Energy Efficient Computing with Time-Based Digital Circuits

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF MINNESOTA

BY

Luke R. Everson

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Prof. Chris H. Kim, Advisor

May 2019

Acknowledgements

The PhD has been a special journey, and although it is a tremendous personal achievement, it has been the furthest thing from a singular effort. I cannot thank my wife, Megan, enough for her ceaseless support, love, and patience through these past four years. I have joked that she has heard me rehearse presentations enough, so that if I were to fall down the stairs on the day of a talk, she could give them in my place! Even through the seemingly perpetual challenges during this PhD, we grew stronger as a couple and always put our commitment to each other first.

I am fortunate that my family has always been there to encourage me. My in-laws deserve credit for their enduring patience in many regards as at times this investment might not have always looked like it would come to fruition. My parents have been the biggest influence in my life, and I am lucky for that because they have lived a life that I am proud to follow.

I treasure this opportunity afforded to me by Professor Kim. His support and vision gave me intellectual freedom to create some very creative, unconventional circuits. The relentless pursuit of simple, elegant, and intuitive designs drove me each and every day to give my best. At times it wasn't always easy, but I will forever cherish the time I got to work in his lab.

My committee members; Professor Sapatnekar, Professor Parhi, and Professor Yang have provided a sounding board for some of my work and their encouragement was instrumental during my career. I am also grateful for the time they spent reading this thesis and providing insightful comments.

My group members; Ibrahim Ahmed, Muqing Liu, Po Wei Chiu, Gyusung Park, Nakul Pande, Minsu Kim, Jeehwan Song, Chen Zhou, Saurabh Kumar, Qianying Tang (VLSI I & II TA, owe her a great deal of thanks for teaching me the tools), Somnath Kundu, Jongyeon Kim, Won Ho Choi, and Hoonki Kim were an ideal audience for vetting new ideas at team meetings, helping at any time with measurements or to discuss a circuit. I was fortunate to have them as colleagues, and now as friends. I also want to thank Kim Lab alumni John Keane for the mentorship during summer 2018.

This journey actually began during my junior year of undergraduate study in Professor David Lilja's lab under the direction of Dr. Manas Minglani. This opportunity was initiated by the Department, of which I am grateful for the financial support, academic advising, and providing a relevant and engaging curriculum. Additional research experiences along the way enriched my skillset and broadened my mindset. I want to thank Robert Lattin of Thermo King for building my confidence in hands-on engineering. Dr. Dwaipayan Biswas, Dr. Nick Van Helleputte and Dr. Chris Van Hoof of imec for providing an environment unparalleled for innovation, and Wes Santa and Dr. Erik Peterson of Medtronic for mentorship during the final year of my study.

Dedication

For Megan

Abstract

Advancements in semiconductor technology have given the world economical, abundant, and reliable computing resources which have enabled countless breakthroughs in science, medicine, and agriculture which have improved the lives of many. Due to physics, the rate of these advancements is slowing, while the demand for the increasing computing horsepower ever grows. Novel computer architectures that leverage the foundation of conventional systems must become mainstream to continue providing the improved hardware required by engineers, scientists, and governments to innovate. This thesis provides a path forward by introducing multiple time-based computing architectures for a diverse range of applications. Simply put, time-based computing encodes the output of the computation in the time it takes to generate the result. Conventional systems encode this information in voltages across multiple signals; the performance of these systems is tightly coupled to improvements in semiconductor technology. Time-based computing elegantly uses the simplest of components from conventional systems to efficiently compute complex results. Two time-based neuromorphic computing platforms, based on a ring oscillator and a digital delay line, are described. An analog-to-digital converter is designed in the time domain using a beat frequency circuit which is used to record brain activity. A novel path planning architecture, with designs for 2D and 3D routes, is implemented in the time domain. Finally, a machine learning application using time domain inputs enables improved performance of heart rate prediction, biometric identification, and introduces a new method for using machine learning to predict temporal signal sequences. As these innovative architectures are presented, it will become clear the way forward will be increasingly enabled with time-based designs.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

The uniting theme for all the work in this thesis is time. In Chapters 2, 3, and 4, time is used as the state variable in the computation. Traditional computing uses voltage as the state variable; digital recognizes binary states of "on" and "off" or VDD and VSS to encode computation. Analog computers use a real-valued representation of the voltage to encode a value with infinite resolution. Time based circuits encode the computation value in the evaluation time. This idea has recently been gained popularity [1] due to the flattening of Moore's law [2]. However, the idea of encoding data in time delays is nothing new. In a seminal review [3] of digital computer memories, delay-line memories are described to be used by injecting a pattern into a medium, such as mercury, which delays the propagation of the pattern until it reaches the end of the medium and it is read out. The purpose of these memories was to store, or delay, the information from the time it was generated until the time the next computation could use it. This paradigm is no different than that of some of the work presented in this thesis; time-based computing can be used to when digital systems fall short. In Chapter 5, temporal signals are used directly for computing, compared to conventional methods which derive features from the signals and compute on those. To meet the requirements of modern applications in the face of slowing technical innovation, time-based computing architectures will be presented on a diverse set of applications; neuromorphic and machine learning, analog to digital conversion, graph computing, and biosignal processing.

1.1 Time-based Neuromorphic Circuits

1.1.1 A Scalable Time-Based Integrate-&-Fire Digitally Controlled Oscillator

A fully scalable light-weight integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition features is implemented in 65nm. The core computes the neural net algorithm entirely in the time domain using standard digital circuits. A parallel two-layer architecture realized using the proposed core achieves a 91% handwritten digit recognition accuracy. The 0.24mm^2 neuromorphic core including 64 digitally controlled oscillator (DCO) circuits consumes 320.4µW per DCO at a maximum throughput of 746M pixels/s.

1.1.2 A One-Shot Neuromorphic Core with Dynamic Threshold Error Correction

As neural networks continue to infiltrate diverse application domains, computing will begin to move out of the cloud and onto edge devices necessitating fast, reliable, and low power solutions. To meet these requirements, a time-domain core using one-shot delay measurements and a lightweight post processing technique, Dynamic Threshold Error Correction (DTEC) will be presented. This design differs from traditional digital implementations in that it uses the delay accumulated through a simple inverter chain distributed through an SRAM array to intrinsically compute resource intensive multiply-accumulate (MAC) operations. Implemented in 65nm LP CMOS, the design achieves an energy efficiency of 104.8TOp/s/W at 0.7V with 3b resolution for 19.1fJ/MAC.

1.2 Time-based Graph Computing

1.2.1 A 40×40 2D Gradient-enabled A Path Planning ASIC*

A mixed-signal time-based 65nm application specific integrated circuit is developed for solving shortest-path problems. The core follows similar principles from wave routing and additionally incorporates a gradient on the periphery of the core to implement the A* algorithm. A leading pulse is propagated from start nodes and is asynchronously latched in neighboring vertex cells and pushed to its four neighbors. Applications include collision avoidance for self-driving cars, shortest path planning, scientific computing, and is shown to be scalable across many cores. The chip achieves 559 million traversed edges per second at $10^5\times$ improved energy efficiency compared to existing platforms such as FPGA and CPU. The processor operates nominally at 1.79ns per node with peak power consumption of 26.4mW.

1.2.2 A 20^3 cube for 3D Graph Traversal

Building on the work presented in the prior section, the primary limitation of adopting the 2D path finding framework in drones and UAVs is the lack of the vertical spatial dimension. This has been addressed in the 20^3 cube purpose designed for drone navigation. A novel 3D-to-2D transformation is developed to create a practical, scalable architecture in standard 2D CMOS. A diverse range of applications including path planning, routing, 3D Voronoi diagram generation, and k nearest neighbor classification is demonstrated on the core. The chip achieves throughput of 1.38 billion traversed edges per second at 0.162pJ/node at 1.2V in a $2mm^2$ die area making it ideal path planner for resource constrained platforms such as quadcopters.

1.3 Time-based Biosignal Recording System

A digital-intensive, low-area, time-based ADC optimized for in-situ neural recording is fabricated in a 65nm test chip and validated with *in-vivo* data. The intrinsic inversely proportional gain of a beat frequency based quantizer allows recording of sub-millivolt neural signals without any sophisticated amplifiers or filters. A low-area analog-front-end (AFE) is implemented with a standard digital logic inverter transimpedance amplifier and tunable low pass and high pass filters. The test chip achieves 20.9dB SNDR for a 1mVpp input at 416Hz with a bandwidth of 4.2 kHz and consumes 52µW at 0.8V. *In-vivo* evoked potentials and spontaneous activity were measured directly from a mouse cerebellum without any external components, validating the efficacy of the aggressive tradeoffs. These results are achieved in an area of 0.0094mm^2/channel, including on-chip AC coupling and filter passives, which makes this an attractive architecture for complete integration in ultra-high channel count neural recording systems.

1.4 Time-based Photoplethysmography Machine Learning Algorithms

1.4.1 BiometricNET: Deep Learning based Biometric Identification using Wrist-Worn PPG

Rapid advances in semiconductor fabrication technology have enabled the proliferation of miniaturized body-worn sensors capable of long term pervasive biomedical signal monitoring. In this section, a novel deep learning-based framework (BiometricNET) on biometric identification using data collected from wrist-worn Photoplethysmography

(PPG) signals in ambulatory environments will be presented. A completely personalized data-driven approach is formulated, using a four-layer deep neural network - employing two convolution neural network (CNN) layers in conjunction with two long short-term memory (LSTM) layers, followed by a dense output layer for modelling the temporal sequence inherent within the pulsatile signal representative of cardiac activity. The proposed network configuration was evaluated on the TROIKA dataset collected from 12 subjects involved in physical activity, achieved an average five-fold cross-validation accuracy of 96%.

1.4.2 CorNET: Deep Learning Framework for PPG based Heart Rate Estimation

The proliferation of miniaturized body-worn sensors capable of long-term pervasive biomedical signal monitoring has ignited a data revolution in healthcare. Remote cardiovascular monitoring has been one of the beneficiaries of this development, resulting in non-invasive, photoplethysmography (PPG) sensors being used in ambulatory settings. Wrist-worn PPG, although a popular alternative to electrocardiogram (ECG), inferring cardiac information (e.g. heart rate) is challenging owing to artifacts induced by motion inherent in daily life. Hence, in this chapter, a novel deep learning framework (CorNET) to efficiently estimate heart rate (HR) information and perform biometric identification (BId) using only wrist-worn, single-channel PPG signal collected in ambulant environment is described. A completely personalized data-driven approach is formulated, using a four-layer deep neural network - employing two convolution neural network layers in conjunction with two long short-term memory layers, followed by a dense output layer for modelling the temporal sequence inherent within the pulsatile signal representative of

cardiac activity. The final dense layer is customized with respect to the application, functioning as: a) regression layer - having a single neuron to predict HR; b) classification layer - two neurons which identifies a subject among a group. The proposed network was evaluated on the TROIKA dataset having 23 PPG records collected during various physical activities, achieved mean absolute error of 0.48±0.19 BPM for HR estimation and an average accuracy of 96% for BId on 20 subjects.

1.4.3 BioTranslator: Deep Learning Framework for Converting Time Series Biosignals

Advancements in wireless sensor networks (WSN) technology and miniaturization of wearable sensors have enabled long-term continuous pervasive biomedical signal monitoring. Wrist-worn photoplethysmography (PPG) sensors have gained popularity given their form factor. However the signal quality suffers due to motion artifacts when used in ambulatory settings, making vital parameter estimation a challenging task. In this section, a novel deep learning framework, BioTranslator, is presented for computing the instantaneous heart rate (IHR), using wrist-worn PPG signals collected during physical activity. Using one-dimensional Convolution-Deconvolution Network, it translates a single channel PPG signal to an electrocardiogram (ECG)-like time series signal, from which relevant R-peak information can be inferred enabling IHR measures. The proposed network configuration was evaluated on 12 subjects of the TROIKA dataset, involved in physical activity. The proposed network identifies 92.8% of R-peaks, besides achieving a mean absolute error of 51ms with respect to reference ECG-derived IHR.

Chapter 2. Time-based Neuromorphic Circuits

2.1 Introduction

Machine Learning (ML) has become one of the hottest topics in mainstream research as well as IC design. The incredible performance on diverse applications has made it a silver bullet for problems typically ill-suited for traditional computers [4]. This has led to a proliferation of custom silicon solutions targeted at meeting the demand [5]. The remainder of this section will introduce time-based neuromorphic computing and compare and contrast to alternative neuromorphic architectures proposed in the literature. In Section 2.2 A Scalable Time-Based Integrate-&-Fire Digitally Controlled Oscillator is described [6]. Next, Section 2.3 details An Energy Efficient Time-Based One-Shot Neuromorphic Chip [5]. Finally, conclusions will be drawn in Section 2.4.

2.1.1 Time-based Neuromorphic Foundations

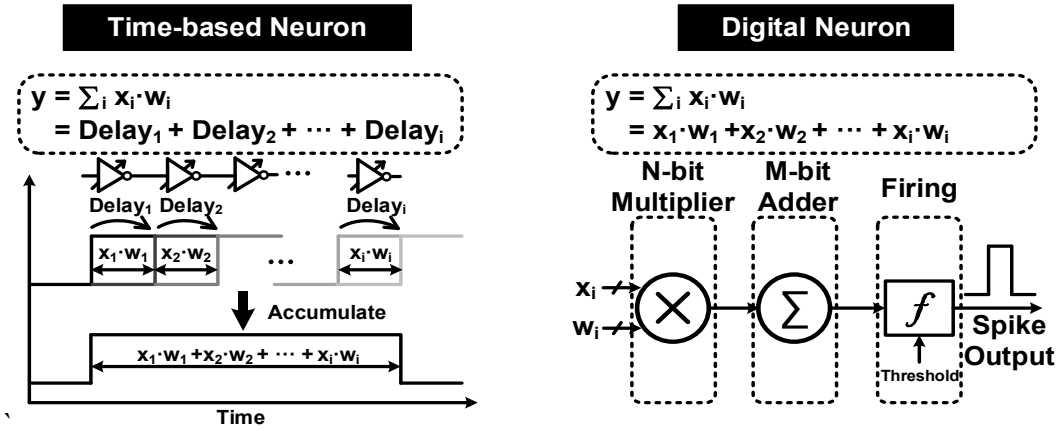


Figure 2.1: Time-based neurons utilize the delay through basic circuit elements such as inverters to implement the dot-product.

