7 Converters and Sensors

In this chapter, we shall discuss the basic principles of data converters and give an overview of their architecture. The data converter design and implementation is also discussed, with emphasis on basic building blocks, the use of comparators, voltage reference, sampling structures, etc… We also discuss the implementation of temperature and light sensors, compatible with the CMOS standard process.

1. Introduction

Our environment is full of analog signals that we need to monitor, to capture, to treat, to store, to modify and transmit, such as sound, temperature, humidity, light, radio frequency waves or acceleration. A modern way to treat analog signals is to convert them into digital signals. The advantages of using digital techniques called signal processing are the programmability, the stability, the repeatability, the accuracy, the noise immunity, but also the ability to implement special functions such as linear phase filters or error correcting codes.

The digital signal is a variable whose possible values are 0 or 1, which corresponds to a low or a high voltage. As the opposite of an analog signal is a continuous time signal whose response with respect to time is uninterrupted.

The analog to digital converters (ADC) and digital to analog converters (DAC) are the main links between the analog signals and the digital world of signal processing. The ADC and DAC viewed as black boxes are shown in figure 7-1. On the right side, the ADC takes an analog input signal *Vin* and converts it to a digital output signal *A*. The digital signal *A* is a binary coded representation of the analog signal using N bits: A_{N-1} … A_0. The maximum number of codes for N bits is 2^N. The digital signal is usually treated by a microprocessor unit (MPU) or by a specific digital signal processor (DSP) before being restituted as an output *B*. In this case example, *B* has the same dimension as for the input signal *A*. Then, the DAC, which has the opposite function compared to the ADC, converts the digital signal to the final analog output signal *Vout*.

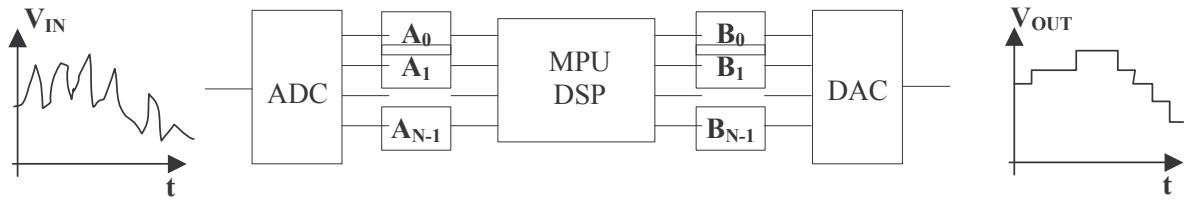


Figure 7-1. Basic principle of N bits analog to digital and digital to analog converters.

A typical function of this signal processing circuit is to filter the high frequency components of the input *Vin*. Consequently, *Vout* only contains the slow-varying portion of *Vin*. The figure below shows some target applications of ADC and DAC converters, with the frequency range in X axis and the converter resolution in Y axis [1][3]. Low frequency, low resolution data conversion mainly concern low quality voice, as found in phones. Mobiles phones typically operate at 8000Hz, 12 bits. Digital audio in CD players works with 20 bit converters at a frequency of 44KHz (figure 7-2). Digital video and high speed internet have created specific demand on high-speed data converters. Finally, sampling rates around 1GHz are still a specificity of very high speed instrumentation such as GHz bandwidth oscilloscopes.

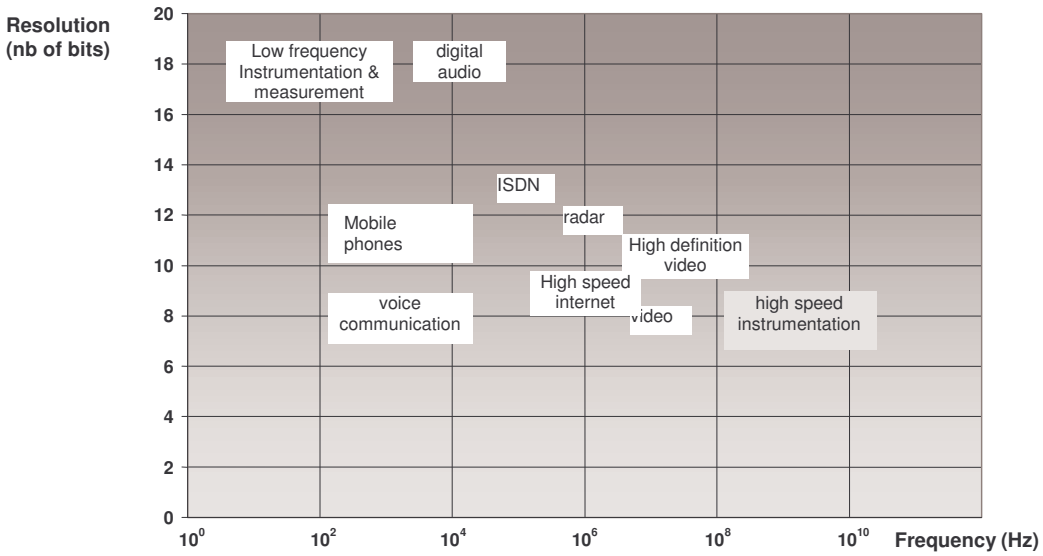


Figure 7-2. Speed and resolution requirements on ADCs

2. Digital-Analog Converters architectures

Digital-to-analog converters (DAC) traduce a digital number *B* into an analog signal V_{OUT}*. The output of the DAC is not as smooth as we could wish, due to a finite number of available analog levels. A low pass filter eliminates the higher order harmonics caused by the conversion on the signal V_{OUT}*, and returns an analog signal V_{OUT} (figure 7-3).

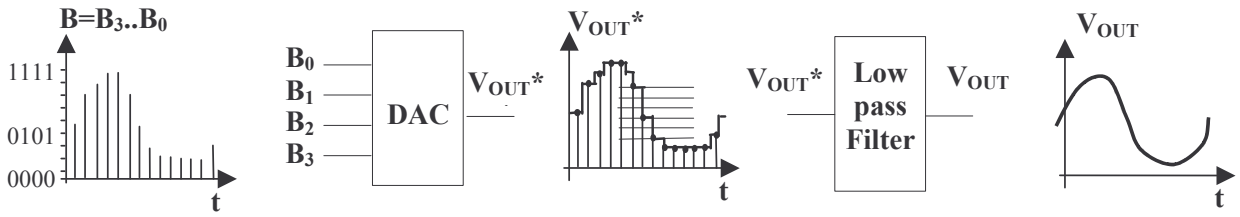


Figure 7-3. A four-bits digital conversion including a filter module.

A wide variety of DAC architectures exists, from very simple to complex ones. Each of them has its own merits and limits. The digital signal can be provided in many different codes, depending on the final application: binary, thermometer, Gray, two's complement , offset binary, and so on. This concepts are presented in the following paragraphs.

Resistor string converter

The most basic DAC is based on a resistance ladder. This type of DAC consists of a simple resistor string of 2^N identical resistors, and a binary switch array whose inputs are a binary word. The analog output is the voltage division of the resistors flowing via pass switches. In the example of figure 7-4, the resistance ladder includes 8 identical resistors which generate 8 reference voltages equally distributed between 0 and *Vdac*.

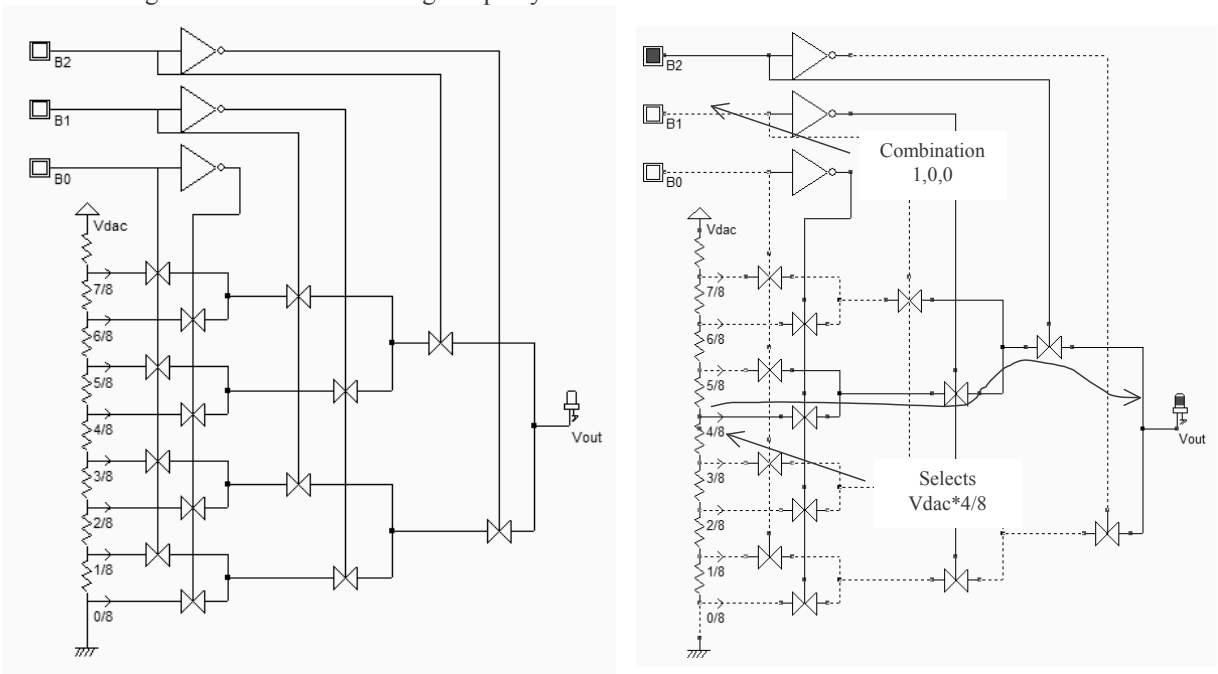
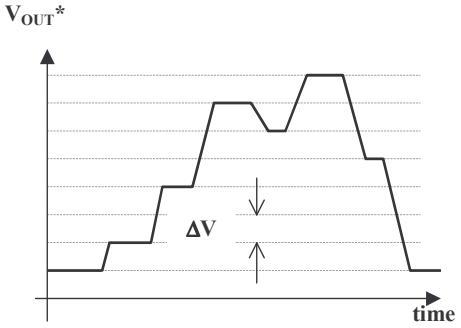


Figure 7-4. Schematic diagram of the digital-analog converter (DAC3bit.SCH)

The digital-analog converter uses the three-bit input (B2,B1,B0) to control the transmission gate network which selects one of the voltage references (A portion of *Vdac*) which is transferred to the output *Vout*. Let us consider *Vdac*=1.2V, which corresponds to the core voltage of the CMOS 0.12µm process. The voltage step can be expressed by equation 7-1.

$$\Delta V = \frac{V_{dac}}{2^N} = \frac{1.2}{8} = 0.15V \qquad \text{(Eq. 7-1)}$$

The correspondence between the input B and the output *Vout** is given in figure 7-5, considering *Vdac*=1.2V.



B2	B1	B0	Vout*	Analog output Vout* (V) with Vdac=1.2V
0	0	0	0/8 Vdac	0.0
0	0	1	1/8 Vdac	0.15
0	1	0	2/8 Vdac	0.3
0	1	1	3/8 Vdac	0.45
1	0	0	4/8 Vdac	0.6
1	0	1	5/8 Vdac	0.75
1	1	0	6/8 Vdac	0.9
1	1	1	7/8 Vdac	1.05

Figure 7-5. The specifications of a 3-bit digital-to-analog converter

Layout considerations

A long path of polysilicon between VDD and VSS may give intermediate voltage references required for the DAC circuit. Unfortunately, the polysilicon has a low resistance due to a surface deposit of metal, called salicidation. The resistance per square is quite small (Around 4 Ω per square) due to this thin metal coat, as seen in the cross-section of figure 7-6. In order to increase the sheet resistance value, the polysilicon resistor must be surrounded by the specific "Option" layer that may be found in the upper part of the palette of layers. The salicide is removed, and the sheet resistance is increased to 40 Ω per square (figure 7-6 right).

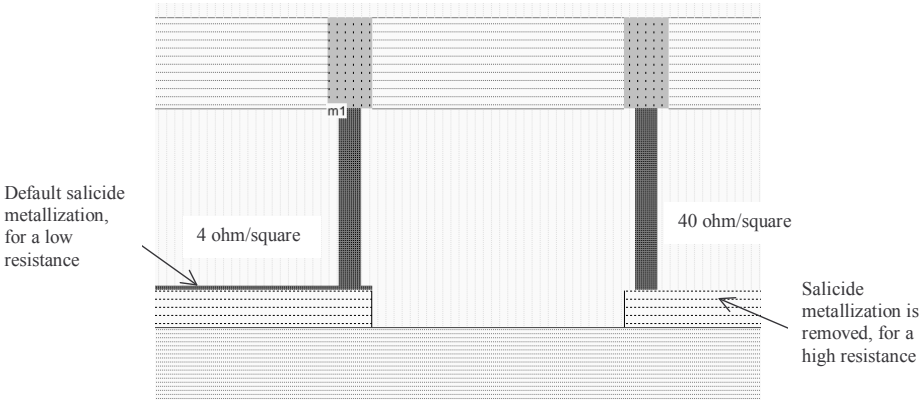


Figure 7-6. Removing the salicidation to increase the sheet resistance

Following a double click in this layer, we activate the property "**Remove salicide to increase resistance**" (Figure 7-7). Consequently, the resistor value is multiplied by 10 and can be used to design an area-efficient resistor network.

