A DOT PRODUCT KERNEL USING RAPIDLY SWITCHED ANALOG CIRCUIT

BY

IHAB NAHLUS

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Professor Naresh R. Shanbhag
In a world driven by technology and hand-held devices, there is ubiquitous demand for high-performance, low-energy processing engines. In this thesis, we present rapidly switched analog circuit (RSAC), a new circuit architecture, to implement an energy-efficient mixed-signal dot product (DP) kernel for machine learning and signal processing applications. RSAC operates by fast switching the analog inputs to the output via variable width digital pulses. A description of the different components of RSAC, along with a detailed accuracy and energy consumption analysis is presented. We show two RSAC designs that span the different design options and technology nodes. Simulations for the first design in a 130 nm process show energy savings of $19 \times$ to $32 \times$ compared to a digital implementation for signal-to-quantization-noise ratios (SQNRs) of 30 dB to 24 dB, respectively. Simulations for the second design in a 28 nm FDSOI process show energy savings of $15.7 \times$, $4 \times$, $2.1 \times$ compared to a digital implementation running at the same sampling frequency for SQNRs of 8 dB, 14 dB and 20 dB, respectively. Finally, we present the design of an emotion recognition system composed solely of SAC-based dot-products. Based on the behavioral and energy models developed in this thesis, we obtain energy savings of 45% and 49% compared to a digital implementation for average probabilities of error of 0.23 and 0.07, running at frequencies of 1.87 MHz and 1.7 MHz, respectively.
To my family, for their love and support
ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Naresh R. Shanbhag for his continuous guidance, patience, enthusiasm and invaluable comments throughout this work. I also thank our collaborators Dr. Eric Kim and Prof. David Blaauw for their comments and feedback for this work. I want to also thank my labmates: Pourya, Tianqi, Yingyan, Sai, Mingu, Sujan and Ameya. They eased me into the research group when I first arrived and made my research environment very friendly.

I want to thank my parents: Nazih and Ghada, my brothers: Saad and Ramzi, and sisters: Farah, Juliana and Dunia, for their love and support during my studies. They are my backbone and I feel blessed for having them by my side. Life would not be complete without having friends to share the journey with. I want to thank Mohammad, Peter, Marc, Laurence and Farah for the wonderful company during my stay in Illinois. It hasn’t been an easy ride but they definitely made it enjoyable.

Finally, I want to thank Brooke Newell and my friend Laurence for helping me in making this thesis easy to read, grammatically correct, and typo-free.
TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ........................................... 1
  1.1 Motivation ..................................................... 1
  1.2 Background ................................................... 2
    1.2.1 Digital approach ........................................ 2
    1.2.2 Analog approach ......................................... 5
    1.2.3 Stochastic computing approach ......................... 6

CHAPTER 2 RAPIDLY SWITCHED ANALOG CIRCUIT (RSAC) PRINCIPLES ........................................... 10
  2.1 RSAC architecture ............................................ 10
  2.2 Principle of operation ....................................... 12
  2.3 RSAC circuit implementations ............................... 16
  2.4 Select generation ............................................ 20
    2.4.1 Constant coefficients .................................. 21
    2.4.2 Programmable coefficients .............................. 21
  2.5 Behavioral model ............................................. 24
  2.6 Energy model ................................................ 25

CHAPTER 3 RSAC DESIGN METHODOLOGY AND EXAMPLES ........................................... 27
  3.1 Design I ...................................................... 28
    3.1.1 Simulation setup ........................................ 28
    3.1.2 Choice of $R$ and $C$ .................................... 29
    3.1.3 Validation of MSE model ................................ 29
    3.1.4 Validation of energy model ............................. 30
    3.1.5 Design optimization .................................... 33
    3.1.6 Comparison to digital implementation ................ 33
  3.2 Design II ..................................................... 36
    3.2.1 Simulation setup ........................................ 36
    3.2.2 Choice of $R$ and $C$ .................................... 38
    3.2.3 Validation of behavioral model ........................ 39
    3.2.4 Energy model ........................................... 42
    3.2.5 Design optimization .................................... 43
    3.2.6 Comparison to digital implementation ............... 43
CHAPTER 4 RSAC-BASED EMOTION RECOGNITION

4.1 Feature extraction using Gabor filtering . . . . . . . . . . . 53
4.2 Feature selection using Adaboost . . . . . . . . . . . . . . . 54
4.3 Support vector machine (SVM) . . . . . . . . . . . . . . . . . 55
4.4 Simulation results . . . . . . . . . . . . . . . . . . . . . . . . 57

CHAPTER 5 CONCLUSION AND FUTURE DIRECTIONS . . . . . . . . 60

REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
CHAPTER 1

INTRODUCTION

1.1 Motivation

In a world driven by technology and hand-held devices, the demand for ubiquitous computing with learning and decision making capabilities has grown tremendously in the past several years. These applications need to process large amounts of data acquired by sensing the surrounding environment and are subject to strict energy demands. In fact, smartphone users have reported that the single most important feature when choosing a phone is battery life [1]. Additionally, these applications have strict speed demands and require sufficient accuracy at the output. Current devices in the market lack intelligence, for they rely heavily on the cloud for processing. In order to embed intelligence into hand-held devices, we should ensure that the cost of computing in the learning algorithms is less than the cost of communicating with a server. Our work targets the computing front in order to achieve that goal.

A typical machine learning algorithm consists of: (i) a feature extraction engine to extract the relevant information from the input data and (ii) a classifier to make a decision (Fig. 1.1). The dot product (DP) kernel is a key kernel in both the feature extraction engine and the classifier. This kernel implements a variety of functions, including but not limited to, vector inner products, correlators, filters, convolutions, multiply-accumulate, $\ell_1$ and $\ell_2$ norms.

Conventionally, there are many ways to design the DP kernel: digital approach, analog approach and stochastic computing approach. In the remaining sections in this chapter, we will discuss these approaches in more detail.
Figure 1.1: A machine learning processing chain consists of a feature extractor and a classifier. The dot product is a key kernel.

1.2 Background

1.2.1 Digital approach

Sensory input $x(t)$ has to pass through an analog-to-digital converter (ADC), which consists of a sampler and a quantizer, before being processed digitally. The output is first sampled with a period $T_s$, and then quantized into a discrete value,

$$x_q[n] \equiv x(nT_s) + q[n],$$

where $x_q[n]$ is the $n$‘th discrete sample and $q[n]$ is the $n$‘th quantization error. Typically, for a uniform quantizer, the quantization step, $\Delta$, is constant:

$$\Delta \equiv \frac{x_{\text{max}} - x_{\text{min}}}{2^B_x},$$

(1.1)

where $x_{\text{max}}$ is the maximum input value, $x_{\text{min}}$ is the minimum input value, and $B_x$ is the number of bits used to represent the analog values. Assuming $x_{\text{max}} = x_m$ and $x_{\text{min}} = -x_m$, Equation (1.1) reduces to:

$$\Delta = \frac{x_m}{2^{B_x}-1}.$$  

(1.2)

It is typically assumed that the quantization error, $q$, is uniformly distributed between any two quantization levels, as shown in Fig. 1.2(b) [2],

$$(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{N} A_i B_i.$$
Figure 1.2: (a) Input-output relationship for a uniform quantizer, and (b) the probability density function of the quantization noise $q$.

Figure 1.3: A two-input digital DP kernel.
and independent of the input. Under these assumptions, we can compute the mean and variance of $q$ as follows:

$$E[q] = \frac{1}{\Delta} \int_{-\Delta}^{\Delta} q \, dq = 0.$$  

$$E[q^2] = \frac{1}{\Delta} \int_{-\Delta}^{\Delta} q^2 \, dq = \frac{\Delta^2}{12}.$$  

This will determine the precision at the output, which can also be cast in the form of a signal-to-quantization noise ratio (SQNR). The SQNR is defined as the ratio between the energy of the input and the quantization error:

$$SQNR \triangleq \frac{E[x^2]}{E[q^2]} = \frac{12(2B_x - 2)}{PAR}, \quad (1.3)$$

where $PAR$ is the peak-to-average ratio defined as:

$$PAR = \frac{E[x^2]}{x_m^2}.$$  

In dB, Equation (1.3) reduces to:

$$SQNR_{dB} = 6B_x + 4.8 - PAR_{dB},$$

where $PAR_{dB} = 20 \log_{10} PAR$. Essentially, for a 6 dB improvement in SQNR, $B_x$ would have to increase by only 1. To compute the DP, we first multiply each two numbers together using a multiplier, a Baugh-Wooley multiplier (BWM) for example, and then the numbers are added in a tree fashion using adders, ripple-carry adders (RCAs) for example. This is shown in Figure 1.3 for a length 2 DP. While there are many ways to model the energy consumption of digital circuits, a general one would be:

$$E_{tot} = g(B_x) \left( \alpha CV_{dd}^2 + E_{lkq} \right),$$

where $\alpha$ is the average transition activity, $C$ is the average output capacitance, $V_{dd}$ is the supply voltage, $E_{lkq}$ is the average leakage energy, and $g(.)$
is an increasing function that depends on the algorithm and architecture. For example, \( g(.) \) is linear if we are performing addition using a RCA and quadratic if performing multiplication using a BWM.

We will further discuss this in Chapter 3 when comparing the techniques developed in this thesis versus the digital approach.

### 1.2.2 Analog approach

An alternative but less common approach to digital processing is analog processing which is typically used when accuracy requirements are low and speed requirements are high. In the past, it was reported that analog processing was more energy efficient than digital processing for precision less than 8 bits [3]. However, currently this point of operation has shifted down to 5-6 bits [4]. While quantization is the main source of error in a digital system, sources of error in an analog system are due to various physical phenomena, such as thermal noise, flicker noise, shot noise which are even more prominent in advanced processes [5].

One possible design for a DP circuit is described in Fig. 1.4. The scheme
shows two differential amplifiers where the multiplication is done through the gain of each pair and the summing is done through the connection between the two amplifiers (i.e. Kirchhoff’s current law) \cite{6}. For a DP of length \(N\), we need \(N\) drain-coupled differential pairs as shown in Fig. 1.4. The operational transconductance of the transistors can be written as:

\[
g_{m_{1,2}} = \frac{2 \left( I_1 \right)}{V_{ov_{1,2}}} = \frac{f(\alpha_1)}{V_{ov_{1,2}}},
\]

\[
g_{m_{3,4}} = \frac{2 \left( I_2 \right)}{V_{ov_{3,4}}} = \frac{f(\alpha_2)}{V_{ov_{3,4}}},
\]

where \(I_1\) and \(I_2\) are the biasing currents in the first and second branch, respectively, \(V_{ov_i}\) is the overdrive voltage of the \(i\)th transistor, and \(\alpha_i\) is the control knob to the biasing currents through a function \(f(.)\). The output voltage is then obtained by the gain equation for differential amplifiers \cite{6} as:

\[
V_{out} = R_D g_{m_{1,2}} v_1 + R_D g_{m_{3,4}} v_2.
\]

Assuming \(V_{ov_{1,2}} = V_{ov_{3,4}} = V_{ov}\) and \(f(x) = kx\) for some \(k\), Equation (1.4) reduces to:

\[
V_{out} = \frac{k R_D}{V_{ov}} (\alpha_1 v_1 + \alpha_2 v_2).
\]

There are other ways to compute DPs using analog circuits. The most common one is the Gilbert cell, which is an extension to the circuit described above so that it computes a four-quadrant product. The Gilbert cell is widely used in modern receivers as a high-speed mixer \cite{7}. Another approach that has reported large energy savings is through a mixed-signal circuit that utilizes switched capacitors \cite{8}.