Equation 2.1 shows the characteristic kernel in ML is the dot-product:

$$y = \sum_i x_i \cdot w_i \quad \textbf{2.1}$$

Here x_i is the input data and w_i is the synaptic weight learned during the training phase of the application. Time-based neurons utilize the delay through basic circuit elements such as inverters to implement the multiplication. The primary benefit of time-domain circuits is that the accumulate portion of the multiply-accumulate (MAC) is intrinsic to the architecture; realized by accumulating the delays. The delay can be measured by counting oscillation cycles in a ring oscillator (ROSC) [6] or comparing delays with a time-to-digital converter (TDC) [5]. Digital neurons use conventional Boolean logic for arithmetic operations such as arrays of multipliers and wide accumulation adders [7]. Both architectures can be mapped to deep learning applications. It should be noted that additional operations such as non-linear transfer functions, pooling (down-sampling), and drop-out during training can be implemented readily in digital circuits. Additionally, data movement of inputs, weights, and outputs between main memory, on-chip cache, and processor needs to be facilitated. These concerns are considered open research [8] and not addressed in this thesis. The following section will discuss prior art from different neuromorphic architectures.

2.1.2 Alternative Neuromorphic Architectures

2.1.2.1 Digital Systems-on-Chip

Digital Systems-on-Chips (SoCs) have found success through many algorithmic techniques. Weight resolution modulation is employed in [7] by leveraging a reconfigurable multiplier block capable of 16-bit, 8-bit, or 4-bit operations to enable a 40x energy-precision scalability. However, there is a significant overhead as the MAC unit has

unused cells at full precision. Another feature is inputs and weights that have a value of zero assert a "guard" flag that bypasses that particular computation. Careful design of the memory-hierarchy in [9] minimizes data movement between off-chip DRAM memory by exploiting data reuse. They also incorporate knowledge of the data statistics to reduce energy by gating computations, as in [7], and using data compression to limit the size of data transferred off-chip. A 14x12 array of Processing Engines is used to compute the MACs in parallel. These cells each have over 4kb of registers and memories which reduces area efficiency and greatly increases the footprint of the SoC. Each Processing Engine is power gated to further increase energy efficiency.

2.1.2.2 Analog SRAM Solutions

SRAM memory-based designs that leverage charge sharing have also been proposed. In [10], SRAMs are used to store weights. Interspersed in the array are local analog moving average blocks, the control unit that implements the charge sharing across bitlines. This design makes use of low power analog techniques to drive down power, but relies on charge sharing and utilizes a time-dependent pre-charge scheme to implement the input. These two techniques limit the scalability of the design and as such they are limited to convolutional operations which have reduced input lengths due to the filter size. As a result of using charge-sharing, the SRAMs cannot be directly connected together which is why 10T bitcells are utilized instead of the conventional, denser 6T bitcells. Reference [11] also leverages charge-sharing between SRAMs in order to implement signed multiplication. They demonstrate that using in-memory computing can reduce power consumption by as much as 4.5x. However, when used as an analog device, process variation from SRAMs

necessitates on-chip learning. Models learned on one chip and applied to another can cause a 43% drop in accuracy. This requires a massive increase in overhead, reducing the efficiency of the core.

2.1.2.3 Analog Neuromorphic Systems

Early approaches relied on analog circuits to mimic synapse and neuron functions [12]. Analog circuits use the sub-threshold region of MOSFET devices to implement the computation. Analog neurons have low area and power consumption thanks to their subthreshold operation. However, they are slower and more sensitive to process, voltage, and temperature and compensation techniques require a large amount of overhead. Neurogrid [13] is an example of an analog neuron system. The system was intended to simulation large-scale neural interactions such as in a rat brain. The main drawback of using analog circuits to implement brain-inspired computing models is that they are sensitive to noise and process variation, so homogeneity and precision cannot be guaranteed for large scale designs. Scaling of CMOS technology also poses a challenge for analog designs as matching becomes challenging as process spreads increase.

2.1.2.4 Digital SRAM Crossbar

Digital SRAM implementation of neural nets has been a more popular approach lately. Compared to analog neural nets, they are less vulnerable to noise and process variation, and can benefit from technology scaling, enabling massively parallel neuromorphic ASIC systems such as IBM's TrueNorth [14]. Weights are stored in the SRAM array and wordlines represent input bits to implement the partial products of the

multiplication which is called a crossbar array. Accumulation is implemented with local registers on the bitline. The registers can be cleared randomly to implement plasticity.

2.1.2.5 Non-volatile Crossbar Designs

One of the key limitations of using SRAMs is the volatile storage. This requires the power supply to be connected constantly drawing static leakage power. If it is disconnected, an overhead will be incurred to reprogram the array to prepare the array for computation. Nonvolatile storage devices such as ReRAM [15] and eFlash [16] present opportunities to have persistent weight storage. The arrays work much in the same manner as SRAM crossbar arrays where the accessed bitline currents are summated to implement the MAC. ReRAM is considered an ideal candidate due to not only the low access latency and energy, but the small footprint could enable very dense arrays. ReRAM can be thought of as a programmable, analog, nonvolatile resistor. However, due to the non-uniform analog resistance states, [15] asserts that it can cause errors in the convolution. They work around this by using ReRAM as a digital device which has benefits including: better programming accuracy, binary voltages applied to WL is scalable due to lower IR-drop in large arrays, and it does not require a large on/off resistance ratio previously required in analog ReRAM [15]. The key limitation is the lack of a capable commercially available process and even [15] does not have measurement results to support their claims. eFlash arrays utilize multi-level storage element and are logic compatible to reduce cost and available in all processes [16]. The main drawback is the large cell size due to the I/O devices required to limit gate leakage on the storage node, on the order of 4x larger than an SRAM even accounting for the eFlash multilevel cell storage.

All of these different architectures have benefits and drawbacks. Time-based computing will be studied closer in the following sections. A strong case will be made for the adoption of these architectures due to their field leading energy efficiency, area density, and application performance. Two designs will be described in detail in the following sections.

2.2 A Scalable Time-Based Integrate-&-Fire Digitally Controlled Oscillator

2.2.1 Integrate-&-Fire Architecture and Concept

Figure 2.2 shows the time-based integrate-&-fire digital controlled oscillator (DCO) neuromorphic core. Each DCO has 128 stages with 3b programmable weight and binary input.

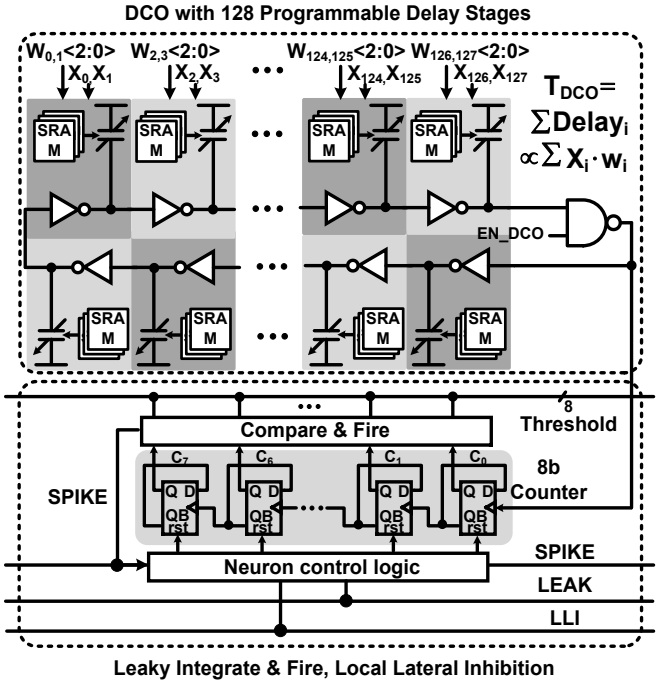


Figure 2.2: Conceptual circuit diagram of the proposed time-based integrate & fire (I&F) DCO neuromorphic core.

The detailed implementation of programmable unit cell delay and the leafcell layout of two delay stages are shown in Figure 2.2. Each input stage of the DCO is composed of an inverter and binary-weighted MOSFET capacitors mixing an input pixel with a 3-bit weight. The weights for each stages are stored locally in SRAM cells.

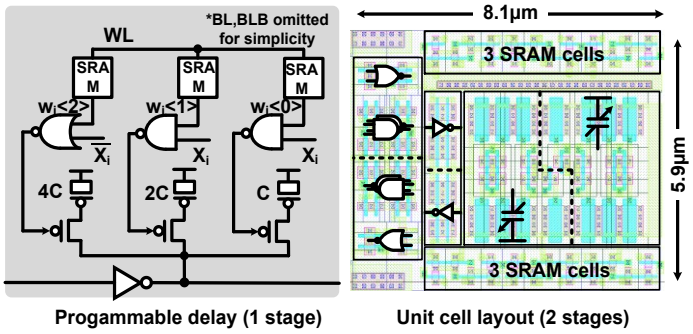


Figure 2.3: Detailed implementation of programmable delay stage and unit cell

Input pixels determine whether a stage is enabled, and weights determine how many capacitors are enabled to load the driver in that stage. Weight 100_2 is defined as weight zero. Positive, or exhibitory, weights are defined as less than 100_2 (i.e. $001\sim011_2$), turning on fewer load capacitors. Thus, reducing the delay of that stage. On the contrary, weights greater than 100_2 (i.e. $101\sim111_2$) represent inhibitory synapses, negative weights. Delay of all stages accumulates naturally in the DCO loop and the output is fed to an 8-bit counter. The counter increments every rising-edge of the DCO cycle. When the counter value reaches a target count, a spike is generated and the counter is cleared. The spike count of the other blocks will be recorded and used as the output of each DCO unit. The counting and threshold blocks can implement the integrate-and-fire using simple hardware. The measurement precision of the time based DCO circuit can be easily programmed by changing the spiking threshold. With a higher spiking threshold for instance, a smaller frequency difference can be detected at the cost of higher energy dissipation and decreased throughput.

The DCO circuit is also very robust against jitter, since the jitter will be averaged out over many DCO cycles. This is due to the first-order noise shaping principle of using

a DCO for a time-to-digital converter [17]. When the rising edge is traveling around the DCO at each stage some random noise will be present and manifest as noise in the channel of the MOSFET which affects the current, and subsequently the delay of the stage. This error is accumulated in each stage until the input to the counter is reached. The counter will sample and the pulse will continue. However, any deviation between measurements is not lost, but carried over to the next stage. This averaging in the discrete time domain is equivalent to first-order noise shaping in the frequency domain [17].

Figure 2.4 shows the overall architecture of the time-based neuromorphic core with 64 parallel DCO circuits. The array is divided into 8 groups, each consisting of 8 DCO circuits, to realize the local lateral inhibition feature discussed in section 0. Each DCO can be enabled or disabled independently allowing activation of any number of DCOs simultaneously. The neuromorphic core compares the raw spike count of each DCO to determine which neuron output is dominant.

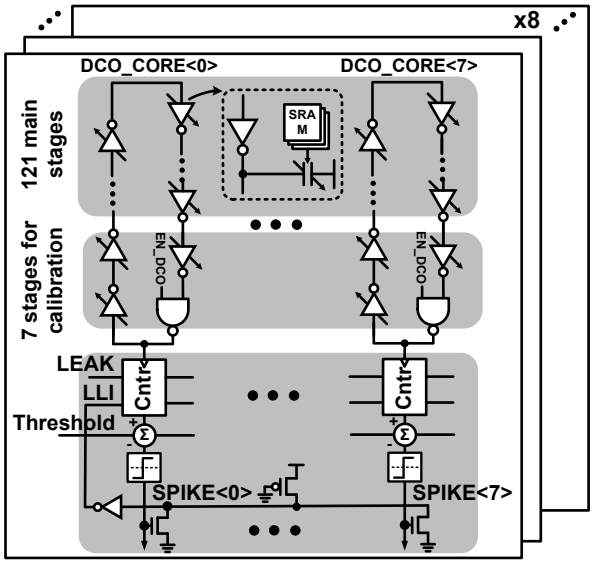


Figure 2.4: Time-based DCO neuromorphic architecture.

Due to process variation however, different DCOs have slightly different oscillation frequencies for identical inputs. This is due to variations in the channel doping of the transistors at manufacturing which manifests as different threshold voltages. If each DCO is composed of gates that have different threshold voltages the oscillation frequencies will not be matches. It is crucial that the DCO frequencies are uniform to start with to have a meaningful measurement for the ML application. Unlike process variation, voltage and temperature variation affect all DCOs uniformly, so although the inter-trial count may vary, the dominant neuron will stay the same under voltage and temperature deviations. Seven of the 128 DCO stages are reserved for frequency trimming while the remaining 121 stages are used for the MAC function. For frequency calibration, all DCOs are configured using nominal inputs and weights, and then the frequency counts are measured for a fixed time period. By tuning the weights of the 7 frequency trimming stages, it is ensured that the DC DCO frequencies are matched.

