SELF-CALIBRATION APPROACH FOR MIXED SIGNAL CIRCUITS IN SYSTEMS-ON-CHIP

A Dissertation Presented
by
In-Seok Jung
to
The Department of
Electrical and Computer Engineering
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in the field of
Electrical Engineering
Northeastern University
Boston, Massachusetts
AUGUST 2014
Abstract

MOSFET scaling has served the semiconductor industry very well for a few decades through improvements in transistor performance, power, and cost. However, modern integrated systems require high test complexity and cost due to several issues such as limited pin count and integration of analog and digital mixed circuits.

Self-calibration is an excellent and promising method to improve yield and to reduce manufacturing cost by simplifying the test complexity because it is possible to address the process variation effects by means of self-calibration techniques. Since many prior published calibration techniques were developed for a specific targeted application, it is not easy to utilize them for other applications.

In order to solve the aforementioned issues, several novel self-calibration design techniques in mixed-signal mode circuits are proposed in this dissertation for an analog to digital converter (ADC) to reduce mismatch error and improve performance. ADCs are essential components in systems-on-chips (SOCs) and the proposed self-calibration approach compensates for process variations.

The proposed novel self-calibration approach targets the successive approximation (SA) ADC. First of all, the offset error of the comparator in the SA-ADC is reduced using the proposed approach by tuning the capacitor array in the input nodes for better matching. In addition, the auxiliary capacitors for each capacitor in the DAC of the SA-ADC are controlled by a synthesized digital controller to minimize the mismatch error of the DAC. Since the proposed technique is applied during foreground operation, the power overhead in SA-ADC case is minimal because the calibration circuit is deactivated during normal operation time.

Another benefit of the proposed technique is that the offset voltage of the comparator is continuously adjusted for the decision of each bit to not only the reduce offset voltage of the comparator but also to compensate for the mismatch of DAC.
The proposed digital calibration control circuit operates in the foreground during calibration mode, and the controller has been optimized for low power consumption and high performance.

In order to increase the sampling clock frequency of proposed self-calibration approach, a novel variable clock period method is proposed. To achieve high-speed SAR operation, a variable clock time technique is used to reduce not only peak current but also die area. The technique avoids conversion time loss and extends the SAR ADC’s speed of operation.

To verify and demonstrate the proposed techniques, a prototype charge-redistribution SA-ADCs with the proposed self-calibration was implemented in a 130nm standard CMOS process. The prototype circuit’s silicon area is 0.0715 \text{mm}^2 and consumes 4.62mW with 1.2V power supply.
Acknowledgements

This dissertation could not have been written without Dr. Kim who not only served as my supervisor but also encouraged me throughout my academic program. He and the other faculty members, Professor Marvin Onabajo and Professor Fabrizio Lombardi, guided me through the dissertation process, never accepting less than my best efforts. I really appreciate them all.

My thanks also go to my dear friends, colleagues and my laboratory co-workers, who are Hojoon Lee, Yongsuk Choi, and Gyunam Jeon. During Ph.D courses, they have encouraged and helped me.

I would finally like to express my greatest gratitude toward my family; my parents and my sister for the unlimited love and support they provided throughout my entire life without which I would have never been even close to where I am today.
Contents

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv
Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vi
List of Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
List of Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xi

1 Introduction 1
1.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.1.1 Systems-on-chips (SOCs) for mixed signal circuits . . . . . . 2
1.1.2 Built-in-self-test (BIST) . . . . . . . . . . . . . . . . . . . . . . . . 3
1.1.3 Analog to digital converters (ADCs) . . . . . . . . . . . . . . . . 4
1.2 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
1.3 Objectives and contributions . . . . . . . . . . . . . . . . . . . . . . . 6
1.4 Organization of this dissertation . . . . . . . . . . . . . . . . . . . . . 6

2 Prior self-calibration techniques and analysis 8
2.1 Analysis of SAR ADC 8
2.1.1 SAR ADC structure . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.1.2 Performance metrics . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.1.2.1 Static specifications . . . . . . . . . . . . . . . . . . . . . . . . 12
2.1.3 DAC mismatch . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.1.4 Offset mismatch . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.5</td>
<td>Gain mismatch</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Current self-calibration techniques for SAR ADC</td>
<td>17</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Analog calibration</td>
<td>18</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Digital calibration</td>
<td>18</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Foreground &amp; background calibration</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Proposed self-calibration research approach</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Proposed calibration approach for SAR ADC</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>Proposed offset calibration for comparator</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>Proposed DAC mismatch calibration in SAR ADC</td>
<td>24</td>
</tr>
<tr>
<td>3.3.1</td>
<td>An inverter based comparator to consider offset error</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Simulation Result</td>
<td>35</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Offset voltage calibration</td>
<td>35</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Capacitor mismatch calibration</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>One-step self-calibration approach</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Proposed self-calibration methodology</td>
<td>44</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Digital self-calibration scheme</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>Simulation result</td>
<td>47</td>
</tr>
<tr>
<td>5</td>
<td>High sampling speed approach using a variable clock period method</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Conventional SAR ADC</td>
<td>51</td>
</tr>
<tr>
<td>5.2</td>
<td>Proposed SAR ADC</td>
<td>54</td>
</tr>
<tr>
<td>5.2.1</td>
<td>SAR Control Logic Implementation</td>
<td>55</td>
</tr>
<tr>
<td>5.3</td>
<td>Simulation result</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>6.2</td>
<td>Future work</td>
<td>63</td>
</tr>
</tbody>
</table>
6.2.1 Calibration application for the mismatch of other SAR ADC architectures

A Verilog-HDL source code for the digital controller for one-step calibration

B SPICE netlist for 10-bit SAR ADC using variable clock period method

Bibliography
List of Tables

3.1 Performance summary for the proposed SAR ADC  . . . . . . . . . . 39

4.1 Performance summary for the proposed SAR ADC  . . . . . . . . . . 48

5.1 Performance summary for the proposed SAR ADC  . . . . . . . . . . 58
List of Figures

1.1 An example of a mixed-signal SOC ........................................ 3

2.1 ADC architectures for different application .......................... 9

2.2 A conventional charge-redistribution based SAR ADC block diagram .................................................. 10

2.3 Waveform example of SAR ADC ........................................ 11

2.4 The example of transfer curves of a 12-bit SA-ADC ................ 13

2.5 Maximum possible resolution of a charge redistribution SAR ADC with a fully binary-weighted capacitor array depending on the matching of an unity capacitor .................................................. 14

2.6 $\sigma(\Delta V_T)$ (blacksquare) and $\sigma(\Delta \beta/\beta)$ (♦) for a minimal nMOS device in different technology nodes .................................................. 16

2.7 The self-testing and calibration architecture in [1] .................. 20

3.1 Schematic of the comparator with pre-amplifier for the SAR ADC .................................................. 22

3.2 Proposed self-calibration approaches (a) Schematic of the comparator with capacitor array to reduce the offset voltage and (b) block diagram for the self-calibration approach .................................................. 23

3.3 Proposed self-calibration strategy for the DAC in SAR ADC .......................... 25

3.4 (a) Calibration algorithm and (b) diagram of the calibration process .................................................. 26

3.5 Flow chart of proposed algorithm for N-Bit DAC for self-calibration .................................................. 29

3.6 Conventional comparison concept of $C_1$ and $C_0$ .................. 30
3.7 The schematic of proposed $\Delta V$ sensing circuit ........................................... 32

3.8 The DAC with the integrated $\Delta V$ sensing circuit .......................................... 34

3.9 Monte Carlo histogram of the offset voltage of the pre-amplifier and comparator ................................................................. 36

3.10 The compensation range of the digitally controlled capacitive calibration ................................................................. 37

3.11 Monte Carlo histogram of the offset voltage of the pre-amplifier and comparator using the proposed calibration approach ................................................................. 38

3.12 Transfer curve of a 12-bit SAR ADC with 5% mismatch error of capacitance (a) without proposed calibration technique (b) with proposed calibration technique ........................................................................................................................................... 40

3.13 Simulated INL and DNL at 32MS/s ...................................................................... 41

3.14 Simulated SNDR and SFDR performance versus input frequency at 32MS/s ................................................................................................................................. 42

4.1 An example of 4-bit SAR ADC operation (a) and SAR ADC architecture (b) ................................................................................................. 45

4.2 Proposed SAR ADC structure ................................................................................. 46

4.3 The layout of the proposed ADC .............................................................................. 47

4.4 Simulated INL and DNL at 64MS/s ...................................................................... 49

4.5 Simulated SNDR and SFDR performance versus input frequency at 64MS/s ................................................................................................................................. 50

5.1 A conventional SAR ADC architecture .................................................................. 51

5.2 Example of 4-bit SAR ADC operation .................................................................. 53

5.3 Settling time with a large capacitor (a) and a small capacitor (b) .................. 53

5.4 A comparison of a (a) conventional clock timing for SAR ADC, (b) ideal clock timing and (c) proposed clock timing for SAR ADC ........................................................................................................................................... 54

5.5 The proposed clock generator ................................................................................. 56

xii
5.6 Layout of the proposed ADC ........................................... 57
5.7 Simulated INL and DNL at 64MS/s ............................... 59
5.8 Simulated SNDR and SFDR performance versus input frequency at 64MS/s .......................................................... 60
6.1 Example of a split DAC for 6-bit SAR ADC ....................... 64
6.2 Block diagram of time interleaved ADC ............................ 64
Both digital and analog circuits test technology have been developing for nearly 40 years and have evolved into hardware and software based testing techniques. In early years, a test bench had to be designed and constructed for each circuit. Later, automatic test equipment (ATE) has been used for a general test solution for most devices. During the use of the ATE, the complexity and density of the circuits increased dramatically while better quality and reliability were required by consumers and the market at the same time. While the cost per chip has been decreased with advances in fabrication, the cost for test has not been decreased more than expected in the market. Especially, as analog circuits need a variety of target specification, built-in-self-test (BIST) methodology has been developed to reduce the total cost. Although the chip size is increased, the total cost can be reduced by using BIST as the cost for test can be reduced. Both external ATE machines used in the IC production stage and embedded test solutions such as BIST required for chip diagnosis and test are necessary in the design of modern electronic systems. The need to adopt or establish automated testing standards has been recognized as essential for higher yield and lower cost by most manufacturing companies. However, no such automatic process exists for mixed-signal circuits where the interface between digital and analog
components is impossible to be directly accessed by the test circuit and equipment. Moreover, additional pin count is most likely required. This dissertation investigate a novel self-calibration approach for BIST in mixed-mode integrated circuits in SOCs.

1.1 Overview

1.1.1 Systems-on-chips (SOCs) for mixed signal circuits

The advancement in fabrication and design technology enables the integration of various digital and analog modules, and it has helped to bring single systems-on-chips (SOCs) [2, 3]. Traditional circuit components developed as reusable cores are now integrated into one complex SOC. Most SOCs include a microprocessor, DSP processor, memory, RF components, analog block, and application specific CMOS logic units. In addition to digital units, analog and mixed-signal circuits are becoming essential parts of SOCs and wireless modems including transceiver also have been integrated into SOCs. Analog and mixed-signal circuits have been widely used for multimedia, wireless communication, networking and control systems [4]. For example, the emerging applications such as micro-electro-mechanical systems (MEMS) and application processor for mobile devices are integrated with electronic circuits in a single substrate and analog circuitry is essential to enable interaction between digital logic blocks and other blocks. It is expected that most future ICs will be mixed-signal circuits rather than pure digital logic or analog logic.

The Figure 1.1 shows an example of a mixed-signal SOC. The most common analog circuits are analog to digital converters (ADCs) and digital to analog converters (DACs) to interface digital processors with the real world. It is also common that complex mixed-signal SOCs include more than one ADC and DAC. These data converters require anti-aliasing filters and reconstruction filters to remove aliasing noise [5]. The programmable-gain amplifier (PGA) is another common analog block
input of an ADC or output of an DAC to control the signal power. These analog circuits also require a band-gap reference (BGR) circuit, which will generate reference voltage and currents for other analog circuits, and LDO, which supply low voltage to analog circuits to reduce power consumption. The phase-locked loop (PLL) is another common block in SOCs that is usually used to generate a high frequency modulation signal for communication or to synthesize a digital clock. Normally, more than two PLLs are used in many SOCs.

1.1.2 Built-in-self-test (BIST)

Digitally assisted analog design \cite{6,7} and integrated transceiver calibration \cite{8,9} approaches are gaining popularity in ensuring efficient and reliable mixed-signal systems in nano-scale CMOS technologies. One design aspect is to equip analog blocks with performance-tuning features that allow the recovery from process variations and faults. Examples of such tuning mechanisms include input impedance matching, gain
and center frequency tuning for low-noise amplifiers [9, 11], second-order nonlinearity and mismatch correction for mixer [12, 14] as well as linearity enhancements for baseband filters [15]. The other aspect related to digitally assisted design is the extraction of performance metrics on the chip to enable one-time or periodic calibrations. Many performance characteristics can be observed based on the output spectrum of a circuit under test (CUT) or a chain of analog blocks, which has led to on-chip spectrum analyzers that emulate conventional off-chip instrumentation [16, 17].

Research articles have been published to propose calibration methods that incorporate existing or dedicated analog-to-digital converter (ADC) and digital signal processing resources to directly quantize the output signals of analog circuits for the computation of the Fast Fourier Transform (FFT) and automatic tuning with digital-to-analog converters (DACs) [18, 19]. In the articles, the frequencies for two-tone tests can be selected by the designer of the built-in test (BIT) or built-in calibration (BIC) scheme to circumvent inaccuracies due to spectral leakage while using a small number of FFT points. This capability to accurately measure the power of the tones as well as their distortion and intermodulation products with an efficient on-chip FFT can also find application in loop-back testing techniques with spectral estimation such as in [20, 21].

1.1.3 Analog to digital converters (ADCs)

Communication systems such as cellular radios, fiber optic links, and cable modems employ analog-to-digital converters (ADCs) in their receivers to convert analog signals to digital signals. There is a growing trend in building all-digital receivers by moving the ADC, traditionally placed at the end of the receiver chain, closer to the front-end [22]. By doing so, many of the analog circuits such as mixers and filters, can be implemented digitally to benefit directly from technology scaling.
On the other hand, with the scaling down of CMOS technology, successive approximation register (SAR) ADCs are widely used because of their high power efficiency and small area. In a 10-bit resolution, the operating frequency of the conventional SAR ADCs is increased up to 100 MHz by asynchronous internal clocks, error compensation, capacitor switching techniques, and etc [23]. Nevertheless, conventional SAR ADCs still have speed limitations due to their serial signal processing. Most high-speed SAR ADCs have a short sampling and conversion period causing an accuracy issue by not enough conversion time.

