Measuring the power efficiency of subthreshold FPGAs for implementing portable biomedical applications

Shahin Sanayei Lotfabadi, Andy Ye *, Sridhar Krishnan

Department of Electrical and Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, Ontario, Canada M5B 2K3

ARTICLE INFO

ABSTRACT

Power is a significant design constraint for implementing efficient portable biomedical applications. Operating transistors in the subthreshold region can significantly reduce power consumption; it, however, also reduces performance. While this performance reduction can be significant in many applications, the low frequency nature of biosignals makes subthreshold region a good candidate for implementing biomedical applications. In this work, we investigate the feasibility of designing a specialized FPGA for implementing portable biomedical applications. In particular, we perform a case study on the performance of the Burg algorithm, a widely used biomedical signal processing algorithm, to determine the minimum operating frequency required for the processing of biosignals in real time. Based on the requirement, the trade-off between power consumption and performance is measured for FPGA routing resources operating in the subthreshold region. It is found that operating FPGA routing resources in the subthreshold region can significantly reduce power consumption while allowing the Burg algorithm to operate in real time. For the 32 nm Predictive Technology Model studied in this work, we observed a power reduction of 197.7 times (which corresponds to a power-delay-product reduction of 10.78 times) for operating FPGA routing tracks in the subthreshold region under a supply voltage of 0.4 V. Under this voltage, the FPGA can operate at 2.0 MHz while allowing signals to propagate unregistered through 45 routing tracks. Furthermore, the 2.0 MHz operating frequency meets the real-time requirement of the Burg algorithm for processing 20,000 samples per second.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Digital signal processing has played a significant role in the diagnostic and research activities in the health care field. Field Programmable Gate Arrays (FPGAs), on the other hand, are an important platform for implementing a variety of digital applications due to their short time to market, re-programmability and low non-recurring engineering costs. While FPGAs have been successfully used in many applications including digital signal processing, aerospace, medical imaging, computer vision, speech recognition, and ASIC prototyping, they have not been widely used in portable applications.

FPGAs are not widely used in portable applications primarily due to their significant power consumption. In particular, previous studies [1,2] have shown that applications implemented on FPGAs can consume significantly more power than the same applications implemented on ASICs. Portable applications, however, demand long battery life and consequently require an ultra low power implementation platform.

One way to extend the battery life of portable applications is to design digital systems that operate in the subthreshold region. Previous work has shown that, by operating in the subthreshold region, digital circuits often achieve minimum power-delay product thus minimizing their overall energy consumption [3]. Operating circuits in the subthreshold region, however, does significantly reduce their performance. This reduction in performance limits the applicability of subthreshold design in many applications.

Performance reduction, however, has significantly less impact on biomedical applications due to the low frequency nature of biosignals. In this research, we investigate the feasibility of designing a specialized FPGA that operates in the subthreshold region in order to reduce the power consumption of biomedical applications on FPGAs while still maintaining real-time signal processing capabilities.

Our study is based on a case study of the Burg algorithm [4], a widely used signal processing algorithm in biomedical applications. We first implemented a scalable RTL implementation of the Burg algorithm targeting Autoregressive (AR) modeling applications [5,6]. Based on the design, the maximum operating frequency that guaranties the real-time processing of biosignals is calculated.

* Corresponding author.
  E-mail addresses: shahin.lotfabadi@ryerson.ca (S.S. Lotfabadi), aye@ee.ryerson.ca (A. Ye), krishnan@ee.ryerson.ca (S. Krishnan).

This maximum operating frequency is then used as the performance constraint in the design of FPGA routing resources that operate in the subthreshold region. The power efficiency of the subthreshold design is then measured by determining the minimum supply voltage that is required to meet the real time performance constraint. These performance and power consumption figures are compared to the performance and power consumption of the conventional FPGA routing resources to quantify power efficiency.

The rest of this paper is organized as follows. Section 2 reviews previous work, Section 3 motivates the idea of using FPGAs as a reliable platform for implementing biomedical applications by describing the implementation of AR modeling using Burg algorithm. Section 4 presents the advantages of subthreshold design in reducing power consumption. Section 5 describes the architecture of FPGA routing resources investigated in this work. Section 6 presents the experimental results, and Section 6 gives concluding remarks.

2. Related work

Subthreshold circuit design for low power application has been investigated previously. It has been shown in [7] that operating in the subthreshold region minimizes energy per operation and that the minimum energy-delay product occurs at supply voltage of 3 V. In [8] an analysis of operating both CMOS and pseudo NMOS logic families in the subthreshold region and a comparison with normal operation in strong inversion region is presented. The results of this research reveal operating in subthreshold region reduces the energy per switching of an 8×8 carry save array multiplier by a factor of two. In [9] different leakage components have been modeled to estimate and reduce the leakage power for low-power applications.