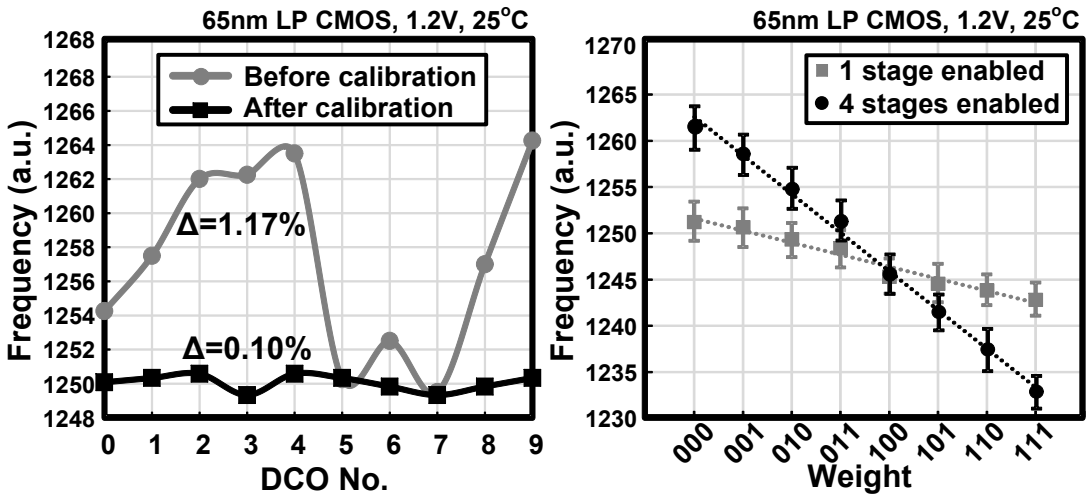


Figure 2.5: Measured DCO Frequency before and after tuning.

Measurement results in Figure 2.5 (left) confirms that after calibration the frequency variation of 10 DCOs reduces from 1.17% to 0.10%. Figure 2.5 (right) shows the mean and 3σ error bars of the frequency count when different number of stages are activated as the weight of each stage is swept. Measured results show adequate linearity.

2.2.2 *Leak and Local Lateral Inhibition Technique*

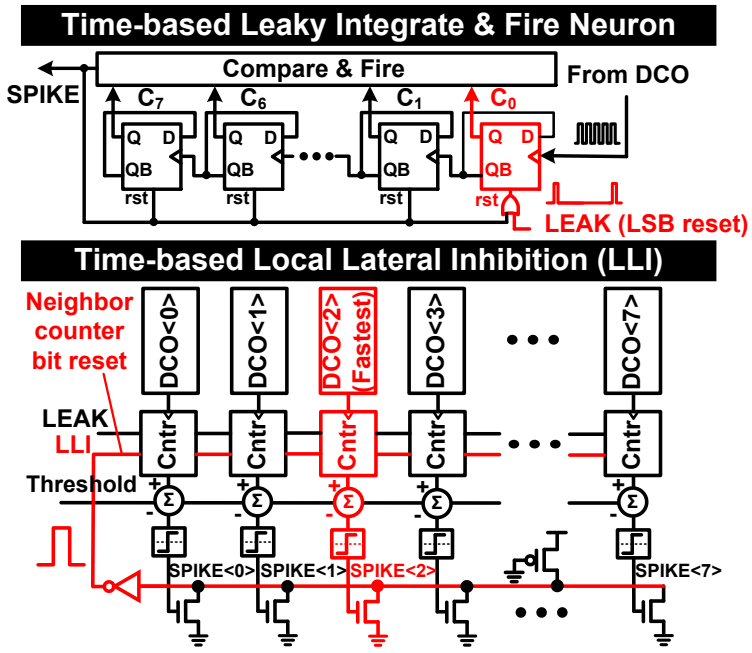


Figure 2.6: Illustration of time-based leaky neuron and local lateral inhibition

Neuro-inspired leak and local lateral inhibition (LLI) features are also implemented in our design, and they can enhance the contrast between neuron outputs. Figure 2.6 explains the concept of leak and LLI. When the leak feature is enabled, the LSB of the counter is reset periodically by a low-frequency LEAK signal. This has the effect of periodically reducing the accumulated count, imitating a leaky neuron [18]. Note that the period of the LEAK signal should be several times longer than that of the DCO in order to allow the count to increment. The main benefit of the leak operation is that it can increase the relative difference between DCO counts as shown in Figure 2.7(left). The frequency difference between the dominant DCO and the others becomes larger and this can be thought of as increasing the contrast.

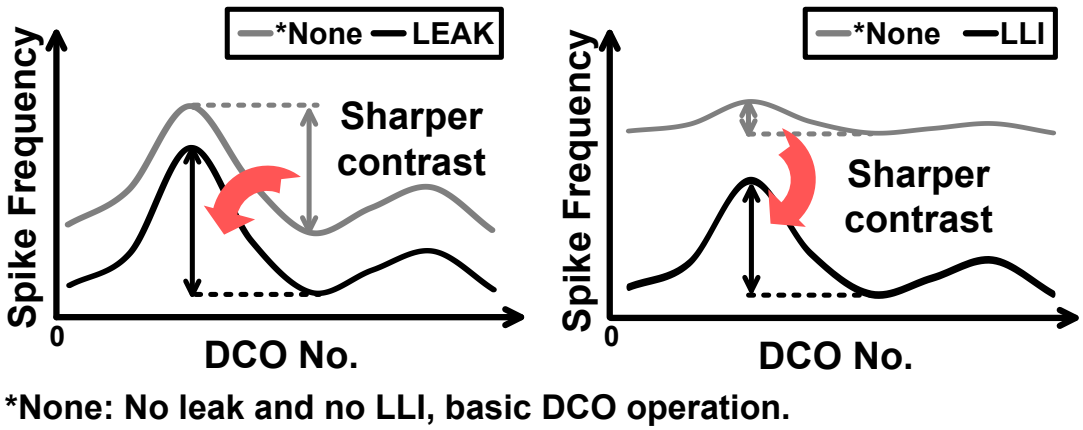


Figure 2.7: Effect of leak and LLI features.

Lateral inhibition is a phenomenon in which the dominant neuron strives to suppress the activation of its neighbors. In this design, eight DCO cores are grouped together to realize LLI. The inhibition amount (count decrease) is determined by which bits of the neighbor counter are reset. Once a DCO in the group spikes, an LLI pulse generated, which resets predetermined bits of the neighboring counters. The fastest DCO in the group resets the other DCOs more often than it is reset by the other DCOs, enhancing the contrast between different DCO outputs, which is illustrated in Figure 2.7 (right).

2.2.3 Application and Measurement Results

A test chip was fabricated in a standard 1.2V, 65nm LP CMOS process to demonstrate the time-based neuromorphic core. Due to die area constraints, a single core was implemented. However, a multi-core architecture can be realized to handle deep neural network algorithms by tiling additional DCO cores and operating them in parallel.

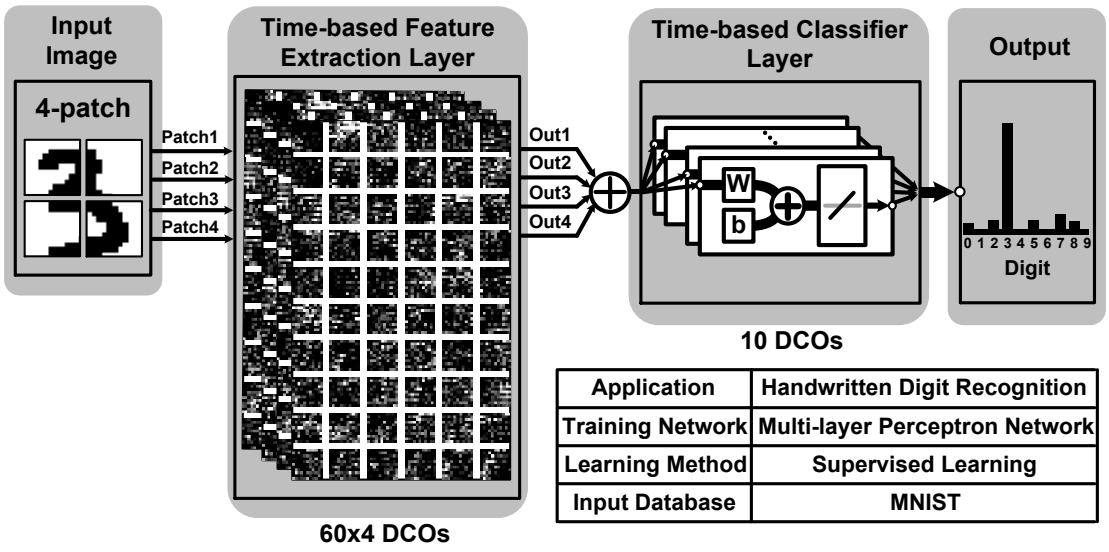


Figure 2.8: Multi-layer digit recognition test architecture and summary for time-based neuromorphic chip demonstration.

Figure 2.8 shows a 2-layer test architecture for handwritten digit recognition used to showcase the versatility of the proposed core. Handwritten text images were obtained from the MNIST database [19]. The original image size is 28x28 pixels. First, the image is reduced to 22x22 pixels by removing three pixels on each side as they contain little information. Each image is then divided into four regions so that they can be assigned to different cores for increased throughput. The first layer of the neural net can extract 60 distinct features from each patch. The outputs of the four patches from the first layer are summed, encoded, and used as the inputs for the second classifier layer. Weights of both layers are trained off-chip using supervised learning and downloaded to the chip.

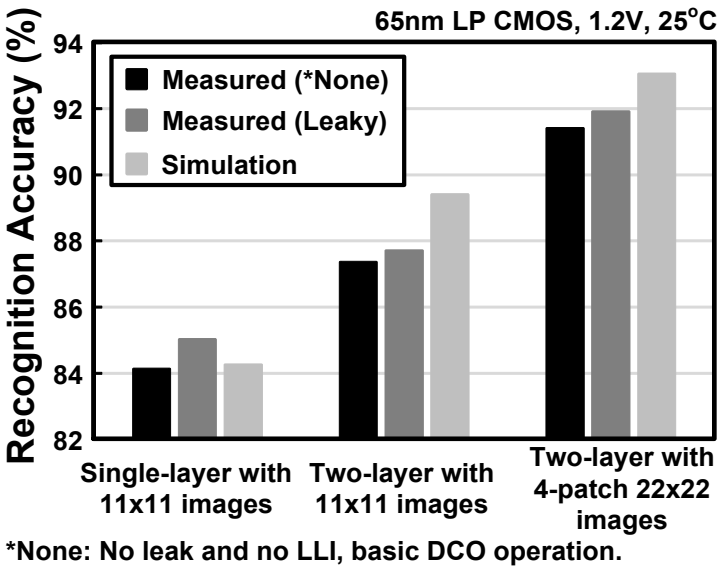


Figure 2.9: Measured application results.

Figure 2.9 compares the accuracy between different configurations. The 2-layer architecture with 4-patch inputs (=22x22 pixels) achieves a recognition accuracy of 91.4%. With the leak feature enabled, the accuracy increases modestly to 91.9%. The measured accuracy is comparable to software simulation results. As seen from the measurement results in Figure 2.9, the recognition accuracy of a single-layer architecture increases from 84.1% to 85.0% after enabling the leak feature, while the accuracy doesn't improve as much in the two-layer architecture. This is because in the two-layer architecture, there are more weights available to improve the contrast between the neuron outputs at training. This makes the leak feature less effective.

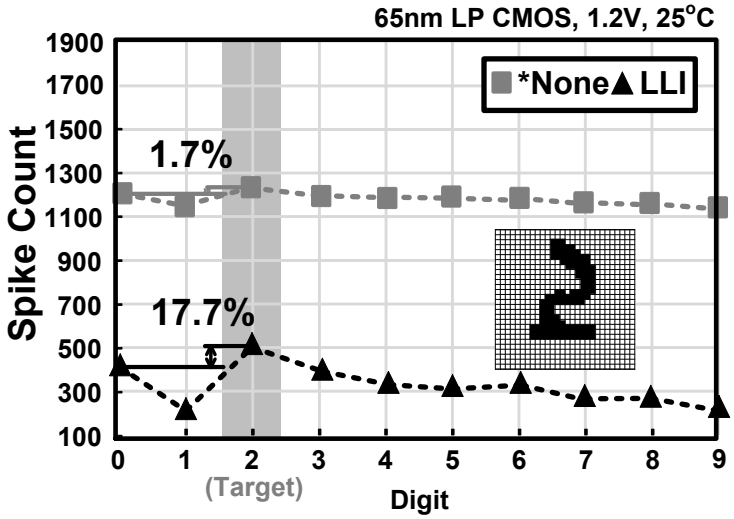


Figure 2.10: Example of measured output from one digit.

Figure 2.10 shows the output spike count with and without the LLI feature for an image containing handwritten digit "2". The spike count difference between digit "2" and runner-up digit "0" is improved from 1.7% to 17.7% using LLI. This greatly improves the confidence in the prediction.

