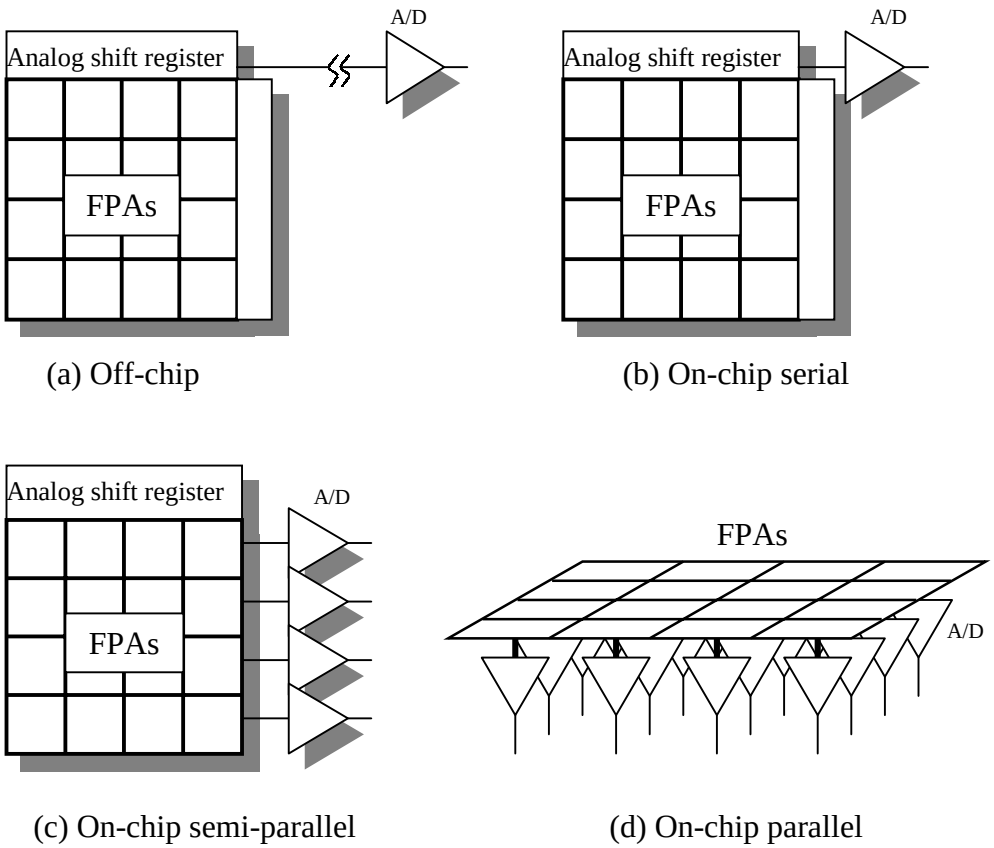
CHAPTER 1

INTRODUCTION

High speed imaging applications such as combustion imaging [1],[2], transMach fluid flow imaging [3], and aerooptic imaging [4] require a high frame rate image acquisition system with frame rates in excess of 100 kfps (frames per second). Currently, imaging systems implemented using charge-coupled device (CCD) technology with off-chip analog-to-digital data converters are limited to continuous frame rates of approximately 1 kfps [5]. In focal-plane-array (FPA) technology, the number of pixels has already been increased to thousands by thousands, and the resolution of the arrays also has been increased [6]. The combination of high resolution with the large number of pixels has resulted in data rates that can not be transmitted off the FPA through a one port readout system. For example, a second-order sigma-delta analog-to-digital converter (ADC) for serial readout system converting 1000x1000 14 bit images at a frame rate of 1MHz must be clocked at more than 62 THz.

The conventional off-chip readout system, which uses X-Y multiplexing to readout the analog array data and transports the analog data to an off-chip analog-to-

digital (A/D) converter as shown in Figure 1.1(a), is not optimal when considering speed, cost, noise-pickup, reliability, noise bandwidth, system complexity, and weight. On-chip A/D conversion has the possibility of becoming a superior choice, since a monolithic solution improves all of the above mentioned concerns except noise bandwidth. The noise bandwidth is dependent on the on-chip readout architecture. There are three basic on-chip readout architectures [7],[8]. The first is serial A/D conversion with only one on-chip ADC for the whole focal-plane-arrays as shown in Figure 1.1(b).



(a) Off-chip (b) On-chip serial

(c) On-chip semi-parallel (d) On-chip parallel

Figure 1.1 Four different readout systems

Figure 1.1(c) shows a semi-parallel A/D conversion with an ADC per each column. The third option is a parallel A/D conversion with an ADC dedicated to each pixel as shown in Figure 1.1(d). When this parallel readout architecture is combined with 3-D interconnection such as through-wafer optical communication technique, a scalable fully parallel readout system can be realized.

Among these architectures, pixelwise A/D conversion for the parallel readout system is the best choice because it performs the A/D conversion as early as possible in a signal chain, and avoids processing and transportation of analog signals. The advantages are that no signal degradation occurs when digital data are read out from the detector arrays and that the radiation on the detector can be sensed and integrated electrically during the whole frame period. The digital readout and the low possible bandwidth requirement of the ADC indicate that this type of structure can give the highest signal-to-noise ratio (SNR). Therefore, in this dissertation, we will investigate a scalable high speed readout system for the FPA system using a fully parallel readout system.

There are other advantages to the FPA when it is designed with an on-chip ADC with a fully parallel readout system [9]. When introducing intelligence into or close to the focal plane, fundamental changes in a conventional FPA system may be expected. Windowing or active vision techniques allow random or multi-resolution access to pixels. Smart cameras will be practical when accompanied by an on-chip A/D conversion and a new readout system. Image sensing and processing facilities may be closely gathered into cameras or even mixed in the focal plane. At that point, image acquisition may be

improved thanks to closer control, possibly at the pixel level. Also, large incident image data flow can be reduced into compact forms.

This dissertation is composed of seven chapters. In chapter 2, existing FPA systems are reviewed. New technologies and research trends on readout systems, photodetectors, and A/D converters are also presented. Chapter 3 starts with a discussion about a several readout architectures and ends with a speed and noise comparison among the architectures. Chapter 4 describes the oversampling A/D converter which is designed for the fully parallel FPA readout system in this dissertation. Design, simulation, and layout are also presented in detail. Other circuits such as a current buffer, an integrator, and a current digital-to-analog converter are also presented. Test and measurement results of these circuits are presented in Chapter 6. Two different image sensors are described in Chapter 5. In Chapter 6, test setup and test results are summarized.

The final chapter of this dissertation is devoted to summarizing the results and discussing the contributions of this research. In addition, future research and possible enhancements for the FPA readout system are discussed.

CHAPTER 2

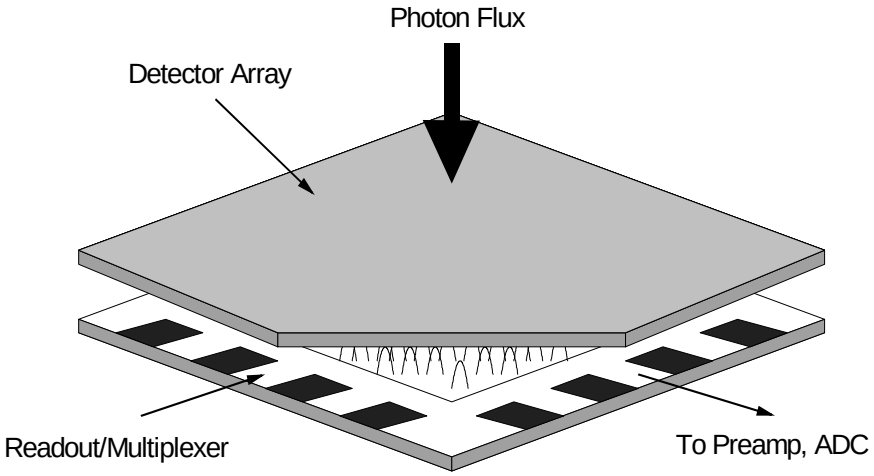
BACKGROUND

There are many kinds of electronic image capture systems accompanied by different characteristics. Focal-plane-arrays are one of those systems and have progressed rapidly with the development of new detectors, readout systems, and A/D converters. With the progress of several techniques, it is possible to build a single chip image capture system that includes integrated timing and control electronics, sensor arrays, signal processing electronics, analog-to-digital converters, and interface circuits.

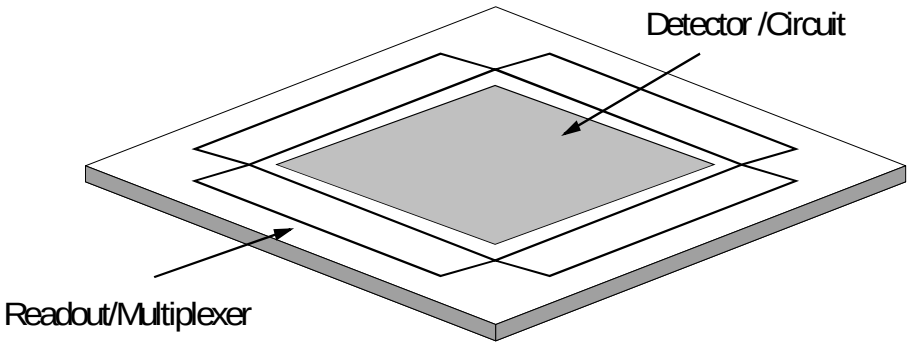
In this chapter, the four major parts of the focal-plane-array system are reviewed. The first one reviewed is photodetectors which generate an electronic signal related to the input photon quantity. The next one reviewed is an A/D converter which translates the analog signal to a digital signal. Readout system architecture is another important block which determines the speed of data acquisition system. The last building block is the signal processing part which generates valuable information from the binary code train.

2.1 Photodetectors

Photodetectors generate electronic signals and are located at the front end of the image acquisition system. There are two options which can be explored regarding the integration of the detector onto the silicon circuitry. One option is that the detectors can be fabricated using hybrid integration on silicon CMOS as shown in Figure 2.1(a). This integration technique includes bump bonded and thin film detectors and has several advantages [10]. The first advantage is that the material which comprises the detector array is fabricated separately from the electronics. Due to this fact, the detectors need not be the same material as the electronics, which enables the use of higher responsivity materials such as direct gap compound semiconductors. This scenario also implies that the detector materials need not be lattice matched to the circuit materials. The second advantage of hybrid integration is that the detectors are integrated directly on top of the silicon circuitry; which, enables scalability with a high fill factor. Each detector is directly connected to the circuitry, which lies beneath the detector. However, the disadvantage of hybrid integration of detector arrays is that the detector array substrate must be transparent; whereby, yielding a high fabrication cost compared to the other detector types. Thin film bonded detector arrays do not suffer from substrate transparency problems because the substrate is removed from the devices [11],[12].



(a) Hybrid integration



(b) Monomaterial integration

Figure 2.1 Photodetectors

Monomaterial integration is another option which has been actively researched for the integration of detector arrays [13],[14],[15]. The detectors can be fabricated using monomaterial (single material) integration in the silicon CMOS as shown in Figure 2.1(b); however, the detector types and materials must be compatible with the silicon process. Low absorption coefficient is another drawback of monomaterial detectors. The low absorption coefficient translates to low responsivity due to the small achievable absorption length in a CMOS process. Nevertheless, there are several advantages with monomaterial detectors. The advantages are 1) the compatibility with integration of on-chip electronics, and 2) the potentially lower cost, weighed against the conventional hybrid image system.

The integration of on-chip timing, control, signal chain, and analog-to-digital converter, lowers focal-plane-array system cost since these components must otherwise be supplied and assembled with the focal-plane-array chip. The low cost of monomaterial image sensors is derived from the assumption that the cost of technology development will have been amortized by the huge sales volume of “standard” CMOS logic and memory chips.

2.2 Analog-to-digital converters

A number of ADC algorithms are available for use in focal-plane-array applications [16],[17]. The conversion methods differ from each other in terms of operating speed, power consumption, achievable accuracy, and chip area. A major difference between on-focal-plane ADC and a single-chip ADC is that, unlike the latter, an on-focal-plane ADC must occupy a relatively small chip area. Therefore, in choosing an ADC architecture, the immunity of the ADC performance to the circuit parameter mismatch is an important issue. In FPAs, the required resolution and the conversion rate vary widely depending upon the applications. The conversion rate depends on the array size, the integration time, and the choice of ADC architecture. Scientific images usually demand a high resolution, but several other applications do not require that high a resolution. Consequently, there is a wide range of operating requirements with a conversion rate requirement varying from 30Hz to 100kHz, and a bit resolution requirement varying from 8 to over 16 bits.

Flash ADCs are the simplest and potentially fastest converters; whereby, achieving 8-10 bits of resolution at a very high speed, however requiring a large chip area and a high power consumption [18]. Figure 2.2 is a block diagram of an m-bit Flash ADC. The circuit consists of 2^m comparators, a resistor ladder comprising 2^m equal segments, and a decoder. The ladder subdivides the main reference into 2^m equally spaced voltages, and the comparators compare the input signal with these voltages. For

example, if the analog input is between V_j and V_{j+1}, comparator A_1 through A_j produce *1s* at their outputs while the rest generates *0s*.

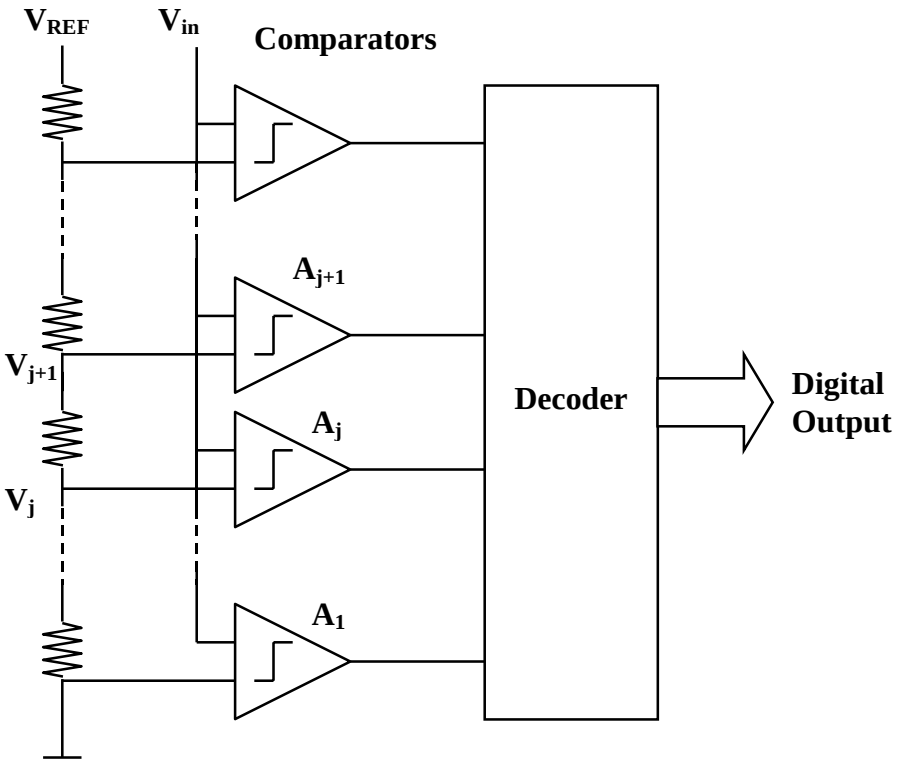


Figure 2.2 Flash ADC

Successive approximation employs a "binary search" algorithm in a feedback loop which includes a 1 bit A/D converter [18]. Figure 2.3 illustrates this architecture, with a front end S/H (sample-and-holding) circuit, a comparator, a shift register, a decision logic, a decision register, and a DAC. Successive approximation ADCs achieve high resolution at medium speed, but power consumption and area are still quite large. For a

resolution of *m* bits, the successive approximation architecture is at least *m* times slower than the flash converter. Circuit complexity and power dissipation of this ADC are, in general, less than those of other architectures.

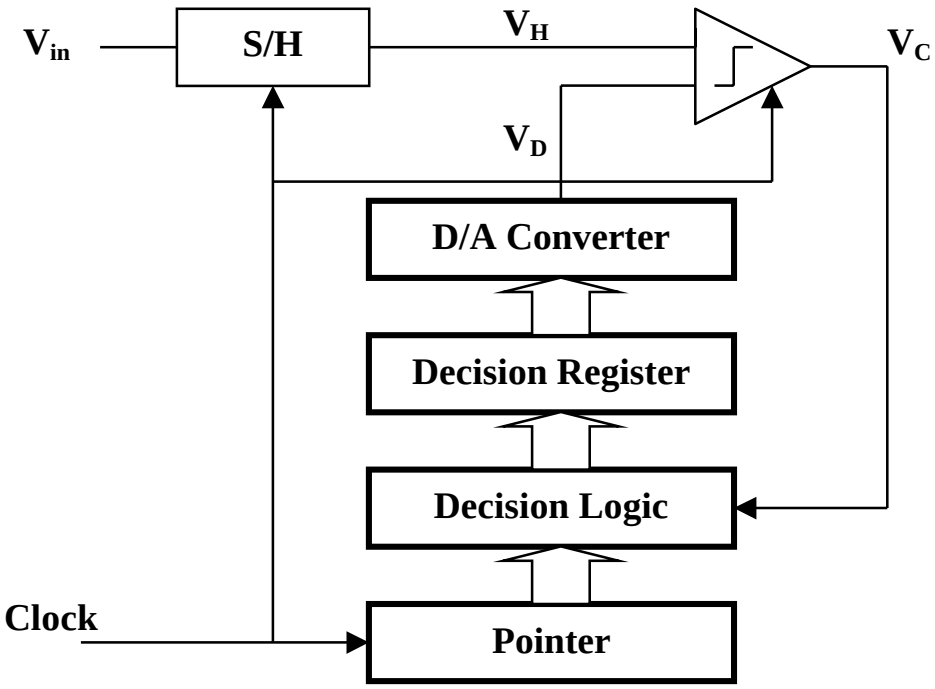


Figure 2.3 Successive approximation ADC

Cyclic ADCs occupy a smaller area, compared to the other ADCs, however the resolution is not adjustable and the noise is higher than the other ADCs [19]. Figure 2.4 is a Cyclic ADC. Its digital output is serial, starting with the MSB (most significant bit). The A/D converter samples an input signal then compares it against a voltage reference

$V_{ref}/2$, which is half of the full scale voltage V_{ref}. If the input is greater than $V_{ref}/2$, the MSB is set to *1*, and $V_{ref}/2$ is subtracted from V_{NOW}; otherwise the MSB is set to *0*, and V_{NOW} is not modified. V_{SUM} is then multiplied by *2* to get V_{NEXT}, and the cycle is repeated with V_{NEXT} replacing V_{IN} on the subsequent cycle.

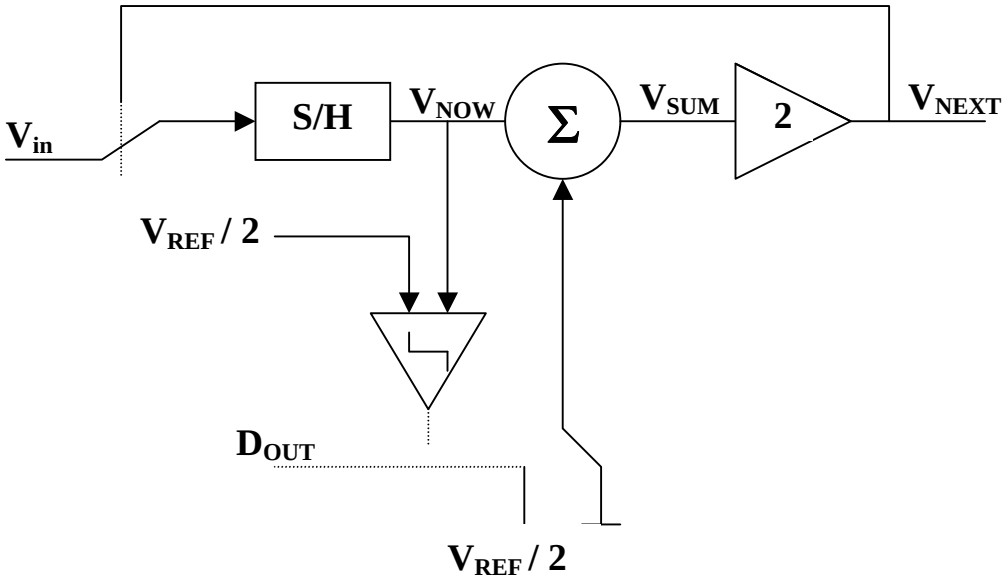


Figure 2.4 Cyclic ADC

Figure 2.5 shows the block diagram of a single-slope A/D converter [18]. This type of converter consists of a ramp generator, an interval counter, a comparator, an AND gate, and a counter that generates the output digital word. At the beginning of conversion cycle, the analog input is sampled and held and applied to the positive terminal of the comparator. The counters are reset and a clock is applied to both the interval counter and the AND gate. On the first clock pulse, the ramp generator begins to integrate the reference voltage V_{ref}. If V_{in} is greater than the initial output of the ramp generator, then

the output of the ramp generator, which is applied to the negative terminal of the comparator, begins to rise. Because V_{in} is greater than the output of the ramp generator, the output of the comparator is high and each clock pulse applied to the AND gate causes the counter at the output to count. Finally, when the output of the ramp generator is equal to V_{in}, the output of the comparator goes low and the output counter is now inhibited. The binary number representing the state of the output counter can now be converted to the desired digital word format. A disadvantage of the single-slope A/D converter is that it is subject to error in the ramp generator. Another disadvantage of the single-slope A/D converter is that a long conversion time is required if the input voltage is near the value of V_{REF}.

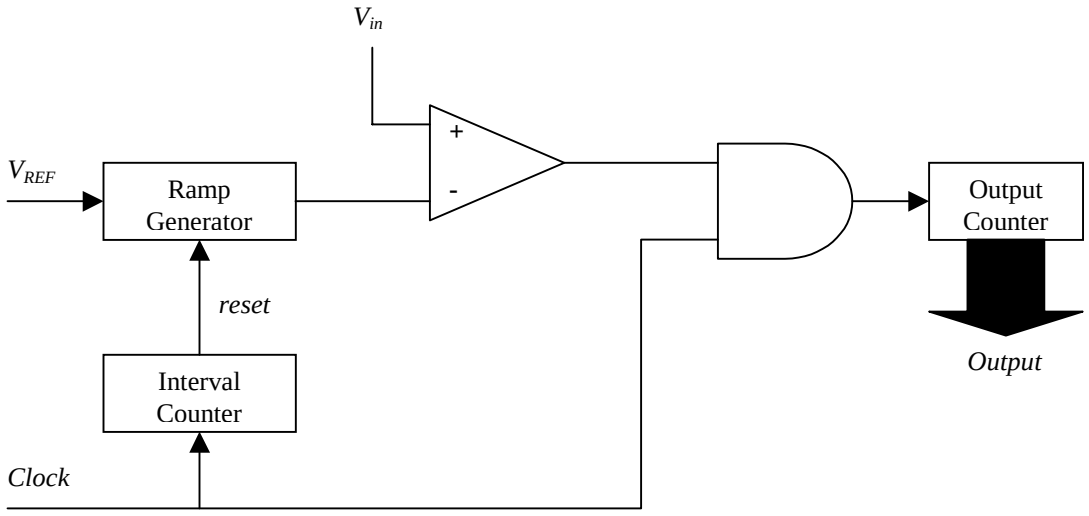


Figure 2.5 Single-slope ADC

Oversampled ADCs show good robustness for component mismatching while their modulator components occupy a small chip area [20],[21]. Oversampled ADCs also have resolution and readout rates that are adjustable. Oversampled ADCs consist of two major parts as shown in Figure 2.6 [22]: the modulator, which samples analog input and develops a corresponding digital bit stream, and the digital signal processor, which compresses the bit stream into the Nyquist rate multibit codes and performs noise filtering. Only the modulator needs to be on the focal-plane which reduces the development problem of fitting the ADC into the allocated space. Therefore, the oversampling ADC is a promising algorithm for integration on the FPA with a fully-parallel readout system.

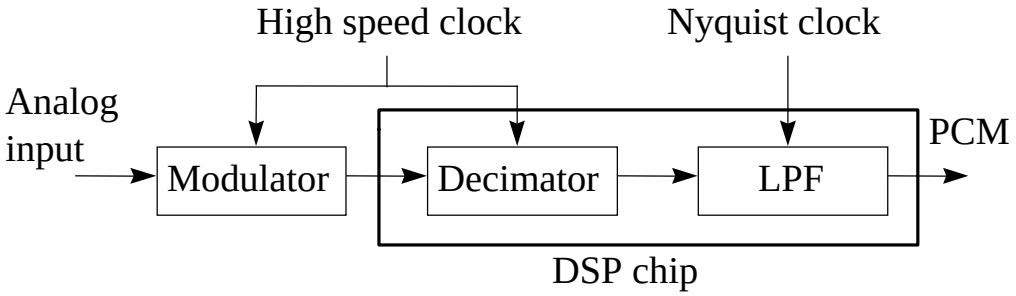


Figure 2.6 Oversampling ADC

As shown in Figure 2.6, a digital signal processing unit is needed to complete the oversampling A/D conversion. Several techniques have been developed for image collection and processing. Beginning with Mahowald and Mead's silicon retina [23], on-focal-plane processing which has increased in complexity from simple logic gates to

latches [24] to 2-bit registers and counters in [25]. These image processing solutions are compact and efficient, but lack computing power and flexibility.

The CM-2 [26] and MasPar [27] are more powerful processors for the general purpose image processing applications; however, these systems achieve performance and generality at the expense of focal plane I/O coupling and physical size.

The Scan Line Array Processor (SLAP)'s serial loading and unloading limits frame rates. The Morphological Image Processor (MIP) [28] combines dedicated processors with an on-chip focal-plane-array; whereby, functionality is limited to morphological operations on binary images.

There is another image processor which has been designed by Georgia Institute of Technology and named SIMPil [29]. It combines features from both focal-plane-array systems and image processing architectures. This SIMPil system is designed for the fully parallel processing between focal-plane-arrays and digital signal processor. The SIMPil system is an embedded, programmable, focal-plane image processing system. The processing power of the SIMPil node will surpass the computational needs of a single pixel; however, the desired frame rates may not be achieved if the number of pixels assigned to a node is too large. Simulations of image processing applications suggests a good balance of 36 to 64 pixels per SIMPil node.

Table 2.1 shows the summary of the on-chip focal-plane-array ADCs comparison. Flash ADCs and Successive-approximation ADCs require a large chip area and high power consumption; therefore, these are not suitable for some readout systems. The

noise of Cyclic and Single-slope ADCs are higher than the other ADCs, and need a matched capacitor which is not compatible with a digital CMOS process.

Table 2.1 Characteristics of ADC algorithm

TECHNIQUE	BITS	POWER	AREA	ROBUSTNESS	SPEED	NOISE	COMPATIBILITY
Flash	8-10	High	Large	Medium	High	Medium	No (linear R)
Succ. App.	10-12	Medium	Medium	Low	Medium	Medium	No (linear C)
Cyclic	7	Low	Small	Medium	Medium	High	No (matched C)
Single-slope	10-12	Medium	Small	Low	Medium	High	No (Linear C)
Oversampling	8-20	Low	Small (1)	High	Medium	Low	Yes (unmatched C)

(1) modulator only

2.3 Readout systems

In the development of FPAs, the readout system is a major part of determining the data acquisition rate. Readout circuits are designed to support an optimum interface between the detectors and the following signal processing stage. Different techniques have been developed for FPAs using different systems and circuits. In conventional focal-plane readout systems, off-focal-plane readout systems are used as shown in Figure 2.7. In this case, noise can be introduced into long analog shift register chains; whereas, this particular noise cannot be avoided and will cause a reduction in dynamic range. In addition, analog circuits in a serial ADC are required to operate with the highest bandwidth of all focal-plane components.

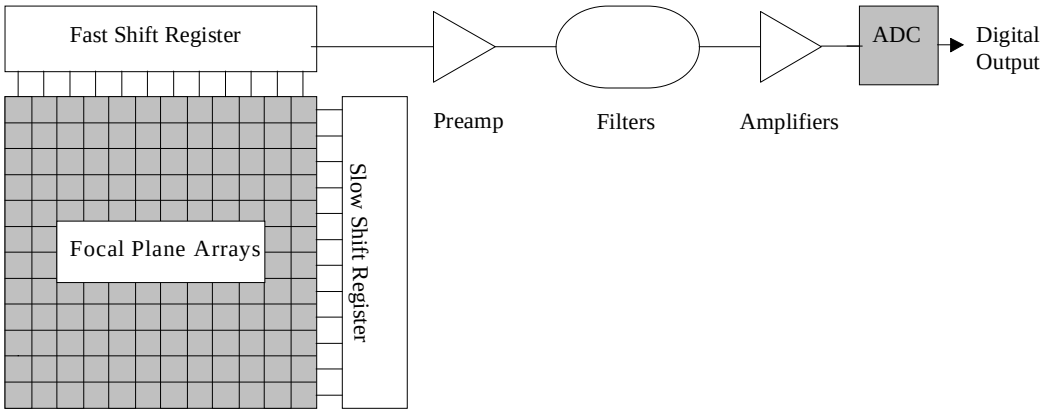
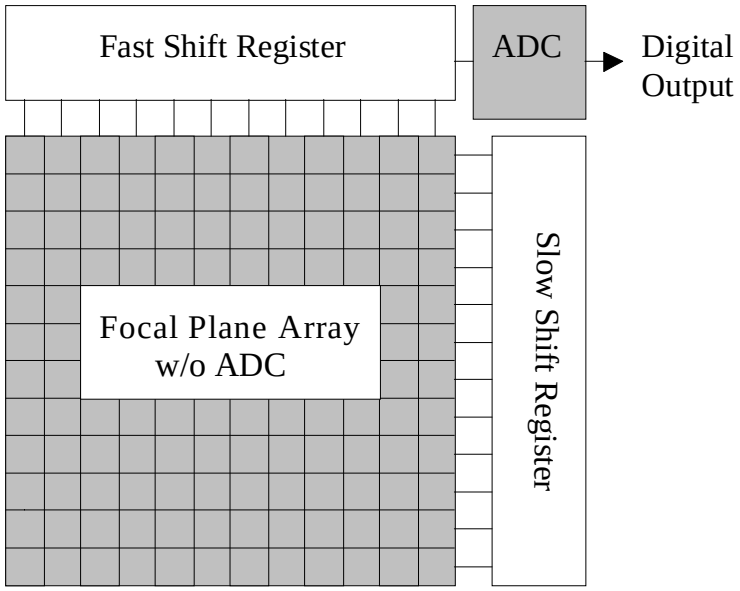


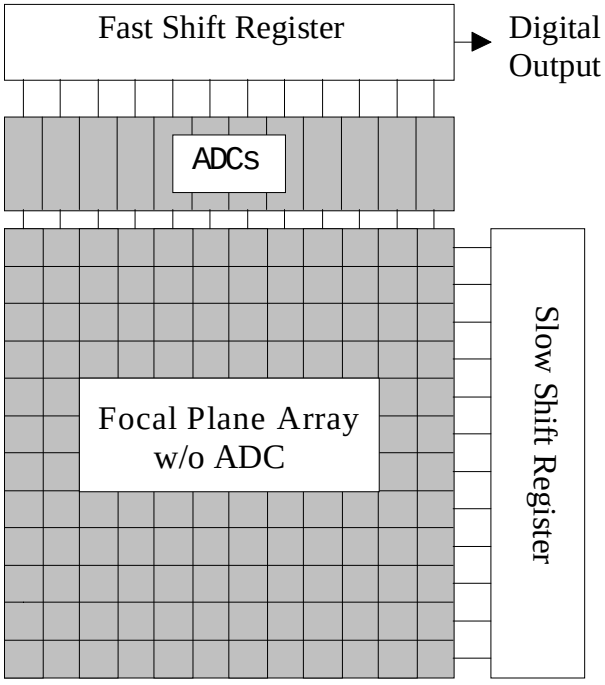
Figure 2.7 Off-focal-plane readout system

To improve signal-to-noise ratio (SNR), a serial on-focal-plane readout system is proposed as shown in Figure 2.8 (a) [30]. This architecture removes pick-up and vibration sensitivity since no off-chip analog cabling is required. For the serial ADC architecture, all available power and area can now be used for the single ADC; however, the bandwidth is several thousand times higher because only a short sampling time per detector element is available. This readout system has problems because the readout rate is not changed, and noise can still be introduced in the analog shift register between the focal-plane-arrays and ADCs.

The semi-parallel architecture proposed to overcome these problems is shown in Figure 2.8(b) [30], [31], [32]. The semi-parallel architecture has a ADC per each column; whereby, reducing the readout speed by a factor of the number of column lines. The readout of the semi-parallel sensor uses a row-by-row principle, where a row is selected at a given time. The data of the selected row is columnwised and independently amplified then A/D converted during the row selection time. Before a new A/D conversion starts, the digitized data is read into the parallel digital register; therefore, the data can be read out during the next conversion period. The area for the signal conditioning and ADC circuits is limited to the column pitch of the sensor in the horizontal direction; however, there is no restriction in the vertical direction other than fabrication costs related to chip area and maximum acceptable chip size. The conversion rate is the frame rate times the number of sensor columns. Compared to the serial concept, there is one level reduction of the analog multiplexing and a bandwidth reduction proportional to the number of sensor columns.



(a) Serial



(b) Semi-parallel

Figure 2.8 On-focal-plane readout systems

These make the concept considerably less sensitive to noise at the analog signal path, in comparison to the serial readout system. Nevertheless, this architecture is not scaleable and the noise can still be introduced between the output of the FPAs and the input of the ADCs through the analog line.

CHAPTER 3

READOUT SYSTEM ARCHITECTURES

To achieve image processing systems that operate in real time, and on large images with frame rates in the high kHz or MHz, is beyond the capability of today's imaging systems. For example, a first-order sigma-delta analog-to-digital converter (ADC) converting 500x500 8 bit images at a frame rate of 100 kHz must be clocked at more than 655 GHz. Even when parallel ADCs are placed along the edge of the imaging array, the problem is only partially mitigated because the speed at which the ADCs must operate still increases with image size. To generate 500x500 8 bit images at a frame rate of 100 kHz, 500 ADCs need to be clocked at more than 1.31 GHz [33].

In this dissertation, a fully parallel readout system was designed as a scalable focal-plane-array readout system. This readout system provides a scaleable solution to the real time high frame rate image capture problem when it is coupled to a massively parallel optically interconnected processor. To keep the design scaleable the processors must reside beneath the imaging chip, which use 3-D electrical interconnect for a parallel connection to the detector plane. A through-substrate parallel optical data link was also used to connect the two layer system.

3.1 A new readout system design

To overcome the serial and semi-parallel readout limitations, a focal-plane-array (FPA) was implemented using a fully parallel readout architecture. The fully parallel readout system was designed so that the pixels can be grouped into sub-arrays; whereby, each is served by one digital signal processing (DSP) unit to perform the conversion. Each pixel has its own ADC. To keep the area of the ADC circuitry small, a first-order current input sigma-delta oversampling ADC was chosen. This type ADC was chosen because only the front end of the sigma-delta A/D converter need be implemented, on a per pixel basis. Since sigma-delta converters process only digital data after the front end, further noise, by shifting the digital data, could not be introduced to the signal. An integrated optoelectronic emitter on each sub-array allowed through-silicon wafer output of digital image data from the focal plane to the processor stacked below each sub-arrays as shown in Figure 3.1. This vertical coupling to the image plane allowed the detector and processor arrays to be scaled while maintaining a fixed level of processing per pixel. Therefore, the processing rate does not depend on the array size anymore, consequently the readout system is scalable. The number of pixels included in the sub-array depends on the bandwidth of the DSP circuits. An example of the proposed processor is the SimPil processor [34][35]. If an 8x8 sub-array is used, the size of the processor and focal plane sub-array seem to match reasonably well. A 256×256 pixel 8 bits resolution focal plane could be achieved by a tiling 8x8 array processor working at 168 MHz.

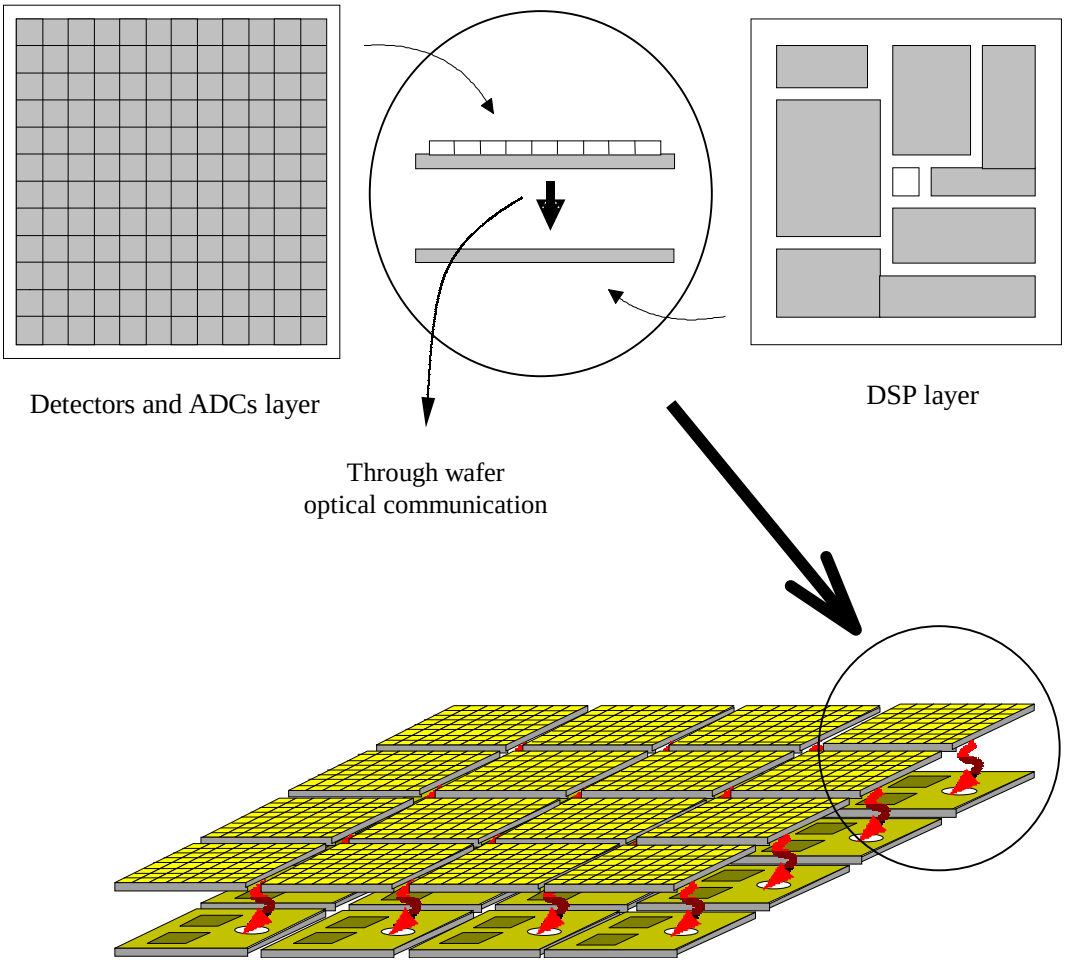


Figure 3.1 A stacked two layer focal plane system

Figure 3.2(a) shows a sub-array of the area image sensors. Each sub-array consists of an array of 8X8 multiplexed pixel blocks, consisting of a photodetector and a single A/D converter. Each pixel block was connected to a bit line using a pass transistor. The bit lines were “read” using an 8 row address decoder, and amplified using an array of 8 digital sense amplifiers. The topology was identical to read only memory circuits [36]. Each pixel block converted the analog light intensity into a digital code, and the entire system was synchronized by a master clock. After each clock pulse every pixel block produced one bit data, generating a two dimensional array of bits. Although two kinds of photodetectors were used for the FPA systems, the readout systems were exactly same. The only difference was a hybrid detector which had a contact pad and a bigger capacitor, compared to the monomaterial detector. Those differences did not affect the readout speed of the entire system. Figure 3.2(b) shows a pixel with a monomaterial detector and a pixel with a hybrid detector. All the generated digital output signals were amplified by the emitter driver in order to drive an integrated optoelectronic emitter on each sub-array.