1.2 Motivation

Although integrated SOCs have a lot of advantages of low cost and low power, the test complexity and cost are increased to improve yield. Many techniques have been published to address those issues. However, the issues have not been resolved because process, voltage, and temperature (PVT) variation have been increased and design consideration have been more complexed with nano scaling. Self-calibration technique is one of the excellent methods to improve yield and reduce manufacturing cost, because it is possible to address the PVT variation effects by means of calibration technique. Although the presented self-calibration techniques help solve the issues, the techniques are designed for only target applications so that it could not be utilized for other general applications. In this dissertation, a novel calibration based mixed-signal mode circuit design technique is proposed for SAR ADCs, and the technique uses an enhanced self-calibration approach to compensate the process variations. The proposed self-calibration approach can be utilized for other ADC or the other general applications using SAR logic algorithm.
1.3 Objectives and contributions

The objective of this dissertation is to describe and demonstrate a novel self-calibration techniques to reduce mismatches and errors due to process variations of chips in portable applications such as mobile phone, smart-phones, and tablets. The main contributions of this research are:

1. A novel reconfigurable self-calibration architecture for ADC in SOCs
2. Extension of the self-calibration technique to ADC in SOCs
3. A BISC ADC for BIST using the proposed technique in SOC for communication applications

The proposed technique reduces both mismatch error of DAC in ADC and offset error of comparator. The error can be measured by using additional circuitry and minimized during non-normal operation time.

1.4 Organization of this dissertation

The main chapter of the proposed begins from Chapter 2, briefly discusses the causes of error occurrences and the variations such as mismatch error and offset error. This chapter also presents prior self-calibration techniques for ADCs to prevent the explained errors for SOCs.

Chapter 3 introduces and describes the proposed new self-calibration techniques for ADC. It chapter illustrates how the proposed techniques reduce the DAC mismatch and offset error of comparator. The simulation results shows how much the performance is improved by using the proposed approach.

Chapter 4 shows a one-step self-calibration techniques that is similar to the one in Chapter 3. Compared to the techniques in the previous chapter, the offset voltage
of the comparator and capacitor mismatch of the DAC are minimized by adjusting the auxiliary capacitors in the comparator.

Chapter 5 demonstrates that proposed control method to increase the sampling frequency with small chip die area to use the proposed self-calibration techniques with experimental result.

Lastly, the conclusion and future work are presented with the summary of this dissertation in Chapter 6.
Chapter 2

Prior self-calibration techniques and analysis

2.1 Analysis of SAR ADC

There are several major types of ADC architectures. Each type entails different trade-offs among required sampling speed, resolution, power, and area. Different ADC types and their working ranges are depicted in Figure 2.1.

Sigma-Delta (Σ-Δ) ADCs are suitable for applications with low sampling frequency and high resolution. While, pipeline and flash ADCs have been used for medium resolution high-speed data conversion [24–31]. Flash ADC dissipates a significant power and pipeline ADC requires linear amplifiers that are difficult to realize at low supply voltages. With the continued advancement in modern process technologies, SAR ADCs are becoming increasingly attractive for low-power, high-speed, and medium resolution data conversion [32–41].

Very high-speed ADCs are also required in modern telecommunication and system on chip (SOC) applications. Very high-speed ADC can be realized using time-interleaved architecture that can be combined with practically any ADC technology.
Very high-speed SAR ADCs using time-interleaved technology have been reported in literature [22, 42–47]. Time-interleaved SAR ADCs suffer from channel mismatches. To achieve the desired ADC performance, either the mismatch should be kept small by drawing accurate layouts or the ADC should be calibrated using extensive digital circuitry.

Different techniques have been proposed to increase the speed of SAR ADC in [48, 49]. The speed of an N-bit SAR ADC is limited primarily by the delay of the N bit cycles. The SAR ADC bit cycle period is defined by DAC settling time, comparator decision time, and the SAR logic speed. To reduce the DAC settling time, binary DAC with redundancy have been employed to correct both comparator decision and DAC settling errors. These methods use extra hardware and may require additional bit cycles and more complex SAR logic.

![Figure 2.1: ADC architectures for different application](image-url)
2.1.1 SAR ADC structure

A block diagram of a charge-redistribution based SAR ADC is shown in Figure 2.2. The ADC consists of a sampler, digital to analog converter (DAC), comparator, and the SAR logic. The input is sampled at the beginning of the conversion. The quantization steps are generated using the DAC. The SAR logic uses the comparator output to generate the DAC control codes. A successive approximation ADC generally uses a binary search algorithm to quantize the ADC input voltage. The binary search algorithm uses \( N \) calculation cycles to find the quantization step that is the closest to the input signal. The plot of a SAR ADC waveform example is shown in Figure 2.3. The signal is first compared with the middle step \((V_{\text{ref}}/2)\). If the signal is higher than \( V_{\text{ref}}/2 \), in the next cycle, it will be compared with \( 3V_{\text{ref}}/4 \), otherwise it will be compared with \( V_{\text{ref}}/4 \). At the third cycle \( V_{\text{ref}}/8 \) will be added or subtracted from the comparison step. This also can be done similarly by adding or subtracting the fractions of \( V_{\text{ref}} \) from the input.

![Figure 2.2: A conventional charge-redistribution based SAR ADC block diagram](image)

As can be seen in Figure 2.2, total capacitance of capacitor array of DAC is \( 2^N \) times of unit capacitor \( C \) for LSB, where \( N \) is the resolution of SAR ADC. 

For example, 10-Bit ADC needs $1024C$ of total capacitance for DAC. Each size of capacitor of the binary weighted capacitance array should have high accuracy because capacitance mismatch error causes the non-linearities.

One of the method to reduce the size of total capacitance is to use a split DAC [51]. A split-DAC uses reduced total capacitance in the capacitor array of the DAC. However, the ADC using split-DAC has a high sensitivity capacitor that is used for the splitter so that the capacitor mismatch as parasitic capacitance is more important for split-DAC based ADC.

In order to reduce the capacitance mismatch error, optimized place and routing techniques for layout is published in [52]. However, high complexity for metal line routing in layout causes other issues such as improvement of parasitic capacitance for routing.

### 2.1.2 Performance metrics

The performance of the data converter in the presence of circuit non-idealities also should be considered. Due to different practical non-idealities, performance metrics
need to be defined and used for the evaluation of the ADCs. Noise and mismatch are two main sources of non-linearities/error sources. These errors reduce the dynamic range of the ADC and reduce the ADC’s effective number of bits (ENOB).

2.1.2.1 Static specifications

The transfer function or transfer characteristic of an ADC is a function that assigns a digital code to the analog value of the input signal. Ideally, in the case of a perfect binary SAR ADC, the input signal range is divided into $2^N$ equal segments and each segment is assigned a unique digital code from 0 to $2^N - 1$ in ascending order, so that lower digital codes correspond to the smaller input analog voltage. In practical implementations, a perfect radix two can never be achieved due to capacitor mismatches. Sufficiently small mismatches can be obtained if large capacitors are used, but that it comes with power and area penalty. It is often beneficial to use minimum-sized capacitors dictated by the thermal noise requirements, and to allow bigger mismatches if they can be somehow corrected. These mismatches will create deviations from the ideal transfer function.

2.1.3 DAC mismatch

The conversion linearity of the SAR ADC is subject to circuit component non-idealities. In case of charge redistribution SAR ADC, when matching is accurate, the SAR ADC performs an ideal binary search to convert the sampled analog input into an N-bit binary code. The resulting ADC transfer curve is shown as the dotted line in Figure 2.4 for a 12-bit example. In this case, the conversion is free of any differential or integral nonlinearity.

The mismatch of capacitor is composed of global and local effects. The edge and the oxide of the capacitor are other two variables in different point of view. The relationship is given by the equation (2.1).
where the $K_{le}$ is the local edge effect factor, $K_{ge}$ is the global edge effect factor, $K_{lo}$ is the local oxide effect factor, and $K_{go}$ is the global oxide effect \[54\].

When capacitor mismatch is present, the ADC transfer curve is highly distorted, especially when small capacitors are used for fast settling and to reduce power consumption. As a result, the decision levels may no longer be uniformly distributed over the full input range, e.g., the one indicated by the solid line in Figure 2.4. The vertical and horizontal misalignments are known as missing codes and missing decision levels, respectively.

In \[55\], the relationship between capacitor mismatch error $\delta$ and the resolution of N-bit charge redistribution SAR ADC is given by equations (2.2) and (2.3).
\[ N_{\text{max}} = \log_2 \left( \frac{1 + \delta + \sqrt{1 + 2\delta - 3\delta^2}}{2\delta} \right) \]  \hspace{1cm} (2.2)

\[ \delta_{\text{max}} = \frac{2^N}{2^{2N} - 2^N + 1} \]  \hspace{1cm} (2.3)

Equation (2.2) and equation (2.3) describe actually the same relationship in different forms. When the technology is definite, the capacitor mismatch error \( \delta \) is fixed, the maximum value of capacitor array \( N \) can be calculated by equation (2.2). On the other hand, when the technology is not definite, it can be chosen by the result calculated by equation (2.3).

Figure 2.5: Maximum possible resolution of a charge redistribution SAR ADC with a fully binary-weighted capacitor array depending on the matching of an unity capacitor.
The Figure 2.5 shows that the maximum possible resolution of a charge redistribution SAR ADC with a fully binary-weighted capacitor array depends on the matching of an unity capacitor in [56].

2.1.4 Offset mismatch

An ADC offset is a random additive error typically resulting from the comparator offset. In a single-channel ADC, the offset error creates a DC tone and the comparator is usually designed to have input-referred noise less than 1 LSB. With low power supply voltage, 1 LSB should be less than $V_{ref}/2^{N_{bit}}$. For example, 1 LSB is 244.14 µV in the 12bit SAR ADC using 1V $V_{ref}$. The impact of the offset errors is much more detrimental in time-interleaved ADCs [57]. If $o(i)$ is the offset of the $i$th channel, then, for a given input signal $v_{in}(t)$, assuming no other errors, the output signal can be written as:

$$D_{out}(n) = v_{in}(nT) + o((n - 1) \mod M + 1), \quad (2.4)$$

where mod is a modulo operation, and $o((n - 1) \mod M + 1)$ is a periodic discrete signal with a period equal to M. This means that in addition to our desired signal $v_{in}(t)$, the spectrum of the output signal will have tones at frequencies that are multiples of $f_s/M$. The magnitude and relative strength of these tones depends on the amplitude and the shape of the introduced periodic error signal.

The input referred offset of the comparator used in the ADC will possibly degrade the overall ADC performance, so it is better to be minimized. Such offset is primarily functions of both threshold mismatch and current factor mismatch, and transistor dimension as below [58]

$$\sigma(\Delta V_t) = \frac{A_{Vt}}{\sqrt{W \cdot L}} \quad (2.5)$$

$$\frac{\sigma(\Delta \beta)}{\beta} = \frac{A_{\beta}}{\sqrt{W \cdot L}} \quad (2.6)$$
where $\Delta V_t$ is the threshold voltage defences, current factor differences $\Delta \beta$ ($\beta = \mu C_{ox} W/L$ [59]), $W$ is the gate-width and $L$ the gate-length, and the proportionality constants $A_{Vt}$ and $A_\beta$ are technology-dependent.

The effect of the mismatch become serious with scaling process. The matching of the minimal size device degrades with scaling as can be seen in Figure 2.6 for nMOS transistors. This is an important concern for the design of digital circuits since the device mismatch starts affecting the noise margin [60] and mismatch mitigation techniques cannot be widely applied due to their large area overhead [58].

![Matching of minimum size nMOS device](image)

Figure 2.6: $\sigma(\Delta V_T)$ (blacksquare) and $\sigma(\Delta \beta/\beta)$ (diamond) for a minimal nMOS device in different technology nodes

Obviously, according to equation (2.5), the transistor size need to be increased by 4 times in order to reduce the offset by 2 times. This calculation shows that the sizing is not practical due to the excessive area and power consumption cost. However offset calibration is necessary to efficiently achieve good accuracy.

The basic idea for the calibration is to deliberately introduce some imbalance to compensate for the offset. Such imbalance can be either by capacitance loading [61] or
current injection \[62\], or even voltage difference \[63\]. To calibrate the offset voltage, the differential inputs of the comparator are usually tied together to a certain common mode voltage, which should be the same when the comparator is in real operation, and the comparator output is monitored and fed back to the state machine to control the imbalance.

### 2.1.5 Gain mismatch

Gain errors manifest itself as a change in the slope of the transfer function of an ADC. The gain error comes from a difference in reference voltages or from the sampling operation (e.g. charge injection). The gain of the \(i^{th}\) channel can be expressed as \(1 + \Delta g_i\), where \(\Delta g_i\) is the gain error in the \(i^{th}\) channel. The composite output of the time-interleaved ADC can be written as:

\[
D_{\text{out}}(n) = v_{in}(nT) + \Delta g((n - 1) \mod M + 1)v_{in}(nT)
\]  

(2.7)

where \(\Delta g((n - 1) \mod M + 1)\) is a periodic discrete signal with a period of \(M\) and can be represented in frequency domain by discrete tones at frequencies \(k f_s / M\), \(k = 0..M - 1\). If the input signal is a sinusoid with the frequency \(f_{in}\), the mixing effect of multiplying the input signal with the periodic signal \(\Delta g((n - 1) \mod M + 1)\) will create tones at frequencies \(k f_s / M \pm f_{in}\).

### 2.2 Current self-calibration techniques for SAR ADC

As discussed in section 2.1, the capacitor mismatches and offset error in SAR ADCs creates non-linearities in the ADC transfer function. Many different calibration
techniques have been developed in order to fix the effect of mismatch-caused non-linearities. The techniques can be analog or digital circuitry [64].

2.2.1 Analog calibration

One group of analog techniques corrects the mismatches by subtracting a signal equal to the error caused by the mismatches from the output of the capacitive DAC. A single calibration DAC combined with a digital logic can be used for this subtraction [65] or every capacitor in the main DAC can have its own calibration DAC [66].

The second analog offset calibration circuit was presented for a modified resistive divider based dynamic comparator in SAR ADC for biomedical applications [67]. The calibration circuit works on the basis of the comparator working as a zero crossing detector.

The third group of analog techniques corrects the mismatches by modifying the effective value of the capacitors in the main DAC. This can be achieved by using small trimming capacitors that are connected in parallel with the main DAC capacitors [68]. The mismatches can be measured by using either a known precise input signal or a self-calibration technique [65]. In the self-calibration technique, the difference between each capacitor in the array and the sum of all capacitors at the lower bit-positions is measured and later used for the mismatch correction.

2.2.2 Digital calibration

A digital calibration techniques measures or infers the values of the capacitors, represent them as a set of digital coefficients, and then corrects the non-linearities in the digital domain by calculating the weighted sums of those coefficients for each conversion [69].

The calibration method includes not only additional circuitry to assist the main circuits but also using different algorithm. The algorithm in [70] uses a dynamic error
correction (DEC) capacitor to cancel the static errors occurring in each capacitor of the array as the first step upon power-up and eliminates the need for an extra calibration DAC.

Another group uses a charge-pump-based offset tracking method to reduce offset voltage of the comparators in flash-assisted time-interleaved SAR ADC. The comparator reflects the feedback information from the charge pump [71].

The recently published method is to use redundancy dithering technique [72]. The digital calibration technique of SAR ADC is based on the principle of internal redundancy dithering, a technique in which the bit decision thresholds are dithered by a pseudo-random bit sequence within the redundancy region [72].