Table 2.1: Performance Comparison

	This work	ISSCC'16 [20]	VLSI'16 [21]	ISSCC'15 [22]	ISSCC'14 [23]	CICC'11 [14]
Application	Hand writing recognition	Object detection + intention prediction	Object recognition	Big data analysis	Pattern recognition	Hand writing recognition
Function	Multi-layer perceptron network	Deep neural network	Deep neural network	Deep neural network	Unsupervised online clustering	Restricted Boltzmann Machine
Circuit Type	Time-based	Analog + Digital	Digital	Digital	Analog + Floating gate	Digital
Technology	65nm	65nm	40nm	65nm	0.13μm	45nm
Area	0.24mm^2 (64 DCOs)	16.0mm^2	1.4mm^2	10.0mm^2	0.36mm^2	4.2mm^2
Voltage	1.2V	1.2V	0.9V	1.2V	3V	0.85V
Frequency	99MHz (nominal DCO freq.)	250MHz	240MHz	200MHz	8.3kHz	-
Power	320.4 μW/DCO	330mW	140.9mW	185mW	11.4μW	45pJ/spike
Performance	99M ÷ N spikes/s/DCO (*N=spiking threshold)	502.0GOPS	898.2GOPS	411.3GOPS	0.012GOPS	-
Power Efficiency	309G ÷ N spikes/s/W (*N=spiking threshold)	862GOPS/W	6.37TOPS/W	1.93TOPS/W	1.04TOPS/W	-

*N=16 in our measurements

Table 2.1 shows the performance comparison with recent neuromorphic chip designs [20], [21], [22], [23], [14]. It is worth noting that an apples-to-apples comparison between our time-based scheme and traditional ASIC chips can be tricky. Here, metrics (e.g. spikes/s/DCO) specific and relevant to our design are presented. The proposed DCO array can generate $3.09 \times 10^{11}/16 = 1.93 \times 10^{10}$ spikes per second per watt, for a spiking threshold value of 16.

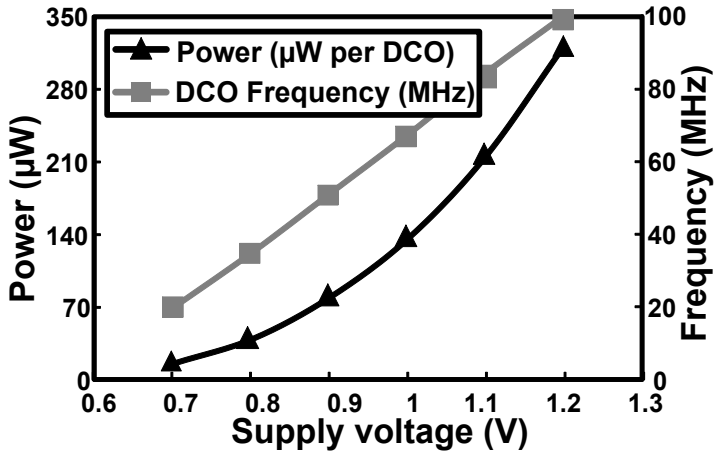
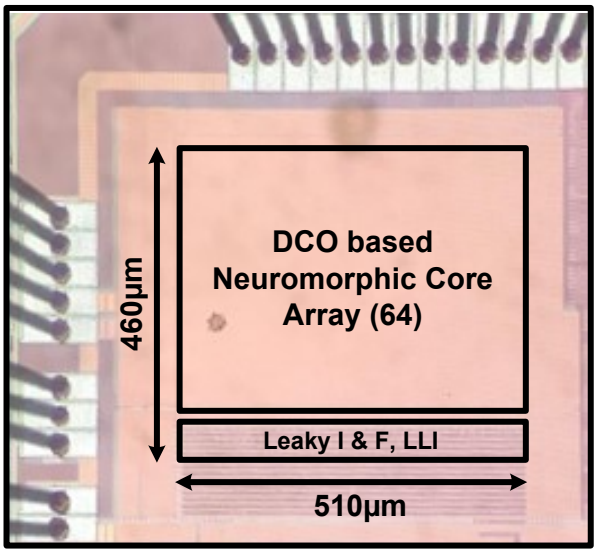


Figure 2.11: Measured power consumption and DCO frequency for the I&F core



Technology	65nm LP CMOS
Core Size	510μm x 460μm = 0.24mm^2
VDD range	0.7V~1.2V
# of Neurons	64 (single core)
# of Synapses	8192 (8K)
Throughput	746Mpixels/s ÷ N (1.2V) 148Mpixels/s ÷ N (0.7V) (*N=spiking threshold)
Power	320.4μW (per DCO, 1.2V) 17.5μW (per DCO, 0.7V)

*N = 16 in our measurements

Figure 2.12: Die photo and performance summary

Figure 2.11 shows the measured power consumption and DCO frequency under different supply voltages. The test chip has a wide operating voltage range of 1.2V to 0.7V. The DCO circuit oscillates at 99MHz consuming 320.4µW under a nominal 1.2V supply. At 0.7V, the DCO oscillates at 20MHz consuming 17.5µW. Figure 2.12 shows the chip micrograph and performance summary.

2.2.4 Conclusions

In this section the idea of implementing neuromorphic function purely in time domain with programmable delay stages based on a leaky integrate-&-fire mechanism was presented. Brain-inspired leak and local lateral inhibition (LLI) are introduced to increase the contrast and confidence in the predictions. The time-based neuromorphic core is tested with digit recognition application and achieves 91.4% recognition accuracy. The energy-efficiency and versatility of the presented time-based DCO neuromorphic core makes it a promising building block for future large scale deep neural network applications.

2.3 An Energy Efficient Time-Based One-Shot Neuromorphic Chip

2.3.1 One-Shot Neuromorphic Architecture

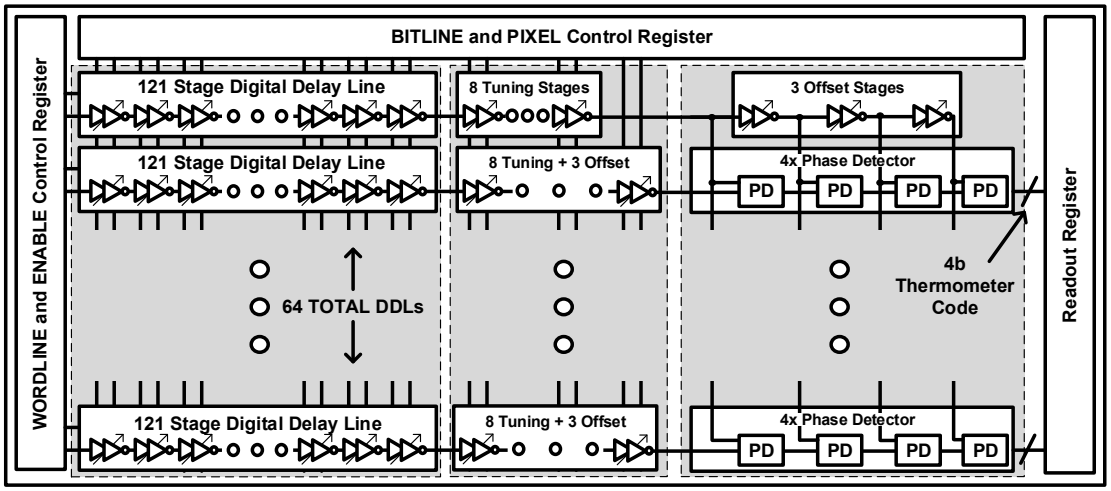


Figure 2.13: Top Schematic of the one-shot time-based neuromorphic core

Conventionally, Boolean computations are used to realize arithmetic operations in hardware. However, time domain circuits can also be used at an advantage of lower area and power per processing unit, and reduced design complexity. The kernel of all ML algorithms can be distilled into a dot product shown in Equation 2.1. The high level architecture is shown in Figure 2.13. An input pulse is presented on the left side of the core and the delay of each stage is modulated based on the application inputs. It is described as one-shot because each pulse gets evaluated once.

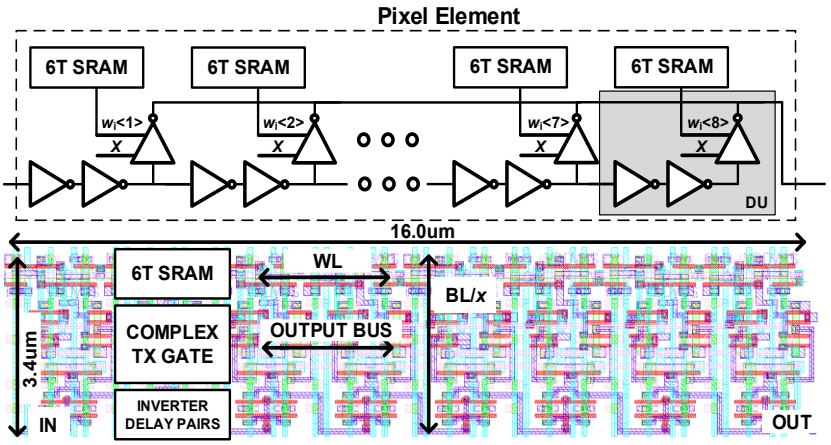
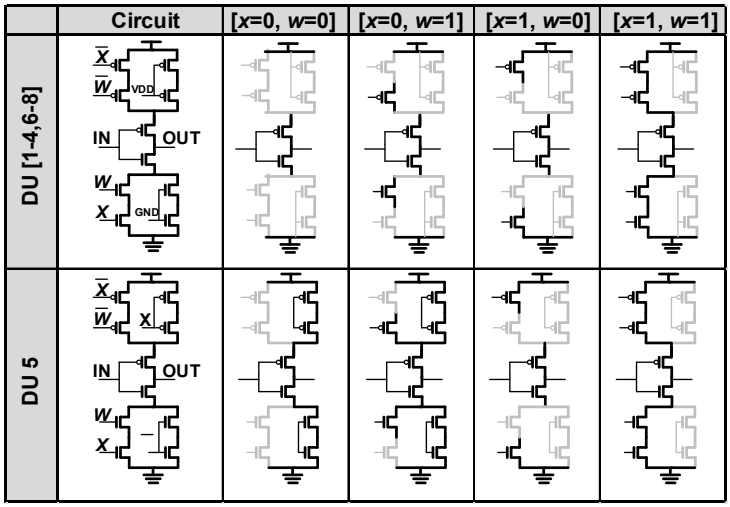


Figure 2.14: (Top) Schematic of pixel stage. (Bottom) Layout of pixel element.



Weight	Delay
-3	8
-2	7
-1	6
0	5
1	4
2	3
3	2
4	1

Figure 2.15: (Left) Complex tristate wiring. (Right) Trained weight to DU stages

Each stage has 8 delay units (DU) with output taps which the pulse travels through as seen in Figure 2.14. The number of DU enabled depends on the one-hot encoded weight, stored locally in SRAM cells, and the input pixel, which is applied across the array on the bitlines. Each DU has two inverters to retain consistent polarity between stages. This is critical in the event that the rising and falling propagation matched, as well as ensuring correct polarity at the TDC. The output tap is realized as a complex tristate gate and the functionality is described in Figure 2.15. The first column shows the circuit schematic and

corresponding connections between the different DU. The right four columns show the activated paths, shown with black lines, depending on the values of the input, x, and weight, w. DU_5 is the nominal stage delay, and is activated through the right branch of the circuit when the input bit is off, representing "zero delay." Figure 2.15 (right) shows the mapping between the algorithm-trained weights and the delays realized in the chip at each stage. When the input, x, is present the left branch is enabled in the DU corresponding the weight bit of the stage. Larger positive weights map to shorter delays relative to the reference DDL, and conversely negative weights correspond to longer delays. The accumulation in the MAC is achieved naturally as the pulse passes sequentially through the DDL, stage by stage. The layout of each DU in the stage is pitch matched to a 6T SRAM so the layout is regular, compact, and scalable. The bias vector is applied in the same way for the last eight units. Additionally, it can be used to tune process variation, so that during evaluation those pixels are always activated.

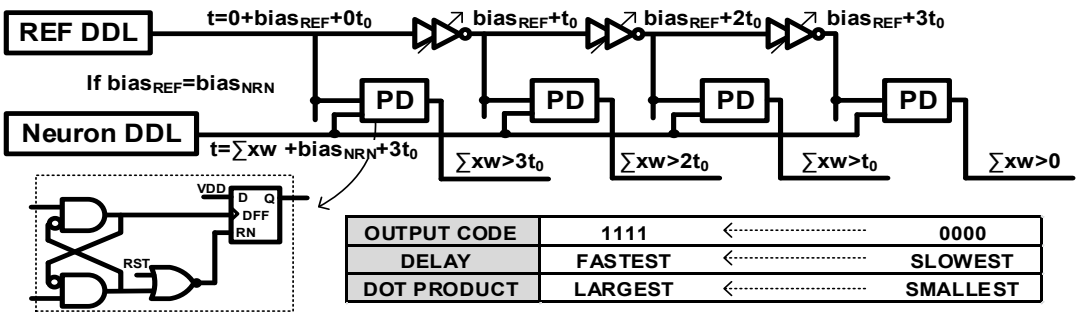


Figure 2.16: Timing details of the 2 bit TDC.

Figure 2.16 shows the relationship between the time domain computation in the chip and the expected arithmetic output. The phase detector output maps roughly to the rectified linear unit (ReLU) transfer function. When the reference pulse beats the neuron rising edge all four thermometer bits are zero, regardless of the magnitude. The transfer

function between the four bits is linear and then clips, or saturates, once the neuron pulse is faster than all the offsets.

2.3.2 TDC Performance Analysis

The delay of time-based circuits can be tuned to cancel out inter-DDL process variation. Measured one-time calibration results are shown in Figure 2.17. Calibration was performed by evaluating each DDL and measuring the DDL phase detector output. After each evaluation, the bias bits (8 Tuning in top DDL shown in Figure 2.13) of the reference was increased and the process repeated.

