))begin
    if(V(iconvert) < vth)begin
      if(counter > 0)begin
        i_ctrlvin=0;
```

```
          counter = counter -1;
        end
        if(counter < 1)begin
          i_oadceoc = 1;
          i_dout = result;
        end
      end
    end // @ (cross(CLKL-vth, dir_neg))

    @(cross(CLKL-vth, dir))begin
      if(counter < 8)begin
        if(V(qp)>V(qn))begin
          result = result &~(1<<(counter));
          i_ctrlp = i_ctrlp &~(1<<(counter -1));
        end
        else if(V(qp) < V(qn)) begin
          i_ctrln = i_ctrln &~(1<<(counter -1));
        end
      end
    end // @

    for(i=0; i<8; i=i+1)begin
      V(oadcout[i]) <+ transition(i_dout &(1<<i) ? vdd : vss, td, tt);
    end

    V(oadceoc) <+ transition(i_oadceoc ? vdd : vss, td, tt);
    V(ctrlvin) <+ transition(i_ctrlvin ? vdd : vss, td, tt);

    for(i=0; i<7;i=i+1)begin
      V(ctrlp[i]) <+ transition(i_ctrlp & (1<<i) ? vss : vdd, td, tt);
      V(ctrln[i]) <+ transition(i_ctrln & (1<<i) ? vss : vdd, td, tt);
    end


    i_adcen = (V(iadcen) > vth) ? 1:0;
    i_clk = (V(iclock) > vth) ? 1:0;
    out_clk=(i_adcen & i_clk);

    V(clk) <+ transition(out_clk ? vdd : vss, td, tt);

  end // analog begin
```

```
endmodule
```

5.0.8 DAC for test benches

```
// VerilogA for adc_sar_project, test_dac, veriloga

`include "constants.vams"
`include "disciplines.vams"

module test_dac(oadcout, analog_out, oadceoc);

  input [7:0]oadcout;
  input oadceoc;
  output analog_out;

  voltage [7:0]oadcout;
  voltage analog_out;
  voltage oadceoc;

  integer i_value;
  real i_out;
  integer CLK;
  genvar i;

  parameter real tr = 10n from [0:inf);
  parameter real td = 10n from [0:inf);
  parameter real tt = 10n from [0:inf);
  parameter real vth = 0.1;
  parameter integer dir_pos = 1 from [-1:1] exclude 0;


  analog begin

    CLK = (V(oadceoc)>0.25) ? 0:1;

    @(cross(CLK-0.25 , dir_pos)) begin

      i_value = 0;
      i_out=0;

      for (i=0; i<8 ; i=i+1) begin
```

```
          if (V(oadcout[i]) > vth) begin
            i_value = i_value + pow(2,i);
          end
        i_out = (i_value);
      end


        end // UNMATCHED !!
        V(analog_out) <+  (i_out)/512;

  end // UNMATCHED !!



endmodule
```