### 1.2.3 Stochastic computing approach

In 1967, B. R. Gaines \cite{9} proposed an alternative system, where a number is represented by a binary stream with the information stored in the 1s density. For example, to store the number 0.5, half the binary stream should consist of 1s. The stream length \(M\) is a design parameter. The longer the stream, the more accurate the output representation, as we will show later in this...
Figure 1.5: (a) Analog-to-Stochastic Converter (ASC), and (b) Stochastic-to-Digital Converter (DSC).

Figure 1.6: Example of a DP computed in stochastic computing using a multiplexer.
section. A typical way to obtain the stream from an analog voltage is through an analog-to-stochastic converter (ASC) [10]. This ASC compares the input to a random signal $U$, which is uniformly distributed between 0 and 1. An example is shown in Fig. 1.5(a). The probability of obtaining a 1 at the output is:

\[
p_O \triangleq P(O = 1) = P(U < I) = I.
\]

We use a stochastic-to-digital converter (SDC) [10] to go back to the digital domain. The SDC consists of a counter that counts the number of 1s, as shown in Fig. 1.5(b). The computing elements are simple such as an AND gate used for multiplication and multiplexer used for weighted-addition. Next, we show the conditions under which those computing elements work. This work focuses on the DP kernel and thus, we will focus on the multiplexer. For a multiplexer with inputs shown in Fig. 1.6, the output Boolean equation can be written as:

\[
O = I_1I_3 + I_2\bar{I}_3,
\]

where $I_1$ and $I_2$ are the two inputs to the multiplexer, and $I_3$ is the select input. Assuming that $I_1$, $I_2$, and $I_3$ are independent Bernoulli streams with parameters $p_1$, $p_2$, and $p_3$ respectively, we obtain that $O$ is also a Bernoulli stream with parameter:

\[
p_O \triangleq P(O = 1) = P(I_1I_3 + I_2\bar{I}_3 = 1) = P(I_1 = 1)P(I_3 = 1) + P(I_2 = 1)P(I_3 = 0) = p_1p_3 + p_2(1 - p_3).
\]

The counter at the backend counts all the 1s in the stream and thus, it is essentially summing $M$ independent Bernoulli random variables with parameter $p_O$ and dividing by the length $M$. Hence, the mean and variance
of the output can be written as:

\[
\mu = \frac{1}{M} (M_p O) = p_O
\]

\[
\sigma^2 = \frac{1}{M^2} (M_p O (1 - p_O)) = \frac{p_O (1 - p_O)}{M}.
\]

The energy consumed by a multiplexer used in this configuration can be written as:

\[
E_{tot} = M \left( \alpha CV_{dd}^2 + E_{lkg} \right),
\]

where \( M \) is the stream length, \( \alpha \) is the average transition activity, \( C \) is the output capacitance, \( V_{dd} \) is the supply voltage, and \( E_{lkg} \) is the leakage energy consumed by the multiplexer. Most recently, Blaauw showed how to use analog inputs instead of digital Bernoulli bit streams, which reduces the swing in the circuit and hence decreases energy consumption [11]. However, the stream length \( M \) is proportional to the energy consumption and thus, this technique becomes energy inefficient for large stream length.

Among the three approaches discussed in this chapter, the digital implementation remains the mainstream approach when designing systems today. Digital designers have the ability to use extensive tools that take the system from a behavioral description to a circuit layout. For low precision, the energy savings due to using the analog approach are typically barely significant (\(< 2\times\)[4]) and are hardly worth the extra complexity and time spent in designing such a system. For high precision, stochastic computing does not provide competition due to its energy inefficiency. In Chapter 2, we will introduce a new circuit, rapidly switched analog circuit (RSAC), that implements the DP kernel [12]. RSAC is based on fast switching analog inputs to the output via variable width digital pulses. Unlike the analog approach, RSAC is oblivious of the transistor models, which greatly reduces the complexity and time needed to design a system. Like stochastic computing, RSAC uses a multiplexer to compute the DP. However, the operation mechanism is very different and leads to an output convergence that is exponential and thus, the energy consumption and delay are greatly reduced.
CHAPTER 2

RAPIDLY SWITCHED ANALOG CIRCUIT (RSAC) PRINCIPLES

In this chapter, we introduce rapidly switched analog circuit (RSAC), a new energy-efficient mixed-signal circuit. RSAC implements the dot product (DP) kernel by fast switching the analog input to the output via variable width digital pulses. The input analog voltages constitute the first vector in the DP and are passed through an \( N \) input multiplexer with \( N \) select signals (Fig. 2.1(a)). By having only one select signal active at a time, and switching among the inputs at a high frequency with certain duty cycles, which constitute the second vector in the DP, the output voltage is obtained as the weighted sum of the input voltages. An example operation with \( N = 3 \) is depicted in Fig. 2.1(b). The output is sampled at \( T_S \), which is a multiple of the iteration period \( T \) (i.e. \( T_S = KT \) for some \( K \)). The iteration period \( T \) is the average on-time for each path. Section 2.1 presents the architecture in which a RSAC-based DP kernel can be implemented. A detailed accuracy analysis is presented in Section 2.2, followed by a description of circuit implementations in Section 2.3. Section 2.4 details two different implementations for the select generator. Finally, a behavioral model and an energy model are presented in Sections 2.5 and 2.6, respectively.

2.1 RSAC architecture

A RSAC-based processor architecture consists of \( J \) stages of DP kernels as shown in Fig. 2.2. \( SG_{jh} \) represents the \( h^{th} \) select generator in stage \( j \), with a total of \( H_j \) such select generators. Here, \( K_{jhi} \) represents the \( i^{th} \) computation kernel sharing \( SG_{jh} \), with a total of \( I_{jh} \) such kernels. The number of inputs going into kernel \( K_{jhi} \) is \( N_{jh} \). Accordingly, the total number of inputs going into stage \( j \) is \( \sum_{h=1}^{H_j} I_{jh} N_{jh} \) and the total number of outputs leaving stage \( j \) is \( \sum_{h=1}^{H_j} I_{jh} \). A system-wide multi-phase clock generator (MPCG) provides
Figure 2.1: Rapidly switched analog circuit (RSAC)-based DP kernel: (a) conceptual operation, and (b) output waveform for $N = 3$, $V_1 = 0.4$, $V_2 = 0.9$, $V_3 = 0.1$, and $p_1 = p_2 = p_3 = 1/3$.

Figure 2.2: Architecture of a rapidly switched analog circuit (RSAC) processor.
clock inputs to the $SG_{jh}$ select generators that generate the vector $\Phi_{jh}$, the select signals of period $N_{jh}T$ required for switching. $N_{jh}T$ will also be referred to as the round period, which is the time required to go through one round over all inputs.

The computation kernel consists of a RSAC-based DP kernel, as shown in Fig. 2.1(a). For a given input voltage vector $\mathbf{V} = (V_0, V_1, \ldots, V_{N-1})$ and weight vector $\mathbf{p} = (p_0, p_1, \ldots, p_{N-1})$, with $p_i \geq 0$ and $\sum_{i=0}^{N-1} p_i = 1$, the circuit computes the output voltage:

$$V_Z = \mathbf{V} \cdot \mathbf{p} = \sum_{i=0}^{N-1} V_i p_i.$$ 

In general, for a positive coefficient vector $\mathbf{c} = (c_0, c_1, \ldots, c_{N-1})$, the circuit computes the output voltage $V_Z = \frac{1}{\sum_{i=0}^{N-1} c_i} \sum_{i=0}^{N-1} V_i c_i$. This can be seen by identifying the relationship between the weight and the coefficient:

$$p_i = \frac{c_i}{\sum_{i=0}^{N-1} c_i}.$$ 

The weights $p_i$ are implemented by a set of non-overlapping pulses $\phi_i$ ($i = 0, \ldots, N-1$) of period $NT$, which is also the round period, and duty cycle $p_i$. By designing $T$ to be significantly smaller than the time constant of the RC network, $V_z$ will converge to $\sum_{i=0}^{N-1} V_i p_i$, with accuracy increasing at an exponential rate with the number of cycles until it settles. This circuit, being analog in nature, is also affected by various physical and mechanical phenomena that will affect its accuracy. Those will be discussed in detail in Section 2.2.

### 2.2 Principle of operation

The basic implementation for RSAC is shown in Fig. 2.3. Transmission gates are used to implement the switches to allow full swing at the output. $C_d$ is the total drain capacitance at node I and $R_i$ is the equivalent resistance of the $i^{th}$ path. A series resistance $R$ and a capacitor $C$ are added for proper operation, as will be shown shortly. By imposing condition:

$$C1): \quad (R + R_i) C \gg \max_{i=0, \ldots, (N-1)} R_i C_d,$$
Figure 2.3: Basic implementation of RSAC.
we can treat the circuit as a first-order RC circuit with step inputs. We define $V_{z,i}$ as the output sequence of iterating over all paths but ending in the $i$th path. From basic principles, we obtain the following relations:

$$V_{z,i}[n+1] = \begin{cases} V_i + (V_{z,(i-1)}N[n] - V_i)X_i, & i = 0 \\ V_i + (V_{z,(i-1)}N[n+1] - V_i)X_i, & i = 1, \ldots, N-1, \end{cases} \tag{2.1}$$

where $X_i = \exp(-\frac{p_iNT}{(R+R_i)C})$ and $(x)_N \triangleq x \mod N$. The relationship between the total number of iterations $K$ and the total number of rounds $n$ is $K = nN$.

Using Equation (2.1), we obtain the recurrence relations:

$$V_{z,i}[n+1] = \sum_{k=0}^{N-1} [V_{(i-k)}N(1 - X_{(i-k)}N) \prod_{j=0}^{k-1} X_{(i-j)}N] + XV_{z,i}[n], \tag{2.2}$$

where $X \triangleq \prod_{k=0}^{N-1} X_k$. The recurrence relations in Equation (2.2) are first-order linear difference equations with constant coefficients. Their convergence is guaranteed by the fact that $X < 1$ and they converge to:

$$\tilde{V}_{z,i} = \lim_{n \to \infty} V_{z,i}[n] = \frac{\sum_{k=0}^{N-1} [V_{(i-k)}N(1 - X_{(i-k)}N) \prod_{j=0}^{k-1} X_{(i-j)}N]}{1 - X}. \tag{2.3}$$

We note that $\exp(-x) = 1 - x + O(x^2)$ and for small $x$, we obtain $\exp(-x) \approx 1 - x$. By making $T \ll RC$, we can approximate $X_i$ by $1 - \frac{p_iNT}{(R+R_i)C}$. By imposing condition:

$$C2) \quad R \gg R_i,$$

we make $R$ large enough to dominate the equivalent resistances of the paths. Thus, we can approximate $X_i$ by $1 - \frac{p_iNT}{RC}$. This allows us to write Equation (2.3) as:

$$\tilde{V}_{z,i} = \frac{\sum_{k=0}^{N-1} [V_{(i-k)}N \left( \frac{p_{(i-k)}NT}{RC} \right) \prod_{j=0}^{k-1} \left( 1 - \frac{p_{(i-j)}NT}{RC} \right) \left( \frac{NT}{RC} \right) ]}{\prod_{j=0}^{k-1} \left( 1 - \frac{p_{(i-j)}NT}{RC} \right) \left( \frac{NT}{RC} \right)}. \tag{2.3}$$

Since $T \ll RC$, we will approximate $\left( \frac{T}{RC} \right)^2$ by 0 and obtain:
where $\sum_{k=0}^{N-1} V_k p_k$ is the ideal output. We note that any of those $N$ outputs can serve as an approximation to the ideal output and hence sampling the output at any instance of time (beyond convergence) can serve as the final output.