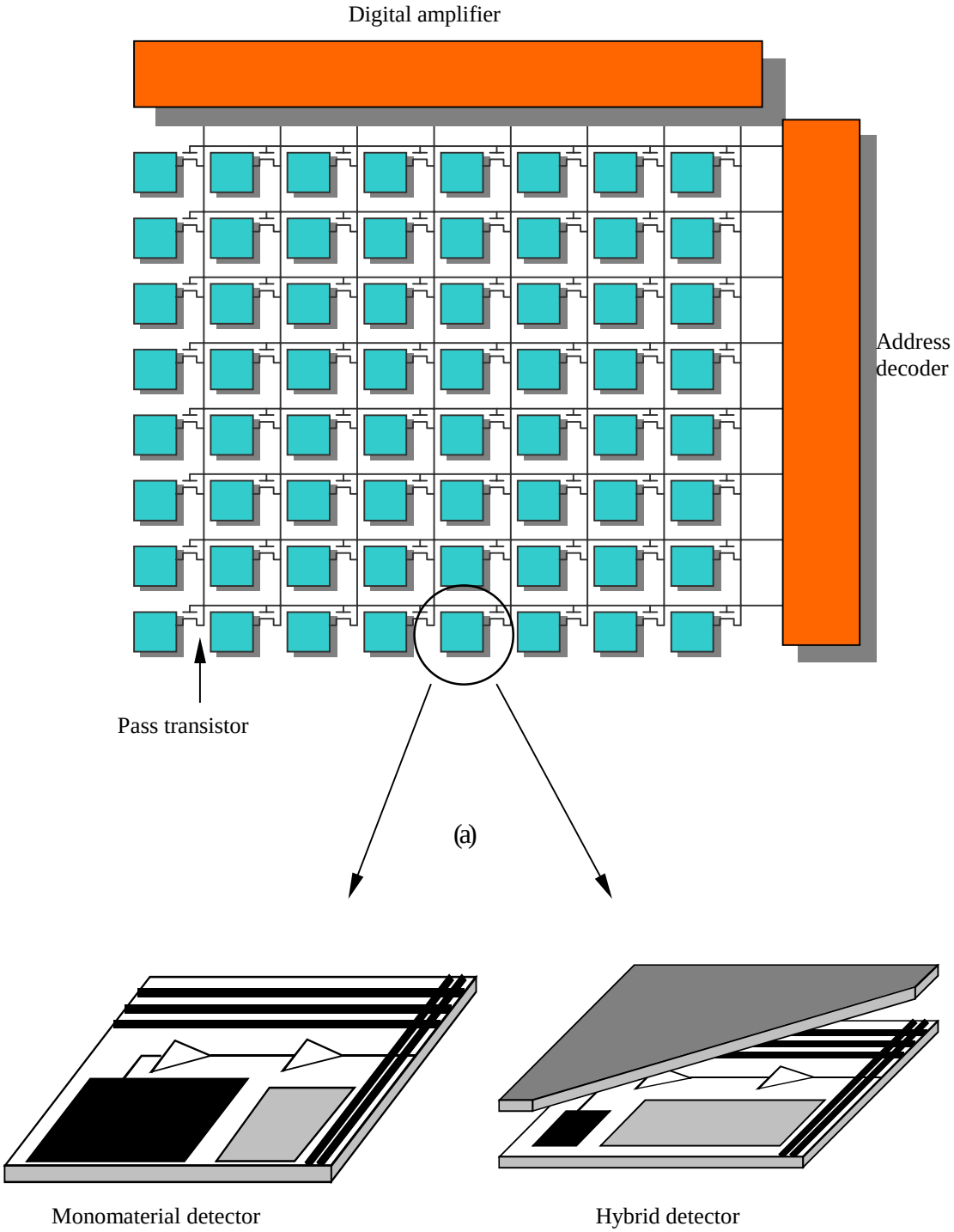


Figure 3.2 8X8 Focal-plane-arrays

After the optic signal was received by the receiver, located underneath the focal-plane-array, the optic signal was amplified by the receiver then synchronized to the SIMPil processor by a clocked comparator. The serial output of the comparator was read into the SIMPil processor by serial-to-parallel conversion. The signal path from image detector to signal processor is shown in Figure 3.3, and a photomicrography of two layer chip is shown in Figure 3.4. To achieve a 100 kHz frame rate, each SIMPil processor needs to process data at 167 Mbps (for an 8x8 sub-array image oversampled by 26).

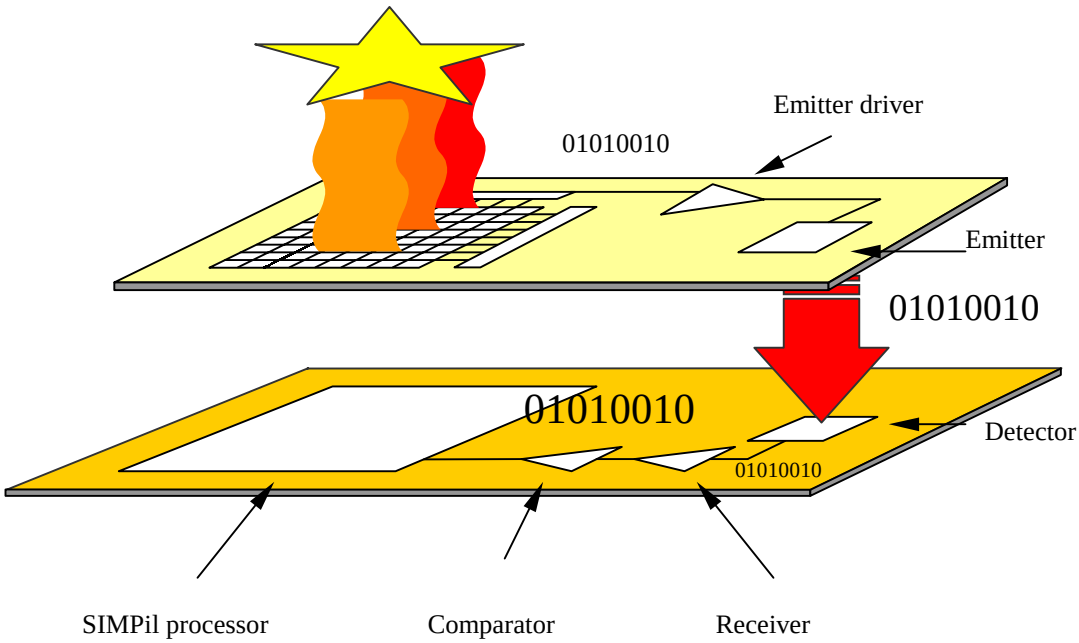
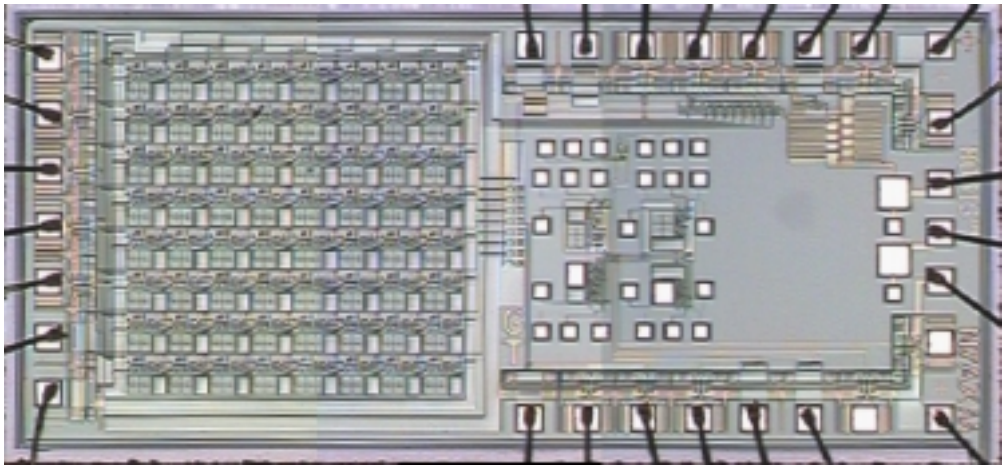
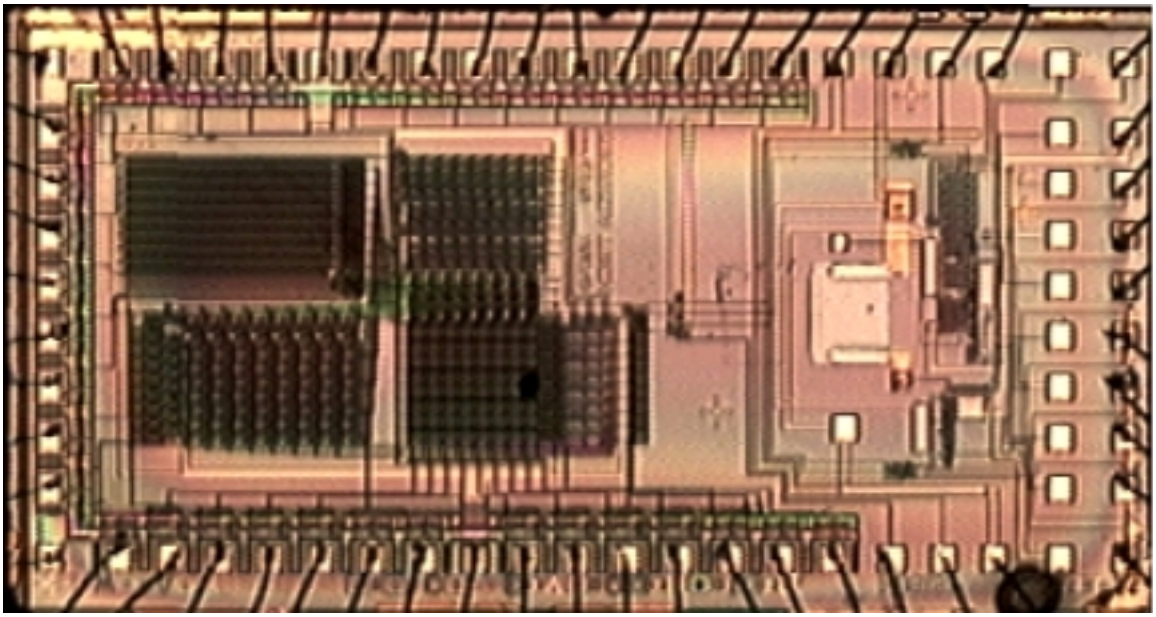


Figure 3.3 Signal flow of the FPA sub-array



(a) Focal-plan-array chip



(a) Digital signal processing chip

Figure 3.4 Two layer FPA system photomicrographs

3.2 Comparison

The main reason to use the fully-parallel readout architecture for the focal-plane-array system was its high readout speed capacity. In this section, three different readout systems are compared to readout speed and signal-to-noise ratio. Before the difference between the three readout systems is compared, there must be some assumptions for the proper comparison. A/D converter types, area, and power consumption must be considered first. The readout speed is highly related to the A/D converter type, and some A/D converters can not be fit into parallel or semi-parallel readout systems. Power consumption is another factor of the focal-plane-array system.

In this dissertation, a first-order sigma-delta A/D converter was chosen as a standard A/D converter because its small and will fit into any architecture. If the A/D converters are identical and the readout speed is same, the power consumption will be the same for the different architecture focal-plane-array systems. In analyzing the power consumption of an CMOS circuits two components, static and dynamic power consumption, need to be considered; whereby, static power is dissipated when the circuit is not changing states. For example, there is almost no power dissipation for the idealized CMOS inverter of Figure 3.5. After the output capacitance (C_L) has been fully charged or discharged, only one of the pullup and pulldown transistors is on. While there is a slight ($\sim 10^{-12}$A) leakage current through the channel of an off-transistor, there is almost no current flowing through the circuit and consequently almost no power consumption.

Power is consumed when gates drive their outputs to new values. It is called dynamic power consumption. The dynamic power consumption depends only on the size of the capacitive load at the output and the rate at which the inverter's output switches (See Appendix A).

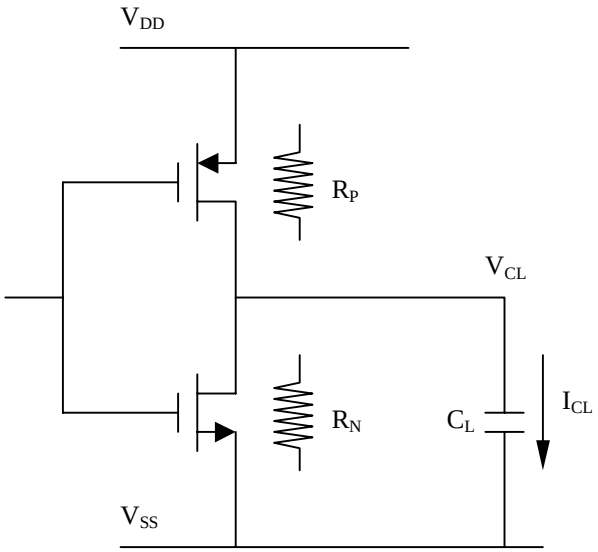


Figure 3.5 CMOS inverter

3.2.1 Readout speed comparison

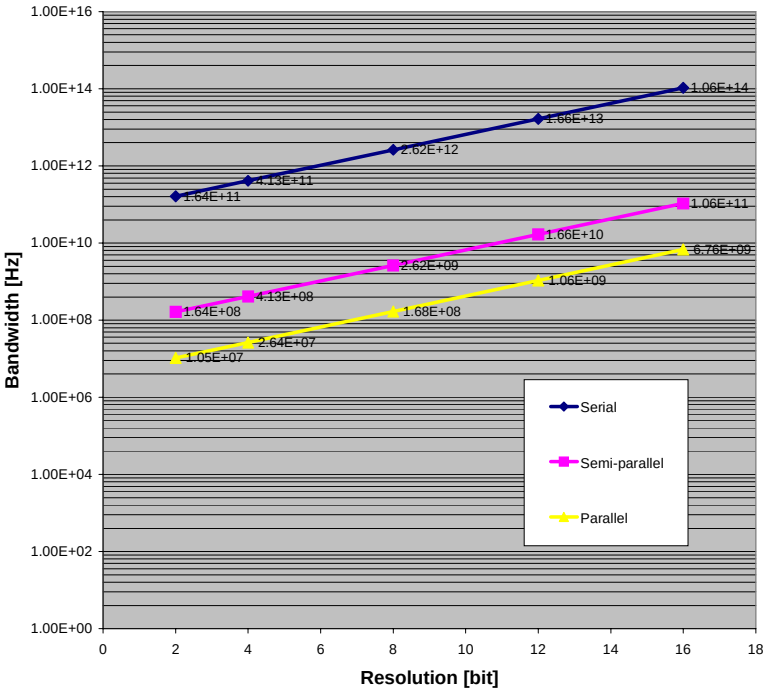
As discussed in a previous section, if the operating speed and the number of A/D converters are considered, the total power consumption with the FPA readout system is almost identical with the different architectures. Under this assumption, the operating speed is compared without any consideration of the power consumption. Figure 3.6 shows simulation results about the bandwidth against resolution and array size. Figure 3.6(a) is obtained under following assumptions:

1. First, the array size is 1000x1000.
2. Second, the frame rate of the system is 100kfps.
3. Third, 8x8 arrays are used for a sub-array of fully parallel system.

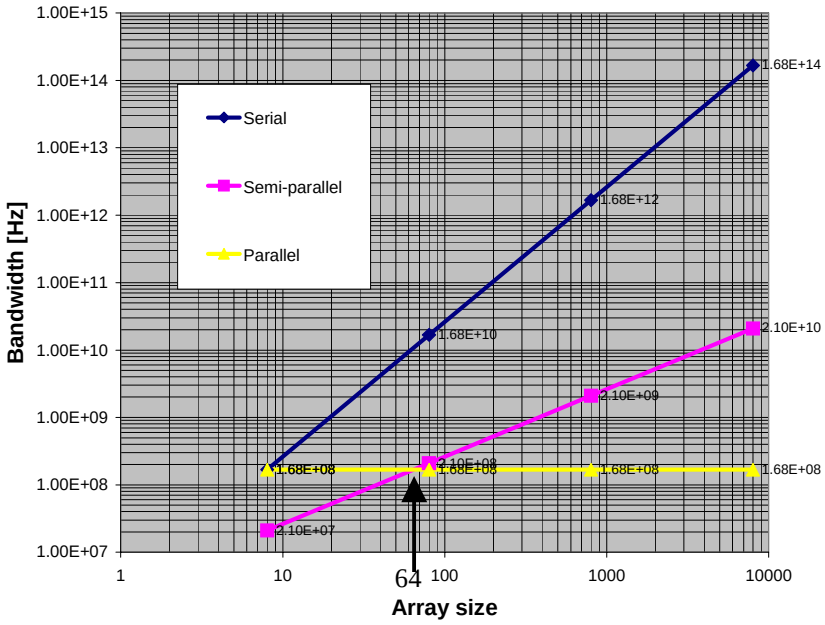
With the condition of an 8 bit 100kfps image the bandwidth of the first-order sigma-delta ADC is 168 MHz for parallel system, 2.62 GHz for semi-parallel system, and 2.62 THz for serial readout system. The system bandwidth is the same as the A/D converter bandwidth for the parallel and serial systems; however, it is increased for the semi-parallel readout system because there is only one processor for the whole FPA system.

Figure 3.6(b) is another simulation result with a fixed 8 bit resolution and a different array size. The bandwidth of a parallel system is independent of the array size; however, the semi-parallel and serial readout system bandwidth exponentially increases with array size. From the simulation results, the semi-parallel readout system has less bandwidth compared to the parallel readout system when its array size is smaller than 64x64. These simulation results are coming from the assumption that each row had its own signal processor for the semi-parallel system. If there is only one processor for the whole system, the semi-parallel system bandwidth would be the same as the serial readout system. From the above two graphs, it is clear that with same resolution and array size the parallel readout system has less bandwidth than the other two readout systems.

To compare the bandwidth among the readout systems, it is assumed that the same type A/D converter is used for different readout systems. This assumption is not true if there is no area restriction and power consumption. For the semi-parallel readout system,



(a) Bandwidth vs. resolution



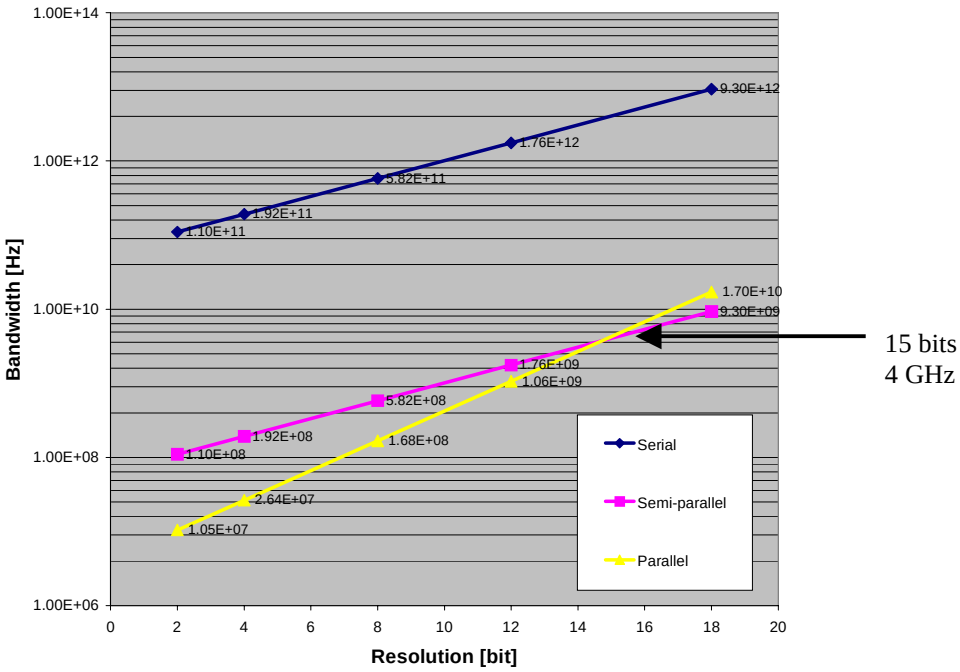
(a) Bandwidth vs. array size

Figure 3.6 Readout system bandwidths

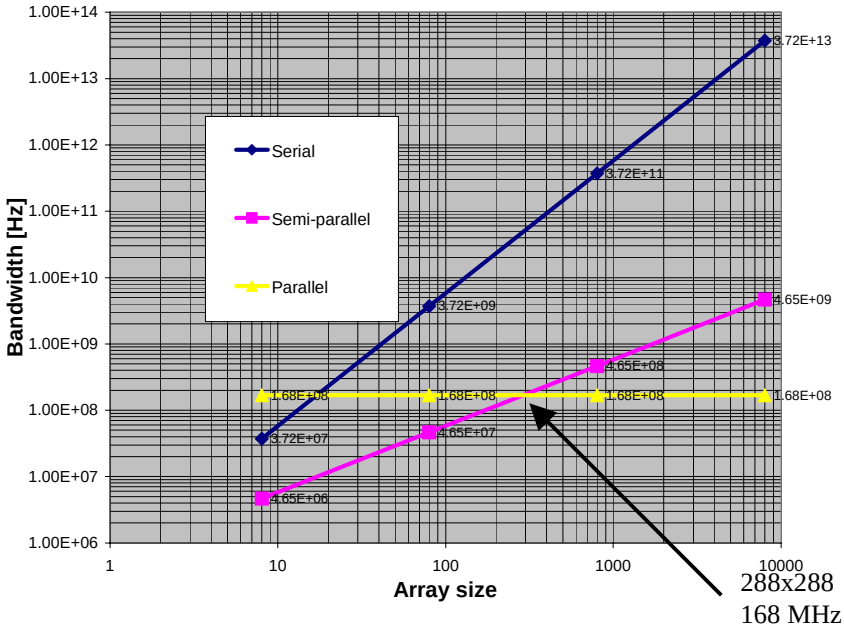
it is a good idea to use second-order sigma-delta A/D converters rather than first-order sigma-delta A/D converters as long as the area is available. By using a second-order sigma-delta A/D converter, the oversampling ratio is decreased significantly [37]. For the serial readout system, there is no limitation in choosing an A/D converter type. In the following simulation, a second-order sigma-delta A/D converter is used for the comparison. All the other assumptions are the same as the previous assumptions.

Figure 3.7(a) shows an interesting simulation result concerning bandwidth. Although 4 GHz is not a realistic number, the semi-parallel readout system shows less bandwidth compared to the parallel readout system over 15 bits resolution. Figure 3.7(b) shows another interesting simulation result; which is that, the semi-parallel readout system is better under a 288x288 array size as long as each signal processor supports each row.

It can be concluded from the above simulation results that the parallel readout system has best performance with low resolution and large image arrays.



(a) Bandwidth vs. resolution



(a) Bandwidth vs. array size

Figure 3.7 Readout system bandwidths with different ADCs

3.2.2 Signal-to-noise comparison

One of the biggest noise sources for the serial and semi-parallel readout system is the switching noise coming from the array addressing switch. As it is shown in Figure 3.8, each pixel needs a select switch and it generates a noise when it is turned on and off. In this figure, R_b is a detector bridge output resistance and R_d is a detector resistance. When the fully parallel readout system is used, the addressing switch is located after the A/D conversion; whereby, no serious noise problem to the output signal is experienced. The switching noise coming from the addressing switch is as follows for the serial and semi-parallel readout systems [38][39].

The MOS transistor in Figure 3.8 is operating in the linear region when activating the selected pixel. In this region, it acts as a resistor with resistance,

$$R_T = \frac{1}{\mu C_{OX} \frac{W}{L}(V_{GS} - V_T)} \tag{3.1}$$

where

μ = Mobility C_{OX} = Gate-oxide capacitance

W = Transistor width V_{DD} = Transistor length

V_{GS} = Gate source voltage V_T = Threshold voltage

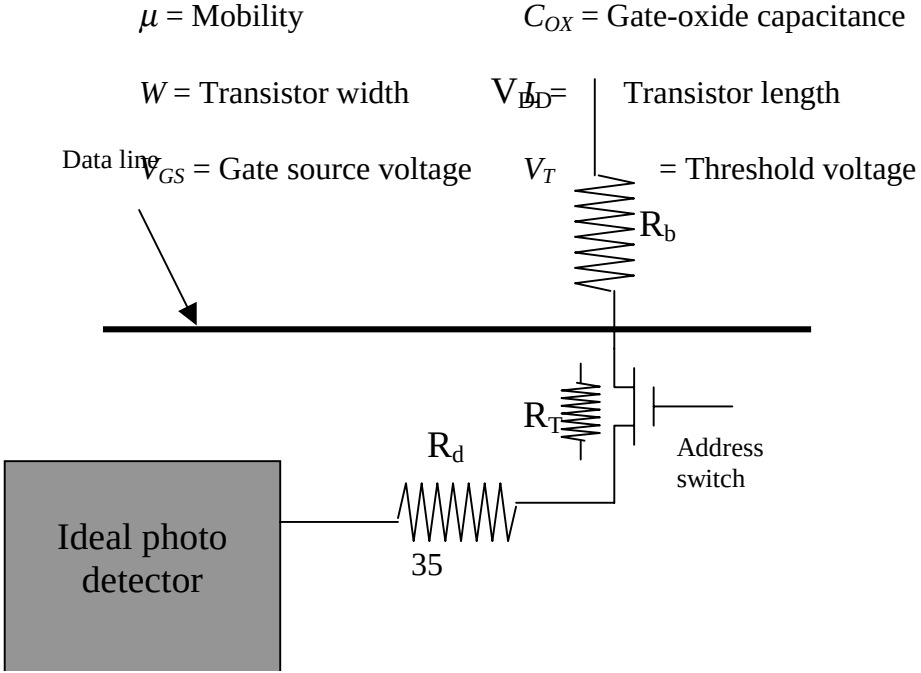


Figure 3.8 Photo detector with address switch

The thermal noise for an MOS transistor working in the linear region is same as that would be produced by a simple ohmic resistor located in the drain-source channel, and having a value given by R_T. Because of low resistance in the MOS channel, the thermal noise can be neglected when compared to the thermal noise in the detector bridge. The thermal noise in the detector bridge is calculated by following equation [40].

$$e_t^2 = 4kT\left(\frac{R_b R_d}{R_b + R_d}\right)BW \tag{3.2}$$

By using the proposed readout system, the switching noise from the addressing switch can be removed. This is because the analog signal is directly converted to a digital

signal through the A/D converter; whereby, the signal is directly connected to the detector then the digitized output is connected to the switch. The digital signal actually has a much higher SNR compared to the analog signal.

In addition to the switching noise, there is another noise source from the serial and the semi-parallel readout systems. This noise source is *1/f* noise. The *1/f* noise for a MOS transistor can be characterized in a number of ways. In the actual process used, the *1/f* noise can be characterized by the equation (3.3) for a transistor working in the saturation region [40],

$$e_f^2 = \frac{K_f I_{DS}^{A_f}}{C_{ox} L^2 f} \tag{3.3}$$

where K_f is the flicker noise coefficient, and A_f is a constant.

A large amount of parallelism would reduce the *1/f* noise, because *1/f* noise is inversely proportional to the gate area, which can be increased when bandwidth is reduced. It should be easier to make a low noise MOS transistor using a semi-parallel structure than using a serial structure.

CHAPTER 4

COMPACT OVERSAMPLING CONVERSION

Oversampling converters trade speed for accuracy. They allow simple compact digitizing units or modulators, to be built and with DSP, achieving significantly lower noise operation than other converters operating at the same effective original sampling rate.

Modern short-channel CMOS processes offer a speed performance which far exceeds the requirements of the proposed system. Since shorter channel lengths will be available in the future, speed will be further improved; however, accuracy and component matching are expected to become worse. It could be a serious problem for the fully parallel FPA readout system because there are thousands of ADCs working simultaneously, and needing good uniformity to get a good image. Hence, it will be interesting to trade off speed for accuracy, and obtain some accuracy advantages at the cost of a speed limitation for the focal-plane-array ADCs.

A current input oversampling ADC was designed in this dissertation to obtain good performance. The concept of the oversampling ADC and the proposed current input ADC are described in this chapter.

4.1 Oversampling ADC

Oversampling ADCs contain very simple analog circuits called modulators (comparators, switches, and one or more integrators and analog summing circuits), and complex digital computational circuitry as shown in Figure 4.1 [41]. An analog input enters the modulator, where it is sampled at a very high rate – many times the Nyquist rate. It is called a modulator because the analog signal is pulse density modulated, that is, the density of the pulses at the output over a given period is approximately equal to the mean value of the analog input over the same period. The modulator generates a 1-bit output stream, which is digitally filtered to remove the out-of-band quantization noise produced by the modulator. The cutoff frequency (f_C) of the digital filter depends on the input signal bandwidth, and the resulting resolution of the signal depends on the oversampling ratio. The sampling rate is then lowered to any rate greater than the Nyquist rate,(f_N).

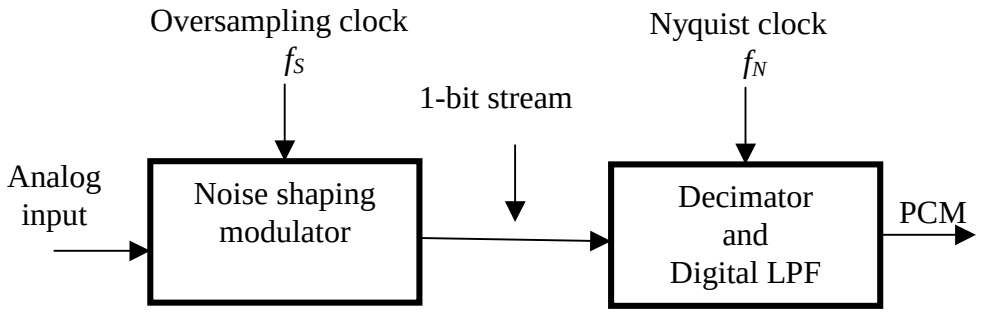


Figure 4.1 Sigma-delta oversampling ADC

Figure 4.2 shows a block diagram of a simple modulator [42], which contains one integrator, a 1-bit A/D converter, and a 1-bit D/A converter. In the sampled-data or discrete-time domain, the previous output of the 1-bit D/A converter is subtracted from the analog input signal *x(n)*, and the resulting signal is integrated and quantized. The resulting 1-bit digital output *y(n)* is then converted into one of two analog levels by the 1-bit D/A converter. The 1-bit quantization that goes on in the modulator generates a high level of quantization noise. This quantization noise is spectrally shaped by the modulator to reshape most of the energy lying at high frequency. In following section, the quantization noise is calculated.

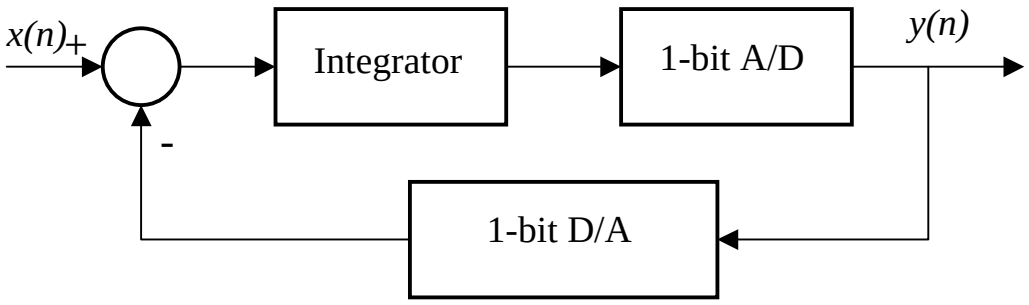


Figure 4.2 First-order sigma-delta modulator

4.1.1 Quantization noise and zero-order sigma-delta modulator

An ADC is a circuit whose digital output is proportional to the ratio of its analog input to its analog reference. Often the scaling factor between the analog reference and the analog signal is unity, so the digital signal represents the normalized ratio of the two.

Figure 4.3 shows the transfer characteristic of an ideal 3-bit ADC. The input to an ADC is analog and is not quantized, but its output is quantized. The transfer characteristic therefore consists of eight horizontal steps. Digital full scale (all 1s) corresponds to 1 LSB (least significant bit) below the analog full scale. This is because the digital code represents the normalized ratio of the analog signal to the reference, and if this were unity, the digital code would be all 0s and 1 in thee bit above the MSB (most significant bit).

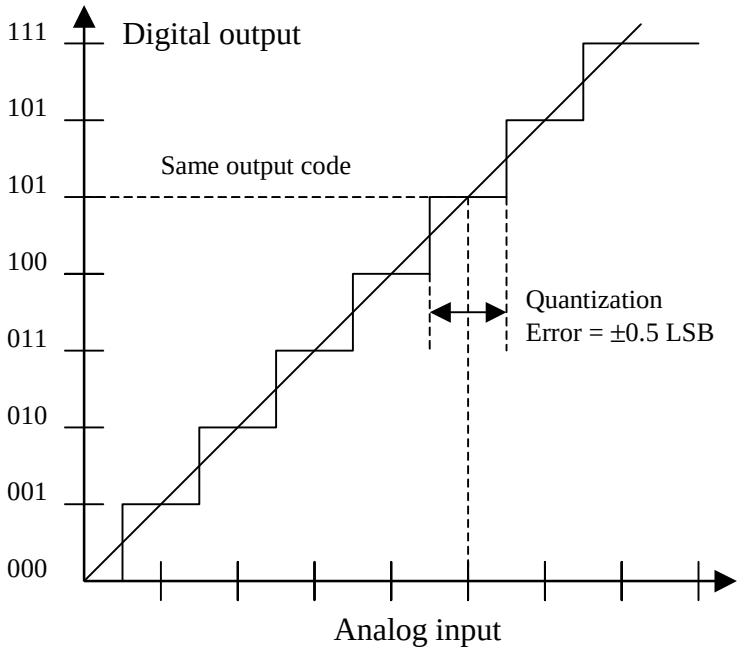


Figure 4.3 Quantization

The ideal ADC transitions take place at 0.5 LSB above zero and thereafter every LSB, until 1.5 LSB below analog full scale. Since the analog input to an ADC can take on any value (the digital output is quantized) there may be a difference of up to 0.5 LSB

between the actual analog input and the exact value of the digital output. This is known as the quantization error. In AC applications, this quantization error gives rise to quantization noise.

Candy and Benjamin [42] have given a simplified analysis of sigma-delta quantization noise, whereby the 1-bit A/D converter is approximated as a white-noise source.

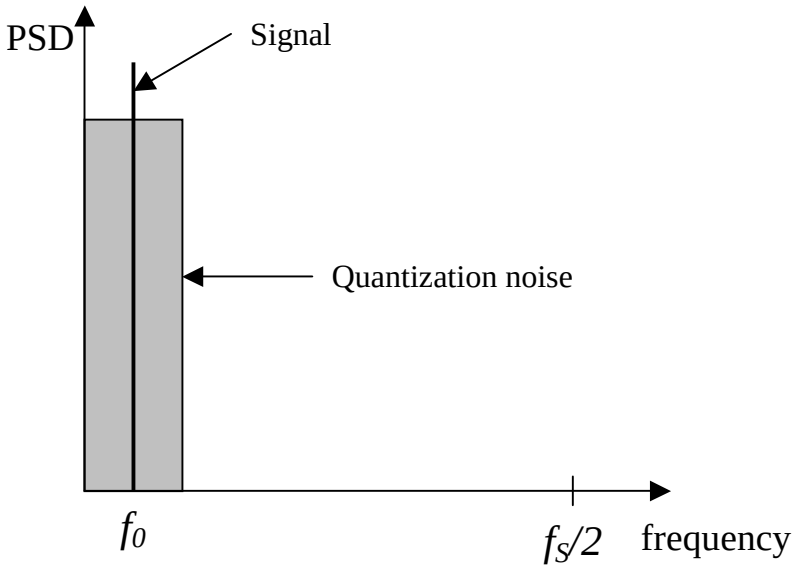
The noise power that falls into the signal band will be

$$n_0^{\,2} = \frac{e_{rms}^{\,2}}{OSR}. \qquad (4.1)$$

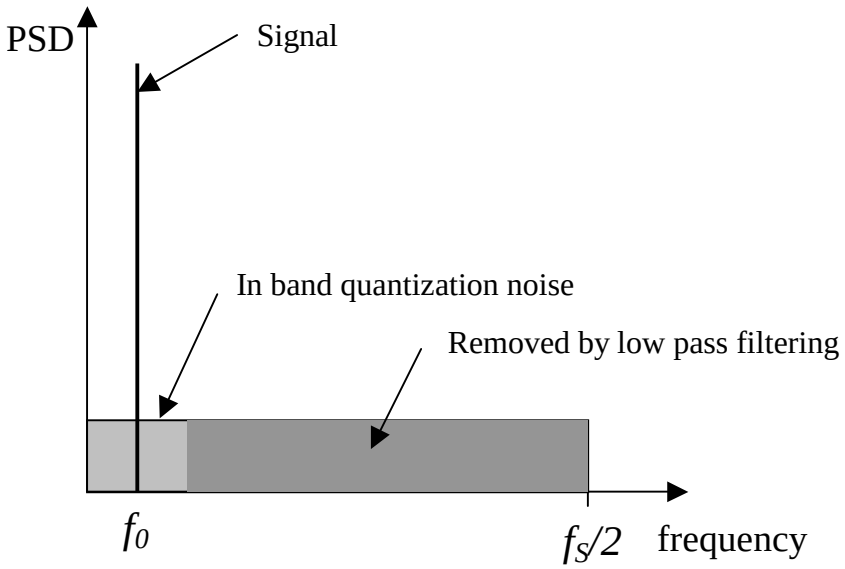
Thus, oversampling reduces the in-band *rms* quantization noise, n_0, by the square root of the oversampling ratio. Therefore, each doubling of the sampling frequency decreases the in-band noise by 3 dB and increases the resolution by 0.5 bit. This modulation is also called zero-order oversampling modulation. Figure 4.4 shows that the quantization noise in the signal band is distributed on the whole bandwidth when the signal is oversampled. The total amount of the quantization noise is same for the both cases.

4.1.2 Sigma-delta modulator

A more efficient oversampling quantizer is a sigma-delta modulator as shown in Figure 4.2. A Sigma-delta modulator is the most common oversampling modulator



(a) Nyquist rate sampling



(a) Oversampling

Figure 4.4 Quantization noise

architecture, which consists of a noise shaping modulator with a 1-bit internal quantizer. The input to the circuit is fed to the quantizer via an integrator, and the quantized output is fed back and subtracted from the input. This feedback forces the average value of the quantized signal to track the average input. The difference between them accumulates in the integrator and eventually corrects itself.

To analyze the operation of a first-order sigma-delta modulator, the aforementioned model is developed. The basic model of a typical first-order modulator is shown in Figure 4.5. Using the switched capacitor technique, the integrator usually has the delay in the forward path. The input of the modulator is defined as $x(t)$, and the output as $y(kT)$. The comparator in the forward loop can be modeled as a unity gain block in conjunction with an additive noise source having error "e", and the D/A converter in the feedback loop can be represented by a unity gain element.

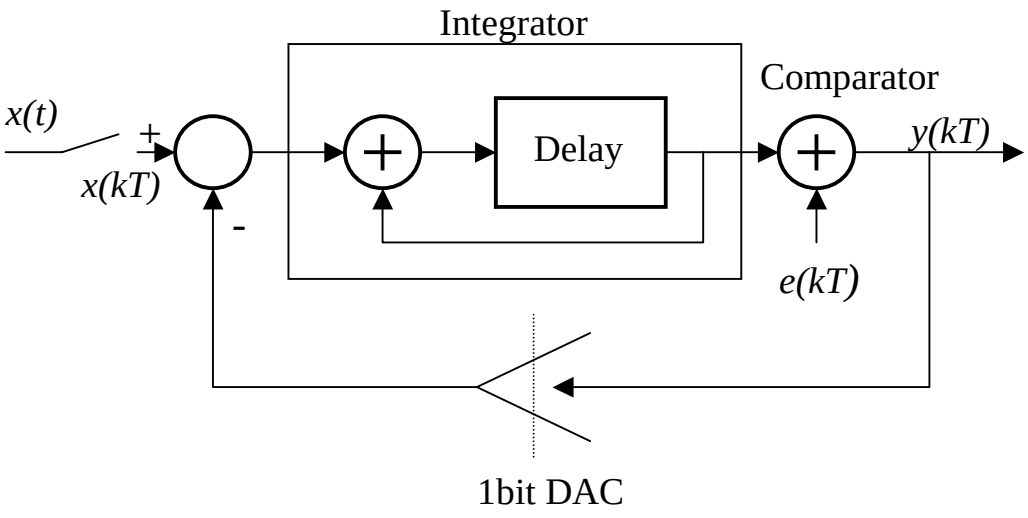


Figure 4.5 First-order sigma-delta modulator

The difference equation describing the entire system is:

$$y(kT) = x(kT - T) + e(kT) - e(kT - T). \tag{4.2}$$

The noise term is represented by:

$$n(kT) = e(kT) - e(kT - T). \tag{4.3}$$

Thus, this circuit differentiates the quantization error, making the modulation noise small while leaving the signal unchanged except for a delay.

To calculate the effective resolution of the sigma-delta modulator, the input signal is assumed sufficiently busy; therefore, the error can be treated as white noise which is uncorrelated with the signal. The spectral density of the modulation noise can then be expressed as:

$$N(f) = E(f)\left|1 - e^{-j\omega\tau}\right| = 2e_{rms}\sqrt{2\tau}\,\sin\left(\frac{\omega\tau}{2}\right). \tag{4.4}$$

In Figure 4.6, the spectral density of the modulation noise expressed by equation (4.4) is compared with that of the quantization noise $E(f)$. Clearly, the feedback around the quantizer reduces the noise at low frequencies but increased it at high frequencies.

The noise power in the signal band is:

$$n_o^2 = \int_0^{f_0} |N(f)|^2 df \approx e_{rms}^2 \frac{\pi^2}{3}(2 f_0 \tau)^3, \quad f_s^2 \rangle\rangle f_0^2 . \qquad (4.5)$$

Each doubling of the oversampling ratio consequently reduces the in-band noise by 9dB and provides 1.5 bits of extra resolution.

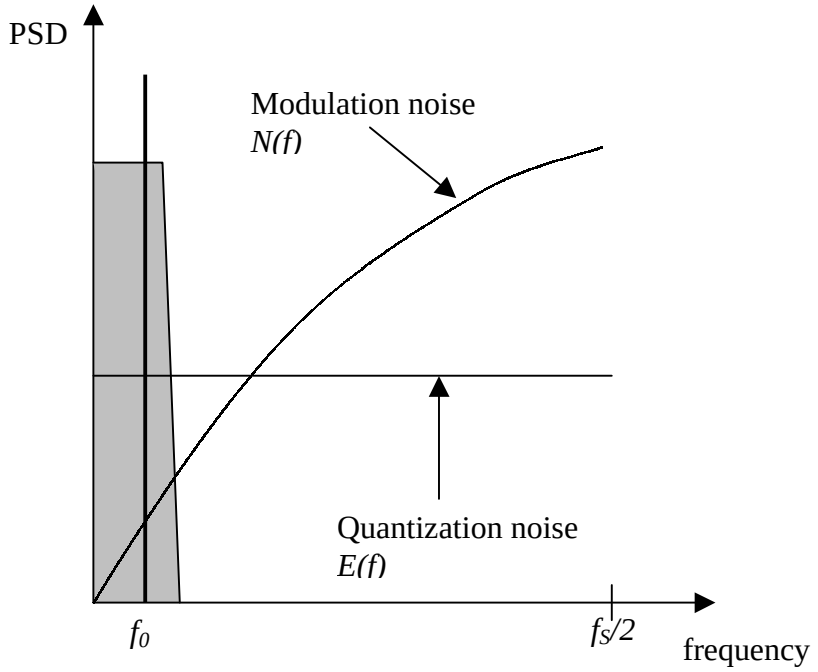


Figure 4.6 Modulation noise of first order oversampling modulator

In essence, the modulator cancels the error by subtracting quantization errors from two adjacent samples. This principle of reducing the error source by exploiting the statistics between them can be extended to higher-order modulators; whereas, more past error samples are involved in the cancellation process to reduce the overall error. Viewed from the frequency domain, this difference operation acts to attenuate the quantization noise at low frequencies.