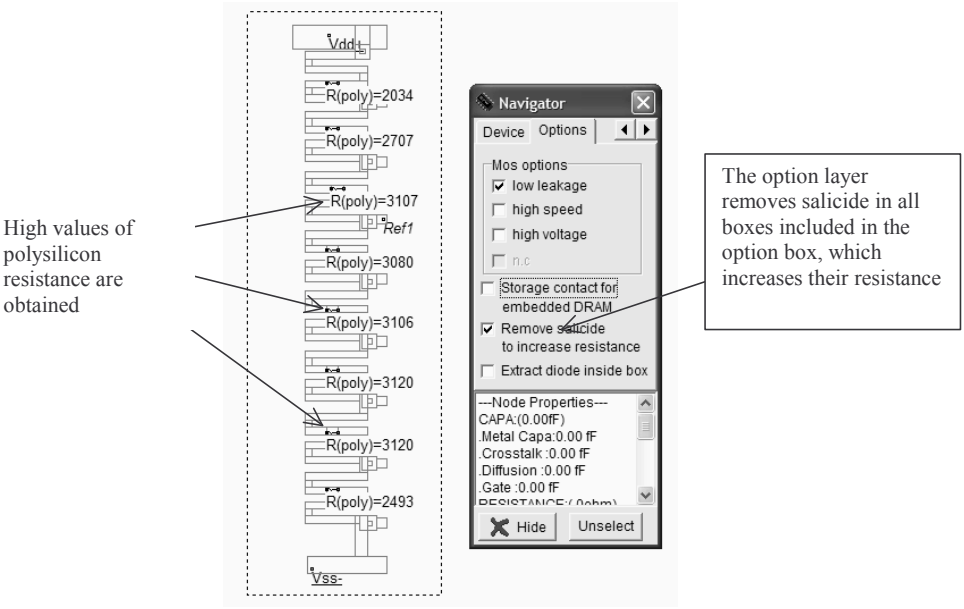


Figure 7-7. The sheet resistance is increased by removing the salicide deposit, thanks to an option layer (ADC.MSK)

The resistor ladder generates intermediate voltage references used by the voltage comparators located in the middle. By default, Microwind does not take into account any serial resistor. This means that the resistor ladder layout on the left of figure 7-8 is considered as a short cut between VDD and VSS.

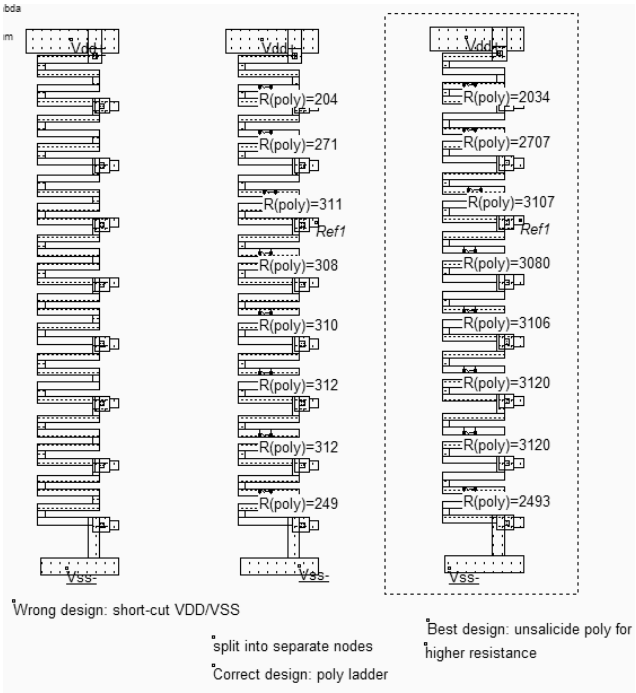


Figure 7-8. Virtual resistor symbols split the polysilicon path into separate electrical regions (ADCRes.MSK)

To account for the serial resistance distributed along the polysilicon path, a virtual resistance symbol must be added, which will force Microwind to split the ladder into separate electrical nodes, and to extract the corresponding polysilicon resistance (Figure 7-8 middle and right). The virtual resistor may be found in the upper part of the palette.

Once inserted, the menu of figure 7-9 appears. It is recommended in this case to select the option **Poly resistance**. At extraction, Microwind will evaluate the equivalent resistance on both sides of the virtual symbol and update the resistance automatically according to the design and technological options.

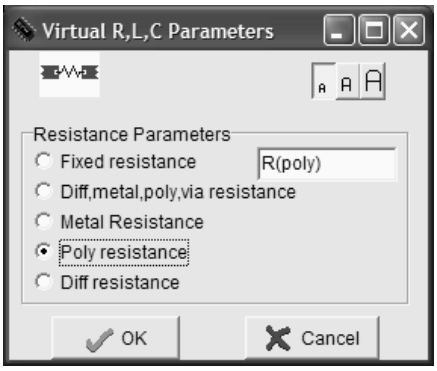


Figure 7-9: Adding a virtual resistor symbol to extract the polysilicon resistance

The resistance symbol is inserted in the layout to indicate to the simulator that an equivalent resistance must be taken into account for the next analog simulation. The layout of the 3-bit digital-to-analog converter is shown in Figure 7-10. The three inverter circuits generate the signals *~B2,~B1* and *~B0* from signals *B2,B1* and *B0*. The transmission gates use minimum MOS device size. The total resistance approaches 24Kohm, which means a stand-by current near 50µA on a 1.2V supply power. Lower DC currents may be obtained by increasing the length of the polysilicon path.

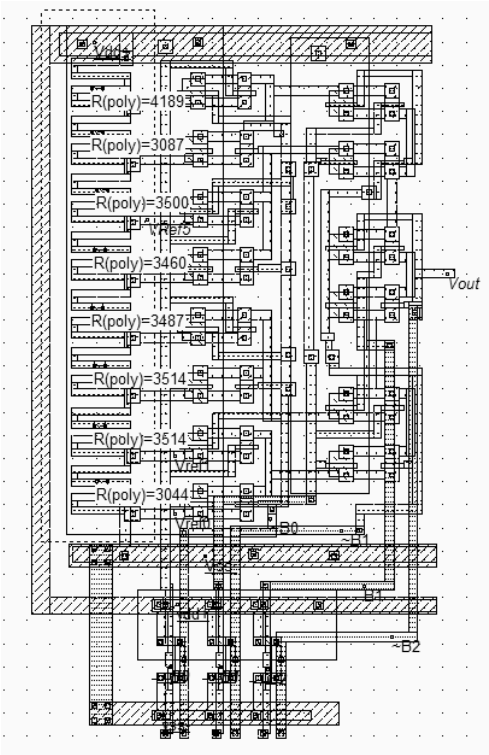


Figure 7-10. Layout of the digital-analog converter (DAC.MSK).

The simulation of the R ladder DAC (Figure 7-11) shows a regular increase of the output voltage *Vout* with the input combinations, from 000 (0V) to 111 (1.2V). Each input change provokes a capacitance network charge and discharge. Notice the fluctuation of the reference voltage *Vref5* (One of the 8 reference voltages) too. This is due to the weak link to VDD and VSS through a highly resistive path.

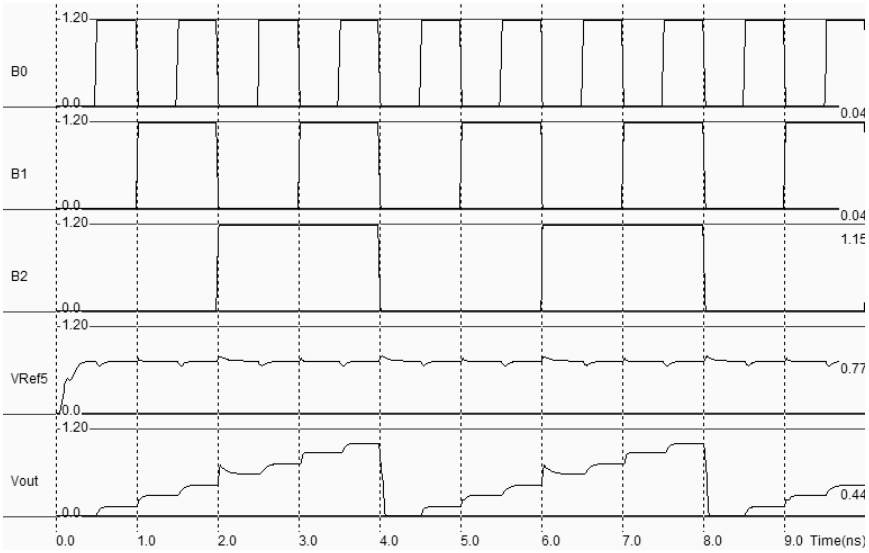


Figure 7-11. Simulation of the digital-analog converter (DAC.MSK).

The analog level *Vout* increases regularly with increasing digit input *B*. The converter is monotonic. However, it must be noticed that for a very short period of time, near t=2.0ns, the internal node discharge leads to a voltage overshoot close to one voltage step ΔV. Also notice that, according to the schematic diagram of figure 7-4, the output is connected to N switches *On* and N switches *Off*.

Converter non-linearity

Due to the non-ideal behavior of switches, process fluctuations, leakages and various gradient effects, there is a small difference between the ideal analog output *Vout_ideal* and the actual analog output *Vout*. The deviation of *Vout* from the ideal value *Vout_ideal* is called the integral non-linearity (INL). The normalized integral non-linearity, according to [3], can be expressed using equation 7-2. The integral non-linearity is illustrated in figure 7-12: when *Vout* is exactly equal to the ideal output, the integral non-linearity INL is equal to 0. However, for several values of *B*, a small difference is usually observed.

$$INL_i = \frac{Vout_i - Vout_{ideal}}{\Delta V} \qquad \text{(Eq. 7-2)}$$

where

INLi=the integral non-linearity for input i (Relative error between -1 and 1)

Vout_i=the real DAC output for input i (V)

Vout_ideal=the ideal DAC output for input i (V)

ΔV=ideal voltage step (V)

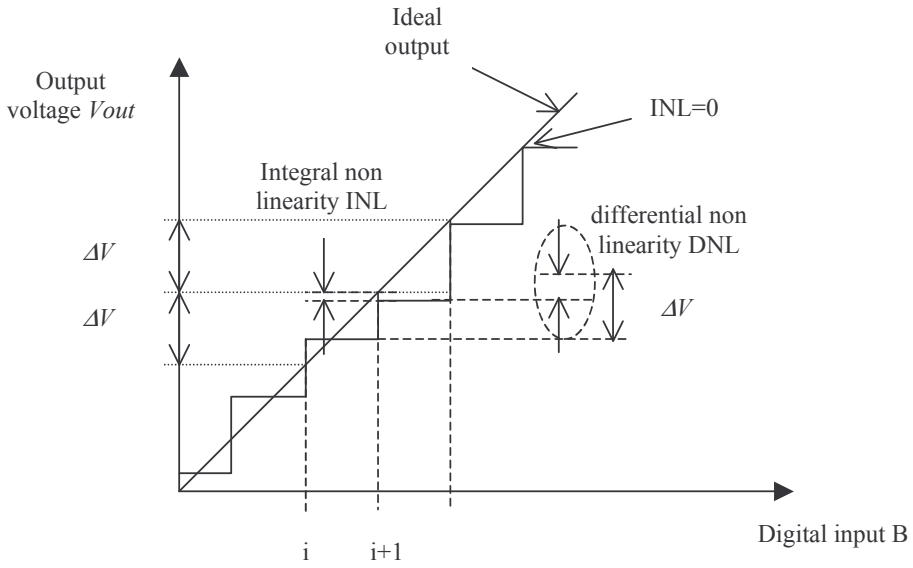


Figure 7-12. The illustration of integrate and differential non-linearity

The difference between two adjacent analog outputs may be significantly different from the theoretical voltage step. This deviation is called the differential non-linearity (DNL). In figure 7-1, the DNL is the vertical mismatch between the ideal voltage step ΔV and the measured step between input *i* and input *i+1*. The normalized differential non-linearity includes the voltage step ΔV to get the relative error, and can be described by equation 7-3.

$$DNL_i = \frac{Vout_{i+1} - Vout_i - \Delta V}{\Delta V} \qquad \text{(Eq. 7-3)}$$

where

DNLi=the differential non-linearity for input i (Relative error, usually between -1 and 1)

$Vout_{i+1}$=the real DAC output for input i+1 (V)

$Vout_i$=the real DAC output for input i (V)

ΔV=ideal voltage step (V)

The illustration of integral non-linearity is given in the simulation of the 3-bit DAC. The ideal reference voltages are placed separately in the layout, as shown in figure 7-13. In the simulation mode **Voltage, Current vs. Time**, all voltage values are placed in the same window. The ladder of reference voltages appears, as well as the DAC output *Vout*. From the simulation shown in figure 7-14, it appears clearly that an integral non-linearity exists which corresponds to the difference between the ideal and actual value of the output, for some values of *B*. The origin of this non-linearity is the resistor ladder design which does not create perfectly regular resistance values. Remember that process fluctuation may affect the value of the resistance, which is one other source of non-linearity.