Another important aspect of this analysis is the convergence speed, since the number of iterations, $K = nN$, is proportional to the overall energy consumption (as will be seen in Section 2.6). $V_{z,i}[n]$ can be unrolled and written as:

$$V_{z,i}[n] = \tilde{V}_{z,i}(1 - X^n). \quad (2.4)$$

Next, we define the error at the output to be $e_i[n] \triangleq V_{\text{ideal}} - V_{z,i}[n]$, where $V_{\text{ideal}} = \sum_{k=0}^{N-1} V_k p_k$ is the ideal output. As such, the mean-square error (MSE) $J[n, X]$ is:

$$J[n, X] \triangleq E\{e_i[n]^2\} = J_{\text{min}} + X^n \alpha_A + X^{2n} \beta_A, \quad (2.5)$$

where $J_{\text{min}} = E\{(V_{\text{ideal}} - \tilde{V}_{z,i})^2\}$, $\alpha_A = 2E\{(V_{\text{ideal}} - \tilde{V}_{z,i})V_{z,i}\}$, and $\beta_A = E\{\tilde{V}_{z,i}^2\}$. Note that as $n$ increases, $X^n$ will go to zero and therefore:

$$\lim_{n \to \infty} J[n, X] = J_{\text{min}}.$$

The speed of convergence is exponential and similar to the one in a simple RC circuit. There is a trade-off between $K$ and $TTR \triangleq \tau T$, the tau-to-T ratio where $\tau \triangleq RC$, which will be studied in the next Chapter where we optimize the choice of $K$ and $TTR$ in the context of minimizing energy consumption for a fixed SQNR at the output.

Finally, the accuracy of the circuit will be affected by various physical phenomena as well as some issues related to the operation mechanics. These issues include: rise and fall times in select signals, charge sharing between $C_d$ and $C$ (Section 2.4.2), finite precision in select generation, clock feed-through,
PVT variations, and thermal noise.

We discuss here the issue associated with finite rise and fall times in select signals. The time when the switch turns on or off depends on the threshold voltage of the switch, the source voltage, and, most importantly, the rise and fall times. The latter are finite and different which will cause the effective duty cycle $\hat{p}_i$ to be different than $p_i$. Another problem with finite rise and fall times is that two select signals might overlap in that period. This overlap will activate two paths at the same time which is problematic. To see that, assume path $i$ and $j$ are active at the same time. This will make the equivalent voltage at node $I$ (see Fig. 2.3), when capacitors are 0:

$$V_I = \frac{RR_i}{R(R_i + R_j) + R_i R_j} V_j + \frac{RR_j}{R(R_i + R_j) + R_i R_j} V_i + \frac{R_i R_j}{R(R_i + R_j) + R_i R_j} V_Z,$$

which is approximately $\frac{R_i}{R_i + R_j} V_j + \frac{R_j}{R_i + R_j} V_i$ when condition $C2$ is imposed ($R \gg R_i, R \gg R_j$). This voltage will be driving the input and will cause interference at the output.

The circuit described in this section cannot be cascaded since it has a finite input impedance and non-zero output impedance. To solve that, we propose two circuit implementations in the next section and we show how the analysis in this section can be extended to explain their functionality.

### 2.3 RSAC circuit implementations

We have developed two circuit implementations for RSAC-based DP kernels, shown in Figure 2.4, that allow cascading without the need of a voltage buffer. This comes at the expense of a bias term at the output. These circuits compute:

$$V_Z = \frac{G}{\sum_{i=0}^{N-1} c_i} \sum_{i=0}^{N-1} V_i c_i - V_t,$$  \hspace{1cm} (2.6)

where $G$ is the attenuation factor of the source follower ($0 < G \leq 1$) and $V_t$ is the threshold voltage. Another limitation for these implementations is the constrained range of allowable inputs. N-RSAC (shown in Fig. 2.4(a)) uses NMOS transistors, where the inputs should be in the range $[V_t n, V_{dd}]$ and will shift the output down ($V_{tn} > 0$). P-RSAC (shown in Fig. 2.4(b)) uses PMOS
Figure 2.4: Circuit implementations of a RSAC-based DP kernel: (a) N-RSAC, and (b) P-RSAC.

Figure 2.5: The equivalent circuit of Fig. 2.4(b) when path $i$ is on and all other paths are off.
transistors, where the inputs should be in the range \([0, V_{dd} - |V_{tp}|]\) and will shift the output up \((V_{tp} < 0)\).

The voltage at node \(I\) at DC voltage (i.e. capacitors are 0) will help us relate the analysis of the implementation in Fig. 2.3 (shown in Section 2.2) and the analysis of the implementations in Fig. 2.4. When path \(i\) is on, in Fig. 2.3, we use KVL to obtain:

\[
V_I = \frac{R}{R + R_i} V_i + \frac{R_i}{R + R_i} V_Z,
\]

which is approximately \(V_i\) when condition \(C2\) is imposed \((R \gg R_i)\). Hence, the RC circuit is being driven by \(V_i\), which makes the analysis in Section 2.2 valid. When path \(i\) is on in Fig. 2.4(a), the circuit reduces to the one shown in Fig. 2.5. This circuit is a variation to the source follower configuration and thus, the voltage at node \(I\) can be written as:

\[
V_I = GV_i - V_t, \tag{2.7}
\]

where \(G\) is the attenuation factor of the source follower and \(V_t\) is the threshold voltage. The exact value of \(G\) depends on \(V_{BIAS}\) and will be investigated in the Chapter 3. By substituting the value of \(V_I\) in Equation \(2.7\) for \(V_i\) in the analysis above, we obtain the relation in Equation \(2.6\). The same analysis holds for the implementation in Fig. 2.4(b).

When cascading RSAC-based DP circuits, the inputs to the second stage consist of the outputs of the first stage, which are not constant values but
rather exponentials. As discussed in Section 2.2, the exponentials can be substituted by a linear approximation. With this approximation, we will describe the behavior of the latter stages of a RSAC-based system with ramp inputs. Figure 2.6 shows the input which is described as:

\[ V_i(t) = \begin{cases} V_i(0) + \alpha t, & 0 \leq t \leq p_iNT \\ \beta, & p_iNT < t, \end{cases} \]

where \( \alpha \triangleq \frac{\beta - V_i(0)}{p_iNT} \) is the slope. The one-sided Laplace transform of \( V_i(t) \) can be written as:

\[ V_i(s) = \frac{V_i(0)}{s} + \alpha \frac{1 - e^{-sp_iNT}}{s^2}. \]

For an RC circuit, the one-sided Laplace transform of the output is:

\[ V_o(s) = \frac{V_i(s)}{1 + RCs} + \frac{V_o(0)}{1 + RCs}. \]

We then obtain the output by replacing \( V_i(s) \) by its value and taking the inverse Laplace transform:

\[
V_o(t) = V_i(0) + (V_o(0) - V_i(0)) \exp(-\frac{t}{RC})u(t) \\
+ \alpha \left( t - RC + RC \exp(-\frac{t}{RC}) \right) u(t) \\
+ \alpha \left( (t - p_iNT) - RC + RC \exp(-\frac{t - p_iNT}{RC}) \right) u(t - p_iNT),
\]

where \( u(t) \) is the unit step function. We make here the approximation that \( \exp(-x) \approx 1 - x \) and thus, we can approximate the second term in Equation (2.8) as:

\[
t - RC + RC \exp(-\frac{t}{RC}) \approx t - RC + RC(1 - \frac{t}{RC}) = 0.
\]

Similarly, the third term can also be approximated by 0. This reduces Equation (2.8) to:
Figure 2.7: Length $M$ ring oscillator.

\[
V_o(t) = V_i(0) + (V_o(0) - V_i(0)) \exp\left(-\frac{t}{RC}\right)u(t),
\]

which is the same response as an RC circuit excited by a step input of $V_i(0)$. Thus, the analysis provided in Sections 2.2 and 2.3 applies for the latter stages of a RSAC-based system. We will verify this in Chapter 3.

In Section 2.4, we discuss the different design alternatives for the select generator.

2.4 Select generation

Generation of the select signals is needed for proper RSAC operation. The coefficients $p_i$ could be constant (e.g. filtering) or programmable (e.g. read from memory), each resulting in a very different design for the select generator. However, in both cases, the select generator is fed by a MPCG that provides the clock inputs at different phases. The MPCG is a ring counter of length $M$, running at a frequency of $f_{\text{Clk}} \triangleq \frac{1}{T_{\text{Clk}}}$ (Fig. 2.7). The choice of $M$ depends largely on the precision required in the select generation and the topology used for the select generator, as will be shown shortly.

Alternatives such as counter-based oscillators or inverter-based ring oscillators provide similar accuracy but the ring counter in Figure 2.7 makes the design of the following select generator simpler and more energy efficient.

In both designs, the coefficients are implemented using a finite precision of $B$ bits. For a coefficient vector $c = (c_0, c_1, \ldots, c_{N-1})$, we define the quantized version $\hat{c} = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{N-1})$, where:
2.4.1 Constant coefficients

When the coefficients are constant, such as in the case of filtering, we can obtain the select signals by ORing the according outputs of the MPCG. For example, consider the case of applying a $3 \times 3$ Gaussian blur filter with $\sigma^2 = 0.85$ with coefficients shown in Fig. 2.8(a). In this case, $c = \hat{c}$ and thus the coefficients are unaltered in the implementation. The length of the MPCG is obtained by computing $M = \sum_{i=0}^{N-1} \hat{c}_i$, which is 16 in this case. The implementation of the select generator is shown in Fig. 2.8(b). The relationship between the iteration period $T$ and the clock period $T_{Clk}$ is $T = \frac{M}{N}T_{Clk}$ in this design, which is the average on-time of any path.