Modulators with a second-order transfer function involve the cancellation of two past samples and exhibit stronger attenuation at low frequencies. The noise shaping functions of a first and second-order modulator are compared in Figure 4.7. As the order of the system increases, the quantization error in the signal band is decreased. However, modulators with more than two integrators suffer from potential instability owing to the accumulation of large signals in the integrators [43]. Second-order sigma-delta modulators are therefore particularly attractive for high-resolution A/D conversion. A detailed description about the second-order sigma-delta modulator is explained in appendix B. In this dissertation a first-order sigma-delta ADC was designed rather than a second-order because the second-order sigma-delta ADC is too big to fit in the pixel.

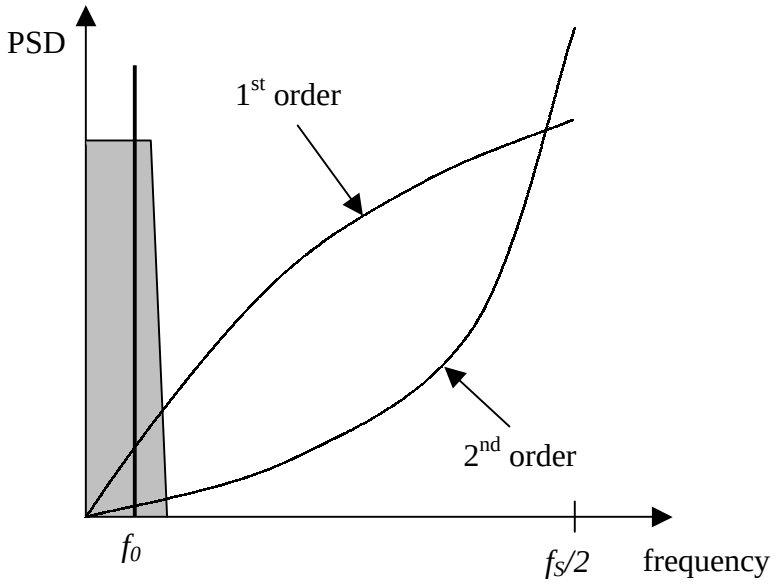


Figure 4.7 Power spectral density for the first and second-order modulators

4.2 Current input oversampling ADC

The simplified architecture of the designed current input first-order modulator is shown in Figure 4.8. The blocks that made up the system is briefly described below.

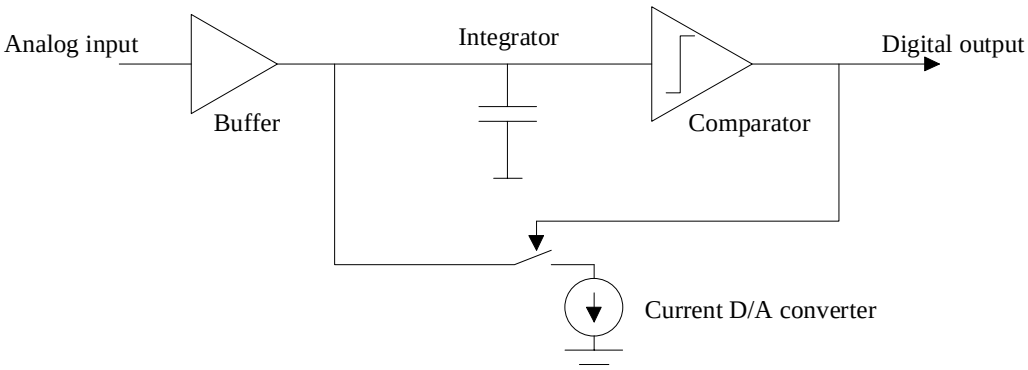


Figure 4.8 first order oversampling modulator

In the FPA application, a current buffer was needed between the photodetector and oversampling modulator to provide low input impedance and a stable bias to the detector. A current buffer typically must provide a low input impedance to reduce the effects of the nonzero output admittance of the detector. It also had to supply a specified DC bias voltage for the input device to improve the linearity of the detector.

An integrator was realized by a floating capacitance because the integration value was current. The output of the current buffer and the current D/A converter were integrated by the capacitor. The capacitor size was limited by pixel size, maximum speed, and resolution of the FPA system.

The D/A converter was realized by a cascode current mirror circuit with a feedback switch which was controlled by the output status. The cascode method increased the linearity of the current mirror circuit. The D/A converter injected current pulses into the integrating capacitor; whereby, the terminal of this capacitor was the node where the system feedback loop closed.

A comparator was used for the quantization of the integrator output.

4.2.1 Design features

There were several design features used to build a current input sigma-delta A/D modulator. The first design feature was to improve the oversampling loop linearity. Since a feedback structure reduced the effects of the nonlinearities of the elements following the gain block in its forward path, the architecture maximized the linearity of the entire system by putting most circuit elements inside the forward path of the feedback loop. The main function of the loop was to shape the quantization noise; whereas, only the input stage was outside of the loop. The second design feature was that the amplifiers were removed from the feedback. Operational amplifiers were not require in the system, so the difficult problems of providing frequency compensation and reducing the settling time were avoided. The third design feature was that a linear D/A conversion was available; whereby, a single bit D/A converter had ideal linearity. In practice, however, when the D/A converter switched from one state to another, spikes were generated in its output waveform. The fourth design feature was that the modulator was working under continuous-time. The oversampling modulator was not realized as a switched capacitor

(SC) network, so the problems associated with SC circuits, such as clock feedthrough and digital noise, were avoided.

In following sections, each part of the current input sigma-delta modulator is explained in detail.

4.2.2 Current buffer

The proposed parallel readout system used a current buffer as a front end of a readout circuit to provide low input impedance and a stable bias to the detector. A current buffer typically must provide a low input impedance to reduce the effects of the nonzero output admittance of the detector. It also has to supply a specified DC bias voltage for the input device to improve the linearity of the detector. In this dissertation, a CMOS current buffer was designed as shown in Figure 4.9. It Assumes that M_1 and M_2 as well as M_3 and M_4 are matched, and all transistors are working under the saturation regions. Therefore, the DC voltage of node D is maintained at V_{bias} voltage by the matching condition. The input impedance is represented by equation (4.6). Small signal equivalent circuit and the calculation procedure are in appendix C.

$$Z_{in} \approx \frac{1}{g_{m1}}\left(\frac{g_{d1}+g_{d3}}{g_{m3}}+\frac{g_{d2}+g_{d4}}{g_{m2}}\right). \tag{4.6}$$

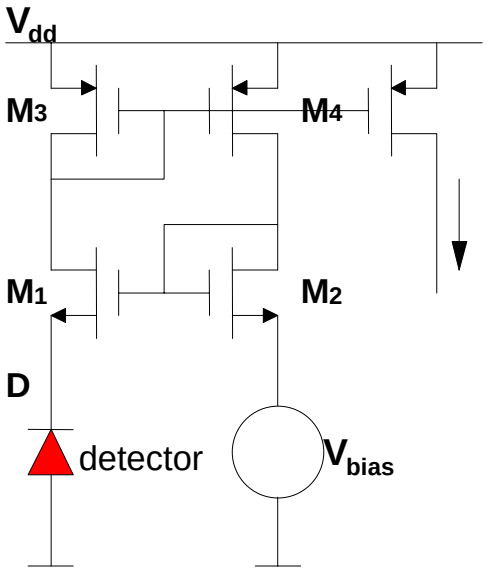
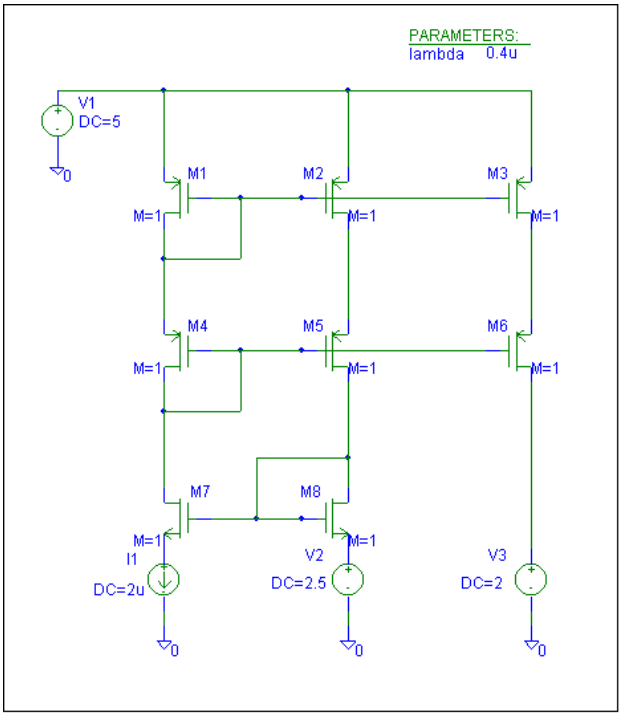


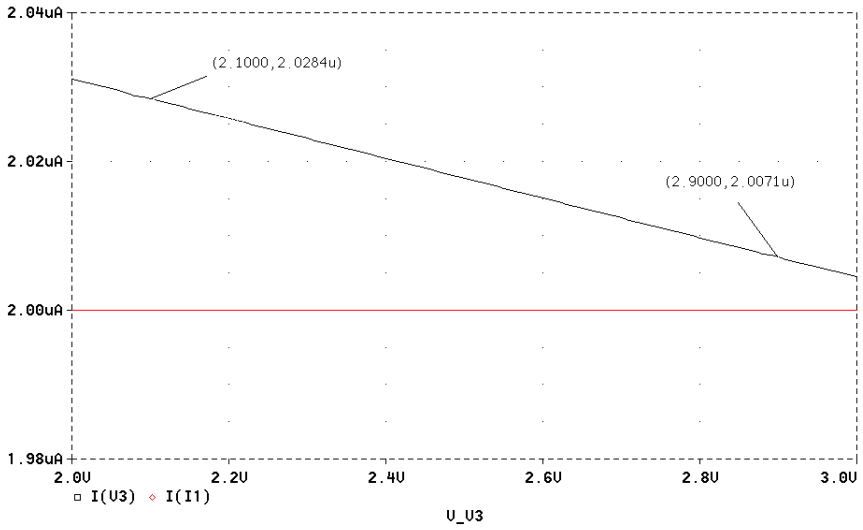
Figure 4.9 Current buffer

Thus, ordinary input impedance ($1/g_{m1}$) is multiplied by a factor which is typically around 0.1, consequently Z_{in} can be as low as few hundred ohms. The simulated input impedance of the cascode minimum size current mirror is approximately 40kΩ.

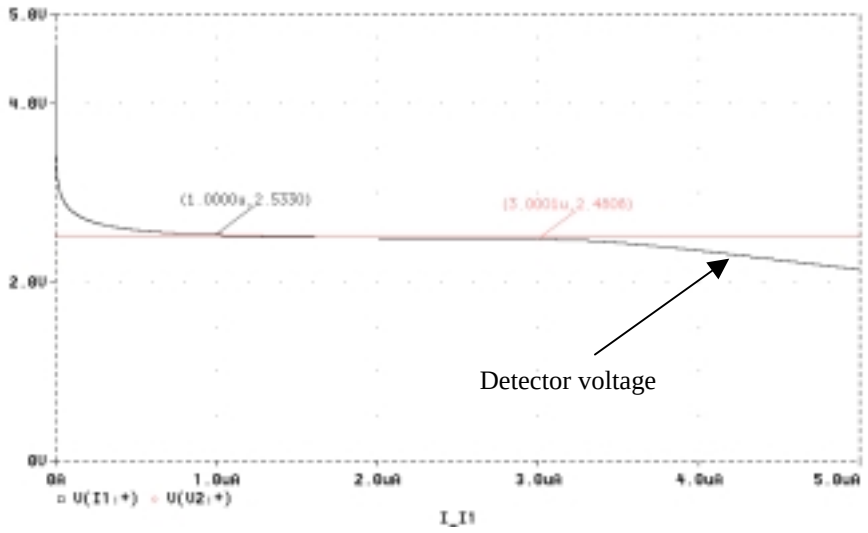
Figure 4.10 shows the schematic and simulation results of the output current with 2uA input with 0.8v output voltage variation. The 0.8v voltage variation comes from the assumption that an 8 bits 100kfps system with a maximum of 2uA input current and an 800fF integrator size. The center of the output voltage variation was set to 2.5V because the output node was connected to the integrator input node, and then was adjusted to switching at 2.5V. From the simulation results, the maximum current variation was 0.0213μA. This variation was attenuated by an oversampling technique. Figure 4.10(c)



(a) Schematic



(b) Output current related to the load bias



(c) Detector bias voltage variation

Figure 4.10 Cascode current buffer

shows the detector bias voltage variation related to the current variation; whereas, the ideal bias voltage is 2.5V. To get good linearity out of the photo current the bias voltage needs to keep constant. This simulation result shows the robustness of the detector bias voltage.

4.2.3 Current D/A converter

The current D/A converter was implemented by cascoded method as shown in Figure 4.11. The current mirror output node was connected to the current buffer output node and a capacitor. The output current can be changed by the effects of channel-length

modulation which is caused by the output node voltage variation. This error is reduced by using the cascode method.

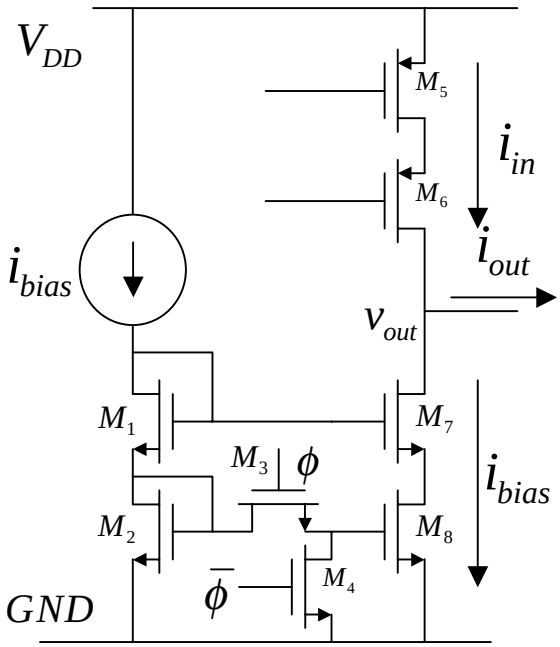


Figure 4.11 Cascode DAC

Cascoding is a well-known technique [44] for lowering the output conductance of a MOS transistor. The cascode transistors, M_6 and M_7, are biased with fixed voltages so that all transistors are kept in saturation. The low-frequency output conductance of the cascoded cell is shown to be as follows:

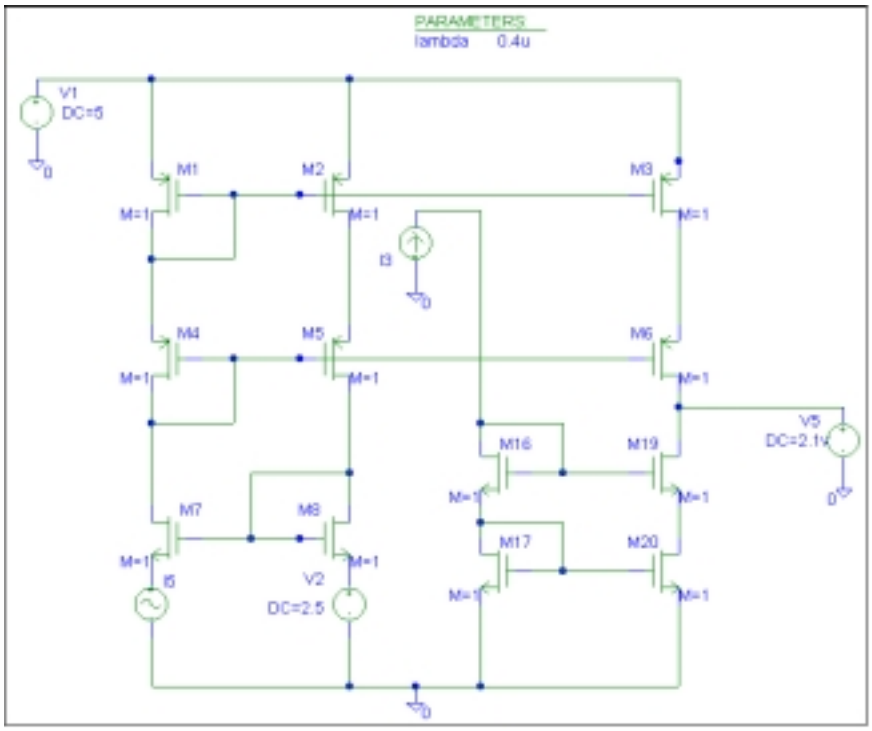
$$g_{OC} = g_o \left[\frac{g_{dsc}}{g_{ds} + g_{dsc} + g_{mc}} \right], \tag{4.7}$$

where g_o and g_{dsc} are the output conductance of the open-gate transistor and the drain conductance of the cascode transistor respectively, and g_{mc} is the transconductance of the cascode transistor. Now since $g_o, g_{dsc} \langle\langle g_{mc}$,

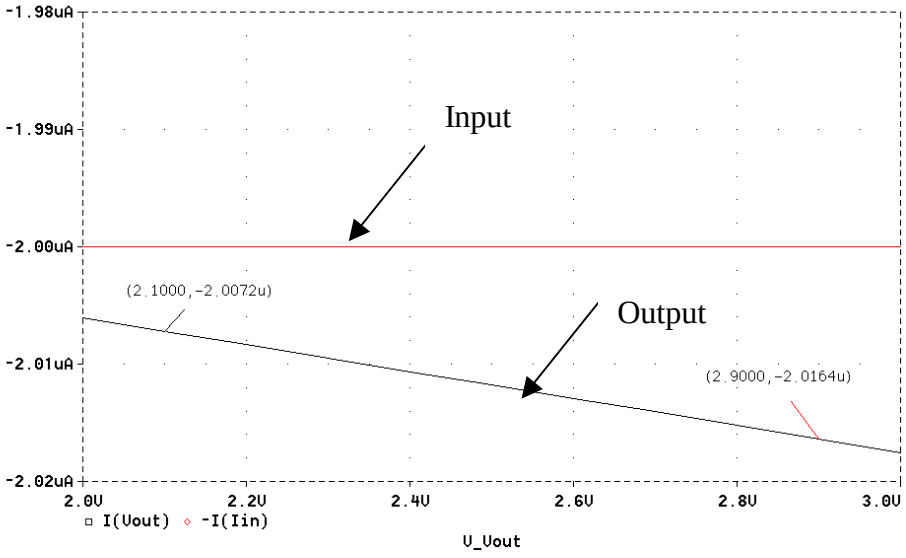
$$g_{OC} = g_o \left[\frac{g_{dsc}}{g_{mc}} \right]. \tag{4.8}$$

Thus, cascoding has decreased the output conductance by a factor approximately equal to g_{mc} / g_{dsc}. This factor is typically about 100 and gives the cascoded current mirror a transmission error resulting from a non-zero output to input conductance ratios; which is, typically 100 times lower than that of the basic current mirror. There are other methods such as a regulated cascode method [45] or a ground-gate method [46] to reduce the channel-length modulation effect.

To determine the size of the feedback current amplitude of the D/A converter, it should be bigger than the maximum detector output current to prevent overflow. It also can not be bigger than the saturation current that saturates the capacitor during one cycle. Therefore, the minimum current is decided by the photodetector, and the maximum current is determined by the capacitor size and clock frequency. Figure 4.12 shows the schematic and the simulation results with a 2uA input and 0.8v output voltage variation. The 0.8v voltage variation comes from the assumption of 8 bits 100kfps system with maximum 2uA input current and 800fF integrator size.



(a) Schematic

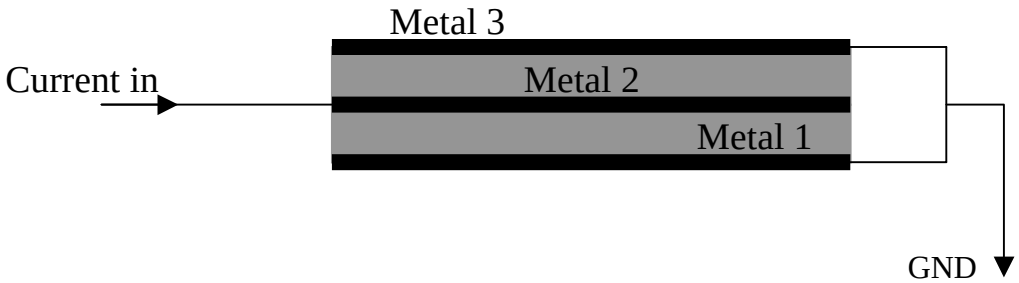


(b) Simulation result

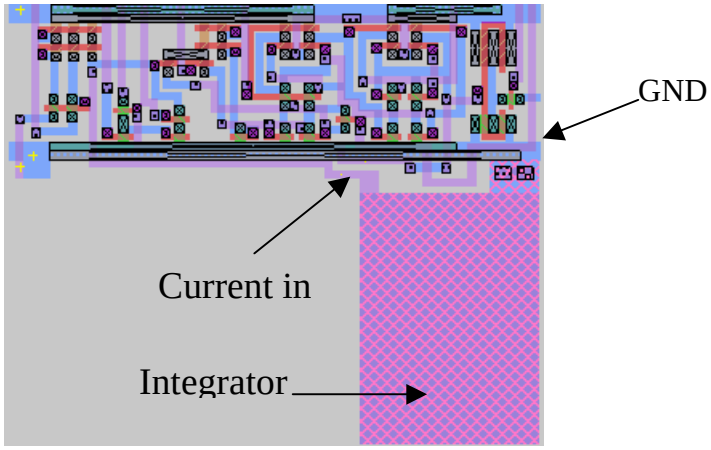
Figure 4.12 Current mirror schematics and simulation result

4.2.4 Integrator

The integrating capacitor was realized by using conductors as shown in Figure 4.13(a). There are floating capacitances among metal 2, metal 1, and metal 3. The capacitor size is 800fF which is designed for 100kfps 8 bits focal-plane-arrays. However, the capacitor should be as small as possible if the modulator has a high sampling rate. Figure 4.13(b) shows the layout of the capacitor.



(a)



(b)

Figure 4.13 Integrator

4.2.5 Comparator

In the proposed FPA application, a high speed CMOS comparator was designed. This CMOS comparator was originally designed by G. M. Yin [47] and is shown in Figure 4.14. This comparator consists of a differential input stage, two regenerative flip-flops, and an S-R latch. No offset cancellation is exploited, which reduces the power consumption as well as the die area and increases the comparison speed. By G. M. Yin, 8bit accuracy with 65 MHz sampling rate was achieved which is enough to get a 100kfps FPA system. The dimension of the comparator was optimized to obtain good performance. However, in this application all the transistors were designed with minimum gate length to minimize the size. Although the transistor sizes of the comparator were not optimized, there was no significant problem with this application.

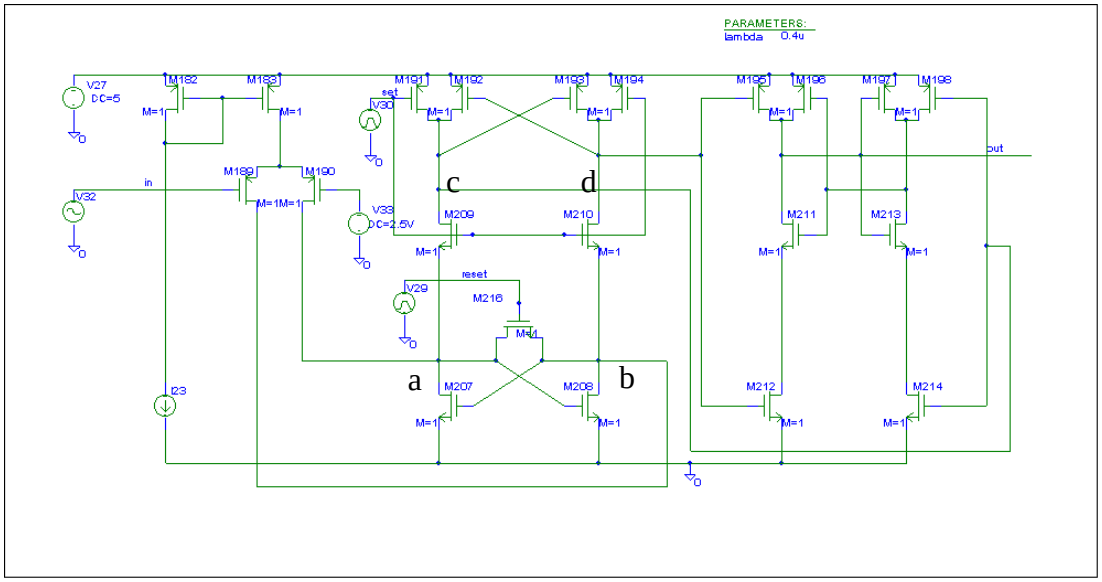


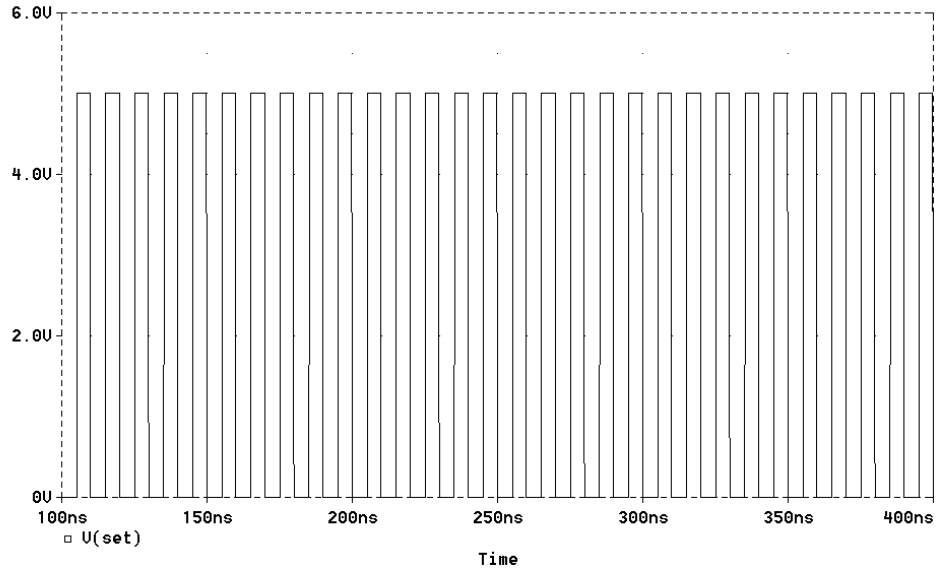
Figure 4.14 Schematic of the comparator

The only consideration for this application was the size and the speed. All the noise coming from the comparator was suppressed by the modulation technique and removed by low pass filtering.

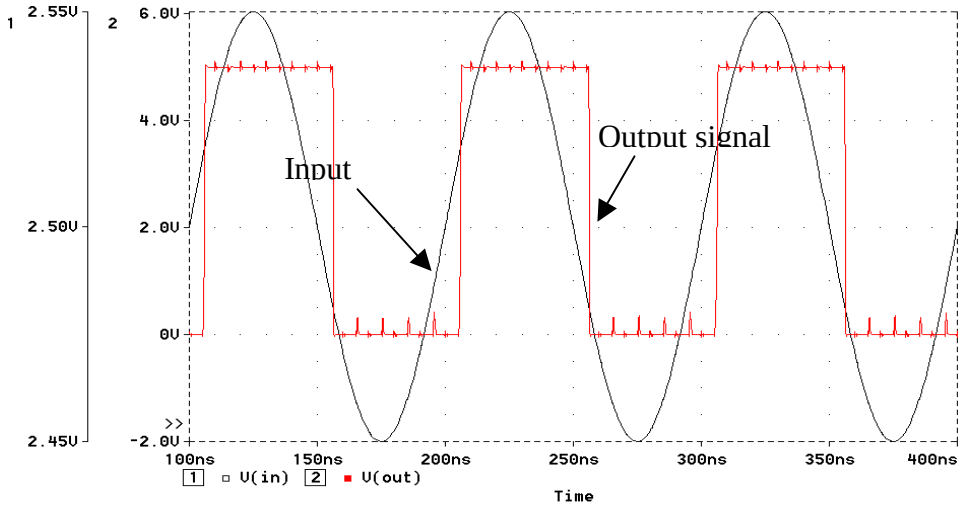
The comparator consists of a differential input pair (M_{189}, M_{190}), a CMOS latch circuit, and a S-R latch. The CMOS latch is composed of an n-channel flip-flop (M_{207}, M_{208}) with a pair of n-channel transfer gates (M_{209}, M_{210}) for strobing, an n-channel switch (M_{218}) for resetting, and a p-channel flip-flop (M_6, M_7) with a pair of p-channel precharge transistors (M_{192}, M_{193}). *Set* and *reset* are the two non-overlapping clocks. The dynamic operation of this circuit is divided into a reset time interval and a regeneration time interval.

During ϕ_2, the comparator is in the reset mode. The current flows through the closed resetting switch M_{218}, which forces the previous two logic state voltages to be equalized. After the input stage settles on its decision, a voltage difference is established between nodes *a* and *b* in the end. This voltage acts as an initial imbalance for the following regeneration time interval. In the meantime, as the n-channel flip-flop is reset, the p-channel one is also reset by the two closed precharge transistors which charge nodes *c* and *d* to the positive power supply voltage. As a result, the CMOS latch is set to the astable high-gain mode.

The regeneration is initialized by the opening of switch M_{12}. Since the strobing transistors M_{209} and M_{210} isolate the n-channel flip-flop from the p-channel flip-flop, when *set* is low, the use of two non-overlapping clocks perform the regeneration which is



(a) Sampling clock signal



(b) Input and output signal

Figure 4.15 Comparator simulation results

within the short time slot between *reset* going low and ϕ_1 going high. The second generation step starts when *set* goes high and M_{209} and M_{210} are closed. The n-channel flip-flop together with the p-channel flip-flop regenerates the voltage differences between nodes *a* and *b* and between nodes *c* and *d*. The voltage difference between node *c* and node *d* is soon amplified to a voltage swing nearly equal to the power supply voltage. The following S-R latch is driven to full complementary digital output levels at the end of the regenerative mode and remains in the previous state in the reset mode.

Figure 4.15 shows a simulation result with 100MHz sampling rate. The reference voltage was 2.5V and the input signal swung between 2.45V and 2.55V with a 10 MHz frequency rate. The total power consumption of this circuit was 120 uW and the layout size was 70uX250um; whereby, occupying 28 % of the pixel area.

4.2.6 Overall systems

Figure 4.16 shows the schematic of the combined modulator circuits. The output of the current buffer is connected to the D/A converter and integrator. The detector bias is controlled by the V_{bias} that is connected commonly with other pixels. The current source is also connected commonly with other pixels on the outside of the focal plane to save space and power. The current integrator is implemented with a capacitor whose value is determined by the input current size, readout speed, and noise. The last stage is a comparator, which compares the integrator voltage and reference voltage then makes a one bit output data stream. The digital output of the comparator would be decimated and filtered by the filters that are programmed in the following DSP chip. The comparator

output is also feedback to the DAC to control the feedback current which makes the comparator output average track the input value.

Figure 4.17 shows simulation results with 50kHz 2uA sinusoidal input, and is oversampled by 32. Figure 4.17(a) shows the integrator voltage variation and the modulator output. As it was calculated in previous section, the integrator voltage does not exceed 800mV and did not saturate with maximum input current. The power density function (PDF) of the output code is shown in Figure 4.17(b). As it was explained in sigma-delta modulator properties, the noise was increased with high frequency area and decreased with low frequency area. The out-of-the modulator signal is decimated and low pass filtered to secure the binary code. The decimating and low-pass filtering algorithms are explained in Appendix D.

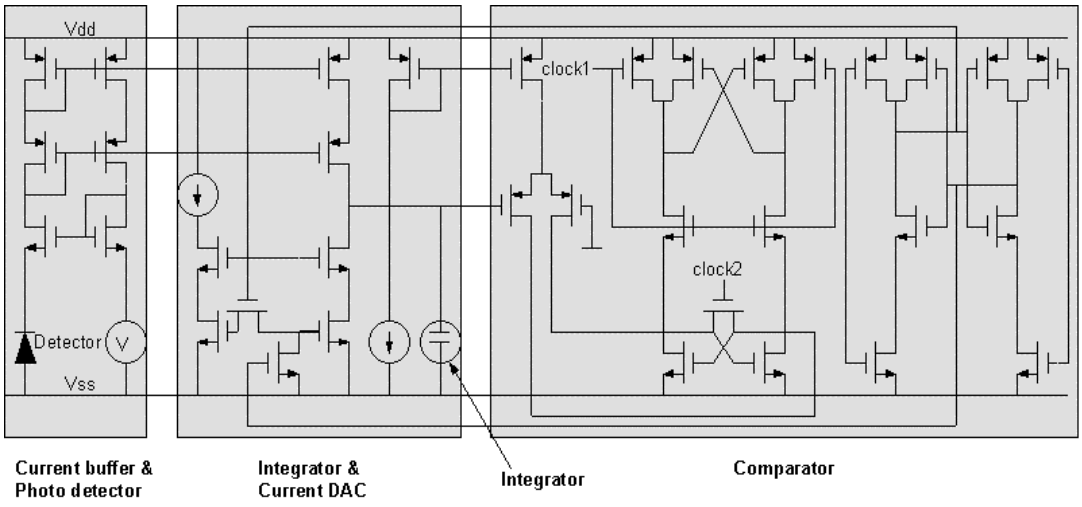
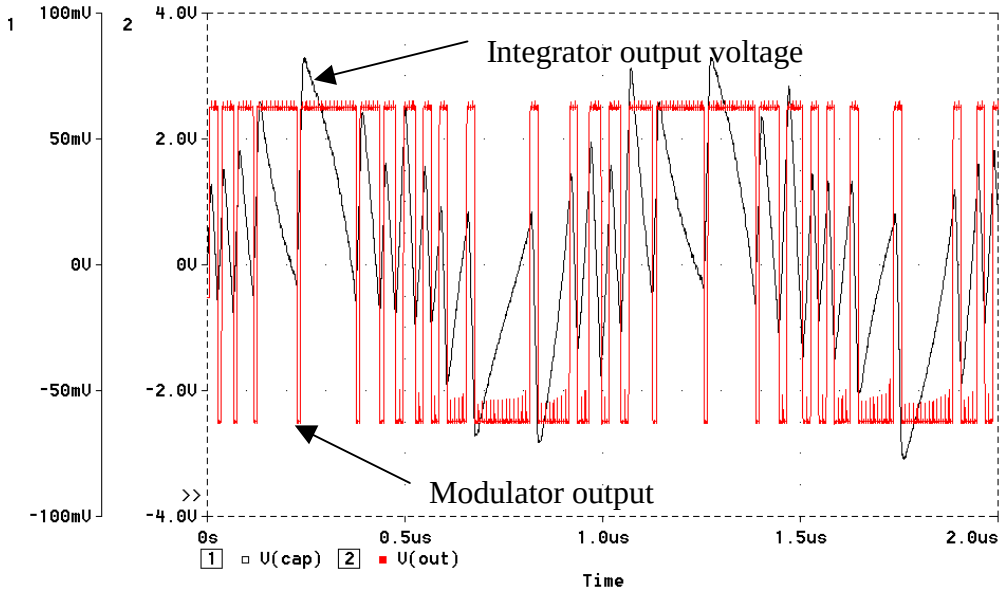
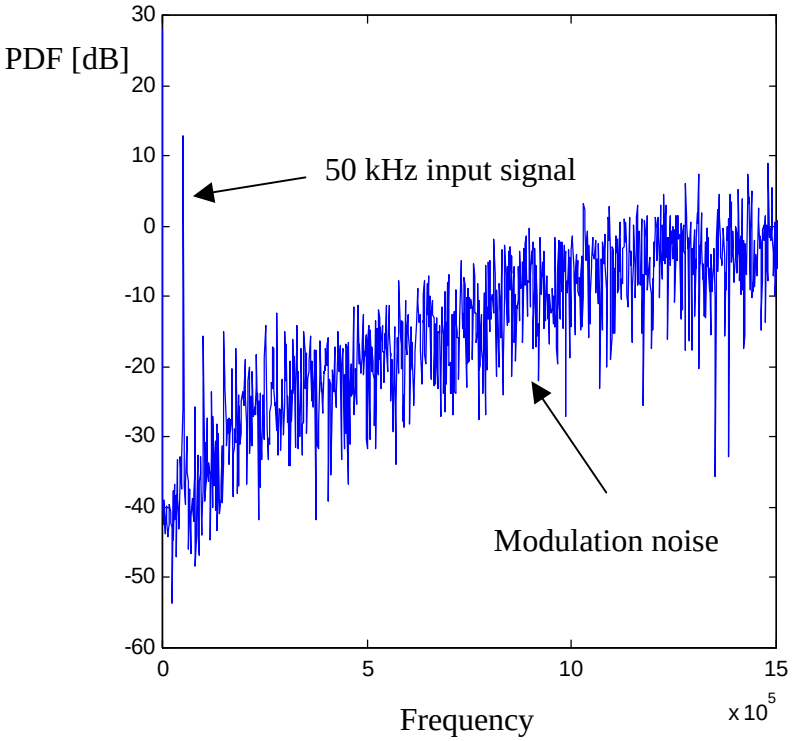


Figure 4.16 First-order oversampling modulator with current buffer



(a) Integrator and output nodes voltage



(b) Power density function

Figure 4.17 First-order oversampling modulator simulation results

4.2.7 Circuit noise attenuation

Figure 4.18 shows the block diagram for first-order oversampling modulator with several noise sources including detector noise and circuit noise. In fact, fixed pattern noise of the detector, threshold offset, and nonlinear gain are equivalent to DC (extremely low frequency content) noise and go to zero when they are differentiated [48]. Furthermore, white noise and *1/f* noise are attenuated by the oversampling technique.

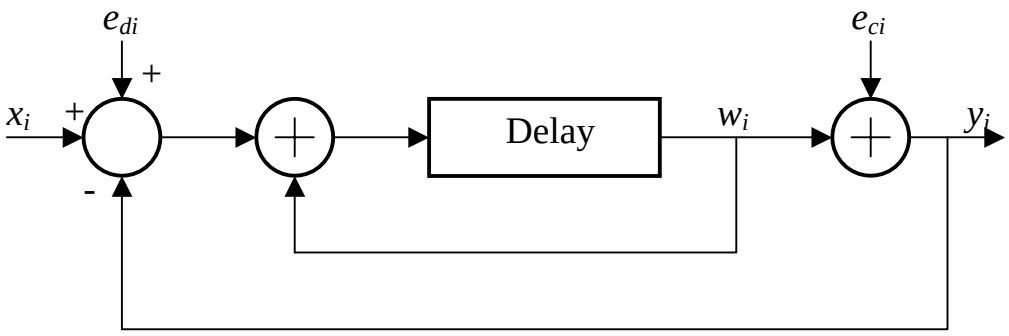


Figure 4.18 Noise sources of modulator

The difference equation describing the above system is:

$$y_i = x_{i-1} + \left(e_{ci} - e_{ci-1}\right) + e_{di-1} \tag{4.9}$$

where, e_{di} = detector, current buffer, and current D/A converter noise and

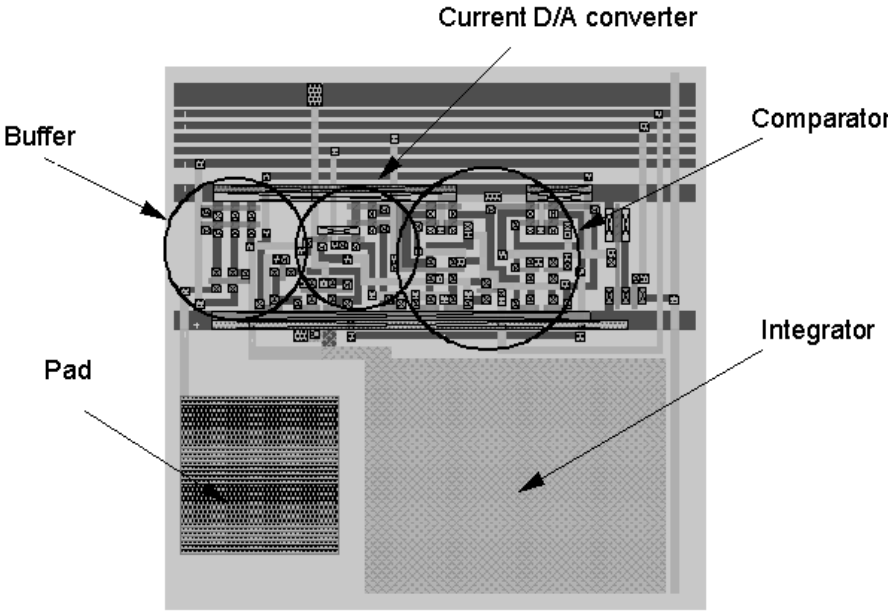
e_{ci} = quantization and comparator noise.

There are two different noise terms in previous equation. The first term will be removed by noise shaping as described by equation (4.4). The second term will be attenuated by oversampling effect as described by equation (4.5). Finally, the quantization noise is reshaped as shown in Figure 4.6 and the other noise should be removed with the filtering.

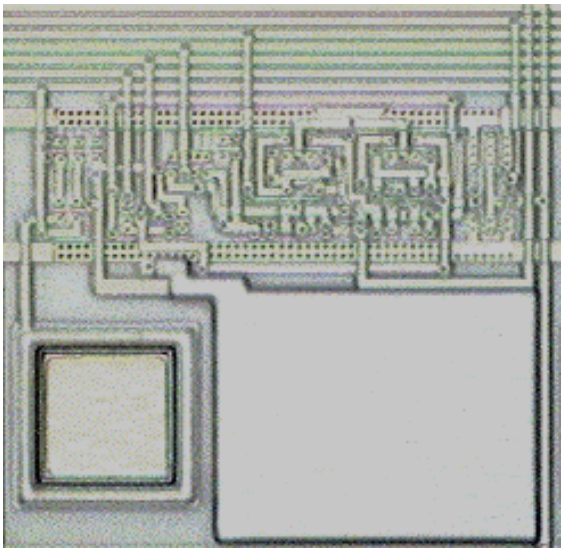
4.2.8 Layout

Figure 4.19 shows two different layouts and photomicrographs of the modulator circuits for FPA pixels. All the circuits including data lines and detector were laid-out to fit into 125umX125um space. To make a large detector, all efforts were applied to design a compact circuit. Input parts of the circuits were carefully designed not to overlapped with digital data lines. To reduce the offset and improve the switching time of the comparator, all the components were carefully laid-out to make a matched comparator. When the capacitor was laid-out, metal 1 and metal 3 layers were connected to the GND to prevent the metal-substrate capacitor. The latch transistor size was optimized to drive a high capacitor load which is connected to several pixels through a long data line.

