Embedded CMOS Basecalling for Nanopore DNA Sequencing

Chengjie Wang

A thesis submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements
for the degree of

Master of Engineering

Graduate Programme in Computer Science and Engineering
York University
Toronto, Ontario
December, 2016

Abstract

DNA sequencing is undergoing a profound evolution into a mobile technology. Unfortunately the effort needed to process the data emerging from this new sequencing technology requires a compute power only available to traditional desktop or cloud-based machines. To empower the full potential of portable DNA solutions a means of efficiently carrying out their computing needs in an embedded format will certainly be required. This thesis presents the design of a custom fixed-point VLSI hardware implementation of an HMM-based multi-channel DNA sequence processor. A 4096 state (6-mer nanopore sensor) basecalling architecture is designed in a 32-nm CMOS technology with the ability to process 1 million DNA base pairs per second per channel. Over a 100 mm^2 silicon footprint the design could process the equivalent of one human genome every 30 seconds at a power consumption of around 5 W.

Acknowledgements

I would like to thank my supervisor Prof. Sebastian Magierowski of the Lassonde School of Engineering at York University. He took me into the palace of graduate studies inspiring me a lot from the very beginning. I always feel deeply grateful that he has been patient all the time and taught me how to tackle the problems I had with endless supports. He was always willing to help whenever I needed. Thanks Prof. Sebastian for all your supports and suggestions that mean a lot in my life.

I also would like to express my gratitude to my co-supervisor Prof. Ebrahim Ghafar-Zadeh of the Lassonde School of Engineering at York University. Without his passionate and patient help, I could not be able to clearly detail certain foundational knowledge of the DNA sequencing nanopore basecaller.

I would also like to acknowledge my colleague Yiyun Huang who is a digital standard cell designer working at TSMC presently. I am gratefully indebted to

his very valuable comments and suggestions on my work. Thank him for sharing a lot of time with me giving different visions and perspectives of the work. To my good friend, Mingbin Xu who is natural language scientist at Apple, I am extremely grateful for his help and suggestions of optimizing algorithms. A hearty thanks to all my friends for understanding me and supporting me.

Finally, I must express my very profound gratitude to my parents and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you!

Table of Contents

Abbreviations

ASICs Application-Specific Integrated Circuits

A/D Analog-to-Digital

BC Basecaller

BER Base-Error Rate

BMG Branch Metric Generator

CMOS Complementary Metal-Oxide-Semiconductor

COP Core Operations

COPS Core-Operations-Per-Second

DC Direct Current

DNA Deoxyribose Nucleic Acid

ED Event Detector

EDA Electronic-Design-Automation

FIFO First-In-First-Out

FPGAs Field-Programmable-Gate-Arrays

GPU Graphics-Processing-Unit

HMM Hidden Markov Model

HW Hardware

IP Intellectual Property

IT Information Technology

LPF Low-Pass Filter

MCBC Multi-Channel Basecaller

NGS Next-Generation Sequencers

NP Node-Parallel

ONT Oxford Nanopore Technologies

PCB Printed Circuit Board

SCBC Single-Channel Basecaller

SNR Signal-to-Noise-Ratio

SoC System-on-Chip

TIA Transimpedance Pre-Amplifier

UP Uni-Processor

VA Viterbi Algorithm

VLSI Very-Large-Scale-Integration

ZMW Zero-Mode Waveguide

Chapter 1

Introduction

1.1 Motivation

The process of *sequencing* in the context of genetics refers to the act of determining the *primary structure* of a linear biomolecule, that is the sequence of molecular sub-components of which a large molecule is comprised. Very commonly, this biomolecule is the deoxyribose nucleic acid (DNA), the familiar double-helix molecular arrangement and so-called "blueprint of life" found in the cells of terrestrial organisms. In the case of DNA, the primary structure is stipulated by the sequence of its monomeric components, the nucleotides: Adenine (A), Cytosine (C), Guanine (G), Thymine (T). A simplified diagram of DNA as well as its message carrying alternate, RNA, along with their nucleotide constituents is shown in Fig. 1.1.

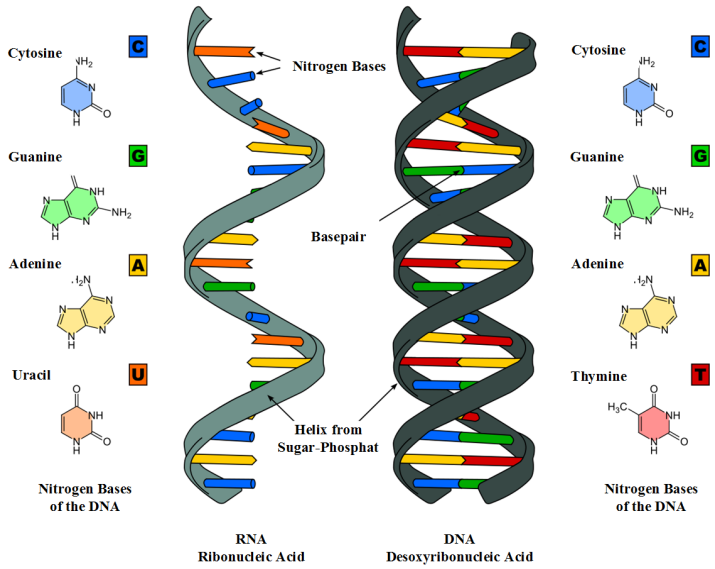

Figure 1.1: Comparison of a single-stranded RNA and a double-stranded DNA with their corresponding nucleobases. Image credit: Chemical Structures of Nucleobases ©Roland Mattern, CC BY-SA 3.0

As may be expected, sequencing DNA in terms of its three billion monomeric constituents (in the case of the human genome) is extremely challenging. Since the dawn of first-generation DNA sequencers in the late 70s [1], DNA sequencing technology has been revolutionized and improved in many profound ways.

Presently, so-called third-generation DNA sequencers [2], emergent only over the last 2-3 years, have been able to measure a single DNA molecule and produce a real-time stream of data in proportion to those measurements. That is, a continuous electronic output signal related to the primary structure of the DNA is available as soon as the molecule interacts with the sensor. This is a truly

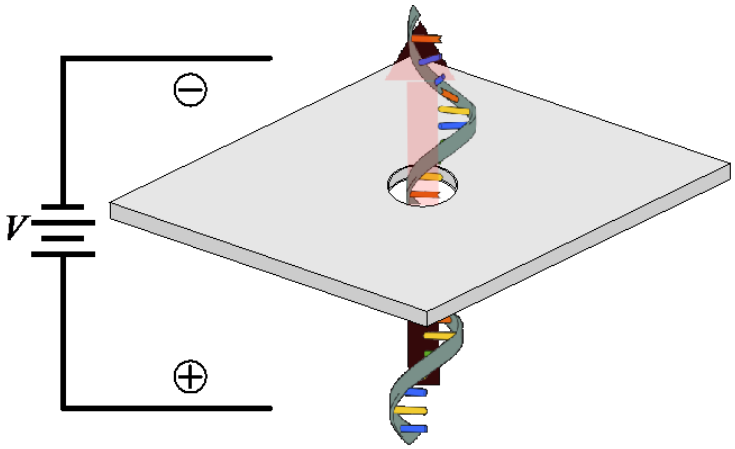

Figure 1.2: Oblique view of simplified nanopore structure.

profound development as it effectively endows molecular sensors with the ability to identify not only structure, but also the influence of dynamic, temporal, phenomena on that structure. Among these third-generation devices are molecular sensor arrays based on a nanometer-sized orifice called a nanopore sensor [3].

An abstract sketch of a nanopore sensor apparatus is given in Fig. 1.2. As shown, it is a small hole ($\approx$ 2 nm in diameter) created within some thin support ($\approx$ 5 nm in thickness) structure through which a single DNA molecule may be threaded by applying a voltage. As the molecule moves through the pore (i.e. as it *translocates*) it disrupts a pre-established direct current (DC) baseline signal, I_{dc}, resulting in a modulated time-series as shown in Fig. 1.3. This measured signal is proportional to the molecular make-up of the DNA, the aforementioned monomers: A, C, G, and T. A suitable feature-detection device, the *basecaller* of

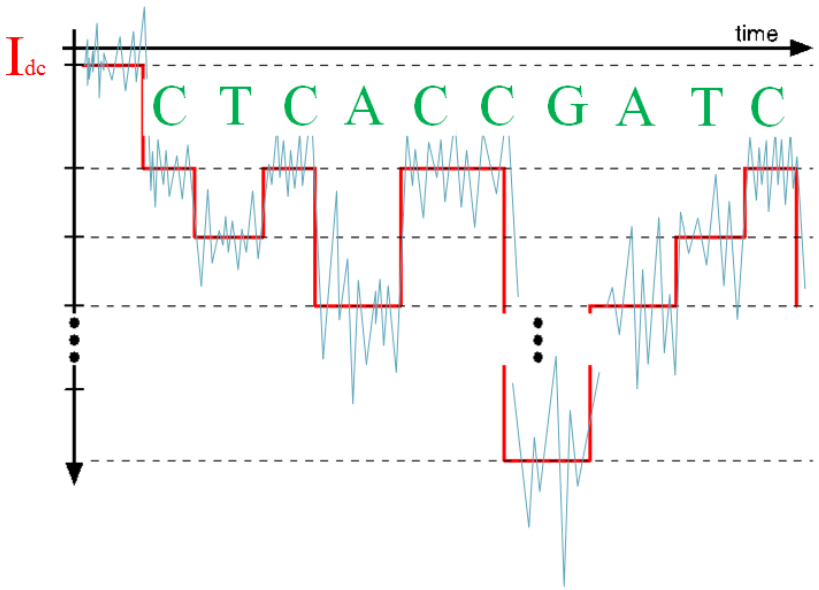

Figure 1.3: Example illustration of modulated current through a nanopore while the light blue trace denotes the baseline corrupted by noise.

concern in this thesis, is responsible for converting this time-series (the piecewise-constant signal in Fig. 1.3) to text label estimates of the DNA's molecular make-up, that is, its primary structure. This step, the basecalling step, is a critical part of the DNA sequencing process [4] and often directly follows a mixed-signal pre-processing stage as shown in Fig. 1.4.

Nanopore-based sensor arrays housed in palm-sized packages as shown in Fig. 1.5 have entered the market over the last two years with the ability to process DNA at rates in excess of 100 bp/s (basepairs-per-second) over 100s of channels for dozens of hours in-a-row leading to data generation in the 100s of GB per

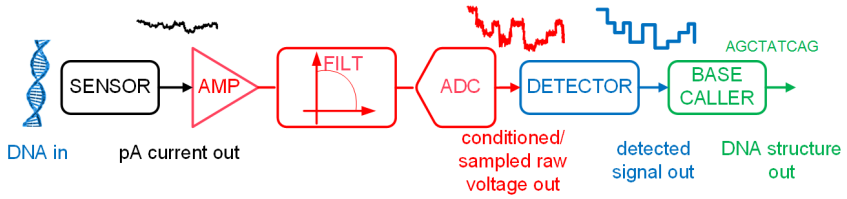

Figure 1.4: A block diagram sketch of the sensor, analog pre-processing, and basecaller.

Figure 1.5: A close up of the MinION. Image credit: Oxford Nanopore Technologies.

"modest" experiment. These numbers are astonishing compared to the metrics of their proof-of-concept nanopore predecessors reported 20 years ago [5]; original demonstrations of nanopore sensors consisted of a large desktop apparatus housing only one-sensor and lacked the ability to discern individual molecular constituents of the DNA molecule. As the technology improves, in no small part due to its close interface with CMOS (complementary metal-oxide-semiconductor) microelectronics technology, substantially higher performance can be expected.

At this time, the information processing load from such DNA measuring devices is handled by traditional desktop and cloud-based computers wherein critical processing blocks such as the aforementioned basecaller are implemented in code. However, the extremely compact physical dimensions of the new sensor platform call out for a similarly scaled compute resource. In other words, instead of realizing key features of a third-generation sequencer's data processors in commodity CPUs it is critical to devise custom embedded hardware versions of these blocks. Molecular sensing coupled with embedded measurement processing offers the possibility of significantly miniaturize DNA sequencing units and hence opens the door to an extremely broad range of application opportunities for DNA sequencing [6].

1.2 Approaches and Contributions

This thesis focuses on very-large-scale-integration (VLSI) implementations of the basecalling block in nanopore-based molecular sensors. The intention is that specialized hardware blocks of this function will be included as at least a part of complete embedded sequencing solutions for miniature, portable DNA sequencers.

Although as already mentioned this is just one part of a sophisticated DNA sequencing pipeline [7], the basecalling step needs to process large amounts of raw data. Thus, the basecalling step is especially advantageous for processing with a dedicated compute engine. This advantage is amplified if we achieve real-time basecalling functionality alongside the small form-factors already inherent to micro/nano technologies. Such a combination of size and speed in DNA sequencing machines should be particularly critical in promoting the vision of ubiquitous genomics [8] a scenario where the sequencing of biological molecules become commoditized and thus available to a broad range of the population. (e.g. real-time Ebola surveillance [9])

In this thesis I detail the potential of CMOS technology for a real-time embedded DNA basecaller for nanopore sensors. Key contributions of this research include:

1. Definition of a HMM-based algorithm for nanopore basecalling and its characterization for fixed-point implementation.

2. The translation of this algorithm into a custom hardware implementation (32-nm CMOS).

3. The expansion of the custom hardware basecaller implementation into a multi-channel sequence processor.

4. The characterization of the physical performance potential of basecaller ASICs over a broad design space.

1.3 Thesis Outline

The organization of this thesis is as follows:

In Chapter 2 the details of nanopore basecalling are discussed. Models for outlining the conversion of DNA features to electronic signals are described and the Viterbi algorithm used to convert these electronic signals to DNA bases explained.

In Chapter 3 the performance potential of the Viterbi-based basecaller is quantified in terms of fixed-point operation and its performance placed in context of other technologies.

In Chapter 4 a single-channel 32-nm CMOS ASIC implementation of the Viterbi-based basecaller is described, designed, and simulated.

In Chapter 5 the single-channel design is expanded to a multi-channel design and its performance as part of a multi-core realization predicted.

In Chapter 6 conclusions from the thesis research are given and future work discussed.

Chapter 2

Nanopore-Base Basecalling: Models and Algorithms

2.1 Sequencing and Basecalling: The State-of-the-Art

As mentioned in Chapter 1, in the nucleic acid sequencing context, the term *basecalling* refers to the process of converting physical measurements of molecules to a text prediction of their primary structure. Due to a myriad of technical limitations, no sequencing machine has yet managed to measure an entire genome directly, rather many short regions of DNA (very roughly, 1000 base-long regions) are physically sensed. Thus, basecalling only forms the initial primary structure prediction of these samples; basecalling results are then subjected to ensuing bioinformatics methods to reconstruct the genome being analyzed. These ensuing

sequencing procedures fall out of the scope of this thesis.

Besides the nanopore-based DNA measurement method outlined above, a number of other molecule measurement methods exist. These alternate methods are in fact the current market leaders in terms of sales of sequencing machines that employ them and represent the so-called next-generation sequencers (NGS). Most prominent today is the sequencing technology from the biotechnology company Illumina whose DNA measurements consist of optical pulses [10]. In fact, since the evolution of the original Sanger-based sequencing into a colour-fluorescence sensing technique [11], the use of optical methods to detect molecule feature have been dominant.