### 2.2.3 Foreground & background calibration

Depending on the operation method to calibrate the mismatch error, there are two methodologies to modify the mismatch-caused non-linearities, which are foreground and background performed techniques. The foreground techniques require to interrupt the operation of the ADC as can be seen in Figure 2.7 [1, 73–79]. In Figure 2.7, the additional DAC for design for test (DFT), called d-DAC, is added for calibration. However, the circuitry can not be used during operation of the ADC to save power. The background techniques do not require to interrupt the operation of the ADC. However, the circuitries have to work during operation time of the ADC [69, 72, 78, 80, 91].
transfer curve can be constructed by a series of inductive calculations. Note that MCT (i) of the DAC capacitor array is just $V_{MCT} = V_{i} - V_{j}$, for $j < i$ and $C_{j}$ is set to be less than $C_{i}$. For ease of illustration, we now consider a 4-bit SAR ADC, with $V_{ref} = 2V$.

**Figure 2.7: The self-testing and calibration architecture in [1].**
Chapter 3

Proposed self-calibration research approach

This chapter discusses a digital-compatible self-calibration approach for SOC building blocks such as DAC and comparator to improve the linearity of SAR based ADC.

Since the calibrated performance is not sensitive to mismatches and process variations, the calibrated circuits can be easily used without additional cost. Simulation and experimental results verify the test performance of the proposed technique.

3.1 Proposed calibration approach for SAR ADC

3.2 Proposed offset calibration for comparator

The primary objective of this dissertation is to reduce the offset error of the comparator in SAR ADC. The main cause of the offset of the comparator is the device mismatch as can be seen in subsection 2.1.4.

The comparator in SAR ADC should be able to compare the voltage difference of at least 1 LSB. For example, for 10-Bit ADC with 1V reference voltage $V_{REF}$, the 1 LSB should be $1/2^{10} = 0.976mV$. However, the offset voltage of reported comparators
used for offset calibration is larger than 10mV \[92\text{–}95\]. Therefore, in order to use the comparator for the proposed self-calibration technique for DAC mismatch, the offset voltage should be minimized.

![Comparator schematic](image)

Figure 3.1: Schematic of the comparator with pre-amplifier for the SAR ADC

Figure 3.1 shows that the schematic of the comparator with pre-amplifier for the SAR ADC. The output of the OR gate is used for the valid signal for asynchronous clock operation. When the comparator is in the pre-charge operation, the valid signal is low because the two input voltage of the OR gate, i.e., two output of the comparator is low. When the comparator is in comparison operation and the operation is finished, the valid signal is high because the one of the input voltage of the OR gate, i.e., one of the output of the comparator is high.

The offset voltage can be compensated by controlling the capacitance, current or threshold voltage of the differential output pair \((Di+, Di−)\) of pre-amplifier \[93\text{–}96\]. In order to reduce the offset voltage of the comparator, the proposed self-calibration techniques change the capacitance of the output node of the pre-amplifier.
Figure 3.2: Proposed self-calibration approaches (a) Schematic of the comparator with capacitor array to reduce the offset voltage and (b) block diagram for the self-calibration approach.
The capacitor arrays to control the capacitance of the output pair of the pre-amplifier consisting of MOM (Metal-Oxide-Metal) capacitors by using PMOS transistors as shown in Figure 3.2(a). Each node has 6 capacitor arrays ($C_0 > C < 5$, $C' < 0 > C' < 5$). The capacitor array has the binary-weighted size like the capacitor array of DAC in the main SAR ADC, while the size of the unit capacitance is minimized as small as possible.

Figure 3.2(b) shows that overall block diagram of the self-calibration approach for the comparator. When the self-calibration mode for the comparator is selected ($En$ is on), DAC driver is disconnected to SAR logic and the voltage of the two inputs of the comparator goes to $V_{cm}$. The function of calibration is similar to the function of SAR ADC, and the SAR logic can be used by modifying the logic for the self-calibration operation. The modified SAR logic can be used for both normal ADC operation and self-calibration approach.

### 3.3 Proposed DAC mismatch calibration in SAR ADC

Figure 3.3(d) the proposed SAR based ADC structure and the main conceptual strategy of the proposed self-calibration for DAC in SAR based ADC to reduce mismatch error. The proposed self calibration method adjusts the auxiliary capacitors that are connected in parallel to each capacitor ($C_0, C_1, C_2, .., C_4$ in Fig.1) of the DAC to minimize the mismatch error of the DAC in SAR based ADC. In calibration mode, the auxiliary capacitances are determined using 4-bit control codes, then the digital code to adjust the auxiliary capacitors is stored in the register with 4-bit word-line. For 12-bit SAR based ADC, 12 registers with 4-bit word-line are required to control the auxiliary capacitance array to compensate the capacitance mismatch error. After
all of the 12 codes are saved, the SAR based ADC operates without the calibration controller causing no additional power consumption.

The calibration controller controls the auxiliary capacitors so that the values of the binary weighted capacitors can be nearly mismatch-less values. Each auxiliary capacitors consist of four capacitors which are unit auxiliary capacitor and binary weighted capacitors of the unit auxiliary.

Figure 3.4 shows the calibration cases and its diagram. First procedure is to compare the unit capacitor, $C_0$ and $C_1$. Depending on the comparison outcome, auxiliary capacitor, $\alpha$ or $\beta$ are added to $C_0$ or $C_1$ to make $C_0$ and $C_1$ euqal. Any auxiliary capacitance is made larger than the mismatch capacitance, and the compensated capacitance using the axiliary capacitance is a little larger than the ideal capacitance. In the second procedure, two unit capacitors, $C_1$ and $C_0$ are added, and the $C_2$ ($=2C_1$
Figure 3.4: (a) Calibration algorithm and (b) diagram of the calibration process
ideally) and $C_0 + C_1$ are compared, then the auxiliary capacitor array, $\gamma$ is connected (added) to $C_2$. By repeating above operation, the auxiliary capacitors of $2^NC_{N+1}$ are controlled, and the calibration controller saves the each switching code of the axillary capacitors. If $D$ case is selected in Figure 3.4(b), it means that $C_2$ is larger than $C_0 + C_1$. However, this case is avoided by making any auxiliary capacitance larger than the capacitance mismatch value assuming that the capacitance mismatch value in the given technology node is given. Therefore, the proposed algorithm always choose $C$ case, and $D$ case is never chosen, i.e. the auxiliary capacitor is only added to the left side. The flow chart of the proposed algorithm for self-calibration is in Figure 3.5.

For example, firstly, compare $C_0$ and $C_1$ as following:

\[
A : C_1 < C_0 \rightarrow C_1 + \alpha = C_0 \\
B : C_1 > C_0 \rightarrow C_1 = C_0 + \alpha
\]

Then, one of the two case, $A$ and $B$, should be chosen. Regardless of the choice, the sum of $C_1$, $C_0$ and $\alpha$ is the same. Then, compare $C_2$ and the sum of $C_1$, $C_0$, and $\alpha$

\[
C : 2C_2 < C_0 + C_1 + \alpha \rightarrow 2C_2 + \beta = C_0 + C_1 + \alpha \\
D : 2C_2 > C_0 + C_1 + \alpha \rightarrow 2C_2 = C_0 + C_1 + \alpha + \beta
\]

In this case, Regardless of the choice of $C$ or $D$, the sum of $2C_2$, $C_1$, $C_0$, $\alpha$, and $\beta$ is also the same. However, if $D$ is selected, the $\beta$ value should be included to $C_a \sim C_c$, which means that the algorithm needs to reconfigure $C_c$. Therefore, the proposed algorithm let $C$ set be chosen, i.e. the auxiliary capacitor is only added to the left side. In order to do this operation, $2C_2$ are set up 10% smaller than $C_0 + C_1 + \alpha$ in
second comparison, so that only $C$ should be selected. The flow chart of the proposed algorithm for self-calibration is in Figure 3.5.

The detailed procedure for the modified proposed algorithm is as follows:

1. compare $C_0$ and $C_1$ as follows
   
   $A : C_1 < C_0 \rightarrow C_1 + \alpha = C_0$
   
   $B : C_1 > C_0 \rightarrow C_1 = C_0 + \alpha$

2. compare $2C_2$ and $C_1 + C_0 + \alpha$ as follows
   
   $2C_2 < C_0 + C_1 + \alpha \rightarrow 2C_2 + \beta = C_0 + C_1 + \alpha$

3. compare $4C_3$ and $2C_2 + C_1 + C_0 + \beta + \alpha$ as follows
   
   $4C_3 < C_2 + C_0 + C_1 + \beta + \alpha \rightarrow 4C_3 + \gamma = 2C_2 + C_1 + C_0 + \beta + \alpha$

4. compare $2^NC_{N+1}$ and $2^{N-1}C_N + 2^{N-2}C_{N-1} \cdots + C_0 + \beta + \alpha$ as follows
   
   $2^NC_{N+1} < 2^{N-1}C_N + 2^{N-2}C_{N-1} \cdots + C_0 + \beta + \alpha \rightarrow 2^NC_{N+1} + x = 2^{N-1}C_N +$
   
   $2^{N-2}C_{N-1} \cdots + C_0 + \beta + \alpha + x$

Each the value of $\alpha$ or $\beta$ is decided by SAR-logic operation and the SAR-logic for main ADC is reused for the operation.

To avoid a high-frequency clock generator and a pulse width modulator (PWM) for SAR logic, the proposed ADC uses an asynchronous control circuit to internally generate the necessary clock signals and to reduce switching power consumption. The generated clock signal is used for the comparator and SAR logic. However, for self-calibration mode, the SAR ADC uses input sampling clock frequency for the self-calibration controller and its operation because the self-calibration mode does not have to be run fast. In the proposed design, the sampling frequency is 32MHz and the calibration controller and the comparator in SAR ADC is operated by the same clock during calibration mode. The calibration controller and 4-bit registers are synthesized using digital standard cell library supported by the process foundry company.
Figure 3.5: Flow chart of proposed algorithm for N-Bit DAC for self-calibration.
Mismatch errors are inevitable due to process variations. Special layout techniques as well as laser trimming are used to reduce matching errors. However, these methods lead to significant cost increases. For the independent DAC for general purpose, the dynamic element matching (DEM) technique accepts matching errors as inevitable and dynamically rearranges the interconnections of the mismatched elements so that all element values are nearly equal on the average \[97\]. However, because the DAC in SAR ADC operates with binary search algorithm normally and consists of binary weighted capacitor or resister, the DEM is not suitable technique for DAC in SAR ADC. Therefore, new approach is required to reduce the mismatch errors.

### 3.3.1 An inverter based comparator to consider offset error

There is a need to convert the charge level quantities to voltage level quantities in order to compare the value of \(C_1\) and \(C_0\) in Figure 3.6. The Figure 3.6 shows the conventional comparison concept of \(C_1\) and \(C_0\). In first step, \(C_1\) and \(C_0\) are charged to zero. Then, \(C_1\) is charged to \(V_{REF}\) as next step. The the voltage of the node \(V_1\) is calculated as follows.

![Diagram showing conventional comparison concept of \(C_1\) and \(C_0\)](image)

Figure 3.6: Conventional comparison concept of \(C_1\) and \(C_0\)
\[Q_1 = V_{REF} \times C_1\]
\[= V_1 \times (C_1 + C_0)\]  \hspace{1cm} (3.1)

Therefore, the voltage of the node \(V_1\) for the next step should be

\[V_1 = \frac{C_1}{C_1 + C_0} V_{REF}\]  \hspace{1cm} (3.2)

By using the comparator, \(V_1\) and \(V_{REF}/2\) are compared. If there are assumptions that the values of \(C_1\) and \(C_0\) are same, the voltage of the node \(V_1\) is should be \(V_{REF}/2\). The voltage difference between \(V_1\) and \(V_{REF}/2\), \(\Delta V\) is decided by the difference between the values of \(C_1\) and \(C_0\), i.e, \(\Delta C/C\) in equation (2.1). For instance, the capacitor mismatch error \(\delta\) is estimated about 0.005 for 0.18\(\mu\)m standard CMOS process from the analysis of the equation (2.1), where \(\delta\) in equation (2.2) is equal to \(\Delta C/C\) in equation (2.1) \[98\]. In addition, ideal \(\Delta V\) is 4.5mV for 1.8V supply voltage and the realistic value should be smaller than 4.5mV with 0.18\(\mu\)m standard CMOS process so that the \(\Delta V\) is smaller than the offset voltage of normal comparator \[99\]. The reported offset voltage \(\delta V_{os}\)s are 1.68mV \(\sim\) 31.8mV \[92\, 95\]. Therefore, the inverter based \(\Delta V\) sensing circuit is proposed.

The Figure 3.7 shows that the schematic of the proposed \(\Delta V\) sensing circuit. The circuit consists of three switches and one inverter. The operation of the circuit is described as following:

When the \(CLK_1\) is high, \(VR_1\) is connected to \(V_1\) and then, \(V_2\) and \(V_3\) are connected so that the charge of \(C_2\) is zero. Therefore, the charge of \(C_1\) and \(C_2\) is given by equation (3.3) and (3.4), respectively.
Figure 3.7: The schematic of proposed $\Delta V$ sensing circuit

\[ Q_1 = (V_1 - V_2) \times C_1 \]
\[ = (V_{R1} - \frac{V_{DD}}{2}) \times C_1 \]  \hspace{1cm} (3.3)

\[ Q_2 = (V_3 - V_2) \times C_2 \]
\[ = 0 \times C_2 \]  \hspace{1cm} (3.4)

where, $V_2$ will be high if $V_3$ is low, and $V_3$ will be low if $V_2$ is high. Therefore, the voltage of $V_2$ and $V_3$ goes to $\frac{V_{DD}}{2}$. 

32
When the $CLK2$ is high, $VR_2$ is connected to $V_1$ and then, $V_3$ will be changed by equation (3.5) and (3.6).

$$Q_1' = (V_1 - V_2) \times C_1$$
$$= (VR_2 - \frac{VDD}{2}) \times C_1 \quad (3.5)$$

$$Q_2' = (V_3 - V_2) \times C_2$$
$$= (V_3 - \frac{VDD}{2}) \times C_2 \quad (3.6)$$

Since the total charge is conserved, $V_3$ is given by equation (3.7).

$$Q_1 + Q_2 = Q_1' + Q_2'$$
$$V_3 = (VR_1 - VR_2) \frac{C_1}{C_2} + \frac{VDD}{2} \quad (3.7)$$

where, $V_3$ is decided by the multiplication between the ratio of $C_1$ and $C_2$ and the difference $V_1$ and $V_2$. By adding the $(V_1 - V_2)$ term, $V_3$ is amplified. The maximum gain of proposed circuit is about $3V/V$ as verified in simulation.

Figure 3.8 shows the DAC with the proposed $\Delta V$ sensing circuit. The $\Delta V$ sensing circuits are only connected to capacitor array of DAC when the self-calibration mode turns on. The operation of calibration does not use the clock for DAC comparison but the ADC sampling clock. The ADC sampling clock is at least 10 times slower than the clock for DAC comparison, and the large switch size is not necessary. However, each size of the switch is different to reduce settling time for suitable operation margin. For instance, the size of the inverter of the last sensing circuit connected to the $2^{10}C$ capacitor for MSB decision is 3 times larger than size of inverter of the first sensing circuit of unit $C$ capacitor for LSB decision.
Figure 3.8: The DAC with the integrated ΔV sensing circuit


3.4 Simulation Result

This section presents the simulation result for the proposed self-calibration approach. The digital controller is written in Verilog code and the overall simulations were performed with Cadence Virtuoso™ in AMS (Analog-Mixed Signal) mode using device models of a 130nm standard CMOS process.