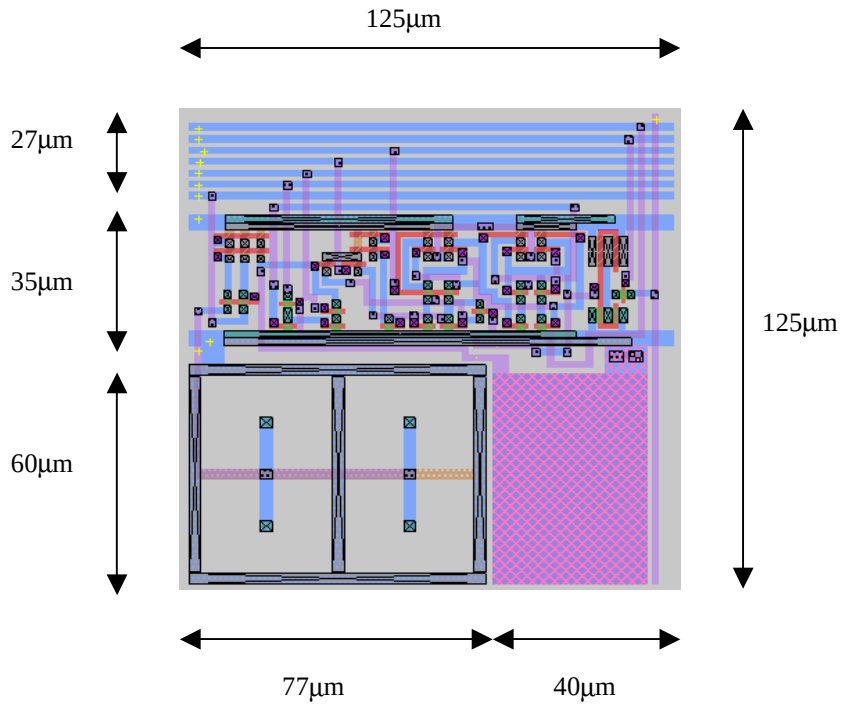
There were two different layouts: one was for the hybrid detector, and the other one was for the monomaterial detector. Hybrid focal plane had a pad to bond the detector (38um X 38um). The monomaterial focal plane had a smaller capacitor (56um X 40um) than the hybrid detector (56um X 72um), and all surfaces except the detector area were covered by metal 3 to prevent unnecessary light induced current through the circuit area.



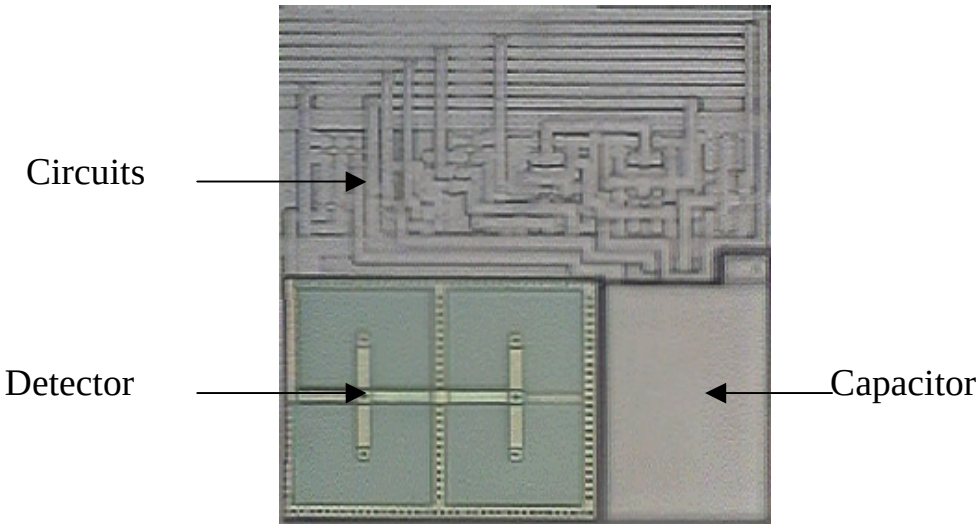
(a) Hybrid detector circuit layout



(b) Hybrid detector circuit photomicrograph



(c) Monomaterial detector layout



(d) Monomaterial circuit photomicrograph

Figure 4.19 Layouts and photomicrographs

CHAPTER 5

PHOTODETECTORS

To integrate the detector imaging array onto the silicon circuitry, there are two options which can be explored. One is a hybrid detector [49], and the other is a monomaterial detector [50]. The hybrid detectors were attractive as a focal-plane-array image sensors until now because of their high responsivity and high fill factor. The other image sensor is a monomaterial detector. The major reasons for the interest in monomaterial detector are related to miniaturized and cost effective imaging systems. CMOS-based image sensors offer the potential opportunity to integrate a significant amount of VLSI electronics on-chip and reduce components and packaging cost. Two kinds of monomaterial detectors are available. In this dissertation, a photodiode was used for the photodetector instead of a photo transistor because it has better linearity characteristic [51]. The problems of monomaterial detectors are their noise level and scalability. The monomaterial pixel does not scale well to a larger array size and a faster pixel readout rate. This is because the bus capacitance and the readout noise is increased. In this dissertation, these kinds of drawbacks were improved by using a current buffer for the front end of the readout system, and by building an ADC per each pixel.

5.1 Hybrid photodetector

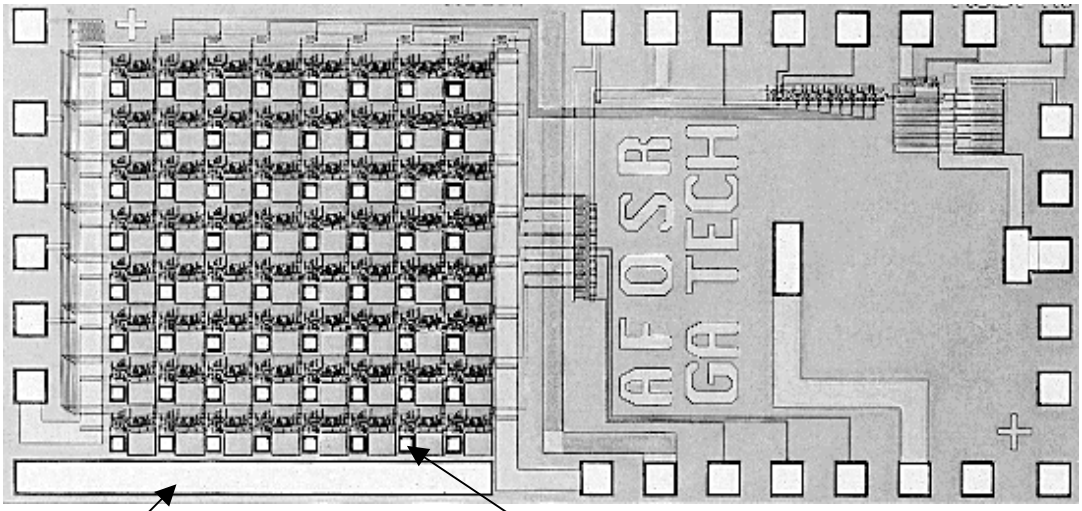
Hybrid integration is an attractive alternative to monomaterial integration for two reasons. First, the material which comprises the detector array is fabricated separately from the electronics. Because of this fact, the detectors need not be the same material as the electronics (Si), which enable the use of higher responsivity materials such as direct gap compound semiconductors. This also implies that the detector materials do not have to be lattice matched to the circuitry materials (for example, direct growth onto the silicon circuitry). The second advantage of hybrid integration is that the detectors are integrated directly on top of the silicon circuitry, which enables scalability with high fill factors and the direct interconnection of every detector to circuitry which lies beneath it. On the other hand, one disadvantage of using flip chip bonding for hybrid integration of detector arrays is that the substrate must be transparent. Thin film bonded detector arrays do not suffer from this limitation since the substrate is removed from the devices [52].

There are two basic types of hybrid detectors which can be integrated with the silicon circuits: P-i-N and MSM (metal-semiconductor-metal) detectors. While MSM detectors are a viable option for integration because they are high speed, low capacitance detectors [52], they have a lower responsivity than the P-i-N detector, and their geometry does not lend itself as well as the P-i-N to array integration. The MSM is a planar device which has two bonding pads in the same plane (the interdigitated fingers of the MSM). To achieve individually addressable pixels using MSMs, the space of the devices has to

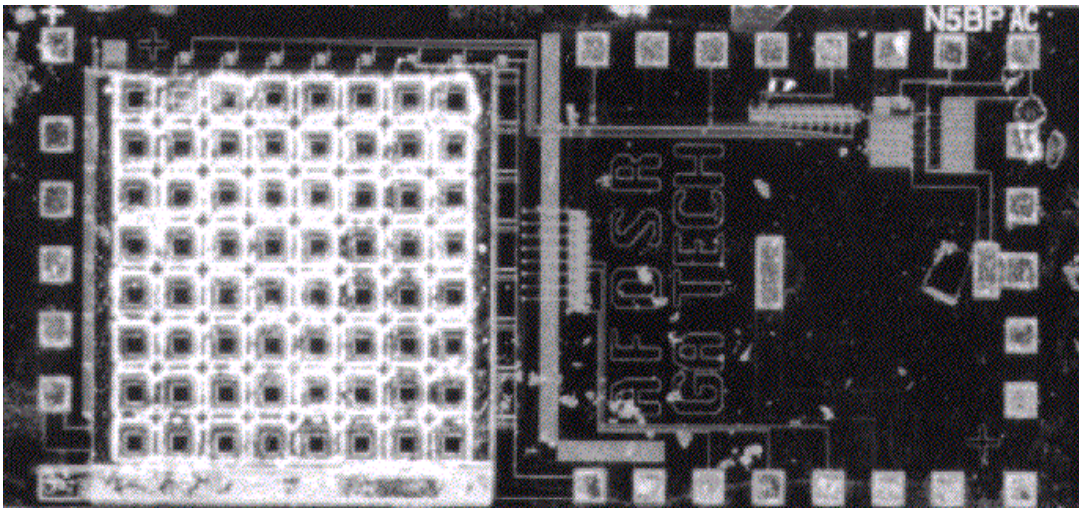
be large enough to allow space for the contact pads, thus decreasing the fill factor. The integration method used with the P-i-N results in a higher fill factor than that of the MSM. This increased fill factor comes from exploiting the non-planarity of the P-i-N structure. The P-i-N device contacts are on the top and bottom of the structure; this is useful in terms of the integration because the pixels can be individually addressed through the device side bonded to the silicon circuit and share a common top contact, eliminating the pad space between detectors. For example, to electrically connect an NxN array of P-i-N detectors only N^2 +1 dedicated pads on the silicon circuit are necessary with N^2 pads located underneath the detector array.

Examples of hybrid integration of compound semiconductor detectors onto silicon circuits include: GaAs MSMs [53] which detect at 850nm, and InP/InGaAs/InAlAs MSMs which detect at 1.3 and 1.55 micron wavelengths. Arrays of GaAs/AlGaAs double heterostructure P-i-N detectors [54] were integrated directly on top of silicon circuits by Suzanne M. Fike who is a student of Dr. Nan M. Jokerst. Figure 5.1(a) is a photomicrograph of the unintegrated silicon circuit as received from the MOSIS foundry. Figure 5.1(b) is a photomicrograph of the GaAs thin film detector array integrated onto the top of the circuitry with the common top contact deposited onto the detector array connecting the array to the silicon circuit. Each detector was individually interconnected to the silicon circuitry, which lies beneath it through a metallized overglass cut on the circuit. Unfortunately, there was no test result of this photodetector because it did not work properly when it was tested. Before it was integrated onto the circuit, the same

photodetector had been integrated onto the other circuit and the test results guaranteed the functionality of this detector [7].



(a) Unintegrated chip



(b) Integrated chip

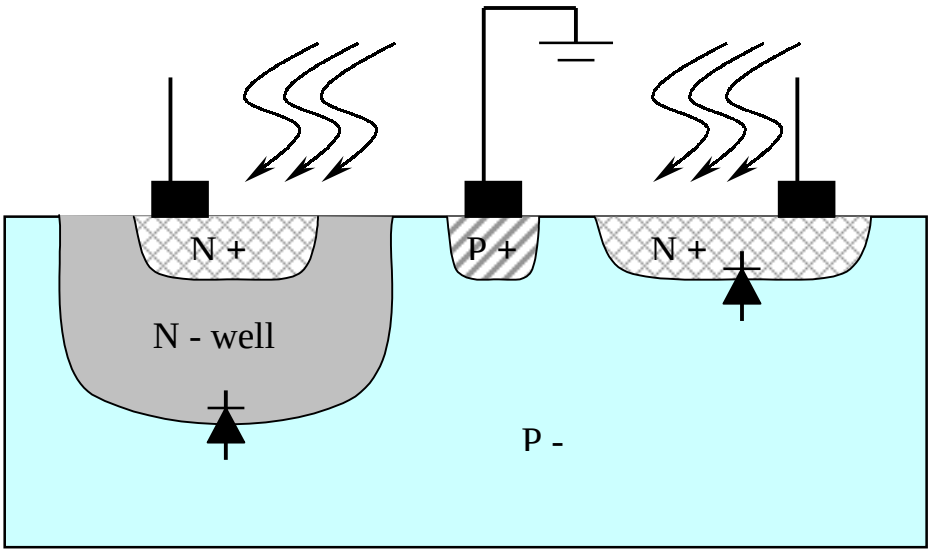
Figure 5.1 Hybrid photodetector photomicrographs

5.2 Monomaterial photodetector

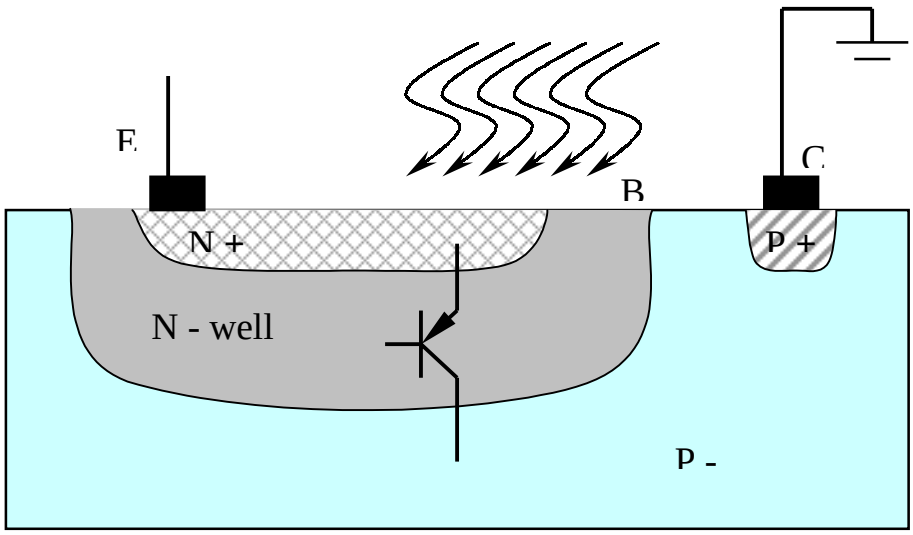
Monomaterial potodetectors using standard “CMOS” technologies have attracted much attention in the past few years [55]. The main advantage of using CMOS is the ability to integrate most or all of a digital imaging system on a single chip, thus reducing cost and power dissipation.

On a standard CMOS processes, basically two sensors can be designed: photodiodes or vertical bipolar phototransistors [56][57]. A trade-off can be found between the area of the sensor, its sensitivity, and its bandwidth. The photodiode has an advantage of better linearity and faster response time; whereas, the phototransistor benefits from higher gain [58]. Two pn-junctions have been mainly utilized to implement photodiodes: the well substrate, and the well-diffused junctions as shown in Figure 5.2(a). The well-substrate photodiode has the best responsivity due to a wide depletion region, and because it is able to collect the minority carriers photogenerated deeply in the substrate provided that they are generated within the diffusion length of the minority carriers. The well-substrate photodiode also has the lowest capacitance, which helps to achieve a high bandwidth. However, the disadvantages of the well-substrate photodiode are its sensitivity to substrate noise and crosstalk from the neighboring photodiodes due to the long diffusion length of the carriers [59][60].

Vertical bipolar transistors are made using drain-well as emitter-base junction and well-bulk as base-collector junction show in Figure 5.2(b) [61].



(a) Photodiodes

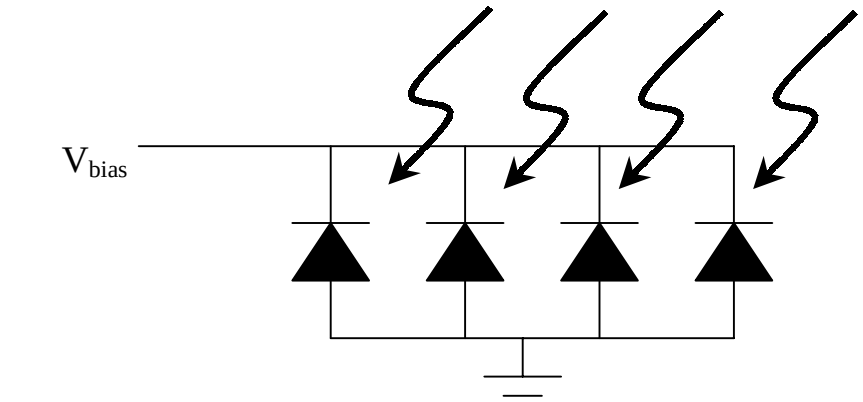


(b) Phototransistor

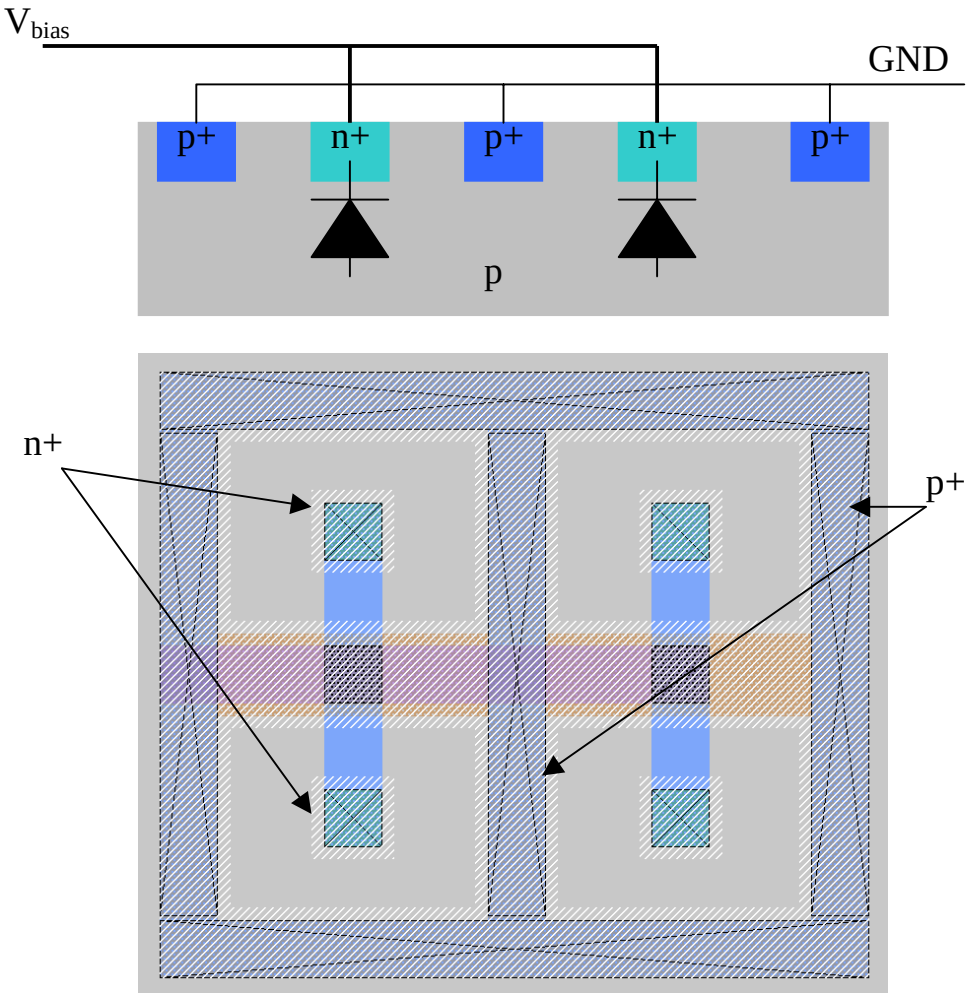
Figure 5.2 Monomaterial photodetectors

The photo current I_B generated in the base-collector photodiode is amplified by the transistor gain β to produce a collector current $I_C = \beta I_B$. The lateral transistor provides a higher β but has a complicated structure and large device-to-device variations, both of which can be problems to array-type detector implementation and performance. The vertical parasitic PNP bipolar transistor in the n-well CMOS process provide high compactness, moderate gain, and speed. Using a special layout, over one hundred gains can be achieved with this structure [62].

In this dissertation, a photodiode was used for the photosensor. The photodetector was originally designed by Jinsung Park who is a student of Dr. Martin A. Brooke. The only modification was to make it small to fit into the already decided standard pixel. The implemented photosensor structures for the FPA system is presented in Figure 5.3. The photodiodes were realized using the standard 0.8μm n-well CMOS process. The physical available size of the photodiode was restricted to 60μmX77μm, because the pixel space was shared by the other circuits and data lines. This phtodiode could not be a optimal photosensor for the FPA system because it was designed for a high speed communicaiton receiver. To make a high speed photodetector, it was designed with a parallel methodology. Four parallel photodiodes were designed rather than one big photodiode to reduce the parasitic capacitor as shown in Figure 5.3. By using this layout skill, it was possible to increase the bandwidth of the photodiode.



(a) Schematic



(b) Layout

Figure 5.3 Designed monomaterial photodetector

CHAPTER 6

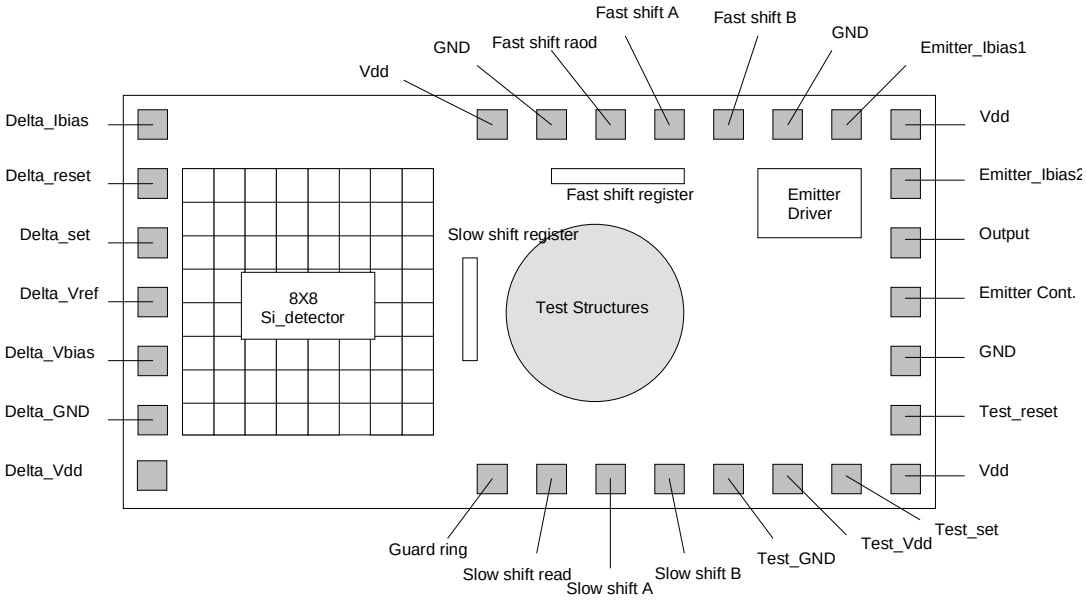
TEST

To verify the functionality of the system, several tests were completed. There were also several steps to prepare for testing. The first step was to design a board to generate all clock signals and bias voltages to supply the test chip. Eight different clock signals and three different bias voltages were required for testing. For low speed testing a board was built to generate clock signals, and for high speed testing an Arbitrary Waveform Generator (AWG2041) was used. The next step was to select a suitable package for the chip. The chip was wire bonded to the 40 pin dip package because it was the most convenient package for the testing, and the maximum frequency for the testing was not over than 100 MHz. After the board and wire bonding chip were ready, electrical testing was scheduled and completed. The electrical test was performed with different DC input currents. The detector was analyzed with a low speed testing environment. Another serial-to-parallel board and data acquisition board were used to import the FPA output data into a PC. A C program was written for downloading the data. Linearity, uniformity, and system noise were measured by using the collected data. For

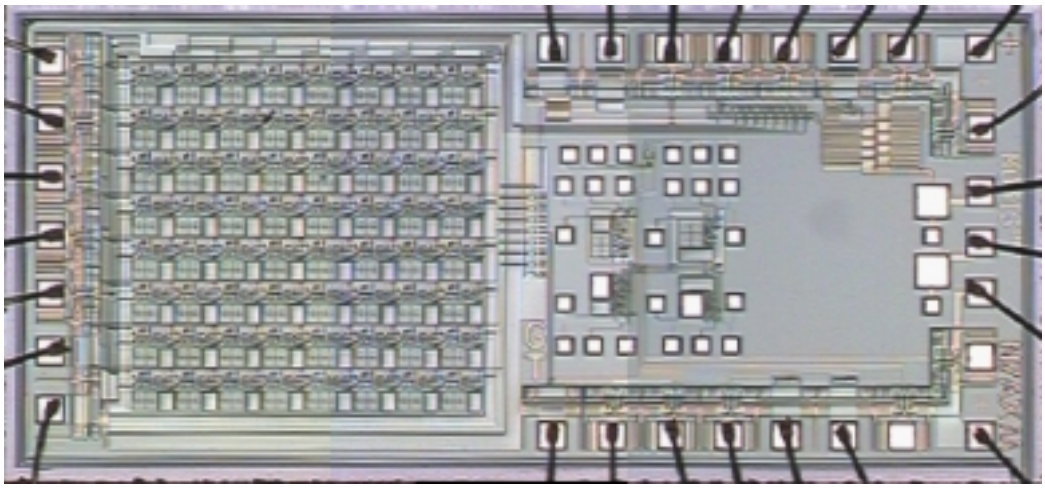
high speed testing, the Arbitrary Waveform Generator and Transient Capture Oscilloscope (Tektronix 11403A) were used. The FPA chip will be stacked on a DSP chip and tested together for completion in the future. In following sections, the test procedure and test results are explained in detail.

6.1 Test setup

Figure 6.1 shows a pin diagram and a photomicrograph of the test chip. Three non-overlapping signals were used for the oversampling modulator and shift registers. Power supplies were separated to reduce the digital noise. Reference voltage (Delta_V_{ref}) was used to provide the switching point for the comparator, and bias voltage (Delta_V_{bias}) was used for the detector biasing. One non-overlapping clock signal was used for the oversampling modulator, and one bit output data was generated at the rising edge of the set signal (Delta_set). The generated data was available during the next 64 cycles. During the first 8 of 64 cycles, the first row of the FPAs was selected by the slow shift register then readout to the digital amplifier. The amplified data was then serialized by the parallel-to-serial shift register. The digital output of the register was changed to a current, through the emitter driver, to drive the integrated emitter. Table 6.1 describes pin descriptions of the chip. To reduce the digital noise, two kinds of technique were applied. One was to use a guard ring to capture the digital noise passing through the substrate and the other was to use separate power supply for digital and analog circuits. Figure 6.2



(a) Pin diagram



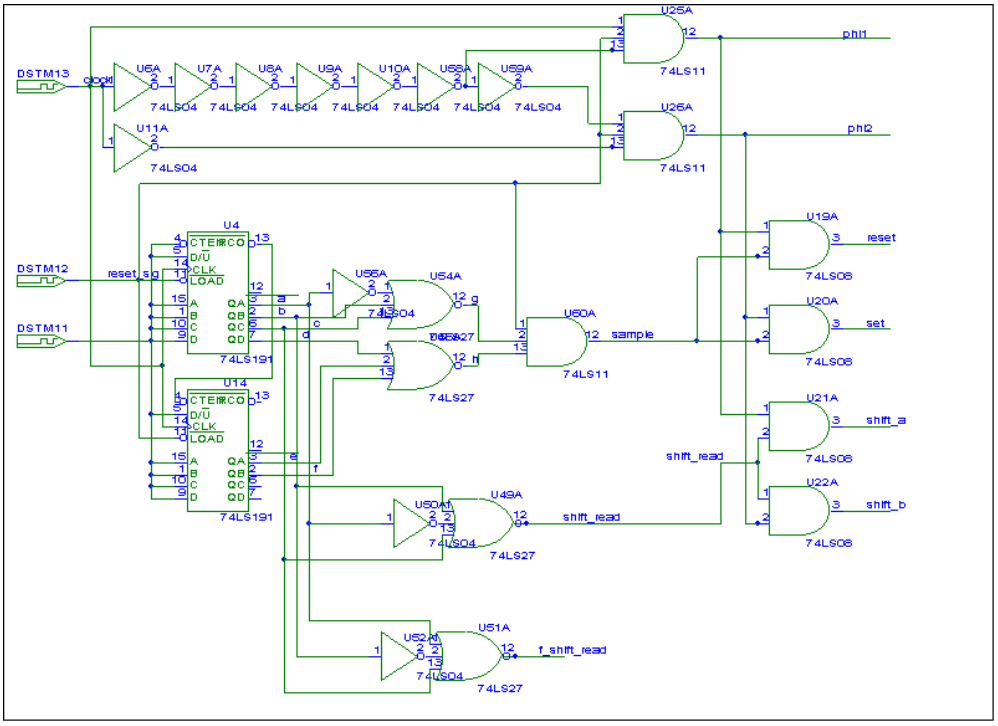
(b) Photomicrography

Figure 6.1 Pin diagram and photomicrograph

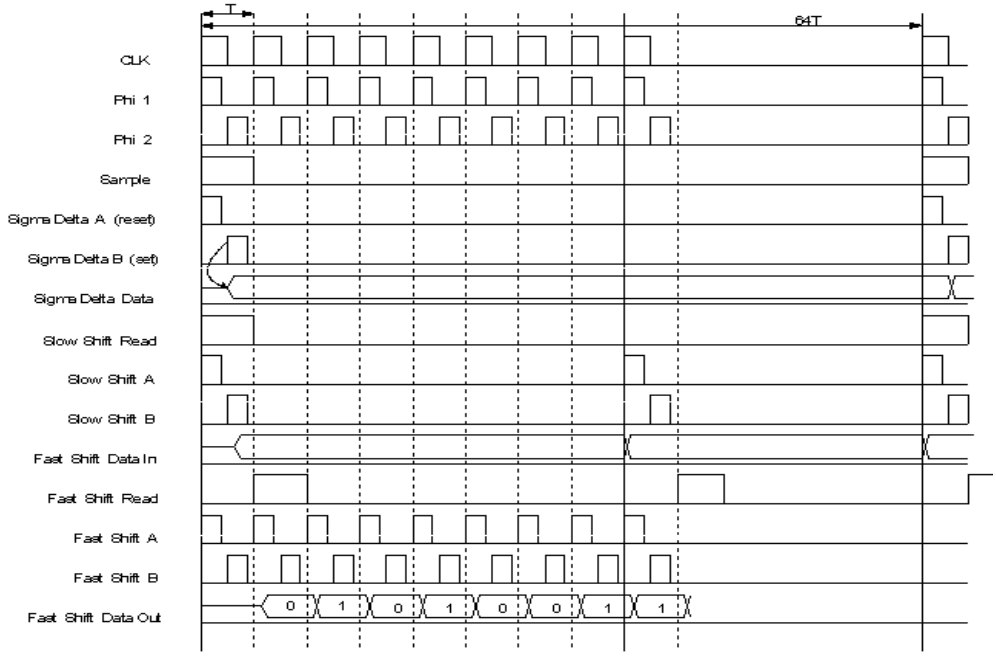
Table 6.1 Pin description

Pin Name	Specification
Vdd	Emitter driver and shift registers power +5V DC 0.1 uF and 0.01 uF ceramic disk capacitor decoupling to GND pin <100 mA current
GND	Emitter driver and shift registers power 0V DC <100 mA current
Delta_Vdd	Sigma-delta modulator power +5V DC 0.1 uF ceramic disk capacitor decoupling to Delta_GND < 300uA current
Delta_GND	Sigma-delta modulator power 0V DC < 300uA current
Delta_Ibias	Bias current for the current mirror of Sigma-delta circuit > 64 × Iin (input current)
Delta_Vbias	Bias voltage for photo detector Low voltage :1V High Voltage :2.5V
Delta_Vref	Reference voltage for the comparator Low voltage :0V High Voltage :2.5V
Top contact	Bias voltage for photo detector 0V DC
Delta_set	Nonoverlapping clock signal with Delta_reset signal 50% duty cycle +- 1ns, Frequency < 10 MHz , > 10 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Delta_reset	Nonoverlapping clock signal with Delta_set signal 50% duty cycle +- 1ns, Frequency < 10 MHz , > 10 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Emitter_Ibias	Bias current for the Emitter driver circuit 200uA ~ 1mA
Fast shift_road	Latches Data on rising edge of Fast shift_road should not have falling edge within Fast shift_A rising edge

	Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Fast shift_A	Nonoverlapping clock with Fast shift_B 50% duty cycle +- 1ns, Frequency < 10 MHz , > 150 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Fast shift_B	Nonoverlapping clock with Fast shift_A 50% duty cycle +- 1ns, Frequency < 10 MHz , > 150 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Slow shift_in	Latches Data on rising edge of Slow shift_A Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Slow shift_A	Nonoverlapping clock with Slow shift_B 50% duty cycle +- 1ns, Frequency < 10 MHz , > 50 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Slow shift_B	Nonoverlapping clock with Slow shift_A 50% duty cycle +- 1ns, Frequency < 10 MHz , > 50 kHz Low voltage < 0.5V High Voltage > 3.5V Capacitance < 10 pF
Guard ring	Isolate analog part from digital noise 0V DC



(a) Schematic



(b) Timing diagram

Figure 6.2 Clock generator schematic and timing diagram

shows the schematic and the timing diagram of the clock generator that was built for low speed testing. Another consideration of the testing procedure was to protect static sensitive gates from static damage. All the sensitive pads were protected by internal protection pads and external protection circuits as follows.

To prevent static damage, a conventional protection circuit was used for the sensitive pads [63]. A 6000V static voltage can be easily generated and damage the transistor gate which can be withstand 800v/μm. In 0.8μm technology, the gate oxide thickness is 17.4nm according to the MOSIS data sheet; therefore, the maximum withstanding voltage of the gate is 13.52V for an 0.8um CMOS technology. The protection circuit is designed with 1kΩ, 1MΩ registers and one zenor diode to restrict the gate voltage under the breakdown voltage as shown in Figure 6.3.

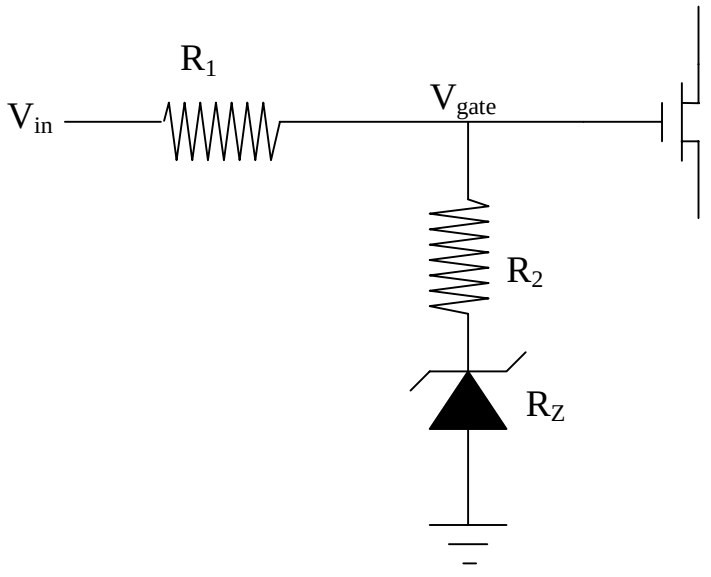


Figure 6.3 Pad protection circuit

The impedance of the zenor diode (R_Z) is between 5 to 500 Ω when it is turned on; therefore, the maximum transistor gate voltage (V_{gate}) is calculated by following equation.

$$V_{gate} = \left(\frac{R_2 + R_Z}{R_1 + R_2 + R_Z} \right) \times V_{in}$$

$$= \left(\frac{1 + 0.5}{100 + 1 + 0.5} \right) \times 6000 = 9V \ . \qquad (6.1)$$

By using the described pad protection circuit, the gate voltage is restricted under 9V with 6000V static voltage.

6.2 Electrical testing

To verify the functionality of the circuit, the chip was tested electrically first. For the electrical testing, one DC current source (Keithley 236 Source Measurement Unit), probe station, and oscilloscope (Tektronix 11403A) were used. One of the pixels was chosen from several different places on the arrays, then the input pad was probed to provide a DC current. A Keithley source measurement unit was used to supply a precise input current. The input node voltage was also monitored to follow the bias voltage, while the test was in progress, by watching the indicator of the Keithley unit. By adjusting the input current amplitude, or the V_{ref} voltage of the comparator, the

oscilloscope displayed the output signal change. Figure 6.4 shows the test setup with a Tektronix oscilloscope, a Keithley source measurement unit, and a Probe station. The output was measured at the output pad of the emitter driver. It was tested with four different DC current inputs. As test results are shown in Figure 6.5, the output signal changing speed was increased with larger input current because it took less time to charge the capacitor when the input current was increased. Table 6.2 shows the testing environment including bias voltages and input current frequency.

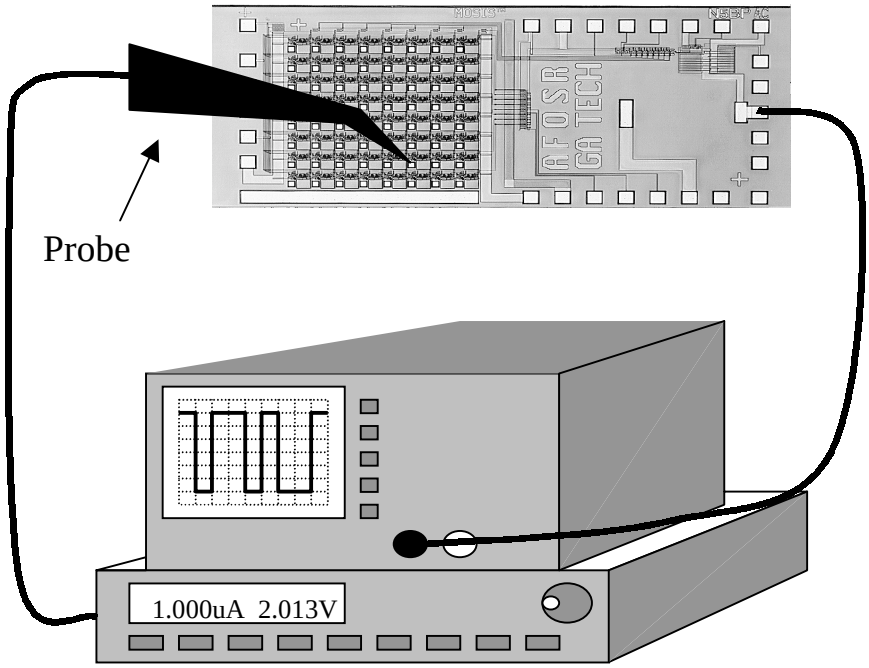
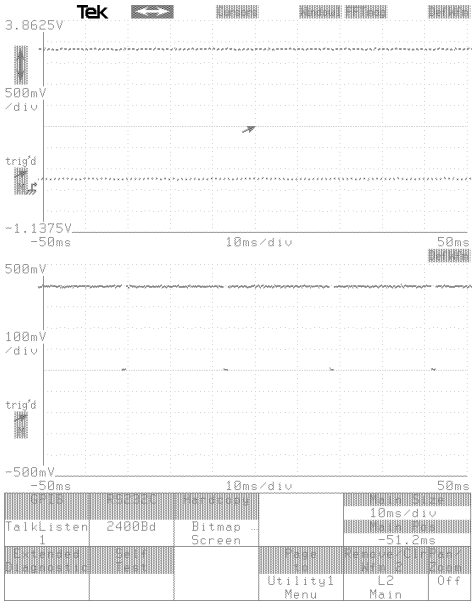
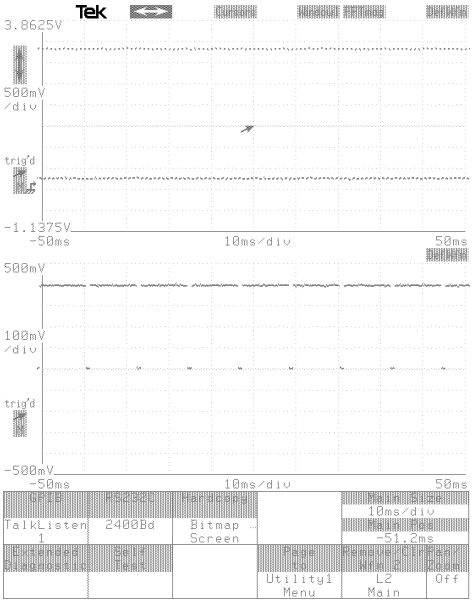


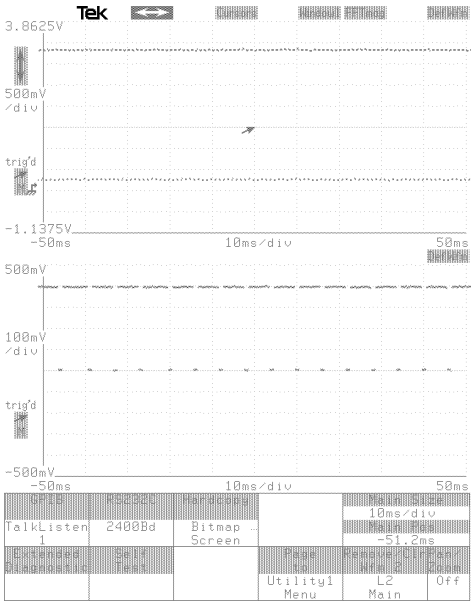
Figure 6.4 Electrical test setup



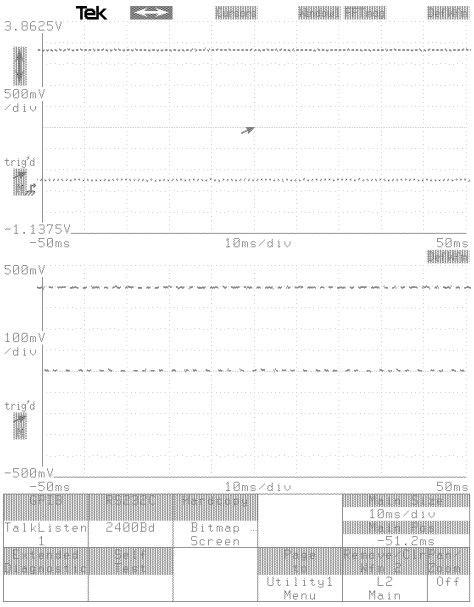
(a) 0.03 μA



(b) 0.06 μA



(c) 0.12 μA



(d) 0.24 μA

Figure 6.5 Electrical test results

Table 6.2 Test values

Name	Value	Compliance
Delta_Vdd	2.5V	0.1145mA
Emitter_Ibias	0.25mA	1.034V
Delta_Ibias	1uA	1.109V
Iin	-0.04uA	1V
Vdd	4V	80mA
Frequency	150kHz	NA

6.3 Slow speed testing

Characteristics of the photodetector such as linearity, uniformity, and system noise were tested using a slow speed test setup. For slow speed testing, a data acquisition card and a PC were used to store all generated data. The speed of this test setup was limited by the bandwidth of the data acquisition card, and by using a low cost digital data acquisition card, the maximum data acquisition rate was limited to 1MHz. For this test a C program, written by Keeshik Chung who is a student of Dr. Scott Wills, was used.