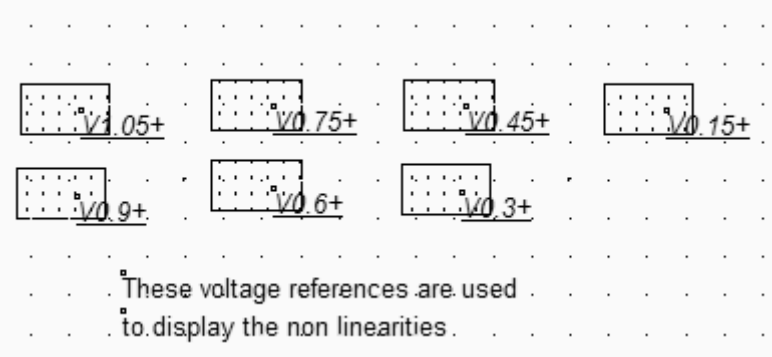


Figure 7-13. The illustration of integral and differential non-linearity (DacNonLinearity.MSK)

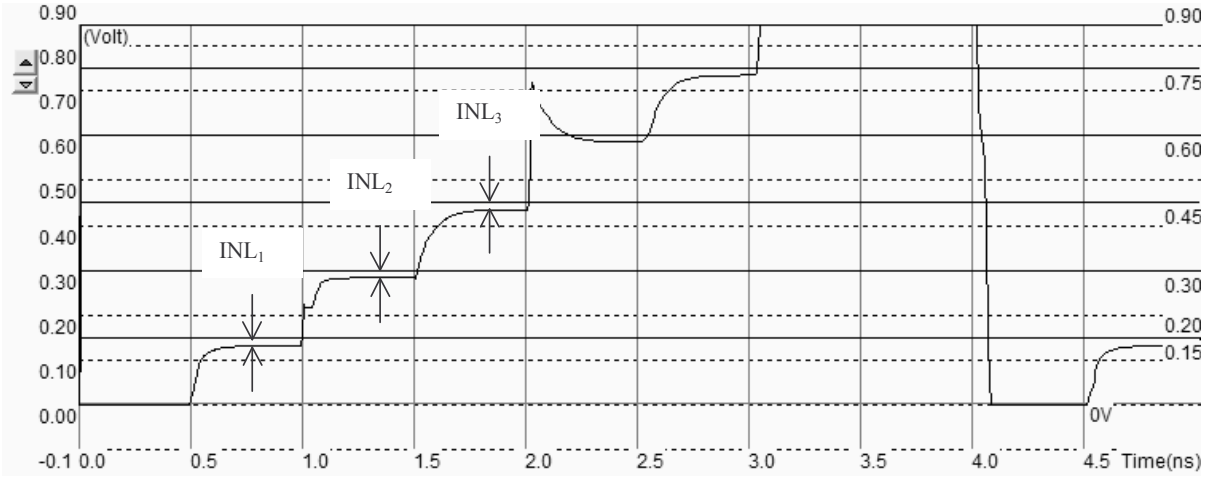


Figure 7-14. A zoom at the analog voltage Vout reveals a non negligible integral non linearity (DacNonLinearity.MSK)

R-2R ladder converter

It is not easy to construct a resistor-based DAC with a high resolution, due to the resistance spread and to the needs for 2^N serial resistors. A more compact choice is the R-2R ladder [Gustavsson]. Its configuration consists of a network of resistors alternating between R and 2R. For a N bits DAC, only N cells based on 2 resistors R and 2R in series are required. The 4-bit and 8-bit implementation of this circuit are reported in figure 7-15. At the right end of the network is the output voltage *Vout*.

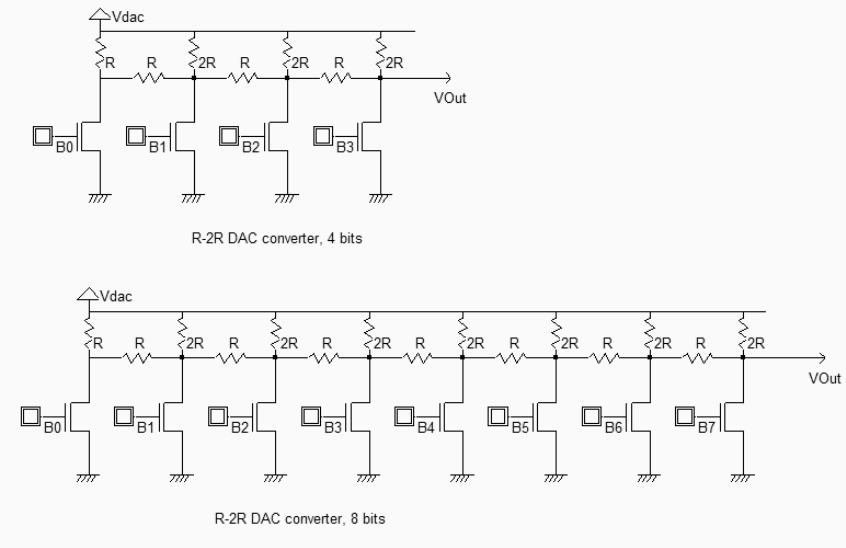


Figure 7-15. 4-bit and 8-bit DAC converter using the R-2R ladder (DACR2R.SCH)

Seven resistors were used for the 4-bit implementation of the R-2R DAC, that is half of the previous R-ladder. The difference is even more significant in the 8-bit circuit, with only 15 resistors, while the simple ladder would require 255 resistors in series.

In the 4-bit implementation of the DAC, the digital inputs (B_3,B_2 B_1 B_0) determine whether each cell is switched to ground or tied to V_{dac}. Each cell's output voltage is a ratio of V_{dac} because of the ladder network voltage division. The final output voltage V_{OUT} depends on the value of *B (0* to *15)*, following the formula 7-4:

$$V_{OUT} = V_{dac} \cdot \frac{(2^N - B)}{2^N} \qquad \text{(Eq. 7-4)}$$

On this principle, table 7-1 gives the value of V_{OUT} versus the input code, with V_{dac} equal to 1.2V.

B_3	B_2	B_1	B_0	V_{OUT}
0	0	0	0	1.2
0	0	0	1	1.125
0	0	1	0	1.05
0	0	1	1	0.975
0	1	0	0	0.9
0	1	0	1	0.825
0	1	1	0	0.75
0	1	1	1	0.675
1	0	0	0	0.6
1	0	0	1	0.525
1	0	1	0	0.45
1	0	1	1	0.375
1	1	0	0	0.3
1	1	0	1	0.225
1	1	1	0	0.15
1	1	1	1	0.075

Table 7-1. Output voltage produced by the 4-bit R-2R DAC versus input code B

Layout considerations:

The resolution of the R-2R DAC is linked with the accuracy of the resistors and of the resistance of the switches which must be negligible to avoid a voltage drop and associated non-linearity. It is important to implement a low R_{on} switch (Large width, minimum length), together with large resistors. In figure 7-16, the design of a four-bit digital-to-analog converter is reported. The elementary resistor pattern has a fixed value of 500 ohm.

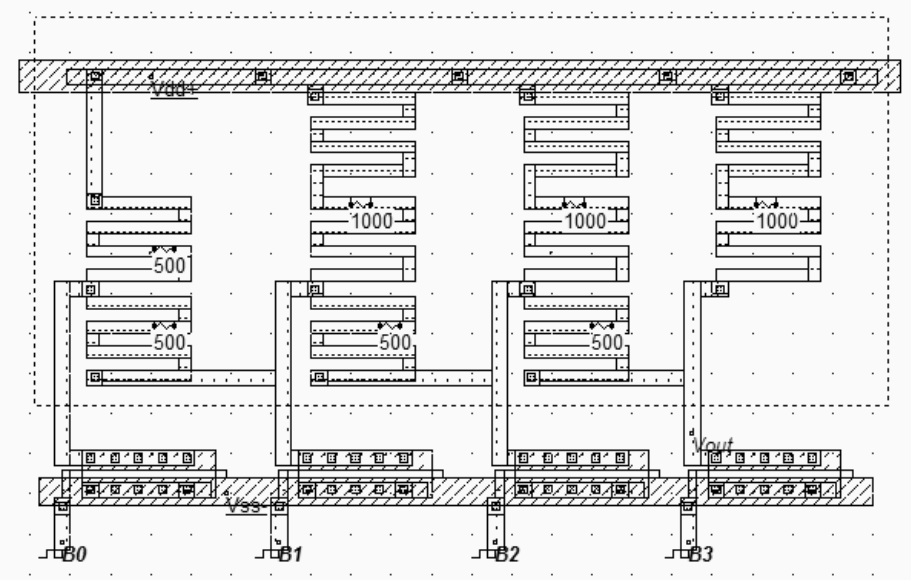


Figure 7-16. A four-bit 2-2R digital to analog converter (DacR2R4Bit.MSK)

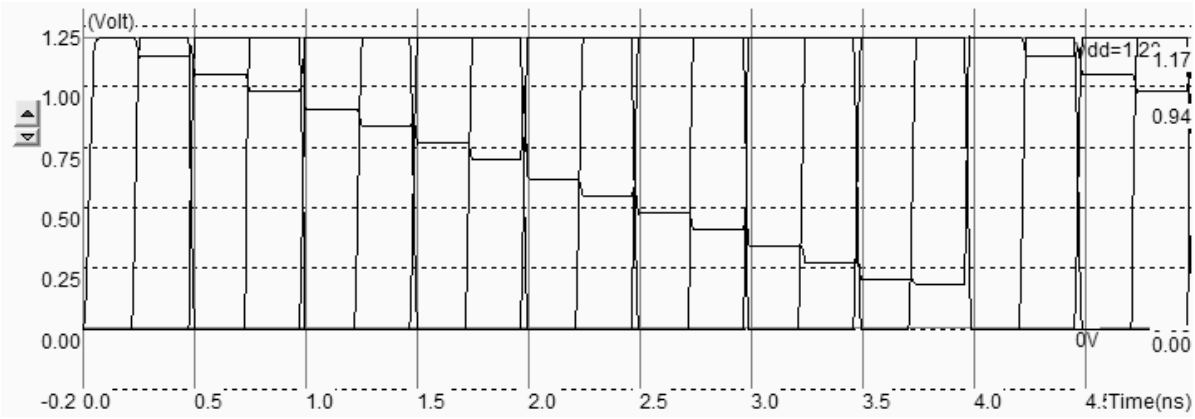


Figure 7-17. Simulation of the 4-bit 2-2R digital to analog converter (DacR2R4Bit.MSK)

The simulation of the four bit 2-2R digital to analog converter (Figure 7-17) shows a regular decrease of the output voltage *Vout*. As the *Ron* of the MOS devices is not negligible, the final value Vout (B=1111) is higher than the predicted 0.075V (Table 7-1). This non-linearity may be corrected by enlarging the MOS switch and increasing the length of the serpentine resistor. Alternatively, a dummy switch, whose pass resistance is half R_{on}, may be inserted inside each cell in serial with R.

Switched capacitors

A very popular DAC architecture used in CMOS technology is based on switched capacitor [Baker]. An array of capacitors is connected to switches, in parallel, as described in figure 7-18. The capacitors are connected in parallel and share one common electrode which is connected to a follower amplifier. Notice that the capacitors are binarly weighted, which means that C, 2C, 4C capacitances are implemented. The capacitor array totals 2^N C. Figure 7-18 gives an example of 3-bit (B_0,B_1,B_2) charge scaling DAC.

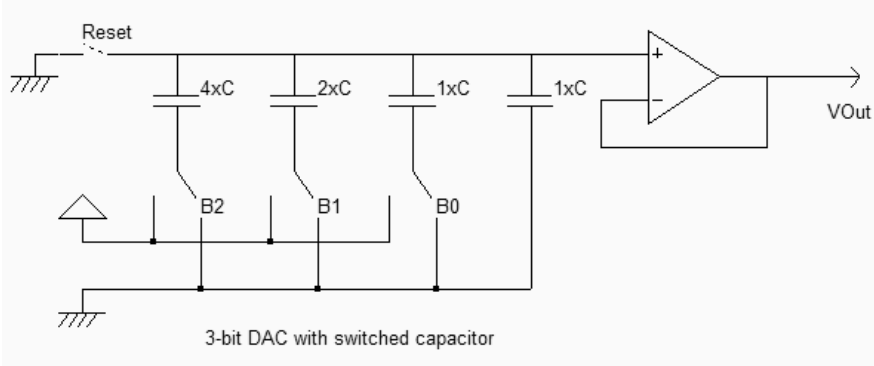
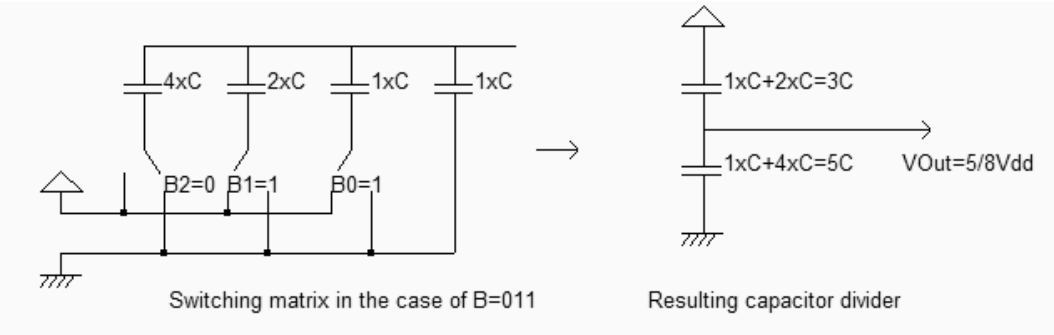


Figure 7-18. The charge-scaling digital to analog converter (DacCapacitor.SCH)

The first step is to discharge all capacitors thanks to the *Reset* switch connected to the ground. Then the switch is disconnected. After the initialization, the digital switches B_0,B_1 and B_2 connect each capacitor to VDD or to VSS, according to the logic value. The output voltage *Vout* is then a function of the voltage division between capacitors. As an example, if the number to convert *B=011*, B_2 is connected to VSS, B_1 and B_0 are connected to VDD as shown in figure 7-19. The equivalent capacitor divider corresponds to a value of the output equal to 5/8Vdd. The conversion table gives the value of *Vout* versus the input code.



B2	B1	B0	Vout/VDD
0	0	0	0
0	0	1	1/8
0	1	0	2/8
0	1	1	3/8
1	0	0	4/8
1	0	1	5/8
1	1	0	6/8
1	1	1	7/8

Figure 7-19. The digital to analog converter at work for B=011 (DacCapacitor.SCH)

Layout considerations

The most important problem is the design of precise capacitors, with values from *C* to $2^N xC$. As the number of bits increases, the ratio of MSB to LSB capacitor becomes difficult to control. Moreover, high value capacitors have an important size on the chip. Using metal plates to create them is not realist as the capacitance value per μm^2 is very low. The solution is to use passive double polysilicon capacitors if available in the CMOS process, as they have good matching accuracy and high capacitance value per μm^2. In 0.12µm CMOS technology, the capacitance between metals is around 50 aF/μm^2 and rises to 2000 aF /μm^2 between polysilicon and poly2.

The implementation of the 3-bit DAC requires 8 sets of capacitor, regrouped in 4xC, 2xC and two separate C. In Microwind, the command **Edit → Generate → Capacitor** gives access to a specific menu for generating capacitor (Figure 7-20). The default capacitor is made with poly/Poly2. The typical capacitance value for C is around 1pF. As may been seen in the layout shown in figure 7-21, a 100fF value for C already leads to a large layout.