A relatively recent addition to sequencing methodology has been the charge-based approach introduced by Ion Torrent [12]. Rather than measuring molecules in terms of light, this method does so in terms of charge. The physical advantage of this approach relative to the optical methods rests largely with the simplicity and cost of the measured signal detector. In the case of optical measurement methods, a sophisticated apparatus must be installed to adequately pick-up the released light. In the case of charge-based method, a detector made out of standard CMOS technology as used in production microelectronic chip fabrication can be used. The benefits are a significantly reduced system size (roughly from a large

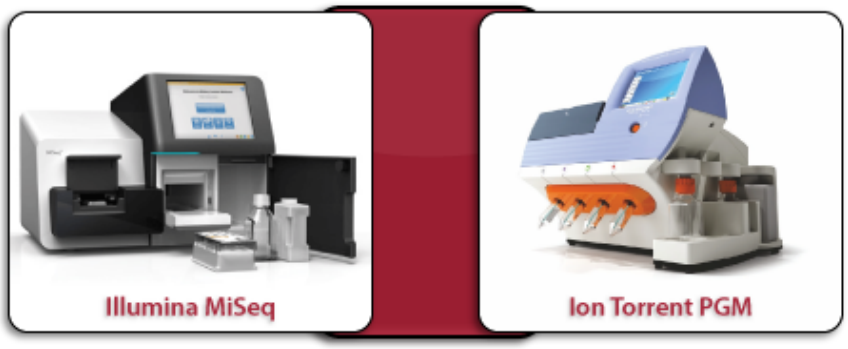

Figure 2.1: Representation of Both Illumina's MiSeq and Life Technologies' Ion Torrent Personal Genome Machine (PGM) platforms which support a broad range of applications and library construction protocols. Image credit: Retrogen Inc,. Platform Comparison. 2016. Web. 17 Oct. 2016.

table-top apparatus to a profile matching a desktop PC as shown in Fig. 2.1) and cost (roughly $10,000s to $100,000s). Other methods in the field include the pyrosequencing approach [13] and the zero-mode waveguide (ZMW) sensing method from Pacific Bioscience [14]. These approaches also leverage optical techniques as part of their sensing.

The rates at which NGS machines work are impressive compared to their Sanger-based predecessors. Where even relatively advances Sanger-based machines can produce measurements corresponding to 1-2M base pairs (bp) per day [15], core NGS machines like those from Illumina's HiSeq family can approach 1M bp/s. Nanopore machines just now entering the market (e.g. the PrometION from Oxford Nanopore Tech. — ONT) are capable of achieving throughputs on

this order as well. Also impressive are palm-sized sequencing units (e.g. the MinION also from ONT) which can achieve raw measurement throughputs (not to be confused with basecalling throughput) of roughly 25,000 bp/s. For context, such a throughput gives the ability to sequence a human genome at 1× coverage in three hours and 20 minutes. For a typical bacterial genome consisting of 1 Mbp, the 1× coverge can be completed in 4 seconds.

The diversity of available sequencing systems and the current state of the sequencing market have largely combined to prevent the emergence of any one basecalling standard. Unlike communication technologies like wireless and wireline communicators, hard-disk drives, etc. the signal channel in sequencing machines varies profoundly depending on the sensing modality and system employed (i.e. optics, charge, flow cell, laser, chemistry, etc.) and therefore requires a basecaller tuned to the sensor used. Further, given that the main market for sequencing machines is still the research community [16, 17], very often end-users still rely on (or prefer) custom-designed bioinformatics tools to process their signals, thus diversifying the basecalling field. The various approaches developed in disparate labs are typically shared and, besides the availability of proprietary solutions, have given rise to an active open-source community providing a multitude of options for sequencing, including basecalling.

Examples of basecalling algorithms include *phred* [18] for traditional Sanger-based methods (a very cost-effective and hence very common sequencing technique in many university research labs), *BayesCall* [19], *Ibis* [20] *AYB* [21] for Illumina reads. For nanopore basecalling there is the proprietary *Metrichor* [22] offered by ONT, and the recently released open-source approximation to Metrichor, *Nanocall* [23].

The majority of available basecalling software employs sophisticated machine learning techniques to achieve its objective. This approach presents a tremendous computational burden and thus requires substantial computing resources to complete in a timely manner. To the author's knowledge all these computations are carried out on commodity PC's, on either stand-alone desktops or in data centres. For more established sequencing methods, the impact of the computing delay is tempered somewhat by the intricate sample preparation process (i.e. the chemistry needed to prepare DNA for sequencing) and the sheer size of the sequencing equipment (which tends to promote centralized processing facilities where the need for extensive computing resources can be more easily accommodated). The scientific, research, nature of the work may also relax somewhat the need to minimize "time-to-data".

The research in this thesis is motivated by an anticipation that these cir-

cumstance will change as sequencer technology improves. The aforementioned MinION, a small, cheap, high-speed device capable of interfacing closely to information technology (IT) is an indication that this change may be imminent. Under these circumstances the prevailing emphasis on improving IT speed, power, cost, size, etc. should be stressed for future basecalling units as well.

2.2 Background on Nanopore-Based Molecular Sensing

Nanopore sensors have been extensively studied and discussed in the literature over the last two decades [24]. Building on this extensive history of work, ONT has fabricated an advanced nanopore sensor array and packaged it in a device called the MinION [25] pictured in Chapter 1. The MinION is a hand-held appliance capable of measuring the characteristics of a variety of biomolecules including DNA, RNA, proteins or small molecules in real-time.

The MinION weighs around 100 grams with a maximum dimension of 10 cm making it the first truly portable DNA measurement device. The device is also capable of processing its samples quickly (at present DNA translocation rates in excess of 200 nucleotide-per-second-per sensor are being achieved).

For example, using the MinION, Prof. Nick Loman, a professional in infectious diseases at the University of Birmingham, obtained enough bacterial genome

information to isolate the strain responsible for a Salmonella outbreak within 15 minutes of obtaining laboratory samples [26]. Another impressive demonstration of the MinION's potential was its application to real-time Ebola surveillance in Guinea [9].

The cross section of a typical nanopore is shown in Fig. 2.2. These are modified protein complexes that occur in nature (e.g. as virulence factors) and form openings with diameters on the order of 1-2 nm through which DNA can be threaded. Typically, a lipid or synthetic bilayer roughly 5-nm in thickness is used to support a pore above a pick-up electrode and readout apparatus and described shortly. As indicated in Fig. 2.2 DNA may translocate through the pore as a singled-sided strand.

A very active area of research in nanopore sensors is focused on the realization of solid-state nanopores [27, 28, 29]. Unlike the molecular (i.e. biological) nanopore pictured in Fig. 2.2 solid-state nanopores realize appropriately sized apertures in materials commonly found in standard semiconductor processes such as silicon-nitride. This form of nanopore sensor has undergone substantial advances over the last 15 years and is of interest given its more robust make-up and potential to benefit from established semiconductor techniques used in mass chip production. However, this type of nanopore sensor has yet to demonstrate

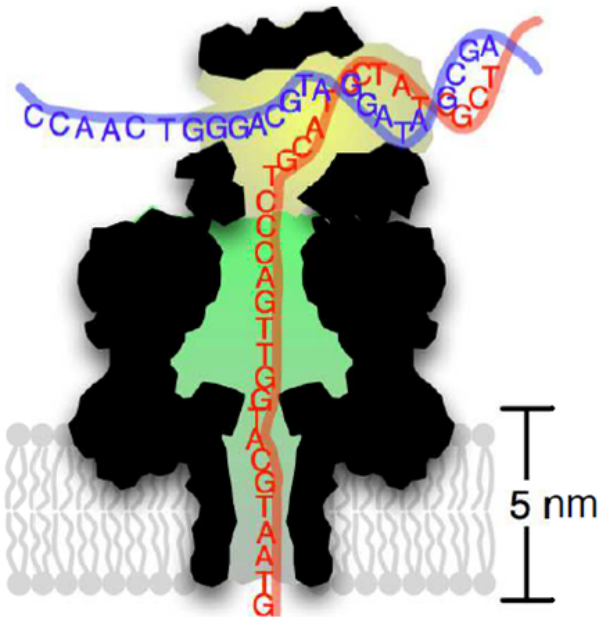

Figure 2.2: Representation of biological pore cross-section undergoing DNA translocation.

functionality of sufficient quality to support DNA sequencing.

During operation, the nanopore is immersed in a conductive fluid and a direct current (dc) voltage on the order of 100-mV is applied across the apparatus. This applied dc voltage causes a dc *baseline* current to flow through the sensor, the current consisting of the ionic charges in the conductive fluid. For typical sensors this dc current is on the order of 100 pA. The translocation of DNA through the nanopore causes fluctuations in the baseline current resulting in a modulated waveform of the type illustrated in Fig. 2.3. These minute ionic current fluctuations I_{signal}, which are about 50-pA peak-to-peak, assume a piecewise constant

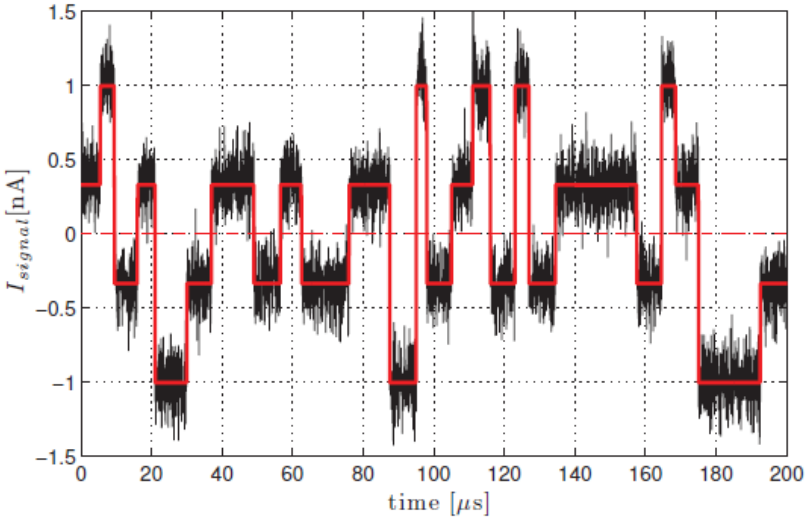

Figure 2.3: The character of the electronic signal from the nanopore before amplification.

(albeit corrupted by noise) profile versus time. Each plateau is associated with an *event* indicative of a discrete structural feature of the molecule segment currently in the pore. In the case of DNA we would expect these events to be indicative of the nucleotides that make up the molecule.

To process the small current signals available from the sensor an amplification and signal processing chain is needed as shown in Fig. 2.4. Typically these consist of an electrode capable of forming Ohmic contacts in an electrolyte, a transimpedance pre-amplifier (TIA), low-pass filter (LPF), analog-to-digital (A/D) converter and finally an event detector (ED). The TIA-LPF-A/D chain amplifies, conditions, and digitizes the raw signal while the ED predicts the event

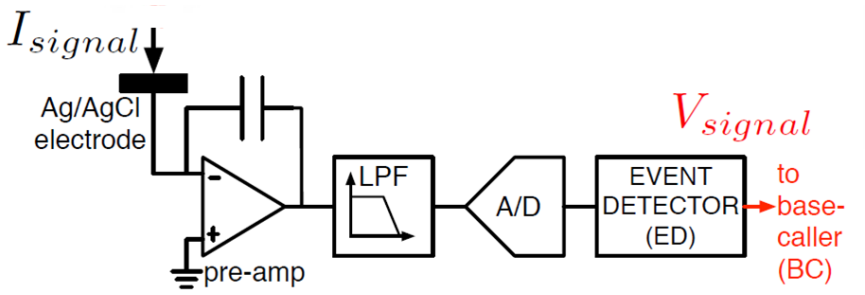

Figure 2.4: Generic mixed-signal CMOS amplification, filtering, and event detection responsible for converting I_{signal} to digital sequence of event values, V_{signal} to be processed by the basecaller.

levels contained therein in terms of voltage V_{signal} (consisting of a mean, standard deviation, start time, and duration). The output of this chain is suitable for processing by the basecalling engine (i.e. the basecaller).

In the majority of experimental nanopore studies the features noted in Fig. 2.4 have been accomplished with off-the-shelf technology, a set-up that encumbers the apparatus with significant parasitics and hence compromises the *event rate* (i.e. the maximum I_{signal} frequency) than can be accurately processed. Instead, employing co-packaged nanopore-CMOS TIAs has been shown to boost the workable event rate by orders of magnitude [30]; integrating the remaining functions noted above in silicon naturally follows [31]. As noted earlier, the integrated function of interest in this thesis is the basecaller (BC) that follows the ED.

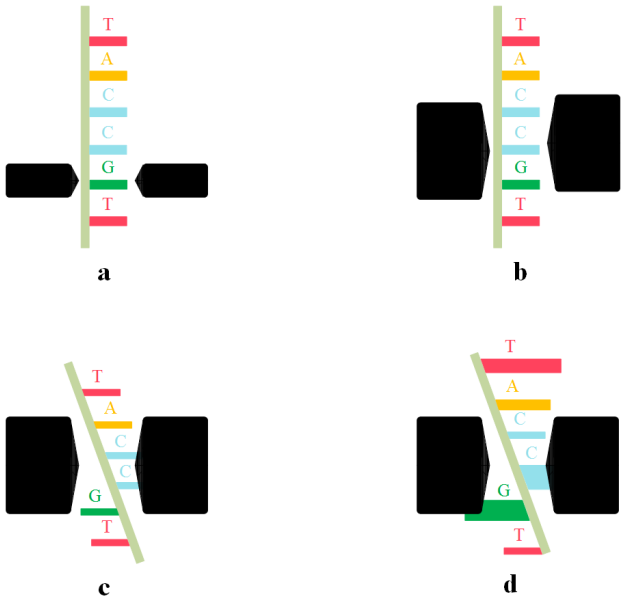

Figure 2.5: a) Abstraction of a nanopore responding to only one polymer unit (base). b) A nanopore responding to a 3-mer. c) A nanopore responding to a 4-mer translocating at an angle. d) A nanopore responding to an angled 3-mer with varying base features (e.g. methylation).

2.3 Realistic Nanopore Sensor Considerations

Ideally, a nanopore sensor used for DNA sequencing would be able to resolve each polymer unit (i.e. nucleotide or base) comprising the molecule. An abstract example of such a case is shown in Fig. 2.5a where the effective thickness of the nanopore is represented to be on the order of a single base. In theory the ramification of such a fine, single nucleotide, nanopore resolution would be a V_{signal} from the event detector exhibiting only four distinct event levels, one for each of the four unique bases of which DNA is composed.

In reality, nanopore sensors operating on complete DNA strands have not yet been engineered to discern individual bases. With roughly 0.5 nm between adjacent bases in a DNA strand and critical thickness around 2-nm it is clear that more than one base will be traversing critical regions of the sensor at any one time. Generally speaking then, the signal emerging from the nanopore will be related to some number, k, of bases (i.e. a k-mer) traversing the pore and its surroundings at any one moment rather than a single base.

An abstracted example of a pore processing a 3-mer is shown in Fig. 2.5b. A clear implication of multiple bases contributing to an output signal is the an increase in the number of observable event levels. Generally, if k bases contribute to the signal at any one time we may expect that 4^k unique event levels are present in the sensor's output signal.

Even more complications behind measured nanopore signals are possible, two of which are represented in Fig. 2.5c and Fig. 2.5d. In Fig. 2.5c a strand is shown translocating through the nanopore at some angle. It can be expected that such a fluctuation in the orientation of translocation will have an impact on the signal levels emerging from the pore. Another possibility, implied in Fig. 2.5d, involves fluctuations in the bases themselves. For example, the addition of molecules such as methyl groups in a process referred to as *methylation* [32, 33, 34] can be

expected to result in a difference between the signals measured for unmodified bases and their methylated counterparts.

2.4 Nanopore Basecalling: Basics and Limits

Ideally, the I_{signal} emerging from the nanopore sensor and consequently V_{signal} emerging from the ED would assume only four possible levels in relation to the four unique bases (A, C, G, T) that constitute DNA. In this case, the BC following the ED could conceivably be realized as a form of threshold detector. Such a thresholding design could be referenced to some model of the expected output voltage levels, that is, a training-based model associating event levels with bases in a one-to-one fashion. Endowing the ED with optimum filtering properties (e.g. matched filter) and the BC with optimum detection properties (e.g. maximum-likelihood) would then achieve the basecalling function sought.