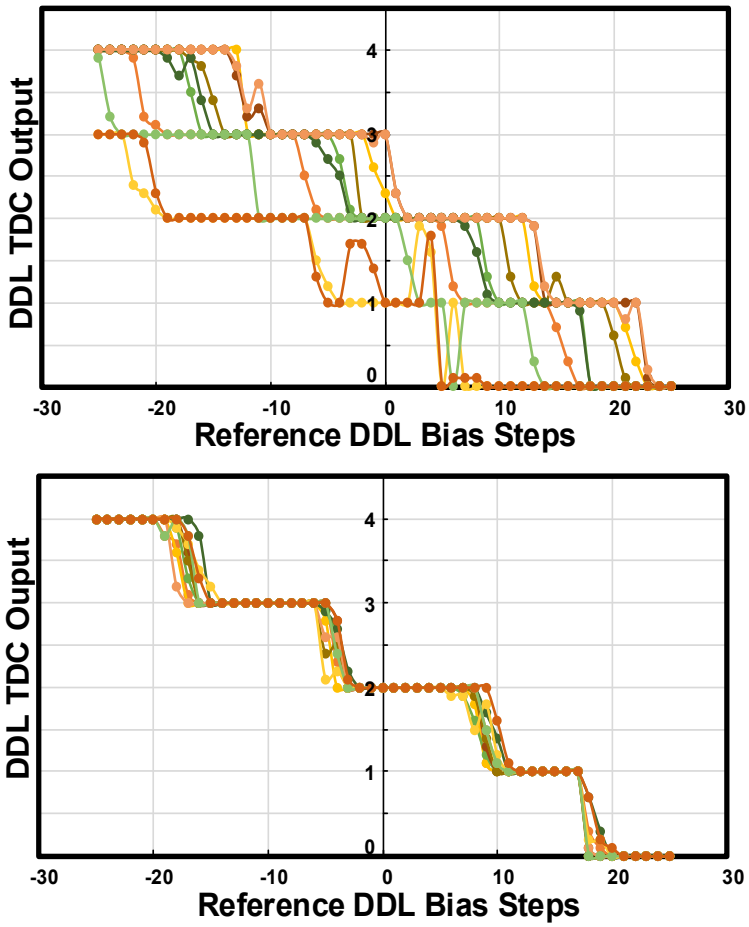


Figure 2.17: Measured data from chip calibration across 10 DDLs.

At each bias point, 10 additional evaluations were run due to quantify trial-to-trial temporal noise, seen as the slope between TDC levels. No other measurements are averaged in the following sections. The reference bias point at which the PD of each DDL trips is applied to the tuning bits of the respective bias to align all DDLs, thus, compensating process variation. The average of the 10 trials is plotted in the Figure 2.17. The inter-DDL spread before calibration is approximately 21 reference bias steps, or tuning steps. After calibration, the spread was reduced to less than three tuning steps. The curves are mostly monotonically decreasing which is expected even though there is meta-stability when the phase of the output and reference DDLs are nearly matched. This supports the effectiveness of the proposed time-based MAC methodology.

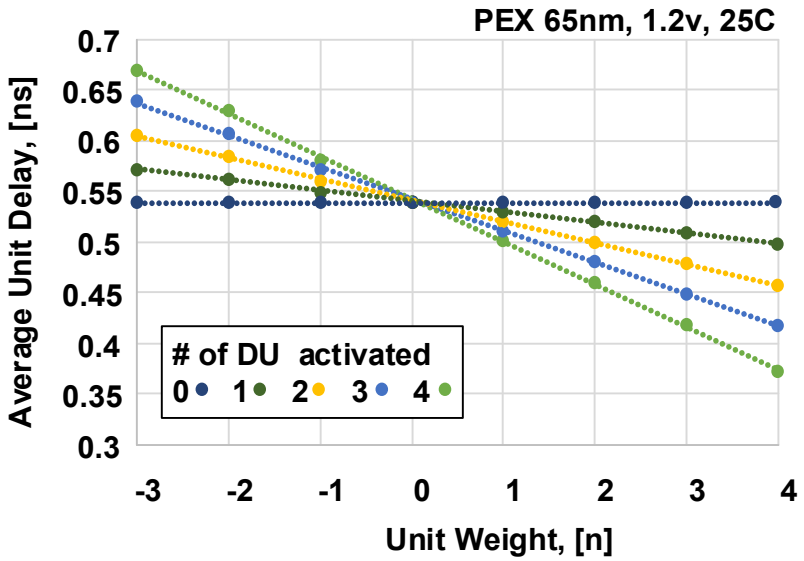


Figure 2.18: Post-layout simulation of DU linearity.

Figure 2.18 shows the simulated average unit delay as a function of the weight in each unit. In this simulation, the extracted layout of a four stage DDL was used to measure the delay of a single weight change. Each stage has the weight programmed from [-3, 4],

corresponding to the x-axis, and the number of active stages is swept from none to all four, corresponding to the different series. This confirms there are no systematic biases between the different weights.

Using both measured and simulated tuning curves it is possible to quantify the TDC performance [24]. It should be noted that while the TDC performance is important, the trade-offs between area, power, and application performance are paramount. For each bit of increased TDC resolution the area and power doubles. This incentivizes the designer to use a minimalistic design to keep overhead low while still managing to meet the requirements of the application, discussed further in section IV. The TDC gain is the slope of the output code to input code. The ideal slope is described by Equation 2.2,

$$k_{TDC} = \frac{1}{T_{LSB}} \tag{2.2}$$

where T_{LSB} is the minimum time interval that can be measured, which in this case would be 12 tuning bits. The gain error describes the difference in the last output code to the expected result based on the gain, quantified by Equation 2.3 [24].

$$E_{gain} = \frac{1}{T_{LSB}}(T_{1111} - T_{0001}) - (2^N - 2) \tag{2.3}$$

In this work, T_{LSB}=12 tuning bits based on the calibrated tuning curve, thus $E_{gain} = -\frac{1}{4}$. Returning to the TDC gain, one can now use the gain error to accurately estimate the actual gain error.

$$k_{TDC} = \frac{1}{T_{LSB}}\left(1 - \frac{E_{gain}}{N_{levels}-2}\right) = \frac{3}{32} \tag{2.4}$$

This is 12.5% steeper than the ideal gain due to the reduced phase window at output code 0001. This could be due in part to the reduced load seen after the third reference buffer and rectified in future work by adding a dummy load to better match the delays of each branch of the TDC.

The previous paragraph studied the performance metrics that affect the linear performance. Next, the non-linear performance due to process variation and noise will be quantified. The total delay can be described as:

$$t_n = nT + \sum_{i=1}^{n} \varepsilon_i \qquad \textbf{2.5}$$

where ε_i is the delay error caused by process variation at stage i. In Table 2.2, μ is equivalent to the nT term, where n is chain length and T is the DU delay. If all the delay units are independent but derived from the same distribution, the standard deviation of the total time is $\sigma(t_n) = \sigma(\varepsilon)\sqrt{n}$.

Table 2.2: DDL Delay for increasing chain lengths

PEX 65nm, 1.2V, 25C

t_{fall}			
Chain Length	**μ**	**σ**	**σ/μ**
4	2.25n	34.6p	0.0154
16	9.24n	68.3p	0.0074
128	72.92n	194.6p	0.0027
t_{rise}			
Chain Length	**μ**	**σ**	**σ/μ**
4	2.25n	37.4p	0.0167
16	9.24n	67.1p	0.0073
128	73.00n	200.5p	0.0027

*Estimated via square root law [11]

This is supported by Table 2.2, where chains of length 4 and 16 were simulated after parasitic extraction for 100 Monte Carlo samples. The distribution of the delays were normally distributed and the standard deviation follows the square root law. Chain lengths of 128 were estimated by the square root law. This has two consequences; the first being a shorter delay chain will have less variation. This is better, however it has lower efficacy because there are fewer elements that can be multiplied at once reducing the throughput and increasing power. The second consequence is that the rate of increase decreases as more stages are added to the delay chain. This means that an increase of 8x stages only results in an increase of 2.8x in the standard deviation. From Figure 2.18 it is estimated that the tuning step delay is 10.5ps, which makes the standard deviation equal to roughly 18.5 tuning steps, or 1.53 output codes. This could be reduced by increasing the transistor size, W, to reduce the Johnson-Nyquist noise, where in saturation the power spectral density of the drain current is shown in Equation 2.6[25].

$$S_i = 4kT\frac{2}{3}\frac{W}{L}\mu C_{ox}(V_{GS} - V_T) \quad \textbf{2.6}$$

Reducing noise comes at a cost of higher power consumption. As temperature and voltage increases it can be seen the current will increase. These shifts would be seen at the global level since all DDLs share the same voltage supply. Additionally, temperature gradient across the small, dense array would be unlikely and if there was a global temperature shift, it would affect all DDLs together. Another method could utilize a closed-loop ring oscillator which integrates the noise over multiple cycles which reduces the total error at a cost of lower throughput and higher power per prediction [6]. With these trade-offs

identified the proposed circuit strikes a balance between performance and a light-weight solution.

2.3.3 Dynamic Threshold Error Correction (DTEC)

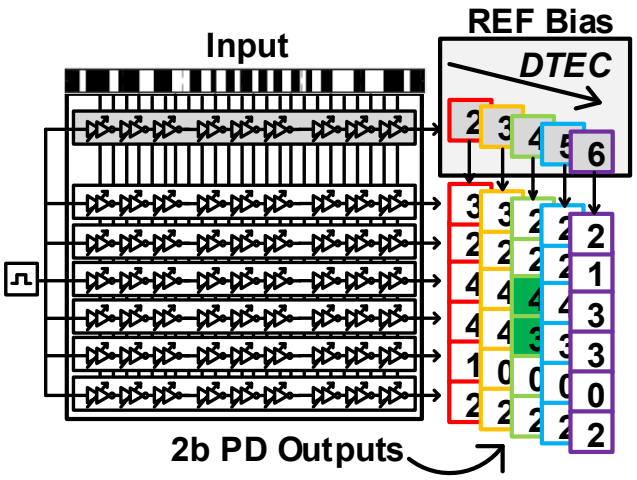


Figure 2.19: DTEC operating concept illustrated where reference DDL bias is swept to boost TDC resolution.

In the prior section the 2 bit TDC was described. It was selected due to the optimal tradeoff between small area and low power, and strong architecture performance. In networks with "winner-take-all" topologies, such as the last stage of classification networks, ambiguous predictions can occur. Unclear outputs in this work can stem from limited resolution between phase detector trip-points or activity outside of the range of the phase detector. To mitigate this issue, Dynamic Threshold Error Correction (DTEC) technique is proposed, which increases the effectiveness of the 2bit TDC. As shown in Figure 2.19, when two or more DDLs have the same output, DTEC works by increasing the threshold bias delay which moves the trip point of the phase detectors. DTEC is dynamic due to the fact that the bias sweep would be terminated after the third evaluation, when the dominant DDL was identified from the phase detectors. Additionally, DTEC can

be stopped after a fixed number of steps if no dominant DDL emerges to conserve power. In Figure 2.21the top row of colormaps shows ambiguous predictions from the core, while successive rows show the output as DTEC is applied. Red rectangles highlight where DTEC has successfully identified the target.

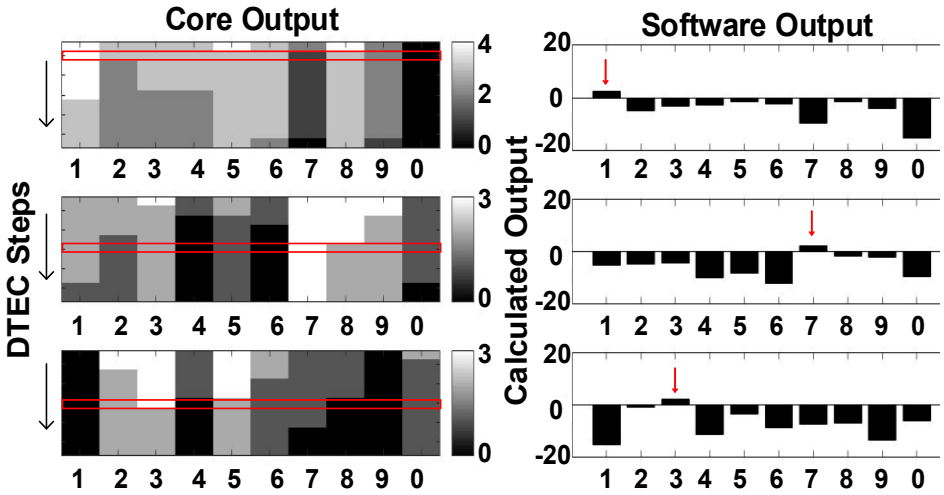


Figure 2.20: Measured effectiveness of DTEC, first rows show ambiguous prediction and as bias is swept winner is isolated.

Figure 2.21 plots the distributions of outputs from the intermediate layers in a two-layer dense neural network with 30 hidden units and 10 output units for all 10,000 test images in the MNIST benchmark [19]. The left column corresponds to the output with full precision weights and the right column corresponds to our rounded three-bit weights. The first row shows the network model used for the analysis. The second row displays the distribution of the MAC output of all 30 hidden units after the ReLU transfer function. The third row shows the winner take all (WTA) output of network, but each curve plots the sorted output instead of each unit (i.e. the correct outputs for the ten cases are grouped to

one unit). The x-axis scales are not normalized to a unit weight. In the 3b precision network, the full precision weights have been scaled up to match the DU range (i.e. [-3,4]).