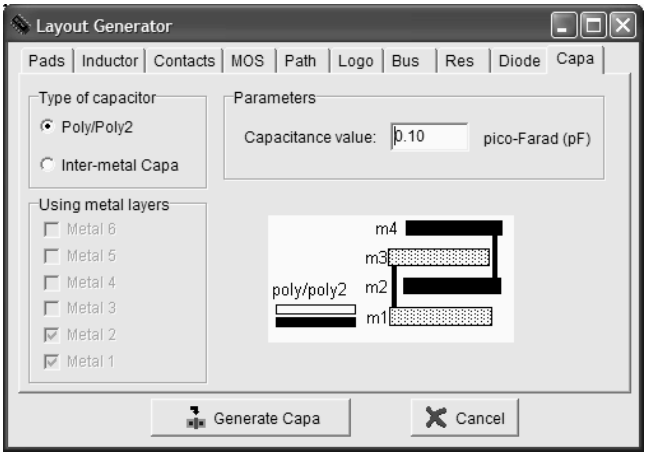


Figure 7-20: The generator menu handles the design of poly/poly2 capacitor and inter-metal capacitors

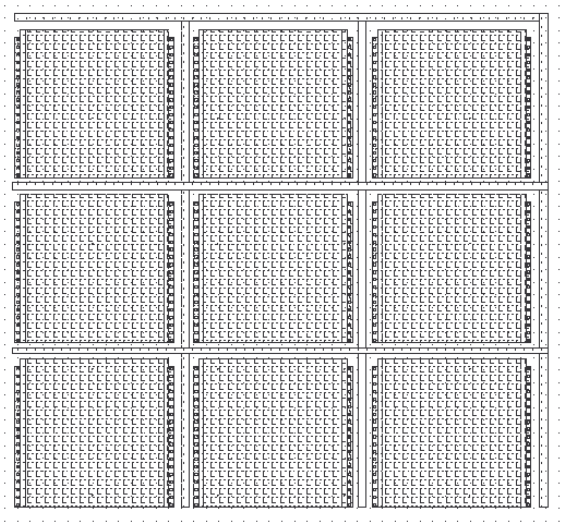


Figure 7-21. Implementation of an array of 100fF capacitor for the 3-bit DAC (DacCapacitor.MSK)

3. Sample and Hold circuits

Sample and Hold (S/H) circuits are critical in converting analog signals into digital signals. The sample-and-hold main function is to capture the signal value at a given instant and hold it until the ADC has processed the information (Figure 7-22).

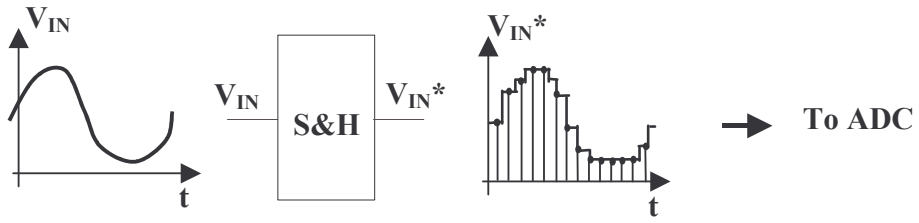


Figure 7-22. The sample-and-hold circuit

The operation is repeated in time with a regular sampling period. We can notice that during the sampling period, the S/H circuit operates alternatively in dynamic mode (sample) and in static mode (hold), as shown figure 7-23. In dynamic mode, the switch lets the input signal flow through the pass transistor and settle *Vin** within the required accuracy. Several parasitic effects may be observed : when the switch is turned off, a parasitic offset may appear due to capacitance couplings which modifies the voltage *Vin**. Also, after some nanoseconds, the stored voltage may be altered by parasitic discharge, appearing as an unpredictable droop.

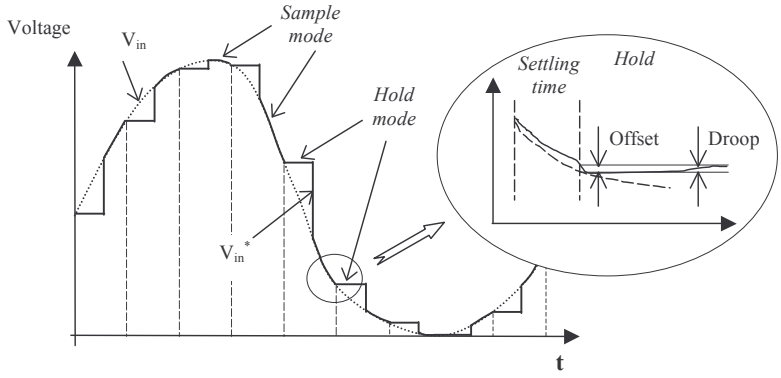


Figure 7-23. Sampling of an analog voltage (sample and hold modes)

The transmission gate can be used as a sample and hold circuit. The schematic diagram of the sample/hold circuit is proposed in figure 7-24. It corresponds to the classical transmission gate. The only important supplement is the storage capacitor, called *Cstore*, appearing at the output *Vin**, the sampled version of *Vin*. The capacitor retains the voltage information during the conversion phase. By default, a parasitic capacitance always exists due to diffusion areas of the p-channel MOS and n-channel MOS devices. However, *Cstore* includes a supplementary capacitor connected to the node *Vin**, with a capacitance value sufficiently high to counterbalance the effects of leakage currents.

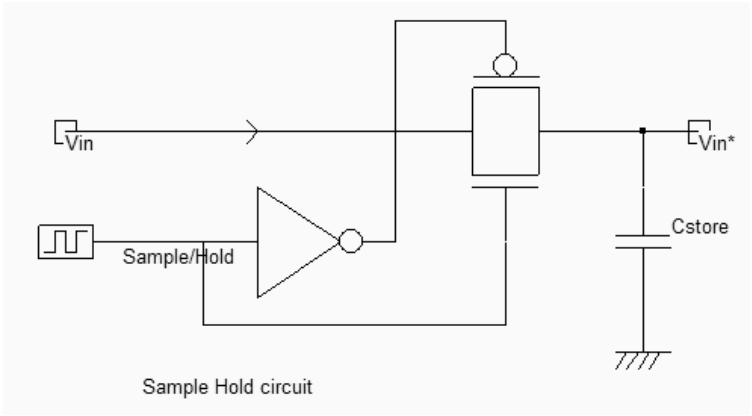


Figure 7-24. Schematic diagram of the Sample-Hold circuit (SampleHold.SCH)

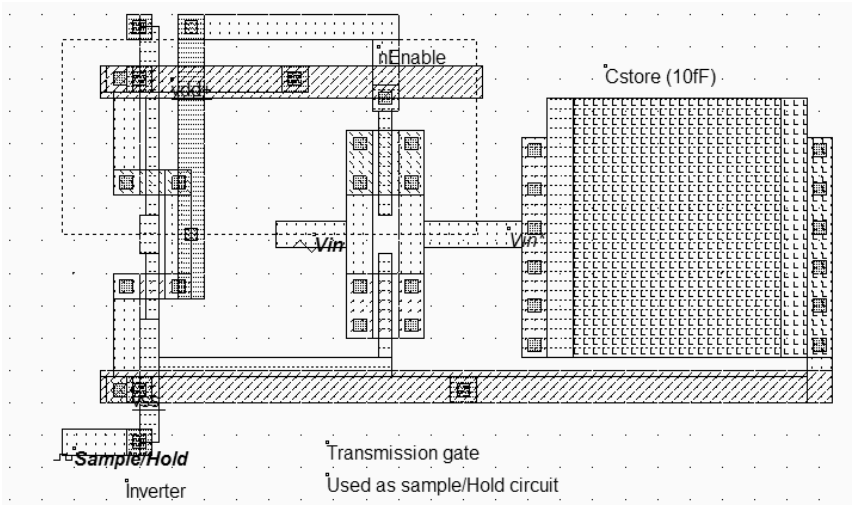


Figure 7-25. The transmission gate used to sample analog signals (SampleHold.MSK)

The layout of the transmission gate is is reported figure 7-25. The *sample/hold* command is situated on the left, and controls the transmission gate. The inverter is required for the pMOS device. The *Vin** signal is connected to a 10fF capacitor made of poly/Poly2. The effect of sample and hold is illustrated in figure 7-26. The voltage curves have been superimposed by using the simulation mode **Current and Voltage vs. Time**. When sampling, the transmission gate is turned on so that the sampled data *Vin** reaches the value of the sinusoidal wave *Vin*.

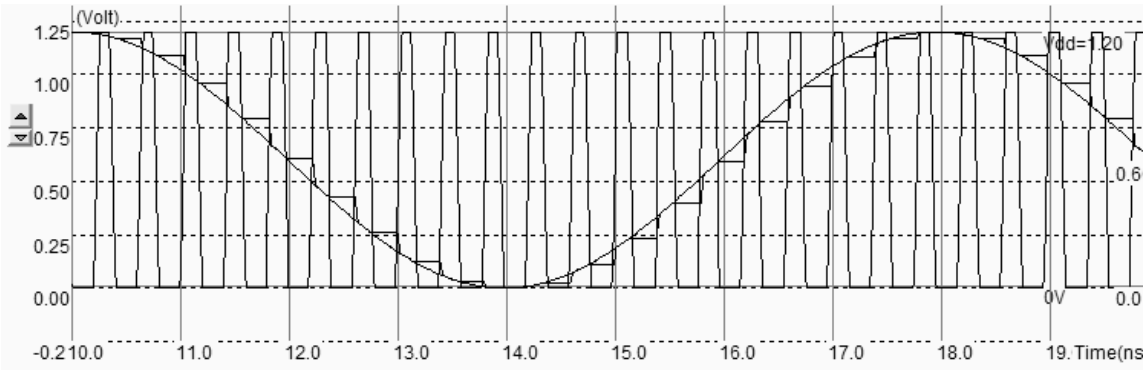


Figure 7-26. Effect of sampling (SampleHold.MSK)

When the gate is off, the value of the sampled data *Vin** remains constant. This is mainly due to the parasitic capacitance of the node which has a value of 10fF as extracted in a CMOS 0.12µm process (figure 7-27). This is sufficient to retain the information for several nanoseconds, even in the presence of leakage currents. Higher values of storage capacitor are required if the duration of the analog to digital conversion is of the order of the µ-second, with a large voltage precision. In all cases, the sampled voltage must not fluctuate more that by 30% of the least significant bit over the whole sampling cycle.

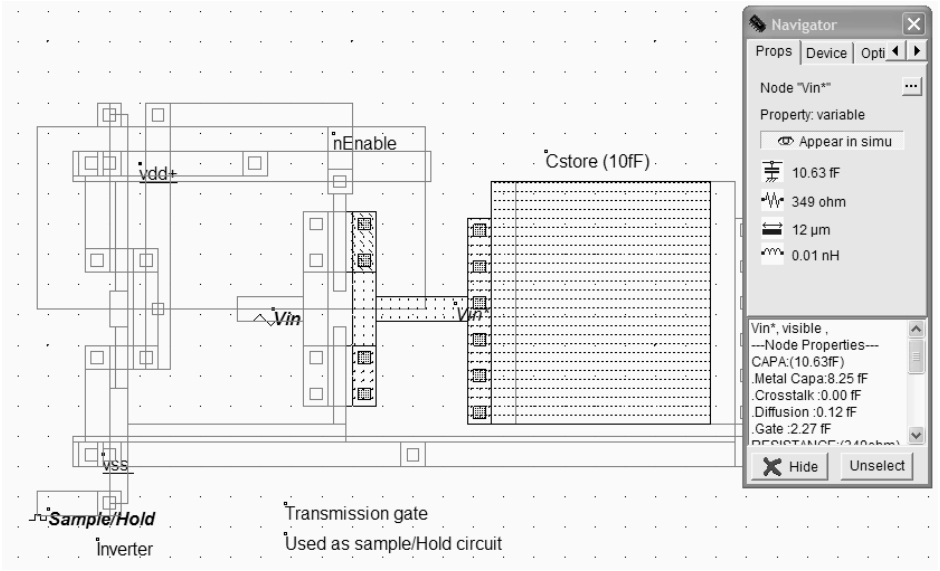


*Figure 7-27. The hold effect is related to the parasitic capacitance of the node Vin**

Layout considerations

If the width and length of the nMOS and pMOS devices are not identical, an error appears between the sampled *Vin** and the original voltage *Vin.* This voltage difference is a parasitic offset, which depends on the value of *Vin* in a non-linear way, as shown in the measured transfer characteristics shown in figure 7-28 (Standard sample circuit). The offset is minimized if the nMOS and pMOS sizes are identical, as it is strongly dependent on the parasitic capacitance between the gate and the drain. With identical channel size, C_{GD} of the pMOS is equal to C_{GD} of the nMOS. However, as the size of the nMOS is identical to the pMOS, the pMOS switching is slower.

The sampling circuit can be improved by switching the pMOS *before* the nMOS device, in contrast to the circuit proposed in figure 7-27, and by adding so-called "dummy transistors" on the storage node. The curves shown in figure 7-27 are complied from measured characteristics on a 0.18µm test chip [Bendhia], and exhibit a significant reduction of the offset. We often experienced a negative offset (10-20mV) for *Vin<Vdd/2*, and a positive offset for *Vin>Vdd/2*. Unfortunately, Microwind do not modelize these characteristics accurately due to a simplification of coupling capacitance in MOS device models.