As established by the discussion in Section 2.3 a 4-level signal from the ED is unlikely in realistic nanopore scenarios. As already noted above, more than one base (i.e. a k-mer) contributes to the electrical signal emerging from the nanopore sensor and the number of observed signal levels is exponentially proportional to this. For example, in one simulation study [35] of solid-state nanopores the pore was modelled as a system responsive to at least three bases at a time and hence

produced a $4^3 = 64$ level output. For the ONT sensor this value can reach levels of $4^5 = 1024$ and $4^6 = 4096$.

In theory, a simple thresholding BC could still be used for the k-mer scenarios. As long as each level is correctly identified by the thresholding unit, the BC could associate its prediction with the appropriate k-mer, say, in a 3-mer case, the base combination: ACT. Using the convention where the right-most letter denotes the most recent base to enter the pore, we expect that the following signal will correspond to one of CTA, CTC, CTG, or CTA 3-mers. Once a simple thresholding unit identifies the appropriate k-mer it is then a simple matter of comparing adjacent k-mer predictions to come up with the appropriate basecall. For example, in the example above, if a signal associated with the 3-mer ACT is followed by a signal associated with the the 3-mer CTG, the bascaller predicts the base A to have entered the nanopore. An example of this process is shown in Fig. 2.6.

In practice this means of level-by-level prediction is insufficient. The range of the underlying signal I_{signal} (roughly 50 pA as noted in Section 2.2) is simply too small to sustain 64+ levels while providing an adequate signal-to-noise-ratio (SNR) for a simplistic level-by-level thresholding scheme.

An alternate strategy, presents in many fields concerned with the general

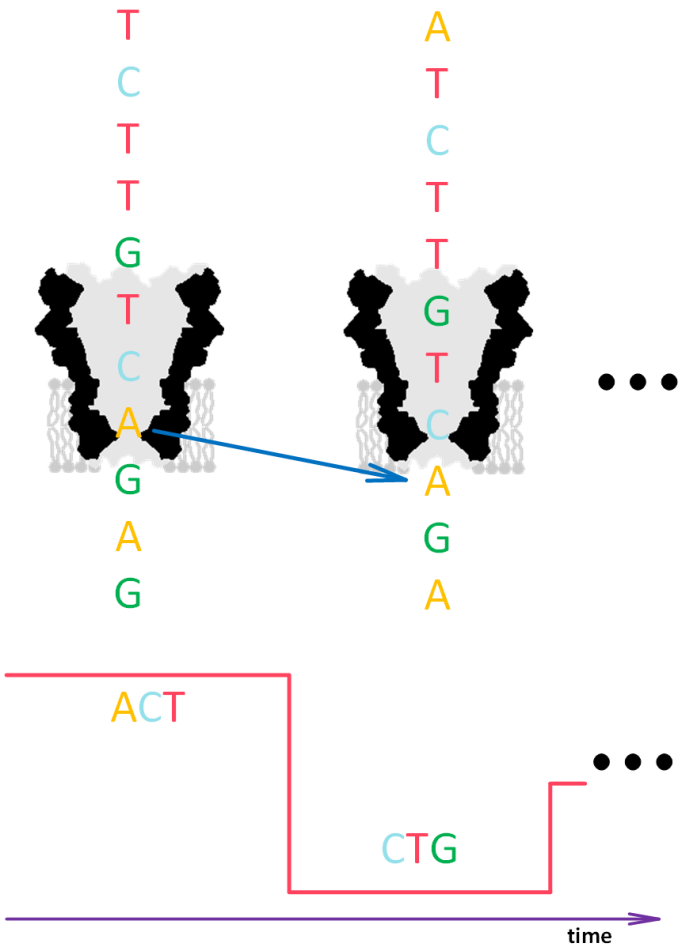

Figure 2.6: An example of GAGACTGTTCT(A) going through the pore and ACT directly in the pore, so a signal associated with the 3-mer ACT followed by a signal associated with the the 3-mer CTG. The basecaller would call an A.

problem of sequence labelling including communications, memory systems, speech processing, etc. is to seek the identification of bases not by looking at one signal at a time, but rather to do so by examining a sequence of multiple values. The definition, design, and CMOS hardware realization such a *sequence detection* scheme for the purpose of nanopore-based basecalling is explored in this thesis. The general strategy is detailed in the following sections.

2.5 Hidden Markov Model of Nanopore Basecalling

As already stated, the signal ultimately presented to the BC for analysis from the ED possesses the stepwise-constant characteristics sketched in Fig. 2.3 (i.e. the constant level approximated from the noisy waveform by the ED). The levels in the stepwise-constant curve constitute an *event* sequence $\{e_i\}_{i\in\mathbb{N}}$ indicative of the molecule passing through the nanopore.

As described above, for a nanopore sensor that responds to a k-mer we would in general expect the event levels e_i (i.e. a discrete-time abstraction of V_{signal}) to assume any one of 4^k different values. Ideally, these values would be drawn from some discrete set $\{\mu_j\}$ of expected levels where $j \in \{0, \ldots, 4^k - 1\}$. These expected values may be extracted via some preliminary sensor modelling or learned (or adapted) via some training method. The means of attaining the $\{\mu_j\}$ values

are not considered in this thesis.

Of course in practice the $\{e_i\}$ produced by the ED will have been corrupted by noise present in both the sensor and signal conditioning blocks. As a result, they cannot be expected to match exactly the expected values $\{\mu_j\}$. An appropriate detection strategy, the aforementioned sequence detection, must be used in the BC in order to decide which $\{\mu_j\}$ an $\{e_i\}$ is most likely derived from.

An intuitive means of describing a sequence detection strategy in the BC is via a picture that conveys all the possible signals that may emerge from the sensor. This is the trellis diagram sketched in Fig. 2.7. In essence it is an unrolled state-diagram denoting all the states that a system may exhibit over some time index i. Along the vertical dimension are shown all of the states which the nanopore sensor may assume at any time that an event e_i is sampled. In the context of basecalling the states merely refer to the k-mer present in the pore ranging in lexicographical order from AA...A to TT...T. As a result, the sensor system can take on any one of 4^k states. The horizontal dimension of the trellis denotes the progression of time, or more generally, samples of event levels taken over time.

The arrows (also referred to as *branches*) drawn joining the states in Fig. 2.7 denote which states at stage i may transition to states at stage $i + 1$. Over time, as the molecule translocates through the nanopore, the state of our sensor

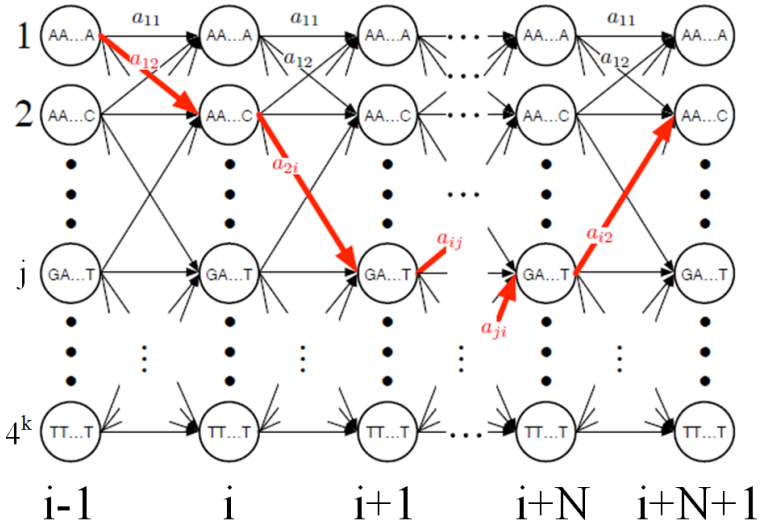

Figure 2.7: HMM trellis model of possible nanopore state progression over time index i. For clarity not all branch metrics labels are included.

effectively traverses the trellis diagram along some route (e.g. the one highlighted in Fig. 2.7) comprised of these transitions/branches.

The job of a BC sequence detector is then to process the sequence of some N event measurements (i.e. events from the ED) $\{e_i, \ldots, e_{i+N}\}$ and from this predict the route traversed by the sensor k-mer states through the trellis. As already mentioned, from these k-mer states, the actual bases may be derived (i.e. called). To emphasize the point, the sequence detector ultimately arrives at its decisions not by processing one event at a time, but rather a sequence of events. In essence it considers the joint probability of a multitude of events to extract

individual decisions.

Of course, a practical detector needs the trellis representation of the sensor's state progression to be informed by, and hence constrained, an appropriate model of the sensors behaviour. Importantly, this implies that only a subset of the 4^k possible transitions from stage i to $i+1$ be considered in the model. Clearly, allowing for all possibilities will impose an impractical computational burden on a detector seeking to find an optimum path through the trellis.

A basic transition constraint to impose is to allow only those transitions which link states whose *suffix* at stage i (i.e last $k-1$ letters as in CT of the state ACT) match the *prefix* of the state at stage $i+1$ (i.e. first $k-1$ letters as in CT of the state CTG). Such links denote the basic case where each event is indicative of a new base in the pore. Given that 4 such transitions are possible from any one state at i to the appropriate state at $i+1$ a total of 4^{k+1} transitions are possible from one stage to the next.

The identification of which transitions are possible and which transitions are not possible is an extreme version of weight assignment wherein some transitions are weighted to zero and hence removed entirely from the diagram and thus from consideration in the BC detector design. Typically more nuance is required and the weights assigned to the transitions are based on their probability of

occurrence. These weights are symbolized by $a_{jj'}[i]$ which are metrics related to the likelihood of a transition from state j at i to state j' at $i+1$. We detail the selection of the weights in Section 2.7. In summary the weights assigned to the branches determine the probability of that branch contributing to the final path identified by the BC sequence detector.

2.6 Basecalling Trellis Reduction

To the extent that the trellis discussed Section 2.5 informs BC detector hardware design (to be explored in the following chapters), our effort can be simplified by reducing the size of the trellis needed to describe our problem.

In particular only the 4^{k-1} need to be enumerated in the trellis since the remaining (most recent) base contribution to the full k-mer in the pore is essentially identified by the transition made from stage i to stage $i+1$. As a result the number of states is reduced by a factor of 4 as is the total number of branches between stages which is now 4^k.

This reduction is helpful, but still leaves us with a substantial problem to solve. In particular, over some discrete time-span N the BC observers N event levels from the nanopore via a space comprised of 4^k branches per stage. The number of complete paths over N event levels thus takes on a complexity measure

of $O(\exp(N \cdot k))$. As discussed next a dynamic programming [36] strategy is used to reduce this to a problem requiring the examination of $O(N \exp k)$ in search for the optimum path.

2.7 Viterbi Detection

A BC sequence-based decision over the HMM can be made in a statistically optimum fashion – in the maximum likelihood sense [37] – by employing the Viterbi algorithm (VA) [38]. As outlined in the preceding sections, this algorithm seeks to identify the path through the trellis over N time steps whose *path metric* Γ (i.e. the sum of its constituent branch metrics $a_{jj'}$) is a minimum. The VA achieves this in an iterative fashion by executing:

1. Calculating the $a_{jj'}$ values as some distance

$$a_{jj'}[i] = \|e_i, \mu(j, j')\| \tag{2.1}$$

   where e_i is the event level provided to the BC from the ED at time i and $\mu(j, j')$ is the expected event for a transition/branch form state j to j'.

2. Using these values to update the set of 4, *candidate* so-called *path metrics*, $\mathbf{\Gamma}_{j'}$ going into each possible state, j', of the 4^{k-1} states constituting a stage

at i

$$\mathbf{\Gamma}_{j'}[i] = \{\Gamma_j[i-1] + a_{jj'}[i] \mid j' \in A,\ |A| = 4, A \in \mathbb{Z}\}. \tag{2.2}$$

3. Culling the candidate paths at each state to arrive at a single *survivor* path: $\Gamma_{j'}[i] = \min(\mathbf{\Gamma}_{j'}[i])$ and noting the state j from which the last step constituting Γ'_j originated; the latter serving as a means of recording the states visited by the survivor paths retained at each state in stage $i+1$.

After N iterations over this three-step procedure a sequence with terminus state $out = \arg\min_j(\Gamma_j[N])$ is selected and the preceding $N-1$ states identified by referring to the associated values stored as part of step 3). The entire process is repeated to extract the next N-state sequence.

In the following chapter the performance prospects for this approach to base-calling are explored. These explorations take particular note of the hardware limitations that such an approach may face.

Chapter 3

VA Basecalling Performance

3.1 Metrics and System Considerations

From a purely bioinformatics perspective, the performance of a basecaller largely boils down to the accuracy with which it calls its bases. Since basecallers do not work on complete genomes, but rather on randomly obtained snippets of genomes (i.e. strands ranging from 10s to 1000s of bases depending on the sequencing machine used) they require a two-step process to evaluate their accuracy in a full experimental setting [39]. This involves first using an alignment algorithm to find an acceptable match between the called strand and a pre-existing *reference genome* of the target organism; for strands with acceptable alignment, the number of bases in error divided by the total number of bases in the strand defines the error-rate for that strand. An average error-rate can be obtained by averaging

over all the extracted (per-strand) error rates.

In practice, the error rate can be highly variable on, not only the method of alignment, but also on the measurement technique and even the location of the sequencing instrument from which measurements are extracted (e.g. lower quality results tend to occur at the fringes of the physical substrate to which DNA is affixed [21]). For basecallers operating on signals from Illumina sequencers error rates around 10^{-2} to 10^{-3} have been reported [39, 40]. For nanopore basecaller the reported rates are presently much worse, around 3 to 3.5 $\times 10^{-1}$ [23] with recent communications of rates around 10^{-1} [41].

A major reason for this difference between Illumina and nanopore accuracies likely originates from the physical signal-to-noise ratio (SNR) available to both systems. Illumina systems benefit from chemical amplification (the creation of 1000s of identical copies of the subject DNA effectively boosting the signal) and extremely slow operation (per-sensor base signals are released roughly one per half-hour, although this includes the need for cyclical chemical treatment between base signals). The high throughput of Illumina systems is realized by the engineering of an extremely large number of sensors (on the order of 1 billion) working in parallel.

Conversely, nanopore-based sensors work directly on one molecule, converting

this structure's translocation through the nanopore into a minuscule current. Of course, this direct measurement brings with it tremendous advantages for sequencing including the small form-factor to which nanopore machines conform. The small signal levels however do leave the device prone to inaccuracies from noise in the physical sensor apparatus as well as noise from the electronic circuitry intended to condition the raw signal to a digital format for BC processing.

We next explore the error rates achievable for a BC using the VA described in the previous chapter. During this analysis we consider a variety of parameter settings that may be encountered in realistic systems subject to hardware limitations.

3.2 VA-Based BC Simulation System

As alluded to above, the performance achievable for a VA-enabled BC is quantified in terms of the "base-error rate" (BER). The BER is a measure of the fraction of bases incorrectly predicted by the BC. The simulations used to obtain the BER consisting of a simplified representation of the nanopore detection system described earlier. In particular, they modelled the nanopore sensor as a filter with the transfer function shown in Fig. 3.1. This transfer function was reported in [35] and obtained from molecular dynamics simulations on a solid-

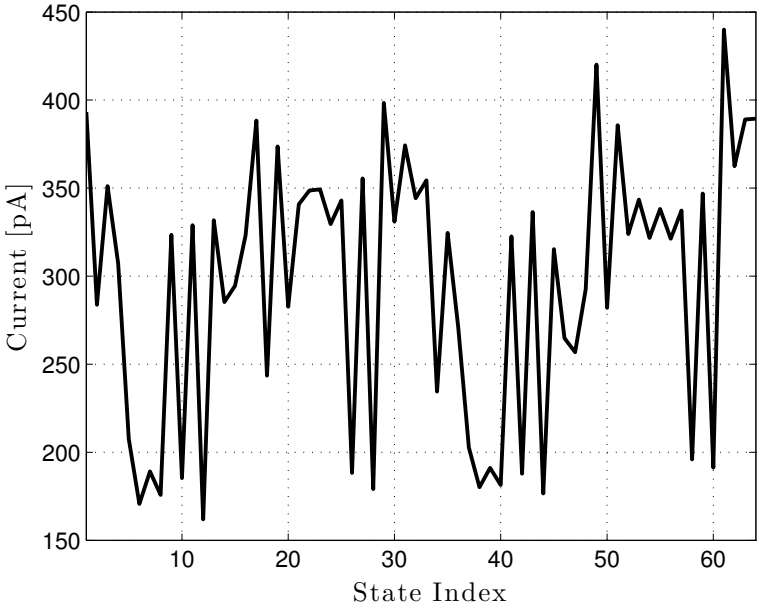

Figure 3.1: Electronic current transfer function of a solid-state nanopore as simulated in [35] vs. ordered 3-mer numerical state index.

state nanopore responsive to a 3-mer. For this reason the transfer function of Fig. 3.1 is shown on an ordinate consisting of 64 states, the states numbered from 0 (AAA) to 63 (TTT). A plot of this transfer function with electronic current output levels arranged in ascending order is shown in Fig. 3.2.