The test circuit block diagram is shown in Figure 6.6. A digital data acquisition card, model CYDIO192T from Cyber Research Inc., was plugged into one of PC's ISA slot. Only three ports were used: port A to read data from FPA chip, port B to synchronize the data being read, and port C to carry reset signal generated by PC. Port A and port B were 8bit wide and were used as inputs. The data bit stream coming out of

FPA had to be converted to an 8bit word. The 8bit-shift register converted the serial data to parallel data, while the octal latch held its value during the next eight clock cycles. This data was read by the PC during this eight clock cycles using Port A.

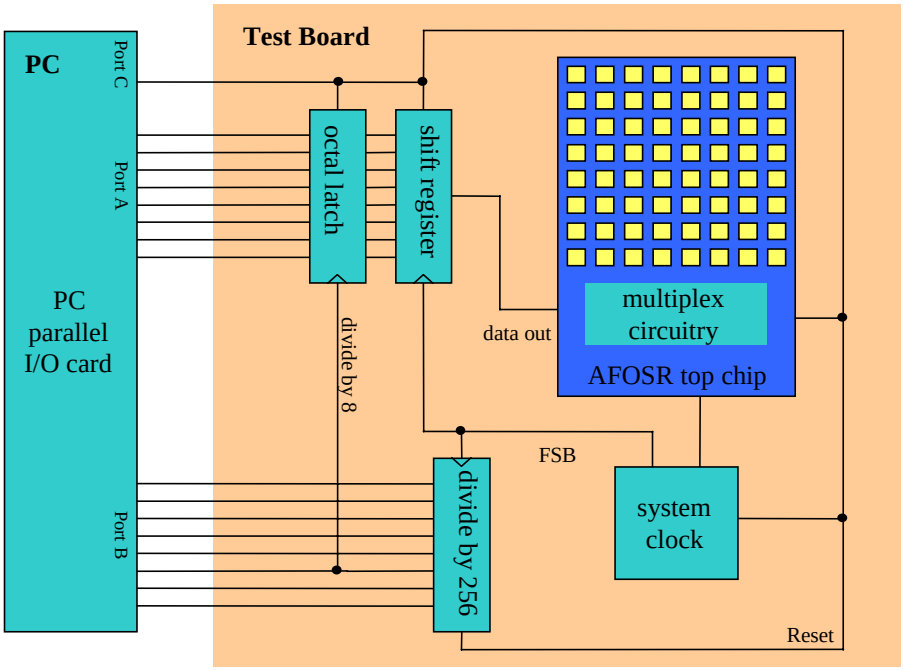


Figure 6.6 Slow speed test setup block diagram

The synchronization information was extracted from the system clock by using an 8bit-counter (divide by 256). A divided by 8 output from this 8bit-counter was used periodically to refresh the octal latch. The data could be synchronized by reading port B regularly and checking its value. Data transmission from the FPA chip to the PC was asynchronous; therefore, there was a need to know if data being read was valid or not. If the data reading rate was too fast, the same data could be read multiple times, or if it was

slower than the actual data stream, some data could be lost. In order to be sure that the data was correct (synchronization), an 8bit-counter was used. The data synchronization was done by reading a byte from port B. As shown in Figure 6.7, the data order and the data timing were extracted from these data points. Using data order the data could be traced and the order could be checked.

When data order value was incremented, it suggested a new datum was present in port A and had to be read. There was a delay from the moment when a datum was presented in the port A and read from it. Besides, during this time a new datum could be arriving and overwritten in port A. Data timing was used to discern how long data had been sitting in port A, and decide whether to read it or abort it because it could read a wrong data. Data timing values range from 0 to 7 clock cycles; whereby, a zero value indicates that data had just arrived, and a value equal to seven indicates that new data would overwrite at the next cycle.

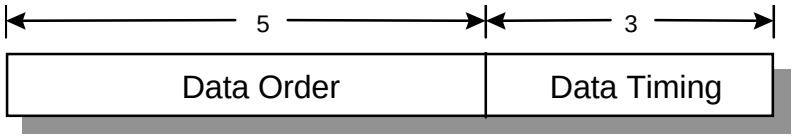


Figure 6.7 Synchronization Data Format

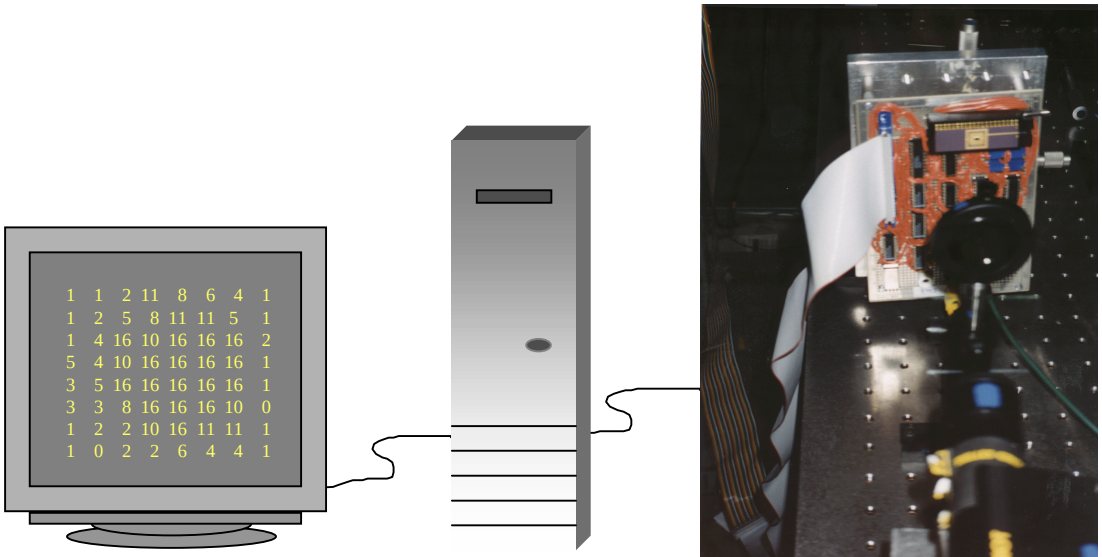
The software was written in Borland C++, and assembler, as described in Appendix E. It permitted the pixels to display what the FPA detects in gray scales. The

number of gray levels was determine by number levels. This program began by sending out a reset signal through port C and initializing all variables. If there was an out of synchronization event, a new set of reading was then initialized and disregarded the data read so far. This successful set of reading was repeated for number samples times. Finally, all successful reading data was computed to get an average for each 8x8 arrays, then displayed on the display.

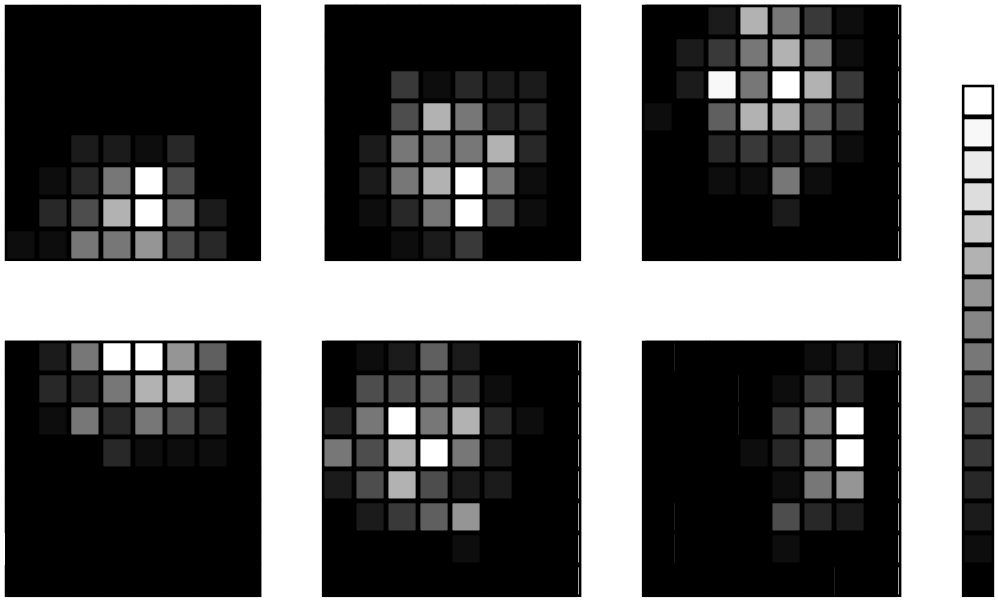
6.3.1 Functional testing

To verify that the entire system was working properly, a test image with a laser source was used to project the image on the FPAs. The dot image on the FPA chip was moving around by changing the chip position. This test was done with two different microprocessors; whereby, one was a 50 MHz 486 processor and the other was a 233MHz Pentium processor. With the 486 processor, the data acquisition rate was limited to 200 kHz. It was increased up to 1 MHz with Pentium processor, which meant that the data acquisition rate was limited not by the FPA readout system but by the test setup.

Figure 6.8(a) shows the test setup. In the right side picture, laser, lens, and a patterned image are shown. The chip was free for the x, y, and z direction and changed the position by rotating the knobs on each direction. The output was displayed on the monitor subsequent to averaging the output data. Figure 6.8(b) shows a 4 bits image with a 1 MHz sampling rate. The actual data on the screen were numbers as shown in Figure 6.8(a). For convenience, the output numbers were transformed to the correspond image



(a) Test setup



(b) Screen image

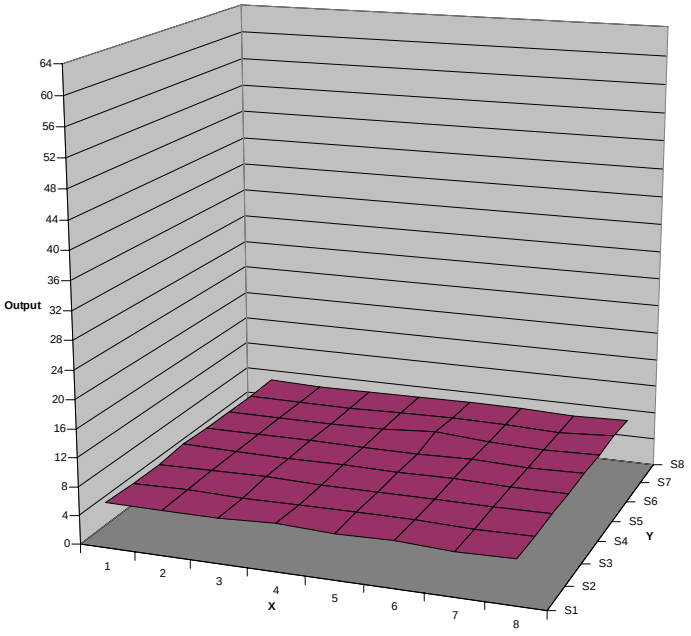
Figure 6.8 FPA functional test

The brightest pixel was colored with white, and the darkest pixel was colored with black. by hand.

6.3.2 Uniformity

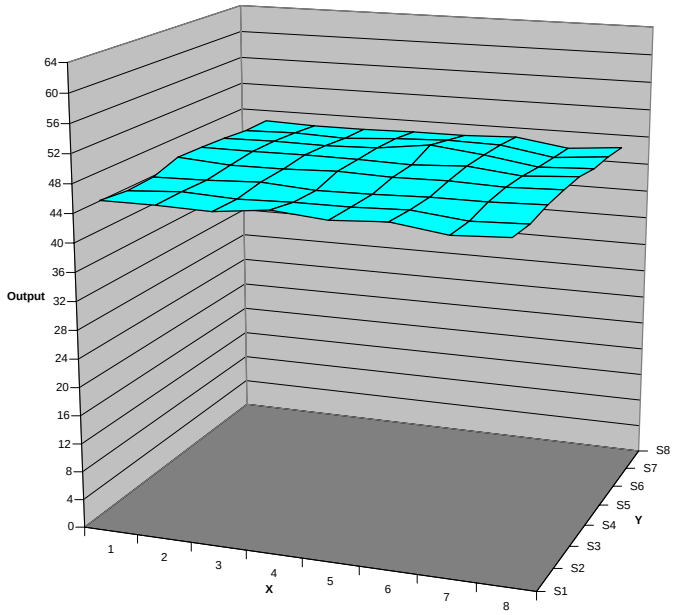
Uniformity is one of the important characteristics of the FPA system. To get a good quality image, all detectors needed to be within a certain amount of variation. For the testing, a uniform light was generated by halogen lamp that was located about 10 ft away from the test chip. The halogen lamp was powered by a 12V DC battery, and the light was filtered by a Newport coated ND filter; whereas, not to saturate the FPAs. The readout system was running under a 1 MHz system clock frequency, and the maximum data value was set to 64.

Figure 6.9 shows the test results of the FPA's uniformity. Standard deviation among the pixels was calculated to measure the uniformity. Twelve different light intensities were used to measure the uniformity; whereas, most of them showed a good standard deviations. The light intensity was controlled to generate an output between 0 to 64; however, the last test result in Table 6.3 was not acceptable because the circuit was saturated. The test results were including all noise sources from the detector and the circuits. Table 6.3 shows the standard deviation for 12 different light intensities. Light intensity meant the transparent light percentage from the original light. When the transparent light was more than 15 %, the output was saturated, and the value did not represent the correct standard deviation. According to the test results, the standard deviation was decreased with more lights.



Maximum : 6.90
Minimum : 6.05
Average : 6.31
Standard deviation :
0.1365 (2.165%)

(a) Low light intensity



Maximum : 47.95
Minimum : 45.48
Average : 46.39
Standard deviation :
0.4741 (1.022%)

(a) High light intensity

Figure 6.9 Uniformity test results

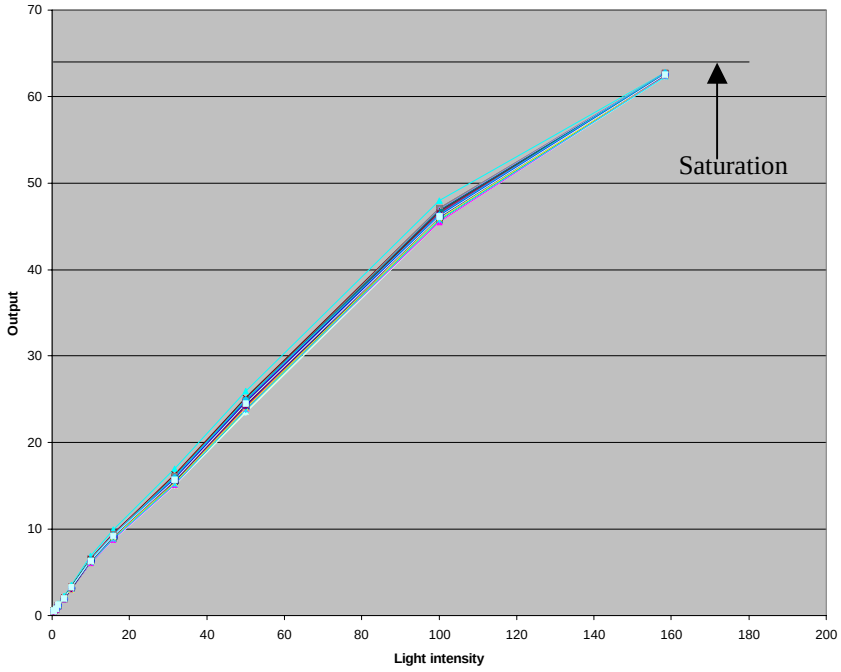
Table 6.3 Uniformity with different light intensities

Light Intensity	Maximum	Minimum	Average	Standard Deviation	%
0.0316%	0.62	0.48	0.52	0.0217	4.19
0.0501%	0.66	0.52	0.56	0.0225	4.04
0.1%	0.96	0.74	0.79	0.0343	4.33
0.158%	1.38	1.12	1.18	0.0390	3.30
0.316%	2.22	1.85	1.95	0.0579	2.97
0.501%	3.55	3.02	3.18	0.0854	2.69
1%	6.90	6.05	6.31	0.1365	2.17
1.583%	9.90	8.74	9.11	0.1965	2.16
3.16%	16.97	15.02	15.68	0.3385	2.16
5.01%	25.95	23.46	24.44	0.4273	1.75
10%	47.95	45.48	46.39	0.4741	1.02
*15.83%	62.83	62.29	62.56	0.1073	0.17

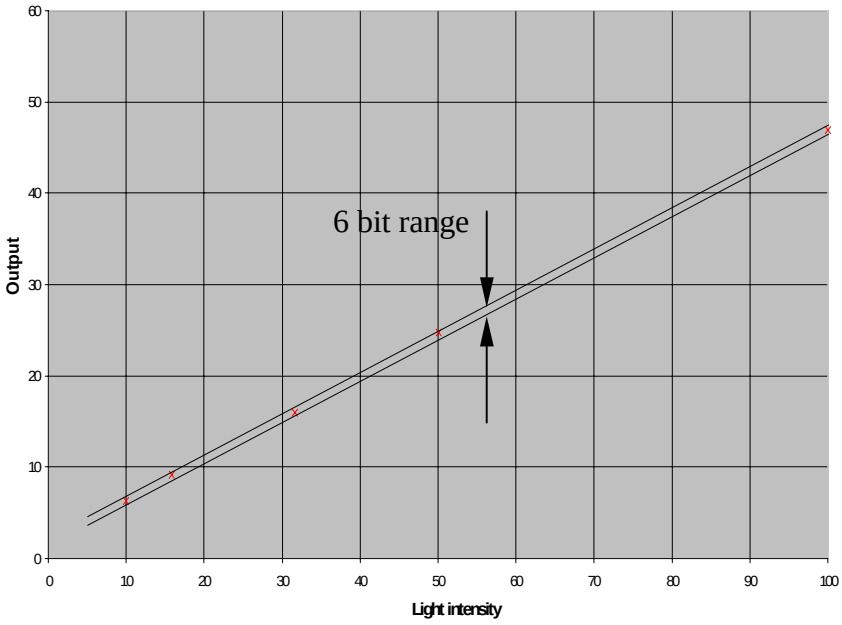
* Saturated

6.3.3 Linearity

Photodetector linearity was tested with four different optical filters (Newport coated N.D. filters). By combining the four optical filters (FBR –ND 03, 05, 10, 20), 16 different light intensities were attained. However, the circuit was saturated with 5 out of 16 light intensities; therefore, they were removed from the evaluation. Figure 6.10(a) shows the linearity of the 64 pixels, and (b) shows 6 bits resolution for one of them. This test setup was not for real time processing; whereby, the PC limits the output resolution less than 6 bits.



(a) 64 pixels data



(a) 6 bits linearity

Figure 6.10 Linearity test results

The nonlinearity of the small light intensity output was not coming from the FPA system but from the optical filters. Figure 6.11 shows another test result which was directly measuring the light intensity using a Multi-Function Optical Meter (Newport 1835-c). For the comparison with the FPA system output, the output of one pixel is shown on the same graph. For this test, three different filters were used; whereby, different filtering effect were made by combining the filters. Figure 6.11 shows the test results by following the measured light intensity trace. This means that the nonlinearity is not caused by the FPA system.

The other nonlinear area of the high light intensity was coming from the saturation of the system.

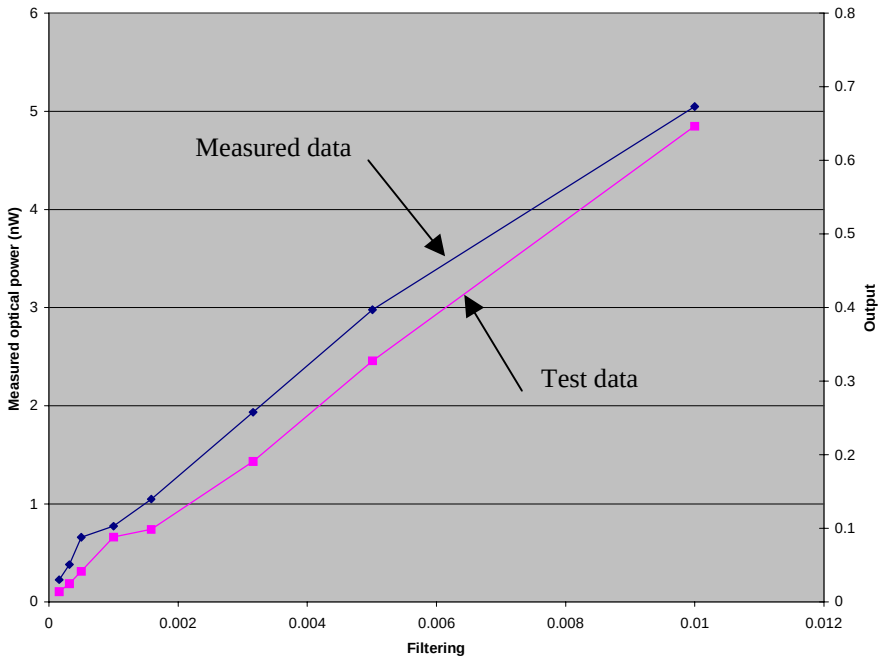


Figure 6.11 Nonlinear test result

6.3 High speed testing

To achieve a 100kHz frame rate with the proposed readout system, it needs to operate with a 167MHz clock speed. In this case, each A/D modulator under the pixel was operated at 2.6MHz to generate an 8 bit image. This oversampling frequency was calculated from the oversampling ratio of 26 to get an 8 bit image. The highest available clock speed using the low speed test setup was 1MHz. The limitation was coming from the PC based data acquisition system. The other limitation was the clock generator circuits. The maximum frequency for the clock generator circuit was 5MHz which was not enough for high speed testing. To overcome these limitations, the test setup was changed. Eight different clock signals were directly generated from the Arbitrary waveform generator (AWG 2041). The output data was read into the transient capture digitizing oscilloscope then readout from the displayed image.

Another high speed test setup was built to measure the error generated by the detectors and circuits. Each control signal was generated by Data generator (DG2020A, Tektronix) and the output data from the test chip was read into the Logic analyzer (TLA704, Tektronix) for the analysis.

6.4.1 Functional testing

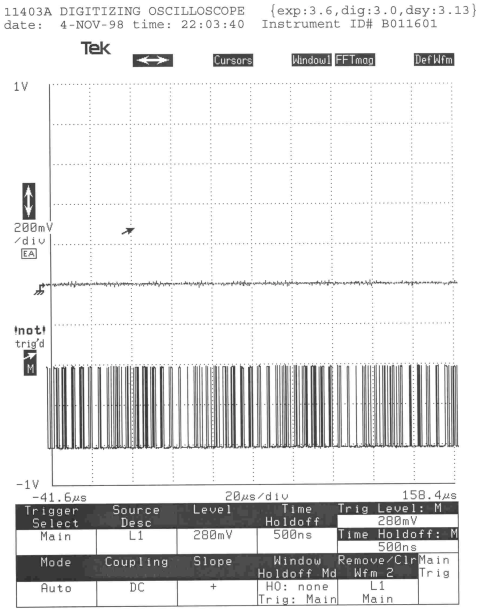
For the first step of high speed testing, a single test pixel was tested at several different clock frequencies. The test pixel output was designed with a small inverter which could not drive a high capacitor load; however, the test results showed the functionality of the circuit with a high frequency sampling rate.

Figure 6.12 shows the output changing with room light intensity. It was tested using a 250kHz system clock speed. Figure 6.12(a) is the output without any light. When the room light was increased there were several pulses which were not periodic, as shown in Figure 6.12(b). The output did not show a clear image because the sampling oscilloscope (Tektronix 11403A) was used to measure the output. When the room light intensity was increased Figure (c) and (d) were obtained.

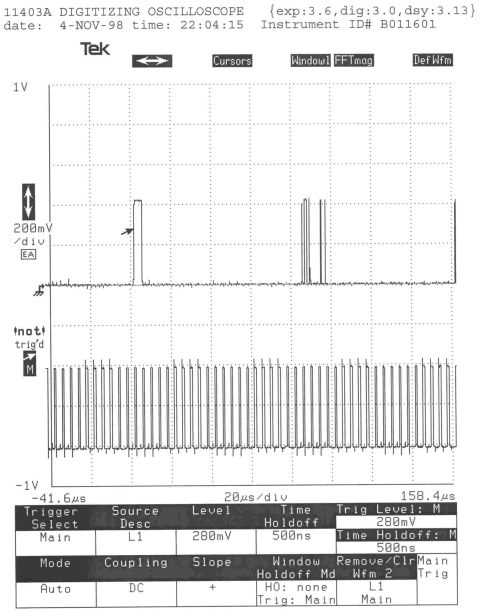
Figure 6.13 shows a maximum sampling rate for the pixel. Figure 6.13 (a) shows the test results using a 250kHz sampling rate for the test structure; whereby, clear images are seen using this medium speed sampling rate. Figure 6.13(b) shows the test results using a 1MHz sampling rate while still displaying a good output waveform. When the sampling rate was increased greater than 1MHz, it became difficult for the output inverter to drive the output load. Figure 6.13 (c) and (d) were the outputs using 2MHz and 4MHz sampling rates. To get a 100kfps 8 bit image with an 8x8 FPAs required a sampling rate of between 2MHz and 4MHz. The maximum sampling with a varied output was 4MHz using this test environment. The actual FPA system has less load capacitance compared

to the test structure. It also has a digital amplifier at the end of the column which promised a larger bandwidth.

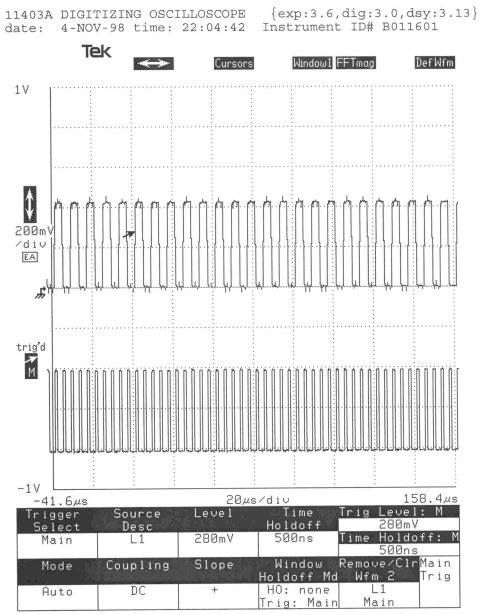
Finally, the 8x8 focal-plane-array system was tested using a 100MHz system clock frequency which meant each pixel was sampled at 1.56MHz. When the test frequency was higher than 100MHz, the non-overlapping clock signal was overlapped and did not give a correct output as shown in Figure 6.14(a) and (b). Figure 6.14 (c) and (d) show the test results using a 2.5 MHz system clock frequency with different light intensity. As it was expected, more 1s shown on the oscilloscope when the light intensity was increased. Figure 6.14 (e) and (f) show the test results using a 40MHz system clock frequency, and (g) and (h) show the test results using a 100MHz system clock frequency.



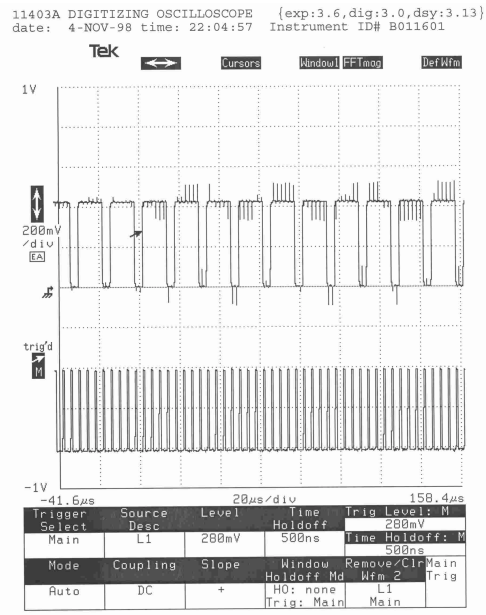
(a) Dark



(b) Bright +

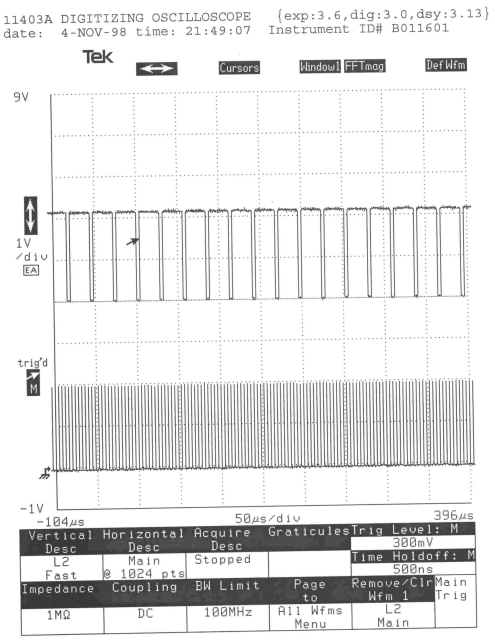


(c) Bright ++

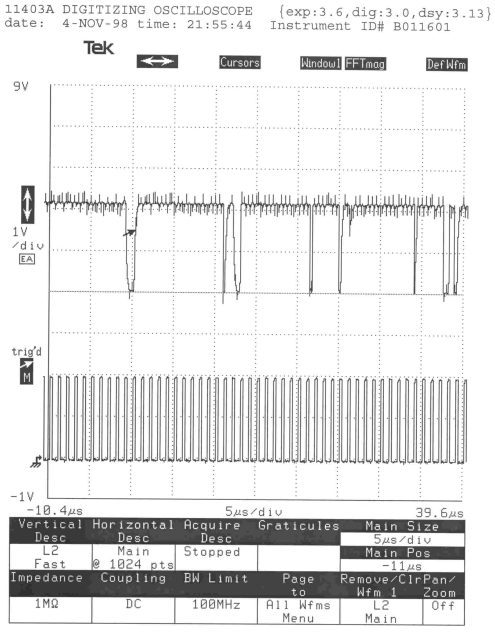


(d) Bright +++

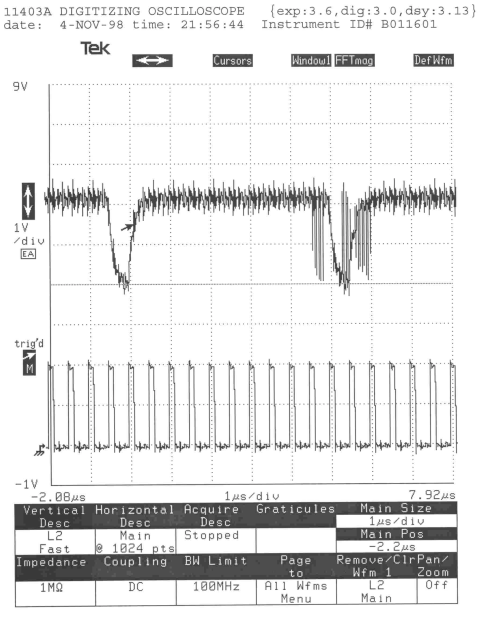
Figure 6.12 High speed test results with room light change



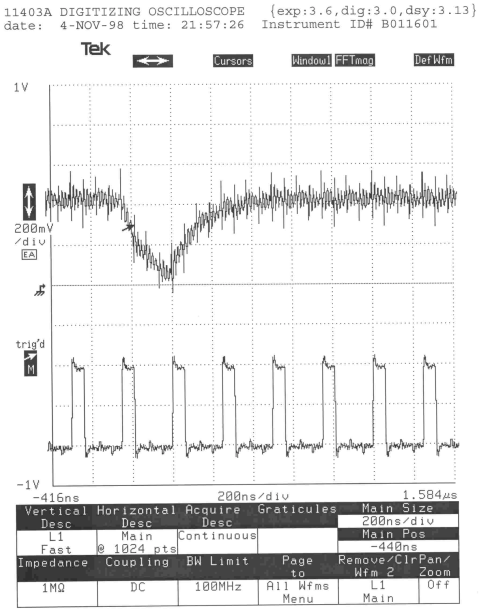
(a) 250 kHz



(b) 1 MHz

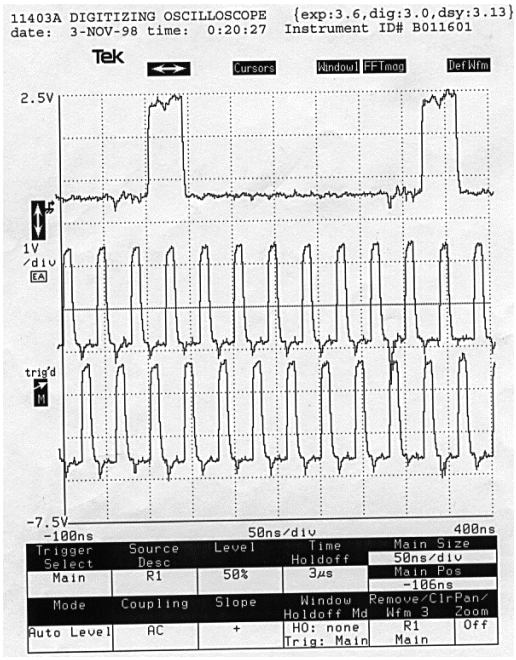


(c) 2 MHz

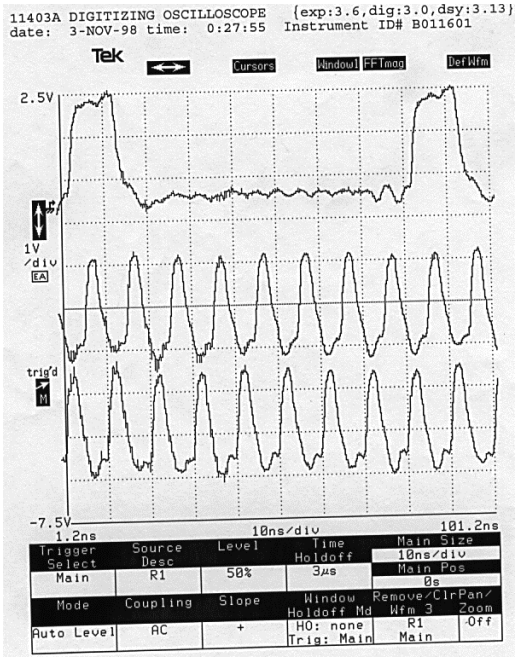


(d) 4 MHz

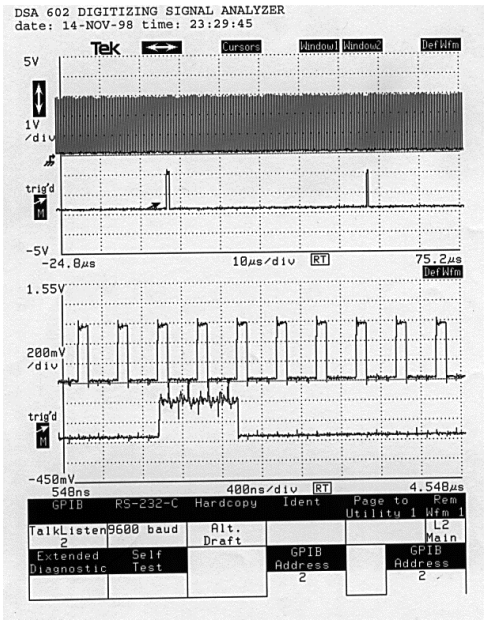
Figure 6.13 High speed test results with test pixel



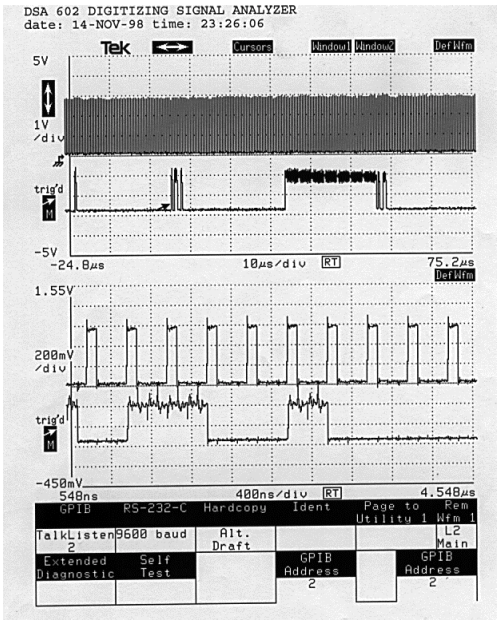
(a) 25MHz clock signals



(b) 100MHz clock signals

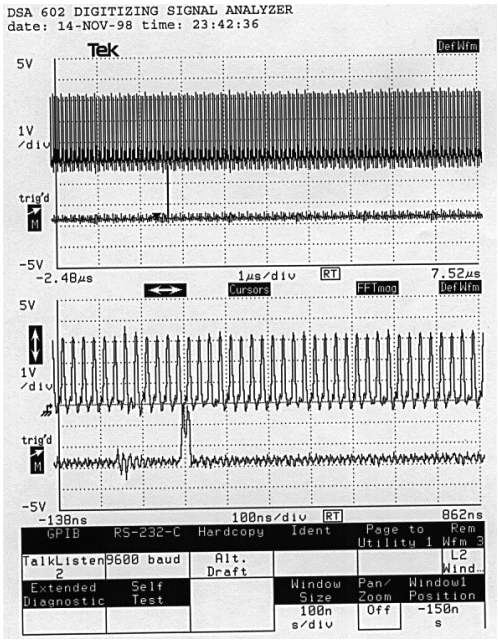


(c) 2.5MHz output with room light

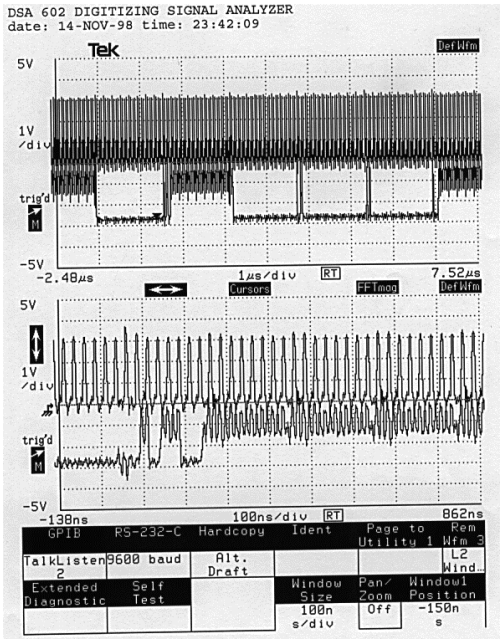


(d) 2.5MHz output with microscope light

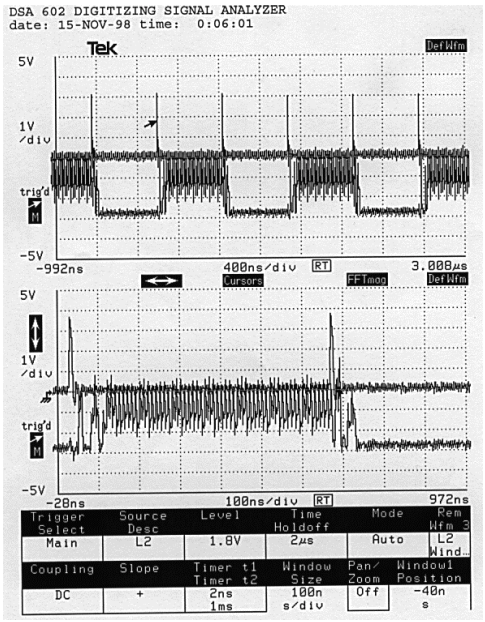
Figure 6.14 High speed test results with whole system (cont.)



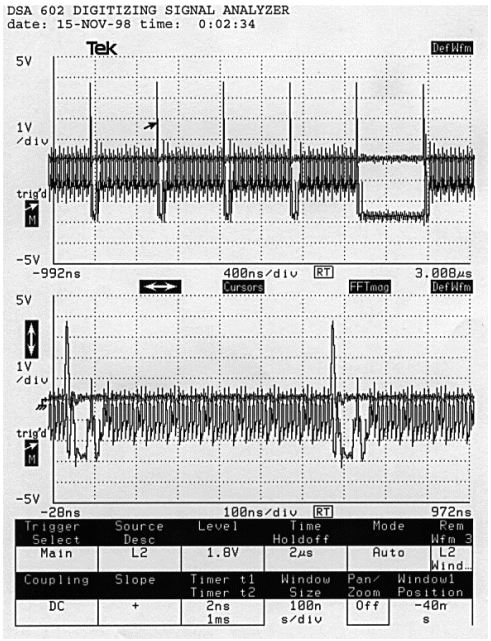
(e) 40MHz output with room light



(f) 40MHz output with microscope light



(g) 100MHz output with room light



(h) 100MHz output with microscope light

Figure 6.14 High speed test results with whole system

6.4.2 System noise

The noise added by the detectors and circuits were tried to measure by compare the test results with ideal simulation results. Figure 6.15 shows the test setup which is working up to 50 MHz system frequency.