The proposed circuits were designed in a 130nm standard CMOS process. The control logic circuit for the self-calibration has been optimized for power consumption and area using asynchronous logic. The input signal capacitance for Sample and Hold Amplifier (S/H) and total capacitance of capacitive DAC are 1.05 pF and 20.24 pF, respectively. The area estimated with RTL-Compiler for the calibration control circuits including registers is 0.0532 mm$^2$, and the area overhead of the proposed method is 14%.

3.4.1 Offset voltage calibration

Figure 3.9 shows the simulated histogram of the offset voltage for the pre-amplifier and comparator in Figure 3.1. It contains 100 Monte Carlo simulations run with the random mismatch model from Equation (2.5) and (2.6), where $A_{Vt} = 3.8mV\mu m$ and $A_{\beta} = 0.8\%$, and the total input referred offset voltage was measured by applying slowly varying slope signals to the comparator inputs. The values of $A_{Vt}$ and $A_{\beta}$ are supported by the PDK (Process Design Kits) document of the foundry. The input differential transistor size is 1.2$\mu m$/0.15$\mu m$. As shown in Figure 3.9, 1-sigma the standard deviation for the input referred offset voltage is 14mV. The result shows that the offset compensation is needed to reuse the comparator for the proposed calibration approach.

Figure 3.10 shows the compensation range of the digitally controlled capacitive calibration. 6 capacitor arrays control the offset voltage from -40mV to 40mV so that
Figure 3.9: Monte Carlo histogram of the offset voltage of the pre-amplifier and comparator

the step of 1 digital code is around 2.5mV, which means that a compensation range of ±40mV is larger than the needed 3-sigma offset voltage.

Figure 3.11 displays the simulated histogram of the offset voltage for the pre-amplifier and comparator with the proposed self-calibration technique by performing 100 Monte Carlo simulations. After calibration, the residual offset voltage is around 3mV.

The calibrated comparator is used for 32MS/s 12-bit charge redistribution SAR ADC using the conventional regular capacitor array.

### 3.4.2 Capacitor mismatch calibration

The proposed calibration technique for DAC mismatch error was implemented for 32MS/s 12-bit charge redistribution SAR ADC using the conventional regular ca-
Figure 3.10: The compensation range of the digitally controlled capacitive calibration capacitor array. The SAR ADC has the DAC capacitor array having intentional 5% capacitance mismatch errors to verify the proposed technique. The SAR logic, the comparator, and the capacitor array in the SAR ADC is implemented by 0.13 µm CMOS process and the switch controller to control the auxiliary capacitor array is implemented by verilog code using behaviour model.

Figure 3.12(a) shows the raw transfer curve of the 12-bit SAR ADC for a case with 5% mismatch error of capacitance. The X-axis is the analog input and the Y-axis is the raw digital codes from the ADC output. Even if the example assumes 5% capacitor mismatch (σ=5%), the transfer curve only exhibits missing codes. The self-calibration can treat the missing codes problem by learning the optimal bit weights, and transfer curve can be had a linearity.

Figure 3.12(b) shows the calibrated raw transfer curve with the proposed calibration technique of the 12-bit SAR ADC with +5% mismatch error of capacitance.
Figure 3.11: Monte Carlo histogram of the offset voltage of the pre-amplifier and comparator using the proposed calibration approach.

Compared to Figure 3.12(a), the linearity has improved conspicuously with the calibration approach.

The summarized overall performance of the proposed SAR based ADC is shown in Table 3.1. The proposed ADC achieves an ENOB of 11.08 bit and SNDR of 65.2 dB for 16 MHz input frequency. At 32 MS/s, the average power consumption including the output buffers and the offset controller is 3.57 mW. Compared to the ADC operation without the offset controller, the power consumption is not increased because the calibration and its controller circuits are active only during the calibration mode before the main circuits are activated. In calibration mode, the power consumption of calibration controller is 723 µW.

The simulated static performance for differential nonlinearity (DNL) and integral nonlinearity (INL) of the proposed ADC are shown in Figure 3.13. The peak DNL values are between -0.87 to 0.91 LSB and the peak INL values are -1.86 to 1.12 LSB.
Table 3.1: Performance summary for the proposed SAR ADC

<table>
<thead>
<tr>
<th>Specifications</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>130 nm</td>
</tr>
<tr>
<td>Resolution</td>
<td>12 b</td>
</tr>
<tr>
<td>Supply</td>
<td>1.2 V</td>
</tr>
<tr>
<td>Input range</td>
<td>0.2 - 1.2 V</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>32 MS/s</td>
</tr>
<tr>
<td>DNL / INL (LSB)</td>
<td>-0.87<del>0.91 / -1.86</del>1.12</td>
</tr>
<tr>
<td>ENOB</td>
<td>11.08 @ Fin = 16 MHz</td>
</tr>
<tr>
<td>SNDR</td>
<td>65.2 dB</td>
</tr>
<tr>
<td>SFDR</td>
<td>76.1 dB</td>
</tr>
<tr>
<td>Power</td>
<td>3.57 mW</td>
</tr>
</tbody>
</table>

Figure 3.14 shows the dynamic performance of the proposed ADC with the data of simulated SNDR and SFDR vs. input frequency at 32MS/s. When the input frequency is 1 MHz, the ADC has peak SNDR and SFDR of 65.2 dB and 76.1 dB, respectively.
(a) Transfer curve of a 12-bit SAR ADC with 5% mismatch error of capacitance without proposed calibration technique

(b) Transfer curve of a 12-bit SAR ADC with 5% mismatch error of capacitance with proposed calibration technique

Figure 3.12: Transfer curve of a 12-bit SAR ADC with 5% mismatch error of capacitance (a) without proposed calibration technique (b) with proposed calibration technique
Figure 3.13: Simulated INL and DNL at 32MS/s
Figure 3.14: Simulated SNDR and SFDR performance versus input frequency at 32MS/s
Chapter 4

One-step self calibration approach

This chapter presents a low-power 12-bit 50MS/s successive approximation register (SAR) analog-to-digital converter (ADC) using a single input condition for Built-In Self Test (BIST) with a novel self-calibration scheme to reduce both offset voltage of a comparator and capacitor mismatch of the DAC.

The calibration approach in the previous chapter reduces both the offset voltage of the comparator and the DAC mismatch as each independent operation. In addition, numbers of auxiliary capacitor arrays are added to the capacitor array of DAC in order to reduce the DAC mismatch, thus unexpected parasitic capacitor can be occurred. The proposed self-calibration scheme reduce the effect of the offset voltage of the comparator and the DAC mismatch by one operation simultaneously. the calibration approach adjusts the offset voltage of the comparator continuously for every step. The adjusted offset voltage of the comparator cancels not only inherent offset of the comparator but also the mismatch in the DAC. Consequently, the total mismatch error of both the comparator and capacitor of DAC can be reduced. Compared to the converters that use the conventional procedure, the INL and DNL are reduced by about 47% and 52% more, respectively. The prototype was designed using standard 0.13\(\mu\)m single poly 6 metal CMOS technology. Using 1.2V supply and the sampling
rate of 50 MS/s, the ADC achieves a SNDR of 65.6 dB and consumes 4.62 mW. The ADC core occupies an active area of only $240\mu m \times 298\mu m$.

4.1 Proposed self-calibration methodology

Figure 4.1 visualizes an example for 4-bit SAR ADC operation to obtain digital output codes and a corresponding SAR architecture. In the case of time period for second MSB decision and LSB decision, the errors can be occurred by mismatch errors in this particular case. In addition, the value of mismatch errors in the second time period, $e(a)$ and $e(b)$ should be same.

The proposed self calibration method adjust the input refereed offset voltage of the comparator in Figure 4.1(b) for each conversion period. In the first step called $MOD1$, the offset controller in Figure 4.2 measure the offset voltage of the comparator, then adjust the 8 capacitor arrays in Figure 4.2. The digital code to adjust offset voltage of the comparator is stored in 8-bit register. For 12-bit SAR ADC, 12 registers with 8-bit word-line are required. After all of the 12 codes are saved, the offset controller changes the offset voltage of the comparator continuously by using the stored code in the registers.

More details about getting the code to adjust the offset voltage of the comparator are as follows. First, the $V_{in}$ is set up to $V_{ref}/2$ as reference voltage, i.e., 0.6V. At the same time, DAC sets up its output voltage at the $V_{ref}/2$ using the capacitor array and charge redistribution algorithm. This voltage is not going to be exactly $V_{ref}/2$ because there is a mismatch error. Therefore, the reference voltage is different from DAC output voltage by as much as mismatch error. The comparator offset controller finds the error code for the capacitor arrays that minimize the mismatch error by changing the number of connected capacitor arrays.
(a) A example of 4-bit SAR ADC operation

(b) Example SAR ADC architecture

Figure 4.1: An example of 4-bit SAR ADC operation (a) and SAR ADC architecture (b)
To avoid a high-frequency clock generator and a pulse width modulator (PWM) for SAR logic, the proposed ADC uses an asynchronous control circuit to internally generate the necessary clock signals and to reduce switching power consumption. The generated clock signal is used for the comparator and SAR logic. However, for $MOD1$ operation, the SAR ADC uses input sampling clock frequency because $MOD1$ process does not have to be run fast. In the proposed design, the sampling frequency is 50MHz and the comparator is operated by the same clock during $MOD1$ operation. For normal operation, the SAR logic does not support the offset controller and it bypasses the output of the comparator to DAC driver. However, SAR needs to support both offset controller and DAC during $MOD1$. The eight capacitor arrays connected to the comparator are used to adjust the effective input voltage of the comparator by compensating the offset voltage.
The offset controller and 8-bit registers are synthesized using digital standard cell library supported by the process foundry.

### 4.1.1 Digital self-calibration scheme

### 4.2 Simulation result

![Figure 4.3: The layout of the proposed ADC](image)

The proposed system was designed in a standard 130nm CMOS process. Figure 4.3 shows the layout and floor-plan of the core parts. The total occupied area including power-ring of the ADC is 0.072 $mm^2$, with the ADC core taking only 0.058 $mm^2$. The switches for capacitors are placed close to the capacitor arrays to reduce any parasitic components. The logic control circuit has been optimized for power consumption and area using asynchronous logic, and the layout of the digital logic circuit is more
Table 4.1: Performance summary for the proposed SAR ADC

<table>
<thead>
<tr>
<th>Specifications</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>130 nm</td>
</tr>
<tr>
<td>Resolution</td>
<td>12 b</td>
</tr>
<tr>
<td>Total area</td>
<td>0.0715 mm²</td>
</tr>
<tr>
<td>Active area</td>
<td>0.0582 mm²</td>
</tr>
<tr>
<td>Supply</td>
<td>1.2 V</td>
</tr>
<tr>
<td>Input range</td>
<td>0.4 - 1.2 V</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>50 MS/s</td>
</tr>
<tr>
<td>DNL / INL (LSB)</td>
<td>-0.82<del>0.75 / -1.42</del>1.05</td>
</tr>
<tr>
<td>ENOB</td>
<td>11.04 @ Fin = 25 MHz</td>
</tr>
<tr>
<td>SNDR</td>
<td>65.6 dB</td>
</tr>
<tr>
<td>SFDR</td>
<td>75.1 dB</td>
</tr>
<tr>
<td>Power</td>
<td>4.62 mW</td>
</tr>
</tbody>
</table>

The summarized overall performance of the proposed ADC is shown in Table 4.1. The input signal capacitance for Sample and Hold Amplifier (S/H) and total capacitance of capacitive DAC are 1.05 pF and 20.24 pF, respectively. The size of the synthesized offset controller including registers is 0.0081 mm², and the area overhead of the proposed method is 12%.

The proposed ADC achieves an ENOB of 11.04 bit and SNDR of 65.6 dB for 25 MHz input frequency. At 50 MS/s, the average power consumption including the output buffers and the offset controller is 4.62 mW. Compared to the ADC operation without the offset controller, the power increased consumption is 23%. However, the power consumption of the offset controller will be decreased further with process scaling.

The simulated static performance for differential nonlinearity (DNL) and integral nonlinearity (INL) of the proposed ADC are shown in Figure 4.4. The peak DNL values are between -0.82 to 0.75 LSB and the peak INL values are -1.42 to 1.05 LSB. Figure 4.5 shows the dynamic performance of the proposed ADC with the data of simulated SNDR and SFDR vs. input frequency of 50MS/s. When the input
Figure 4.4: Simulated INL and DNL at 64MS/s

frequency is 1 MHz, the ADC has peak SNDR and SFDR of 65.6 dB and 75.1 dB, respectively.
Figure 4.5: Simulated SNDR and SFDR performance versus input frequency at 64MS/s
Chapter 5

High sampling speed approach using a variable clock period method

5.1 Conventional SAR ADC

Figure 5.1: A conventional SAR ADC architecture
Figure 5.1 shows the conventional SAR ADC architecture using a charge-redistribution DAC for a single input ADC, where S/H is sample-and-hold amplifier. As can be seen in Figure 5.1, total capacitance of capacitor array of the DAC is \(2^N\) times of the unit capacitor C for LSB, where N is the resolution of SAR ADC \([50]\). For example, 10-Bit ADC needs 1024C of total capacitance for DAC. Especially, the size of the capacitor for MSB is the largest as 512C. Therefore, the switch for MSB should be larger than others, which means a big size buffer is required to drive the switch and the die area of the DAC driver must be increased. Generally, an inverter chain is used to drive big size buffer and the timing margin is reduced due to the delay of the inverter chain. In addition, the current for those capacitors come from \(V_{REF}\) and \(V_{REF}\) should have a small input resistance \(R_{IN}\) in order to charge the capacitor rapidly, which means that \(V_{REF}\) should be designed carefully because \(V_{REF}\) may suffer from IR-drop. The \(V_{REF}\) IR voltage drop has a significant effect on ADC reducing the linearity of the SAR ADC.

Figure 5.2 shows an example of 4-bit SAR ADC operation to get digital output codes, where the \(V_{DAC}\) signal in Figure 5.1 is shown. While the first switch turns on to decide the MSB code, a half of the total current of SAR DAC passes through the switch of MSB and long settling time is needed due to the large capacitance \([100]\). For the second step, the settling time is shorter than first step because the value of capacitor for the second MSB is a half of the capacitor value for the first MSB. Therefore, the settling time for first step to decide the MSB should be taken the longest time and the clock period for SAR logic is dependent on the settling time of first step. Of course, the larger the size of the switch is, the shorter is the settling time. However, the size of the switch cannot be increased too much due to not only the die area but also peak current issues. Even though the high peak current reduces the settling time, it causes a larger IR-drop as a side effect as can be seen in Figure 5.3.
Figure 5.2: Example of 4-bit SAR ADC operation

Figure 5.3: Settling time with a large capacitor (a) and a small capacitor (b)
5.2 Proposed SAR ADC

As mentioned earlier, the settling optimum time for each bit is different. The first settling time should be the longest for deciding MSB and last settling time should be the shortest for deciding the LSB. Therefore, considering the different settling timing for each bit, a different clock period can be used for the ideal case as shown in figure 5.4 (b). In general, the clock for SAR logic comes from outside the chip or can be generated by internal clock generator. If the clock signal comes from outside the chip, the clock period should be constant. Otherwise, the clock signal can be generated by using independent clock generator or timing skill having delay components. The period of clock is constant in either case. Consequently, the clock period of SAR ADC in practical case is typically constant as shown in Figure 5.4 (a).