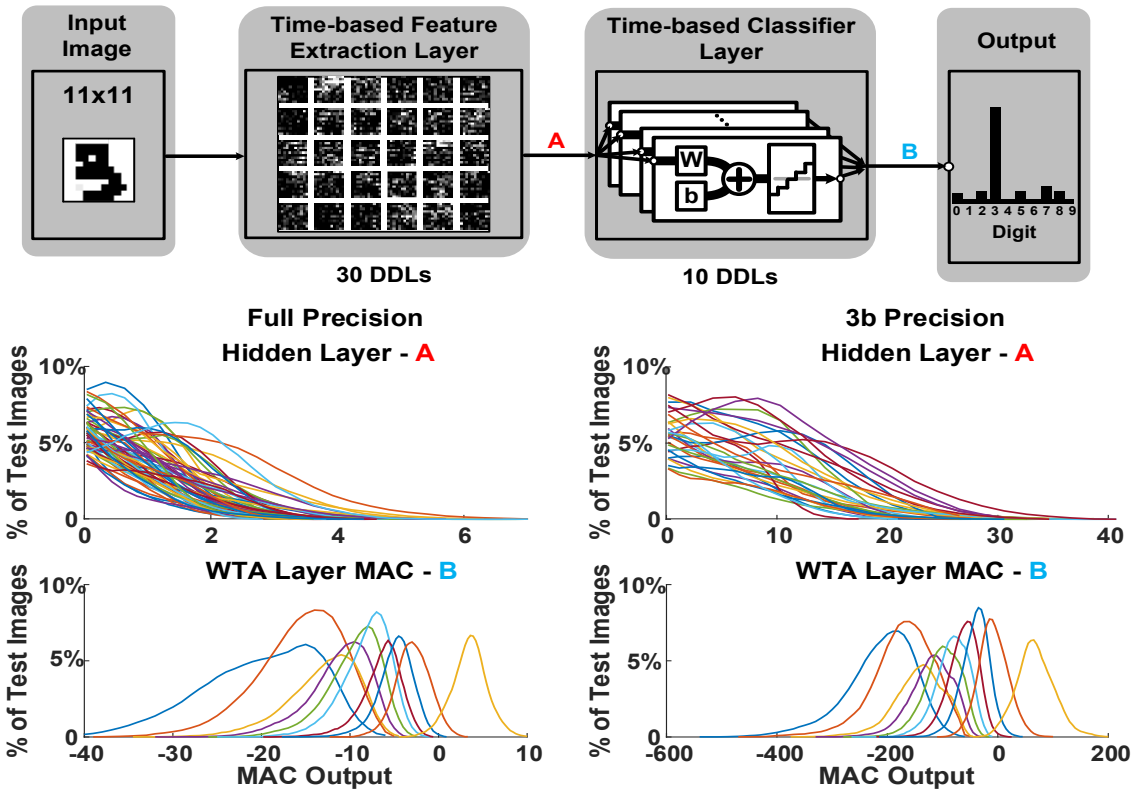


Figure 2.21: Distribution of activation outputs in a 2 layer neural network.

These curves support the assumption that a 2-bit TDC can cover the entire output range because according to Figure 2.17 the width of the PD is 40 units on the x-axis. The hidden layer output would be contained inside that range. In the hidden layer, the results are approximately zero-centered prior to the ReLU activation, but have a large range. Units in neural networks must have zero-centered activations otherwise the predictions would be biased resulting in reduced learning capacity. If the TDC had a unit step of 1 tuning bit (equal to 1 step of the x-axis) this would require at least a 6-bit TDC for each DDL. The area overhead would render this solution infeasible. Additionally, due to the effectiveness

of the training, the correct prediction output histogram has very little overlap with the remaining predictions. This outcome can be leveraged because in the majority of the cases a high precision TDC will not provide additional information when the only relevant outcome is which unit has the highest activation. Another observation is that full precision and fixed point traces match closely. There is a modest amount of spread between the fixed and full precision hidden layer outputs. Nearly all hardware implementations utilize fixed point weights and this is an acceptable transform as the curves match.

Analysis of results from the 3b single layer application (section 2.3.4) show that by applying just two DTEC steps 81.64% of the correctible errors are recovered. This comes at a cost of just 41% additional evaluations per image. After the one-shot evaluation, 73% of all images have a dominant output. The remaining 2,668 images begin DTEC. After the first step 46% are resolved and 37% after the second step leaving less than 1,000 images ambiguous. Thus, 4,108 DTEC evaluations improves the total accuracy from 69.16% to 82.14%. If three DTEC steps are applied 88.8% of errors can be recovered at an overhead of 51%, demonstrating the dynamic scalability of the technique. Hardware results show that DTEC is an economical and scalable approach to significantly improve application performance.

2.3.4 Application and Measurement Results

The core was evaluated on the MNIST benchmark [19]. Figure 2.22 shows the comparison of classification accuracy on an 11x11 image for single and two layer networks between expected simulated software results, one-shot evaluation, and DTEC. To reduce the 28x28 grayscale images to 11x11 binary images, 3 pixels are sliced from all four sides

of the image. Then, a fixed resizing command is applied, and finally the pixels are binary thresholded.

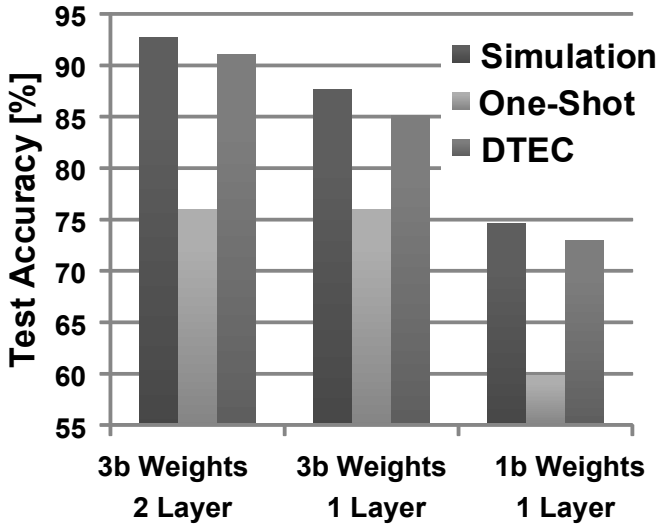


Figure 2.22: Measurement results on MNIST application

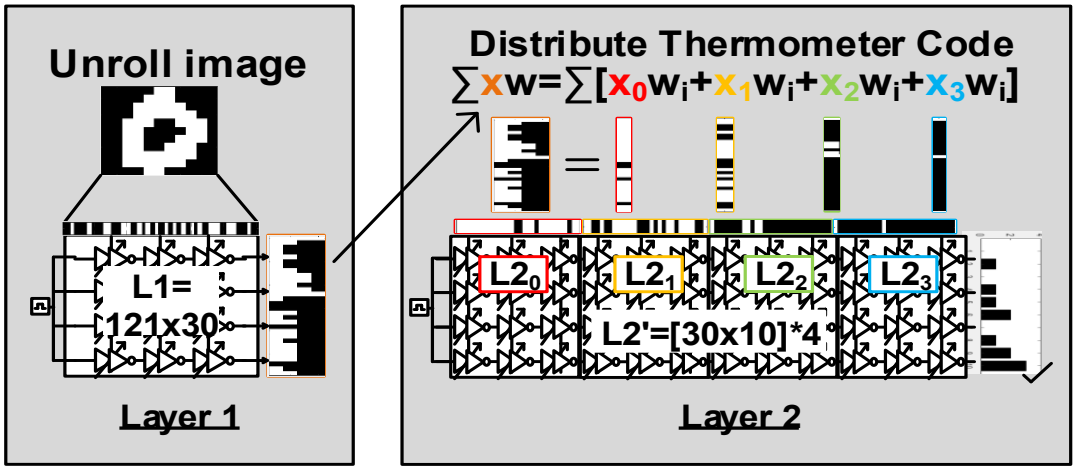


Figure 2.23: Dataflow for multilayer neural network with binary inputs.

Figure 2.23 shows how the core can be used in a multi-layer deep neural net application. Each bit of the thermometer code is expanded as the input in the next layer. The input is divided into four segments, and the weight matrix is copied four times ($L2_0$-$L2_3$), which gives each bit equal weighting. In the example shown in Fig. 12, 30 neurons in layer 1 yield a 120 bit input to layer 2. By applying DTEC, the ambiguous results are

almost completely recovered and the slight loss in accuracy is due to output differences smaller than a single tuning bit. Figure 2.24 shows the tradeoff between power consumption and nominal stage delay for various supply voltages. Power is kept exceptionally low because rarely are more than two stages switching at a time in a DDL. A wide operating voltage range is enabled, due to the all-digital time-based design choices.

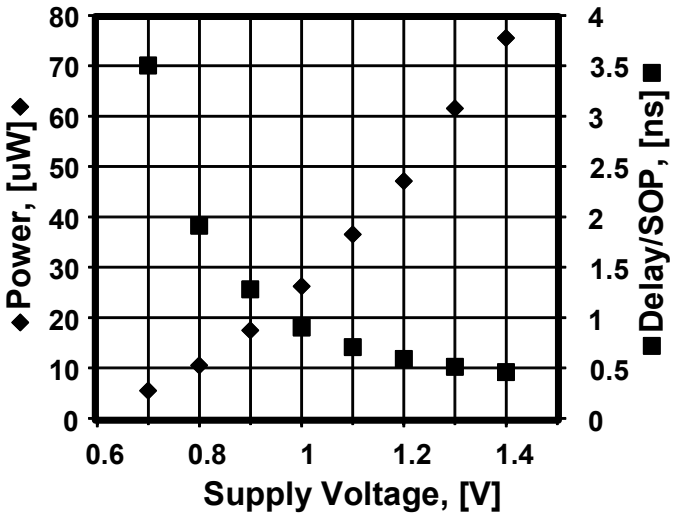
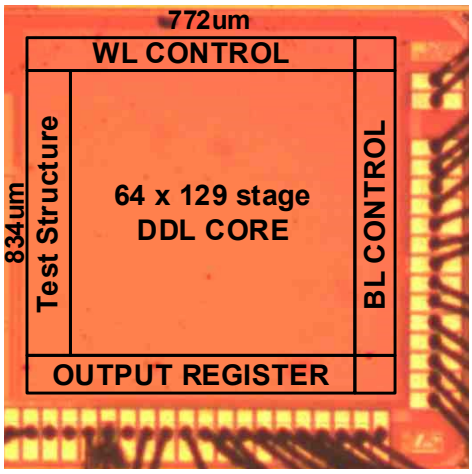


Figure 2.24: Power dissipation of a single DDL and delay/stage across VDD.

If the design incorporated pipelining, it could achieve even greater throughput. That is, multiple pulses could be pushed into the DDL and the input could shift as well. This is ideally suited for convolutional nets where a weight filter slides across an image. In this case, the image could slide across the weights while input pulses are applied to the DDL. Die photo and design specs are highlighted in Figure 2.25.



Process	65nm LP CMOS
Core Area	0.644mm^2
VDD	0.7-1.4V
# of Neurons	64
# of Synapses	8256 (8K)
Throughput	1.7Gpixels/s/DDL
Power	47.1uW/DLL
Energy/SOP	27.6fJ

Figure 2.25: Chip micrograph and chip summary (metrics reported at nominal supply).

Table 2.3: Performance Comparison Table

	This Work		A-SSCC'16 [26]	CICC'17 [6]	ISSCC'17 [7]	ISSCC'17 [27]	ISSCC'16[28]	ISSCC'16[9]	Science'14[4]
Chip Architecture	Time-Based		Time-Based	Time-Based	Digital	Digital	Digital	Sw. Cap	Digital
Algorithm Target	FCDNN & CNN		FCDNN & CNN	FCDNN & CNN	FCDNN & CNN	FCDNN & FFT	CNN	CNN & SGD	FCDNN & CNN
Technology [nm]	65		65	65	28 FDSOI	40	65	40	28
Chip Area [mm^2]	0.644		3.61	0.24	1.87	7.1	12.25	0.012	430
Precision* [b]	[B,T,2,3]		B	3	[4-16]	[6-32]	16	3	[B,T]
On-Chip SRAM [kB]	8.06		20	3	144	270	181.5	[-]	256MB
VDD [V]	1.2 (Nom.)	0.7 (E_{MAX})	1	1.2	0.6	0.65	0.82	1	0.85
Frequency [MHz]	1700	285	23041	792	200	19.3	250	1000	0.001
Energy Efficiency** [TSop/s/W]	36.2	52.4	48.2	2.47	5.0	0.19	.18	3.86	0.04
Hardware Efficiency [GE/PE][1]	38.4		76.5	33.2	7456	18269	50637	288	6.5

*B=Binary, T=Ternary **Synaptic Op=MAC

Table 2.3 shows competitive performance compared with state of the art [26] [6] [7] [27] [28] [9] [4]. All comparisons are made at the highest reported energy efficiency operating point. It should be noted in this chip, 1 SOP is defined as a 1b input × 3b weight MAC without DTEC. Compared to [26] the energy efficiency would be 3x higher than reported. Additionally, [7] reports peak energy efficiency at 4b with 30-60% sparsity which they claim is present in convolutional neural networks. If the power supply can be tuned, very economical energy efficiency can be achieved at 8.7% improvement over [26], and modest gate equivalent count for each processing unit coming in at half the size of [26]. The gate efficiency compared is similar compared to [6]. This is interesting because the

capacitive weights and binary encoded weights stored in local SRAMs are only slightly smaller than one hot encoded SRAMs, linearly unrolled inverters, and tristate output gates. One-hot is less compact, but does not required decoding which causes overhead to control the capacitive connections in [6]. This chip is scalable in voltage, weight resolution, and is versatile in that it is able to tackle fully connected deep networks as well as convolutional nets.

2.3.5 Conclusions

This section described a time-based neuromorphic core based on one-shot DDLs in 65nm LP CMOS and proposed an error recovery technique, DTEC. It uses inverter delays to compute the dot product kernel, making it ideally suited for ML applications. The proposed core is validated on the MNIST dataset and achieves near simulated prediction accuracy on single and multi-layer networks after applying our error correction technique, DTEC. Maximum energy efficiency of 54.2TSOPs/s/W with 3b resolution at 0.7V makes the proposed architecture attractive for edge devices.