In the BER simulations below, nanopore models that process not only 3-mer molecules, but also, 4, 5, and 6-mer molecules are considered which naturally exhibit 256, 1024, and 4096 states, respectively. The transfer functions of these higher-order sensors are derived from the 64-state system of Fig. 3.1 by simply

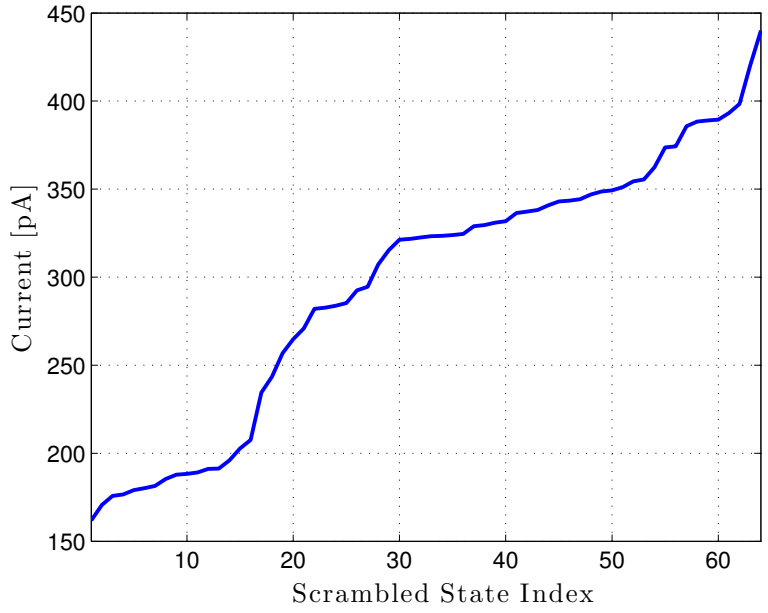

Figure 3.2: The nanopore transfer function of Fig. 3.1 with output current arranged in ascending order.

interpolating current levels between the original 64 states.

The input to the transfer function consists of a 4-alphabet text stream representing the 4 DNA base constituents (i.e. A's, C's, G's, and T's). This input is made uniformly distributed. When applied to the corresponding sensor transfer function, each input produces a corresponding event current output $i[k]$ (depending also on the L preceding mer's into the pore). To this signal, for each output, a random Gaussian noise value, $n[k]$ is added. Over the course of a simulation of

N events the average power of the input signal is approximated with

$$\sigma_i^2 = \frac{1}{N}\sum_k i^2[k] - \left(\frac{1}{N}\sum_k i[k]\right)^2 \tag{3.1}$$

while the noise power is approximated with

$$\sigma_n^2 = \frac{1}{N}\sum_k n^2[k]. \tag{3.2}$$

The signal-to-noise ratio (SNR) of such a simulation is then given by

$$\text{SNR} = \frac{\sigma_i^2}{\sigma_n^2}. \tag{3.3}$$

The net signal $x[k] = i[k]+n[k]$ is input to the VA-enabled BC which processes it and outputs its best prediction of the input text stream. The BER is then calculated as the ratio of incorrectly called bases to the total number of bases processed.

3.3 VA-Based BC Performance

In Fig. 3.3 the simulated BER as a function of the SNR of the signal presented to the BC is shown. A number of scenarios are considered in this picture; among these are the BER of VA BC's implemented using double-precision floating-point computation. There are the so called *ideal* curves in Fig. 3.3. As shown, 4 ideal scenarios are considered, one for BC's designed to process signals from 3-mer,

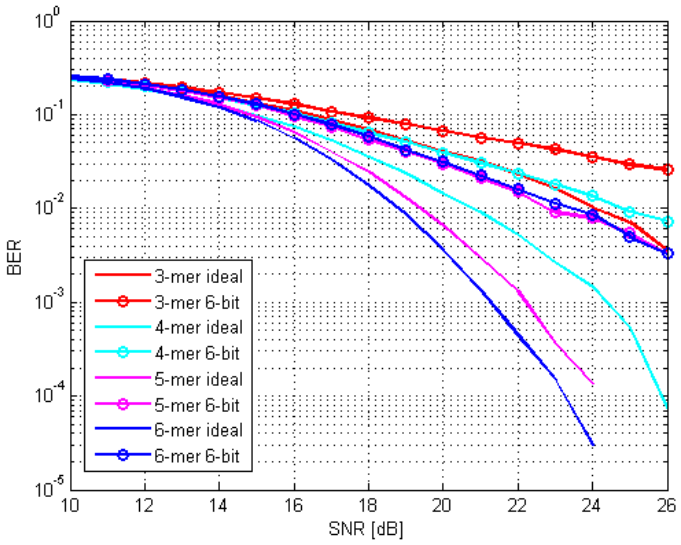

Figure 3.3: Base error rate (BER) as a function of SNR for 3-mer, 4-mer, 5-mer, and 6-mer sensor examples processed by a VA-enabled BC.

4-mer, 5-mer, and 6-mer sensors respectively. In a nod towards more simplified hardware realizations however, all the BER results presented in Fig. 3.3 employ a simple l_1-norm to calculate (2.1). In contrast to branch calculations based on the l_2-norm, this approach reduces hardware complexity.

Raw basecalling (i.e. before application of further bioinformatics) of good quality can attain BERs around 10^{-2}-10^{-3} [42] and in the ideal VA examples considered these are achieved at SNRs of roughly 19, 19.5, 21 and 24 dB for the 6, 5, 4, 3-mer sensors respectively. The exact results are of course heavily dependent on sensor specifics and these examples are meant to be illustrative.

In Fig. 3.3 it is apparent that for a given SNR the BER of the BC operating on 3-mers suffer's relative to the BC operating on 6-mers. This is a consequence of decreased redundancy; that is, to the extent that 6-mer signals are correlated over a longer sequence of event samples than 3-mer signals they better inform the sequence detection computations of the VA and thus result in more accurate base calls. These results however may be slightly deceiving to the extent that available signal ranges are finite and therefore as the number of event states increases the difference between any one transition shrinks.

For example, for a given SNR and a given signal range (i.e. peak-to-peak value) ΔI the transition from a 3-mer signal to a 6-mer signal involves an increase in the number of event states by a factor of $4^6/4^3 = 64$. It would seem that this may lead to a significant degradation in base calling performance, but the fact that only a subset (i.e. 4) of transitions are allowed between a state at i and its ensuing state at $i + 1$ coupled with the fact that VA decisions are made over a sequence of considerations blunts the impact of this. As the length of the mer processed however the advantage drops.

By the time the SNR drops to 15 dB the ideal 4, 5, and 6-mer performance is nearly identical with an error rate around 10%. The 3-mer system crosses this boundary at about 17 dB SNR. In a mature sequencing system where SNRs

typically exceed 100 (i.e. above 20 dB) are expected [43, 44] such performance levels would be very good (e.g. the 6-mer ideal BC simulates to an error rate of 0.4% at 20 dB). In present-day nanopore systems however the SNR currently hovers around 10 dB [45] where all solutions converge to a base calling accuracy of roughly 75%.

Besides the ideal BER results, Fig. 3.3 also shows the VA BC behaviour when calculations are limited to coarse fixed-point calculations. This obviously has benefits for hardware simplifications, but comes at the expense of less accurate base calls. This penalty is quite substantial for a 6-bit system as shown in Fig. 3.3 with an extra 2.5 dB SNR required by the 6-mer system to maintain a BER or 1%. Similarly, for a 20-dB SNR the 6-bit calculation achieves a 3% error rate compared to the ideal BER of 0.3% at that setting.

The character of the degradation from ideal to fixed-point calculations with limited bit depth is shown in Fig. 3.4. Therein are indicated BER improvements of 40-90% as the bit width is increased from 6 to 10 bits for 3, 4, 5, and 6-mer sensing systems and BC's operating on a signal with 20-dB SNR.

Under this SNR setting we see that even a 1-bit improvement from 6 to 7-bit calculations has a significant impact on the accuracy of the results. Ensuing increases clearly saturate by bit-depths of 10. For existing nanopore system with

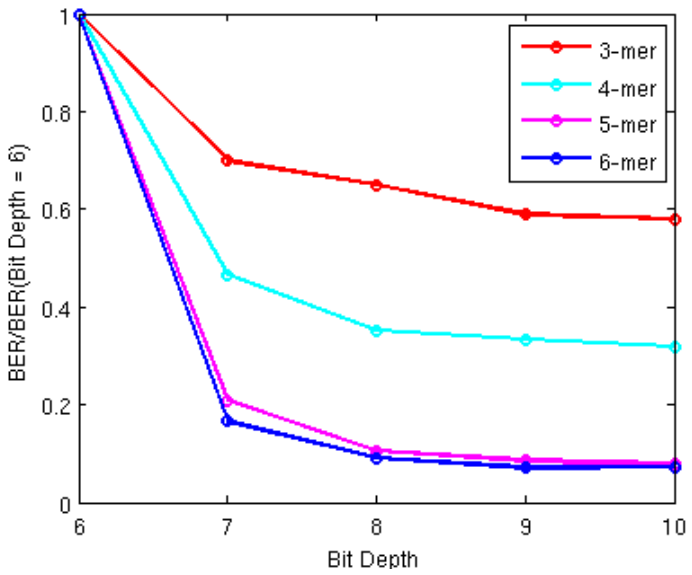

Figure 3.4: Relative BER at 20-dB SNR (referenced to BER at bit depth of 6) vs. bit depth for 6, 5, 4, and 3-mer sensors.

their relatively low SNRs, even a 6-bit implementation may not be out of the question. This is evident from the converged BER performance seen in Fig. 3.3 for SNR below roughly 14 dB.

An examination of the base error rate for the 6-bit VA BC as lower SNR, those between 5 and 10-dB is shown in Fig. 3.5. In this case it becomes clear that noise effects become dominant and only marginal differences exist between pores operating on different mer counts. From this picture it is also clear that minimal sacrifice is made by employing a 6-bit event word size with which to do the basecalling. Of the results shown, the biggest performance drop off between

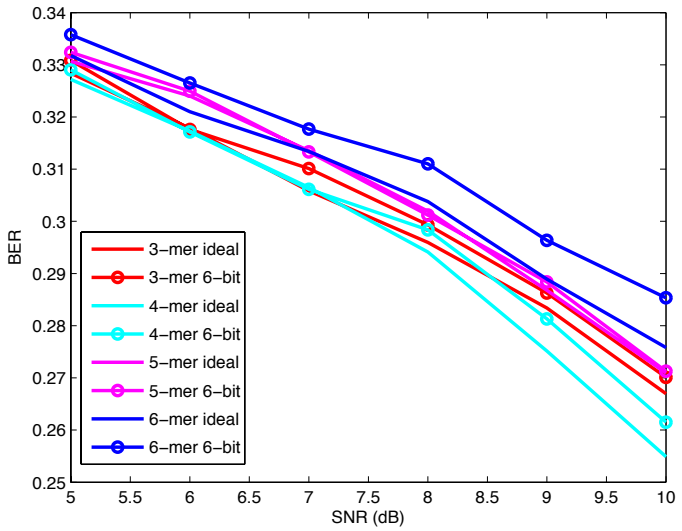

Figure 3.5: Base Error Rate for SNR below 10 dB.

the ideal (floating point) and the 6-bit implementation is that for the 6-mer sensor.

Chapter 4

Single-Channel Architectural and Physical Implementation

4.1 Viterbi Detection Hardware Context

As a means of detecting data based on a sequence of observations the Viterbi algorithm (VA) has found use in a multitude of IT applications. In practice it's role as a decoder of convolutional channel codes is probably its most prominent use in hardware [38, 46, 47]. In this context it has been employed in wireless cellular as well as deep-space communications [48, 49, 50, 49]. Another well-know VA application in hardware IT is in magnetic signal read-out, specifically hard-drive readers [51].

The data-rates at which these applications needed to operate require dedicated

hardware (HW) implementations of the VA. Ensuing improvements in microelectronic semiconductor technology have resulted in higher application data rates and thus largely maintained the realization of these algorithms in specialty chips (rather than migrating to commodity microprocessors). The designs achieved as part of this pursuit serve as inspiration for the VA BC HW researched in this thesis. However, it is the case that the application context in which the present work is done requires substantial modifications to the art. Some of the differences include the presence of potentially 1000s of states for the VA to keep track of in sequencing applications (rather than on the order of 10s for IT). The number of channels needed in sensing applications can also easily reach into the 1000s, orders of magnitude greater than typical IT needs. The nature of the sensing application may also allow the VA to operate with a much lower accuracy than needed by VA's in wireless or data-storage applications although high accuracy would never be a drawback in sensing. Per-channel speed may also be quite different, where IT applications can easily be required to handle data rates in the 100's or 1000's of megabits per second, sensor applications, specifically DNA sequencing applications currently operate at rates of roughly 100 times slower than this (but given their poor signal quality are usually faced with a heavier computational burden).

4.2 Pipelined Viterbi Architecture

To maximize data throughput VA HW typically relies on the well-known strategies of *data parallelism* (i.e. replication) [52] and *pipelined parallelism* as in the architecture example in [53]. The VA is particularly amenable to data parallelization, the simultaneous execution of identical operations on physically distinct, but computationally identical blocks, and this effort for the base calling operation is described in Section 4.3. The VA system design also stands to benefit from pipelining and the approach considered for this aspect is outlined presently.

In the pipelining strategy, a calculation is distilled into a sequence of sub-computations that communicate information only with the computational blocks logically adjacent to them. The ability to break-up a problem and distribute it along such a structure is a form of parallelism as each block can be simultaneously engaged in achieving a part of the total computation. Unlike data parallelism, a significant savings in physical resources can be realized with such a technique. The immediate drawback of pipelining is the latency incurred by the traversal of a computation from the input sub-computation to the final output sub-computation. For many streaming applications, real-time basecalling included, this is not a significant issue.

A proposed 7-stage pipeline for the BC VA is as follows (note that each single

activity does not mean it is executed in one clock):

1. Distance Generation (DG)

    - Calculate the branch metric (i.e. distance $a_{jj'}[i]$ in Ch. 2) for all transitions modelled in the trellis

2. Path Metric Update (UP)

    - Calculate all the new path $\mathbf{\Gamma}'_j[i] = \{\Gamma_j[i-1] + a_{jj'}[i]\}$

3. Path Metric and Pointer Selection (SE)

    - Identify minimum path: $\Gamma_{j'}[i] = \min(\mathbf{\Gamma}_{j'}[i])$
    - Identify pointer to preceding state: $\text{ptr}_{j'}[i] = \arg\min_j(\mathbf{\Gamma}_{j'}[i])$

4. Path Metric Referencing (RE)

    - Select minimum path metric over all states: $\Gamma[i] = \min_{\forall j}(\Gamma'_j[i])$
    - Store pointer to the minimum state $\text{ptr}[i] = \arg\min_{\forall j}(\Gamma'_j[i])$

5. Normalize Path Metrics and Traceback (NO)

    - Offset all path metrics by minimum: $\Gamma_{j'}[i] \leftarrow \Gamma_{j'}[i] - \Gamma[i]$

6. Traceback (TB)

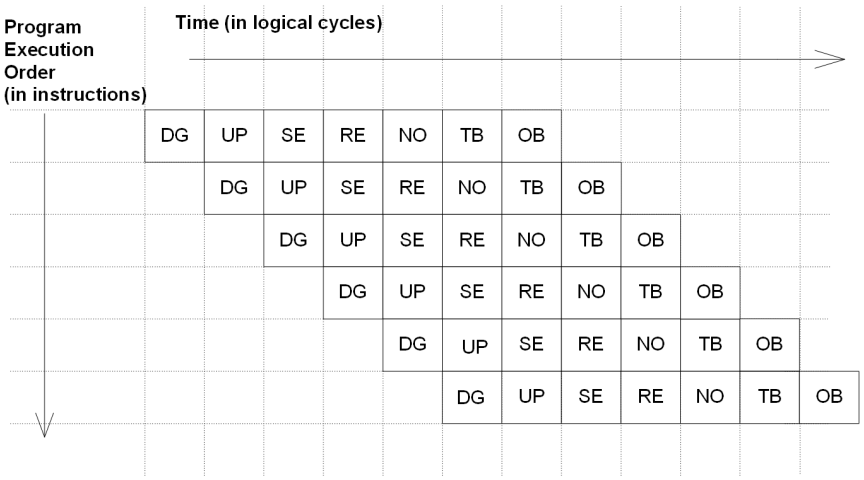

Figure 4.1: Illustrates an instruction execution pipeline of the VA logical-cycle datapath developed previously, with replication in space. The segments are arranged horizontally, and data flows from left to right, synchronously with the clock cycles.