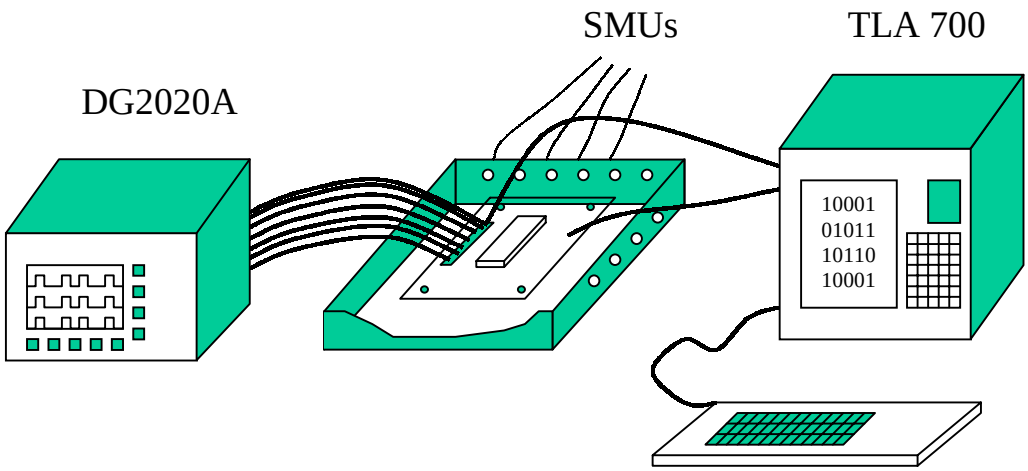


Figure 6.15 High speed test setup

The TLA 700 was programmed to read the data from the circuit every rising edge of the Fast shift_B signal. Slow shift_road is also readout to synchronize the data. The maximum capacity of the logic analyzer is 262144 which means each pixel can store 4096 data. After the data were acquired several times with several different light intensity, they were analyzed using a Matlab program. Figure 6.16 (a) shows the frequency response of the system with three different light intensity. As it was explained in chapter 4, the noise is modulated and the in-band noise is decreased. Figure 6.16(b)

shows the frequency response after filtering at 10kHz. Figure 6.17 shows the ideal simulation results when it was simulated using the Matlab program.

There are many pattern noises in the ideal simulation results which can be removed by adding a white noise. Test results have less pattern noise because there are many noise sources which help to remove the pattern noise [37]. Figure 6.18 (a) shows the time domain responses of the test results. When the light intensity is increased from 0.1345 to 0.5867, the output noise is also increased from 0.0025 to 0.0053. The ideal frequency responses in Figure 6.18(b) show the same results which means the frequency response has a input dependent quantization noise. Also the quantization noise of the ideal simulation is very sensitive to the input signal. From these simulation and test results, it is not able to distinguish the system noise from the quantization noise. We can only check the noise within the signal band is decreased when the oversampling ratio is increased.

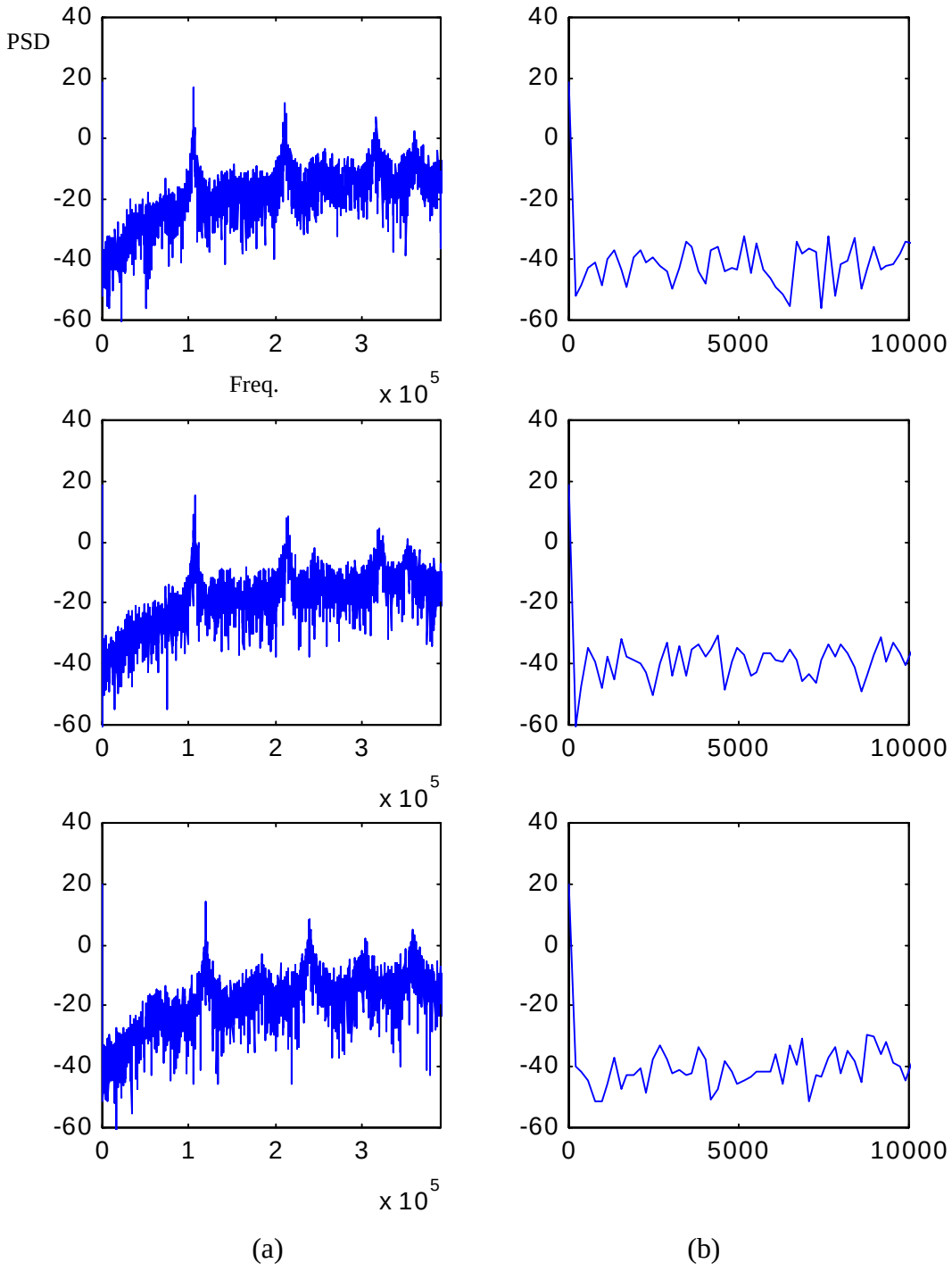


Figure 6.16 Frequency responses of test results

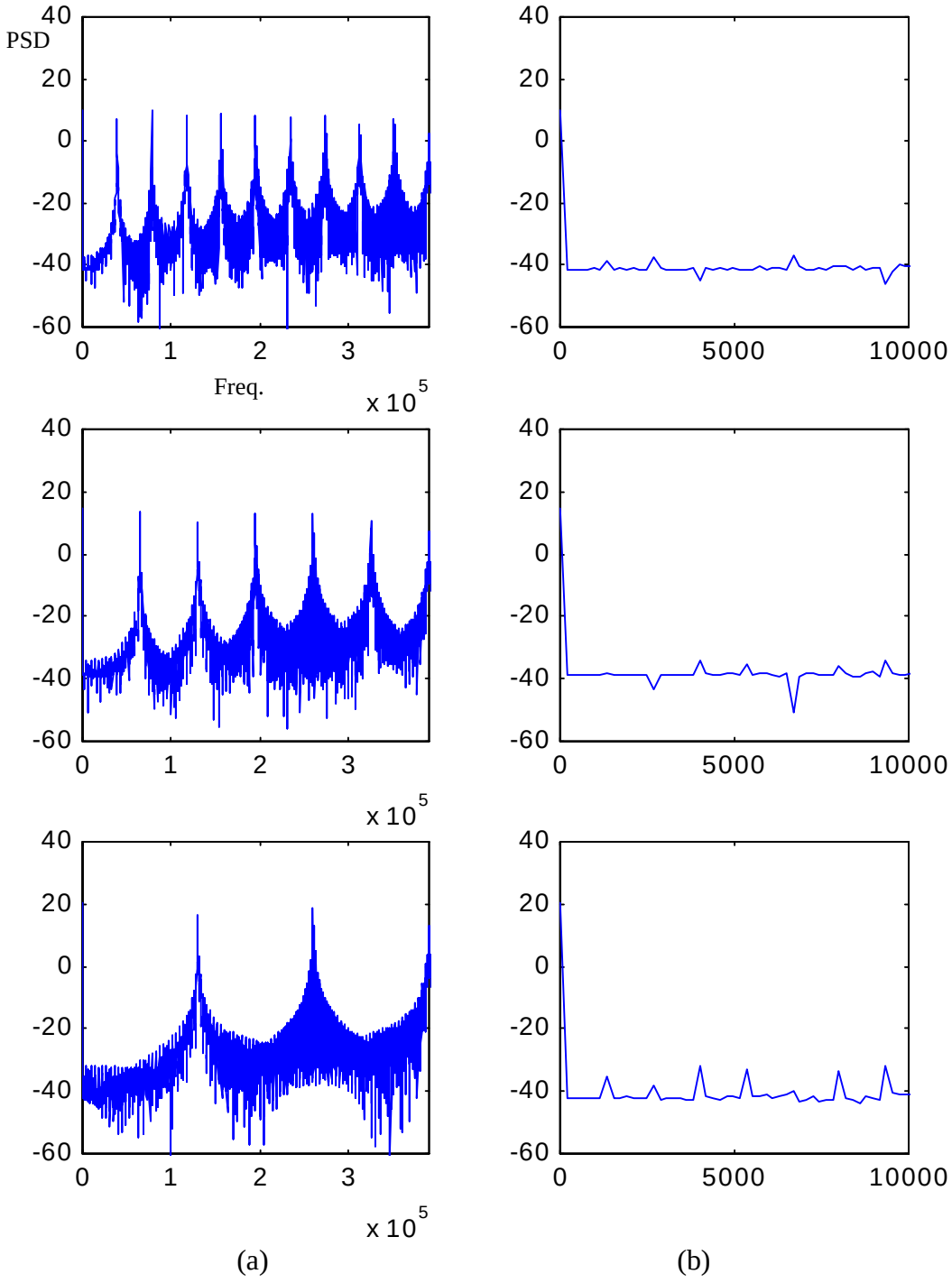


Figure 6.17 Frequency responses of ideal simulation results

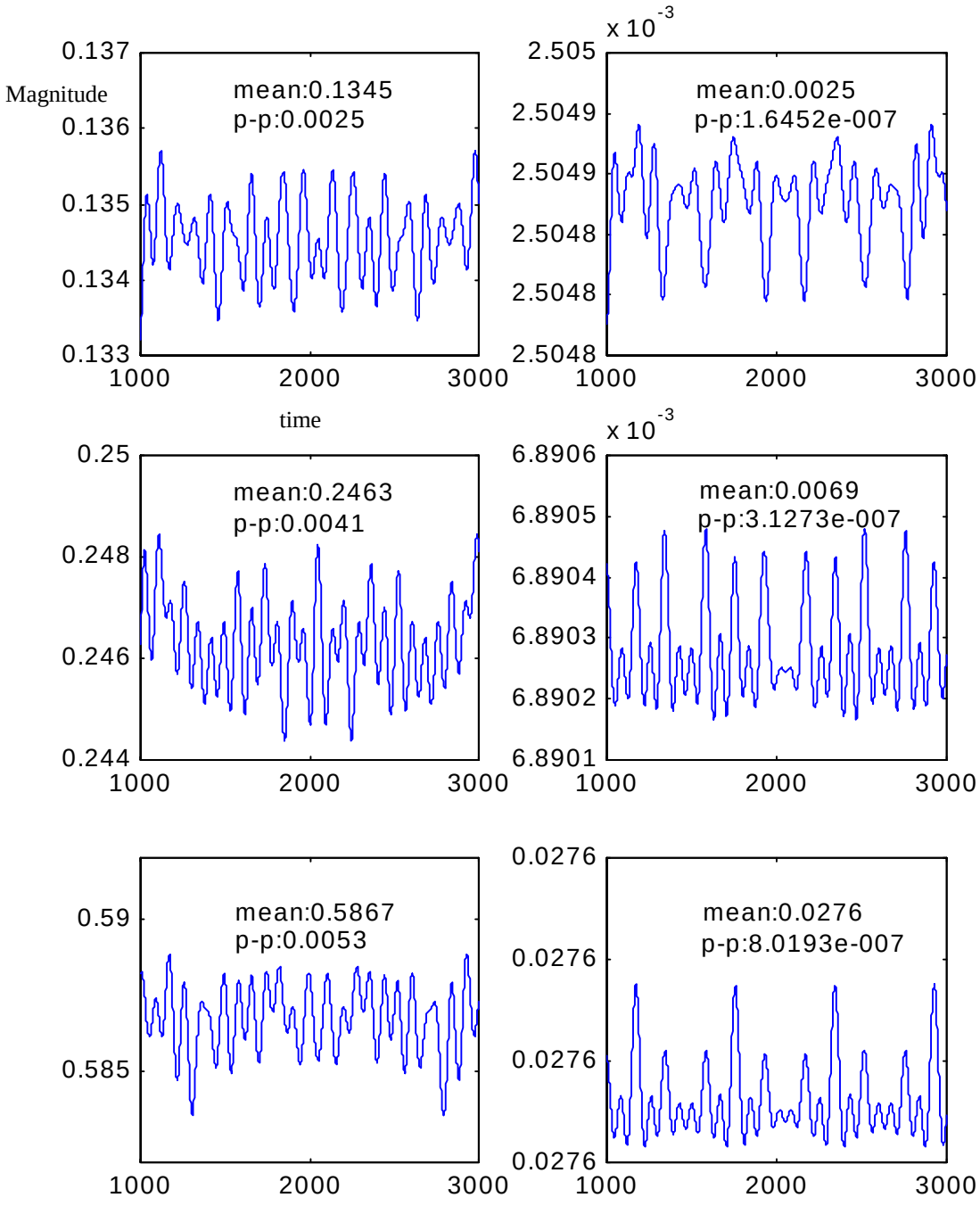


Figure 6.18 Time domain responses

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 Conclusion

In this dissertation, a new high speed readout system for focal-plane-arrays was designed and tested. The main purpose of the proposed readout system was to design a cost effective scalable high speed readout architecture for FPAs that was not available until now. There were several areas explored in this dissertation.

First, a new architecture for the focal-plane-array readout system was designed for the high speed imaging system. Several architectures had been used for the focal-plane-array systems including off-chip serial readout system, on-chip serial readout system, semi-parallel readout system, and parallel readout system. The critical problem of these readout systems was that they were not scalable; which means, the readout speeds are highly related to the array sizes. In this thesis a fully parallel readout system was implemented with through-wafer optical communication technology which was developed at the Georgia Institute of Technology. By using through-wafer optical communication, the processing speed of the proposed sub-array group was not increased while increasing the array size. From the simulation results, the proposed readout system

had an advantage when implemented for the medium resolution and large array size focal-plane-arrays. To get an 8 bit image with 100kfps, an 8x8 sub-array was chosen because that array size gave a reasonable bandwidth for the other circuits including an emitter driver, a receiver, and a digital processor.

Second, a new first-order sigma-delta A/D modulator was designed for each pixel. There were several restrictions in choosing the focal-plane-array A/D converters. It should be small enough to fit into each pixel. It should be robust when facing processing mismatch because thousands of A/D converters for the fully parallel system, and the focal-plane-arrays need good output uniformity. The resolution should be adjustable because several applications require different resolutions. A current input first-order sigma-delta A/D converter was chosen for this dissertation which satisfied all the requirements. It was implemented in a small area because only the modulator part of the converter needed to be designed within the pixel, and there were no switching capacitors or op amps. It was very robust while facing processing mismatch because the sigma-delta algorithm creates a speed for accuracy trade off. The resolution was increased when the oversampling ratio was increased. From the simulation results, the proposed A/D modulator was working properly at a greater than 100MHz sampling rate which was more than enough for the proposed readout system.

Third, two kinds of detectors were utilized. A hybrid detector was integrated on top of the circuits which had high responsivity compared to the monomaterial detector. The integrated hybrid detector was GaAs/AlGaAs double heterostructure P-i-N detector. It had a high fill factor because one of the pads was bump bonded, which saves space.

Unfortunately, it did not work properly when it was tested. The other detector was monomaterial detector which was designed in a standard digital CMOS process. The biggest reason to use the monomaterial detector was to reduce the cost by using the standard CMOS process without any additional processing. For the high speed detector, it was specially designed to have a small output capacitor. It was also designed with four photodiodes rather than a single photodiode; whereby, all were connected in parallel.

Several tests had been done to verify the proposed system. Each pixel has been tested using a 4MHz sampling frequency which was enough to get an 8 bit image with 100kfps. The whole system was tested with three different test setups. One was a slow speed test which yielded uniformity and linearity. The other was a high speed test using up to a 100MHz system bandwidth. Frequency responses were measured by using another test setup.

7.2 Publications

The following publications have resulted from this research:

- **Youngjoong Joo**, J. Park, M. Thomas, K. Chung, M. Brooke, N. Jokerst, D. Wills, "Smart CMOS focal plane arrays: Si COMS detector array and sigma delta analog to digital converter imaging system," Submitted to IEEE Journal of Selected Topics in Quantum Electronics, 1999.

- **Youngjoong Joo**, S. Fike, M. Thomas, K. Chung, M. Brooke, N. Jokerst, D. S. Wills, "High speed, smart focal plane processing using integrated photodetectors and Si CMOS VLSI sigma-delta analog-to-digital converters," IEEE LEOS Summer Tropical Meeting, pp. 55-56, 1998.
- **Youngjoong Joo**, S. Fike, K. S. Chung, M. A. Brooke, N. M. Jokerst, D. S. Wills, "Application of massively parallel processors to real time processing of high speed images," Proceedings of the Fourth International Conference on Massively Parallel Processing Using Optical Interconnections, pp. 96-100, 1997.
- H. H. Cat, A. Gentile, J. C. Eble, M. Lee, O. Vendier, **Youngjoong Joo**, D. S. Wills, M. Brooke, N. M. Jokerst A. S. Brown, "SIMPil: An OE integrated SIMD architecture for focal plane processing applications," Proceedings of the Fourth International Conference on Massively Parallel Processing Using Optical Interconnections, pp. 44-52, 1996.

7.3 Future works

To complete the fully parallel readout system for FPAs, two things need to be tested and verified. One is the speed of the through-wafer optical communication. Although the speed of the emitter and the detector were tested with other applications, they were not tested with this application. To obtain an 8 bit 100kfps images, they need to operate at more than 150MHz bandwidth. The other one is the microprocessor. The

processor was proven to work using a 50MHz data rate, which was enough to get the 8 bit 100kfps images. However, it was not tested with the analog circuits which share the same substrate. As can be expected, there will be noise problems which make the system not function properly. The other group related to this research is trying to get some test results from the mixed-signal circuits; whereby, the focal-plane-array chip and the microprocessor chip will be stacked and test together. Ultimately, a one chip high speed scalable FPAs will be available.

APPENDIX A

POWER CONSUMPTION

For analyzing the power consumption of an CMOS circuits, two components, static and dynamic power consumption, need to be considered. The static power is dissipated while the circuit is not changing states. For example, there is almost no power dissipation for the idealized CMOS inverter of Figure A.1. After the output capacitance (C_L) has been fully charged or discharged, only one of the pullup and pulldown transistors is on. While there is a slight (~10^{-12}A) leakage current through the channel of an off-transistor, there is almost no current flowing through the circuit and, therefore, almost no power consumption. Power is consumed when gates drive their outputs to new values. It is called dynamic power consumption. The dynamic power consumption depends only on the size of the capacitive load at the output and the rate at which the inverter's output switches. To understand why, the energy has to be considered, which is required to drive the inverter's output. There are two ways to calculate the power: by the current through the load capacitor C_L and by the current through the pullup transistor, represented by its effective resistance R_P.

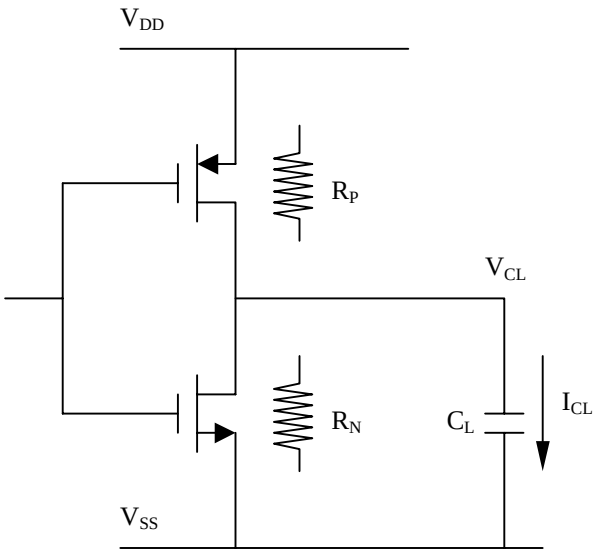


Figure A.1 CMOS inverter

The current through the capacitor and the voltage across it are:

$$i_{CL}(t) = \frac{V_{DD} - V_{SS}}{R_P} e^{-t/R_P C_L} \tag{A.1}$$

$$v_{CL}(t) = (V_{DD} - V_{SS})\left[1 - e^{-t/R_P C_L}\right]. \tag{A.2}$$

So, the energy required to change the capacitor is :

$$E_C = \int_0^\infty i_{C_L}(t) v_{C_L}(t) dt$$

$$= C_L (V_{DD} - V_{SS})^2 \left(e^{-t/R_P C_P} - \frac{1}{2} e^{-2t/R_P C_P} \right)_0^\infty$$

$$= \frac{1}{2} C_L \left(V_{DD} - V_{SS}\right)^2 . \qquad \text{(A.3)}$$

This formula depends not on resistance of the pullup transistor but on the size of the load capacitance. The current through and voltage across the pullup transistor are

$$i_P(t) = i_{C_L}(t), \text{ and} \qquad \text{(A.4)}$$

$$v_P(t) = Ve^{-t/R_P C_L} . \qquad \text{(A.5)}$$

The energy required to charge the capacitor, as computed from the resistor's point of view, is

$$\begin{aligned} E_R &= \int_0^\infty i_P(t) v_P(t) dt \\ &= C_L \left(V_{DD} - V_{SS}\right)^2 \left(e^{-2t/R_P C_L}\right)_0^\infty \\ &= \frac{1}{2} C_L \left(V_{DD} - V_{SS}\right)^2 . \end{aligned} \qquad \text{(A.6)}$$

Once again, even though the circuit's energy consumption is computed through the pullup, the value of the pullup resistance drops from the energy formula. The two energies have the same value because the currents through the register and capacitor are equal.

The energy consumed in discharging the capacitor can be calculated the same way. The discharging energy consumption is equal to the charging power consumption: $C_L(V_{DD}-V_{SS})^2/2$. A single cycle requires the capacitor to both charge and discharge, so the energy consumption over the cycle is $C_L(V_{DD}-V_{SS})^2$. Power is energy per unit time, so the power consumed by the circuit depends on how frequently the inverter's output changes. The worst case is that the inverter alternately charges and discharges its output capacitance. This sequence takes two clock cycles. The clock frequency is $f=1/t$ and the total power consumption is

$$fC_L\left(V_{DD}-V_{SS}\right)^2. \qquad \text{(A.7)}$$

Figure A.2 shows the total power consumption simulation result of the inverter according to the operating frequency. As the equation A.7 shows, the simulation result shows the power consumption is highly linear to the operating frequency. The static power is 22.8 pW for this simulation, which is almost negligible compared to the dynamic power.

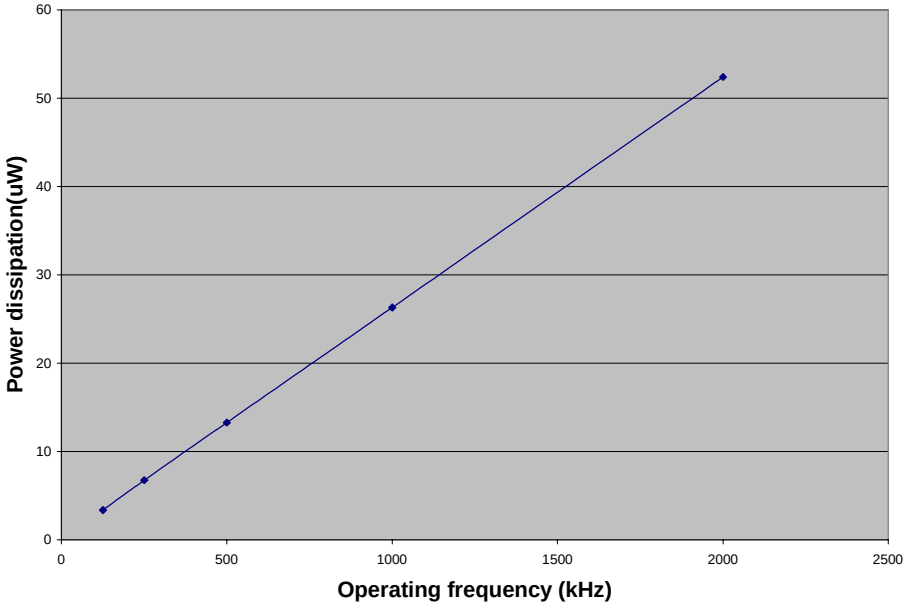


Figure A.2 Power dissipation vs. operating frequency of the inverter

APPENDIX B

HIGH-ORDER SIGMA-DELTA MODULATORS

In a sigma-delta circuit, feedback via an integrator shapes the spectrum of the modulation noise, placing most of its energy outside the signal band. In general, the characteristics of the filter included in the feedback loop determine the shape of the noise spectrum [64][65]. The objective of using improved filters is to reduce the net noise in the signal band. Ordinary first-order sigma-delta modulation subtracts the previous value of the quantization error from the present error. Higher order prediction should give better results than this first-order prediction.

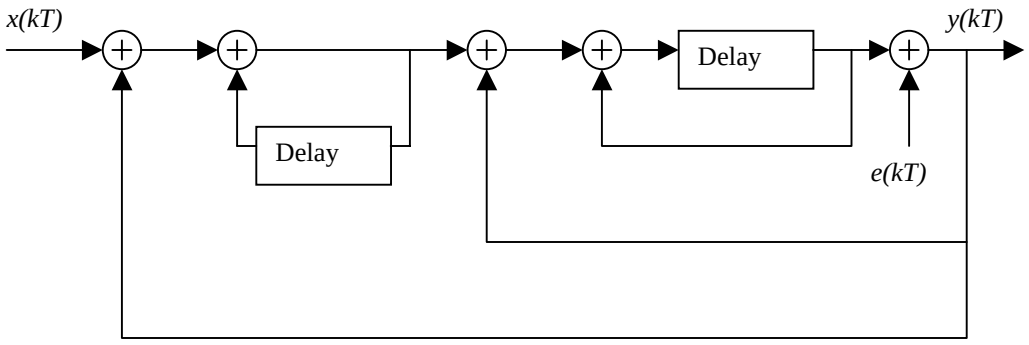


Figure B.1 A second-order sigma-delta modulator

There are several circuit arrangements that give second-order predictions of the quantization error [66]. The one shown in Figure B.1 is easy to build and is tolerant of circuit imperfection.

It is an iteration of sigma-delta feedback loops. The output of this modulator can be expressed as

$$y(kT) = x(kT - T) + \big(e(kT) - 2e(kT - T) + e(kT - 2T)\big). \qquad \text{(B.1)}$$

The noise term is represented by

$$n(kT) = e(kT) - 2e(kT - T) + e(kT - 2T)\,. \qquad \text{(B.2)}$$

The spectral density of the modulation noise then be expressed as

$$N(f) = E(f)\left|1 - e^{-j\omega\tau}\right| = 4e_{rms}\sqrt{2\tau}\,\sin^2\left(\frac{\omega\tau}{2}\right). \qquad \text{(B.3)}$$

The noise power in the signal band is

$$n_o{}^2 = \int_0^{f_0} |N(f)|^2\, df \approx e_{rms}{}^2\,\frac{\pi^2}{\sqrt{5}}(2f_0\tau)^{5/2}, \quad f_s{}^2 \rangle\rangle f_0{}^2\,. \qquad \text{(B.4)}$$

The noise falls by 15 dB for every doubling of the sampling frequency, providing 2.5 extra bits of resolution [67][68].

The techniques can be extended to provide higher-order predictions by adding more feedback loops to the circuit. In general, when there are L loops and the system is stable, it can be shown that the spectral density of the modulation is

$$N_L(f) = e_{rms}\sqrt{2\tau}\left(2\sin\left(\frac{\omega\tau}{2}\right)\right)^L. \qquad \text{(B.5)}$$

and for oversampling ratios greater than 2, the *rms* noise in the signal band is given approximately by

$$n_o \approx e_{rms}\frac{\pi^L}{\sqrt{2L+1}}(2f_0\tau)^{L+1/2}, \qquad f_s{}^2 \rangle\rangle f_0{}^2. \qquad \text{(B.6)}$$

This noise falls *3(2L+1)* dB for every doubling of the sampling rate, providing *L+0.5* extra bits.

APPENDIX C

CURRENT BUFFER INPUT IMPEDANCE

The current buffer is redrawn in Figure C.1 and the small signal equivalent circuits are shown in Figure C.2.

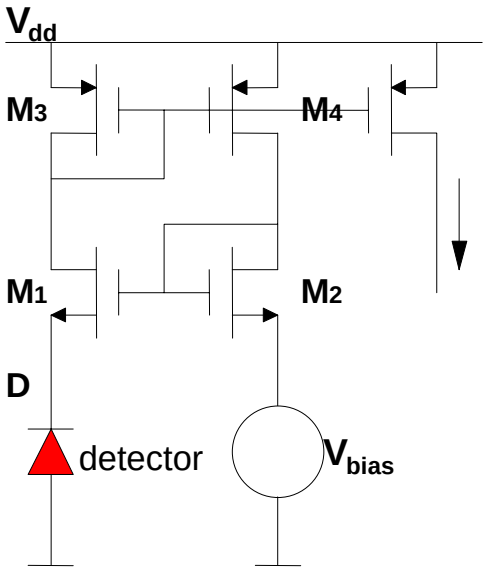
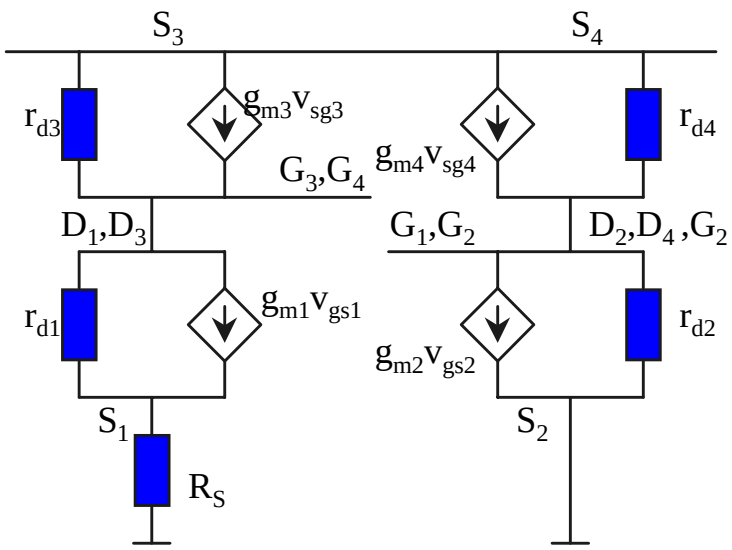
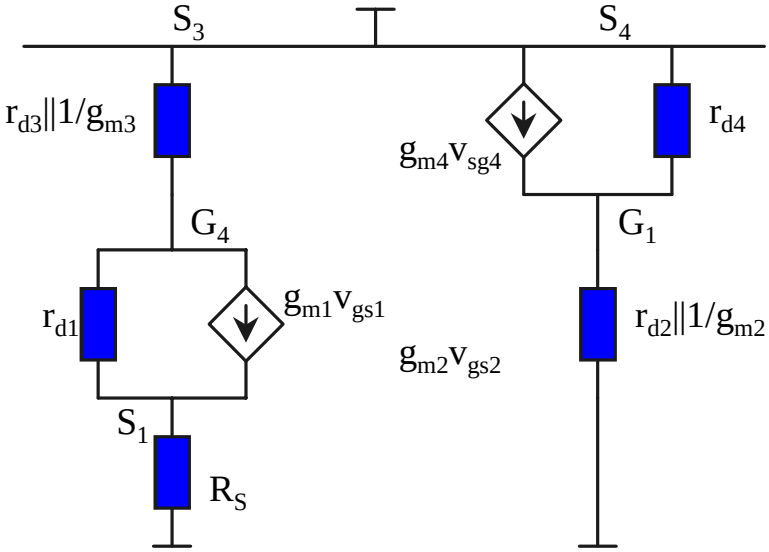


Figure C.1 Current buffer



(a)



(b)

Figure C.2 Current buffer small signal equivalent circuits

The gate voltage of the transistor M_1 is as follows.

$$V_{g1} = g_{m3} v_{sg4} \left(r_{d2} \parallel \frac{1}{g_{m2}} \parallel r_{d4} \right) = g_{m3} \left(v_{s4} - v_{g4} \right) \left(\frac{1}{G_2 + g_{d4}} \right)$$

$$= -g_{m4} v_{g4} \left(\frac{1}{G_2 + g_{d4}} \right) \qquad \text{(C.1)}$$

The input impedance of the detector can be calculated by using test voltage source connect to the source of transistor M_1. The input voltage of the test voltage source is

$$V_t = r_{d1} \left[I_t - G_S V_t + g_{m1} \left(v_{g1} - v_{s1} \right) \right] + \left(I_t - G_S V_t \right) / G_3 \qquad \text{(C.2)}$$

The input impedance of the current buffer is obtained by dividing the V_t with I_t.

$$\frac{V_t}{I_t} = Z_{in} \approx \frac{1}{g_{m1}} \left(\frac{g_{d1} + g_{d3}}{g_{m3}} + \frac{g_{d2} + g_{d4}}{g_{m2}} \right) \qquad \text{(C.3)}$$

APPENDIX D

DECIMATION FILTER

The output of the modulator represents the input signal together with its out-of-band components, modulation noise, circuit noise, and interference. The decimation filter shown in the Figure 4.1 serves to attenuate all of the out-of-band energy of this signal so that it may be resampled at the Nyquist rate without incurring significant noise penalty because of aliasing.

In addition to the attenuation of signal and noise components outside the baseband, several other requirements influence the design and complexity of the decimation filter. For example, specifications for the flatness, or ripple, of the transfer function in the passband and the width of the transition from passband to stopband strongly influence the complexity of the filter. Typically the order of an FIR low-pass filter is directly related to a function of the required ripples δ_p and δ_s in the passband and stopband, respectively, and inversely related to the normalized width of the transition band [69]; that is

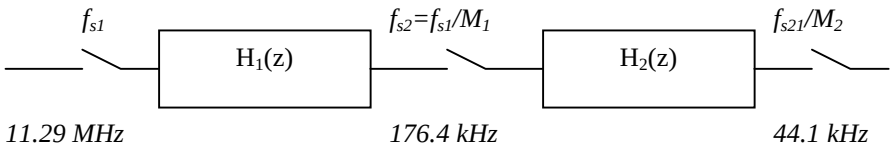
$$N = \frac{D_\infty(\delta_p, \delta_s)}{\Delta F / F} \qquad \text{(D.1)}$$

where

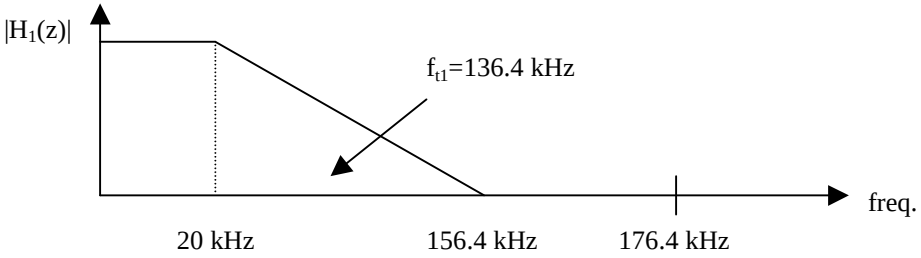
$$D_{\infty}(\delta_p, \delta_s) = \log_{10}\delta_s\left[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3\right] + \left[a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_5\right]. \quad \text{(D.2)}$$

where a_1=0.005309, a_2=0.07144, a_3=-0.4761, a_4=-0.00266, a_5=-0.5941, a_6=-0.4278, and ΔF is the transition bandwidth, and the F is the sampling frequency at which the filter design is referred. When large oversampling ratios are required, as in the case of sigma-delta converters, it can be seen that the cutoff frequency requirements on the digital filters defined above can become extremely severe: that is, ΔF becomes small relative to F, leading to excessively large filter orders and high-word-length requirements on these filters. Fortunately, these constraints can be overcome by considering multistage designs in which the decimator is defined as a cascade of two or more stages such that the overall conversion ratio is the product of the ratios of the stages [70]. Figure D.1 illustrates this principle for an example of a two-stage decimation filter. The two stages are designed to attenuate noise above 24.1 kHz so that the sampling rate can be reduced by a factor of 256 from 11.29 MHz to the 44.1 kHz Nyquist rate. The decimation ratio is the ratio of the input and output rates of the filter. The decimation ratio is equivalent to the oversampling ratio, M, in an oversampled A/D converter. In Figure D.1 the sampling rate is reduced in the first stage by M_1=64 and in the second stage by M_2=4. Note that the first-stage filter need only provide attenuation at those frequencies that are aliased into the baseband upon resampling at the intermediate rate of 176.4 kHz. Thus, the first stage

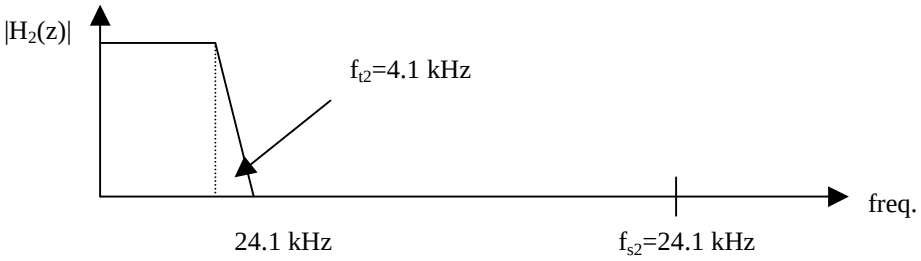
filter can have a relatively gentle cutoff characteristic. The ratio of the sampling rate to the width of the transition band in the first stage (f_{s12}/f_{t1}) is 82.8; in the second stage,



(a) Block diagram



(b) Magnitude response of the first stage



(c) Magnitude response of the second stage

Figure D.1 Two-stage decimation filter

f_{s2}/f_{t2}=43. The sum of these two, 125.8, is less than 5% of the ration f_s/f_t=2754 for a single stage filter that meets the same specifications.

The principle of decimation filtering in stages can be extended to produce further savings in the required number of coefficients. The optimal number of stages depends on the specifications of a particular decimation filter. In the case of oversampled A/D converters based on sigma-delta modulation, two or three stages result in efficient hardware implementations. This is because a very efficient comb filter may be used for the first stage of decimation. Comb filters are suitable for reducing the sampling rate to four times the Nyquist rate. Therefore, only one or two additional stages are needed to complete the filter.

A comb filter computes a simple running average of its last M_1 input samples,

$$y(n) = \frac{1}{M_1} \sum_{i=0}^{M_1-1} x(n-i) . \qquad \text{(D.3)}$$

The corresponding z-transform is

$$H_1(z) = \left(\frac{1}{M_1} \times \frac{1 - z^{-M_1}}{1 - z^{-1}} \right). \qquad \text{(D.4)}$$

Evaluating $H_1(z)$ on the unit circle in the z-plane gives the frequency response,

$$H_1(f) = \left(\frac{\sin \pi f M_1 T_1}{M_1 \sin \pi f T_1} \right) \times e^{-j\pi f T_1 (M_1 - 1)} \qquad \text{(D.5)}$$

where $T_1 = 1/f_{s1}$ is the input sampling period. Note that the phase response of a comb filter is linear. The magnitude of the gain, shown in Figure D.2, has zeros at integer multiples of the intermediate resampling frequency, f_{s1}/M_1. Noise in the frequency bands centered on these frequencies is aliased into the baseband when the output of the comb filter is resampled at f_{s1}/M_1. Noise in other frequency bands is aliased outside the baseband and removed by provide attenuation stages of the decimation filter.