2.4 Summary

In this chapter two time-based designs for machine learning have been described. Section 2.1 began by introducing time-based computing for neuromorphic computing and studied the prior art by looking at different architectures that have been proposed in the literature; digital SoCs, analog SRAM solutions, analog computing, and crossbar arrays in SRAM and non-volatile memories. Next, a ring oscillator time-based machine learning chip was presented which achieved 91.4% accuracy on the target application and consumed 320.4μW at 1.2V. The main drawback of the ring oscillator was the many cycles required to get an output, which the next chip presented solved. The DDL one-shot chip achieved 54.2TSOPs/s/W making it one of the most energy efficient processors reported in the literature. DTEC was proposed which enables a scalable approach to limit design complexity and trade-offs increased accuracy for additional evaluations which reduces the energy efficiency. These two chips make a very strong case for time-based neuromorphic systems in the future. Both designs are able to achieve such low power consumption due to the unique multiplication structure that only toggles a single gate at a time, as well as the intrinsic addition through time accumulation. This architectural approach is not only simple and elegant, but has the potential to unleash the utility of machine learning out of the cloud and into every device.

Chapter 3. Time-based Graph Computing

3.1 A 40×40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control

3.1.1 Introduction

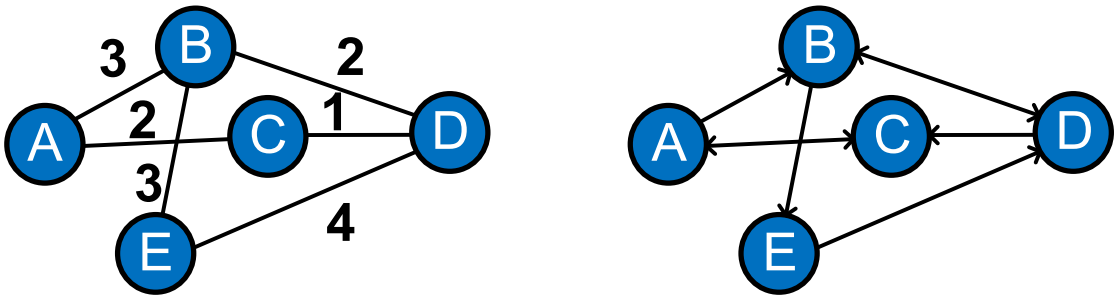


Figure 3.1: Examples of an integer weighted undirected graph (left) and unweighted directed graph (right).

A graph can be thought of as a set of objects, vertices, and their connections, edges. Edges can be directed meaning a one-way connection, or undirected. Additionally, edges can have weights which incentivizes certain paths. The left graph has undirected connections with weights. The right graph has directed edges and this can be seen by the arrows on ends of the edges. Graphs as an algorithmic construct are not very interesting. Their utility is derived from their ability to solve real world problems. If problems that manifest in the real world can be mapped onto specific structures, then known algorithms can be applied to solve them. Single-Source Shortest Path (SSP) problems have a rich history of algorithm development [29] [30] [31]. One of the most widely used frameworks for SSP is Dijkstra's Algorithm [30]. Dijkstra's is considered a greedy algorithm; it always

searches the best available option. It does this by using a priority queue. Nodes are added to the queue with priority based on how far they are from the source node. Smaller distances signify a possibility to find a shorter path to the target and are popped from the queue before others. One advantage of Dijkstra's is that it is guaranteed to find the shortest route because at each node the shortest path to it is recorded as the algorithm progresses.

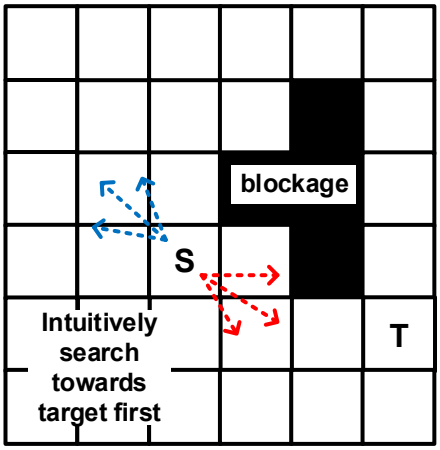


Figure 3.2: Illustration of the intuition of the A* heuristic

In most applications there is no reason that all directions of search should be treated equally. This intuition is shown in Figure 3.2. Arrows pointing towards *T* clearly should be searched before those cells in the direction of the arrows pointing away from *T*. This is precisely the modification that the A* algorithm [31] makes to Dijkstra's.

$$Cost(n) = F(n) + H(n) \qquad \textbf{3.1}$$

The cost assigned to a node, *n*, in the priority queue is shown in 3.1. *F(n)* is the actual distance of the current node from the source; the cost assigned in Dijkstra's algorithm. *H(n)* is a heuristic that predicts the distance of the current node to the target.

Figure 3.3: Manhattan Distance as the heuristic in A* SSP

Figure 3.3 details a common heuristic used commonly referred to as the Manhattan distance in reference to the dense, orthogonal streets in Manhattan. Outside the grid are the coordinates in both directions, and the predicted cost of the distance at any given point to the target is the sum of the differences of their horizontal and vertical coordinates. Another distance heuristic that is more familiar is the Euclidean distance, or straight-line path. SSP has many applications including AI decision making, robot navigation, VLSI signal routing, and for autonomous vehicles as will be seen in the following sections.

Conventional algorithms rely on sequentially traversing the search space, described above as the priority queue, which is inherently limited by von Neumann systems in traditional computer architecture. As graphs become very large, this slow processing time can become a bottleneck in real world applications. In this section a first-of-its-kind time-based ASIC is presented to address this issue. The design leverages a dedicated hardware implementation to solve these problems in linear time complexity and at unparalleled energy efficiencies. A 40x40 four-neighbor grid implements a wavefront (WF) expansion

with a first-in lockout mechanism to enable traceback. Outside the array, a programmable resistive ladder provides bias voltages to the edge cells which enables pulse shaping reminiscent of the A* algorithm [31]. Section 3.1.2 described the operating principle of the ASIC as well presents the core circuits that enable the unique functionality. Next, section 3.1.3 briefly describes quantitative measurement results. The heart of the work is described in section 3.1.4, applications. Finally, the subchapter is concluded in section 3.1.5.

3.1.2 Principle of Operation

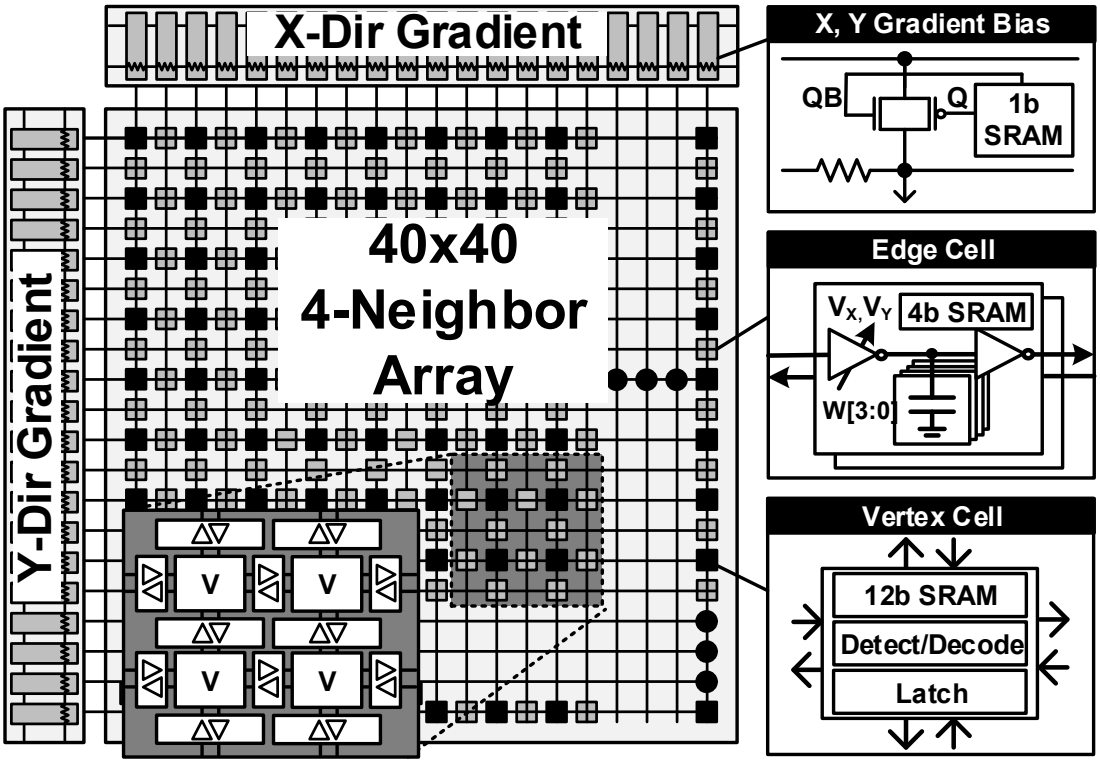


Figure 3.4: 40x40 graph ASIC chip for solving single-source shortest path problems based on 2-dimensional wavefront expansion.

Figure 3.4 provides a high-level schematic of the chip. The chip is structured to model a graph with a regular Manhattan grid structure. Each of the 1600 vertices have four connections to its neighbors in the cardinal directions (N, S, E, and W). The chip functions

by propagating a pulse between the vertices through the edges. The time it takes for a pulse to travel from each cell is proportional to the distance, or cost, to travel through that edge. Each vertex operates autonomously; it senses and stores the direction of the input, prevents other pulses from overwriting it, and propagates it to neighboring stages. The first pulse latched, or set of simultaneous pulses, represents the fastest way to reach that cell. Since the first pulse is the only pulse latched in each cell, tracing the pulse chain back to the start will reveal the shortest path. Connections are predetermined based on the description of the graph at runtime. Although the core was initially designed for SSP, each evaluation contains all shortest paths to the start node. In the following subsections each of the core circuit blocks will be described.

3.1.2.1 Vertex Circuit Functionality

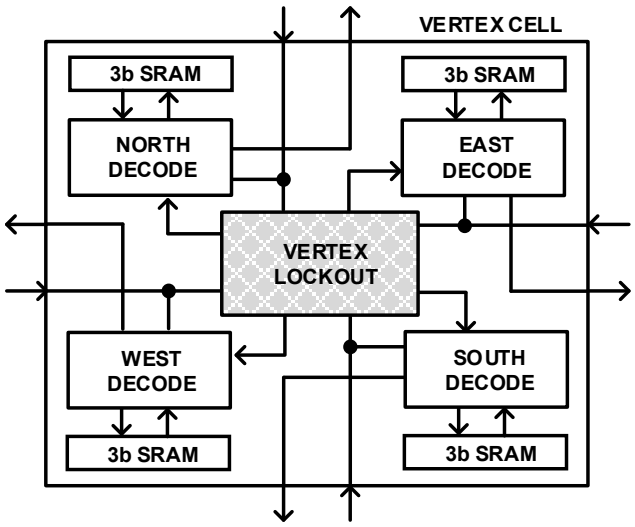


Figure 3.5: Block diagram of the vertex.

Figure 3.5 shows the block diagram of the key units that comprise the vertex cell. The vertex consists of four input/output ports, a lockout mechanism, and direction decoder cells that connect with local storage. Each of the directional storage is allocated to: output

edge enabled, input arrived from that edge, and one spare bit to align the layout. One of the spare bits is utilized to store if the vertex is a start cell. The key lockout operation is described in Figure 3.6. In the pre-input state, the SR latch output is low signifying no input has occurred. An input is presented to the vertex from the north. This input latches the vertex and pulses are passed to the edges. Additionally, inputs are prevented from overwriting the cell.

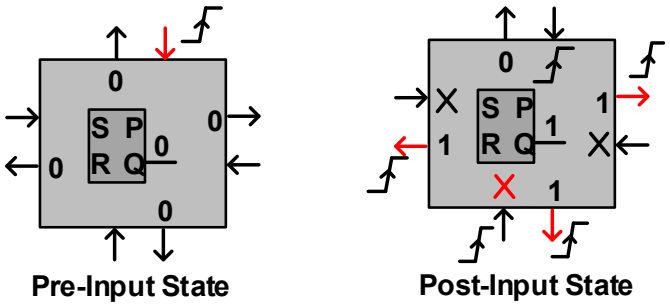


Figure 3.6: Operation of the lockout mechanism.

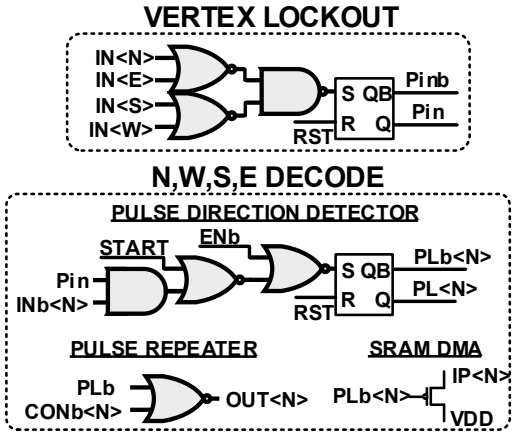


Figure 3.7: Vertex circuit schematic.

Figure 3.7 shows the schematic of the vertex cell. The lockout consists of a merging network to determine if a pulse has arrived. The decoder is responsible for determining which direction carried the pulse to the vertex. Once the pulse has been decoded it is

propagated to the output by the pulse repeater. Additionally, the local SRAMs are programmed with a direct memory access circuit in-situ.