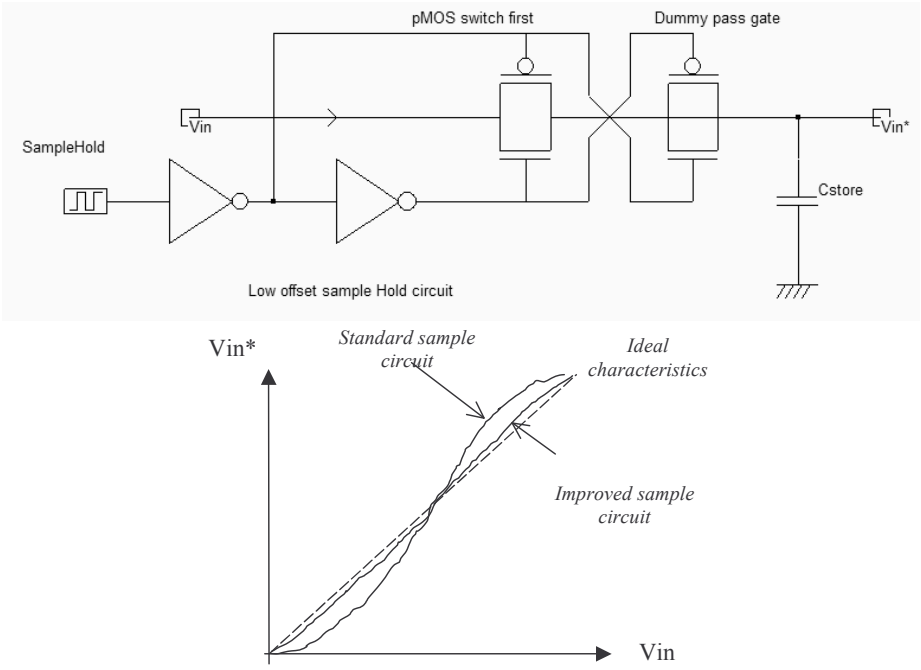


Figure 7-28. Offset reduction techniques (SampleHold.SCH)

Shannon's Sampling Theorem

The critical element when capturing the analog input voltage accurately is the number of sampled data in the considered time window. We can also talk of the sampling frequency compared to the input voltage frequency. Shannon's sampling theorem gives the minimum frequency required to represent the analog input voltage accurately. The minimum sampling frequency f_{sample} must be greater than twice the highest frequency component of the original signal f_{signal} (equation 7-5).

$$f_{sample} > 2.f_{signal} \qquad \text{(Equ. 7-5)}$$

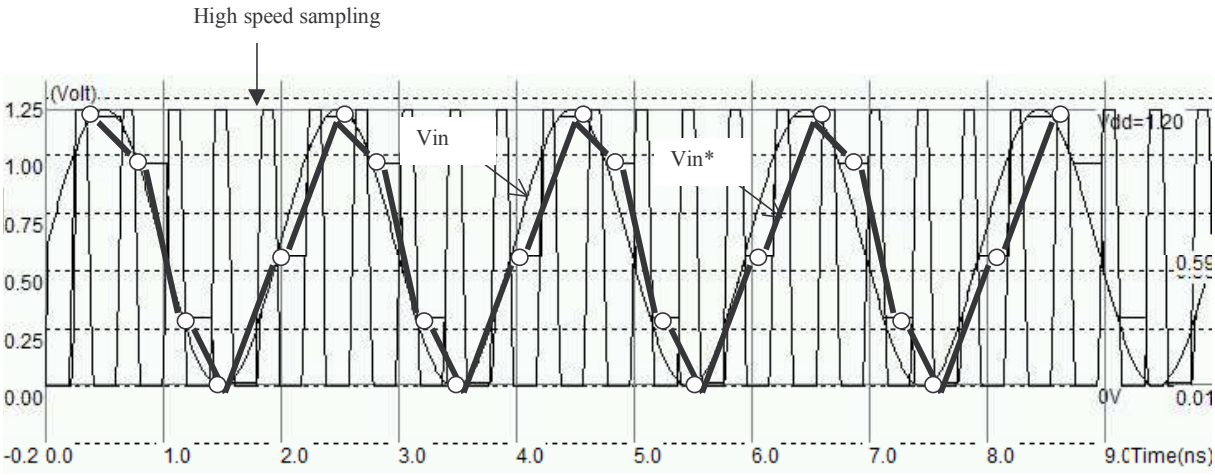


Figure 7-29: The sampling frequency is fast enough to comply with Shannon's theorem (SampleHoldShannon.MSK)

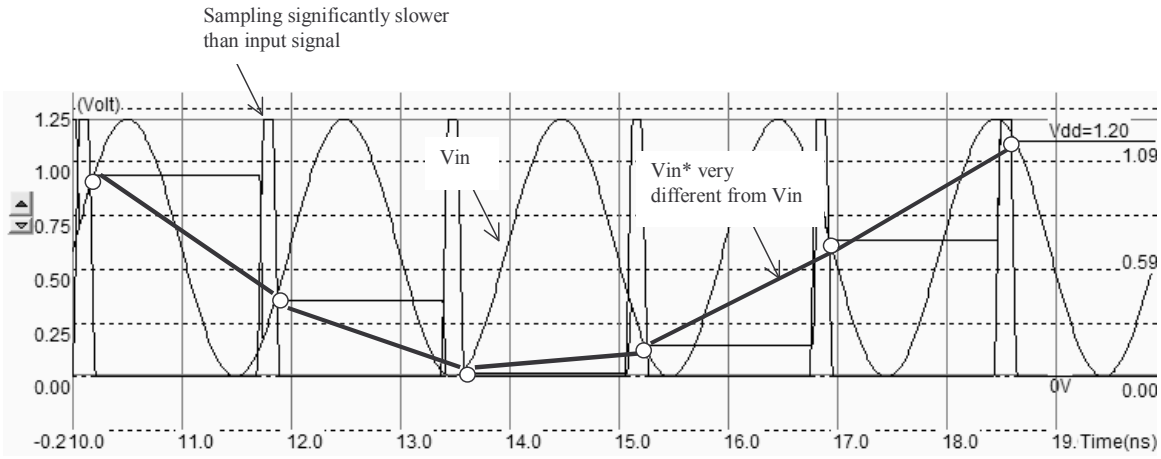


Figure 7-30. The sampling frequency is too slow. Vin differs from Vin (SampleHoldShannon.MSK)*

Figure 7-30 shows the sampling of a 500MHz sinusoidal input wave (f_{signal}) with a sampling frequency f_{sample} of 2.5GHz which complies largely with the Shannon's Theorem. In figure 7-30 the sampling frequency f_{sample} is too low (600MHz), consequently the sampled output *Vin** is significantly different from *Vin*.

4. Analog-Digital Converters architectures

The analog to digital converter is considered as an encoding device, where an analog sample is converted into a digital quantity with a number N of bits. Figure 7-31 shows the complete chain from the analog signal to the digital data using a sampled and hold module and a 4-bit ADC. ADCs can be implemented by employing a variety of architectures. In the following chapters, we describe the Flash converter and successive approach converters.

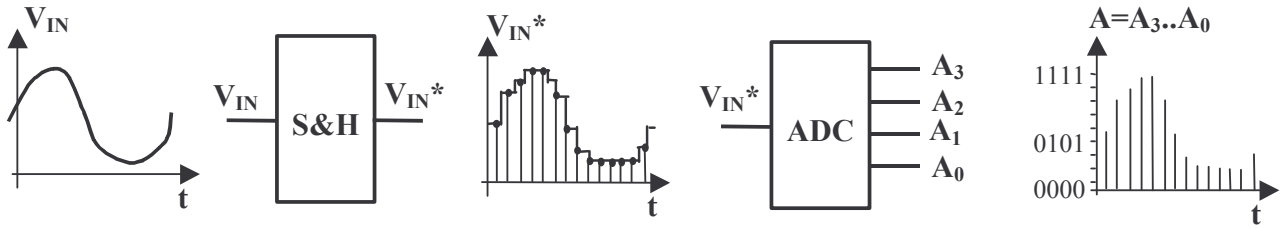


Figure 7-31. A 4 bits digital conversion of a sampled analog voltage

The Flash converter Principles

The 2-bit analog-digital converter converts an analog value *Vin* into a two-bit digital value A coded on 2-bit A_1,A_0. The flash converter uses three amplifiers, which produce results C_0,C_1 and C_2, connected to a coding logic to produce A_1 and A_0 in a very short delay (Figure 7-32). The flash converters are widely used for very high sampling rates, a the cost of very important power dissipation.

Analog Input Vin	C2	C1	C0	A1	A0
Vin<Vref0	0	0	0	0	0
Vref0<Vin<Vref1	0	0	1	0	1
Vref1<Vin<Vref2	0	1	1	1	0
Vin>Vref2	1	1	1	1	1

Table 7-3. The specifications for a 2-bit flash ADC converter

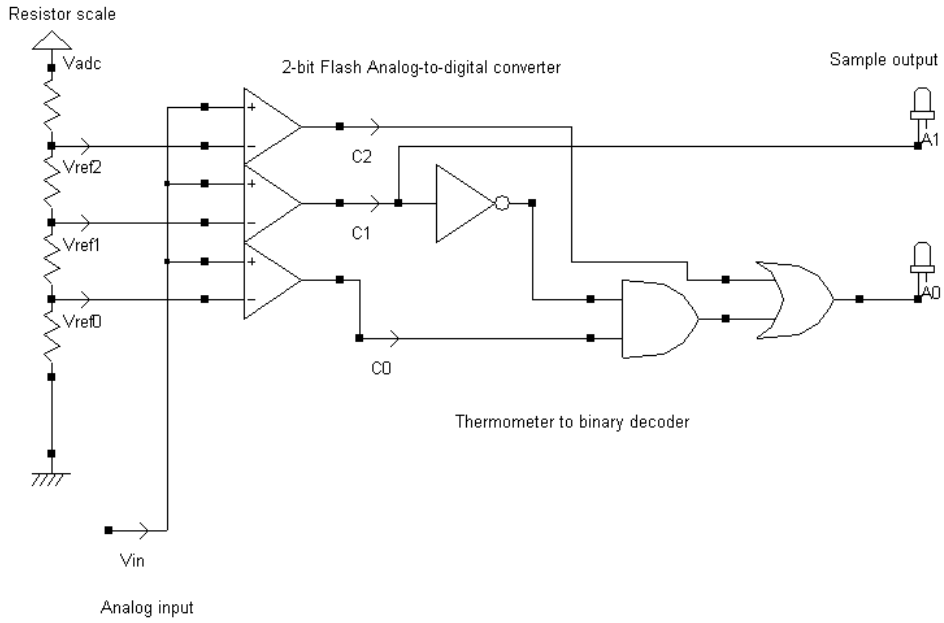


Figure 7-32. The schematic diagram of the 2-bit flash ADC converter (AdcFlash2bits.SCH)

A schematic diagram for the 2-bit flash converter is proposed in figure 7-33. The resistor scale produces reference voltages *Vref0*, *Vref1* and *Vref2*. Three comparator circuits compute the difference between *Vin* and the reference voltage. Their outputs C_2,C_1 and C_0 are almost logic signals as the comparators are connected in open-loop.

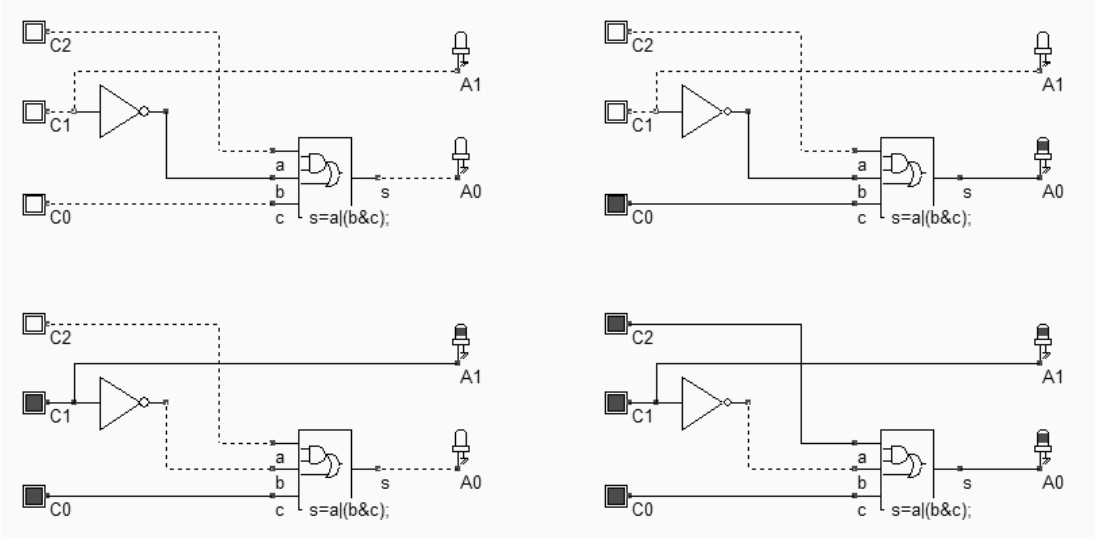


Figure 7-33. The thermometer to binary coder (AdcFlash2bits_coder.SCH)

The main problem of the comparator-based architecture is that the output A_1,A_0 is not directly available from C_2,C_1 and C_0. The comparator outputs represent the "thermometer coding" of the input. The ones propagate from C_0 to C_2 as the input *Vin* rises, as specified in table 7-3. A conversion circuit from thermometer code to binary code is needed. In the case of a 2-bit flash converter, the circuit is quite simple (figure 7-34), and can be efficiently implemented using one inverter (A_1) and a complex gate (A_0). For a 3-bit flash converter, the logic circuit starts to rise in complexity. The thermometer code is described in table 7-4.

Analog Input Vin	C6	C5	C4	C3	C2	C1	C0	A2	A1	A0
Vin<Vref0	0	0	0	0	0	0	0	0	0	0
Vref0<Vin<Vref1	0	0	0	0	0	0	1	0	0	1
Vref1<Vin<Vref2	0	0	0	0	0	1	1	0	1	0
Vref2<Vin<Vref3	0	0	0	0	1	1	1	0	1	1
Vref3<Vin<Vref4	0	0	0	1	1	1	1	1	0	0
Vref4<Vin<Vref5	0	0	1	1	1	1	1	1	0	1
Vref5<Vin<Vref6	0	1	1	1	1	1	1	1	1	0
Vin>Vref6	1	1	1	1	1	1	1	1	1	1

Table 7-4. The specifications for a 3-bit flash ADC converter

A 3 bit flash converter requires 7 converters and a complex logic circuit which converts the thermometer code into a binary code, as specified in table 7-4. The 8-bit flash converter would require 255 comparators and a very complex logic decoder. An interesting approach for the encoding consists in using a small memory array, taking into account the specific condition of the thermometer coder. For example, a 0 on C_4 and a 1 on C_3 means that *Vref3<Vin<Vref4*, which is a sufficient condition to produce *A=100 (4)* at the output. Using this principle, the implementation of the 3-bit coder is detailed in figure 7-32.