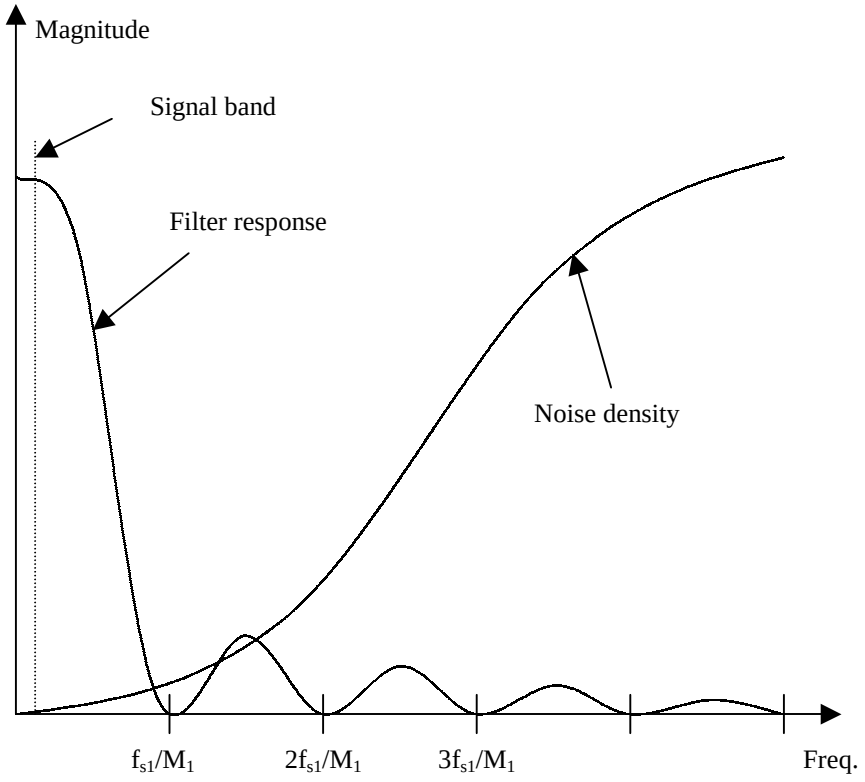


Figure D.2 Magnitude response of comb filter

APPENDIX E

DATA ACQUISITION PROGRAM

E.1 Algorithm

```
RESET;

DO
   FOR (Counter=0; Counter<NumberLevels; Counter++)    {
      DO
         READ divider's output from Port B;
         EXTRACT DataOrder and DataTiming;
      WHILE (DataOrder == PreviousDataOrder)
      PreviousDataOrder ←  (PreviousDataOrder + 1) % 32;
      IF ( DataOrder == PreviousDataOrder )
         IF ( DataTiming < Maximum Tolerable )
            Value ← data read from Port A;
            Matrix[Counter] = Value;
      ELSE
         Error ← YES;
    }
   IF (Error == NO)    {
      IF (PacketCounter == NumberSamples)
         FOR (i=0; i<8; i++)
            FOR (j=0; j<8; j++)
               DISPLAY ( PixelValue[i][j]/NumberSamples );
         PacketCounter ← 0;
         PixelValue[i][j] ← 0 for all i,j ;
      ELSE
         PacketCounter++;
         FOR (FrameNumber=0; FrameNumber<NumberLevels; FrameNumber++)
            FOR (ByteNumber=0; ByteNumber<8; ByteNumber++)
```

```
            FOR (I=0; I<8; I++)
               IF ((7-I)th bit of Matrix[8*FrameNumber +   ByteNumber] = 1)
                  PixelValue[I][ByteNumber] ++;
    }
   DO
      READ divider's output from Port B;
      EXTRACT DataOrder;
   WHILE (DataOrder == 0) ;

WHILE (no end of test is requested) ;
END
```

E.2 Program

```
/* Include files */
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <math.h>

#include "cb.h"

#define YES 5;
#define NO  7;

/* Prototypes */
void Clear_Pixel_Value_Matrix (int Matrix[8][8]);
void ClearScreen (void);
void GetCursor (int *x, int *y);
void MoveCursor (int x, int y);

void main (int argc, char *argv[])
   {
   /* Variable Declarations */
   int num_frames, num_samples;
   int Row, Col, I;
   int BoardNum = 0;
   int ULStat = 0;
   int PortNum, Direction;
   int PowerVal, RowNumber;
```

```
int ByteNumber, PreviousByteNumber;
int FrameNumber;
int DividerState;
int Counter;
int PacketCounter=0;
unsigned DividerOutput, Value, DataValue[2048];
unsigned PixelValue[8][8];
int Error;

if (argc != 3) {
   printf("\n ----Please enter: topview num_frames num_samples \n");
   exit(1);
  }

num_frames = atoi(argv[1]);
num_samples = atoi(argv[2]);

/* Initiate error handling
       Parameters:
          PRINTALL :all warnings and errors encountered will be printed
          STOPALL  :if any error is encountered, the program will stop */
ULStat = cbErrHandling (PRINTALL, STOPALL);


/* configure FIRSTPORTA, FIRSTPORTB  for digital input
   and FIRSTPORTCL for digital output
       Parameters:
          BoardNum    :the number used by CB.CFG to describe this board.
          PortNum     :the input port
          Direction   :sets the port for input or output */

PortNum = FIRSTPORTA;
Direction = DIGITALIN;
ULStat = cbDConfigPort (BoardNum, PortNum, Direction);
ULStat = cbDConfigPort (BoardNum, FIRSTPORTB, DIGITALIN);
ULStat = cbDConfigPort (BoardNum, FIRSTPORTCL, DIGITALOUT);

/* Reset circuit */
ULStat = cbDOut (BoardNum, FIRSTPORTCL, 0 );

/* set up the display screen */
ClearScreen();
printf ("\n\n                          -- TOP CHIP -- \n");
```

```
printf ("                (result after averaging %d samples)\n\n\n",num_samples);
printf ("                 <Press any key to quit this test>\n\n");
printf ("               ****************************************\n");
printf ("               *                                      *\n");
GetCursor (&Col, &Row);
printf ("           0   *                                      *\n");
printf ("           1   *                                      *\n");
printf ("           2   *                                      *\n");
printf ("           3   *                                      *\n");
printf ("           4   *                                      *\n");
printf ("           5   *                                      *\n");
printf ("           6   *                                      *\n");
printf ("           7   *                                      *\n");
printf ("               *                                      *\n");
printf ("               ****************************************\n");

ULStat = cbDOut (BoardNum, FIRSTPORTCL, 15 );

Clear_Pixel_Value_Matrix(PixelValue);

do {
      /* +++Read 8 bits digital input and display+++
        ULStat = cbDin(boardNum, PortNum, DataValue)
        Parameters:
              BoardNum    :the number used by CB.CFG to describe this board
              PortNum     :the input port
              DataValue   :the value read from the port   */

  PreviousByteNumber = 0;
  Error = NO;

  /* ---begin reading 'num_frames' consecutives frames--- */
  for (Counter=0 ; Counter < 8*num_frames ; Counter++)  {

       /* ---wait for next byte--- */
       do {
        /* ULStat = cbDIn(BoardNum, FIRSTPORTB, &DividerOutput);  */
        asm   mov DX,341h;   /* first port B*/
        asm   in  AL,DX;     /* read        */
        DividerOutput = _AL;

        ByteNumber = (int) DividerOutput/8 ;
        }
```

```
      while (ByteNumber == PreviousByteNumber) ;

      PreviousByteNumber = (PreviousByteNumber +1) % 32;
      if (ByteNumber == PreviousByteNumber) {
        DividerState = DividerOutput % 8;

        if ( DividerState < 6 )  {
          /* ULStat = cbDIn(BoardNum, FIRSTPORTA, &DataValue); */
           asm   mov DX,340h;   /* first port A*/
           asm   in  AL,DX;     /* read        */
           Value = _AL;
           DataValue[Counter] = Value;
         }
        }
      else {
         /* reports an error has occured */
         Error=YES; /*there is missing byte*/
       }  /*else*/
     }  /*for*/

if (Error==7) {  /*there was no missing byte*/

      if (PacketCounter==num_samples) {
        /* --Display Pixel Value to the screen-- */
        for (ByteNumber=0; ByteNumber<8; ByteNumber++)
          for (I = 0; I < 8; I++)    {
                MoveCursor (24+Col+4*I, Row+ByteNumber);
                printf("%3u", PixelValue[I][ByteNumber]/PacketCounter);
               }
        PacketCounter=0;

        Clear_Pixel_Value_Matrix(PixelValue);
        /* for (i=0; i<8; i++)
          for (j=0; j<8; j++)
                PixelValue[i][j] = 0;   */
        }

      else {
        PacketCounter++;

        /* ---Analyze 'num_frames' frames read from top chip--- */
        for (FrameNumber=0; FrameNumber<num_frames; FrameNumber++)
          for (ByteNumber=0; ByteNumber<8; ByteNumber++)  {
```

```
                    Value = DataValue[8*FrameNumber+ByteNumber];

                    /* parse Value into bit values */
                    for (I = 0; I < 8; I++)    {
                      PowerVal = (int)pow(2, 7-I);  /*most significant bit first*/

                      if (Value & PowerVal)
                           PixelValue[I][ByteNumber] ++;
                      }
                    }
               } /*else*/

          }


     /* ---synchronize input to the begining of next frame set--- */
     do {
           /* ULStat = cbDIn(BoardNum, FIRSTPORTB, &DividerOutput);  */
          asm   mov DX,341h;   /* first port B*/
          asm   in  AL,DX;     /* read        */
          DividerOutput = _AL;

          ByteNumber = (int) DividerOutput/8 ;
         }
     while (ByteNumber != 0) ;


     }
    while ( bioskey(1) == 0 ) ;
    printf("\n\n\n");

   }

/*************************************************************************
*
* Name:      ClearScreen
* Arguments: ---
* Returns:   ---
*
* Clears the screen.
*
*************************************************************************/
```

```
#define BIOS_VIDEO   0x10

void
ClearScreen (void)
{
   union REGS InRegs,OutRegs;

   InRegs.h.ah = 0;
   InRegs.h.al = 2;
   int86 (BIOS_VIDEO, &InRegs, &OutRegs);
   return;
}

/*************************************************************************
*
* Name:      Clear_Pixel_Value_Matrix
* Arguments: ---
* Returns:   ---
*
* Clears the matrix called PixelValueMatrix to zero its elements.
*
*************************************************************************/

void

Clear_Pixel_Value_Matrix (int Matrix[8][8])
{
   int i,j;

   for (i=0; i<8; i++)
      for (j=0; j<8; j++)
            Matrix[i][j] = 0;
}
/*************************************************************************
*
* Name:      MoveCursor
* Arguments: x,y - screen coordinates of new cursor position
* Returns:   ---
*
* Positions the cursor on screen.
*
*************************************************************************/
```

```
void
MoveCursor (int x, int y)
{
   union REGS InRegs,OutRegs;

   InRegs.h.ah = 2;
   InRegs.h.dl = (char) x;
   InRegs.h.dh = (char) y;
   InRegs.h.bh = 0;
   int86 (BIOS_VIDEO, &InRegs, &OutRegs);
   return;
}


/*************************************************************************
*
* Name:      GetCursor
* Arguments: x,y - screen coordinates of new cursor position
* Returns:   *x and *y
*
* Returns the current (text) cursor position.
*
*************************************************************************/

void
GetCursor (int *x, int *y)
{
   union REGS InRegs,OutRegs;

   InRegs.h.ah = 3;
   InRegs.h.bh = 0;
   int86 (BIOS_VIDEO, &InRegs, &OutRegs);
   *x = OutRegs.h.dl;
   *y = OutRegs.h.dh;
   return;
}
```

APPENDIX F

HSPICE INPUT DATA AND SIMULATION MODEL

F.1 Input data

```
Vdd    1 0 DC 2.5V
Vdd2   103 0 DC 2.5V
Vss    2 0 DC -2.5V
Vss2   104 0 DC -2.5V
Vss3   107 0 DC -2.5V
Vss4   113 0 DC -2.5V
Vss5   114 0 DC -2.5V
Vss6   121 0 DC -2.5V

****************************
*      Floating nodes
****************************
Vss7   3 0 DC -2.5V
Vss8   4 0 DC -2.5V
Vss9   102 0 DC -2.5V
Vss10   105 0 DC -2.5V
Vss11   108 0 DC -2.5V
Vss12   112 0 DC -2.5V
Vss13   119 0 DC -2.5V
Vss14   122 0 DC -2.5V
Vss15   123 0 DC -2.5V
Vss16   124 0 DC -2.5V
Vss17   120 0 DC -2.5V
****************************

D2     117 vled D1N3600
Ibias  116 0 DC -2.5m
Vled   vled 0 DC -2V
Ibias2  115 0 DC 6.5m

.PARAM      lambda=0.5U  Delta_Wp=3.7360E-07
Delta_Wn=2.9700E-7

V_V1 8 0
+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 4.6n 10n
V_V2 7 0
+PULSE -2.5v 2.5v 5.1n 0.1n 0.1n 4.6n 10n
V_bias 5 0 -1.5v
V_ref  6 0 0v
I_bias 9 0  -1920u

I_00 13 0 sin 7u  8u  2mega  0 0 0
I_01 15 0 sin 10u  10u  3mega 0 0 0
I_02 17 0 sin 10u  8u  .8mega 0 0 0
I_03 19 0 sin 8u  8u  .5mega 0 0 0
I_04 21 0 sin 10u  3u  2mega 0 0 0
I_05 23 0 sin 10u  8u  .6mega 0 0 0
I_06 25 0 sin 15u  5u  1mega 0 0 0
I_07 27 0 sin 15u  8u  3mega 0 0 0

I_10 29 0 sin 7u  8u  2mega 0 0 0
I_11 30 0 sin 10u  10u  3mega 0 0 0
I_12 31 0 sin 10u  8u  .4mega 0 0 0
I_13 32 0 sin 8u  8u  .5mega 0 0 0
I_14 33 0 sin 10u  3u  3mega 0 0 0
I_15 34 0 sin 10u  8u  .6mega 0 0 0
I_16 35 0 sin 15u  5u  3mega 0 0 0
I_17 36 0 sin 15u  8u  2mega 0 0 0

I_20 38 0 sin 7u  8u  1mega 0 0 0
I_21 39 0 sin 10u  10u  1mega 0 0 0
I_22 40 0 sin 10u  8u  .8mega 0 0 0
I_23 41 0 sin 8u  8u  .5mega 0 0 0
I_24 42 0 sin 10u  3u  1mega 0 0 0
I_25 43 0 sin 10u  8u  .6mega 0 0 0
I_26 44 0 sin 15u  5u  1mega 0 0 0
I_27 45 0 sin 15u  8u  2mega 0 0 0
```

```
I_30 47 0 sin 7u  8u  1mega 0 0 0
I_31 48 0 sin 10u  10u  1mega 0 0 0
I_32 49 0 sin 10u  8u  .8mega 0 0 0
I_33 50 0 sin 8u  8u  .5mega 0 0 0
I_34 51 0 sin 10u  3u  1mega 0 0 0
I_35 52 0 sin 10u  8u  .6mega 0 0 0
I_36 53 0 sin 15u  5u  1mega 0 0 0
I_37 54 0 sin 15u  8u  2mega 0 0 0

I_40 56 0 sin 7u  8u  1mega 0 0 0
I_41 57 0 sin 10u  10u  1mega 0 0 0
I_42 58 0 sin 10u  8u  .8mega 0 0 0
I_43 59 0 sin 8u  8u  .5mega 0 0 0
I_44 60 0 sin 10u  3u  1mega 0 0 0
I_45 61 0 sin 10u  8u  .6mega 0 0 0
I_46 62 0 sin 15u  5u  1mega 0 0 0
I_47 63 0 sin 15u  8u  2mega 0 0 0


I_50 65 0 sin 7u  8u  1mega 0 0 0
I_51 66 0 sin 10u  10u  1mega  0 0 0
I_52 67 0 sin 10u  8u  .8mega 0 0 0
I_53 68 0 sin 8u  8u  .5mega 0 0 0
I_54 69 0 sin 10u  3u  1mega 0 0 0
I_55 70 0 sin 10u  8u  .6mega 0 0 0
I_56 71 0 sin 15u  5u  1mega 0 0 0
I_57 72 0 sin 15u  8u  2mega 0 0 0

I_60 74 0 sin 7u  8u  1mega 0 0 0
I_61 75 0 sin 10u  10u  1mega 0 0 0
I_62 76 0 sin 10u  8u  .8mega 0 0 0
I_63 77 0 sin 8u  8u  .5mega 0 0 0
I_64 78 0 sin 10u  3u  1mega 0 0 0
I_65 79 0 sin 10u  8u  .6mega 0 0 0
I_66 80 0 sin 15u  5u  1mega 0 0 0
I_67 81 0 sin 15u  8u  2mega 0 0 0

I_70 83 0 sin -7u  8u  1mega 0 0 0
I_71 84 0 sin 10u  10u  1mega 0 0 0
I_72 85 0 sin 10u  8u  .8mega 0 0 0
I_73 86 0 sin 8u  8u  .5mega 0 0 0
I_74 87 0 sin 10u  3u  1mega 0 0 0
I_75 88 0 sin 10u  8u  .6mega 0 0 0
I_76 89 0 sin 15u  5u  1mega 0 0 0
I_77 90 0 sin 15u  8u  2mega 0 0 0


V_mux_vin 99 0
*+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 79.8n 640n
+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 9.8n 640n

V_mux_clock+ 100 0
*+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 39.6n 80n
+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 4.6n 80n

V_mux_clock- 101 0
*+PULSE -2.5v 2.5v 40.1n 0.1n 0.1n 39.6n 80n
+PULSE -2.5v 2.5v 5.1n 0.1n 0.1n 4.6n 80n

V_shift_read 106 0
*+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 9.8n 80n
+PULSE -2.5v 2.5v 10.1n 0.1n 0.1n 9.8n 80n

V_shift_clock+ 110 0
*+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 4.6n 10n
+PULSE -2.5v 2.5v 0.1n 0.1n 0.1n 4.6n 10n

V_shift_clock- 111 0
*+PULSE -2.5v 2.5v 5.1n 0.1n 0.1n 4.6n 10n
+PULSE -2.5v 2.5v 5.1n 0.1n 0.1n 4.6n 10n

.include 'ext.lib'
.tran 0.1ns 300ns
.include 'si_delta5.spice'
.OP
.probe
.END
```

F.2 HSPICE model

```
*****  Subcircuit from file ./adder.ext
.SUBCKT adder 1 2 3 4 5

M1 6 5 1 1 PMOS W=4.0U L=1.0U
+ AD=3.2P PD=16.0U AS=8.2P PS=16.0U
M2 6 4 1 1 PMOS W=4.0U L=1.0U
+ AD=3.2P PD=16.0U AS=8.2P PS=16.0U
M3 3 6 1 1 PMOS W=8.0U L=1.0U
+ AD=6.5P PD=32.0U AS=16.5P PS=32.0U
M4 7 4 6 8 NMOS W=4.0U L=1.0U
+ AD=0.0P PD=2.0U AS=0.0P PS=16.0U
M5 2 5 7 8 NMOS W=4.0U L=1.0U
+ AD=0.0P PD=16.0U AS=0.0P PS=2.0U
M6 2 6 3 8 NMOS W=4.0U L=1.0U
+ AD=0.0P PD=16.0U AS=0.0P PS=16.0U

C1 5 0 5.7F
C2 4 0 5.3F
C3 6 0 8.7F
C4 3 0 4.4F
C5 2 0 12.3F
C6 1 0 13.6F
*** Node Listing for subckt:  adder
**  0     Node 0 is the global ground node
**  1     Vdd!
**  2     GND!
**  3     Vout
**  4     Vin2
**  5     Vin1
**  6     a_2_57#
**  7     a_22_29#
**  8     Gnd!
.ENDS

*****  Subcircuit from file ./delta_sub2.ext
.SUBCKT delta_sub2 1 2 3 4 5 6 7 8 9 10 11

M1 12 12 1 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=4.2P PS=19.0U
M2 13 12 1 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=4.2P PS=19.0U
M3 14 12 1 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=4.2P PS=19.0U
M4 15 16 1 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=4.2P PS=19.0U
M5 16 16 1 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=4.2P PS=19.0U
M6 17 7 1 1 PMOS W=2.0U L=1.0U
+ AD=2.2P PD=4.0U AS=4.2P PS=19.0U
M7 18 7 1 1 PMOS W=2.0U L=1.0U
+ AD=2.2P PD=4.0U AS=4.2P PS=19.0U
M8 19 19 12 1 PMOS W=2.0U L=1.0U
+ AD=4.5P PD=8.0U AS=0.0P PS=4.0U
M9 20 19 13 1 PMOS W=2.0U L=1.0U
+ AD=4.5P PD=8.0U AS=0.0P PS=4.0U
M10 21 19 14 1 PMOS W=2.0U L=1.0U
+ AD=4.5P PD=8.0U AS=0.0P PS=4.0U
M11 22 23 15 1 PMOS W=2.0U L=1.0U
+ AD=0.0P PD=13.3U AS=0.0P PS=4.0U
M12 23 23 16 1 PMOS W=2.0U L=1.0U
+ AD=4.5P PD=8.0U AS=0.0P PS=4.0U
M13 24 21 22 1 PMOS W=2.0U L=1.0U
+ AD=13.5P PD=8.0U AS=0.0P PS=13.3U
M14 25 10 22 1 PMOS W=2.0U L=1.0U
+ AD=13.5P PD=8.0U AS=0.0P PS=13.3U
M15 26 18 1 1 PMOS W=2.0U L=1.0U
+ AD=2.2P PD=4.0U AS=4.2P PS=19.0U
M16 27 17 1 1 PMOS W=2.0U L=1.0U
+ AD=2.2P PD=4.0U AS=4.2P PS=19.0U
M17 1 18 17 1 PMOS W=2.0U L=1.0U
+ AD=4.2P PD=19.0U AS=2.2P PS=4.0U
M18 1 17 18 1 PMOS W=2.0U L=1.0U
+ AD=4.2P PD=19.0U AS=2.2P PS=4.0U
M19 1 27 26 1 PMOS W=2.0U L=1.0U
+ AD=4.2P PD=19.0U AS=2.2P PS=4.0U
M20 1 26 27 1 PMOS W=2.0U L=1.0U
+ AD=4.2P PD=19.0U AS=2.2P PS=4.0U
M21 1 27 28 1 PMOS W=16.0U L=1.0U
+ AD=33.5P PD=151.6U AS=11.5P PS=32.0U
M22 9 20 19 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M23 11 20 20 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M24 24 7 17 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M25 25 7 18 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M26 30 27 26 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M27 31 26 27 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M28 32 27 8 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M29 28 3 4 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.8U AS=0.0P PS=8.0U
```

```
M30 33 6 21 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M31 24 5 25 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.0U AS=0.0P PS=8.0U
M32 2 26 32 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M33 2 32 33 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=4.0U
M34 2 23 23 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M35 2 25 24 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M36 2 24 25 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M37 2 18 30 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M38 2 17 31 29 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=36.0U AS=0.0P PS=8.0U
M39 2 27 28 29 NMOS W=8.0U L=1.0U
+ AD=0.0P PD=144.0U AS=0.0P PS=19.2U

C1 2 21 164.8F
C2 27 17 1.4F
C3 5 3 6.5F
C4 5 27 1.3F
C5 2 5 5.7F
C6 25 24 2.9F
C7 11 8 4.8F
C8 26 17 1.1F
C9 3 7 4.8F
C10 25 23 1.4F
C11 6 10 4.8F
C12 8 26 1.2F
C13 3 1 1.3F
C14 10 7 4.8F
C15 17 18 4.0F
C16 2 27 8.1F
C17 26 1 4.7F
C18 4 21 2.5F
C19 6 8 5.4F
C20 27 26 3.4F
C21 5 4 2.6F
C22 17 1 1.1F
C23 11 26 2.9F
C24 27 21 1.6F
C25 21 0 18.9F
C26 19 0 4.3F
C27 11 0 22.2F
C28 12 0 2.9F
C29 31 0 1.2F
C30 18 0 14.0F
C31 3 0 21.5F
C32 20 0 3.9F
C33 23 0 6.7F
C34 16 0 2.5F
C35 8 0 20.2F
C36 24 0 9.2F
C37 6 0 22.1F
C38 28 0 3.9F
C39 17 0 17.7F
C40 22 0 1.7F
C41 32 0 3.3F
C42 27 0 23.7F
C43 5 0 27.3F
C44 26 0 26.7F
C45 30 0 1.2F
C46 4 0 11.7F
C47 9 0 14.3F
C48 7 0 27.0F
C49 2 0 121.0F
C50 1 0 36.6F
C51 10 0 21.1F
C52 25 0 9.
.ENDS

***** Subcircuit from file ./delta_Ibias.ext
.SUBCKT delta_Ibias 1 2 3 4 5

M1 3 3 4 6 NMOS W=128.0U L=1.0U
+ AD=0.0P PD=512.0U AS=0.0P PS=512.0U
M2 4 4 5 6 NMOS W=128.0U L=1.0U
+ AD=0.0P PD=512.0U AS=0.0P PS=512.0U

C1 4 3 9.7F
C2 4 5 9.6F
C3 3 0 61.9F
C4 4 0 72.4F
C5 5 0 45.6F
.ENDS

***** Subcircuit from file ./driver_delta.ext
.SUBCKT driver_delta 1 2 3 4

M1 1 3 5 1 PMOS W=6.0U L=1.0U
+ AD=7.2P PD=18.0U AS=5.5P PS=24.0U
M2 1 5 4 1 PMOS W=18.0U L=1.0U
+ AD=21.8P PD=54.0U AS=11.5P PS=48.0U
M3 2 3 5 6 NMOS W=3.0U L=1.0U
+ AD=0.0P PD=9.0U AS=0.0P PS=12.0U
M4 2 5 4 6 NMOS W=9.0U L=1.0U
+ AD=0.0P PD=27.0U AS=0.0P PS=24.0U

C1 3 0 2.7F
C2 4 0 5.7F
```

```
C3 5 0 8.4F
C4 2 0 12.5F
C5 1 0 13.5F
.ENDS

*****  Subcircuit from file ./register1.ext
.SUBCKT register1 1 2 3 4 5 6 7 8 9 10 11 12 13

M1 1 11 14 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M2 1 15 10 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M3 1 16 17 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M4 1 18 9 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M5 1 19 20 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M6 1 21 8 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M7 1 22 23 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M8 1 24 7 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M9 1 25 26 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M10 1 27 6 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M11 1 28 29 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M12 1 30 5 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M13 1 31 32 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M14 1 33 4 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=26.0P PS=160.0U
M15 1 34 35 1 PMOS W=20.0U L=1.0U
+ AD=8.3P PD=60.0U AS=13.0P PS=80.0U
M16 1 36 3 1 PMOS W=60.0U L=1.0U
+ AD=25.0P PD=180.0U AS=25.5P PS=160.0U
M17 2 11 14 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M18 14 12 15 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M19 2 15 10 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M20 10 13 16 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M21 2 16 17 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M22 17 12 18 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M23 2 18 9 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M24 9 13 19 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M25 2 19 20 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M26 20 12 21 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M27 2 21 8 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M28 8 13 22 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M29 2 22 23 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M30 23 12 24 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M31 2 24 7 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M32 7 13 25 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M33 2 25 26 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M34 26 12 27 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M35 2 27 6 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M36 6 13 28 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M37 2 28 29 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M38 29 12 30 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M39 2 30 5 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M40 5 13 31 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M41 2 31 32 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M42 32 12 33 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M43 2 33 4 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=60.0U
M44 4 13 34 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M45 2 34 35 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=30.0U AS=0.0P PS=20.0U
M46 35 12 36 37 NMOS W=10.0U L=1.0U
+ AD=0.0P PD=20.0U AS=0.0P PS=40.0U
M47 2 36 3 37 NMOS W=30.0U L=1.0U
+ AD=0.0P PD=90.0U AS=0.0P PS=80.0U

C1 9 1 3.0F
```

```
C2 7 1 3.0F
C3 5 1 3.0F
C4 3 1 3.0F
C5 2 13 17.1F
C6 10 1 3.0F
C7 2 12 11.7F
C8 8 1 3.0F
C9 6 1 3.0F
C10 4 1 3.0F
C11 13 12 9.3F
C12 11 2 1.2F
C13 26 0 5.5F
C14 17 0 5.5F
C15 20 0 5.5F
C16 23 0 5.5F
C17 15 0 8.4F
C18 12 0 54.4F
C19 13 0 46.1F
C20 36 0 8.4F
C21 30 0 8.4F
C22 33 0 8.4F
C23 27 0 8.4F
C24 31 0 4.6F
C25 18 0 8.4F
C26 21 0 8.4F
C27 24 0 8.4F
C28 28 0 4.6F
C29 16 0 4.6F
C30 29 0 5.5F
C31 11 0 6.2F
C32 22 0 4.6F
C33 10 0 15.0F
C34 9 0 15.0F
C35 8 0 15.0F
C36 7 0 15.0F
C37 6 0 15.0F
C38 5 0 15.0F
C39 4 0 15.0F
C40 3 0 15.1F
C41 34 0 4.6F
C42 25 0 4.6F
C43 2 0 137.2F
C44 35 0 5.5F
C45 1 0 171.1F
C46 14 0 5.5F
C47 19 0 4.6F
C48 32 0 5.
.ENDS

***** Subcircuit from file ./avdd.ext
.SUBCKT avdd 1 2 3 4 5

C1 4 3 231.3F
C2 4 5 122.9F
C3 3 0 359.5F
C4 5 0 1266.4F
C5 4 0 316.7F
*** Node Listing for subckt:  avdd
**  0     Node 0 is the global ground node
**  1     Vdd!
**  2     GND!
**  3     GND_ring
**  4     Vdd_ring
**  5     w_80_202#
.ENDS

*****  Subcircuit from file ./agnd1.ext
.SUBCKT agnd1 1 2 3 4 5 6

C1 5 3 83.8F
C2 5 4 156.8F
C3 3 4 74.5F
C4 5 6 122.9F
C5 5 0 31.0F
C6 4 0 359.5F
C7 6 0 1266.4F
C8 3 0 257.1F
.ENDS

*****  Subcircuit from file ./register_new.ext
.SUBCKT register_new 1 2 3 4 5 6 7 8 9 10 11 12
13 14

xadder_0 1 2 15 16 13 adder
xadder_1 1 2 17 16 5 adder

M1 16 11 1 1 PMOS W=4.0U L=1.0U
+ AD=4.5P PD=16.0U AS=13.3P PS=15.1U
M2 1 18 19 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M3 1 20 21 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M4 1 22 23 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M5 1 24 25 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M6 1 26 27 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M7 1 28 29 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M8 1 30 31 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M9 1 32 33 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
```

```
M10 1 34 35 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M11 1 36 37 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M12 1 38 39 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M13 1 40 41 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M14 1 42 43 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M15 1 44 45 1 PMOS W=4.0U L=1.0U
+ AD=13.3P PD=15.1U AS=5.0P PS=16.0U
M16 1 46 4 1 PMOS W=12.0U L=1.0U
+ AD=39.8P PD=45.3U AS=9.5P PS=32.0U
M17 18 11 12 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=14.0U AS=0.0P PS=8.0U
M18 22 11 3 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M19 26 11 14 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M20 30 11 10 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M21 34 11 9 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M22 38 11 8 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M23 42 11 7 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M24 46 11 6 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.0U AS=0.0P PS=8.0U
M25 2 18 19 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M26 19 17 20 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M27 2 20 21 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M28 21 15 22 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M29 2 22 23 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M30 23 17 24 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M31 2 24 25 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M32 25 15 26 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M33 2 26 27 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M34 27 17 28 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M35 2 28 29 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M36 29 15 30 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M37 2 30 31 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M38 31 17 32 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M39 2 32 33 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M40 33 15 34 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M41 2 34 35 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M42 35 17 36 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M43 2 36 37 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M44 37 15 38 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M45 2 38 39 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M46 39 17 40 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M47 2 40 41 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M48 41 15 42 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M49 2 42 43 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M50 43 17 44 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=8.0U
M51 2 44 45 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=4.0U
M52 45 16 46 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=4.0U AS=0.0P PS=7.0U
M53 2 46 4 47 NMOS W=6.0U L=1.0U
+ AD=0.0P PD=22.7U AS=0.0P PS=16.0U
M54 2 11 16 47 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=7.6U AS=0.0P PS=8.0U

C1 7 11 1.2F
C2 12 1 1.2F
C3 2 17 10.2F
C4 14 1 1.2F
C5 8 1 1.2F
C6 13 16 1.3F
C7 10 11 1.2F
C8 6 11 1.2F
C9 12 11 1.2F
C10 9 1 1.2F
C11 14 11 1.2F
C12 8 11 1.2F
C13 3 1 1.2F
```

```
C14 2 15 13.3F
C15 15 17 8.4F
C16 7 1 1.2F
C17 2 16 1.2F
C18 9 11 1.2F
C19 11 1 13.2F
C20 2 16 1.4F
C21 10 1 1.2F
C22 3 11 1.2F
C23 17 16 8.8F
C24 6 1 1.2F
C25 12 0 3.1F
C26 3 0 3.1F
C27 35 0 2.8F
C28 14 0 3.1F
C29 10 0 3.1F
C30 9 0 3.1F
C31 8 0 3.1F
C32 7 0 3.1F
C33 6 0 3.1F
C34 24 0 4.4F
C35 44 0 4.4F
C36 5 0 4.1F
C37 13 0 3.6F
C38 18 0 4.9F
C39 22 0 4.9F
C40 26 0 4.9F
C41 30 0 4.9F
C42 34 0 4.9F
C43 38 0 4.9F
C44 42 0 4.9F
C45 46 0 8.7F
C46 28 0 4.4F
C47 23 0 2.8F
C48 40 0 4.4F
C49 43 0 2.8F
C50 25 0 2.8F
C51 45 0 2.8F
C52 15 0 43.9F
C53 32 0 4.4F
C54 21 0 2.8F
C55 17 0 60.0F
C56 29 0 2.8F
C57 27 0 2.8F
C58 37 0 2.8F
C59 19 0 2.8F
C60 36 0 4.4F
C61 39 0 2.8F
C62 41 0 2.8F
C63 4 0 6.5F
C64 33 0 2.8F
C65 31 0 2.8F
C66 2 0 95.7F
C67 11 0 76.8F
C68 1 0 99.9F
C69 16 0 41.7F
C70 20 0 4.
.ENDS

***** Subcircuit from file ./avdd1.ext
.SUBCKT avdd1 1 2 3 4 5 6

C1 6 3 83.8F
C2 6 4 156.8F
C3 3 4 74.5F
C4 6 5 122.9F
C5 6 0 31.0F
C6 4 0 359.5F
C7 5 0 1266.4F
C8 3 0 257.1F
*** Node Listing for subckt:  avdd1
**  0     Node 0 is the global ground node
**  1     Vdd!
**  2     GND!
**  3     Vdd_ring
**  4     GND_ring
**  5     w_80_202#
**  6     m2_84_206#
.ENDS

***** Subcircuit from file ./jung2.ext
.SUBCKT jung2 1 2 3 4 5 6 7

M1 1 3 5 1 PMOS W=720.0U L=1.0U
+ AD=432.4P PD=7316.1U AS=0.0P PS=8943.2U
M2 1 7 5 1 PMOS W=800.0U L=1.0U
+ AD=480.4P PD=8129.0U AS=0.0P PS=9936.8U
M3 1 7 7 1 PMOS W=80.0U L=1.0U
+ AD=48.0P PD=812.9U AS=42.9P PS=811.6U
M4 1 8 9 1 PMOS W=24.0U L=1.0U
+ AD=14.4P PD=243.9U AS=14.5P PS=112.0U
M5 7 9 1 1 PMOS W=3.0U L=1.0U
+ AD=1.6P PD=30.4U AS=1.8P PS=30.5U
M6 8 6 1 1 PMOS W=4.0U L=1.0U
+ AD=4.5P PD=16.0U AS=2.4P PS=40.6U
M7 3 3 1 1 PMOS W=80.0U L=1.0U
+ AD=0.0P PD=1600.0U AS=48.0P PS=812.9U
M8 10 9 7 11 NMOS W=40.0U L=1.0U
+ AD=0.0P PD=83.0U AS=0.0P PS=166.0U
M9 2 6 8 11 NMOS W=2.0U L=1.0U
+ AD=0.0P PD=8.2U AS=0.0P PS=8.0U
M10 2 8 9 11 NMOS W=12.0U L=1.0U
+ AD=0.0P PD=49.0U AS=0.0P PS=48.0U
M11 2 4 4 11 NMOS W=20.0U L=1.0U
```

```
+ AD=0.0P PD=81.6U AS=0.0P PS=80.0U
M12 2 4 10 11 NMOS W=40.0U L=1.0U
+ AD=0.0P PD=163.2U AS=0.0P PS=83.0U

C1 9 1 9.4F
C2 7 1 63.8F
C3 7 10 1.1F
C4 3 1 47.6F
C5 2 4 1.8F
C6 5 1 499.4F
C7 8 1 3.4F
C8 2 10 1.1F
C9 6 1 1.6F
C10 10 0 6.5F
C11 6 0 2.8F
C12 7 0 27.7F
C13 4 0 10.0F
C14 3 0 30.6F
C15 8 0 3.5F
C16 9 0 10.0F
C17 5 0 376.5F
C18 2 0 35.3F
C19 1 0 9881.1F
.ENDS

*****  Subcircuit from file ./agnd.ext
.SUBCKT agnd 1 2 3 4 5

C1 3 5 158.3F
C2 5 4 122.9F
C3 5 0 443.4F
C4 4 0 1266.4F
C5 3 0 257.1F
.ENDS

*****  Subcircuit from file ./in_ub.ext
.SUBCKT in_ub 1 2 3 4 5

C1 3 5 45.1F
C2 4 3 13.3F
*C3 6 3 22.2F
*C4 3 7 240.9F
C5 4 5 187.8F
*C6 3 8 22.2F
*C7 6 5 12.0F
*C8 5 8 12.0F
C9 3 0 458.7F
C10 5 0 624.9F
*C11 7 0 1266.4F
*C12 8 0 16.9F
C13 4 0 326.3F
*C14 6 0 16.9F
.ENDS

*****  Subcircuit from file ./cv.ext
.SUBCKT cv 1 2 3 4 5

C1 5 3 208.9F
C2 5 4 124.1F
C3 4 0 1280.6F
C4 3 0 254.4F
C5 5 0 172.7F
.ENDS

****** top level cell is ./si_delta5.ext
xavdd_1 1 2 2 1 3 avdd
xagnd_3 1 2 1 4 2 agnd
xin_ub_5 1 2 5 1 2 in_ub
xin_ub_6 1 2 6 1 2 in_ub
xin_ub_7 1 2 7 1 2 in_ub
xin_ub_8 1 2 8 1 2 in_ub
xin_ub_16 1 2 9 1 2 in_ub
xdelta_1[7][7] 1 2 10 11 8 9 7 12 13 6 5 delta_sub2
xdelta_1[7][6] 1 2 10 14 8 9 7 12 15 6 5 delta_sub2
xdelta_1[7][5] 1 2 10 16 8 9 7 12 17 6 5 delta_sub2
xdelta_1[7][4] 1 2 10 18 8 9 7 12 19 6 5 delta_sub2
xdelta_1[7][3] 1 2 10 20 8 9 7 12 21 6 5 delta_sub2
xdelta_1[7][2] 1 2 10 22 8 9 7 12 23 6 5 delta_sub2
xdelta_1[7][1] 1 2 10 24 8 9 7 12 25 6 5 delta_sub2
xdelta_1[7][0] 1 2 10 26 8 9 7 12 27 6 5 delta_sub2
xdelta_1[6][7] 1 2 28 11 8 9 7 12 29 6 5 delta_sub2
xdelta_1[6][6] 1 2 28 14 8 9 7 12 30 6 5 delta_sub2
xdelta_1[6][5] 1 2 28 16 8 9 7 12 31 6 5 delta_sub2
xdelta_1[6][4] 1 2 28 18 8 9 7 12 32 6 5 delta_sub2
xdelta_1[6][3] 1 2 28 20 8 9 7 12 33 6 5 delta_sub2
xdelta_1[6][2] 1 2 28 22 8 9 7 12 34 6 5 delta_sub2
xdelta_1[6][1] 1 2 28 24 8 9 7 12 35 6 5 delta_sub2
xdelta_1[6][0] 1 2 28 26 8 9 7 12 36 6 5 delta_sub2
xdelta_1[5][7] 1 2 37 11 8 9 7 12 38 6 5 delta_sub2
xdelta_1[5][6] 1 2 37 14 8 9 7 12 39 6 5 delta_sub2
xdelta_1[5][5] 1 2 37 16 8 9 7 12 40 6 5 delta_sub2
xdelta_1[5][4] 1 2 37 18 8 9 7 12 41 6 5 delta_sub2
xdelta_1[5][3] 1 2 37 20 8 9 7 12 42 6 5 delta_sub2
xdelta_1[5][2] 1 2 37 22 8 9 7 12 43 6 5 delta_sub2
xdelta_1[5][1] 1 2 37 24 8 9 7 12 44 6 5 delta_sub2
xdelta_1[5][0] 1 2 37 26 8 9 7 12 45 6 5 delta_sub2
xdelta_1[4][7] 1 2 46 11 8 9 7 12 47 6 5 delta_sub2
xdelta_1[4][6] 1 2 46 14 8 9 7 12 48 6 5 delta_sub2
xdelta_1[4][5] 1 2 46 16 8 9 7 12 49 6 5 delta_sub2
xdelta_1[4][4] 1 2 46 18 8 9 7 12 50 6 5 delta_sub2
xdelta_1[4][3] 1 2 46 20 8 9 7 12 51 6 5 delta_sub2
xdelta_1[4][2] 1 2 46 22 8 9 7 12 52 6 5 delta_sub2
xdelta_1[4][1] 1 2 46 24 8 9 7 12 53 6 5 delta_sub2
xdelta_1[4][0] 1 2 46 26 8 9 7 12 54 6 5 delta_sub2
```

```
xdelta_1[3][7] 1 2 55 11 8 9 7 12 56 6 5 delta_sub2
xdelta_1[3][6] 1 2 55 14 8 9 7 12 57 6 5 delta_sub2
xdelta_1[3][5] 1 2 55 16 8 9 7 12 58 6 5 delta_sub2
xdelta_1[3][4] 1 2 55 18 8 9 7 12 59 6 5 delta_sub2
xdelta_1[3][3] 1 2 55 20 8 9 7 12 60 6 5 delta_sub2
xdelta_1[3][2] 1 2 55 22 8 9 7 12 61 6 5 delta_sub2
xdelta_1[3][1] 1 2 55 24 8 9 7 12 62 6 5 delta_sub2
xdelta_1[3][0] 1 2 55 26 8 9 7 12 63 6 5 delta_sub2
xdelta_1[2][7] 1 2 64 11 8 9 7 12 65 6 5 delta_sub2
xdelta_1[2][6] 1 2 64 14 8 9 7 12 66 6 5 delta_sub2
xdelta_1[2][5] 1 2 64 16 8 9 7 12 67 6 5 delta_sub2
xdelta_1[2][4] 1 2 64 18 8 9 7 12 68 6 5 delta_sub2
xdelta_1[2][3] 1 2 64 20 8 9 7 12 69 6 5 delta_sub2
xdelta_1[2][2] 1 2 64 22 8 9 7 12 70 6 5 delta_sub2
xdelta_1[2][1] 1 2 64 24 8 9 7 12 71 6 5 delta_sub2
xdelta_1[2][0] 1 2 64 26 8 9 7 12 72 6 5 delta_sub2
xdelta_1[1][7] 1 2 73 11 8 9 7 12 74 6 5 delta_sub2
xdelta_1[1][6] 1 2 73 14 8 9 7 12 75 6 5 delta_sub2
xdelta_1[1][5] 1 2 73 16 8 9 7 12 76 6 5 delta_sub2
xdelta_1[1][4] 1 2 73 18 8 9 7 12 77 6 5 delta_sub2
xdelta_1[1][3] 1 2 73 20 8 9 7 12 78 6 5 delta_sub2
xdelta_1[1][2] 1 2 73 22 8 9 7 12 79 6 5 delta_sub2
xdelta_1[1][1] 1 2 73 24 8 9 7 12 80 6 5 delta_sub2
xdelta_1[1][0] 1 2 73 26 8 9 7 12 81 6 5 delta_sub2
xdelta_1[0][7] 1 2 82 11 8 9 7 12 83 6 5 delta_sub2
xdelta_1[0][6] 1 2 82 14 8 9 7 12 84 6 5 delta_sub2
xdelta_1[0][5] 1 2 82 16 8 9 7 12 85 6 5 delta_sub2
xdelta_1[0][4] 1 2 82 18 8 9 7 12 86 6 5 delta_sub2
xdelta_1[0][3] 1 2 82 20 8 9 7 12 87 6 5 delta_sub2
xdelta_1[0][2] 1 2 82 22 8 9 7 12 88 6 5 delta_sub2
xdelta_1[0][1] 1 2 82 24 8 9 7 12 89 6 5 delta_sub2
xdelta_1[0][0] 1 2 82 26 8 9 7 12 90 6 5 delta_sub2
xdelta_Ibias_1 1 2 9 12 2 delta_Ibias
xdriver_1[7] 1 2 11 91 driver_delta
xdriver_1[6] 1 2 14 92 driver_delta
xdriver_1[5] 1 2 16 93 driver_delta
xdriver_1[4] 1 2 18 94 driver_delta
xdriver_1[3] 1 2 20 95 driver_delta
xdriver_1[2] 1 2 22 96 driver_delta
xdriver_1[1] 1 2 24 97 driver_delta
xdriver_1[0] 1 2 26 98 driver_delta
xregister1_1 1 2 10 28 37 46 55 64 73 82 99 100
101 register1
xavdd_0 1 2 2 1 102 avdd
xin_ub_14 1 2 99 103 104 in_ub
xin_ub_13 1 2 100 103 104 in_ub
xin_ub_12 1 2 101 103 104 in_ub
xagnd_2 1 2 1 105 2 agnd
xin_ub_3 1 2 106 1 2 in_ub
xagnd1_0 1 2 103 104 107 108 agnd1
xregister_new_1 1 2 97 109 110 91 92 93 94 95
106 98 111 96 register_new
xin_ub_2 1 2 110 1 2 in_ub
xin_ub_1 1 2 111 1 2 in_ub
xavdd1_0 1 2 103 104 112 113 avdd1
xin_ub_9 1 2 114 103 104 in_ub
xjung2_1 1 2 115 116 117 109 118 jung2
xagnd_1 1 2 1 119 2 agnd
xin_ub_0 1 2 116 1 2 in_ub
xcv_0 1 2 104 120 103 cv
xin_ub_15 1 2 121 103 104 in_ub
xin_ub_4 1 2 115 1 2 in_ub
xcv_1 1 2 2 122 1 cv

C1 103 120 11.6F
C2 120 103 9.9F
C3 1 122 10.0F
C4 122 1 12.3F
C5 120 103 5.4F
C6 120 103 4.6F
C7 122 1 4.1F
C8 1 122 5.1F
C9 1 1 1.3F
C10 1 117 1.2F
C11 109 1 1.5F
C12 109 1 3.2F
C13 113 112 13.9F
C14 112 113 10.3F
C15 109 1 3.2F
C16 119 2 14.0F
C17 119 2 15.2F
C18 113 112 4.2F
C19 113 112 3.1F
C20 109 1 3.2F
C21 2 119 1.1F
C22 119 2 1.2F
C23 108 107 11.2F
C24 108 107 5.3F
C25 108 107 10.1F
C26 108 107 4.8F
C27 2 105 3.8F
C28 2 105 10.4F
C29 2 105 4.7F
C30 105 2 12.6F
C31 1 102 1.9F
C32 1 1 1.4F
C33 1 102 8.4F
C34 102 1 20.4F
C35 3 1 1.4F
C36 3 1 6.3F
C37 4 2 1.7F
C38 1 3 4.2F
C39 1 3 19.6F
C40 2 4 21.0F
```