![Figure 5.4: A comparison of (a) conventional clock timing for SAR ADC, (b) ideal clock timing and (c) proposed clock timing for SAR ADC](image.png)
This chapter presents an idea to improve the sampling rate of the conventional SAR ADC. The clock period used for the proposed SAR ADC is constant like in other SAR ADCs. However, two clock periods are assigned for the first step while the same one clock period is assigned for all other steps to increase the conversion time for the first bit only as shown in figure 5.4 (c). Therefore, all other subsequent steps do not have to use the longer clock period than necessary, which makes it possible to increase sampling rate. In the conventional SAR based ADC, the clock period of SAR is determined by the time required for the first bit decision that takes longer time than other bits. Then all other subsequent steps have to use a clock period longer than necessary because the subsequent bits do not need a clock period as long as the first bit (MSB). Therefore, time loss and ADC sampling rate limitation are inevitable in the conventional SAR based ADC design. In the proposed SAR design, the total conversion time can be reduced by assigning two clock periods to the first MSB decision and assigning one clock period to all other steps using the same clock frequency. If the resolution of the ADC is increased, the overall conversion time will be more reduced by using the proposed method.

5.2.1 SAR Control Logic Implementation

To avoid a high-frequency clock generator and a pulse width modulator (PWM) for the SAR logic, the proposed ADC uses an asynchronous control circuit to internally generate the necessary clock signals and to reduce switching power consumption. Figure 5.5 shows the logic schematic and timing diagram of the proposed clock generator. The conversion process starts once the system clock $CLK$ is switched to low. Sample is the signal which turns on the sampling switches. After the first valid signal for the first comparison, one clock is skipped so that the time for the first comparison is increased to two clock periods. Other comparisons have the same one clock period for each bit decision. After ten bits decisions, the Sample signal will be set to high to
Figure 5.5: The proposed clock generator
sample the next input signal until the system clock, \( CLK \), switches to low. Therefore, the duty cycle of \( CLK \) makes no odds. The dynamic comparator generates an initial valid signal after each comparison, then the valid signal becomes reversed by a single inverter. The inverted signal is called "valid" signal. Signal \( Clkc \) is the control signal of the dynamic comparator. \( C1 \) to \( C10 \) go to SAR logic which consists of D-flip-flop, and \( D0 \) to \( D9 \) in Figure 5.5 are generated by the SAR logic and they generate control signals for the capacitor arrays with the the digital output codes of the comparator sampled by SAR logic to perform the switchback switching procedure.

By using a similar method, it is possible to assign different clock periods to a certain bit decision step (ex: three periods to the first bit and two periods to the second bit decision steps) with faster internal clock, \( Clkc \), thereby reducing the total A/D conversion time.

## 5.3 Simulation result

![Figure 5.6: Layout of the proposed ADC](image)
Table 5.1: Performance summary for the proposed SAR ADC

<table>
<thead>
<tr>
<th>Specifications</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>130 nm</td>
</tr>
<tr>
<td>Resolution</td>
<td>10 b</td>
</tr>
<tr>
<td>Total area</td>
<td>0.0389 mm²</td>
</tr>
<tr>
<td>Active area</td>
<td>0.0274 mm²</td>
</tr>
<tr>
<td>Supply</td>
<td>1.2 V</td>
</tr>
<tr>
<td>Input range</td>
<td>0.4 - 1.2 V</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>64 MS/s</td>
</tr>
<tr>
<td>DNL / INL (LSB)</td>
<td>-1<del>1 / -2</del>1.7</td>
</tr>
<tr>
<td>ENOB</td>
<td>8.04 @ Fin = 32 MHz</td>
</tr>
<tr>
<td>SNDR</td>
<td>50.2 dB</td>
</tr>
<tr>
<td>SFDR</td>
<td>60.8 dB</td>
</tr>
<tr>
<td>Power</td>
<td>1.325 mW</td>
</tr>
</tbody>
</table>

The ADC was designed in a standard 130nm CMOS technology. Figure 5.6 shows the layout and floor-plan of the core part. The active area including power-ring of the ADC is 0.0389 mm², with the ADC core taking up only 0.0274 mm². The switches for capacitors are placed close to the capacitor arrays. The logic control circuit has been optimized for power consumption and area using asynchronous logic, and the layout of the digital logic circuit is more compact. The input signal capacitance for Sample-and-Hold Amplifier (S/H) and total capacitance of capacitive DAC are 1.05 pF and 5.6 pF, respectively. Consequently, the size of the buffers to drive the switches of capacitive DAC is reduced by 25.2% comparing to the existing approaches.

The overall performance of the proposed ADC is summarized in Table 5.1. The proposed ADC achieves an ENOB of 8.04 bit and SNDR of 50.2 dB for 32 MHz input frequency. At 64 MS/s, the average power consumption including the output buffers is 1.325 mW.

The simulated static performance for differential nonlinearity (DNL) and integral nonlinearity (INL) of the proposed ADC are shown in Figure 5.7. The peak DNL values are between -1 to 1 LSB and the peak INL values are -2 to 1.7 LSB. One
of reason of the nonlinearity of the ADC is the limitation of the comparator. The nonlinearity can be reduced by increasing the size of comparator at the cost of power and area. Figure 5.8 shows the dynamic performance of the proposed ADC with the data of simulated SNDR and SFDR vs. input frequency of 64MS/s. When the input frequency is 2 MHz, the ADC has peak SNDR and SFDR of 53.5 dB and 63.5 dB, respectively.
Figure 5.8: Simulated SNDR and SFDR performance versus input frequency at 64MS/s
Chapter 6

Conclusion

6.1 Conclusion

The self-calibration approaches in mixed-mode circuits in a SOC have been studied in this dissertation. Several novel self-calibration design techniques were proposed for an ADC to reduce mismatch error and improve performance. These are essential in future SOCs and an enhanced self-calibration approach is used to compensate for the process variations.

Firstly, a self-calibration approach to reduce the offset error of the comparator in SAR ADC was proposed. The approach controls the auxiliary capacitor array connected to the output of pre-amplifier to reduce offset voltage. In order to control the capacitor array, a SAR algorithm is used so that the SAR logic in SAR ADC is reused for the approach. The simulation result of the calibrated offset error was obtained as expected in the theoretical analysis. The $1-\sigma$ offset error is reduced from 14mV to 3mV.

Secondly, a self-calibration approach to reduce the DAC mismatch in SAR ADC was proposed. The approach controls the auxiliary capacitor array to reduce mismatch of DAC by using SAR algorithm in a similar way to the first approach. The
modified SAR logic in the SAR ADC is used for the approach to save area and power and the modified SAR logic can be used for both calibration mode and normal SAR ADC operation mode. The simulation results show that the proposed approach works ordinarily with 5% intentional capacitor mismatch error.

Thirdly, the one-step self-calibration approach reduce both the DAC mismatch and the offset voltage of the comparator simultaneously with low parasitic capacitance of the calibration circuitry is proposed. By adjusting one of input voltage of the comparator, both the offset voltage of the comparator and the mismatch of DAC can be cancelled, thus the linearity of the ADC can be increased by 52% with 23% overhead compared to the conventional ADC. Consequently, ENOB is increased by 1.6 in spite of the use of single ended-input SAR ADC. The prototype ADC in 130nm standard CMOS process shows a simulated SNDR of 65.6dB for 25 MHz input frequency and consumes 4.62 mW with output buffer. The proposed ADC will be a good reference to overcome the presumed limit of the resolution of SAR based ADC using single-ended input.

Finally, the high sampling approach using variable clock period method was proposed to increase the sampling clock frequency for the SAR ADC using the self-calibration approach. The proposed clock generator generates internal asynchronous clock, and the high-frequency clock generator is not required. The proposed method adjusts internal clock period to reduce the total conversion time. By using proposed method, not only the conversion time is reduced by about 13% but also the size of DAC driver is reduced by 25.2% compared to the conventional ADC. Consequently, peak current is reduced by 33.8%, thus complicated power design issue such as reference voltage for SAR can be mitigated. The prototype ADC using 130nm standard CMOS process shows the SNDR of 50.2dB for 32 MHz input frequency and consumes 1.325mW with output buffer.
6.2 Future work

The self-calibration techniques for SAR ADC presented in this dissertation reduce test cost and to improve performance in SOC. The proposed approach uses SAR algorithms to tune the capacitor arrays to reduce process variation and to improve yield. Moreover, the approach can be easily applied for other applications. The designs demonstrate that the proposed approaches are useful solutions to address the comparator and DAC mismatch issues. However, more simulation results and experimental results are needed to fully address the known issues. In addition, the proposed approach needs to be refined to reduce area and power overhead to be a better solutions. Therefore, there is a need to research or to develop additional algorithms.

6.2.1 Calibration application for the mismatch of other SAR ADC architectures

A conventional binary weighed capacitor array is generally used for charge charge redistribution SAR ADCs. However, inherently the total size of capacitance is large because the total capacitance is $2^N \cdot C$ ($C$ is unit capacitance). Therefore, split-DAC is one of solution to reduce the total size of capacitance of the DAC. The power consumption and area can be reduced by using split-DAC instead of the conventional DAC. For example, 10-bit regular binary weighed capacitor array requires $2^{10}C$ while split capacitor array needs $2 \times 2^5 = 2^6C$. Figure 6.1 shows the example of a split DAC for 6-bit SAR ADC. The total capacitor required is only $16C$. However, the method is sensitive to parasitic capacitor because the value of capacitance for calculation is multiplied by split capacitor.

If the proposed calibration approach is applied for the SAR ADC using split DAC, the power and area of the SAR ADC can be reduced significantly. In addition, the
effect of offset voltage is more sensitive in time interleaved ADCs. Figure 6.2 displays the block diagram of a time interleaved ADC and its transient clock signal. The ideal output of the time interleaved ADC should have same offset voltage for each N-bit ADC. Therefore, the time interleaved ADC including the proposed calibration technique should be tested.

Figure 6.2: Block diagram of time interleaved ADC
Appendix A

Verilog-HDL source code for the digital controller for one-step calibration

```verilog
module controller_test1(

  mode_sel,
  clk,
  clk_en,
  measure_rst,

  clk_sampling,

  //input measure_end,
  cmp_cap_data_n,
  cmp_cap_data_p,
```


measure_en,
mode_1_finish_flag,
controlled_read_out_data

);

parameter WORD_LENGTH = 7;

input mode_sel;
input clk;
input clk_en;
input measure_rst;
input clk_sampling;

//input measure_end,
input [WORD_LENGTH-1:0] cmp_cap_data_n;
input [WORD_LENGTH-1:0] cmp_cap_data_p;

output measure_en;
output mode_1_finish_flag;
output [WORD_LENGTH-1:0] controlled_read_out_data;
// generate measure_en signal
// frequency dividers
// : enabled when mode_sel == 1
reg clock_div_2;
reg clock_div_4;
reg clock_div_8;
reg clock_div_16;

always @(posedge clk, posedge measure_rst, negedge clk_en)
    if (!clk_en)
        clock_div_2 <= 1'b0;
    else if (measure_rst)
        clock_div_2 <= 1'b0;
    else if (mode_sel) begin
        clock_div_2 <= ~clock_div_2;
    end

always @(posedge clock_div_2 or posedge measure_rst, negedge clk_en)
    if (!clk_en)
        clock_div_4 <= 1'b0;
    else if (measure_rst) begin
        clock_div_4 <= 1'b0;
end else if (mode_sel) begin
  clock_div_4 <= ~clock_div_4;
end

always @(posedge clock_div_4 or posedge measure_rst, negedge clk_en)
  if(!clk_en)
    clock_div_8 <= 1'b0;
  else if (measure_rst) begin
    clock_div_8 <= 1'b0;
  end else if (mode_sel) begin
    clock_div_8 <= ~clock_div_8;
  end

always @(posedge clock_div_8 or posedge measure_rst, negedge clk_en)
  if(!clk_en)
    clock_div_16 <= 1'b0;
  else if (measure_rst) begin
    clock_div_16 <= 1'b0;
  end else if (mode_sel) begin
    clock_div_16 <= ~clock_div_16;
  end

assign measure_en = clock_div_16;

wire [3:0] register_num;
reg [3:0] reg_n;

reg reg_mode_1_finish_flag;
always @(negedge clk or posedge measure_rst, negedge clk_en)
  if (!clk_en)
    reg_mode_1_finish_flag <= 0;
  else if (measure_rst)
    reg_mode_1_finish_flag <= 0;
  else if (register_num == 13)
    reg_mode_1_finish_flag <= 1;

assign mode_1_finish_flag = reg_mode_1_finish_flag;

// binary up counter
// : clocked by clock_div_16
// enabled by mode_sel
always @(negedge clock_div_16, posedge measure_rst, negedge clk_en)

if (!(clk_en) || measure_rst)
    reg_n <= 0;
else if (reg_n == 4'b1111)
    reg_n <= reg_n;
else if (mode_sel)
    reg_n <= reg_n + 1;

assign register_num = reg_n;

reg [1:0] SR_clk_sampling;
wire [1:0] delayed_clk_sampling;

always @(posedge clk)
if (clk_en)
    SR_clk_sampling <= {SR_clk_sampling[0], clk_sampling};

assign delayed_clk_sampling = SR_clk_sampling;
wire [3:0] mode_2_register_num;
reg [3:0] mode_2_reg_n;

// always @(posedge clk, posedge measure_rst, posedge clk_sampling)
always @(posedge clk, posedge measure_rst, negedge clk_en)
if (!clk_en)
    mode_2_reg_n <= 0;
else if (measure_rst || ((delayed_clk_sampling[0] == 1) && (delayed_clk_sampling[1] == 0)))
    mode_2_reg_n <= 0;
else if (!mode_1_finish_flag)
    mode_2_reg_n <= 0;
else if (mode_2_reg_n == 4'b1111)
    mode_2_reg_n <= mode_2_reg_n;
else if (clk)
    mode_2_reg_n <= mode_2_reg_n + 1;

assign mode_2_register_num = mode_2_reg_n;