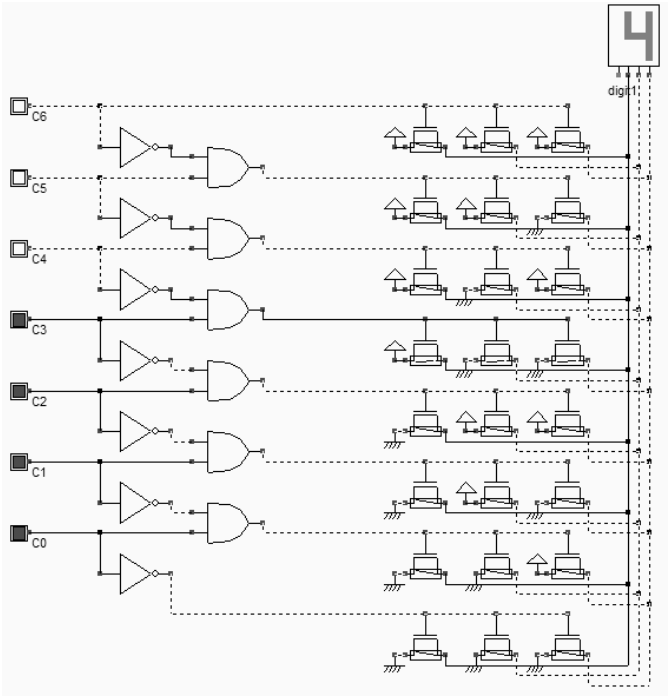


Figure 7-34. The thermometer to 3-bit binary coder using a logic array (AdcFlash3bits_coder.SCH)

Flash converter Implementation

The resistor ladder generates intermediate voltage references used by the voltage comparators located in the middle of the layout (figure 7-35). An unsalicide option layer multiplies the sheet resistance of the polysilicon ladder for an area-efficient implementation. The resistance symbol *R(poly)* is inserted in the layout to indicate to the simulator that an equivalent resistance must be taken into account for the analog simulation. Open-loop amplifiers are used as voltage comparators. The comparators address the decoding logic situated to the right and that provides correct A_0 and A_1 coding.

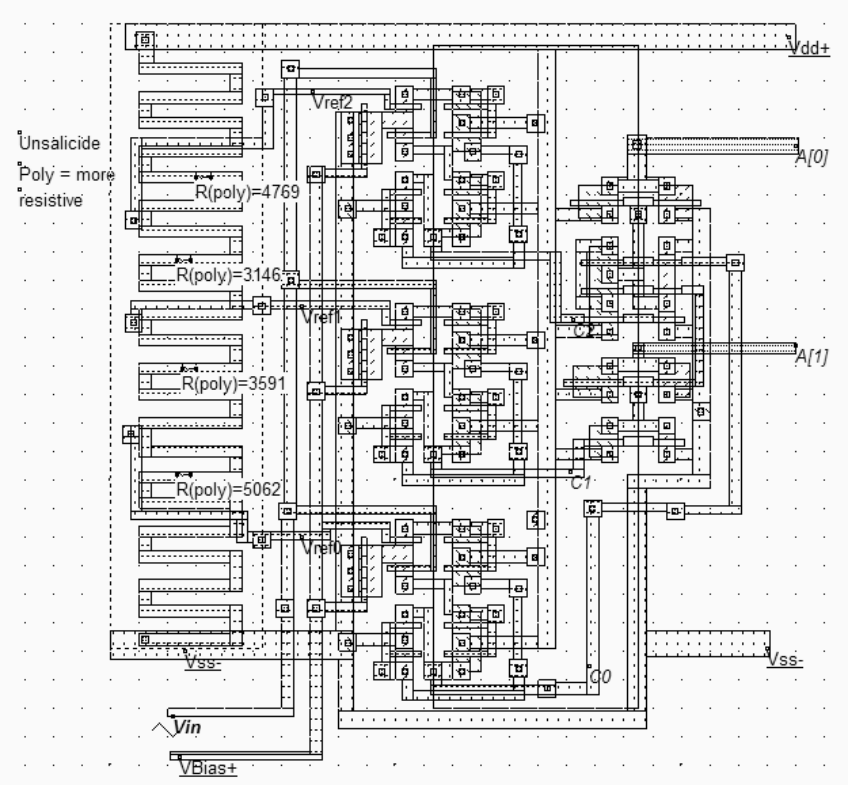


Figure 7-35. Design of the analog-digital converter (ADC.MSK).

In the simulation shown in Figure 7-36, the comparators C_0 and C_1 work well but the comparator C_0 is used in the lower limit of the voltage input range. The generation of combinations "01", "10" and "11" is produced rapidly but the generation of "00" is slow. The comparator C_0 may be modified to provide a faster response in comparison with low voltage, by changing the biasing conditions. An alternative is to reduced the input voltage range, which means that the resistance scale would be supplied by *Vdac-* larger than VSS and *Vdac+* smaller than VDD.

The main drawback of flash converters is the silicon area and the power consumption: every bit increase in resolution almost doubles the size of the ADC circuit and significantly increases the power consumption.

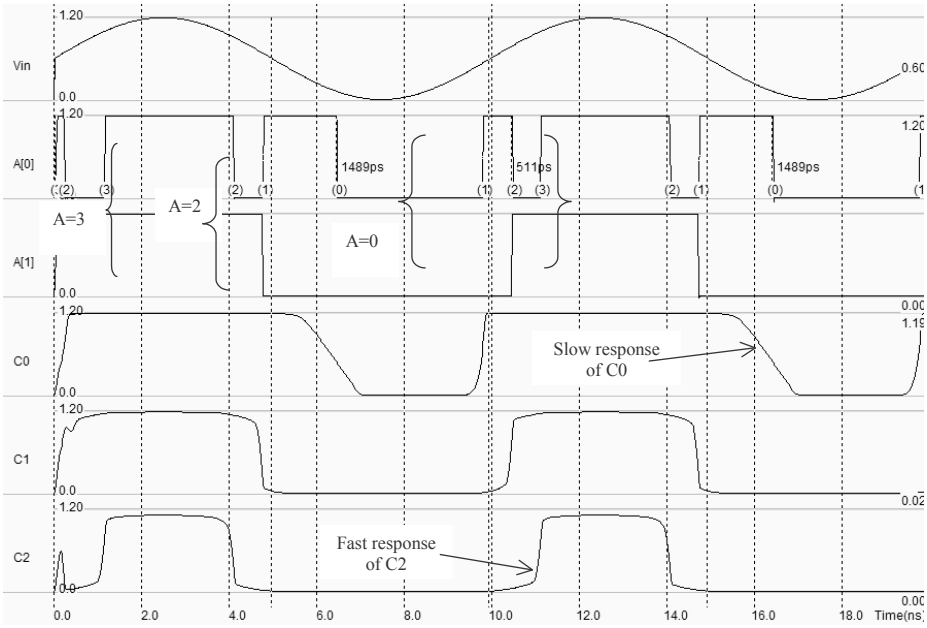


Figure 7-36. Simulation of the analog-digital converter (ADC.MSK).

Low speed ADC Converters

The most common low speed converter is the iterative converter. As shown in figure 7-37, it consists of a digital-to-analog converter, a counter and an analog comparator. Starting with the lowest voltage, the counter is increased until the DAC voltage *Vdac* is higher than the input voltage *Vin*.

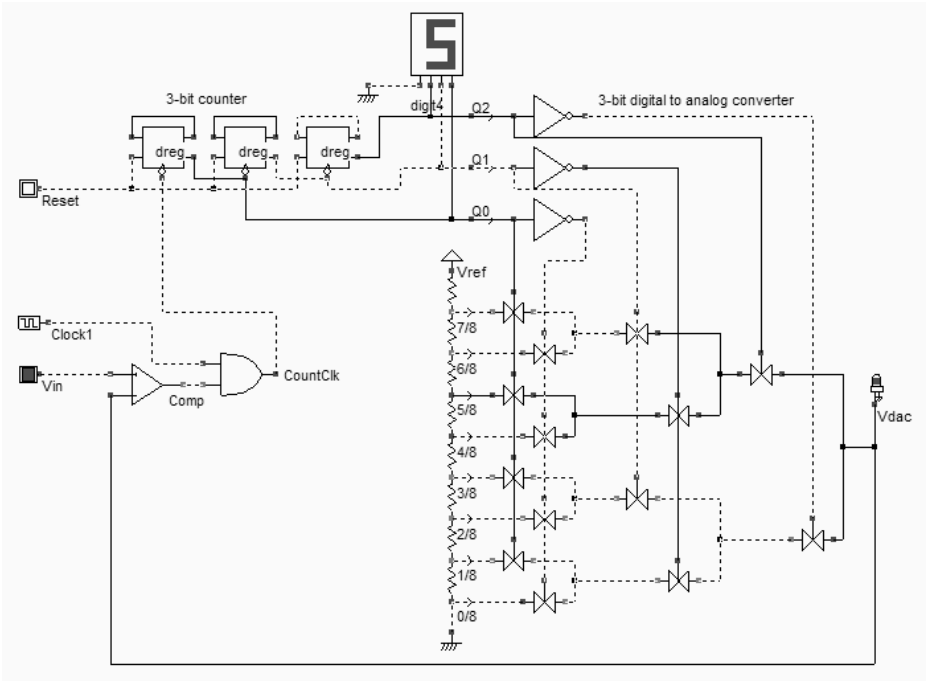


Figure 7-37. Iterative converter using a DAC (ADCInterative.SCH)

In the particular example shown in the figure, we suppose that *Vin* is a little higher that *Vref/2*. The counter has reached the value 5 (101), which corresponds to the transfer of the reference voltage *Vrefx5/8* to *Vdac*. As *Vin* is lower that this reference, the comparator produces a 0, which stops the counter clock *CountClk*.

This converter is very simple to design but slow. Up to 2^N clock cycles are necessary to complete the conversion, where N is the resolution of the DAC and of the ADC converter. For example, with a 16-bit data converter with a 100MHz clock frequency, the conversion rate is as low as 750 Hz. The implementation of figure 7-37 corresponds to a 3-bit converter. With a high resolution DAC and a high precision amplifier, high resolution ADC converters may be constructed.

A better solution consists in examining the most significant bit a_{n-1} first and then in determining whether *Vin* is larger or smaller than VDD/2. The comparator gives the value of that bit directly. Then, the comparison is performed for the next bit, and so on until all bits are extracted, finishing by the least significant bit a_0. This type of converter is called successive approximation converter. The complete process is faster than the iterative converter as only N comparisons are necessary. The algorithm is detailed in figure 7-38.

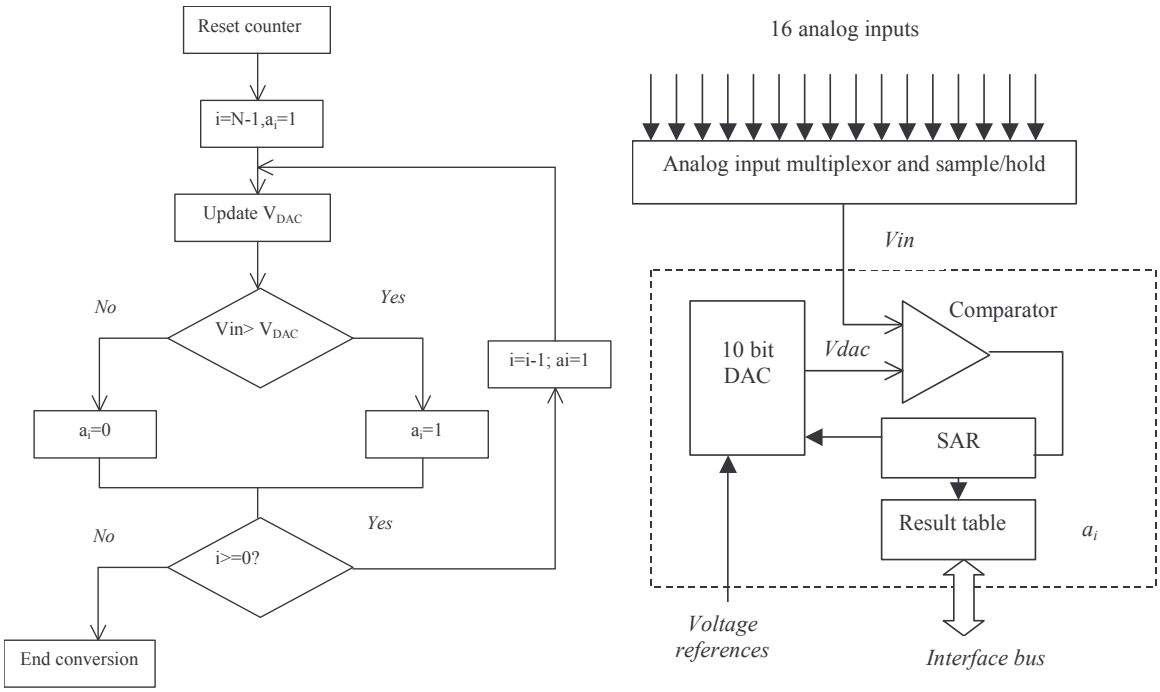


Figure 7-38. Iterative converter algorithm and typical implementation for a 10bit converter (Motorola MPC555)

The schematic diagram of a successive approach converter is given at the right of Figure 7-38. The analog input signal *Vin* is retained by a sample/hold circuit during the data conversion process. In the first clock cycle, the most significant bit a_i of the successive approximation register (SAR) is set to 1. The DAC converts the SAR value to an analog voltage *Vdac* that is compared to *Vin*. If *Vdac* is smaller than *Vin*, the bit a_i is validated at 1, and the SAR register is unchanged. Conversly, if *Vdac* > *Vin*, a_i is set to 0. The DAC generation and comparisons processes are repeated for N clock cycles to complete the conversion.