There also have been a number of studies to reduce the power consumption of the interconnect fabric on the FPGAs. Some of these studies are focused on the subthreshold design and body biasing techniques. In [10] a subthreshold FPGA that uses a low-swing dual-VDD global interconnect fabric was implemented on a 90 nm device. This implementation resulted in $4.7\times$ energy reduction and $14\times$ improvement in speed as compared to a subthreshold FPGA using conventional interconnect. The energy reduction for this implementation was $22\times$ as compared to an FPGA with transistors operating in saturation region. In [11] a stack of half-width transistors in conjunction with adaptive body biasing has been proposed to reduce the leakage power for a switch block and a switch matrix on an FPGA. In [12] body biasing technique is used at a coarse grained architecture level to reduce leakage power and a clock skew scheduling scheme offered to improve performance. It required 3.35% area overhead to implement clock skew control blocks which are placed at every configurable logic block (CLB). None of the above studies, however, has investigated the performance requirement of biomedical applications on FPGAs.

3. Implementation of ar burg algorithm

Biomedical signal processing involves the collection and analysis of signals generated by human and other living organisms for medical diagnosis purposes. In portable applications, these signals often need to be processed in real time. For example, Autoregressive (AR) modeling [4–6] is a widely used feature extraction method that is used in biomedical applications to extract key diagnostic features from a range of biomedical signals such as knee joint vibroarthrographic (VAG) signals [5], pathological voice signals [5] and polysomnographic sleep data.

Extracting key features from signals also reduces the amount of memory that is required to store these signals. For portable applications, AR modeling performed in real time can be used to reduce their memory requirements. In particular, instead of directly storing VAG signals from the output of an Analog to Digital Converter (ADC), AR modeling can be used to extracts key features, called Autoregressive (AR) parameters, from the digitized version of these signals and stores only the AR parameters [5,13] in memory for later diagnostic use [5].

In particular, with 16 kHz sampling rate, 16 gigabits of memory is required to directly store an uncompressed stream of biomedical signals (for an ADC with 12-bit output) for 24 h. Using a 32-stage AR model, the required memory can be reduced by a factor of 184 to around 90 megabits. This reduction in memory can be extremely important in the design of portable medical devices for monitoring patient activities for an extended period of time.

Burg Algorithm is one of the most widely used algorithms for computing AR parameters and the advantages of using the algorithm over other methods are discussed in detail in [14,15]. The algorithm computers AR parameters based on minimizing the least squares of the forward and backward prediction errors using the lattice structure as shown in Fig. 1.

In particular, the cost function for minimizing the forward and backward prediction errors is given by:

$$\xi_m = \sum_N^{n=m+1} f_m^2(n) + b_m^2(n) \tag{1}$$

where m is the order of the AR model, $f_m(n)$ and $b_m(n)$ are the forward and backward predication errors respectively, and N is the length of the input data. As shown in Fig. 1, the forward and backward prediction errors are recursively updated based on the following equations:

$$b_m(n) = b_{m-1}(n-1) - \gamma_m f_{m-1}(n) \tag{2}$$

$$f_1 m(n) = f_1(m-1)(n) - \gamma_1 m b_1(m-1)(n-1) \tag{3}$$

where γ_m is the reflection coefficient of the algorithm. It is defined as:

$$\gamma_m = 2 \frac{\sum_{n=m}^{N-1} f_{m-1}(n) b_{m-1}(n-1)}{\sum_{n=m}^{N-1} f_{m-1}(n) b_{m-1}(n-1)} \tag{4}$$

Knowing the reflection coefficient, the AR model parameters for iteration k can be computed by following relationship:

$$a_{m,k} = a_{m-1,k} - \gamma_m a_{m-1}, m - k \tag{5}$$

Note that the value for the initial AR parameter $a_{0,0}$ is always equal to 1, and the $a_{k,n}$ parameter can be calculated using the following matrix operation:
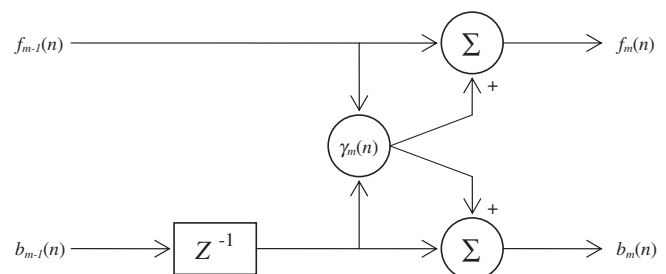

Fig. 1. The lattice structure of the recursion equations for forward and backward prediction errors.