// 12-set data save

parameter ROM_WIDTH = 7;
always @(posedge measure_en, posedge measure_rst, negedge clk_en)
if (!clk_en || measure_rst) begin
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_01;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_02;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_03;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_04;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_05;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_06;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_07;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_08;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_09;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_10;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_11;
    reg [ROM_WIDTH-1:0] cmp_cap_data_n_12;
    cmp_cap_data_n_01 <= 0;
    cmp_cap_data_n_02 <= 0;
    cmp_cap_data_n_03 <= 0;
    cmp_cap_data_n_04 <= 0;
    cmp_cap_data_n_05 <= 0;
    cmp_cap_data_n_06 <= 0;
    cmp_cap_data_n_07 <= 0;
end
cmp_cap_data_n_08 <= 0;
cmp_cap_data_n_09 <= 0;
cmp_cap_data_n_10 <= 0;
cmp_cap_data_n_11 <= 0;
cmp_cap_data_n_12 <= 0;
end

else if (mode_sel)
  case (register_num)
    4'b0001: cmp_cap_data_n_01 <= cmp_cap_data_p;
    4'b0010: cmp_cap_data_n_02 <= cmp_cap_data_p;
    4'b0011: cmp_cap_data_n_03 <= cmp_cap_data_p;
    4'b0100: cmp_cap_data_n_04 <= cmp_cap_data_p;
    4'b0101: cmp_cap_data_n_05 <= cmp_cap_data_p;
    4'b0110: cmp_cap_data_n_06 <= cmp_cap_data_p;
    4'b0111: cmp_cap_data_n_07 <= cmp_cap_data_p;
    4'b1000: cmp_cap_data_n_08 <= cmp_cap_data_p;
    4'b1001: cmp_cap_data_n_09 <= cmp_cap_data_p;
    4'b1010: cmp_cap_data_n_10 <= cmp_cap_data_p;
    4'b1011: cmp_cap_data_n_11 <= cmp_cap_data_p;
    4'b1100: cmp_cap_data_n_12 <= cmp_cap_data_p;
    // default: cmp_cap_data_n_ <= 0;
  endcase
// read data

reg [WORD_LENGTH - 1 : 0] reg_controlled_read_out_data;
always @(posedge clk, posedge measure_rst, negedge clk_en)
if (!clk_en || measure_rst)
reg_controlled_read_out_data <= 0;
else

case(mode_2_register_num)
4'b0001: reg_controlled_read_out_data <= cmp_cap_data_n_01;
4'b0010: reg_controlled_read_out_data <= cmp_cap_data_n_02;
4'b0011: reg_controlled_read_out_data <= cmp_cap_data_n_03;
4'b0100: reg_controlled_read_out_data <= cmp_cap_data_n_04;
4'b0101: reg_controlled_read_out_data <= cmp_cap_data_n_05;
4'b0110: reg_controlled_read_out_data <= cmp_cap_data_n_06;
4'b0111: reg_controlled_read_out_data <= cmp_cap_data_n_07;
4'b1000: reg_controlled_read_out_data <= cmp_cap_data_n_08;
4'b1001: reg_controlled_read_out_data <= cmp_cap_data_n_09;
4'b1010: reg_controlled_read_out_data <= cmp_cap_data_n_10;
4'b1011: reg_controlled_read_out_data <= cmp_cap_data_n_11;
4'b1100: reg_controlled_read_out_data <= cmp_cap_data_n_12;
endcase

assign controlled_read_out_data =
    reg_controlled_read_out_data;
endmodule
Appendix B

SPICE netlist for 10-bit SAR ADC using variable clock period method

** Generated for: hspiceD
** Generated on: Jul 29 10:22:20 2014
** Design library name: sar
** Design cell name:
    sar_logic_test_sampling_final_DBOUT_hspice
** Design view name: schematic

.TEMP 25.0
.OPTION
+    ARTIST=2
+    INGOLD=2
+    PARHIER=LOCAL
+    PSF=2
.LIB "MODEL_LIBRARY$" $mode$

** Library name: sar
** Cell name: inv4
** View name: schematic
.subckt inv4 in out vddd vssd

mmm0 out in vssd vssd nch m=1 w=540e−9 l=129e−9 ad=162e−15 as =162e−15 pd=1.68e−6 \textbf{ps}=1.68e−6 nrd=316.667e−3 nrs=316.667e−3 sa=300e−9 sb=300e−9

mmm1 out in vddd vddd pch m=1 w=1.26e−6 l=129e−9 ad=378e−15 as=378e−15 pd=3.12e−6 \textbf{ps}=3.12e−6 nrd=135.714e−3 nrs =135.714e−3 sa=300e−9 sb=300e−9
.ends inv4

** End of subcircuit definition.

** Library name: sar
** Cell name: inv
** View name: schematic
.subckt inv in out vddd vssd

mmm0 out in vssd vssd nch m=1 w=135e−9 l=129e−9 ad=80.595e−15 as=80.595e−15 pd=1.194e−6 \textbf{ps}=1.194e−6 nrd=1.46667 nrs =1.46667 sa=327e−9 sb=327e−9

mmm1 out in vddd vddd pch m=1 w=315e−9 l=129e−9 ad=94.5e−15 as=94.5e−15 pd=1.23e−6 \textbf{ps}=1.23e−6 nrd=542.857e−3 nrs =542.857e−3 sa=300e−9 sb=300e−9
.ends inv

** End of subcircuit definition.
** Library name: sar
** Cell name: inv64
** View name: schematic

.subckt inv64 in out vddd vssd

.mmm0 out in vssd vssd nch m=8 w=1.08e−6 l=129e−9 ad=324e−15
    as=324e−15 pd=2.76e−6 ps=2.76e−6 nrd=158.333e−3 nrs
    =158.333e−3 sa=300e−9 sb=300e−9

.mml1 out in vddd vddd pch m=8 w=2.52e−6 l=129e−9 ad=756e−15
    as=756e−15 pd=5.64e−6 ps=5.64e−6 nrd=67.8571e−3 nrs
    =67.8571e−3 sa=300e−9 sb=300e−9

.ends inv64
** End of subcircuit definition.

** Library name: sar
** Cell name: inv8
** View name: schematic

.subckt inv8 in out vddd vssd

.mmm0 out in vssd vssd nch m=2 w=540e−9 l=129e−9 ad=162e−15 as
    =162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs=316.667e
    −3 sa=300e−9 sb=300e−9

.mml1 out in vddd vddd pch m=2 w=1.26e−6 l=129e−9 ad=207.9e−15
    as=378e−15 pd=1.59e−6 ps=3.12e−6 nrd=135.714e−3 nrs
    =135.714e−3 sa=443.981e−9 sb=443.981e−9

.ends inv8
** End of subcircuit definition.

78
** Library name: sar
** Cell name: inv16
** View name: schematic
.subckt inv16 in out vddd vssd

```
mmm0 out in vssd vssd nch m=4 w=540e-9 l=129e-9 ad=89.1e-15
  as=125.55e-15 pd=870e-9 ps=1.275e-6 nrd=316.667e-3 nrs
  =316.667e-3 sa=697.253e-9 sb=697.253e-9
mmml out in vddd vddd pch m=4 w=1.26e-6 l=129e-9 ad=207.9e-15
  as=292.95e-15 pd=1.59e-6 ps=2.355e-6 nrd=135.714e-3 nrs
  =135.714e-3 sa=697.253e-9 sb=697.253e-9
.ends inv16
** End of subcircuit definition.
```

** Library name: sar
** Cell name: inv32
** View name: schematic
.subckt inv32 in out vddd vssd

```
mmm0 out in vssd vssd nch m=4 w=1.08e-6 l=129e-9 ad=324e-15
  as=324e-15 pd=2.76e-6 ps=2.76e-6 nrd=158.333e-3 nrs
  =158.333e-3 sa=300e-9 sb=300e-9
mmml out in vddd vddd pch m=4 w=2.52e-6 l=129e-9 ad=756e-15
  as=756e-15 pd=5.64e-6 ps=5.64e-6 nrd=67.8571e-3 nrs
  =67.8571e-3 sa=300e-9 sb=300e-9
.ends inv32
** End of subcircuit definition.
```

** Library name: sar
** Cell name: sar_logic_buffer_11ch
** View name: schematic
.subckt sar_logic_buffer_11ch in<0> in<10> in<1> in<2> in<3> in<4> in<5> in<6> in<7> in<8> out<9> out<6> out<7> out<8> out<9> vdd vss
xi80 in<6> net036 vdd vss inv4
xi25 in<7> net0170 vdd vss inv4
xi21 in<8> net0171 vdd vss inv4
xi16 net017 out<10> vdd vss inv64
xi79 net036 out<6> vdd vss inv8
xi28 net0170 out<7> vdd vss inv8
xi24 net0171 out<8> vdd vss inv8
xi19 in<9> net0174 vdd vss inv8
xi15 in<10> net017 vdd vss inv16
xi20 net0174 out<9> vdd vss inv32
.ends sar_logic_buffer_11ch
** End of subcircuit definition.

** Library name: sar
** Cell name: or2
** View name: schematic
.subckt or2 a b out vdd vss
mpm2 net26 a vdd vdd pch m=1 w=540e−9 l=129e−9 ad=162e−15 as =162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs=316.667e −3 sa=300e−9 sb=300e−9

80
mpm1 net013 b net26 vdd pch m=1 w=540e−9 l=129e−9 ad=162e−15
as=162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs
=316.667e−3 sa=300e−9 sb=300e−9

mmm4 net013 a vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15
as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
=1.46667 sa=327e−9 sb=327e−9

mmm3 net013 b vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15
as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
=1.46667 sa=327e−9 sb=327e−9

xi0 net013 out vdd vss inv
.ends or2
** End of subcircuit definition.

** Library name: sar
** Cell name: inv2
** View name: schematic
.subckt inv2 in out vddd vssd

mmm0 out in vssd vssd nch m=1 w=270e−9 l=129e−9 ad=81e−15 as
=81e−15 pd=1.14e−6 ps=1.14e−6 nrd=633.333e−3 nrs=633.333e
−3 sa=300e−9 sb=300e−9

mmm1 out in vddd vddd pch m=1 w=630e−9 l=129e−9 ad=189e−15 as
=189e−15 pd=1.86e−6 ps=1.86e−6 nrd=271.429e−3 nrs=271.429e
−3 sa=300e−9 sb=300e−9
.ends inv2
** End of subcircuit definition.