- Identify next state to output from traceback unit

7. Output base (OB)

- Convert predicted state to its corresponding base call and output

Fig. 4.1 the temporal arrangement of the pipeline actions outlined above is shown. As this diagram shows, every one of these instructions in the processor is operative in every logical cycle, increasing the instruction execution rate by 7 times. Setting up storage registers between each state of the pipeline allows for the gathering of regional results between cycles.

4.3 Single-Channel Hardware Design

I now characterize two extremes of VA-based BC design to express the boundaries of a design-space for nanopore based CMOS basecallers: a uni-processor (UP) serial iterative architecture geared to process one state per cycle (i.e. one of 4^{k-1} states per trellis stage per clock cycle) and a node-parallel (NP) architecture geared to process one stage per cycle (i.e. all 4^{k-1} states per trellis stage per clock cycle). These approaches trade size for speed; the UP minimizes area at the expense of speed while NP does the opposite. As described, the NP is the extreme form of structural parallelism mentioned in Section 4.2.

A functional diagram of the NP is shown in Fig. 4.2; it is composed of three main blocks: the *state*, the *stage*, and the *traceback* blocks. The system's function in terms of these blocks is now elaborated.

4.3.1 The State Block

The conversion of event input signals, e_i, into base predictions starts with the state block. A finite bit width of d is assumed for this input. This input, along with previous calculations and the nanopore model is used.

For a NP architecture, the state block shown is one of 4^{k-1} identical components (the jth block is drawn in Fig. 4.2) working in parallel; a classic data

parallel arrangement. The objective of this block is to carry out the first three steps of the pipeline strategy described above. That is the steps: DG, UP, and SE.

For each state, j', associated with time index i the state block's *branch metric generator* (BMG) computes the four branch metrics, $a_{jj'}[i]$, associated with that state. That is, the four possible transitions into that state.

As discussed earlier, this BMG calculation may take the form of the l_1-norm between the measured event signal and the expected model value. In Fig. 4.2 the four branch metrics emerging from the BMG are labelled with the subscript $4j$, $4j-1$, $4j-2$, and $4j-3$, respectively denoting the four possible transitions into state j' (j is used in the subscript to simplify notation). This is a reflection of the state labelling scheme employed in the present work where a state j (at time i) may be transitioned into by any four states j (from time $i-1$) with the aforementioned four labels.

The branch metrics $a_{jj'}[i]$ are then added to their corresponding path metrics $\Gamma_j[i-1]$ which are available in the state block's *Path Metric Memory*$_{K_p}$. The subscript K_p denotes the number of nanopore channels being processed by a basecalling unit. The topic of multiple-channel basecalling is discussed in the following chapter and for the present discussion K_p may be ignored or simply

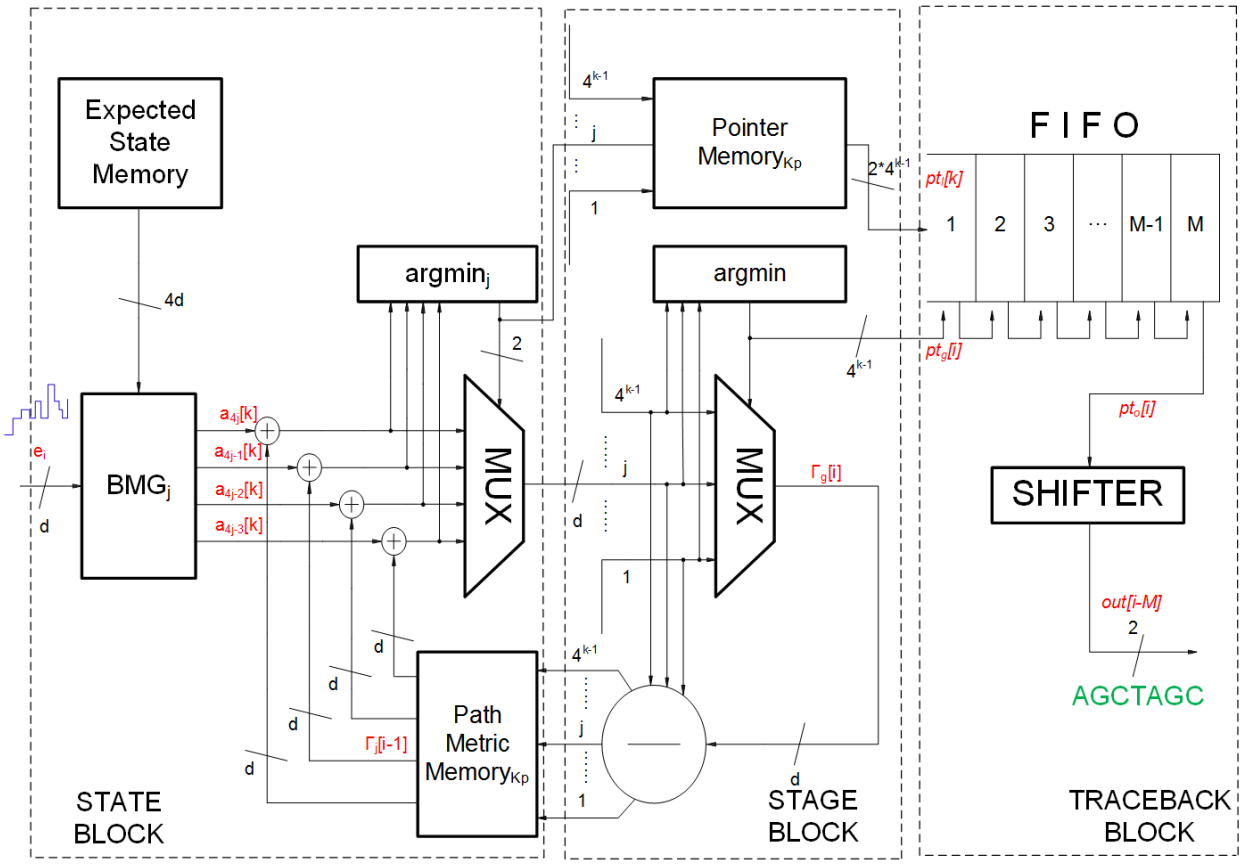

Figure 4.2: Block diagram of VA.

assumed to be $K_p = 1$.

The event input, e_i, into the state block's BMG at time i, with bit-depth d is fed into all 4^{k-1} state blocks which, in aggregate, calculate all 4^k candidate path metrics in one clock cycle and then whittle them down to the 4^{k-1} *survivor* states corresponding to time index i.

This reduction to 4^{k-1} states is done with the help of the minimum-argument

function $\arg\min_i$ that effectively creates a *local pointer*, pt_l, to the state labels of minimum distance path values (to j') . Given that each state block only processes the transitions pertinent to a state (i.e. the four transitions), these pointer values are local; they are essentially just referenced to the state block's four transition calculations. In ensuing parts (i.e. the stage block discussed in Section 4.3.2) a global state reference is computed. Since only four states are considered in each state block the local pointers only required two bits.

The UP's state block is identical to the NP's state block saving the fact that the UP has only one state block that executes the 4^k states in series. The UP is simply a time-multiplexed version of the NP system.

4.3.2 The Stage Block

The state block's results are gathered and processed by the stage block. As shown in Fig. 4.2 the stage block considers the 4^{k-1} survivor paths from the preceding 4^{k-1} stage blocks and, using the arg min function, determines the global minimum path metric $\Gamma_g[i]$. The pointers generated by the preceding states are also organized in a scratchpad memory, the *Pointer Memory*$_{K_p}$.

A coarse *global pointer*, pt_g referencing the state block with the minimum path is simultaneously computed. As discussed in Section 4.3.3, pt_g is combined with

pt_l in the traceback block to identify the exact identity of states on the optimal path through the trellis.

The stage block also re-references all accrued path metric calculations as indicated by the subtractor present in the feedback path from the stage block to the state block. This re-referencing at each stage prevents overflow in a fixed-point system.

4.3.3 The Traceback Block

The pointers generated in the state and stage blocks are used to populate (with pt_l) and drive (with pt_g) the traceback block's M-register first-in-first-out (FIFO) component. Specifically, the collection of 4^{k-1} 2-bit $pt_l[i]$ pointers organized in the preceding stage block are stored in the FIFO with every new sample of e_i at time index i. At the same time, $pt_g[i]$, is used to identify the state block corresponding to the minimum metric.

The combination of $pt_l[i]$ and $pt_g[i]$ can be used to calculate $pt_l[i-1]$ and $pt_g[i-1]$ (i.e. preceding minima). This process is repeated over the length of the FIFO until it's final (Mth) entry from which the emerging global pointer pt_o can be used (with a logical shifter) to identify the base identity corresponding to the input signal at time $i-M$ (i.e. $out[i-M]$).

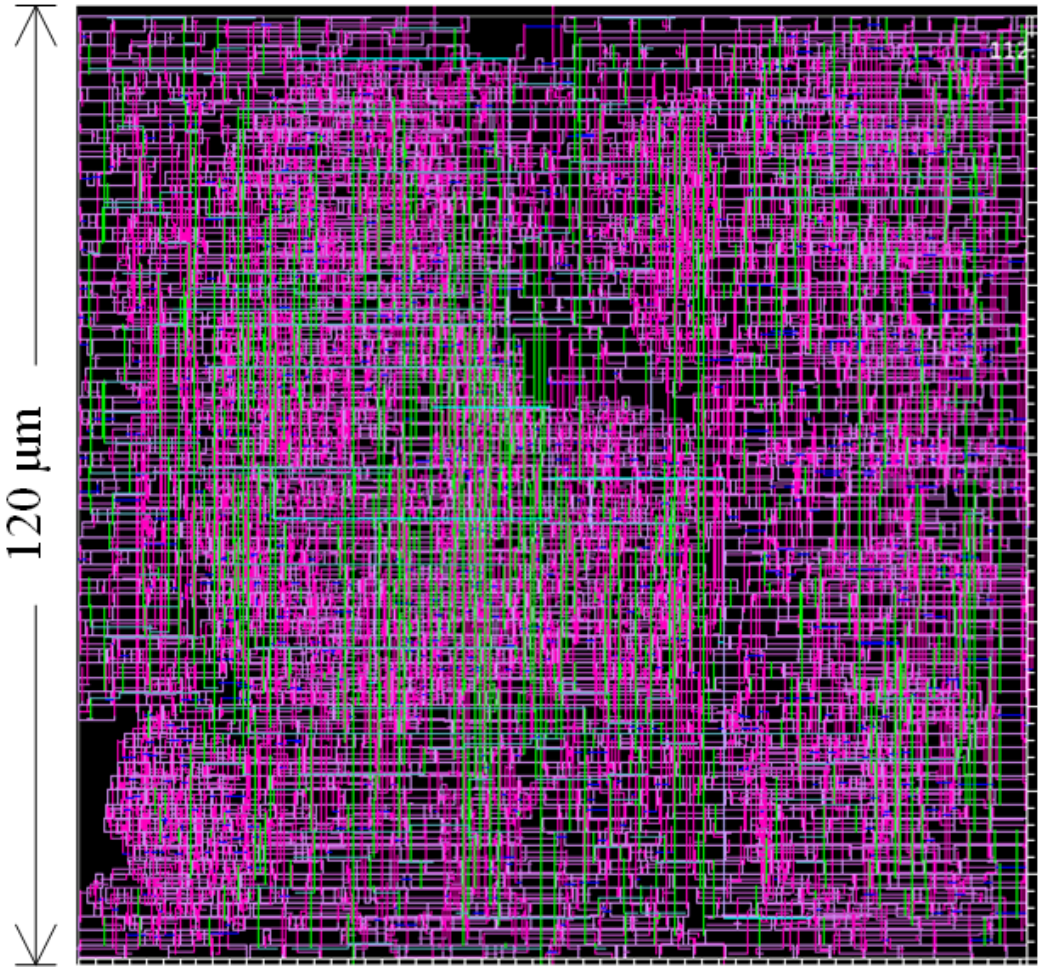

Figure 4.3: Example layout of a 64-state VA-based BC whose bitwidths is 6 in a 32-nm CMOS technology.

4.4 Single-Channel Hardware Realization

An example layout of a VA BC in a 32-nm CMOS technology is shown in Fig. 4.3. As with all the physical instantiations in this thesis, this construction was completed automatically with an electronic-design-automation (EDA) tool. In this particular example, the layout was accomplished using *IC Compiler* from Synopsys [54].

Chip areas vary from 120^2 μm^2 for a 64-state NP design to 510^2 μm^2 for a 1024-state chip with bitwidth $d = 6$, a footprint ratio of $16\times$ in line with the boost in states processed.

Chip power shows a similar characteristic, with an average value of 1.5 mW for the 64-state NP and 20 mW for its 1024-state counterpart.

The UP area advantage progressively increases as the bit depth grows, dropping from 75% to 60% to 55% relative to NP at $d = 6, 8, 10$, respectively.

Chapter 5

Multi-Channel Basecalling

5.1 Multi-Channel DNA Sequencing

The basecalling system considered in the preceding chapter operated on a single event sequence $\{e_i\}$. All advanced sequencers, including the nanopore-based sequencer from ONT generate a multitude of event sequences simultaneously. For example, ONT's palm-sized MinION system consists of 512 sequencing channels [55]. each one originating with a nanopore sensor and ending with some digital encoding device from which a corresponding $\{e_i\}$ sequence emerges. A DNA sample consisting of 1000s of DNA strands (on average about 5,000 bases per strand) is poured over the channels which then start to simultaneously operate on whatever strands begin translocating through their respective nanopore sensor.

As the nanopore-technology matures these numbers are expected to scale. Indeed, a desktop sequencing unit has already been developed that possesses 144,000 nanopore channels.

Given these practical circumstances we now propose a means of adapting the one-channel BC described in the previous chapter for the multi-channel scenario.

5.2 Multi-Channel Basecaller Arrangement

A high-level block diagram of a proposed multi-channel BC is shown in Fig. 5.1. The system may be imagined as an *intellectual property* (IP) block that would be included as part of a broader system. A common scenario would be for the system to be implemented as a stand-alone chip and interfaced to a bigger system via a printed circuit board (PCB). A growing trend, especially for mature communications technologies would be to realize the system as component part of a system-on-chip (SoC) and thus serve as potential contributor to a broader complex of applications (e.g. basecalling plus down-stream bioinformatics, encryption, communications, etc.)