2.4.2 Programmable coefficients

When the coefficients are programmable, such as when they are read from memory, we can formulate the problem as a finite state machine for which the state transition diagram is shown in Fig. 2.9. The $cnt$ signal is the output of a $B$-bit counter and is an input to this state machine, in addition to the coefficients $\hat{c}_i$. The output of this state diagram is $\Phi$, the $N$-bit signal that
Figure 2.9: State transition diagram for the input-dependent select generator.
Figure 2.10: Optimized logic design for the programmable select generator. An SR latch is used to distinguish the trigger point between the white and gray states in Fig. 2.9.

represents a one-hot encoding of $N$ states (in particular, the ones in white). The extra $N$ states (the ones shown in gray) represent idle states, which were added to keep the control of the counter unaltered since it is shared by all state machines in the processor. The implementation of this select generator is optimized, in Fig. 2.10:

- Each successive white and gray state is merged into one state. An extra bit, $b$ in Fig. 2.10, serves to distinguish between the original states at the final output.

- The states are one-hot encoded in order to be re-used as the outputs, thus avoiding extra circuitry.

- A circular shifter is used to represent the states since the transitions in the output are one-directional.

- A ring-counter is used instead of a counter to facilitate the comparison circuitry. Although this counter is more expensive, it is shared by all select generators running in parallel. This extra cost will be attenuated by a factor of $(\sum_{j=1}^{J} \sum_{h=1}^{H_j} I_{jh})$, which is typically large.
Figure 2.11: The equivalent circuit of the RSAC-based DP kernel when all paths are off.

We note that the relationship between iteration period $T$ and the clock period $T_{Clk}$ is $T = 2^{B-1}T_{Clk}$ in this design, which is the average on-time of any path.

This design gives rise to charge sharing between $C_d$ and $C$ when the signal path is all idle. To see that, assume path $i$ has been on for $\hat{c_i}T_{Clk}$ and now turns off for the remaining period, $T_{off} = (2^B - \hat{c_i})T_{Clk}$. Figure 2.11 shows the equivalent circuit in that period. Solving for the output voltage after $T_{off}$, we obtain:

$$V_z(T_{off}) = V_f + K \exp\left(-\frac{T_{off}(C_d + C)}{RC_d C}\right)$$

$$V_f = \frac{C_d}{C_d + C} V_i(0) + \frac{C}{C_d + C} V_z(0),$$

where $V_i(0)$ is $V_i$, $V_z(0)$ is the expected output, and $K$ is some positive factor. For $C \gg C_d$ and $T_{off} \gg RC_d$ (which is typically the case), we obtain $V_z(T_{off}) \approx V_f$. This will cause interference at the output.

In Sections 2.5 and 2.6, we introduce a behavioral model that predicts the output of RSAC and an energy model that predicts the energy consumed by RSAC.

2.5 Behavioral model

Given the issues described in Sections 2.2 and 2.3, we can establish a behavioral model for the RSAC circuit in Fig. 2.3:
\[ V_{out}[K + 1] = V_{(K)N} + (1 - \epsilon) \left( V_{out}[K] - V_{(K)N} \right) \exp \left( -\frac{(\hat{p}(K)_N + \Delta)N}{TTR} \right) + \eta, \]

where \( \epsilon \triangleq \frac{C_d}{C_d + C} \) is the factor due to charge sharing, \( \hat{p}_i \) is the quantized version of the weight, \( \Delta \) is the adjustment to the weight due to rise and fall times and PVT variations in the threshold voltages, and \( \eta \) is the term due to thermal noise and the various other phenomena not captured in the analysis. Accordingly, \( \Delta \) and \( \eta \) will be obtained using simulations before using this model for system design. This same behavioral model can be used for the circuit implementations in Fig. 2.4(a) and (b) by adjusting \( V_{(K)N} \) to \( GV_{(K)_N} - V_t \) to obtain:

\[ V_{out}[K + 1] = GV_{(K)_N} - \epsilon V_t \]
\[ + (1 - \epsilon) \left( V_{out}[K] - GV_{(K)_N} \right) \exp \left( -\frac{(\hat{p}(K)_N + \Delta)N}{TTR} \right) + \eta. \]

2.6 Energy model

A RSAC implementation of \( \sum_{j=1}^{J} \sum_{h=1}^{H_j} I_{jh} \) DP kernels requires one MPCG. Hence, the total energy consumption for a kernel can be written as:

\[ E_{tot}[n, X] \triangleq E_{RSAC}[n, X] + \frac{E_{MPCG}}{\sum_{j=1}^{J} \sum_{h=1}^{H_j} I_{jh}}, \tag{2.9} \]

where \( E_{RSAC,jh}[n, X] \) is the energy consumption of the RSAC signal path and select generation over \( n \) rounds, and \( E_{MPCG} \) is the energy consumption of the MPCG. \( E_{MPCG} \) depends largely on the topology used and thus will be estimated through simulations, while \( E_{RSAC}[n, X] \) can be written as:

\[ E_{RSAC}[n, X] \triangleq E_{sig}[n, X] + \frac{E_{sel}[n]}{I_{jh}}, \]

where \( E_{sig}[n, X] \) is the energy consumed in the signal path over \( n \) rounds and \( E_{sel}[n] \) is the energy dissipated by the select generator over \( n \) rounds. These quantities have different expressions, depending on the implementation used. When using the basic circuit implementation described in Section 2.2, the
energy consumed in the signal path is denoted by \( E_{\text{sig,bas}}[n, X] \) and can be written as:

\[
E_{\text{sig,bas}}[n, X] = n \frac{C_d}{2} \sum_{k=0}^{N-1} (V_{(k+1)_{N}} - V_k)^2 + C \sum_{l=0}^{n-1} V_0 (V_{z,l}[l + 1] - V_{z,N-1}[l]) \\
+ C \sum_{l=0}^{n-1} \sum_{k=1}^{N-1} V_k (V_{z,k}[l + 1] - V_{z,k-1}[l + 1]) \\
= nC_d c_E + C \left( \tilde{V}_z^2 (1 - X^n) + nd_E \right),
\]

with \( c_E \triangleq \sum_{k=0}^{N-1} \frac{(V_{(k+1)_{N}} - V_k)^2}{2} \) and \( d_E \triangleq \left( \sum_{k=0}^{N-1} V_k^2 (1 - X_k) - \tilde{V}_z^2 (1 - X) \right) \).

In the case of using the implementations in Figure 2.4, the energy consumed in the signal path is denoted by \( E_{\text{sig,casc}}[n, X] \) and can be written as:

\[
E_{\text{sig,casc}}[n, X] = nI_{\text{Bias}} V_{dd} NT.
\]

The energy dissipated by the select generator depends on the topology used. When the constant coefficient select generator topology in Section 2.4.1 is used, we obtain:

\[
E_{\text{sel,cst}}[n] = nC_{\text{sel}} V_{dd}^2,
\]

where \( C_{\text{sel}} \) includes the total gate capacitance of the access transistors in the DP kernel and total drain capacitance of the constant coefficient select generator. When the programmable select generator topology in Section 2.4.2 is used, we can write a more explicit expression:

\[
E_{\text{sel,prog}}[n] = n \left( (N + 2)E_{\text{NOR},2} + 2^B E_{\text{MUX},2} + BE_{\text{MUX},N} \right), \quad (2.10)
\]

where \( E_{\text{NOR},2} \) is the energy dissipated by a two-input NOR gate, \( E_{\text{MUX},2} \) and \( E_{\text{MUX},N} \) are the energies dissipated by a two-input multiplexer and \( N \)-input multiplexer, respectively. Equation (2.10) shows the dependency of the energy consumed by the select generator and the length \( N \) and the precision \( B \).

In Chapter 3, we will introduce a design methodology for RSAC followed by two design examples.
CHAPTER 3

RSAC DESIGN METHODOLOGY AND EXAMPLES

RSAC has a rich design space. Namely, $R$, $C$, and $T$ are left as parameters to be controlled by the designer. Once these parameters are chosen, $TTR = \frac{RC}{T}$ and $K = \frac{T_s}{T}$ are automatically decided and will affect the accuracy and the energy consumption of the circuit. Our objective is to minimize energy consumption for a given SQNR constraint. Hence, the design methodology is as follows:

**Step 1:** Obtain $R_{\text{min}}$ and $C_{\text{min}}$ that satisfy conditions $C_1$ and $C_2$ (Section 2.2).

**Step 2:** Obtain SQNR vs. energy plots for different $TTR$ while varying $K$.

**Step 3:** Obtain the minimizing pair $TTR_{\text{min}}$ and $K_{\text{min}}$ that have the minimum energy consumption for the given SQNR constraint.

**Step 4:** Choose $R$ and $C$. Initially, $R = R_{\text{min}}$ and $C = C_{\text{min}}$.

**Step 5:** Choose $T = \frac{RC}{TTR_{\text{min}}}$ and $K = \frac{T_s}{T}$. If $T_{\text{Clk}} (\propto T)$ is unfeasible, go to **Step 4** and increase $R$ and/or $C$. If $K < K_{\text{min}}$, go to **Step 3** while ignoring the current $TTR_{\text{min}}$ and $K_{\text{min}}$.

We will assume that $T_{\text{Clk}}$ is always feasible and thus, we will merge **Step 1** and **Step 4**. However, in real life applications, this assumption should be removed and design methodology above should be kept intact.

In this chapter, we will apply this design methodology to two different design examples. Design I in Section 3.1 uses the basic circuit in Fig. 2.3 with a constant coefficient select generator, shown in Fig. 2.8. The simulations are conducted using a 130 nm CMOS process. We verify in this design the mean-square error (MSE) and energy models developed in Sections 2.2 and 2.6, respectively. We apply the design methodology to optimize RSAC and finally compare it against a digital implementation.
Design II in Section 3.2 uses the implementations in Fig. 2.4 with an input-dependent select generator, shown in Fig. 2.10. The simulations are conducted using a 28 nm CMOS fully-depleted silicon-on-insulator (FDSOI) process. We verify the behavioral model for RSAC which was developed in Section 2.5. We apply the design methodology to optimize RSAC and finally compare it against a digital implementation.

3.1 Design I

In this section, a RSAC-based DP kernel is used to implement an image filter. We will investigate two filters, the average filter and the Gaussian blur (GB) filter whose coefficients are shown in Fig. 2.8(a) with its select generator in Fig. 2.8(b).

3.1.1 Simulation setup

A RSAC-based DP kernel (Fig. 2.1(b)) is used to implement an image filter. We only have one processing stage and thus, $J = 1$. Circuit simulations in a 130 nm CMOS process at the nominal corner were performed for image sizes $30 \times 30$, $60 \times 60$, and $120 \times 120$. The MPCG operates at $T_{\text{Clk}} = 400 \text{ ps}$. Three average filters of lengths $M = 9$, 25, and 49 were implemented, which correspond to a $3 \times 3$, $5 \times 5$ and $7 \times 7$ window, respectively. These filters do not require any combinational logic at the output of the ring counter since the different phases already represent the coefficients. A $3 \times 3$ Gaussian blur filter with $\sigma^2 = 0.85$ (Fig. 2.8(a)) has also been implemented. The $D$-flipflops in the ring counter are implemented using true single-phase clocking (TSPC). Select signals are generated using static-CMOS based NOR and NAND gates. Select signals corresponding to coefficients of unit value do not need a logic stage but are still passed through inverters to match the delay of other select signals. The image is processed on a per-row basis but the number of select generators were duplicated, depending on the size of the image, to ensure sharp rise and fall of the select signals. For the $30 \times 30$ image, $H_1 = 1$ and $I_{11} = 30$, while for the $60 \times 60$ image, $H_1 = 2$ and $I_{11} = I_{12} = 30$, and finally for the $120 \times 120$ image, $H_1 = 4$ and $I_{11} = I_{12} = I_{13} = I_{14} = 30$. 
Figure 3.1: Circuit simulation of a RSAC-based GB filter with $TTR = 45$ ($K = 5TTR$). Energy and MSE saturate at $R \geq 700 \text{k}\Omega$.