** Library name: sar
** Cell name: comparator_w_valid
** View name: schematic

```
.subckt comparator_w_valid clk valid vdd vin vip _net0 _net1 vss

mmm9 vout1_n vin net028 vss nch m=2 w=1.8e-6 l=192e-9 ad
    =410.4e-15 as=766.8e-15 pd=2.256e-6 ps=4.452e-6 nrd=130e-3
    nrs=130e-3 sa=629.077e-9 sb=629.077e-9

mmm7 net028 vdd vss vss nch m=2 w=1.8e-6 l=192e-9 ad=766.8e
    -15 as=766.8e-15 pd=4.452e-6 ps=4.452e-6 nrd=130e-3 nrs
    =130e-3 sa=426e-9 sb=426e-9

mmm15 vout1_p vip net028 vss nch m=2 w=1.8e-6 l=192e-9 ad
    =410.4e-15 as=766.8e-15 pd=2.256e-6 ps=4.452e-6 nrd=130e-3
    nrs=130e-3 sa=629.077e-9 sb=629.077e-9

mmm14 vout1_n clkb vout1_p vss nch m=2 w=1.26e-6 l=129e-9 ad
    =536.76e-15 as=536.76e-15 pd=3.372e-6 ps=3.372e-6 nrd
    =185.714e-3 nrs=185.714e-3 sa=426e-9 sb=426e-9

mmm4 net022 net027 net32 vss nch m=1 w=900e-9 l=129e-9 ad
    =383.4e-15 as=383.4e-15 pd=2.652e-6 ps=2.652e-6 nrd=260e-3
    nrs=260e-3 sa=426e-9 sb=426e-9

mmm3 net027 net022 net33 vss nch m=1 w=900e-9 l=129e-9 ad
    =383.4e-15 as=383.4e-15 pd=2.652e-6 ps=2.652e-6 nrd=260e-3
    nrs=260e-3 sa=426e-9 sb=426e-9

mmm2 net32 vout1_n net23 vss nch m=1 w=2.7e-6 l=192e-9 ad
    =1.1502e-12 as=1.1502e-12 pd=6.252e-6 ps=6.252e-6 nrd
    =86.6667e-3 nrs=86.6667e-3 sa=426e-9 sb=426e-9

82
mm1 net23 clk vss vss nch m=2 w=1.8e-6 l=129e-9 ad=766.8e-15
as=766.8e-15 pd=4.452e-6 ps=4.452e-6 nrd=130e-3 nrs=130e-3
sa=426e-9 sb=426e-9

mm0 net33 vout1_p net23 vss nch m=1 w=2.7e-6 l=192e-9 ad=1.1502e-12
as=1.1502e-12 pd=6.252e-6 ps=6.252e-6 nrd=86.6667e-3
nrs=86.6667e-3 sa=426e-9 sb=426e-9

mpm6 vout1_n vout1_n vdd vdd pch m=2 w=2.7e-6 l=129e-9 ad=615.6e-15
as=615.6e-15 pd=3.156e-6 ps=6.252e-6 nrd=86.6667e-3
nrs=86.6667e-3 sa=612.422e-9 sb=612.422e-9

mpm7 vout1_p vout1_p vdd vdd pch m=2 w=2.7e-6 l=129e-9 ad=615.6e-15
as=615.6e-15 pd=3.156e-6 ps=6.252e-6 nrd=86.6667e-3
nrs=86.6667e-3 sa=612.422e-9 sb=612.422e-9

mpm3 net022 clk vdd vdd pch m=1 w=630e-9 l=129e-9 ad=460.08e-15
as=460.08e-15 pd=3.012e-6 ps=3.012e-6 nrd=216.667e-3
nrs=216.667e-3 sa=426e-9 sb=426e-9

mpm2 net027 net022 vdd vdd pch m=1 w=630e-9 l=129e-9 ad=268.38e-15
as=268.38e-15 pd=2.112e-6 ps=2.112e-6 nrd=371.429e-3
nrs=371.429e-3 sa=426e-9 sb=426e-9

mpm1 net027 clk vdd vdd pch m=1 w=630e-9 l=129e-9 ad=460.08e-15
as=460.08e-15 pd=3.012e-6 ps=3.012e-6 nrd=216.667e-3
nrs=216.667e-3 sa=426e-9 sb=426e-9

mpm0 net022 net027 vdd vdd pch m=1 w=630e-9 l=129e-9 ad=268.38e-15
as=268.38e-15 pd=2.112e-6 ps=2.112e-6 nrd=371.429e-3
nrs=371.429e-3 sa=426e-9 sb=426e-9

xi2 _net0 _net1 valid vdd vss or2
xi4 net027 _net1 vdd vss inv
xi3 net022 _net0 vdd vss inv

83
xi21 clk clkb vdd vss inv2
.ends comparator_w_valid

** End of subcircuit definition.

** Library name: sar
** Cell name: delay2_buff2
** View name: schematic
.subckt delay2_buff2 in out vdd vss
xi2 net08 net07 vdd vss inv
xi1 net11 net08 vdd vss inv
xi0 in net11 vdd vss inv
xi3 net07 out vdd vss inv2
.ends delay2_buff2
** End of subcircuit definition.

** Library name: sar
** Cell name: delay4
** View name: schematic
.subckt delay4 in out vdd vss
xi5 in net012 vdd vss inv
xi3 net11 net013 vdd vss inv
xi4 net013 out vdd vss inv
xi0 net012 net11 vdd vss inv
mm0 vss net11 vss vss nch m=16 w=900e-9 l=912e-9 ad=148.5e-15 as=270e-15 pd=1.23e-6 ps=2.4e-6 nrd=190e-3 nrs=190e-3
sa=643.97e-9 sb=643.97e-9
.ends delay4
** End of subcircuit definition.

** Library name: sar
** Cell name: or3
** View name: schematic

.subckt or3 a b c out vdd vss

mpm2 net26 a vdd vdd pch m=1 w=810e−9 l=129e−9 ad=243e−15 as
  =243e−15 pd=2.22e−6 ps=2.22e−6 nrd=211.111e−3 nrs=211.111e−3
  sa=300e−9 sb=300e−9

mpml net27 b net26 vdd pch m=1 w=810e−9 l=129e−9 ad=243e−15
  as=243e−15 pd=2.22e−6 ps=2.22e−6 nrd=211.111e−3 nrs
  =211.111e−3 sa=300e−9 sb=300e−9

mnm2 net012 c net27 vdd pch m=1 w=810e−9 l=129e−9 ad=243e−15
  as=243e−15 pd=2.22e−6 ps=2.22e−6 nrd=211.111e−3 nrs
  =211.111e−3 sa=300e−9 sb=300e−9

mnm4 net012 a vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15
  as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
  =1.46667 sa=327e−9 sb=327e−9

mnm1 net012 c vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15
  as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
  =1.46667 sa=327e−9 sb=327e−9

mnm3 net012 b vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15
  as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
  =1.46667 sa=327e−9 sb=327e−9

xi0 net012 out vdd vss inv

.ends or3

** End of subcircuit definition.
** Library name: sar
** Cell name: nand2
** View name: schematic

.subckt nand2 a b out vdd vss

mpm0 out a vdd vdd pch m=1 w=270e-9 l=129e-9 ad=81e-15 as=81e-15 pd=1.14e-6 ps=1.14e-6 nrd=633.333e-3 nrs=633.333e-3 sa=300e-9 sb=300e-9

mnm2 out b vdd vdd pch m=1 w=270e-9 l=129e-9 ad=81e-15 as=81e-15 pd=1.14e-6 ps=1.14e-6 nrd=633.333e-3 nrs=633.333e-3 sa=300e-9 sb=300e-9

mml1 out b net13 vss nch m=1 w=270e-9 l=129e-9 ad=81e-15 as=81e-15 pd=1.14e-6 ps=1.14e-6 nrd=633.333e-3 nrs=633.333e-3 sa=300e-9 sb=300e-9

mml0 net13 a vss vss nch m=1 w=270e-9 l=129e-9 ad=81e-15 as=81e-15 pd=1.14e-6 ps=1.14e-6 nrd=633.333e-3 nrs=633.333e-3 sa=300e-9 sb=300e-9

.ends nand2

** End of subcircuit definition.

** Library name: sar
** Cell name: nor2
** View name: schematic

.subckt nor2 a b out vdd vss

mpm2 net26 a vdd vdd pch m=1 w=540e-9 l=129e-9 ad=162e-15 as=162e-15 pd=1.68e-6 ps=1.68e-6 nrd=316.667e-3 nrs=316.667e-3 sa=300e-9 sb=300e-9

86
mpm1 out b net26 vdd pch m=1 w=540e−9 l=129e−9 ad=162e−15 as
   =162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs=316.667e−3
   sa=300e−9 sb=300e−9

mm4 out a vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15 as
   =80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9

mm3 out b vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15 as
   =80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9
.ends nor2
** End of subcircuit definition.

** Library name: sar
** Cell name: dff_ver3_Q
** View name: schematic
.subckt dff_ver3_Q d q reset vdd vss clk
xi17 net038 rb net012 vdd vss nand2
xi29 reset rb vdd vss inv
xi24 clk clkbr vdd vss inv
mm6 net012 clk net036 vss nch m=1 w=360e−9 l=129e−9 ad=108e−15 as=108e−15 pd=1.32e−6 ps=1.32e−6 nrd=475e−3 nrs=475e−3
   sa=300e−9 sb=300e−9

mm5 net026 clkbr vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15 as=80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9
.subckt dff_ver3_Q

** End of subcircuit definition.

** Library name: sar

** Cell name: dff_ver3_Q_buff

** View name: schematic

.ends dff_ver3_Q

xi20 net036 reset q vdd vss nor2

88
xi29 reset rb vdd vss inv
xi31 net025 net020 vdd vss inv
xi24 clk clkbr vdd vss inv

mnm6 net012 clk net036 vss nch m=1 w=360e-9 l=129e-9 ad=108e-15 as=108e-15 pd=1.32e-6 ps=1.32e-6 nrd=475e-3 nrs=475e-3 sa=300e-9 sb=300e-9

mnm5 net026 clkbr vss vss nch m=1 w=135e-9 l=129e-9 ad=80.595e-15 as=80.595e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.46667 nrs=1.46667 sa=327e-9 sb=327e-9

mnm4 net036 net025 net026 vss nch m=1 w=135e-9 l=129e-9 ad=80.595e-15 as=80.595e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.46667 nrs=1.46667 sa=327e-9 sb=327e-9

mnm3 d clkbr net038 vss nch m=1 w=360e-9 l=129e-9 ad=108e-15 as=108e-15 pd=1.32e-6 ps=1.32e-6 nrd=475e-3 nrs=475e-3 sa=300e-9 sb=300e-9

mnm2 net032 clk vss vss nch m=1 w=135e-9 l=129e-9 ad=80.595e-15 as=80.595e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.46667 nrs=1.46667 sa=327e-9 sb=327e-9

mnm0 net038 net012 net032 vss nch m=1 w=135e-9 l=129e-9 ad=80.595e-15 as=80.595e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.46667 nrs=1.46667 sa=327e-9 sb=327e-9

mpm1 net036 net025 vdd vdd pch m=1 w=162e-9 l=129e-9 ad=82.134e-15 as=82.134e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.22222 nrs=1.22222 sa=327e-9 sb=327e-9

mpm0 net038 net012 vdd vdd pch m=1 w=162e-9 l=129e-9 ad=82.134e-15 as=82.134e-15 pd=1.194e-6 ps=1.194e-6 nrd=1.22222 nrs=1.22222 sa=327e-9 sb=327e-9

89
xi32 net020 q vdd vss inv4
xi20 net036 reset net025 vdd vss nor2
.ends dff_ver3_Q_buff
** End of subcircuit definition.

** Library name: sar
** Cell name: nor2_dff
** View name: schematic
.subckt nor2_dff a b out vdd vss
mpm2 net26 a vdd vdd pch m=1 w=540e−9 l=129e−9 ad=162e−15 as
   =162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs=316.667e−3 sa=300e−9 sb=300e−9
mpm1 out b net26 vdd pch m=1 w=540e−9 l=129e−9 ad=162e−15 as
   =162e−15 pd=1.68e−6 ps=1.68e−6 nrd=316.667e−3 nrs=316.667e−3 sa=300e−9 sb=300e−9
mnm4 out a vss vss nch m=1 w=135e−9 l=129e−9 ad=80.595e−15 as
   =80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9
mnm3 out b vss vss nch m=1 w=207e−9 l=129e−9 ad=84.699e−15 as
   =84.699e−15 pd=1.194e−6 ps=1.194e−6 nrd=956.522e−3 nrs
   =956.522e−3 sa=327e−9 sb=327e−9
.ends nor2_dff
** End of subcircuit definition.

** Library name: sar
** Cell name: nor2_d
** View name: schematic
.subckt nor2_d a b out vdd vss
mpm2 net26 a vdd vdd pch m=1 w=270e−9 l=192e−9 ad=81e−15 as
   =81e−15 pd=1.14e−6 ps=1.14e−6 nrd=633.333e−3 nrs=633.333e
   −3 sa=300e−9 sb=300e−9
mpml out b net26 vdd pch m=1 w=270e−9 l=192e−9 ad=81e−15 as
   =81e−15 pd=1.14e−6 ps=1.14e−6 nrd=633.333e−3 nrs=633.333e
   −3 sa=300e−9 sb=300e−9
mmm4 out a vss vss nch m=1 w=135e−9 l=192e−9 ad=80.595e−15 as
   =80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9
mmm3 out b vss vss nch m=1 w=135e−9 l=192e−9 ad=80.595e−15 as
   =80.595e−15 pd=1.194e−6 ps=1.194e−6 nrd=1.46667 nrs
   =1.46667 sa=327e−9 sb=327e−9
.ends nor2_d

** End of subcircuit definition.

** Library name: sar
** Cell name: tmg_2input
** View name: schematic
.subckt tmg_2input in out sw sw_b vdd vssd
mmm0 out sw in vssd nch m=1 w=360e−9 l=129e−9 ad=108e−15 as
   =108e−15 pd=1.32e−6 ps=1.32e−6 nrd=475e−3 nrs=475e−3 sa
   =300e−9 sb=300e−9
mpm0 out sw_b in vdd pch m=1 w=720e−9 l=129e−9 ad=216e−15 as
   =216e−15 pd=2.04e−6 ps=2.04e−6 nrd=237.5e−3 nrs=237.5e−3
   sa=300e−9 sb=300e−9
.ends tmg_2input
** End of subcircuit definition.

** Library name: sar
** Cell name: dff_ver4_Q_RS
** View name: schematic
.subckt dff_ver4_Q_RS d q reset set vdd vss clk
  xi14 set net1 net2 vdd vss nor2_dff
  xi20 reset net037 net044 vdd vss nor2_dff
  xi8 net044 net5 vdd vss inv
  xi9 net5 q vdd vss inv4
  xi27 set net044 net045 vdd vss nor2_d
  xi22 reset net2 net029 vdd vss nor2_d
  xi11 clk clkbr vdd vss inv2
  xi18 net037 net045 clkbr clk vdd vss tmg_2input
  xi19 net2 net037 clk clkbr vdd vss tmg_2input
  xi17 net1 net029 clk clkbr vdd vss tmg_2input
  xi21 d net1 clkbr clk vdd vss tmg_2input
.ends dff_ver4_Q_RS
** End of subcircuit definition.

** Library name: sar
** Cell name: sar_logic_11bit_ver71_test
** View name: schematic
.subckt sar_logic_11bit_ver71_test d<0> d<10> d<1> d<2> d<3>
  d<4> d<5> d<6> d<7> d<8> d<9> clk clkc comp sample_short
    valid vdd vss
  xi384 net0168 net093 vdd vss delay2_buff2
xi383 net0128 net0169 vdd vss delay2_buff2
xi382 net0169 net0168 vdd vss delay2_buff2
xi379 net0127 net0172 vdd vss delay2_buff2
xi378 net0171 net090 vdd vss delay2_buff2
xi377 net0172 net0171 vdd vss delay2_buff2
xi362 net0123 net0184 vdd vss delay2_buff2
xi361 net0183 net092 vdd vss delay2_buff2
xi366 net0180 net089 vdd vss delay2_buff2
xi351 net0193 net0192 vdd vss delay2_buff2
xi343 net0198 net086 vdd vss delay2_buff2
xi367 net0181 net0180 vdd vss delay2_buff2
xi345 net0196 net0195 vdd vss delay2_buff2
xi369 net0177 net0138 vdd vss delay2_buff2
xi370 net0125 net0178 vdd vss delay2_buff2
xi371 net0178 net0177 vdd vss delay2_buff2
xi359 net0121 net0187 vdd vss delay2_buff2
xi358 net0187 net0186 vdd vss delay2_buff2
xi373 net0175 net0174 vdd vss delay2_buff2
xi388 net086 net0200 vdd vss delay2_buff2
xi374 net0174 net096 vdd vss delay2_buff2
xi389 net087 net0197 vdd vss delay2_buff2
xi346 net0110 net0196 vdd vss delay2_buff2
xi347 net0195 net087 vdd vss delay2_buff2
xi363 net0184 net0183 vdd vss delay2_buff2
xi354 net0119 net0190 vdd vss delay2_buff2
xi394 net089 net0182 vdd vss delay2_buff2
xi355 net0190 net0189 vdd vss delay2_buff2
xi392 net0137 net0188 vdd vss delay2_buff2
xi349 net0192 net094 vdd vss delay2_buff2
xi350 net095 net0193 vdd vss delay2_buff2
xi397 net090 net0173 vdd vss delay2_buff2
xi391 net091 net0191 vdd vss delay2_buff2
xi365 net0124 net0181 vdd vss delay2_buff2
xi353 net0189 net091 vdd vss delay2_buff2
xi393 net092 net0185 vdd vss delay2_buff2
xi341 net0112 net0199 vdd vss delay2_buff2
xi375 net0205 net0175 vdd vss delay2_buff2
xi357 net0186 net0137 vdd vss delay2_buff2
xi398 net093 net0170 vdd vss delay2_buff2