The multi-channel BC (MCBC) design assumes a two-level hierarchy that, from the top, consists of a data parallel array of K_c basecalling cores. At the second level of the hierarchy, each core is assumed to be able to process K_p

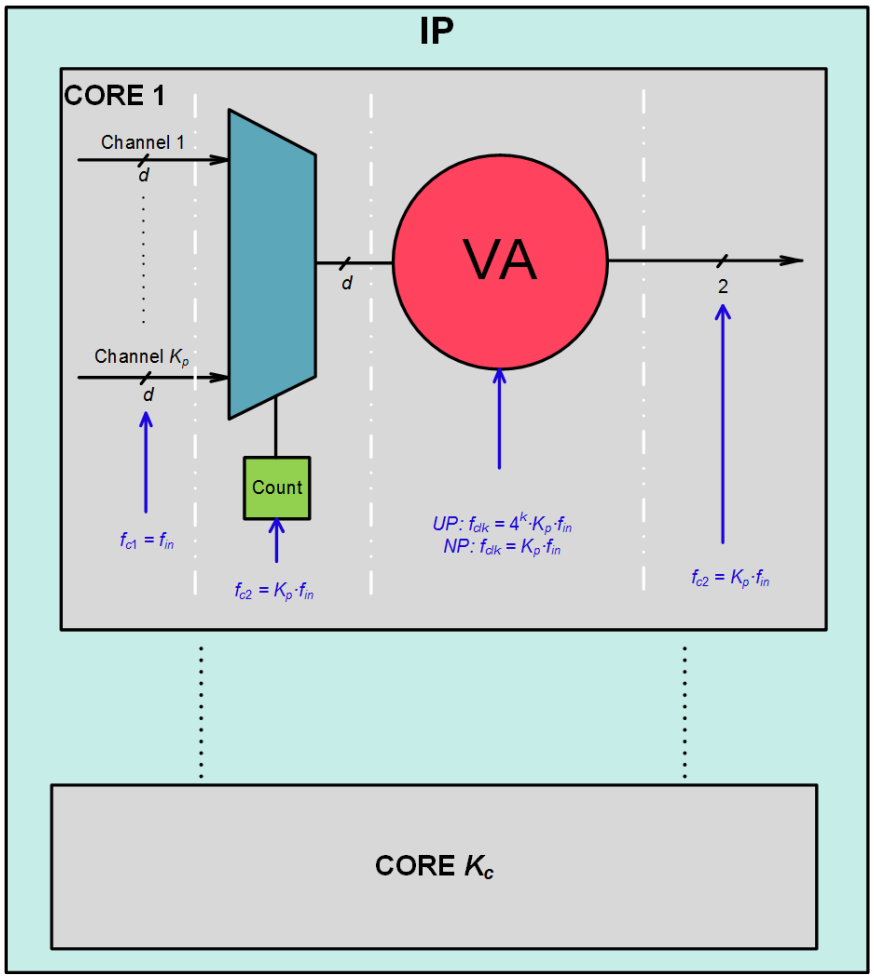

Figure 5.1: Representation of the internal contructure of the bioinformatical IP containing multiple parallel channels.

individual nanopore signals. That is each of the K_c cores receives a d-bit wide input, that input being comprised of K_p multiplexed nanopore channels. As a result, the total number of nanopore channels that can be addressed by such a design is

$$K_t = K_c \times K_p. \tag{5.1}$$

In other words, the MCBC is intended to handle data from a sequencer that operates K_t nanopore channels in parallel. In this thesis, the exact details of this sequencer are not considered. Also not considered in detail is the dispatch block that would be employed between the physical sequencer and the MCBC chip. Certainly, at least by virtue of the K_p multiplexed signals per core input, some pre-processing on raw sequencer data by a dispatch block is assumed by the MCBC. This may be as simple as aggregating signals from a single multi-nanopore sequencer. It may also take on a more sophisticated configuration, as for example aggregating and multiplexing signals from a multitude of multi-nanopore sequencers. As long as the MCBC is given even the implicit facility to identify which signals belong to which nanopore it will be able to consistently apply the sequence detection methods described in previous chapters to achieve correct basecalling.

5.3 I/O Considerations: Pins and Speeds

As with many high-performance application-specific integrated circuits (ASICs), the MCBC can be challenged by the number of signal inputs, K_t that are required to be processed. In particular, the physical footprint of an MCBC chip needs to provide enough space to physically accommodate these inputs. These inputs may appear simply as on-chip metal wires routing signals from other parts of a shared silicon substrate (i.e. an SoC) setting or they may be routed in from outside the chip via PCB traces onto package pins and then to metal pads on the chip.

Given the design arrangement described above, the total number of input and output signals handled by the MCBC is

$$K_{BC} = 2 \times K_c \times K_p. \tag{5.2}$$

Assuming an event word size of d bits as noted above, the number of input bits (and potentially individual input pads) can be expressed with

$$I_{BC} = K_c \times d \tag{5.3}$$

$$= \frac{K_t}{K_p} \times d. \tag{5.4}$$

To sustain real-time operation this input would have to be clocked into the MCBC at

$$f_{clk} = K_p \times f_{in} \tag{5.5}$$

where f_{in} is the frequency at which events e_i that are unique to individual nanopore channels enter the system. For a real-time system f_{in} simply equals the average rate at which bases move through (i.e. *translocate*) the nanopore sensor and thus generate new event levels. Recently, marketed nanopore sequencing technology has exceeded $f_{in} = 100$ Hz. Prototype systems have been reported with $f_{in} = 1000$ Hz. Experimental laboratory systems have reached translocation rates of $f_{in} = 1$ MHz. These numbers are clearly within the clock ranges of modern CMOS systems such as microprocessors, but the number of channels that must be processes and the complexity of the processing required to achieve basecalling combine to make it challenging.

The number of output pins required by an MCBC may vary depending on the output strategy. Assuming a completely parallel output (i.e. no serializer employed at the output of the MCBC) a total of number of output bits

$$O_{BCp} = K_c \times K_p \times 2 \tag{5.6}$$

would be generated where the factor of 2 denotes the fact that base calls assume only one of four labels (i.e. A or C or G or T). As a result the data may simply be clocked out at f_{in}.

Alternatively, assuming each core produces a multiplexed output in agreement

with its multiplexed input then the total number of output pins drops to

$$O_{BCm} = K_c \times 2 \tag{5.7}$$

in agreement with (5.3) save the change to a 2-bit output from a d-bit input. In this case the data is clocked out at the same rate as the input, that is at $K_p \times f_{in}$.

Finally the possibility of completely serializing the output (2-bit) word over all cores is available in which case only two output pins, $O_{BCs} = 2$ would be needed and the data would be clocked out at a frequency of $K_c \times f_{clk}$.

To establish a grasp for the quantities involved one may consider nanopore-based sequencers operating with $K_t = 1000$ nanopore channels that simultaneously gather signals and forward these to the MCBC (perhaps via an appropriate dispatch). Indeed, existing palm-sized platforms already incorporate 2048 sensors that are multiplexed in a redundant processing scheme to 512 channels. Given the emergent nature of this technology, 1000 nanopore channel operation in a mobile sequencer seems foreseeable for the near-term.

Assuming an MCBC with $K_c = 10$ cores, then each core would need to multiplex $K_p = 100$ nanopore signal channels to sustain $K_t = 1000$. Assuming an input of $d = 10$ bits then, as per (5.3), only 100 input pins would need to be connected to the MCBC, each pin operating at $f_{clk} = 10$ kHz, according to (5.5), assuming an f_{in} of 100 Hz. At the output, only $O_{BCm} = 20$ pins would be needed

in this configuration each pin also operating at 10 kHz. Imagining a nanopore channel signal increase to $f_{in} = 1$ MHz, the total number of pins (i.e. 120) does not change but the clocking requirement at the input and output goes from 10 kHz to 100 MHz.

In context of contemporary technology, an I/O count of 120 is not onerous. For example flagship field-programmable-gate-arrays (FPGAs) such as the Virtex-7 in an advanced FLG1925 package contain about 2000 pins of which about 60%, 1200 pins, may be used for signal I/O [56].

In an SoC setting where the MCBC would share a silicon substrate with other blocks this scenario may is even more easily accommodated as the full flexibility of on-chip interconnect could be leveraged. This includes not only the possibility of delivering the inputs to practically any portion of the chip, but also the ability to benefit from the minimal wiring pitch (conservatively, < 1 μm) available to IP embedded with an SoC.

Less convenient of course is the case where the MCBC is implemented as a stand-alone chip and must then somehow accommodate inputs provided via a package as considered above. Such a connection suffers from much more restrictive constraints in terms of arrangement and pitch than an SoC. Two classic chip-to-package interface options include periphery and area bonding [57]. The

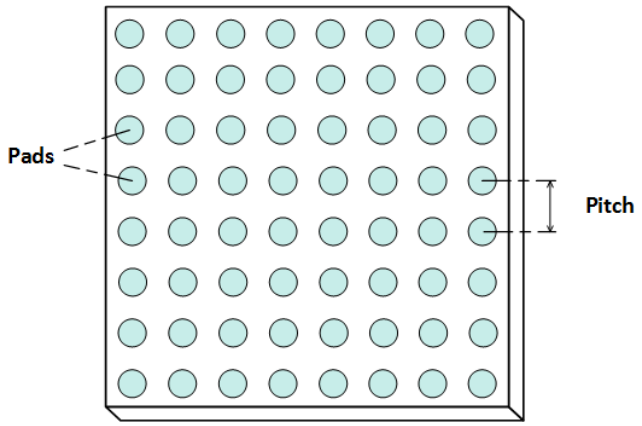

Figure 5.2: Representation of the pad distribution across a chip capable of accommodating area bonding.

former implies the use of wire bonds between the package and the chip's pads, the pads being distributed around the four edges of the chip. Area bonding refers to the ability to make connections to the chip, not only at its periphery but over its surface as well. A representative sketch of the top-view of a chip capable of accommodating area bonding is shown in Fig. 5.2.

For a given pad-pitch, l_p, and chip dimension $l_c = \sqrt{\text{Chip Area}}$ a periphery-bound design can accommodate $4l_c/l_p$ inputs while an area-bound design can accommodate l_c^2/l_p^2 inputs. For a pad pitch of 90 μm [58, 59] and a modest stand-alone design of $l_c = 5$ mm the edge bonded design can accommodate 222 pads while the area bonded configuration could accommodate 3086 pads. This is a clear advantage and a signal that even more aggressive scenarios than the one considered above are possible. We return to this point in Section 5.4.

Besides its ability to host many more I/Os, area bonding is beneficial for power distribution since the global power distribution is allowed to be placed in thick-film (greater than 10 μm normally) power planes in the package instead of in thin-film (less than 1 um typically) wires on the chip. Nevertheless, area bonding is a more costly proposition relative to periphery bonding due to sophisticated process needed to solder it as well as the complex multilayer packages needed to route the multitude of signals from the PCB [60].

5.4 Performance of 64-State MCBC

Fig. 5.3 conveys a more complete picture of the scalability of the VA MCBC for real-time performance over input frequency f_{in} at which nanopore event data is introduced (i.e. the frequency of new data per nanopore channel) and clock f_{clk} frequency at which the BC's VA is clocked. Its derivation and meaning are elaborated on presently.

5.4.1 64-State MCBC Performance Derivation

In particular, Fig. 5.3 summarizes the potential of a 64-state (i.e. $k = 3$, 3-mer) MCBC utilizing either NP or UP VA arrangements with the intention of processing 1000s of nanopore signal channels in a streaming (i.e. real-time) fashion. The

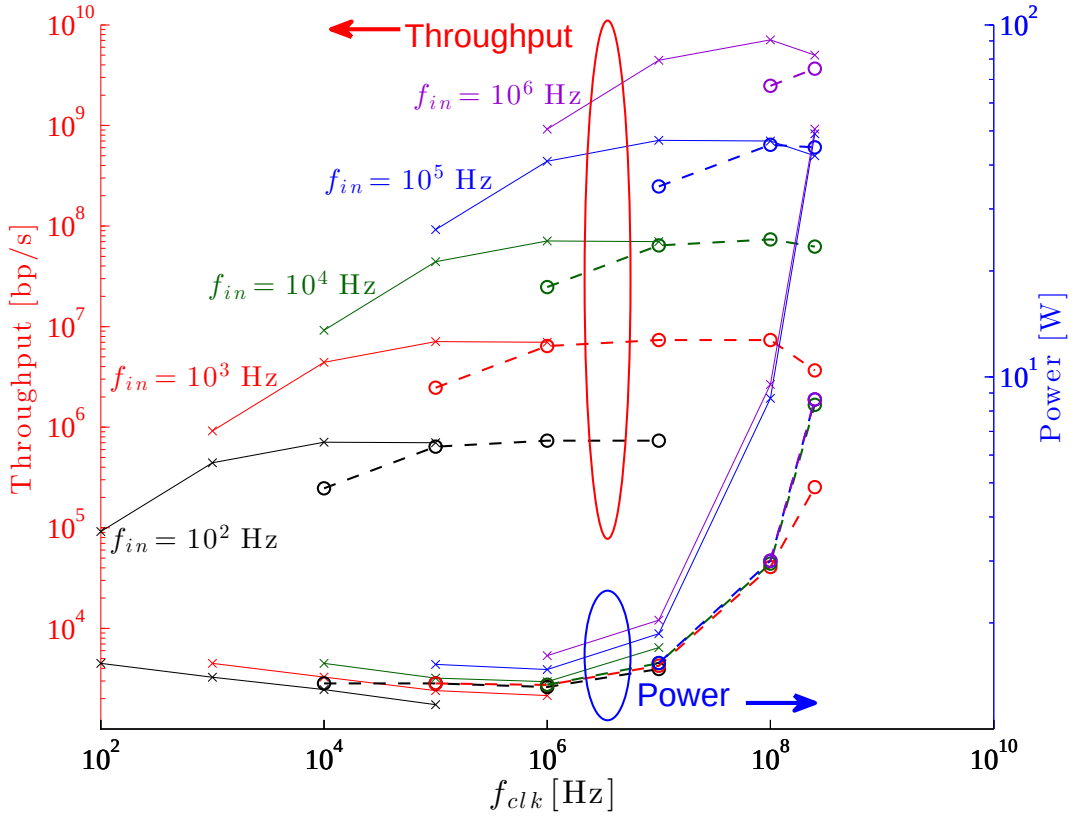

Figure 5.3: Real-time speed and power performance of a 64-state, 10-bit, 32-nm VA BC array in a 25 mm^2 die for NP (x+solid) and UP (o+dashed) arrangements vs. clock rate, f_{clk}, at different nanopore translocation rates, f_{in} (colour-coded).

input event word size is set to be $d = 10$ bits. A nanopore sensor capable of such performance is described in [35].

The chip is designed using a 32-nm CMOS technology and is limited to a 25-mm^2 ASIC die. This total area assumption serves as a constraint. For reference, such an area would take up about 25% of an Apple A6 iPhone 5 chip (also in a 32-nm CMOS technology) [61]. This is slightly below the percentage of silicon real-estate currently consumed by the graphics-processing-unit (GPU) on the A6 SoC.

Another constraint imposed on the study concerns the I/O count. Although we have described the chip in terms of an SoC environment we conservatively constrain it to I/O numbers as might be imposed by an area bonded system. Assuming the aforementioned 90-μm area pad pitch and assuming that 60% of the available connections can be used for signals, then the 25-mm^2 device can support approximately 1,850 I/Os. In general the total number of 10-bit inputs and outputs is

$$\begin{aligned} IO_{bp} &= \frac{K_t}{K_p} \cdot (d+2) \\ &= K_c \cdot (d+2) \end{aligned} \tag{5.8}$$

with the data clocked in at f_{in}. Under the present assumption $IO_{bp} = 1850$

limits the number of on-channel cores to $K_c = 154$, a somewhat restrictive value. Assuming that the I/O can be clocked in and out in bit-serial fashion for each word of size d then the I/O clocking requirement goes up to $d \cdot f_{in}$, a manageable increase given the typical values of f_{in} (i.e. ≤ 1 MHz as discussed below) while changing the I/O balance to

$$IO_{bs} = K_c \cdot \left(1 + \frac{2}{d}\right). \tag{5.9}$$

Thus, in the case of $IO_{bs} = 1850$, the one-channel core count can reach $K_c \approx 1{,}500$. In the calculations associated with Fig. 5.3 the maximum core count does not exceed $K_c = 919$ and thus conforms to this I/O constraint.