3.1.2 Choice of $R$ and $C$

Step 1 in the design methodology is applied here by designing $R$ and $C$ to satisfy conditions $C1$ and $C2$ (Section 2.2). To obtain a value for $R$, the circuit was simulated at different values of $R$ while keeping $TTR = \frac{RC}{T}$ at a fixed value by adjusting $C$ accordingly. $C1$ was satisfied by choosing $RC$ large enough to dominate $\tau_{\text{max}}$. Figure 3.1 shows the plot of the MSE and energy consumption of the circuit vs. $R$ with $TTR = 45$. The same trend was observed for different values of $TTR$. A total of 5 $TTR$ iterations were performed. A reduction in MSE and energy consumption can be seen as $R$ increases until $R \approx 700 \text{k}\Omega$ for the MSE and $R \approx 900 \text{k}\Omega$ for energy. Thus in all simulations, $R$ was chosen to be $1 \text{M}\Omega$, and the value of $C$ was set to obtain a specific value for $TTR$. In practice, $C$ can consist solely of the intrinsic capacitance, as long as condition $C1$ in Section 2.2 is satisfied. If so, we can use $T_{\text{Clk}}$ as a variable to control $TTR$.

3.1.3 Validation of MSE model

Comparison of circuit simulations and Equation (2.5) for MSE vs. iterations are shown in Fig. 3.2. Figure 3.2(a) shows the GB filter applied to three
Figure 3.2: Validation of Equation (2.5) for: (a) GB filter with different images and (b) GB filter with varying $TTR$.

Table 3.1: Fitted accuracy parameter values of a GB filter with $TTR = 45$ ($C = 32\, \text{fF}$).

<table>
<thead>
<tr>
<th>Image</th>
<th>$\alpha_A ,(V^2)$</th>
<th>$\beta_A ,(V^2)$</th>
<th>$J_{\text{min}} ,(V^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>$-3.07 \times 10^{-4}$</td>
<td>$2.3 \times 10^{-2}$</td>
<td>$1.4 \times 10^{-5}$</td>
</tr>
<tr>
<td>I2</td>
<td>$-1.11 \times 10^{-3}$</td>
<td>$9.83 \times 10^{-2}$</td>
<td>$6.17 \times 10^{-5}$</td>
</tr>
<tr>
<td>I3</td>
<td>$-1.44 \times 10^{-3}$</td>
<td>$8.9 \times 10^{-2}$</td>
<td>$5.96 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

different $30 \times 30$ images (I1, I2 and I3). I2 and I3 were chosen to have similar statistics while being different from those of I1. Weighted least-squares fitting was applied to obtain the parameters $\alpha_A$, $\beta_A$ and $J_{\text{min}}$ in Section 2.2. The fitted parameters obtained for $TTR = 45$ ($C = 32\, \text{fF}$) are tabulated in Table 3.1. It can be seen that the parameters for I2 and I3 are similar in value, as expected.

In Fig. 3.2(b), $TTR$ was varied for I1 to see its effect on MSE. As expected, the parameters $\alpha_A$ and $\beta_A$ in this model are a weak function of $TTR$ and depend largely on the input statistics. The asymptotic MSE value $J_{\text{min}}$ decreases as $TTR$ increases. However, the decrease in MSE is minimal due to the overall accuracy being dominated by the imperfection of the various issues described in Section 2.2.

3.1.4 Validation of energy model

The energy consumption of this design can be written as:
Figure 3.3: Validation of Equation (3.1) for: (a) GB filter with different images and (b) average filter with different window sizes.

Figure 3.4: Energy per computation breakdown of the GB filter applied to 30 \times 30, 60 \times 60, and 120 \times 120 images.

Table 3.2: Fitted energy parameter values of a GB filter with different TTR.

<table>
<thead>
<tr>
<th>TTR (C)</th>
<th>( \alpha_E ) (J)</th>
<th>( \beta_E ) (J)</th>
<th>( E_{MPCG} ) (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45 (32 fF)</td>
<td>( 8.12 \times 10^{-15} )</td>
<td>( 2.26 \times 10^{-15} )</td>
<td>( 2.25 \times 10^{-12} )</td>
</tr>
<tr>
<td>90 (64 fF)</td>
<td>( 8.12 \times 10^{-15} )</td>
<td>( 4.07 \times 10^{-15} )</td>
<td>( 2.25 \times 10^{-12} )</td>
</tr>
<tr>
<td>135 (96 fF)</td>
<td>( 8.13 \times 10^{-15} )</td>
<td>( 5.82 \times 10^{-15} )</td>
<td>( 2.25 \times 10^{-12} )</td>
</tr>
</tbody>
</table>
Table 3.3: Fitted energy parameter values of an average filter with \( TTR = 45 \).

<table>
<thead>
<tr>
<th>Window ((C))</th>
<th>( \alpha_E (J) )</th>
<th>( \beta_E (J) )</th>
<th>( E_{MPCG} (J) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 \times 3(18 fF)</td>
<td>(5.97 \times 10^{-15})</td>
<td>(1.47 \times 10^{-15})</td>
<td>(7.29 \times 10^{-13})</td>
</tr>
<tr>
<td>5 \times 5(50 fF)</td>
<td>(1.98 \times 10^{-14})</td>
<td>(3.18 \times 10^{-15})</td>
<td>(5.45 \times 10^{-12})</td>
</tr>
<tr>
<td>7 \times 7(98 fF)</td>
<td>(2.41 \times 10^{-14})</td>
<td>(3.07 \times 10^{-15})</td>
<td>(1.06 \times 10^{-11})</td>
</tr>
</tbody>
</table>

\[
E_{RSAC}[n,X] = n\alpha_E + \beta_E (1 - X^n), \quad (3.1)
\]

for:

\[
\alpha_E = C_d c_E + C d_E + n \frac{C_{sel}}{30} \\
\beta_E = CV z^2,
\]

where \( c_E, d_E, \) and \( C_{sel} \) are defined in Section 2.6. Comparison of circuit simulations and Equation (3.1) for energy vs. iterations are shown in Fig. 3.3. Figure 3.3(a) shows the GB filter applied to the same three images as above. Similarly, we can see the dependence of the parameters on input statistics. To see the dependence of \( \beta_E \) on \( TTR \), the same simulation was run on \( I_1 \) with varying \( TTR \). The fitted energy parameter values \( \alpha_E \) and \( \beta_E \) are tabulated in Table 3.2. It can be seen that \( \beta_E \) changes with varying \( TTR \), while \( \alpha_E \) remains constant, as predicted by our energy model.

Figure 3.3(b) shows the average filter of different window sizes applied to \( I_1 \), with \( TTR = 45 \). The energy parameters are tabulated in Table 3.3. It can be seen that the parameters \( \alpha_E \) and \( \beta_E \) remain relatively constant, as the input statistics and \( I_{1h} \) remains constant. Also, the ring counter energy consumption is constant per row of computation for the different image sizes. In our design, \( E_{MPCG} \gg E_{RSAC} \), and energy per single computation is dominated by \( \frac{E_{MPCG}}{\sum_{h=1}^{H} I_{1h}} \). As more computation kernels can share the same ring counter, more energy benefits can be obtained (Fig. 3.4). Finally, the energy consumption of the ring counter is proportional to \( M^2 \), as can be seen in Table 3.3.
3.1.5 Design optimization

The remaining steps in the design methodology are applied in this section. HSPICE simulations were performed on a GB filter for I1, by sweeping over TTR at various iterations (Fig. 3.5). The number of iterations for a given SQNR should be minimized since the linear component in $E_{RSAC}$ dominates ($\alpha_E \gg \beta_E$). From the close-up view (Fig. 3.6) of $9 \leq TTR \leq 18$, it can be seen that the minimum number of rounds for a given SQNR occur at $n = \frac{5}{9} TTR - 2$. The optimal curve is obtained by joining these minimizing points. SQNR improvements saturate around $TTR = 45$ due to the overall accuracy being dominated by the imperfection of the select-signal generation block.

3.1.6 Comparison to digital implementation

The RSAC-based DP kernel is compared against a digital logic implementation using ripple carry adders (RCA) and Baugh-Wooley multipliers (BWM). To estimate the energy consumption of the adders and multipliers, the energy for a 1-bit full adder ($E_{FA}$) using a mirror-adder structure loaded with FO4 inverters was simulated. In a 130 nm process, $E_{FA} = 18.63$ fJ. Energy
Figure 3.6: Close-up view of SQNR vs. number of rounds of a GB filter for a 30 × 30 image for 9 ≤ TTR ≤ 18.

Figure 3.7: Comparison of energy per DP computation vs. SQNR for a GB filter.
Figure 3.8: Filtered images: (a) original image, (b) noisy image, (c) ideal filtered image, and (d) RSAC filtered image with $TTR = 45$ and $n = \frac{5}{9} TTR - 2$. 
Table 3.4: Fitted source follower parameter values for the N-RSAC and P-RSAC implementations.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>$V_{BIAS}$</th>
<th>$G$</th>
<th>$V_{t}$</th>
<th>$E { v_{SF}^2 }$</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-RSAC</td>
<td>0.24 V</td>
<td>0.868</td>
<td>0.24 V</td>
<td>$6.91 \times 10^{-7}$</td>
</tr>
<tr>
<td>P-RSAC</td>
<td>0.6 V</td>
<td>0.818</td>
<td>$-0.49$ V</td>
<td>$6.93 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

The energy consumption of a $B$-bit RCA is then estimated to be:

$$E_{RCA}[B] = \alpha_{0\rightarrow1} B E_{FA},$$

(3.2)

where $\alpha_{0\rightarrow1}$ is the activity factor of the RCA. We assume the inputs are uniformly distributed and thus $\alpha_{0\rightarrow1}$ is 0.25. The energy consumption of a $B$-bit BWM is lower-bounded [13] by:


The SQNR vs. energy per DP computation is shown in Fig. 3.7 for a 120 $\times$ 120 image. For $SQNR \approx 24$ dB, energy savings are approximately 32$\times$; whereas for $SQNR \approx 30$ dB, the energy savings are approximately 19$\times$. These savings are pessimistic, as $E_{BW}$ was based on a lower bound. The filtered output of our RSAC-based image filter is shown in Fig. 3.8(d).

### 3.2 Design II

In this section, a RSAC-based DP kernel is used to implement a DP between two arbitrary vectors of length 8. We will first discuss the simulation setup, followed by the design optimization for RSAC and finally compare it to a digital design for the same sampling period, $T_S$.