In many industrial 32-bit micro-controller, 8-bit to 16-bit successive approach converters are implemented with 4 to 16 channel multiplexors. In the particular case of a 10-bit converter supplied by a voltage reference *Vref*, as much as 2^{10} reference values (1024) are used. An example of conversion is given in figure 7-39. For the first bit, *Vdac* is set to *Vref/2*. As *Vdac* > *Vin*, a_9 is set to 0. Then *Vdac* is set to *Vref/4*. As *Vdac* < *Vin*, a_8 is set to 1.

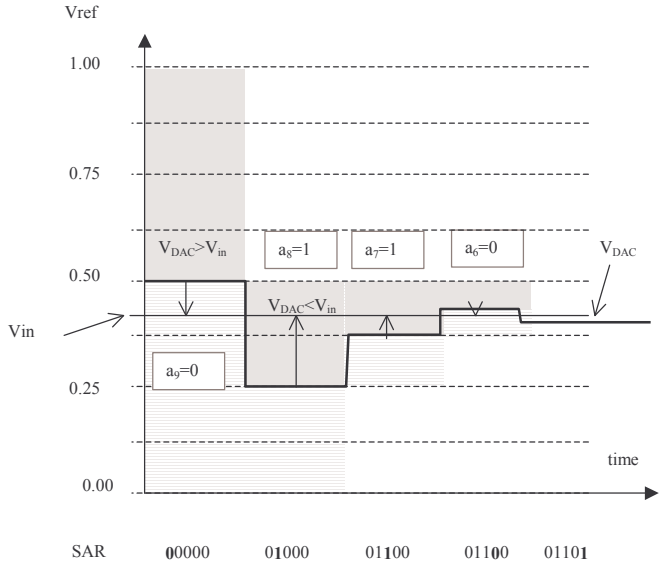


Figure 7-39 The successive approach converter at work

Pipeline ADC Converters

The pipelined Analog-to-digital converter consists of two or more stages connected in serial, each containing a low resolution ADC and DAC converter. The schematic diagram of figure 7-40 corresponds to a 4-bit ADC, based on a 2-bit flash ADC. The role of the first stage is to generate the most significant bits *A3* and *A2*. Then the difference between *Vdac* and *Vin* is computed. In order to convert the residue, the voltage difference is amplified and sent to a second 2-bit DAC which calculates the least significant bits *A1* and *A0*. The pipelining approach is very powerful and may be applied for a large number of stages, which permits to convert analog signals with a high resolution without the need of designing high resolutions DAC or ADC blocks. An extensive study of pipeline ADC may be found in [3].

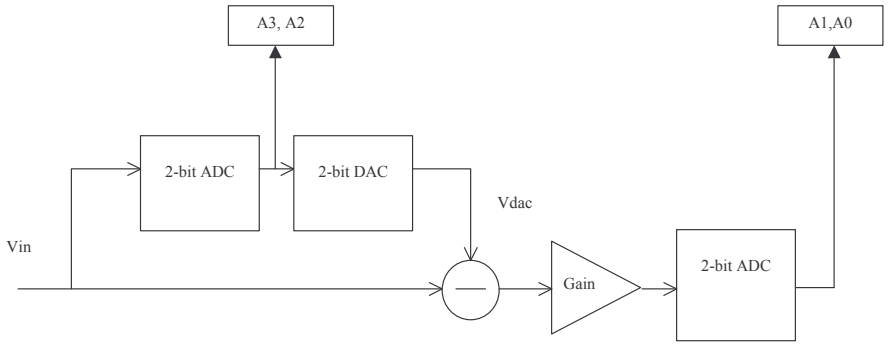


Figure 7-40 Pipeline ADC converter

The successive approach or pipeline converter layout area increase almost linearly with the converter resolution N, as well as withthe power consumption, which represents a key advantage on flash converters.

5. Temperature Sensor

One of the simplest temperature sensing element is the *pn* diode [4]. The classical model of the diode is given by equation 7-7. The expression of the current includes two strongly temperature-dependent parameters, the exponential term and the reverse saturation current.

$$I_{ak} = I_{sat} S(exp[\frac{q}{kT} V_{ak} \ ] - 1) \qquad \text{(Eq. 7-7)}$$

with
Isat= reverse saturation current per µm2 (A/µm2)
S=surface of the diode (µm2)
Q=electric charge
K=Boltzmann's constant
T=absolute temperature (°K)
Vak= diode voltage (V)

For temperature sensing, the *np* diode is forward biased by a small constant current, and the diode voltage *Vak* serves as a measure of the temperature. The proposed circuit is given in figure 7-41. The pMOS device serves as a load while the P+/Nwell diode serves as a temperature sensor. The pMOS device itself is sensitive to temperature, but its dependence is negligible as compared to the diode. In this circuit, *Vak* is equal to *Vref*.

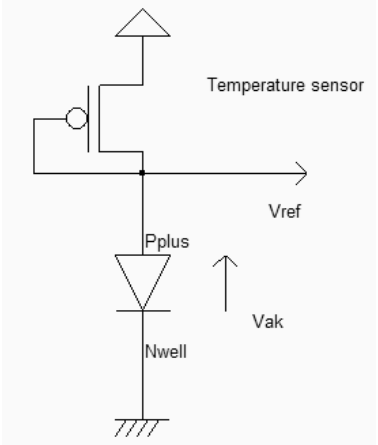


Figure 7-41. Principles for a temperature sensor based on a junction diode (SensorTemperature.SCH)

The implementation of the diode in forward biasing condition cannot be done with a *N+/Psusbtrate* diode, as the substrate is connected to ground, which would imply a negative biasing of the N+ diffusion. The only remaining solution consists in using an *nwell* region connected to ground, and a P+ diffusion, which creates a *P+/Nwell* diode.

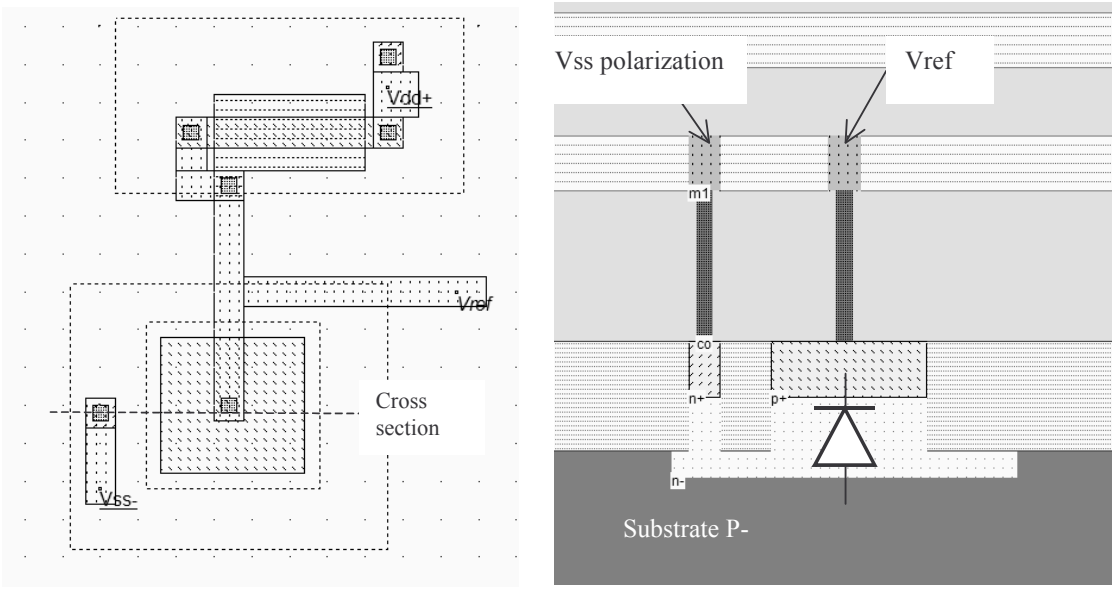


Figure 7-42. Implementing the temperature sensor (SensorTemperature.MSK)

The implementation of the current sensor is reported in figure 7-42. The pMOS channel length is large so as to reduce the DC current and to avoid short channel limiting effects. The simulation of the temperature influence is performed using the parametric analysis, in order to plot the diode voltage *Vref* versus the temperature in °C directly. Invoke the command **Analysis→Parametric Analysis**, click inside the layout corresponding to Vref, and the following screen appears. Select the item "**Temp.",** and the measurement "**Final voltage Vref**". Click **Start Analysis** to perform the iterative simulation from -40°C to 120°C with a step of 20°C. At the end of each simulation, the final value of *Vref* is added to the data array. It can be seen from the result of figure 7-43 that *Vref* decreases nearly linearly with temperature, with a slope of around -1.2mV/°C.

Measured results presented in [4] give around -2.4mV, using a stable current reference of 10µA instead of the diode-connected on-chip pMOS device. The main problem of this type of sensor is its very strong dependence on process variation which requires a calibration procedure to obtain an exact value for the temperature.

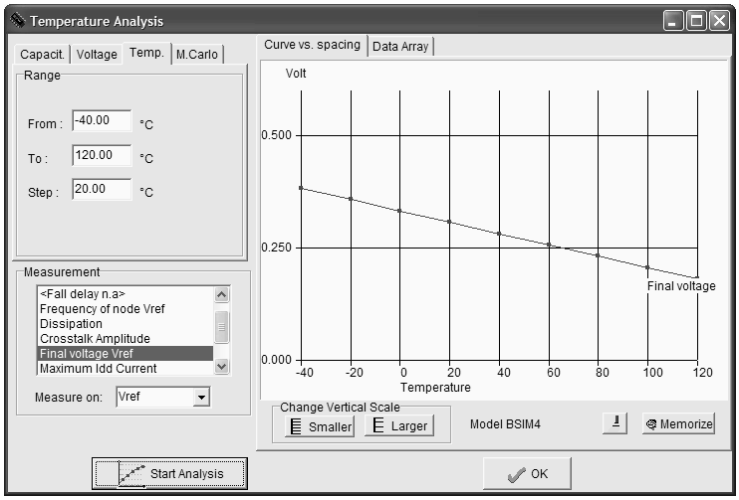


Figure 7-43. Simulation of the voltage dependence with temperature (SensorTemperature.MSK)

6. Image Sensors

Recently, new attention has been paid for CMOS image sensors, due to the proliferation of low cost video cameras such as webcams and video on mobile phones. Most high quality cameras use charge-coupled-device image sensors, which feature superior light sensing characteristics over CMOS sensors. However, the use of standard CMOS technology for image sensing offers two key advantages over CCD technology: the analog signal processing can be realized on the same chip, and the price of CMOS imagers is significantly lower that its CCD counterpart, thanks to the availability of deep sub-micron CMOS foundries all over the world.

The Diode Detector

In CMOS technology, one of the most simple light detection devices is the PN junction. The diode is photo resistive, which means that its characteristics I/V are sensitive to light. The photo diode can be considered as a variable resistor. The most common photo diode consists of an N+ diffusion area in the P-substrate. By default, with no light, the diode is polarized in reverse mode, so that almost no currents flow from the grounded P-substrate to the N+ diffusion region (Figure 7-43). The light photons are converted in the P region into electrons which are attracted by the N+ diffusion and generate a photo current. This current is almost linearly proportional to the incident light. Consequently, the resistance of the diode is linearly decreased with the intensity of the light.

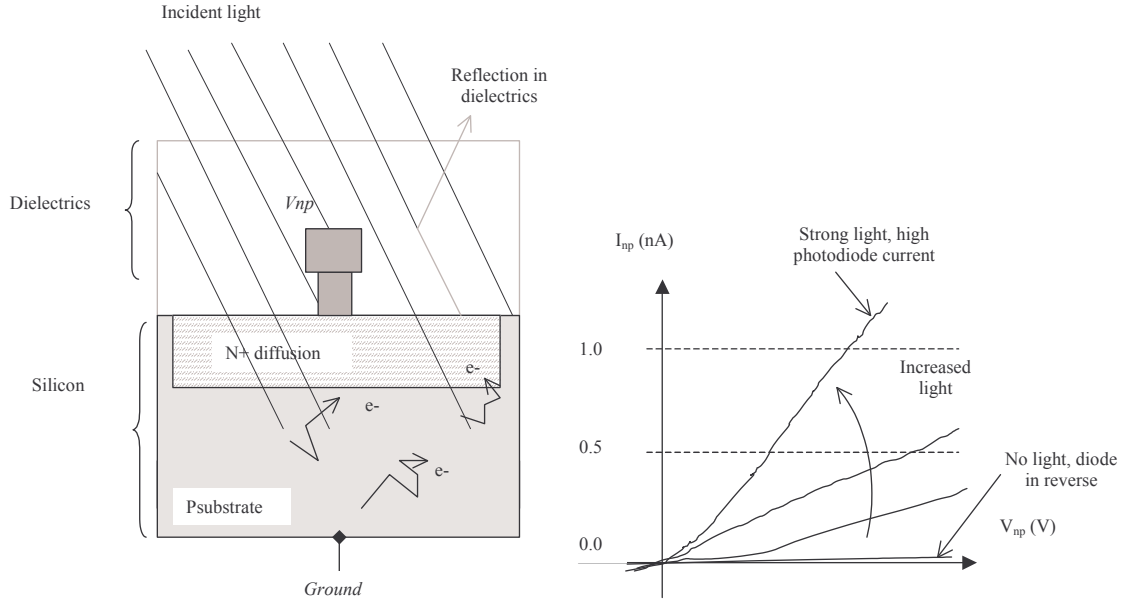


Figure 7-43. The diode as a light detector

Silicon photo resistors are sensitive to the spectral band corresponding to a wavelength from 350 to 1100 nm, which covers the visible spectrum (Figure 7-44). Our eyes are sensitive to a light that corresponds to a wavelength range of 400nm (blue color) to 700nm (red color).

The efficiency in converting photons into electrons is limited by reflection on the dielectric materials which cover the surface of the integrated circuit and by the loss and recombination of electrons during their travel from the substrate to the electrode. The efficiency of the light sensor is maximum near 700nm, with an efficiency around 20%. The current that flows on the photodiode is very small, approximately 1nA for a 10x10µm diode area.