xi342 net0199 net0198 vdd vss delay2_buff2
xi396 net096 net0176 vdd vss delay2_buff2
xi390 net094 net0194 vdd vss delay2_buff2
xi395 net0138 net0179 vdd vss delay2_buff2
xi338 net0126 net052 vdd vss inv16
xi340 net0132 net068 vdd vss inv16
xi9 net088 net166 vdd vss delay4
xi339 net0114 net0132 vdd vss inv4
xi337 net0153 net0126 vdd vss inv4
xi10 clk net088 vdd vss inv4
xi399 net052 net0114 vdd vss inv
xi13 net178 valid net052 clkc vdd vss or3
xi12 vdd net178 net045 vdd vss net052 dff_ver3_Q
xi336 net0130 net0153 net045 vdd vss valid dff_ver3_Q
xi400 net068 sample_short vdd vss inv32

94
xi198 net052 net0110 net0122 vdd vss or2
xi108 net0128 net0130 net052 vdd vss valid dff_ver3_Q_buff
xi333 vdd net045 valid vdd vss net088 dff_ver3_Q_buff
xi37 net0127 net0128 net052 vdd vss valid dff_ver3_Q_buff
xi36 net0205 net0127 net052 vdd vss valid dff_ver3_Q_buff
xi35 net0125 net0205 net052 vdd vss valid dff_ver3_Q_buff
xi8 net0124 net0125 net052 vdd vss valid dff_ver3_Q_buff
xi7 net0123 net0124 net052 vdd vss valid dff_ver3_Q_buff
xi6 net0121 net0123 net052 vdd vss valid dff_ver3_Q_buff
xi5 net0119 net0121 net052 vdd vss valid dff_ver3_Q_buff
xi4 net095 net0119 net052 vdd vss valid dff_ver3_Q_buff
xi3 net0110 net095 net052 vdd vss valid dff_ver3_Q_buff
xi2 net0112 net0110 net052 vdd vss valid dff_ver3_Q_buff
xi1 vdd net0112 net0122 vdd vss net166 dff_ver3_Q_buff
xi387 vss d<0> sample_short net0130 vdd vss vss dff_ver4_Q_RS
xi385 vss net0161 sample_short net0170 vdd vss vss
dff_ver4_Q_RS

xi380 comp d<1> sample_short net0173 vdd vss net0161
dff_ver4_Q_RS
xi376 comp d<2> sample_short net0176 vdd vss d<1>
dff_ver4_Q_RS
xi364 comp d<5> sample_short net0185 vdd vss d<4>
dff_ver4_Q_RS
xi352 comp d<8> sample_short net0194 vdd vss d<7>
dff_ver4_Q_RS
xi368 comp d<4> sample_short net0182 vdd vss d<3>
dff_ver4_Q_RS

95
xi360 comp d<6> sample_short net0188 vdd vss d<5> dff_ver4_Q_RS
xi372 comp d<3> sample_short net0179 vdd vss d<2> dff_ver4_Q_RS
xi344 comp d<10> sample_short net0200 vdd vss d<9> dff_ver4_Q_RS
xi348 comp d<9> sample_short net0197 vdd vss d<8> dff_ver4_Q_RS
xi356 comp d<7> sample_short net0191 vdd vss d<6> dff_ver4_Q_RS
.ends sar_logic_11bit_ver71_test
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D10
** View name: schematic
.subckt switch_2input_INV_D10 a b d gate vdd vss

mmml d net06 a vdd pch m=32 w=2.25e−6 l=129e−9 ad=513e−15 as
   =540.844e−15 pd=2.706e−6 ps=2.87138e−6 nrd=104e−3 nrs=104e−3 sa=4.30068e−6 sb=4.30068e−6

mmm0 d net06 b vss nch m=100 w=360e−9 l=129e−9 ad=513e−15 as
   =568.688e−15 pd=2.706e−6 ps=3.03675e−6 nrd=104e−3 nrs=104e−3 sa=2.51979e−6 sb=2.51979e−6

xi2 gate net06 vdd vss inv64
.ends switch_2input_INV_D10
** End of subcircuit definition.
** Library name: sar
** Cell name: inv3
** View name: schematic

.subckt inv3 in out vddd vssd

mnm0 out in vssd vssd nch m=1 w=405e-9 l=129e-9 ad=121.5e-15
as=121.5e-15 pd=1.41e-6 ps=1.41e-6 nrd=422.222e-3 nrs
=422.222e-3 sa=300e-9 sb=300e-9

mnm1 out in vddd vddd pch m=1 w=900e-9 l=129e-9 ad=270e-15 as
=270e-15 pd=2.4e-6 ps=2.4e-6 nrd=190e-3 nrs=190e-3 sa=300e
-9 sb=300e-9

.ends inv3

** End of subcircuit definition.

** Library name: sar
** Cell name: buff16_1
** View name: schematic

.subckt buff16_1 in out vdd vss

xi0 net08 net11 vdd vss inv3
xi1 net11 net07 vdd vss inv8
xi5 net07 out vdd vss inv16
xi4 in net08 vdd vss inv

.ends buff16_1

** End of subcircuit definition.

** Library name: sar
** Cell name: Resister_10bit
** View name: schematic
.subckt Resister_10bit clk i<0> i<1> i<2> i<3> i<4> i<5> i<6> i<7> i<8> i<9> o<0> o<1> o<2> o<3> o<4> o<5> o<6> o<7> o<8> o<9> vdd vss
xi9 i<0> o<0> net22 vdd vss clk dff_ver3_Q_buff
xi8 i<1> o<1> net23 vdd vss clk dff_ver3_Q_buff
xi7 i<2> o<2> net24 vdd vss clk dff_ver3_Q_buff
xi6 i<3> o<3> net25 vdd vss clk dff_ver3_Q_buff
xi5 i<4> o<4> net26 vdd vss clk dff_ver3_Q_buff
xi4 i<5> o<5> net27 vdd vss clk dff_ver3_Q_buff
xi3 i<6> o<6> net28 vdd vss clk dff_ver3_Q_buff
xi2 i<7> o<7> net29 vdd vss clk dff_ver3_Q_buff
xi10 i<8> o<8> net30 vdd vss clk dff_ver3_Q_buff
xi0 i<9> o<9> net21 vdd vss clk dff_ver3_Q_buff
.ends Resister_10bit

** End of subcircuit definition.

** Library name: sar
** Cell name: tmg_16x_1_2
** View name: schematic

.subckt tmg_16x_1_2 in out sw vddd vssd
mnm0 out sw in vssd nch m=8 w=900e-9 l=129e-9 ad=148.5e-15 as=178.875e-15 pd=1.23e-6 ps=1.5225e-6 nrd=190e-3 nrs=190e-3 sa=1.14052e-6 sb=1.14052e-6
mpm0 out net9 in vddd pch m=7 w=1.35e-6 l=129e-9 ad=248.786e-15 as=248.786e-15 pd=1.91143e-6 ps=1.91143e-6 nrd=126.667e-3 nrs=126.667e-3 sa=1.03477e-6 sb=1.03477e-6
xi0 sw net9 vddd vssd inv8

98
.ends tmg_16x_1_2
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D9
** View name: schematic
.subckt switch_2input_INV_D9 a b d gate vdd vss

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>18</td>
</tr>
<tr>
<td>w</td>
<td>2.25e-6</td>
</tr>
<tr>
<td>l</td>
<td>129e-9</td>
</tr>
<tr>
<td>ad</td>
<td>513e-15</td>
</tr>
<tr>
<td>pd</td>
<td>2.706e-6</td>
</tr>
<tr>
<td>ps</td>
<td>3e-6</td>
</tr>
<tr>
<td>nrd</td>
<td>104e-3</td>
</tr>
<tr>
<td>nrs</td>
<td>104e-3</td>
</tr>
<tr>
<td>sa</td>
<td>2.75508e-6</td>
</tr>
<tr>
<td>sb</td>
<td>2.75508e-6</td>
</tr>
</tbody>
</table>

.mmm1 d net06 a vdd pch m=4 w=2.25e-6 l=129e-9 ad=513e-15 as
   =624.375e-15 pd=2.706e-6 ps=3.3675e-6 nrd=104e-3 nrs=104e-3 sa
   =1.51357e-6 sb=1.51357e-6

.xi2 gate net06 vdd vss inv32
.ends switch_2input_INV_D9
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D8
** View name: schematic
.subckt switch_2input_INV_D8 a b d gate vdd vss

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>8</td>
</tr>
<tr>
<td>w</td>
<td>2.25e-6</td>
</tr>
<tr>
<td>l</td>
<td>129e-9</td>
</tr>
<tr>
<td>ad</td>
<td>513e-15</td>
</tr>
<tr>
<td>pd</td>
<td>2.706e-6</td>
</tr>
<tr>
<td>ps</td>
<td>3e-6</td>
</tr>
<tr>
<td>nrd</td>
<td>104e-3</td>
</tr>
<tr>
<td>nrs</td>
<td>104e-3</td>
</tr>
<tr>
<td>sa</td>
<td>940.292e-9</td>
</tr>
<tr>
<td>sb</td>
<td>940.292e-9</td>
</tr>
</tbody>
</table>

99
xi2 gate net06 vdd vss inv16
.ends switch_2input_INV_D8
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D7
** View name: schematic
.subckt switch_2input_INV_D7 a b d gate vdd vss
  mnm1 d net06 a vdd pch m=4 w=2.25e−6 l=129e−9 ad=513e−15 as
  =735.75e−15 pd=2.706e−6 ps=4.029e−6 nrd=104e−3 nrs=104e−3
  sa=940.292e−9 sb=940.292e−9
  mnm0 d net06 b vss nch m=2 w=2.25e−6 l=129e−9 ad=513e−15 as
  =958.5e−15 pd=2.706e−6 ps=5.352e−6 nrd=104e−3 nrs=104e−3
  sa=612.422e−9 sb=612.422e−9
xi2 gate net06 vdd vss inv16
.ends switch_2input_INV_D7
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D6
** View name: schematic
.subckt switch_2input_INV_D6 a b d gate vdd vss
  mnm1 d net06 a vdd pch m=2 w=2.25e−6 l=129e−9 ad=513e−15 as
  =958.5e−15 pd=2.706e−6 ps=5.352e−6 nrd=104e−3 nrs=104e−3
  sa=612.422e−9 sb=612.422e−9
mnm0 d net06 b vss nch m=1 w=2.25e-6 l=129e-9 ad=958.5e-15 as
  =958.5e-15 pd=5.352e-6 ps=5.352e-6 nrd=104e-3 nrs=104e-3
  sa=426e-9 sb=426e-9
xi2 gate net06 vdd vss inv8
.ends switch_2input_INV_D6
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D5
** View name: schematic
.subckt switch_2input_INV_D5 a b d gate vdd vss
mnm1 d net06 a vdd pch m=1 w=2.25e-6 l=129e-9 ad=958.5e-15 as
  =958.5e-15 pd=5.352e-6 ps=5.352e-6 nrd=104e-3 nrs=104e-3
  sa=426e-9 sb=426e-9
mnm0 d net06 b vss nch m=1 w=1.125e-6 l=129e-9 ad=479.25e-15
  as=479.25e-15 pd=3.102e-6 ps=3.102e-6 nrd=208e-3 nrs=208e
  -3 sa=426e-9 sb=426e-9
xi2 gate net06 vdd vss inv4
.ends switch_2input_INV_D5
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D4
** View name: schematic
.subckt switch_2input_INV_D4 a b d gate vdd vss
** Library name: sar  
** Cell name: switch_2input_INV_D3  
** View name: schematic

.subckt switch_2input_INV_D3 a b d gate vdd vss

mnm1 d net06 a vdd pch m=2 w=720e-9 l=129e-9 ad=164.16e-15 as =306.72e-15 pd=1.176e-6 ps=2.292e-6 nrd=325e-3 nrs=325e-3 sa=612.422e-9 sb=612.422e-9

mnm0 d net06 b vss nch m=2 w=360e-9 l=129e-9 ad=82.08e-15 as =153.36e-15 pd=816e-9 ps=1.572e-6 nrd=650e-3 nrs=650e-3 sa =612.422e-9 sb=612.422e-9

xi2 gate net06 vdd vss inv4 
.ends switch_2input_INV_D4

** End of subcircuit definition.

** Library name: sar  
** Cell name: switch_2input_INV_D2  
** View name: schematic
.subckt switch_2input_INV_D2 a b d gate vdd vss
mm1 d net06 a vdd pch m=1 w=360e-9 l=129e-9 ad=153.36e-15 as
   =153.36e-15 pd=1.572e-6 ps=1.572e-6 nrd=650e-3 nrs=650e-3
   sa=426e-9 sb=426e-9
mm0 d net06 b vss nch m=1 w=270e-9 l=129e-9 ad=115.02e-15 as
   =115.02e-15 pd=1.392e-6 ps=1.392e-6 nrd=866.667e-3 nrs
   =866.667e-3 sa=426e-9 sb=426e-9
xi2 gate net06 vdd vss inv2
.ends switch_2input_INV_D2
** End of subcircuit definition.

** Library name: sar
** Cell name: switch_2input_INV_D1
** View name: schematic
.subckt switch_2input_INV_D1 a b d gate vdd vss
mm1 d net06 a vdd pch m=1 w=360e-9 l=129e-9 ad=153.36e-15 as
   =153.36e-15 pd=1.572e-6 ps=1.572e-6 nrd=650e-3 nrs=650e-3
   sa=426e-9 sb=426e-9
mm0 d net06 b vss nch m=1 w=270e-9 l=129e-9 ad=115.02e-15 as
   =115.02e-15 pd=1.392e-6 ps=1.392e-6 nrd=866.667e-3 nrs
   =866.667e-3 sa=426e-9 sb=426e-9
xi2 gate net06 vdd vss inv2
.ends switch_2input_INV_D1
** End of subcircuit definition.

** Library name: sar
** Cell name: sar_logic_test_sampling_final_DBOUT
** View name: schematic

**.subckt sar_logic_test_sampling_final_DBOUT clk data<0> data
  <1> data<2> data<3> data<4> data<5> data<6> data<7> data
  <8> data<9> vdd vdda vin vss v_04 v_11 vcomp_sample

  vin_sample

  xi88 clk clk_d_b vdd vss inv4
  xi21 net088 comp_clk vdda vss inv4
  xi461 net0128 net0101 vdda vss inv
  xi436 clkc net0128 vdd vss inv
  xi437 net043 net051 vdda vss inv
  xi462 net0101 net088 vdda vss inv
  xi434 net026 net0115 d<1> d<2> d<3> d<4> d<5> net0119 net0118
      net0117 net0116 d<10> d<6> d<7> d<8> d<9> vdd vss

  sar_logic_buffer_11ch

  xi340 comp_clk net043 vdda vcomp_sample vin_sample v_code_pre

  net053 vss comparator_w_valid

  xi41 net026 net0115 d<1> d<2> d<3> d<4> d<5> net0119 net0118
      net0117 net0116 clk_d clkc net055 sample valid vdd vss

  sar_logic_11bit_ver71_test

  xi89 clk_d_b clk_d vdd vss inv8
  xi475 v_11 v_04 net069 d<10> vdd vss switch_2input_INV_D10
  xi23 net051 valid vdda vss buff16_1
  xi438 v_code_pre net055 vdd vss buff16_1
  xi419 net026 d<1> d<2> d<3> d<4> d<5> d<6> d<7> d<8> d<9> d
      <10> data<0> data<1> data<2> data<3> data<4> data<5> data
      <6> data<7> data<8> data<9> vdd vss Resister_10bit
  xc17 vin_sample vss mimcap_1p0 l=1.8369e−6 w=3.9537e−6 m=96
xc161 vcomp_sample net069 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=256
xc162 vcomp_sample net084 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=128
xc163 vcomp_sample net085 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=64
xc164 vcomp_sample net086 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=32
xc165 vcomp_sample net087 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=16
xc166 vcomp_sample net091 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=8
xc167 vcomp_sample net089 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=4
xc168 vcomp_sample net079 mimcap_p0  l=1.8369e-6 w=3.9537e-6
   m=2
xc169 vcomp_sample net080 mimcap_p0  l=1.8369e-6 w=1.8369e-6
   m=1
xc170 vcomp_sample net081 mimcap_p0  l=1.8369e-6 w=1.8369e-6
   m=1
xc171 vcomp_sample v_04 mimcap_p0  l=1.8369e-6 w=1.8369e-6 m
   =1
xi299 vin_sample vin_sample vdd vss tmg.16x_1.2
xi476 v_11 v_04 net084 d<9> vdd vss switch_2input_INV_D9
xi477 v_11 v_04 net085 d<8> vdd vss switch_2input_INV_D8
xi478 v_11 v_04 net086 d<7> vdd vss switch_2input_INV_D7
xi479 v_11 v_04 net087 d<6> vdd vss switch_2input_INV_D6
xi480 v_11 v_04 net091 d<5> vdd vss switch_2input_INV_D5
xi481 v_11 v_04 net089 d<4> vdd vss switch_2input_INV_D4
xi482 v_11 v_04 net079 d<3> vdd vss switch_2input_INV_D3
xi483 v_11 v_04 net080 d<2> vdd vss switch_2input_INV_D2
xi484 v_11 v_04 net081 d<1> vdd vss switch_2input_INV_D1

mmn7 vcomp_sample sample v_04 vss nch m=20 w=3.15e−6 l=129e−9
     ad=718.2e−15 as=780.57e−15 pd=3.606e−6 ps=3.9606e−6 nrd
     =74.2857e−3 nrs=74.2857e−3 sa=2.98584e−6 sb=2.98584e−6
.ends sar_logic_test_sampling_final_DBOUT

** End of subcircuit definition.

** Library name: sar
** Cell name: sar_logic_test_sampling_final_DBOUT_hspice
** View name: schematic

xi0 clk data<0> data<1> data<2> data<3> data<4> data<5> data
     <6> data<7> data<8> data<9> vdd vdda vin vss v_04 v_11
     vcomp_sample vin_sample
     sar_logic_test_sampling_final_DBOUT

.END
Bibliography


110