#### 3.2.1 Simulation setup

Circuit simulations in 28 nm fully-depleted silicon on insulator (FDSOI) at the nominal corner were performed. The select generator in Fig. 2.10 was synthesized with $B = 6$, $N = 8$ and mapped into an HSPICE netlist to simulate with the computation kernel. A precision of 6 bits was chosen in the select generator to make sure that this block will not be the limiting factor
in the accuracy of RSAC. The RSAC-based DP kernel was implemented using the circuits in Fig. 2.4(b) and (c). $V_{B\text{IAS}}$ was swept by steps of 0.01 V from 0.5 V to 0.2 V for N-RSAC and 0.5 V to 0.7 V for P-RSAC. Weighted least-squares fitting was then applied to obtain the parameters $G$ and $V_t$. The value of $V_{B\text{IAS}}$ and the fitted parameters obtained were tabulated in Table 3.4. $e_{SF}$ is the error due to the weighted least-squares fitting and $E\{e_{SF}^2\}$ is the MSE. We chose $V_{B\text{IAS}}$ to achieve an operation point that minimizes energy consumption while keeping the MSE at the output low enough to not dominate the overall accuracy. The allowable input range for N-RSAC is [0.35, 0.80] and [0.00, 0.40] for P-RSAC, which leads to an output range of [0.06, 0.40] and [0.49, 0.82], respectively. The select generator used here allows charge sharing and hence we will fix the value of $C$ and $R$, and use $T_{\text{Clk}}$ as a knob to control $T$ and in turn, $TTR$. Finally, the inputs to the MUX and the coefficients of the select generator were both generated at random using a uniform distribution.
3.2.2 Choice of $R$ and $C$

Step 1 in the design methodology is applied here by designing $R$ and $C$ to satisfy conditions $C1$ and $C2$ (Section 2.2). Unlike Design I, we need to obtain acceptable values of both $R$ and $C$. Indeed, a large enough value of $C$ will limit the charge sharing to obtain acceptable accuracy at the output. Figures 3.9 and 3.10 show the MSE vs. $R$ and $C$, respectively. In Figure 3.9, $C$ is fixed to 400 fF and $T_{Clk}$ is changed to keep $TTR = 10$. The total number of iterations is $5TTR$. It can be seen that the MSE saturates at $R \geq 200$ kΩ. The same trend was observed for different values of $TTR$. As $R$ increases, the area increases and the clock period $T_{Clk}$ will need to increase accordingly in order to keep $TTR$ constant. In turn, $T_S$ will also increase and so will the energy consumption. $R = 200$ kΩ was chosen as a good balance between the improvement in MSE and the increase in the sampling period. In Figure 3.10, $R$ is fixed to 1000 kΩ and $T_{Clk}$ is changed to keep $TTR = 10$. The total number of iterations is $5TTR$. It can be seen that the MSE saturates at $C \geq 200$ fF. The same trend was observed for different values of $TTR$. As $C$ increases, the area increases and the clock period $T_{Clk}$ will need to increase accordingly in order to keep $TTR$ constant. In turn, $T_S$ will also increase and so will the energy consumption. $C = 200$ fF was chosen as a good balance.
Figure 3.11: The probability mass function for the error between the circuit simulation output and behavioral model output with $\eta$ set to 0.

between the improvement in MSE and the increase in the sampling period.

### 3.2.3 Validation of behavioral model

The behavioral model described in Section 2.5 is verified in this section. We review the model again:

$$V_{\text{out}}[K+1] = GV_{(K)N} - \epsilon V_t$$

$$+ (1 - \epsilon)(V_{\text{out}}[K] - GV_{(K)N}) \exp\left(-\frac{(\hat{p}_{(K)N} + \Delta)N}{TTR}\right) + \eta,$$

where $\epsilon \triangleq \frac{C_d}{C_d+C}$ is the factor due to charge sharing, $\hat{p}_i$ is the quantized version of the weight, $\Delta$ is the adjustment to the weight due to rise and fall times and PVT variations in the threshold voltages, $\eta$ is the term due to thermal noise and the various other phenomena not captured in the analysis. $\epsilon$ is obtained through fitting as 0.002. $\Delta$ is negligible in this design due to the high precision in the select generator ($B = 6$). The parameter $\eta$ is obtained as the difference between circuit simulation output and the behavioral model output with $\eta$ set to 0. The values of the inputs were swept from 0.35 V to 0.80 V for N-RSAC and 0.00 V to 0.40 V for P-RSAC in steps of 0.01 V. The
Figure 3.12: Validating Equation (3.3) for a one-stage RSAC-based system. The average normalized error is 8.27%.

Figure 3.13: Validating Equation (3.3) for a two-stage RSAC-based system. The average normalized error is 12.89% for the first stage and 13.17% for the second stage.
Figure 3.14: Validating Equation (3.3) for a three-stage RSAC-based system. The average normalized error is 11.23% for the first stage, 14.54% for the second stage, and 21.17% for the third stage.

value of $T$ was swept from $0.01\tau$ to $\tau$ in steps of $0.01\tau$. The initial value of the output was also swept from $0.00\,\text{V}$ to $0.45\,\text{V}$ for N-RSAC and $0.49\,\text{V}$ to $0.82\,\text{V}$ for P-RSAC in steps of $0.01\,\text{V}$. The probability mass function of $\eta$ is shown in Fig. 3.11. $\eta$ has a normalized variance of $4.23 \times 10^{-6}$, which makes its effect on the behavioral model minimal and will be ignored in the next parts. We now verify the behavioral model in emulating the RSAC-based DP. For a RSAC-based DP of length 8, we perform circuit simulations on 200 uniformly distributed inputs and we compare the output to the output obtained from using the behavioral model. Figure 3.12 compares the output from circuit simulations and the output from the behavioral model. The average normalized error is 8.27%. Figure 3.13 shows the output from circuit simulations and output from the behavioral model for a two-stage RSAC-based DP of length 2. The total number of samples is 200 and the inputs are uniformly distributed. The average normalized error is 12.89% for the first stage and 13.17% for the second stage. Figure 3.14 shows the output from circuit simulations and output from the behavioral model for a three-stage RSAC-based DP of length 2. The total number of samples is 200 and the inputs are uniformly distributed. The average normalized error is 11.23% for
Table 3.5: Actual and estimated energy consumed by the select generators of different lengths and precision.

<table>
<thead>
<tr>
<th></th>
<th>Actual $E_{\text{sel,prog}}(fJ)$</th>
<th>Estimated $E_{\text{sel,prog}}(fJ)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 8, B = 6$</td>
<td>9.75</td>
<td>10.5</td>
</tr>
<tr>
<td>$N = 8, B = 7$</td>
<td>16.7</td>
<td>18</td>
</tr>
<tr>
<td>$N = 8, B = 8$</td>
<td>30.5</td>
<td>32.8</td>
</tr>
<tr>
<td>$N = 25, B = 7$</td>
<td>23.8</td>
<td>25</td>
</tr>
<tr>
<td>$N = 150, B = 7$</td>
<td>205</td>
<td>211</td>
</tr>
</tbody>
</table>

the first stage, 14.54% for the second stage, and 21.17% for the third stage.

3.2.4 Energy model

In this section, we verify the expression in Equation (2.10), which governs the energy consumed by the select generator in this design. This expression is parametrized by the length $N$ and the coefficient precision $B$, which will help in estimating the energy consumption of RSAC-based systems. A $V_{dd}$ of 0.675V is used in the select generator, which has minimal effects on the accuracy of RSAC compared to using 0.9V as supply voltage. Using circuit simulations at $V_{dd} = 0.675V$, we obtain:

$$
E_{\text{NOR},2} = 1.12 \times 10^{-16} J \\
E_{\text{MUX},2} = 1.12 \times 10^{-16} J \\
E_{\text{MUX},4} = 2 \times 10^{-16} J \\
E_{\text{MUX},8} = 3.75 \times 10^{-16} J \\
E_{\text{MUX},16} = 7.08 \times 10^{-16} J.
$$

We use linear fitting to obtain $E_{\text{MUX},N}$ for any $N$:

$$
E_{\text{MUX},N} = (4.25N + 3) \times 10^{-17} J.
$$

We compare the energy consumption of the select generators of different lengths and precision with the estimated energy consumption in Table 3.5. The values show an average overestimate of 6.2%. The energy consumption in RSAC is dominated by the select generator ($E_{\text{sel}} \gg E_{\text{sig}}$). This model will
Figure 3.15: SQNR vs. number of iterations for a RSAC-based DP kernel.

only be used in the next chapter to estimate the energy savings of a system of RSAC-based DPs.

3.2.5 Design optimization

Similarly to Section 3.1.5, the optimal curve is obtained by joining the minimizing points, shown in Fig. 3.15. SQNR improvements saturate around $TTR = 15$ due to the overall accuracy being dominated by the other noise factors described in Section 2.2.

3.2.6 Comparison to digital implementation

In this section, we will compare the RSAC-based DP against the digital implementation, for the same sampling period $T_S$. Thus, we need a more advanced model for digital, compared to the one in Section 3.1.6, that estimates both the delay and energy consumption. We use here an energy model and a delay model for the digital approach that is applicable for both superthreshold and subthreshold operations [14]. The dominant energy sources of a processing element are dynamic energy and leakage energy, expressed as:
Figure 3.16: The superthreshold current vs. $V_{dd}$ with $V_{gs} = V_{ds} = V_{dd}$.

Figure 3.17: The subthreshold current vs. $V_{dd}$ with $V_{gs} = V_{ds} = V_{dd}$.
Figure 3.18: The subthreshold current vs. $V_{dd}$ with $V_{gs} = 0$.

\[
E_{tot} = E_{dyn} + E_{lkg}
\]
\[
E_{dyn} = \alpha MC_{dyn}V_{dd}^2
\]
\[
E_{lkg} = \frac{MIO_{ff}V_{dd}}{f},
\]

where $\alpha$ is the switching activity factor, $M$ is the number of processing nodes each with capacitance $C_{dyn}$, $f$ is the operating frequency, $V_{dd}$ is the supply voltage, and $I_{Off}$ is the leakage current. The superthreshold current as a function of gate-to-source voltage is given by:

\[
I_{SUP}(V_{gs}) = \mu_n C_{ox} \frac{W}{L} (V_{gs} - V_t)^2,
\]

where $\mu_n$ is the electron mobility, $C_{ox}$ is oxide capacitance, and $V_t$ is the threshold voltage. The subthreshold current [15] as a function of gate-to-source and drain-to-source voltage is given by:

\[
I_{SUB}(V_{gs}, V_{ds}) = I_0 10^{\frac{V_{GS} - V_t - \gamma V_{ds}}{S V_T}} (1 - e^{\frac{V_{ds}}{V_T}}),
\]

where $I_0$ is the reference current and is proportional to the transistor $W/L$ ratio, $S$ is the swing factor, $\gamma$ is the DIBL coefficient, and $V_T$ is the thermal
voltage. Using Equations (3.3) and (3.4), the leakage current for an NMOS transistor is $I_{OFF} = I_{SUB}(0, V_{dd})$, while the switching current is:

$$I_{ON} = \begin{cases} I_{SUB}(V_{dd}, V_{dd}), & \text{if } V_{dd} < V_{t} \\ I_{SUP}(V_{dd}), & \text{otherwise.} \end{cases}$$ (3.5)