The f_{in} to which the MCBC is assumed to be subject is swept from 10^2 Hz (speeds realized in state-of-the-art commercial sequencers with modified nanopore proteins as noted above) to 10^6 Hz (the unfettered rate of bp translocation through nanopores) [24].

As outlined in Section 5.3 via (5.5), each NP core can multiplex $K_p = f_{clk}/f_{in}$ signals. For a UP core this multiplexing value must roughly be diminished by 4^k (i.e. 64 in the present example) for sensors that report events based on k-mer inputs since the UP sequentially calculates the 4^k transitions at each event index while the NP does so simultaneously. By arraying K_c such BCs on the ASIC die a total base pair throughput of $H = K_c \times K_p \times f_{in}$ can be managed.

The results shown in Fig. 5.3 use data obtained from simulations on the single-channel BC (SCBC) discussed in the previous chapter interpolated to an MCBC architecture. Power values are extracted using the Synopsys tool *PrimeTime* [62] after synthesis and layout are completed using Synopsys *DC Compiler* [63] and *IC Compiler* [54] tools respectively which allow for area estimation.

In the case of $K_p = 1$ the area of a MCBC core, A_{mc}, is equal to the area of a SCBC, A_{sc}. Otherwise, the following approximation can be used

$$A_{mc} = A_{sc} \cdot [(K_p - 1)P_{tb} + 1] \tag{5.10}$$

where P_{tb} is the percentage of A_{sc} consumed by the traceback block. The reasoning behind this associated is that whereas the state and stage blocks are only needed to carry out blind computations on incoming data and hence can be re-used over the different K_p input channels the traceback block must hold results unique to each channel and hence needs a separate instantiation. In the designs considered for this thesis $P_{tb} \approx 12\%$.

5.4.2 64-State MCBC Performance Summary

In Fig. 5.3 it thus approximated that at current nanopore sequencer translocation rates ($f_{in} = 10^2$ Hz) the manageable H can reach 7×10^5 bp/s, the equivalent of 20 human genomes per day (or 1 human genome in 70 minutes), achieved at

$f_{clk} = 0.1$ MHz for a NP implementation and $K_c = 7$ and $K_p = 1000$. The power consumption of this solution is 1.2 W with a power density of 4.8 W/cm^2, well below the 100 W/cm^2 capability of contemporary air cooling technology [64]. Still, from a smartphone technology perspective this power value is not insubstantial. For example, average smartphone power consumption during a 2G cellular phone call hovers around 1 W [65].

At the other extreme, processing input at the $f_{in} = 10^6$ Hz peak, the NP ASIC can achieve $H = 7 \times 10^9$ bp/s (200k human genomes/day or 1 human genome in about 0.5 seconds) a rate competitive with the abilities of core sequencing facilities. The chip can accomplish this at $f_{clk} = 100$ MHz with 10-W power consumption and with a power flux of roughly 40 W/cm^2.

In all cases we see a couple of interesting trends. First as the clock rate is increased we observer an initial improved in H followed by a saturation in performance and then a decline. The initial improvement is clearly due to the area efficiency gained by multiplexing channels within cores. The area required by a core increases as K_p goes up of course, but not as fast. Eventually however the limited chip area (i.e. 25 mm^2) prevents the ability to host more of these large cores and the performance saturates and eventually drops as more K_p are introduced.

Another interesting trend is the drop in power noticeable for the NP designs as the clock starts to increase from its minimum setting. Normally, the opposite effect would be expected and indeed this increase eventually does occur as the clock exceeds 1 MHz. Clearly, the multiplexing of channels into cores must be at work and likely results in a power drop due to a trade-off between the static power consumed in between switches of a large number (i.e. K_c) of cores and the dynamic power of a limited set of cores.

Although its throughput performance would become seriously compromised for state counts in excess of 64, a UP approach demonstrates the ability to perform competitively relative to its NP counterpart. For example, the UP with $f_{clk} =$ 250 MHz achieves $H = 3.7 \times 10^9$ bp/s, about half of NP's maximum throughput, for $f_{in} = 1$ MHz while consuming less than 10 W for a power density of 35 W/cm^2.

5.5 Performance of 4096-State MCBC

The previous design analysis considered a MCBC for a 3-mer (64-state) nanopore system. Although devices with such resolution are in development, present systems call for basecallers capable of analyzing state counts as high as 4096. We consider the performance potential for an MCBC in such a context presently. Given the scale of the problem, only the NP architecture was explored.

5.5.1 Operations per Second

To gauge the complexity of a 4096 system, we start by enumerating the number of fundamental core operations (COP) per event. Fundamental operations are define here as add, subtract, compare, load, and store. With reference to the system drawn in Fig. 5.4 the operations count per component block are as follows:

1. Controller: 8;
2. BMG: 4^{k+1}+1;
3. PMU: $(K_p$+10)$\cdot 4^{k-1}$+3;
4. Selector: $15\cdot 4^{k-1}$;
5. MPM: $5\cdot 4^{k-1}$;
6. Reductor: $4\cdot 4^{k-1}$+1;
7. Traceback: $90\cdot 4^{k-1}$+$9K_p$+40;
8. Baseout: 7;

These components result in a net COP count of

$$\mathrm{COP}(k, K_p) = (K_p + 130) \cdot 4^{k-1} + 9K_p + 60. \tag{5.11}$$

Since a 4096 state system is assumed, $k = 6$. The setting of K_p is of course variable, but due to the pipelined nature of its design and the fact that there is a six event delay between the calculation of a branch metric and the availability

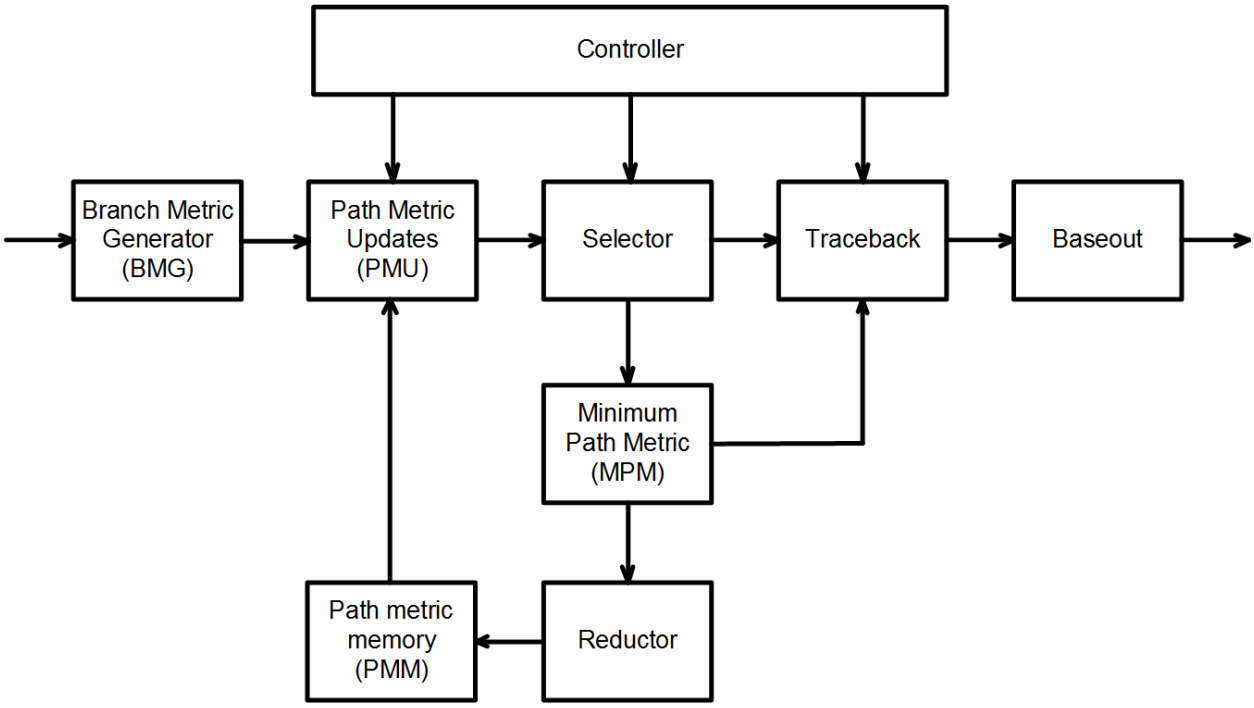

Figure 5.4: Simplified block diagram of the VA BC.

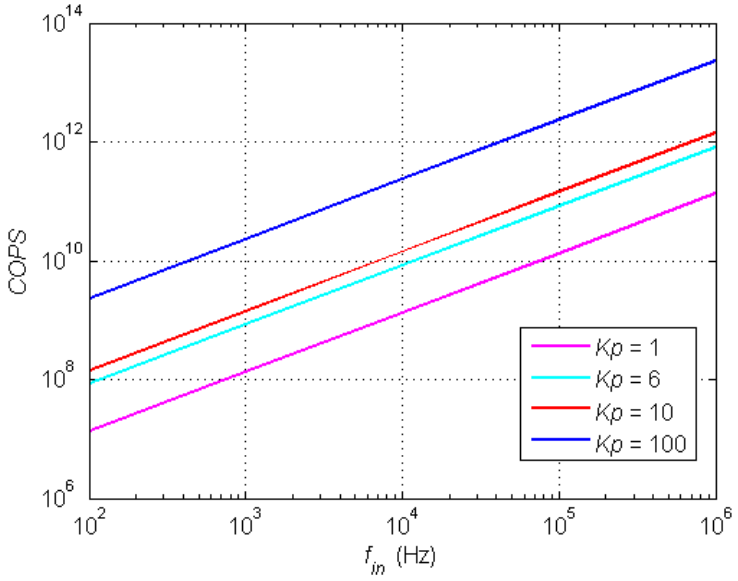

Figure 5.5: Estimate of core operations per second as a function of the nanopore sensor speed, f_{in}, and channels multiplexed per core, K_p.

of a normalized path metric choosing a $K_p \geq 6$ is convenient.

Thus, under the assumption that am MCBC core process $K_p = 6$ nanopore channels and that each nanopore channel produces events in response to a $k = 6$ mer the number of operations that a core must execute per event is COP($k = 6, K_p = 6$) = 557,170.

Of course of most interest is a measure of the number of operations that the unit is asked to perform in a unit time, in our case we invoke the core-operations-per-second (COPS) metric. This measure, based on (5.11), is shown in Fig. 5.5 as a function of f_{in} which, as already noted in (5.5), requires a clock (and hence

an operations execution rate) of $f_{clk} = K_p f_{in}$.

Fig. 5.5 plots the COPS of the 4096 design at various K_p settings over the range of present and anticipated future nanopore channel speeds, f_{in}. At minimal $f_{in} = 100$ Hz and preferred $K_p \approx 10$ we predict that COPS $\approx 10^8$ are required of the 4096-state core. As expected, increasing f_{in} by four orders of magnitude to 1 MHz requires a COPS around 10^{12}. Assuming the need to scale these to value by 100 to accommodate $K_t = 1000$ puts the potential COPS values at 10^{10}–10^{14}. To put these values in context, Intel Core i7 chips have been benchmarked at over 10^{11} instructions per second on Dhrystone tests [66]. These numbers at least imply the computational suitability of commodity processors like the i7 for the problem at hand, but with their roughly 100 W consumption per chip [67], power becomes a concern, and thus practically necessitates custom designs.

5.5.2 Power Consumption

Fig. 5.6 displays information about the power consumption of the 4096-state NP MCBC $K_p = 6$ core at different clock speeds as extracted using the Synopsys PrimeTime tool. The input word size for this design is $d = 6$.

As can be seen from this graph, when the clock speed is lower than 100 kHz, the average power consumption is always at round 0.135 W. Between 10^{-1} MHz

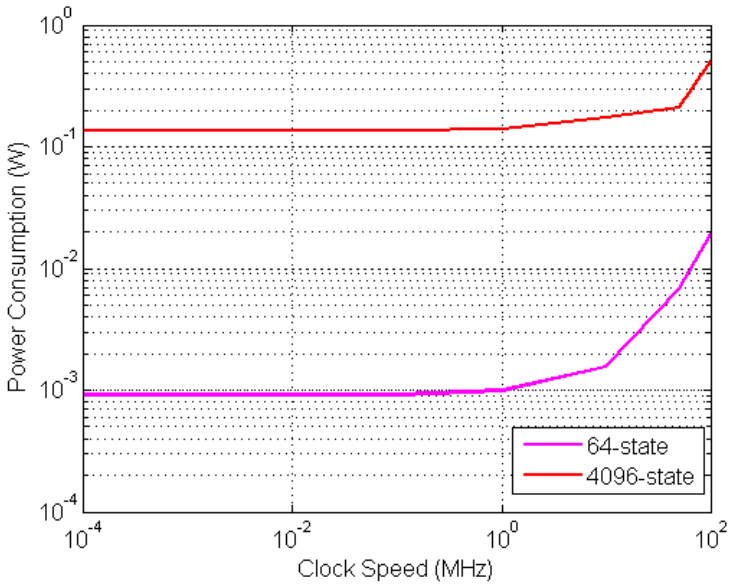

Figure 5.6: The comparison of power consumption and clock speed.

and 50 MHz, the average power consumption increases from round 0.135 W to round 0.21 W. After 50 MHz, it goes dramatically to the peak at round 0.5 W when the clock speed is 100 MHz. These value are roughly two-orders of magnitude higher than a 64-state core an understandable relationship given the relative number of states being processes. The power efficiency of the custom core relative to a commodity solution is clear however.

5.5.3 Core Counts

The number, K_c, of $K_p = 6$, $d = 6$ 4096-state MCBC cores that could be accommodated on a 5×5 mm^2 and a 10×10 mm^2 chip, respectively, are noted

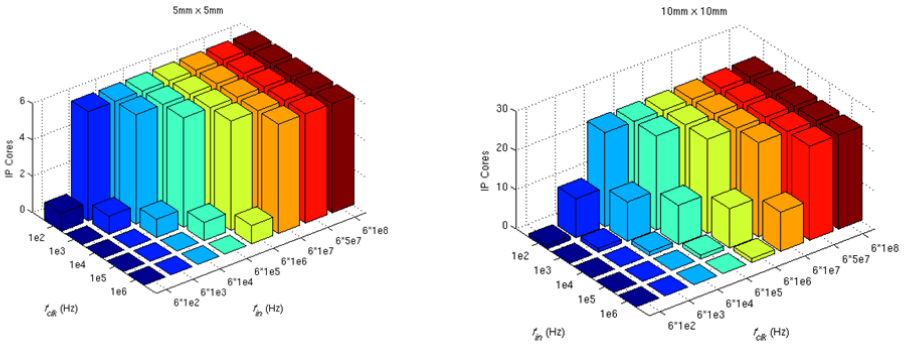

Figure 5.7: The number of 4096-state NP ($K_p = 6$) MCBC cores as a function of die area and clock speed.

in Fig. 5.7. The core counts are considered with respect to the clock frequency, f_{clk}, and the nanopore translocation frequency, f_{in}. As expected, the core counts scale linearly with area and we elaborate on the result for the 100-mm^2 chip only in the following discussion.

In the 100 mm^2 chip a maximum of $K_c = 24$ 4096-state cores can be placed when $f_{clk} >= 6f_{in}$. This means a chip capable of handling $K_t = 24 \times 6 = 144$ nanopore channels over input frequencies, f_{in}, between 100 Hz and 100-MHz. At $f_{in} = 100$ Hz this means a net throughput 14.4 kbp/s or 0.4 human genomes per 24-hours. With reference to Fig. 5.6 the power consumption of such a chip is $0.135 \times 24 = 3.24$ W, roughly 30× less than a run on a typical desktop CPU. This basecalling speed is more than 10× faster than HMM-based basecallers implemented on such CPUs [23].