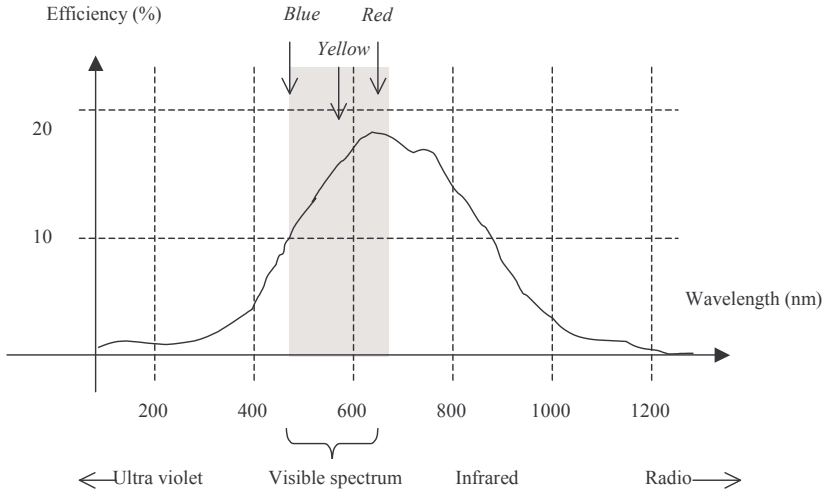


Figure 7-44. Typical performance of a photo diode versus the incident wavelength

Notice that the photoreceptor is subject to an important time delay to reach equilibrium after a change in illumination level. This time constant is of the order of the micro-second.

Diode Detector Setup

The basic setup for the passive light sensor is shown in figure 7-45. The circuit consists of the diode and a capacitor. First, the capacitor C_{store} is charged to a high voltage *Vdd*, using the precharge switch. Second, the precharge is deactivated and the light sensor starts to discharge the capacitor, thanks to the reverse photo-current *Inp*. Without light, the current *Inp* is equivalent to the reverse mode current, which is the range of the pico-ampere (10^{-12} A) for a small PN junction. With a strong light, the current *Inp* enters the range of the nano-ampere (10^{-9} A).

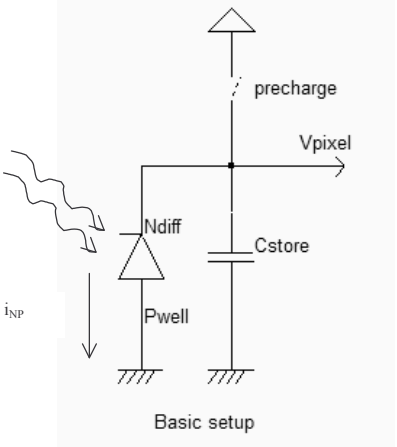


Figure 7-45 Light detector setup including a precharge circuit (ImageSensor.SCH)

The basic diagram for an array of passive light sensors is shown in figure 7-46. Each passive pixel sensor (PPS) converts the photons into an electrical charge which is carried off by the sensor through pass transistors which are laid out as in a memory array. The charges flow through the vertical lines to a voltage amplifier with an important gain. The main problem with the passive sensor is the noise that appears in the resulting image and limits its use to low quality image sensing. Furthermore, a large pixel matrix induces important leakage in the vertical bit lines and limits the use of the passive pixel sensor to very low resolution devices.

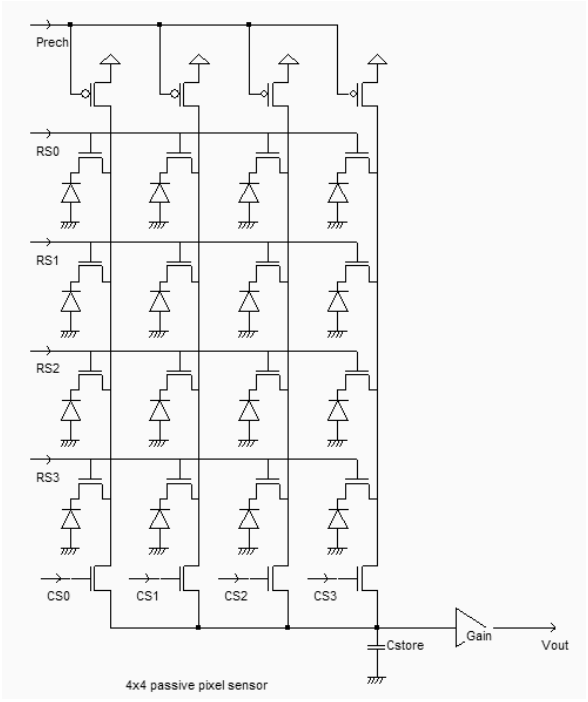


Figure 7-46. A passive pixel array made with the photodiode and pass transistors (ImageSensor.SCH)

The goal of active pixel sensors (APS) is to reduce the noise associated with passive sensors, and to amplify the light-induced charges at each pixel location. The active pixel sensor approach improves the light sensor performances significantly, allowing the design of large pixel arrays, with fast read out access, while keeping power dissipation low. The transistor *N1* sets the photodiode voltage *Vdiode* to a high value when *Set=1*. Once *N1* is cut off, the photodiode discharges *Vdiode*, depending on the light intensity. The source follower *N2* buffers the photodiode voltage to the bus, when the row select transistor *N3* is active (*Sel=1*), as illustrated in figure 7-47.

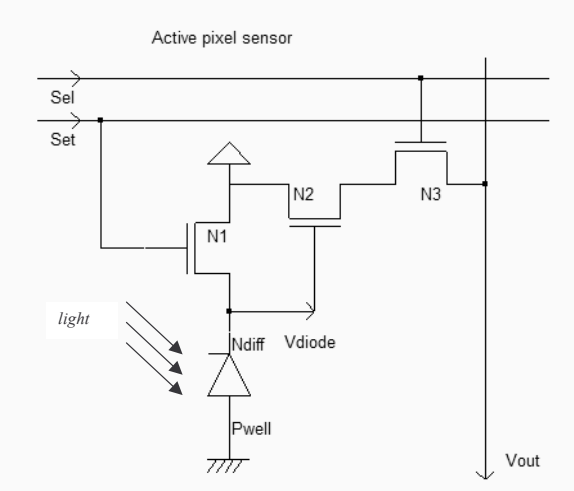


Figure 7-47. The active pixel sensor with a photodiode (ImageSensorActive.SCH)

In MICROWIND, the photodiode effect is modeled by a virtual resistor added between *Vdiode* and the ground. Values ranging from 1MΩ to 9MΩ are added in the 2x2 pixel array to account for variable photocurrent (figure 7-48). However, the real case photocurrent is very small (Around 10-100 pA/µm^2 in the best case). When a 10x10µm pixel is designed, the maximum current is around 10nA, which discharges the pixel capacitance within several microseconds, with an equivalent leakage resistance of hundreds of mega-ohm. Using a mega-ohm resistor speeds up the simulation but accelerates real case chronograms by 3 orders of magnitude.

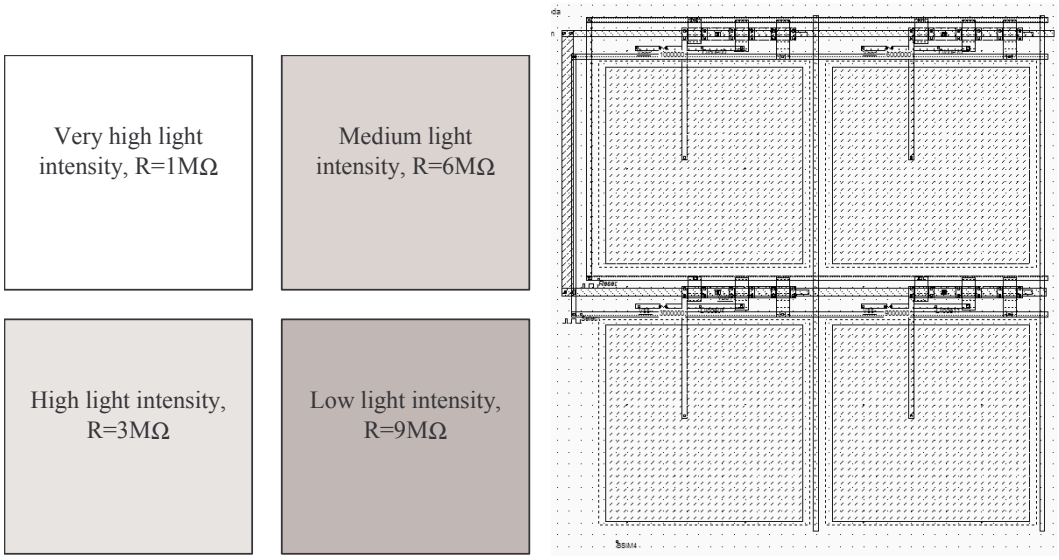


Figure 7-48. A matrix of 4 active pixels(ImageSensor2x2.MSK)

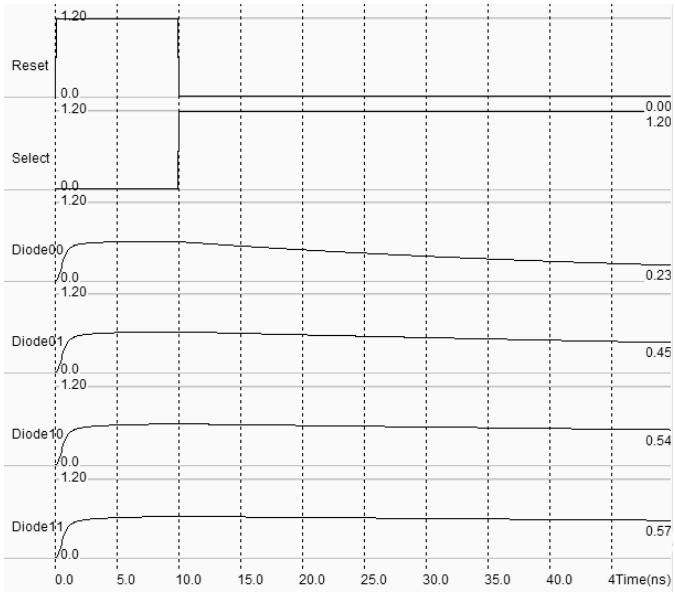


Figure 7-49. Simulation of the active pixels (ImageSensor2x2.MSK)

In the simulation reported in figure 7-49, the role of photocurrents is clearly demonstrated: the diode voltage of each pixel is set to around VDD-Vt, then each photocurrent discharges the pixel voltage with a slope dependent on the light intensity. The final voltage may be sampled at time 50ns. In real case light sensors, 100µs up to 1ms are required before sampling the pixel voltage and converting it into an image information.

Color filter materials are used to capture selectively the blue, green and red components of the incident image. The transmission of light has a general shape shown in figure 7-50. The blue filter passes electromagnetic waves around a 475nm wavelength, the green filter passes mainly 510nm waves, and the red filter passes 650nm waves. Color filters may be placed mechanically on the top of the pixel array, in order to assign one color to one pixel, according to a regular assignation pattern such as the one shown in figure 7-50.

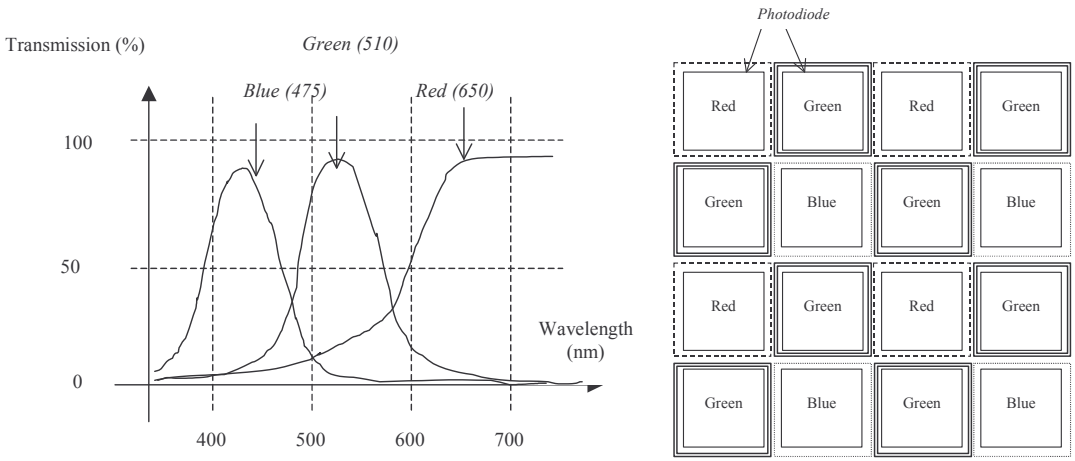


Figure 7-50. Filters used to assign one color to one pixel

7. Conclusion

In this chapter we have introduced the basis of digital to analog signal conversion. The implementation of resistor ladders has been detailed, with an illustration of non-linearity effects. The analog-to-digital convert principles have also been described, through the example of the flash converter and successive approach converter. Finally, temperature and light sensors have been briefly introduced.

Exercises

Exercise 6-1

Design of a 3-bit thermometer to binary coder according to the schematic diagram shown in figure 12-xxx

Exercise 6-2

Design a 3-bit flash converter using the thermometer coder and a set of 7 comparators.

Exercise 6-3

Design a iterative converter using a 4-bit counter and the 2-2R 4-bit DAC.

Exercise 6-4

Design a successive approach converter using a specific register and the R-2R 4-bit DAC.

References

[Backer] R.J Backer, H. W.Li, D. E. Boyce "CMOS design, layout and simulation", IEEE Press, 1998, www.ieee.org

[Razavi] B. Razavi "Design of Analog CMOS integrated circuits", McGraw Hill, ISBN 0-07-238032-2, 2001, www.mhhe.com

[Gustavsson] Mikael Gustavsson, J.Jacob Wikner, Nianxiong Nick Tan "CMOS Data Converters for Communications", Kluwer Academic Publishers, ISBN 0-7923-7780-X, 2000

[Ristic] Ljubisa Ristic "Sensor Technology and Devices", Artech House, ISBN 0-89006-532-2, 1994

[Bendhia] Paper 0.18µm sampling