Figures 3.16, 3.17, and 3.18 show the plots of the fitted model with the output from HSPICE. We fit the parameters using least-squares fitting to obtain:

$$V_{t} = 0.35 \text{ V}$$
$$\gamma = 0.16 \text{ V}$$
$$S = 2.2$$
$$I_{0} = 56 \text{ nA}$$
$$\mu_{n}C_{ox}\frac{W}{L} = 1.16 \text{ mA}.$$  

Assuming the critical path of the processing element consists of $L$ processing nodes each with capacitance $C_{dyn}$, the operating frequency $f_S$ is given by:

$$f = \frac{1}{T_{S}} = \frac{I_{ON}}{\beta LC_{dyn}V_{dd}},$$ (3.6)

where $\beta$ is a fitting parameter needed due to finite rise and fall times. For the length 8 DP at hand, we have:

$$M = N \left( B(B+1) + B^{2} C_{and} \right) + \sum_{i=1}^{\log_{2}N} \frac{N}{2^{i}} (2B + i - 1)$$

$$L = BT_{sum} + BT_{cout} + T_{and} + T_{sum} + \sum_{i=1}^{\log_{2}N} (2B + i - 2)T_{cout},$$

where $C_{and}$ is the ratio between the capacitance of the full-adder block and that of the AND gate, which was found to be 0.25 from simulations. $T_{and}$ is the ratio between the switching period of the the full-adder and that of the AND gate, which was also found to be 0.25 from simulations. Similarly, $T_{sum}$ is obtained as 0.95 ($T_{cout} = 1$ is the reference switching time). Finally, $C_{dyn}$
Figure 3.19: Energy consumption vs. supply voltage.

Figure 3.20: Energy consumption vs. frequency of operation.
Figure 3.21: Energy consumptions vs. frequency of operation.

Figure 3.22: SQNR vs. frequency of operation.
was least-squares fitted and found to be 8.58 fF. Figures 3.19 and 3.20 show the energy consumption vs. $V_{dd}$ and energy consumption vs. frequency, respectively. We use these plots to compare with our RSAC-based DP of length 8. The results are shown in Fig. 3.21 and 3.22. For $SQNR \approx 8$ dB, 14 dB, and 20 dB, the energy savings are approximately $15.7 \times$, $4 \times$, and $2.1 \times$.

In Chapter 4, we will discuss the design of an emotion recognition system that uses only RSAC-based DPs.
CHAPTER 4
RSAC-BASED EMOTION RECOGNITION SYSTEM

In this chapter, we discuss how to implement an emotion recognition system using three cascaded RSAC-based DPs, shown in Figure 4.1. The input to this system is a frontal face and the objective is to classify this face between two emotions, \( \text{Emotion}_0 \) and \( \text{Emotion}_1 \), which could be one of any eight emotions: neutral, anger, contempt, disgust, fear, joy, sadness, surprise. The system in Figure 4.1 uses Gabor filters to extract features from the frontal face, picks the most important features using adaptive boosting (Adaboost), and classifies those features using a linear support vector machine (SVM). This system has three cascaded DP stages, \( J = 3 \), with the algorithmic work largely based on [16, 17]. The first and third stages use N-RSAC, while the second stage uses P-RSAC. This alternation between N-RSAC and P-RSAC is needed to make sure that the output of the current stage is a valid input for the latter stage. Computation in this system is done in stages:

- Compute in stage 1, select generators in stages 2 and 3 are clock-gated.
- Compute in stage 2, select generators in stages 1 and 3 are clock-gated.
- Compute in stage 3, select generators in stages 1 and 2 are clock-gated.

When the select generators in a stage are clock-gated, the output of each computation kernels is stored on its capacitor which is typically large enough to hold the value during the sampling period \( T_S \). This will be analyzed in Section 4.4.

The face inputs are extracted from the extended Cohn-Kanade dataset [18] using the MATLAB face detection algorithm, based on [19]. The faces are rescaled to \( 48 \times 48 \) pixels, with 8-bit precision for the grayscale values. Videos were recorded in analog S-video using a camera located directly in front of the subject. Subjects began each display with a neutral face and then performed a display which has been described and modeled by the experimenter. For
Emotions:
1. Neutral
2. Anger
3. Contempt
4. Disgust
5. Fear
6. Joy
7. Sadness
8. Surprise

Figure 4.1: An emotion recognition system that classifies the face between emotion 0 and emotion 1.

In our study, we selected 327 sequences, from 113 subjects, that were labeled as one of the seven basic emotions. The first frame of every sequence was also collected as a neutral face, totaling 654 faces. For the system at hand, of the 654 faces collected, only the ones exhibiting one of Emotion$_0$ or Emotion$_1$ will be used. We will use a leave-one-subject-out cross validation method due to the limited number of faces available. For a total of $x$ subjects per system, training will be conducted on $x-1$ subjects and testing on the remaining one subject. This procedure will be repeated $x$ times to obtain the probability of error, $P_e$. There are $\binom{8}{2} = 28$ possible systems and we obtain the average probability of error, $P_{e,avg}$ by averaging over all 28 $P_e$ values.

We note that RSAC has two potential drawbacks that need to be addressed before designing a system of cascaded RSAC-based DPs. First, RSAC, in its current implementation, can only handle positive numbers. Thus, we will devise a strategy used throughout this system to resolve this.

Assume we have two vectors: $\mathbf{A}$, consisting of only positive numbers, and $\mathbf{B}$, consisting of positive and negative numbers. Let $\mathbf{I}^+$ and $\mathbf{I}^-$ be the the index vectors that specify the indices of the positive numbers and negative numbers in $\mathbf{B}$, respectively. The vector of elements from $\mathbf{B}$ specified by the index vector $\mathbf{I}$ is denoted by $\mathbf{B}(\mathbf{I})$. With this notation, $\mathbf{B}(\mathbf{I}^+)$ is the vector consisting of the positive elements of $\mathbf{B}$. The DP, $\alpha$, of $\mathbf{A}$ and $\mathbf{B}$ can be
re-written as:

\[
\alpha \triangleq A \cdot B = A(I^+) \cdot B(I^+) + A(I^-) \cdot B(I^-) = A(I^+) \cdot B(I^+) - A(I^-) \cdot (-B(I^-)) \triangleq \alpha^+ - \alpha^-,
\]

where \(\alpha^+\) and \(\alpha^-\) are defined as \(A(I^+) \cdot B(I^+)\) and \(A(I^-) \cdot (-B(I^-))\), respectively. We note that both \(\alpha^+\) and \(\alpha^-\) consist of computing DPs of positive vectors, which is the target of this strategy. In this system, whenever applicable, we will compute \(\alpha^+\) and \(\alpha^-\) while delaying doing the subtraction until the end since DP operations are linear. We denote by \(\alpha^+\) the vector consisting of \(\alpha^+\)s and \(\alpha^-\) the vector consisting of \(\alpha^-\)s. The same strategy can be applied to \(\alpha^+\) and \(\alpha^-\) in the next stage, separately. In our system, we start off with a positive vector consisting of the image pixels. At the end of the third stage, we obtain eight outputs, of which four consist of results due to positive combinations and four consist of results due to negative combinations. We will see in Section 4.3 that the final decision is made by comparing the theoretical output to 0; and thus, the final decision in this system can be made by comparing the sum of the first four outputs to the sum of the second four outputs. This final decision is assumed to be done offline in our system and uses floating-point operations.

The second drawback to using RSAC in a cascaded system, is that instead of computing \(\alpha_i \triangleq A \cdot B_i\), we obtain \(\hat{\alpha}_i = \frac{G_n}{\|B_i\|_1} - \alpha_i - V_{tn}\), where \(G_n\) is the attenuation factor of N-RSAC and \(\|B_i\|_1\) is the \(\ell_1\) norm of \(B_i\). This is a linear transformation that is inversely proportional to \(\|B_i\|_1\), and thus non-constant. This will cause a problem in the next stage, where we want to compute \(\beta \triangleq \alpha \cdot C = \sum_{i=0}^{N} \alpha_i C_i\). Instead, we obtain:
\[
\hat{\beta} \triangleq \frac{G_p}{\|C\|_1} \hat{\alpha} \cdot C - V_t p
\]
\[
= \frac{G_p}{\|C\|_1} \sum_{i=0}^{N} \hat{\alpha}_i C_i - V_t p
\]
\[
= \frac{G_n G_p}{\|C\|_1} \sum_{i=0}^{N} \alpha_i C_i - V_t n G_p - V_t p.
\]
where \(G_p\) is the attenuation factor of P-RSAC. We can clearly see that the original DP \(\beta\) cannot be retrieved from the computed DP. We solve this problem by substituting the vector \(C\) with a vector \(\hat{C}\), whose elements are defined as \(\hat{C}_i = C_i \|B_i\|_1\). With this, we obtain:

\[
\hat{\beta} \triangleq \frac{G_p}{\|\hat{C}\|_1} \hat{\alpha} \cdot \hat{C} - V_t p
\]
\[
= \frac{G_p}{\|\hat{C}\|_1} \sum_{i=0}^{N} \hat{\alpha}_i \hat{C}_i - V_t p
\]
\[
= \frac{G_n G_p}{\|\hat{C}\|_1} \sum_{i=0}^{N} \alpha_i C_i \|B_i\|_1 - V_t n G_p - V_t p
\]
\[
= \frac{G_n G_p}{\|\hat{C}\|_1} \sum_{i=0}^{N} \alpha_i C_i - V_t n G_p - V_t p
\]
\[
= \frac{G_n G_p}{\|\hat{C}\|_1} \beta - V_t n G_p - V_t p.
\]

We now discuss the different algorithms used in this system.

4.1 Feature extraction using Gabor filtering

The feature extraction method used in this work is Gabor filtering. A Gabor filter [20] is a 2-D filter consisting of a complex sinusoid, known as the \textit{carrier}, and a 2-D Gaussian-shaped function, known as the \textit{envelope}. The Gabor filter is defined as:

\[
G(x, y) = \exp \left( -\frac{\hat{x}^2}{2S_x} - \frac{\hat{y}^2}{2S_y} \right) \exp (j2\pi f \hat{x}),
\]
where $S_x$ and $S_y$ are the variances along the $x$ and $y$-axes, respectively, $f$ is the spatial frequency of the sinusoid, and $\hat{x}$ and $\hat{y}$ are defined as:

$$
\hat{x} = x \cos(\theta) + y \sin(\theta) \\
\hat{y} = y \cos(\theta) - x \sin(\theta),
$$

where $\theta$ is the orientation of the Gabor filter. Typically, the dimension of the filter is chosen as $(2S + 1) \times (2S + 1)$, where $S \triangleq S_x = S_y$. In this system, we chose $S = 3$ to obtain a $7 \times 7$ complex filter, of which 25 coefficients are positive and 24 are negative. To obtain useful features from the images, a bank of Gabor filters at eight orientations and five spatial frequencies [21] were used:

$$
f = \{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\} \\
\theta = \left\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\right\}.
$$

Typically, the frontal face is convolved with the filter and every complex sample is merged into one real sample by taking its magnitude. However, in this system, we treat every complex sample as two distinct samples in order to avoid computing the magnitude, which is typically computationally heavy.