Of course the above basecalling rate is a reflection of the input data rate

and not the innate speed of the 4096-state basecaller. At $f_{in} = 1$ MHz (and hence a clock rate is greater than only 6-MHz) the basecalling rate goes up to 14.4×10^7 bp/s or 2.9 human genomes per minute at a power consumption of $0.2 \times 24 = 4.8$ W.

Clearly, the ceiling for a custom 4096-state basecaller's speed is high, but more effort is needed in minimizing the physical footprint — the chip area — of such a design. In this case, the design's layout was entirely automated (using the Synopsys IC Compiler tool). It is likely that an effort at hand guided placement could result in a more efficient use of space and hence a higher throughput per chip area.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis considered the design of specialized ASIC hardware for the processing of signals for emerging miniature sequencing platforms. Roughly it considered this problem over a design space spanning six orders of magnitude (i.e. the range of the Viterbi state, bit-width, and input data speed).

At the present state of the sensing art with measured SNR below 10 dB, a basecaller input bit width of 6-dB seems adequate, but should probably operate at around 8-10 bits once SNRs improve towards 20-dB levels (the values of incumbent sequencing technologies).

For nanopores that produce signals in proportion to 3-mer DNA inputs, at present only an experimental technology under development, a single-channel 32-

nm CMOS basecalling core can be implement in a space of only 120^2 μm^2. For a comparable SoC (e.g. the 32-nm Apple A6) 380 such cores can be place in an area equivalent to the footprint of the GPU IP.

The potential for data throughput of such systems was explored in Chapter 5 where an arrangement of 445 64-state 32-nm cores with 10 channels per core could be placed within a 25 μm^2 area and, assuming a translocation rate of 1 Mbp/s per nanopore channel achieve the equivalent of calling a human genome in less than one second. Although actually achieving this rate depends on a sequencing sensor system capable of providing measurements at this rate from a multitude of channels, the trajectory of sequencer advances does not rule-out such a scenario. Further, the fact that such a calculation could be done at a power level of roughly 2 W underscores the advantage of this approach relative to desktop implementations which range around 100 W.

More contemporary scenarios were considered assuming a 4096-state 32-nm multi channel implementation. Although certainly much more resource intensive than its 64-state counterpart, the 4096-state implementation was predicted, in a 100 mm^2 silicon footprint and per-nanopore channel translocation rate of 1 Mbp/s, to achieve a throughput equivalent to about one human genome every 30 seconds. This is achieved at a power consumption of around 5 W. Again, these

value far outperform what is possible using commodity computing hardware.

6.2 Future Work

Improvements on the work presented herein could be made on a number of fronts. Perhaps the most pressing need is to demonstrate the performance levels predicted herein with actual measurements. An implementation in field-programmable-gate-array (FPGA) form could serve as an initial and very relevant step towards this.

Further, although the performance levels of the 4096-state design are impressive assuming a high enough nanopore channel input (i.e. 1 Mbp/s), present systems operate at four orders of magnitude below this value. For basecaller ASIC designers this implies that slower inputs must be multiplexed more efficiently per core, in other words each core must be kept sufficiently busy for a given operating clock. The present 4096-state system fails at this challenge for nanopore translocations speeds of 100 bp/s where its throughput is equivalent to only 0.4 human genomes per 24 hours.

Another improvement to the design would fall back on the construction of a more sophisticated trellis. The present design assumes that each new input to the base caller corresponds to a new base, but the measurement process may be

less ideal than this. Possible complications include the possibility that adjacent measurements correspond to the same base or that a measurement of a base is missed altogether. A statistical means of accounting for such behaviour is possible, but requires a more sophisticated trellis and hence a more complex hardware implementation.

Bibliography

[1] Artem E Men, Peter Wilson, Kirby Siemering, and Susan Forrest. Sanger dna sequencing. *Next-Generation Genome Sequencing: Towards Personalized Medicine*, pages 1–11, 2008.

[2] Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human molecular genetics*, 19(R2):R227–R240, 2010.

[3] Farzin Haque, Jinghong Li, Hai-Chen Wu, Xing-Jie Liang, and Peixuan Guo. Solid-state and biological nanopore for real-time sensing of single chemical and sequencing of dna. *Nano Today*, 8(1):56–74, 2013.

[4] Sebastian Magierowski. Everybodys basecalling whats what. *Tech. Rep. York University*, September 26, 2016.

[5] John J Kasianowicz, Eric Brandin, Daniel Branton, and David W Deamer. Characterization of individual polynucleotide molecules using a membrane

channel. *Proceedings of the National Academy of Sciences*, 93(24):13770–13773, Novemeber 1996.

[6] Erika Check Hayden. The automated lab. *Nature*, 516(4):131–132, December 2014.

[7] Hugo YK Lam, Cuiping Pan, Michael J Clark, Phil Lacroute, Rui Chen, Rajini Haraksingh, Maeve O'Huallachain, Mark B Gerstein, Jeffrey M Kidd, Carlos D Bustamante, et al. Detecting and annotating genetic variations using the hugeseq pipeline. *Nature biotechnology*, 30(3):226–229, March 2012.

[8] Yaniv Erlich. A vision for ubiquitous sequencing. *Genome research*, 25(10):1411–1416, May 2015.

[9] Joshua Quick, Nicholas J Loman, Sophie Duraffour, Jared T Simpson, Ettore Severi, Lauren Cowley, Joseph Akoi Bore, Raymond Koundouno, Gytis Dudas, Amy Mikhail, et al. Real-time, portable genome sequencing for ebola surveillance. *Nature*, 530(7589):228–232, February 2016.

[10] Shankar Balasubramanian. Polynucleotide sequencing, December 21 2004. US Patent 6,833,246.

[11] Lloyd M Smith, Jane Z Sanders, Robert J Kaiser, Peter Hughes, Chris Dodd,

Charles R Connell, Cheryl Heiner, SB Kent, and Leroy E Hood. Fluorescence detection in automated dna sequence analysis. *Nature*, 321(6071):674–679, 1985.

[12] Jonathan M Rothberg, Wolfgang Hinz, and Kim L Johnson. Methods and apparatus for measuring analytes using large scale fet arrays, October 15 2013. US Patent 8,558,288.

[13] Pal Nyren. Method of sequencing dna based on the detection of the release of pyrophosphate and enzymatic nucleotide degradation, July 10 2001. US Patent 6,258,568.

[14] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. Real-time dna sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

[15] Sebastian Magierowski, Yiyun Huang, Chengjie Wang, and Ebrahim Ghafar-Zadeh. Nanopore-cmos interfaces for dna sequencing. *Biosensors*, 6(3):42, 2016.

[16] Clive G Brown. 'no thanks, i've already got one'. *MinION Talk*.

[17] Clive G Brown. Comment on a tweet – revenue lags. https://community.nanoporetech.com/posts/comment-on-a-tweet, 2016.

[18] Brent Ewing, LaDeana Hillier, Michael C Wendl, and Phil Green. Base-calling of automated sequencer traces usingphred. i. accuracy assessment. *Genome research*, 8(3):175–185, 1998.

[19] Wei-Chun Kao, Kristian Stevens, and Yun S Song. Bayescall: A model-based base-calling algorithm for high-throughput short-read sequencing. *Genome research*, 19(10):1884–1895, 2009.

[20] Martin Kircher, Udo Stenzel, and Janet Kelso. Improved base calling for the illumina genome analyzer using machine learning strategies. *Genome biology*, 10(8):1, 2009.

[21] Tim Massingham and Nick Goldman. All your base: a fast and accurate probabilistic approach to base calling. *Genome biology*, 13(2):1, 2012.

[22] Oxford Nanopore Technologies. Metrichor. https://metrichor.com/s/, 2016.

[23] Matei David, Lewis Jonathan Dursi, Delia Yao, Paul C Boutros, and Jared T

Simpson. Nanocall: An open source basecaller for oxford nanopore sequencing data. *bioRxiv*, page 046086, 2016.

[24] Bala Murali Venkatesan and Rashid Bashir. Nanopore sensors for nucleic acid analysis. *Nature nanotechnology*, 6(10):615–624, September 2011.

[25] Alexander S Mikheyev and Mandy MY Tin. A first look at the oxford nanopore minion sequencer. *Molecular ecology resources*, 14(6):1097–1102, 2014.

[26] J. Medeiros. Your genes can now be sequenced using your usb port. `http://www.wired.co.uk/magazine/archive/2015/04/features/usb-gene-sequence/viewall`, 2015. Accessed: 2016-03-17.

[27] Stefan W Kowalczyk, Alexander Y Grosberg, Yitzhak Rabin, and Cees Dekker. Modeling the conductance and dna blockade of solid-state nanopores. *Nanotechnology*, 22(31):315101, 2011.

[28] GR Willmott and BG Smith. Comment on modeling the conductance and dna blockade of solid-state nanopores. *Nanotechnology*, 23(8):088001, 2012.

[29] Stefan W Kowalczyk, Alexander Y Grosberg, Yitzhak Rabin, and Cees

Dekker. Reply to comment on? modeling the conductance and dna blockade of solid-state nanopores? *Nanotechnology*, 23(8):088002, 2012.

[30] Jacob K Rosenstein, Meni Wanunu, Christopher A Merchant, Marija Drndic, and Kenneth L Shepard. Integrated nanopore sensing platform with sub-microsecond temporal resolution. *Nature methods*, 9(5):487–492, March 2012.

[31] Yiyun Huang, Sebastian Magierowski, Ebrahim Ghafar-Zadeh, and Chengjie Wang. High-speed event detector for embedded nanopore bio-systems. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 2179–2182. IEEE, August 2015.

[32] Meni Wanunu, Devora Cohen-Karni, Robert R Johnson, Lauren Fields, Jack Benner, Neil Peterman, Yu Zheng, Michael L Klein, and Marija Drndic. Discrimination of methylcytosine from hydroxymethylcytosine in dna molecules. *Journal of the American Chemical Society*, 133(3):486–492, 2010.

[33] Jiwook Shim, Gwendolyn I Humphreys, Bala Murali Venkatesan, Jan Marie Munz, Xueqing Zou, Chaitanya Sathe, Klaus Schulten, Farhad Kosari, Ann M Nardulli, George Vasmatzis, et al. Detection and quantification of methylation in dna using solid-state nanopores. *Scientific reports*, 3, 2013.

[34] Jiwook Shim, Younghoon Kim, Gwendolyn I Humphreys, Ann M Nardulli, Farhad Kosari, George Vasmatzis, William R Taylor, David A Ahlquist, Sua Myong, and Rashid Bashir. Nanopore-based assay for detection of methylation in double-stranded dna fragments. *Acs Nano*, 9(1):290–300, 2015.

[35] Winston Timp, Jeffrey Comer, and Aleksei Aksimentiev. Dna base-calling from a nanopore using a viterbi algorithm. *Biophysical journal*, 102(10):L37–L39, 2012.

[36] Stuart Dreyfus. Richard bellman on the birth of dynamic programming. *Operations Research*, 50(1):48–51, 2002.

[37] John Aldrich et al. Ra fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):162–176, 1997.

[38] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[39] Christian Ledergerber and Christophe Dessimoz. Base-calling for next-generation sequencing platforms. *Briefings in bioinformatics*, page bbq077, 2011.

[40] Shreepriya Das and Haris Vikalo. Onlinecall: fast online parameter estimation and base calling for illumina's next-generation sequencing. *Bioinformatics*, 28(13):1677–1683, 2012.

[41] Vladimír Boža, Broňa Brejová, and Tomáš Vinař. Deepnano: Deep recurrent neural networks for base calling in minion nanopore reads. *arXiv preprint arXiv:1603.09195*, 2016.

[42] Barry Merriman, Ion Torrent, Jonathan M Rothberg, R&D Team, et al. Progress in ion torrent semiconductor chip based sequencing. *Electrophoresis*, 33(23):3397–3417, December 2012.

[43] Ken McGrath. Sanger sequencing troubleshooting guide v1.1, November 12, 2014.

[44] Thermo Fisher Scientific Inc. Troubleshooting sanger sequencing data (Revision A.0), January 15, 2016.

[45] Sebastian Magierowski. Ont nanopore signal analysis. *Tech. Rep. York University*, October 8, 2016.

[46] Joachim Hagenauer and Peter Hoeher. A viterbi algorithm with soft-decision outputs and its applications. In *Global Telecommunications Con-*

ference and Exhibition'Communications Technology for the 1990s and Beyond'(GLOBECOM), 1989. IEEE, pages 1680–1686. IEEE, 1989.

[47] Trieu-Kien Truong, Ming-Tang Shih, Irving S Reed, and Edgar H Satorius. A vlsi design for a trace-back viterbi decoder. *IEEE Transactions on Communications*, 40(3):616–624, 1992.

[48] Gennady Feygin, Patrick Glenn Gulak, and Paul Chow. Generalized cascade viterbi decoder-a locally connected multiprocessor with linear speed-up. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 1097–1100. IEEE, 1991.

[49] Gennady Feygin, Patrick Glenn Gulak, and Paul Chow. A multiprocessor architecture for viterbi decoders with linear speedup. *IEEE transactions on signal processing*, 41(9):2907–2917, 1993.

[50] Gennady Feygin, Paul Chow, P Glenn Gulak, John Chappel, Grant Goodes, Oswin Hall, Ahmad Sayes, Satwant Singh, Michael B Smith, and Steven JE Wilton. A vlsi implementation of a cascade viterbi decoder with traceback. In *ISCAS*, pages 1945–1948, 1993.

[51] Ke Han and Richard Spencer. Comparison of different detection techniques

for digital magnetic recording channels. *IEEE transactions on magnetics*, 2(31):1128–1133, 1995.

[52] Kate R Lieberman, Gerald M Cherf, Michael J Doody, Felix Olasagasti, Yvette Kolodji, and Mark Akeson. Processive replication of single dna molecules in a nanopore catalyzed by phi29 dna polymerase. *Journal of the American Chemical Society*, 132(50):17961–17972, 2010.

[53] C Arun and V Rajamani. A low power and high speed viterbi decoder based on deep pipelined, clock blocking and hazards filtering. *International Journal of Communications, Network and System Sciences*, 2(6):575, 2009.

[54] Synopsys. Ic compiler user guide: Zroute, version b-2008.09-sp4, March, 2009.

[55] Adam D Hargreaves and John F Mulley. Assessing the utility of the oxford nanopore minion for snake venom gland cdna sequencing. *PeerJ*, 3:e1441, November 2015.

[56] Xilinx. 7 series fpgas packaging and pinout - product specification v1.14, March 23, 2016.

[57] Claude Louis Bertin, Thomas George Ference, Wayne John Howell, and Ed-

mund Juris Sprogis. Highly integrated chip-on-chip packaging, November 2 1999. US Patent 5,977,640.

[58] Rita N Horner, Rajendra D Pendse, and Fan Kee Loh. Implementation of pad circuitry for radially staggered bond pad arrangements. *Hewlett-Packard Journal*, pages 51–54, 1996.

[59] Chih-Pin Hung, Pao-Nan Li, Hsueh-Te Wang, and Yun-Hsiang Tien. Flip-chip package substrate with a high-density layout, May 6 2005. US Patent App. 11/123,204.

[60] William J Dally and John W Poulton. *Digital systems engineering*. Cambridge university press, 2008.

[61] Jason Tanner, Jim Morrison, Dick James, Ray Fontaine, and Phil Gamache. Inside the iphone 5s. *Chipworks*, September 20, 2013.

[62] Synopsys. Primetime si user guide, version v-2004.06, June, 2004.

[63] Synopsys. Design compiler user guide, version f-2011.09-sp2, December, 2011.

[64] Ghavam G Shahidi. Evolution of cmos technology at 32 nm and beyond. In

Custom Integrated Circuits Conference, 2007. CICC'07. IEEE, pages 413–416. IEEE, September 2007.

[65] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14. Boston, MA, 2010.

[66] Wikipedia. Instructions per second. `https://en.wikipedia.org/wiki/Instructions_per_second#MIPS`, updated 18 October, 2016. [Online; accessed 06-November-2016].

[67] Alternate Wars. Computing power throughout history. `http://www.alternatewars.com/BBOW/Computing/Computing_Power.htm`, updated 13 March, 2014. [Online; accessed 06-November-2016].