C41 2 4 8.2F
C42 8 9 5.2F
C43 7 2 35.1F
C44 92 91 46.4F
C45 117 123 124.5F
C46 5 12 37.9F
C47 6 8 6.8F
C48 55 1 3.0F
C49 2 98 1.2F
C50 2 93 1.2F
C51 116 1 9.6F
C52 2 106 2.4F
C53 8 2 38.4F
C54 12 9 39.2F
C55 2 94 1.2F
C56 124 104 122.2F
C57 1 2 22.3F
C58 7 8 23.7F
C59 6 12 4.0F
C60 5 9 5.9F
C61 46 1 3.4F
C62 2 91 1.2F
C63 2 95 1.2F
C64 1 10 4.0F
C65 5 6 5.9F
C66 99 100 20.1F
C67 82 1 3.0F
C68 94 93 57.2F
C69 2 96 1.2F
C70 1 98 1.2F
C71 12 2 22.3F
C72 1 93 1.2F
C73 2 97 1.2F
C74 1 106 4.3F
C75 7 12 4.0F
C76 6 9 40.1F
C77 5 2 22.3F
C78 37 1 3.0F
C79 1 94 1.2F
C80 95 94 61.5F
C81 101 100 30.4F
C82 5 7 5.9F
C83 73 1 3.0F
C84 2 92 1.2F
C85 1 91 1.2F
C86 1 95 1.2F
C87 97 98 74.2F
C88 9 2 22.3F
C89 110 111 1.3F
C90 1 96 1.2F
C91 8 12 4.8F
C92 7 9 4.0F
C93 6 2 28.9F
C94 28 1 3.0F
C95 96 95 65.7F
C96 93 92 53.0F
C97 1 97 1.2F
C98 6 7 43.9F
C99 5 8 6.7F
C100 64 1 3.0F
C101 97 96 70.0F
C102 1 92 1.2F
C103 124 0 1280.6F
C104 26 0 4.4F
C105 83 0 43.2F
C106 84 0 43.2F
C107 85 0 43.2F
C108 86 0 43.2F
C109 87 0 43.2F
C110 74 0 43.2F
C111 88 0 43.2F
C112 75 0 43.2F
C113 89 0 43.2F
C114 76 0 43.2F
C115 90 0 43.2F
C116 98 0 178.2F
C117 77 0 43.2F
C118 97 0 169.0F
C119 78 0 43.2F
C120 96 0 159.7F
C121 65 0 43.2F
C122 79 0 43.2F
C123 95 0 150.7F
C124 66 0 43.2F
C125 80 0 43.2F
C126 94 0 141.6F
C127 67 0 43.2F
C128 81 0 43.2F
C129 93 0 132.1F
C130 68 0 43.2F
C131 22 0 4.3F
C132 92 0 123.0F
C133 69 0 43.2F
C134 91 0 108.3F
C135 56 0 43.2F
C136 70 0 43.2F
C137 82 0 56.1F
C138 57 0 43.2F
C139 71 0 43.2F
C140 73 0 43.3F
C141 58 0 43.2F
C142 72 0 43.2F
C143 64 0 30.6F
C144 59 0 43.2F

C145 55 0 18.0F
C146 60 0 43.2F
C147 46 0 24.7F
C148 47 0 43.2F
C149 61 0 43.2F
C150 37 0 37.3F
C151 48 0 43.2F
C152 62 0 43.2F
C153 28 0 50.0F
C154 49 0 43.2F
C155 63 0 43.2F
C156 10 0 62.6F
C157 50 0 43.2F
C158 51 0 43.2F
C159 38 0 43.2F
C160 52 0 43.2F
C161 39 0 43.2F
C162 53 0 43.2F
C163 40 0 43.2F
C164 54 0 43.2F
C165 116 0 115.2F
C166 41 0 43.2F
C167 115 0 91.3F
C168 42 0 43.2F
C169 29 0 43.2F
C170 43 0 43.2F
C171 30 0 43.2F
C172 44 0 43.2F
C173 18 0 4.2F
C174 31 0 43.2F
C175 45 0 43.2F
C176 32 0 43.2F
C177 33 0 43.2F
C178 13 0 43.2F
C179 34 0 43.2F
C180 15 0 43.2F
C181 35 0 43.2F
C182 17 0 43.2F
C183 36 0 43.2F
C184 19 0 43.2F
C185 21 0 43.2F
C186 23 0 43.2F
C187 25 0 43.2F
C188 27 0 43.2F
C189 14 0 4.4F
C190 99 0 151.2F
C191 7 0 279.0F
C192 111 0 121.8F
C193 110 0 82.8F
C194 9 0 299.8F
C195 12 0 184.2F
C196 106 0 109.9F
C197 6 0 278.1F
C198 104 0 51.1F
C199 107 0 70.8F
C200 113 0 70.8F
C201 114 0 70.8F
C202 121 0 70.8F
C203 24 0 4.2F
C204 1 0 2065.9F
C205 20 0 4.2F
C206 109 0 31.6F
C207 16 0 4.2F
C208 8 0 294.3F
C209 11 0 4.3F
C210 123 0 1280.6F
C211 2 0 1532.2F
C212 5 0 257.3F
C213 100 0 203.3F
C214 101 0 247.1F
C215 103 0 70.8F
C216 117 0 342.0F
*** Node Listing for subckt: si_delta5
** 0 Node 0 is the global ground node
** 1 delta_Vdd
** 1 jung_Vdd
** 1 mux_Vdd
** 1 Vdd!
** 2 delta_Vss
** 2 jung_Vss
** 2 mux_Vss
** 2 GND!
** 3 avdd_1/w_80_202#
** 4 agnd_3/w_80_202#
** 5 delta_vbias
** 6 delta_vref
** 7 delta_set
** 8 delta_reset
** 9 delta_ibias
** 10 mux_out8
** 11 delta_driver_7
** 12 delta_Ibias_1/delta_Ibias_mid
** 13 I_81
** 14 delta_driver_6
** 15 I_82
** 16 delta_driver_5
** 17 I_83
** 18 delta_driver_4
** 19 I_84
** 20 delta_driver_3
** 21 I_85
** 22 delta_driver_2
** 23 I_86
** 24 delta_driver_1

```
** 25   I_87
** 26   delta_driver_0
** 27   I_88
** 28   mux_out7
** 29   I_71
** 30   I_72
** 31   I_73
** 32   I_74
** 33   I_75
** 34   I_76
** 35   I_77
** 36   I_78
** 37   mux_out6
** 38   I_61
** 39   I_62
** 40   I_63
** 41   I_64
** 42   I_65
** 43   I_66
** 44   I_67
** 45   I_68
** 46   mux_out5
** 47   I_51
** 48   I_52
** 49   I_53
** 50   I_54
** 51   I_55
** 52   I_56
** 53   I_57
** 54   I_58
** 55   mux_out4
** 56   I_41
** 57   I_42
** 58   I_43
** 59   I_44
** 60   I_45
** 61   I_46
** 62   I_47
** 63   I_48
** 64   mux_out3
** 65   I_31
** 66   I_32
** 67   I_33
** 68   I_34
** 69   I_35
** 70   I_36
** 71   I_37
** 72   I_38
** 73   mux_out2
** 74   I_21
** 75   I_22
** 76   I_23
** 77   I_24
** 78   I_25
** 79   I_26
** 80   I_27
** 81   I_28
** 82   mux_out1
** 83   I_11
** 84   I_12
** 85   I_13
** 86   I_14
** 87   I_15
** 88   I_16
** 89   I_17
** 90   I_18
** 91   shift_8
** 92   shift_7
** 93   shift_6
** 94   shift_5
** 95   shift_4
** 96   shift_3
** 97   shift_2
** 98   shift_1
** 99   mux_vin
** 100  mux_clock+
** 101  mux_clock-
** 103  Vdd_ring
** 104  GND_ring
** 106  shift_read
** 107  decoy1
** 109  jung2_1/in
** 109  emitter_in
** 110  shift_clock+
** 111  shift_clock-
** 113  decoy2
** 114  decoy3
** 115  jung_b2
** 115  jung2_1/Ib2
** 116  jung_b1
** 117  jung_out
** 121  decoy4
.END
```

F.3 Additional simulation results

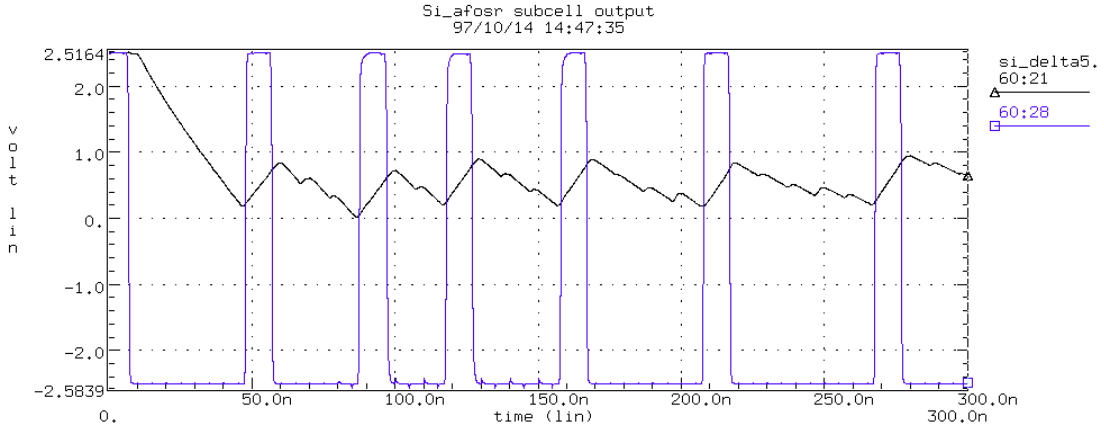


Figure F.1 Sub-cell output

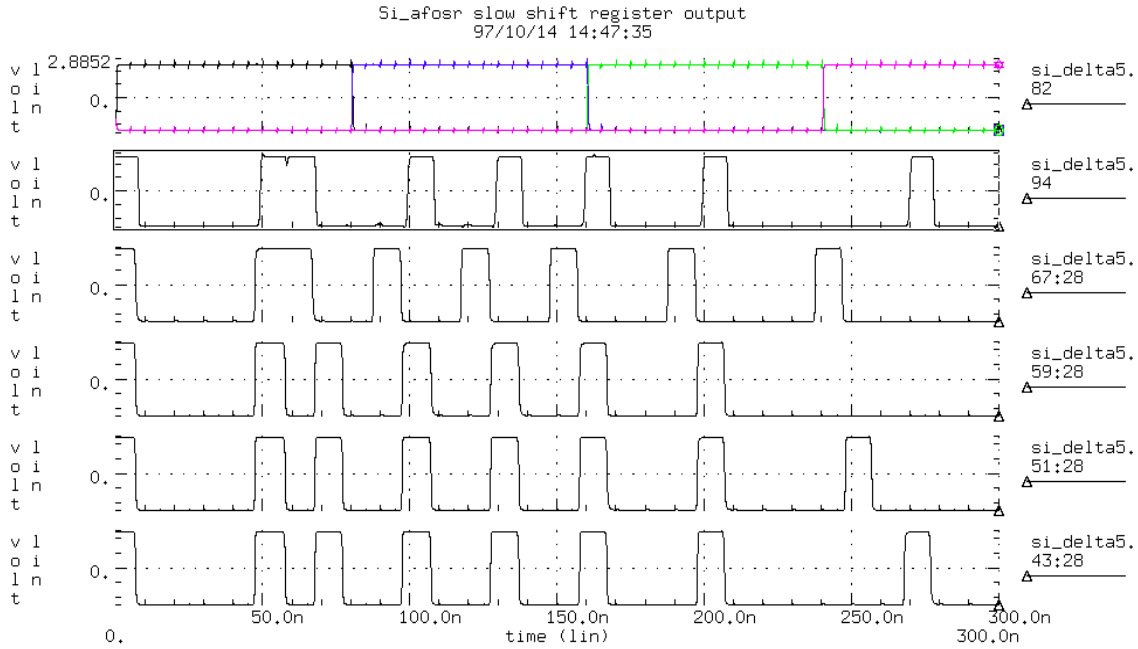


Figure F.2 Slow shift register output

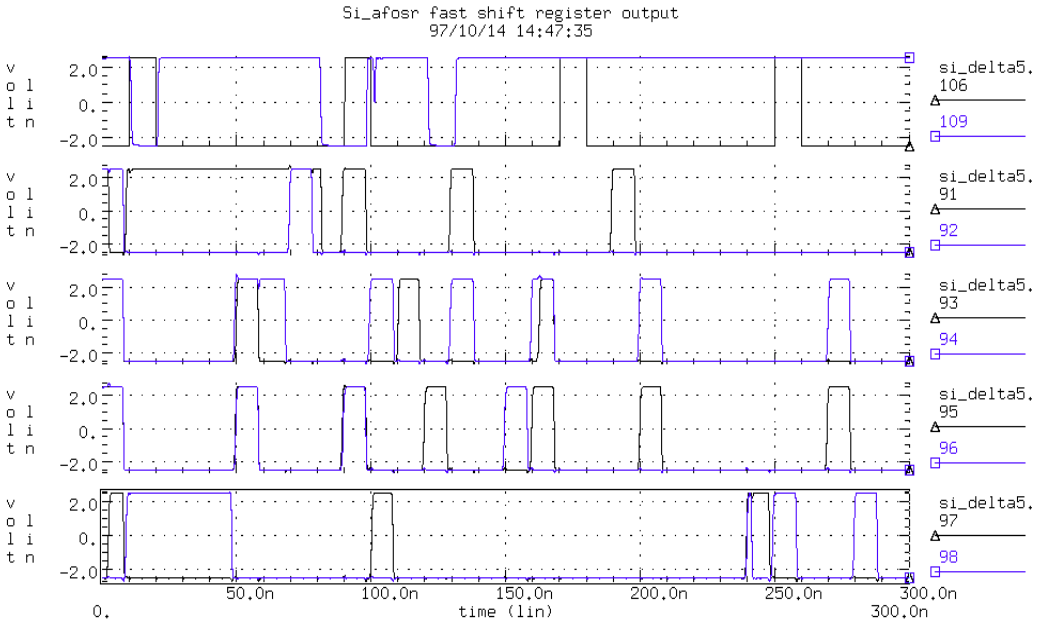


Figure F.3 Slow shift register output

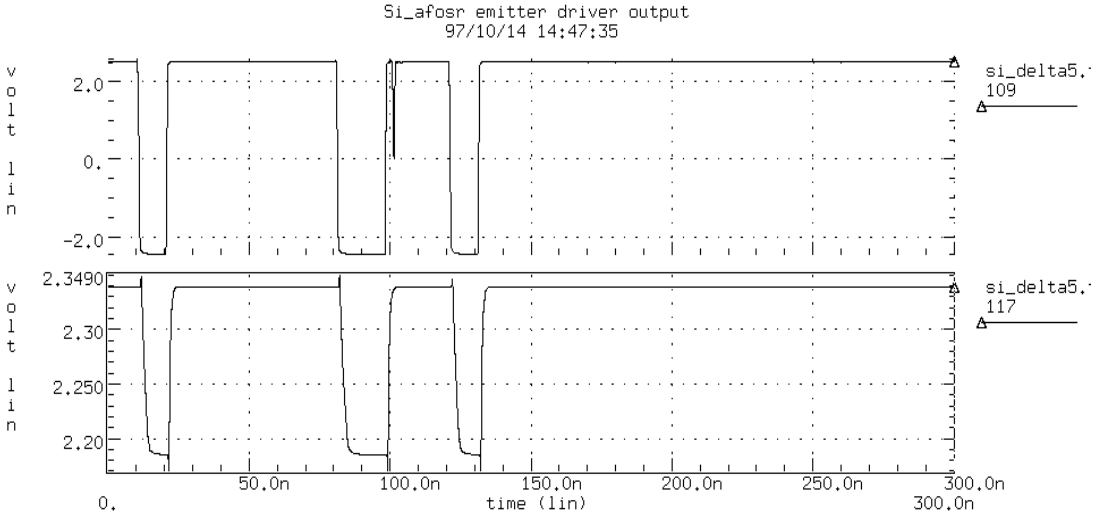


Figure F.4 Emitter driver output

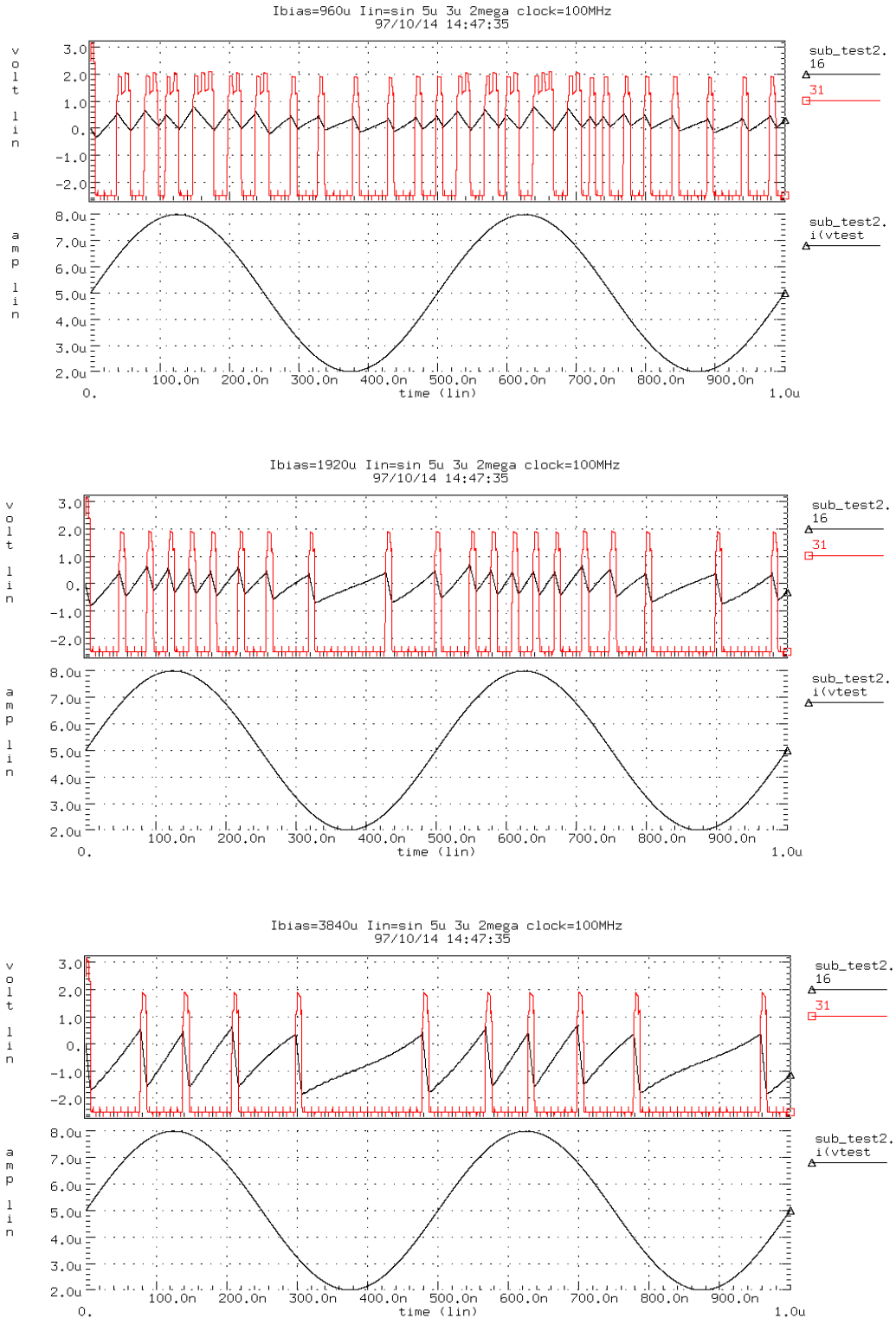


Figure F.5 Simulation results with different bias currents

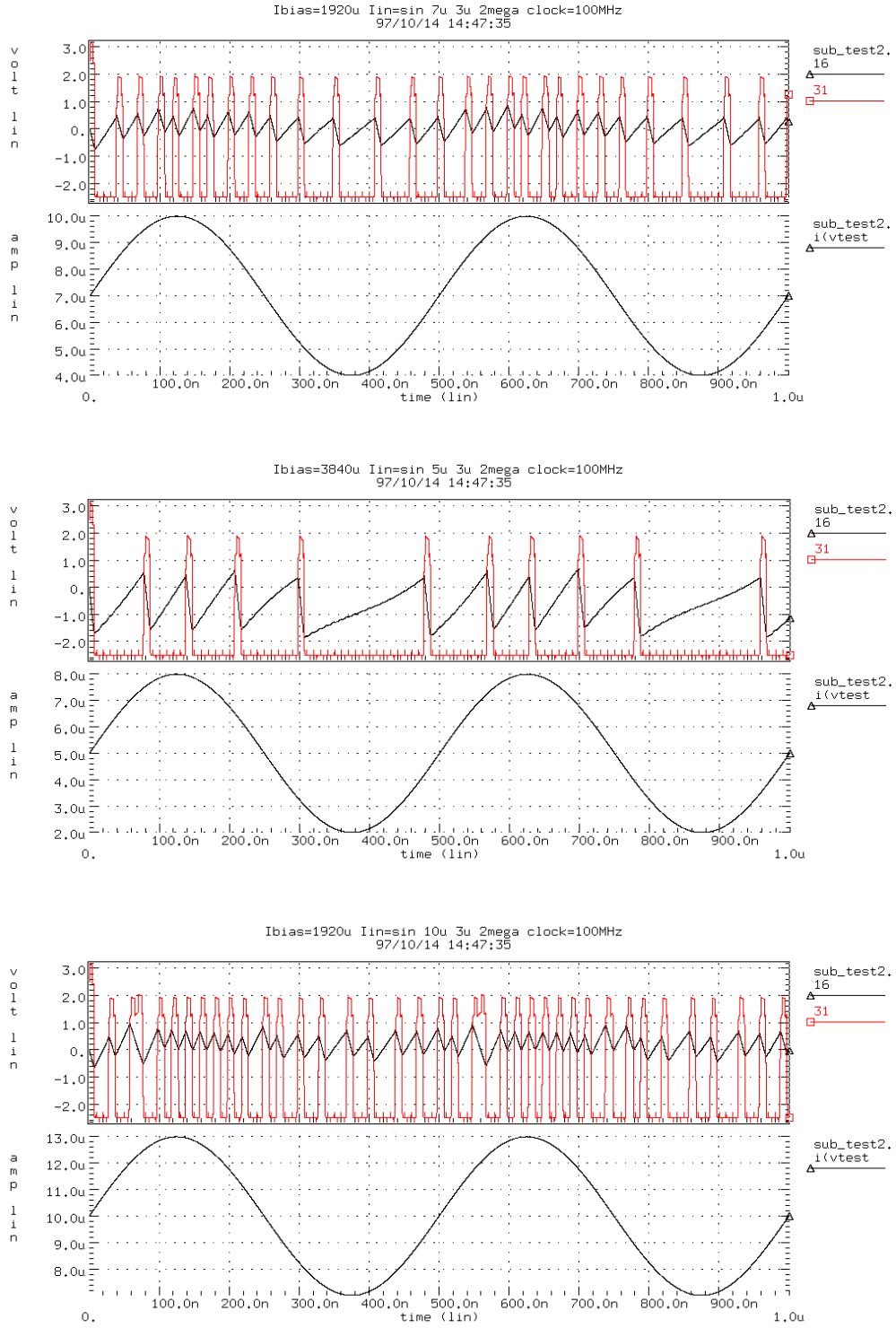


Figure F.6 Simulation results with different input currents

REFERENCES

[1] A. R. Greig, D. Salters, P.A Williams, “Development and application of a twin CCD camera system for in-cylinder fuel spray analysis,” *IEE Conf Publ* vol. 443, pp. 151-155, 1997.

[2] T. McIntyre, P. Rabbath, A. Houwing,” Imaging of combustion processes in a supersonic combustion ramjet,” *Proceedings of the 12th International Symposium on Air Breathing Engines*, pp.1163, 1995.

[3] D. Lytle, B.W. Webb, “Air jet impingement heat transfer at low nozzle-plate spacings,” *Int J Heat Mass Transfer*, vol. 37, no. 12, pp. 1687-1697, 1994.

[4] Paul F. Goldsmith, “Focal-plane-arrays for millimeter-wavelength astronomy,” *IEEE MTT S Int Microwave Symp Dig*, vol. 3, pp. 1255-1258, 1992.

[5] Bojan T. Turko, Georgia J. Yates, Nicholas S. King, “ Processing of multiport CCD video signals at very high frame rates,” Proc. SPIE Int. Soc. Opt Eng., vol. 2549, pp. 11-15, 1995.

[6] Zhimin Zhou, Bedabrata Pain, Roger Panicacci, Barmak Mansoorian, Junichi Nakamura, Eric R. Fossum, “On-Focal-Plane ADC: Recent Progress at JPL,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp. 111-122, 1996.

[7] Bedabrata Pain, Eric R. Fossum, “Approaches and Analysis for On-Focal-Plane Analog-to-Digital Conversion,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2226, pp. 208-218, 1994.

[8] Ulf Ringh, Christer Jansson, Kevin Liddiard, “ Readout Concept Employing a Novel on Chip 16bit ADC for Smart IR Focal-plane-arrays,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp.99-110, 1996.

[9] Tierry M. Bernard, “Introduction,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp. vii, 1996.

[10] “Special Issue on Solid State Image Sensors,” *IEEE Trans. on Electron Devices*, vol. 38, 1991.

[11] C. Camperi-Ginestet, Y.W. Kim, N.M. Jokerst, M.G. Allen, M.A. Brooke, “Vertical Electrical Interconnection of Compound Semiconductor Thin-Film Devices to Underlying Silicon Circuitry,” *IEEE Photonics Technology Letters*, vol. 4, no. 9, pp. 1003-1006, September 1992.

[12] Olivier Vendier, N. Marie Jokerst, R. P. Leavitt, “Thin Film Inverted MSM Photodetectors,” *IEEE Photonics Technology Letters*, vol. 8, no. 2, pp. 266-268, February 1996.

[13] David X.D. Yang, Hao Min, Boyd Fowler, Abbas El Gamal, Mark Beiley, Kit Cham, “ Test Structure for Characterization and Comparative Analysis of CMOS Image Sensors,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2950, pp. 8-17, 1996.

[14] J. Solhusvik, C. Cavadore, F. X. Audoux, N. Verdier and J. Farre, “ Recent experimental results form a COMS Active Pixel Image Sensor with Photodiode and Photogate Pixels,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2950, pp. 18-24, 1996.

[15] Sunetra K. Mendis, Sabrina E. Kemeny, Russell C. Gee, Bedabrata Pain, Craig O. Staller, Quiesup Kim, Eric R. Fossum, “ COMS Active Pixel Image Sensors for Highly Integrated Imaging Systems,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 2, pp.187-196, 1997.

[16] Uif Ringh, Christer Jansson, and Kevin Liddiar, “ Readout Concept Employing a Novel on Chip 16 bit ADC for Smart IR Focal-plane-arrays,” *SPIE*, vol.2745, pp.99-110, 1996.

[17] J. Mandl William, Carl Rutschow, "All-Digital Monolithic Scanning Readout Based on Sigma-Delta Analog-to-Digital Conversion," *Proceedings of SPIE - The International Society for Optical Engineering*, v 1684, pp. 233-246, 1992.

[18] Phillip E. Allen, Dounglas R. Holberg, *CMOS analog circuit design, Saunders College Publishing, 1987.*

[19] A. David Martin, Hae-Seung Lee, Ichiro Hasaki, " A Mixed-Signal Array Processor with Early Vision Applications," *IEEE Custom Integrated Circuits Conference*, pp.275-278, 1997.

[20] J. C. Candy, G. C. Temes, "A Tutorial Discussion of the Oversampling Method for A/D and D/A Conversion," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol 2. pp. 910-913, 1990.

[21] J. C. Candy, G. C. Temes, "Oversampling Methods for A/D and D/A Conversion," *Oversampling Delta-Sigma Data Converters*, IEEE Press, New York, 1992.

[22] J. C. Candy, G. C. Temes, "Oversampling Methods for Data Conversion," *Proc 91 IEEE Pacific Rim Conf Commun Comput Signal Process*, pp. 498-502, 1991.

[23] C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, Massachusetts, 1989.

[24] M. W. Haney, M. P. Christensen, "Smart Pixel Based Viterbi Decoder," *Optical Computing, 1995 Technical Digest Series*, vol. 10, pp. 99-101, 1995.

[25] C. B. Kuznia, A. A. Sawchuk, and L. Cheng, "FET-SEED Smart Pixels for Free-Space Digital Optics Systems," *Optical Computing, 1995 Technical Digest Series*, vol. 10, pp. 108-110, 1995.

[26] CPU Info Center, http://infopad.eecs.berkely.edu/CIC/summary/index.html.

[27] T. Blank, “MasPar MP-1 Architecture,” Proceedings of COMPCON Spring *’90 - The Thirty-Fifth IEEE Computer Society International Conference*, San Francisco, CA, pp. 20-24, 1990.

[28] W-C Fang, T. Shaw, J. Yu, “VLSI Focal-Plane Array Processor for Morphological Image Processing,” *Proceedings of Fifth Annual IEEE International ASIC Conference and Exhibit, Rochester*, NY, pp. 423-426, 1992.

[29] Huy H. Cat, Antonio Gentile, John C. Eble, Myunghee Lee, Olivier Vendier, Youngjoong Joo, Scott D. Wills, Martin Brooke, Nan Marie Jokerst, and April S. Brown, "SIMPil: An OE Integrated SIMD Architecture for Focal Plane Processing Applications," *MPPOI*, pp. 44-52, 1996.

[30] Bedabrata Pain, Eric R. Fossum, “Approaches and Analysis for On-Focal-Plane Analog-to-Digital Conversion,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2226, pp. 208-218, 1994.

[31] Ulf Ringh, Christer Jansson, Kevin Liddiard, “ Readout Concept Employing a Novel on Chip 16bit ADC for Smart IR Focal-plane-arrays,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp.99-110, 1996.

[32] Zhimin Zhou, B. Panicacci Pain, B. N. Roger Mansoorian, E. R. Fossum, “On-Focal-Plane ADC: Recent Progress at JPL,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp. 111-122, 1996.

[33] Youngjoong Joo, S. Fike, Kee Shik Chung, Martin A.Brooke, Nan M. Jokerst, Scott Wills, “ *Application of Massively Parallel Processors to Real Time Processing of High Speed Images,*” *MPPOI*, pp.96-100, 1997.

[34] H. H. Cat., J. C. Eble, D. S. Wills, V. K. De, M. Brooke, N. M. Jokerst, “Low Power Opportunities for a SIMD VLSI Architecture Incorporating Integrated Optoelectronic Devices,” *GOMAC’96 Digest of Papers*, pp. 59-62, March 1996.

[35] Huy H. Cat, D. Scott Wills, Nan Marie Jokerst, Martin A. Brooke, April S. Brown, "Three-dimensional, massively parallel, optically interconnected silicon computational hardware and architectures for high-speed IR scene generation," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2469, pp. 141-145, 1995.

[36] John P. Uyemura, *Circuit design for CMOS VLSI*, Kluwer Academic Publishers, 1998.

[37] J. C. Candy, G. C. Temes, "A tutorial discussion of the oversampling method for A/D and D/A conversion," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 910-913, 1990.

[38] Christer Jansson, Ulf Ringh, Kevin Kiddiard, "ON-chip analog-to-digital conversion suitable for uncooled focal plane detector arrays employed in smart IR sensors," Proceedings of SPIE - *The International Society for Optical Engineering*, vol. 2474, pp. 72-87, 1995.

[39] Uif Ringh, Christer Jansson, Kevin Kiddiard, "Readout conceop employing a novel on chip 16 bit ADC for smart IR focal-plane-arrays," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2745, pp. 99-110, 1996.

[40] W. Marshall Leach Jr., "Fundamentals of low-noise analog circuit design," *Procedings of the IEEE*, vol. 82, no. 10, pp. 1515-1538, 1994.

[41] Steven R. Norsworthy, Irving G. Post, H. Scott Frtterman, "A 14-bit 80-kHz sigma-delta A/D converter: modeling, design, and performance evaluation," *IEEE J. Solid-State Circuits*, vol. SC-24, pp. 256-266, April 1989.

[42] J. C. Candy, G. C. Temes, "Oversampling Methods for Data Conversion," *Proc 91 IEEE Pacific Rim Conf Commun Comput Signal Process*, pp. 498-502, 1991.

[43] J. C. Candy, G. C. Temes, "A Tutorial Discussion of the Oversampling Method for A/D and D/A Conversion," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol 2. pp. 910-913, 1990.

[44] Kenneth R. Laker, Willy M.C. Sansen, *Design of analog integrated circuits and systems*, McGRAW Hall, 1994.

[45] Toumazou, C. Hughes, J. B. Pattullo, "*Regulated cascode switched-current memory cell*," Electron Lett., v26, n5, p303-305, 1990.

[46] John B. Hughes, Kenneth W. Moulding, "Switched-current signal processing for video frequencies and beyond," *IEEE Journal of Solid-State Circuits*, v28, n3, p214-322, 1993.

[47] G. M. Yin, F. Op't Eynde, W. Sansen, "Ahigh-speed CMOS comparator with 8-b resolution," *IEEE J. of Solid-State Circuits*, vol. 27, no. 2, pp.208-211, February 1992.

[48] William J. Mandl, " Focal plane analog-to-digital conversion development," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2474, pp. 63-71, 1995.

[49] Fike, S.M., Buchanan, B., Jokerst, N.M., Brooke, M.A., Morris, T.G., DeWeerth, S.P., "8x8 Array of Thin-Film Photodetectors Vertically Electrically Interconnected to Silicon Circuitry," *IEEE Photonics Technology Letters*, vol. 7, no. 10, pp 1168-1170, October 1995.

[50] Antoine Dupret, Eric Belhaire, "A high current large bandwidth photosensor on standard CMOS processes," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2950, pp. 36-44, 1996.

[51] Pallab, Bhattacharya, *Semiconductor optoelectronic devices*, Prentice Hall, 1994.

[52] Olivier Vendier, N. M. Jokerst, R. P. Leavitt, "Thin Film Inverted MSM Photodetectors," *IEEE Photonics Technology Letters*, vol. 8, no. 2, pp. 266-268, February 1996.

[53] Chow, K., Rode, J.P., Seib, D.H., and Blackwell, J.D., " Hybrid Infrared Focal-plane-arrays," *IEEE Trans. Electron Devices*. vol. 29, no.1, 1982.

[54] S.M. Fike, B. Buchanan, N.M. Jokerst, M.A. Brooke, T.G. Morris, S.P. DeWeerth, "8x8 Array of Thin-Film Photodetectors Vertically Electrically Interconnected to Silicon Circuitry," *IEEE Photonics Technology Letters*, vol. 7, no. 10, pp 1168-1170, October 1995.

[55] T.Nakamura, K.Matsumoto, F.Hyuga,and A. Yusa, "A new MOS image sensor operating in a nonstructive readout mode," *Int. Electron Device Meet.*, pp.353, 1986.

[56] Antonine Dupret, Eric Belhaire, " A High Current Large bandwidth photocwnsor on standard CMOS process," ," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2950, pp. 36-44, 1996.

[57] Robert W. Sandage, J. Alvin Connelly, " Producing Phototransistors in a Standard Digital CMOS Technology," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol 2. pp. 369-372, 1996.

[58] S. P. DeWeerth and C. A. Mead, "A two-dimentional visual tracking array," *Proc. MIT Conf. VLSI*. Pp.259-275, 1988.

[59] A. Gruss, L. R. Carley, and T. Kanade, "Integrated sensor and range finding analog signal processor," *IEEE J. Solid-State Circuits*, vol. 26, pp. 184-191, 1991.

[60] O. Yadid-Pecht, R. Ginosar, and Y. S. Diamand, " A random access photodiode array for intelligent image capture," *IEEE Trans. Electron Devices*, vol. 38, pp. 1772-1780, 1991.

[61] Robert W. Sandage, J. Alvin Connelly, ” Producing Phototransistors in a Standard Digital CMOS Technology,” *Proceedings - IEEE International Symposium on Circuits and Systems*, vol 2. pp. 369-372, 1996.

[62] M. P. Vidal, M. Bafleur, J. Buxo, and G. Sarrabayrouse, “A bipolar photodetector compatible with standard COMS technology,” *Solid-State Electron.*, vol. 34, pp.809-814, 1991.

[63] K. R. Laker, W. M. Sansen, “ Design of analog integrated circuits and systems,” *McGRAW-Hill, Inc*. 1994.

[64] D. Anastassiou, “ Error diffusion coding for A/D conversion,” *IEEE Trans. Circuits Sys*., vol. CAS-36, pp., 1175-1186, 1989.

[65] T. Ritoniemi, T. Karema, and H. Tenhunen, “ The design of stable high order 1-bit sigma-delta modulators,” *IEEE Proc. ISCAS’90*, pp. 3267-3270, 1990.

[66] S. K. Tewksbury and R. W. Hallock, “ Oversampled, linear predictive and noise-shaping coders of order N>1,” *IEEE Trans. Circuits Sys*., vol. CAS-25, pp. 436-447, 1978.

[67] B. P. Brandt, D. E. Wingard, and B A. Wooley, “ Second-order sigma-delta signal acquisition,” *IEEE J. Solid-State Circuits*, vol. SC-26, pp. 618-627, 1991.

[68] J. C. Candy, “ A use of double integration in sigma-delta modulation,” *IEEE Trans. Commun*., vol. COM-33, pp. 249-258, 1985.

[69] R. E. Crochiere and L. R. Raginer, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[70] S. R. Norsworthy, R. Schreier, G. C. Temes, *Delta-Sigma Data Converter*, IEEE Press, Piscataway, NJ, 1998.