The total amount of features obtained from this step is $48 \times 48 \times 5 \times 8 \times 2 = 184,320$. This number is very large and needs to be reduced in order to have a feasible implementation of this system.

4.2 Feature selection using Adaboost

As previously mentioned, a total number of 184,320 features is very large, and thus, we need a feature selection mechanism. In this thesis, we use Adaboost to perform feature selection. Other techniques, such as principal component analysis (PCA), were compared to Adaboost in [22] and the latter was found to be the best option for this system. The feature selection is done offline and only the relevant features are computed in the feature extraction stage. In feature selection by Adaboost, every feature from the Gabor filters
Adaboost selects the best of those classifiers based on a threshold scheme, and then boosts the weights on the errors exhibited by using the selected feature. This procedure is repeated to obtain the best 150 features for every emotion by classifying it against all others. Thus, for every system, we only use 300 features from the Gabor filter and those will be classified using a linear support vector machine, discussed in Section 4.3. Depending on the two emotions chosen, we will have a different set of features and thus, different $H_1$ and different $I_{ih}$ values. However, we know that $\sum_{h=1}^{H_1} I_{ih} = 600$, which is the total number of DPs and total number of outputs, of which half consists of the positive samples and the other half consists of the negative samples. The lengths of these DPs is $\frac{49}{2}$ on average. The implementation using the digital approach has 300 DPs in total, each of length $N = 49$.

### 4.3 Support vector machine (SVM)

We use a linear SVM in this system to classify the features between the two different emotions. As a brief review on SVM, assume that $\{X_i\}_{i=1}^{n}$ are the features, with $X_i \in \mathbb{R}^D$. Accordingly, we have $\{t_i\}_{i=1}^{n}$, which are the target classes, with $t_i \in \{-1, 1\}$. The purpose of SVM is to find the hyperplane separating the two classes. The boundary will have an Equation $W \cdot \Phi(X_i) + b = 0$, where $W$ and $b$ are parameters to be determined and $\Phi(X)$ is some mapping, typically taking $X$ to a higher dimensional space. This is discussed below in more details.

The decision function $y(X)$ is $\text{sign}(W \cdot \Phi(X_i) + b)$. Let $X_1$ and $X_2$ be such that:

\[
W \cdot \Phi(X_1) + b = -1 \\
W \cdot \Phi(X_2) + b = +1,
\]

where the data has been re-scaled such that no points lie between -1 and 1; hence, $X_1$ and $X_2$ represent the boundaries of the two classes. We will see that these vectors play a very important in SVMs and are referred to as support vectors.

One can show that the distance between the two boundaries is $\frac{2}{\|W\|}$. We
would like to maximize this quantity which is equivalent to minimizing \( \|W\|^2 \). This leads to the following Quadratic Programming (QP) problem:

\[
\min_{w,b} \quad \frac{1}{2} \|W\|^2
\]
subject to \( t_i(W \cdot \Phi(X_i) + b) \geq 1 \).

This formulation assumes that the data is perfectly separable. A simple extension of this problem when the data is not perfectly separable is the soft-margin extension, where we introduce slack variables \( \epsilon_i \) for each \( X_i \). The QP problem becomes:

\[
\min_{W,b,\epsilon} \quad \frac{1}{2} \|W\|^2 + C \sum_{i=1}^{n} \epsilon_i
\]
subject to \( t_i(W \cdot \Phi(X_i) + b) \geq 1 - \epsilon_i \)
\( \epsilon_i \geq 0 \),

where \( C \) serves as a limiting quantity to the penalty factors for misclassified data.

One can show that the solution to the problem in (4.1) is:

\[
y(X) = \sum_{i=1}^{n} \alpha_i t_i (\Phi(X) \cdot \Phi(X_i)) + b,
\]

(4.2)

where \( \alpha_i \) are the Lagrange multipliers of the Lagrangian solution to problem (4.1) and \( b \) is a bias. We note that \( \alpha_i \) are 0 for all vectors except the ones on the boundary, the support vectors. Hence, the complexity of this algorithm depends largely on the number of support vectors, which we denote by \( L \).

We can rewrite Equation (4.2) as:

\[
y(X) = \sum_{i=1}^{L} \alpha_i t_i (\Phi(X) \cdot \Phi(X_i)) + b,
\]

(4.3)

where \( \{\alpha_i\}_{i=L+1}^{n} = 0 \) with a slight abuse of re-indexing. The decision is made by taking the sign of the result in Equation (4.3). In the emotion recognition system, a linear SVM is used, where \( \Phi(X) = X \). Hence, Equation (4.3) reduces to:
\[ y(X) = \sum_{i=1}^{L} \alpha_i t_i (X \cdot X_i) + b. \]

In the emotion recognition system, the second stage computes \( \beta_i \triangleq X \cdot X_i \). This stage has \( H_2 = 2L \) select generators, \( L \) for the positive coefficients and \( L \) for the negative coefficients. Each of those select generators feeds the two output chains from the previous stage; hence, \( I_{2h} = 2 \) \( (h = 1, \ldots, H_2) \). A total of \( 4L \) RSAC-based DPs are needed in this stage, for a total of four chains with \( L \) outputs each. The lengths of the DPs is \( N_{2h} = 150 \) \( (h = 1, \ldots, H_2) \), on average. The implementation using the digital approach requires a total of \( L \) DPs, each of length \( N = 300 \).

The third stage computes \( y(X) = \alpha \cdot \beta \), where \( \alpha = (\alpha_1 t_1, \alpha_2 t_2, \ldots, \alpha_L t_L, 1) \) and \( \beta = (\beta_1, \beta_2, \ldots, \beta_L, b) \). This stage has \( H_3 = 2 \) select generators, one for the positive coefficients and one for the negative coefficients. Each of those select generators feeds the four output chains from the previous stage; hence, \( I_{31} = I_{32} = 4 \). A total of eight RSAC-based DPs are needed in this stage. The lengths of these DPs is dictated by \( N_{31} + N_{32} = L + 1 \), with an average length of \( N_{31} = N_{32} = \frac{L + 1}{2} \). The implementation using the digital approach requires only 1 DP of length \( N = L + 1 \).

In Section 4.4, we will show the the effect of quantization on the probability of error when using a digital implementation. Then, we will show the results of using RSAC-based DPs and the energy savings.

### 4.4 Simulation results

The system in Figure 4.1 is implemented using a digital implementation with different precision, \( B \). In this implementation, the result of every DP is truncated back to \( B \) bits before going into the next stage. The last DP output is left unchanged. Figure 4.2 shows the plot of \( P_{e,avg} \) for different precision, compared to using floating-point operations. We can see that the performance saturates around 7 bits of precision. Figure 4.3 shows the energy consumption of the different digital implementations vs. \( f_S \) (the sampling frequency). The minimum energy operating point (MEOP) for these implementations is \( 1.0309 \times 10^{-9}, 1.5012 \times 10^{-9}, 2.0766 \times 10^{-9}, 2.7654 \times 10^{-9}, \) and \( 3.5758 \times 10^{-9} \) for bit precision of 4, 5, 6, 7, and 8, respectively.
Figure 4.2: Quantization effect of the digital implementation of the system in Figure 4.1 on the average probability of error (POE).

Figure 4.3: Energy consumption of the different digital implementations vs. sampling frequency.
The simulation setup for the RSAC-based system is the same as the one for Design II in Section 3.2. Unlike Design II, we will have a different $R$ and $C$ for every stage, denoted by $R_j$ and $C_j$, respectively. $R_j$ will remain fixed to 200 kΩ, while $C_j = c_j \times 200 \text{fF}$ and accordingly, $TTR_j = c_j \times 10$, where $c_1 = 1$, $c_2 = 3$, and $c_3 = 1$. The total number of iterations in stage 1 is $3 \times TTR_1$ and the total number of iterations in stage 2 is $3 \times TTR_2$. The precision of the select generator was chosen to be 7.

Figure 4.4 shows the plot of $P_{e,\text{avg}}$ vs. $f_S$ for the RSAC-based system and the digital implementations. For $P_{e,\text{avg}} = 0.23$ and 0.07, we obtain energy savings of 36% and 49% compared to the digital implementations’ MEOP, running at frequencies of 1.87 MHz and 1.7 MHz, respectively. For $P_{e,\text{avg}} = 0.23$, we can operate RSAC with a select generator with a precision of 6 and thus, increase the energy savings to 45% while not affecting $P_{e,\text{avg}}$. 
CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we presented RSAC, a new circuit architecture, that implements an energy-efficient DP. In Chapter 2, we analyzed the way RSAC operates, its architecture, and its select generation. We also developed a behavioral model for RSAC, and an energy model. In Chapter 3, we showed the energy savings of RSAC compared to digital implementations at the DP kernel level. In a 130 nm process, we showed $19 - to - 32 \times$ in energy savings compared to a digital implementation for SQNRs of $30 - to - 24$ dB, respectively. In a 28 nm FDSOI process, we showed $15.7 \times, 4 \times, 2.1 \times$ in energy savings compared to a digital implementation running at the same sampling frequency for SQNRs of 8 dB, 14 dB and 20 dB, respectively. In Chapter 4, we showed the design of an emotion recognition system composed solely of RSAC-based DPs. We obtained $2 \times$ energy savings, approximately, compared to digital implementations for $P_{e,avg} = 0.23$ and $P_{e,avg} = 0.07$. An extension to the system described in Chapter 4 is one where we need to distinguish one of seven emotions exhibited by a face. For this extension, one can use the emotion recognition system in Chapter 4 to compare every combination of two emotions, totaling 28 combinations, followed by a scoreboard to determine the most likely emotion. This is left as topic for future research. We note that RSAC is most powerful when the select generators are shared, which was not the case in the system studied. One way to improve the energy savings of using RSAC is by adopting another design for the select generator, digital or analog. This is a topic for future study. In particular, in the case of the emotion recognition system, we can improve the energy consumption by using the output of the Gabor filter for the select generation and the support vectors as inputs to the DP kernel. This will allow us to share the select generation for all support vectors but it also requires an analog select generation scheme and digital-to-analog converters for the support vectors. We also note that the cost of analog-to-digital conversions and digital-to-analog
conversions have all been ignored in this study and thus, a more reliable study would be to take them into account for the designed system.

In Chapter 4, we showed how we can split the processing chain to two chains, positive and negative, to accommodate the processing of negative values and thus, doubled the number of DPs needed. Extra energy savings can be achieved by working on a new RSAC implementation that accommodates negatives numbers without doubling the number of DPs. We also note that this technique would possibly not work if there’s a non-linear operation in the processing chain. Further, the emotion recognition system consisted of only three RSAC-based DP stages, but having more stages could be detrimental for RSAC since the dynamic range shrinks with every stage. This should be considered when designing systems with more stages.

Finally, we can improve the sampling period by pipelining, which requires an additional sample-and-hold (SAH) circuit at the output of each RSAC-based DP. This is also left as a future direction.
REFERENCES