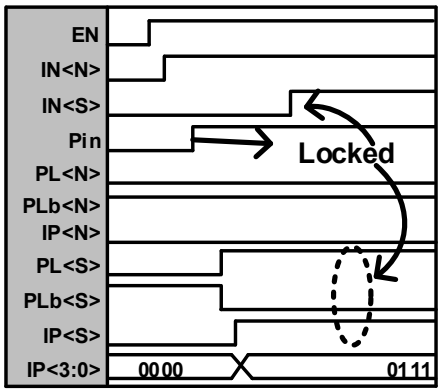


Figure 3.8: Vertex timing diagram of lockout functionality

A timing diagram that demonstrates the functionality of the lockout mechanism in Figure 3.8. The functionality of the cell will be described with an example of two pulses arriving; North and then South. First, a global enable signal, *EN*, is asserted enabling the core. Additionally, not shown in this figure, a pulse is started from somewhere in the array. The four inputs are merged together in a detection circuit to determine if a pulse has arrived in the cell. In the example *IN<N>* will flow through and latch P_{IN}, or Pulse Input. P_{IN} is compared with the input from each of the four directions. If P_{IN} is asserted and the input is not from the direction that asserted P_{IN}, Pulse Latch, in this example *PL<{S, E, W}>*, will assert. *PLb* is connected to the SRAM DMA which will flip the *IP<{S, E, W}>* SRAM that will be read out after evaluation during the path traceback. *IP<3:0>* shows how the vertex stores the input pulse. Initially all four bits are cleared. The bits that do not correspond to the first input are flipped with the DMA circuit. This notation corresponds to the Readout Colormap Key in Figure 3.9. In this example the input arrived from North, or "2", giving the stored cell value 1011_2, or 11_{10} encoded orange. Finally, the pulse is propagated to the

neighbors that were not responsible for latching the cell and have a connection stored in their respective local SRAMs. This is where the "in-memory computing" portion of the title is derived from; all circuits, connections, and pulse propagation occurs inside the memory array. The layout of the chip is constructed on an SRAM architecture which is what enables this concept.

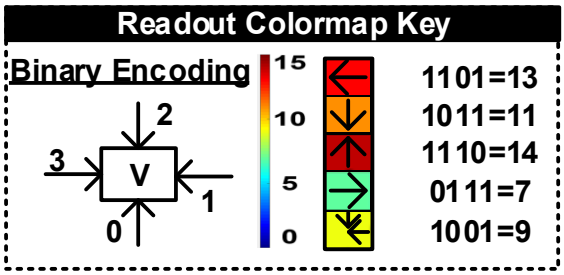


Figure 3.9: Local SRAM storage encoding enabling traceback

3.1.2.2 Edge Unit

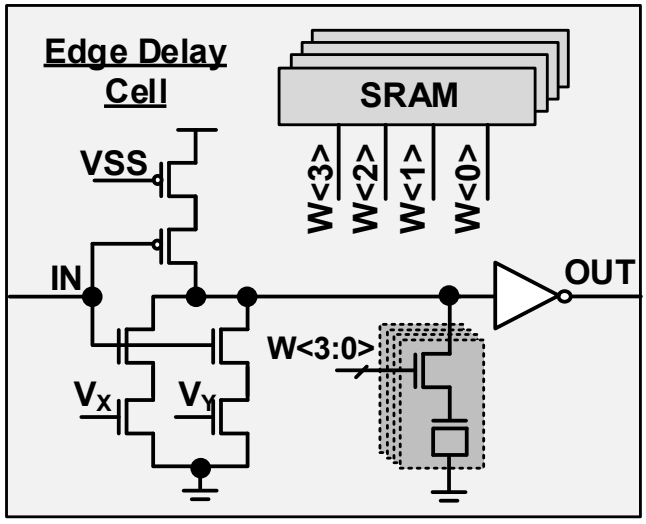


Figure 3.10: Edge Unit schematic

The edge unit is responsible for passing the pulse between vertices via a modulated delay. Figure 3.10 shows the edge schematic. Each edge consists of one current starved

inverter, one standard inverter, and four binary weighted bits of capacitor loading. The delay is modulated by the 4-bit weight stored locally in the SRAM and the voltage bias applied to each branch of the first inverter. The bias voltage values are assigned by the edge's position in the array determined by the gradient, described in section 3.1.2.3. The circuit of the current starved inverter could be reduced by two gates to reduce area, but were opted to stay as overhead. The first gate that could have been removed is the PMOS connected to *VSS*. The circuit would still function as digital logic without an always on path, but it was included in order to have similar pull-up and pull-down strength of the branches. The second gate that could have been removed is one the NMOS gates connected to *IN*. This would require shorting the drains of the two bias controlled NMOS footers. However, these were retained in order to ensure the linearity of the summated currents in the two paths. Shorting the drains could have resulted in voltages as V_{DS} that would have resulted in a non-linear contributions of the currents. As the gate voltages of the NMOS increases, this would have reduced V_{DS} which would have had an outsized effect on the gate with the lower V_G.

3.1.2.3 Gradient A* Mapping

The gradient cell is used to implement the predicted distance to the target. On each axis there is a resistive ladder that has voltage taps at each row/column that connect across the entire core array. Since this is a 40×40 array, there are 41 resistors on each side. Additionally, switches controlled by a register chain are placed at each node which is used to fix the voltage to V3. On the ends of the resistive ladder are two additional bias voltages. The difference between V3 and the other two voltages is linearly dropped across each

resistor shown on the opposite side of the ladder. This is a valid assumption because there should be very little current lost in the array since the taps connect to MOSFET gates.

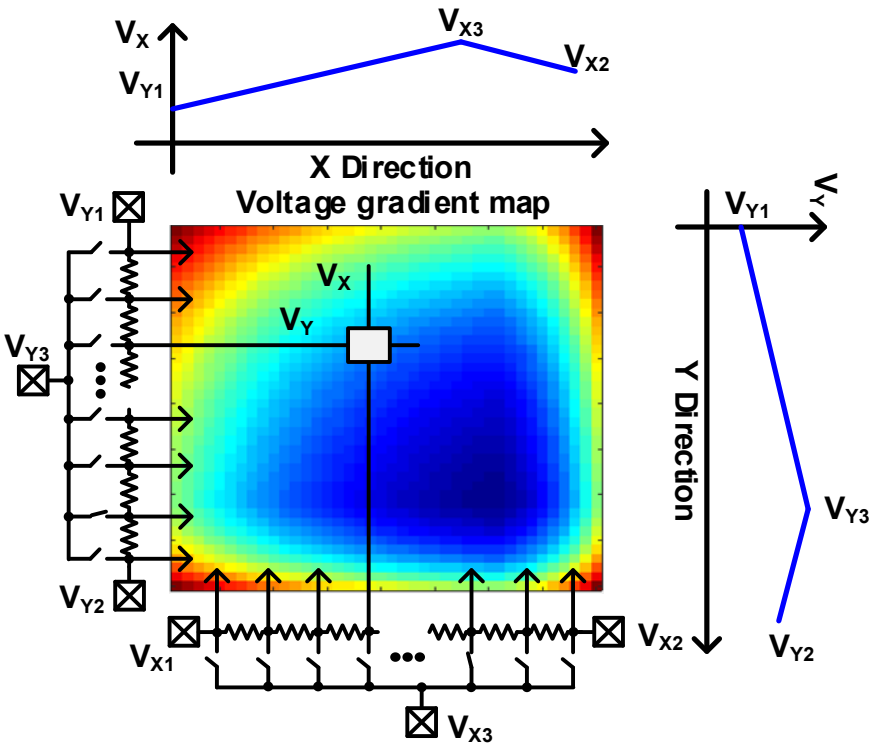


Figure 3.11: Functional example of the gradient ladder

The point at which V3 connects is programmable through a scan chain. Only one tap of the resistive ladder is connected to V3. The position of that connection corresponds to the location of the target, determined by the application requirements. In the SSP problem, the location of the target is the destination cell, whose coordinates would be programmed into the resistive ladder register connections. In Figure 3.11, the delay of each cell based on the bias voltages seen by the X-direction and Y-direction gradient is encoded in the color of the cells in the array map. The dark blue color corresponds to the fastest cells because the bias voltage is maximum at V3. As the distance increases from the target cell, the bias voltage decreases causing the delay to increase. This essentially accelerates the

pulse towards the destination in the same manner that the A* algorithm reduces the predicted cost as the search gets closer to the target node. Figure 3.12 shows an improved gradient circuit. For an additional scan bit and bias voltage (V_{Y4}), an additional bend can be incorporated in the slope curve. Additionally, non-linear resistive elements could be incorporated to provide second-order functions.

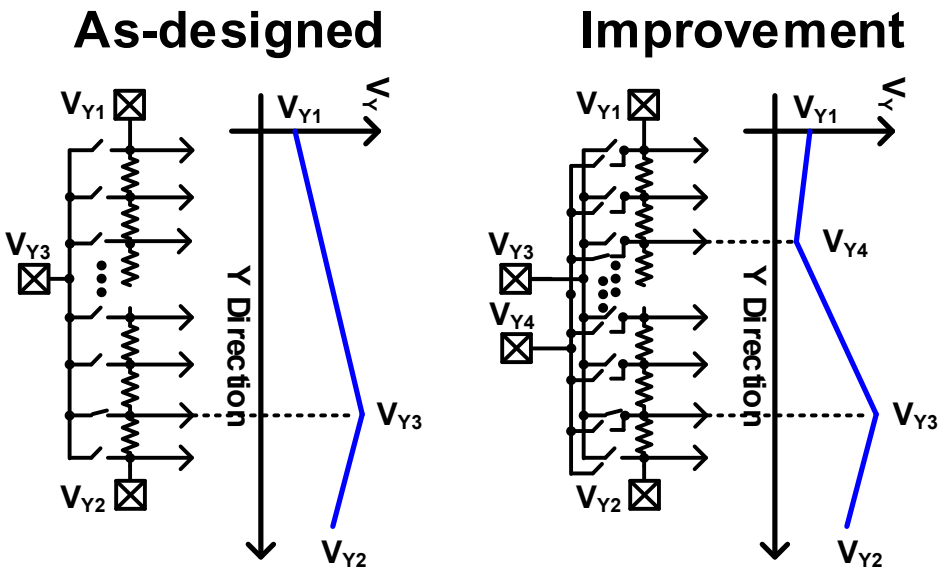


Figure 3.12: Possible Wavefront shaping with increased gradient complexity

3.1.3 Measurement Results

Figure 3.13 contains the measurement results from the edge cell delay. On the left is a plot of average edge cell delay versus digital edge code for three different bias voltages. Due to the lack high precision measurement circuits on chip and no ability to probe internal points during pulse propagation, a unique strategy was required to generate the delay curve on the left. The graph was programed to have only a single route across the chip. The path

started in the top left and traversed across the array to the right until the edge was reached. The path moved down one unit and then proceeded back across the core.

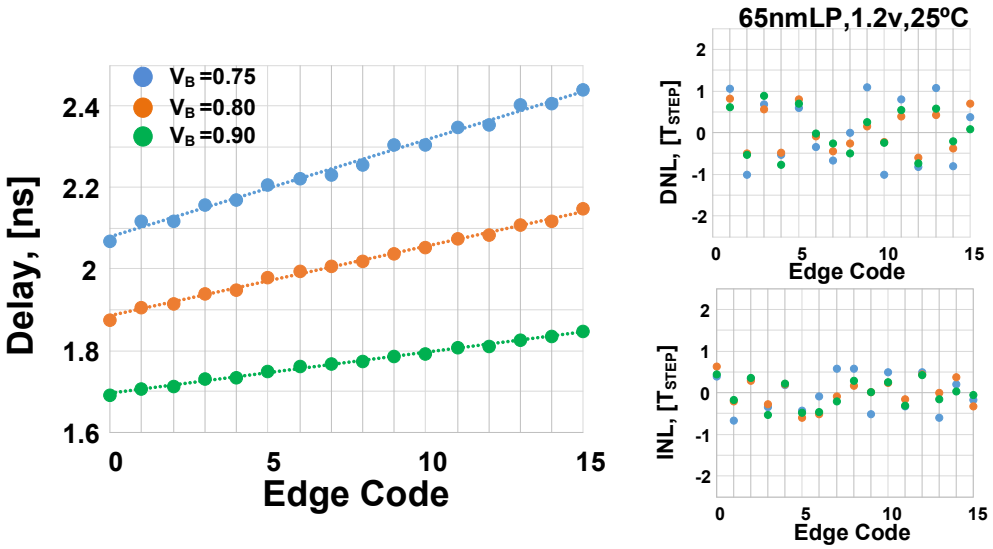


Figure 3.13: Measured delay of the edge cell

This was repeated over each row in the array to yield the longest single path possible in the chip. During measurement, the enable signal was precisely controlled by off-chip test equipment. The total time enable was divided by the number of cells that the pulse traversed. This process was repeated for different edge codes and bias voltages. To apply the same bias voltage to all cells, {V1,V2,V3} were all set to the same voltage and no connection was made on the gradient ladder. Over the three bias voltages shown in the figure, it is possible to see how the delay can be modulated across the array by the gradient resistive ladder framework introduced in section 3.1.2.3. On the right the differential nonlinearity and integrated nonlinearity are shown. The deviations in the DNL and INL appear to toggle between positive and negative. This could be due to the least significant bit in the capacitor bank not having a significant contribution to the delay. This is apparent in the V_B=0.75V trace in the left plot. Every other point has a minimal increase in the delay;

edge code two and three have nearly identical delays. In future designs this could be rectified by increasing the size of the access switch to increase the effect of the connected capacitive load.

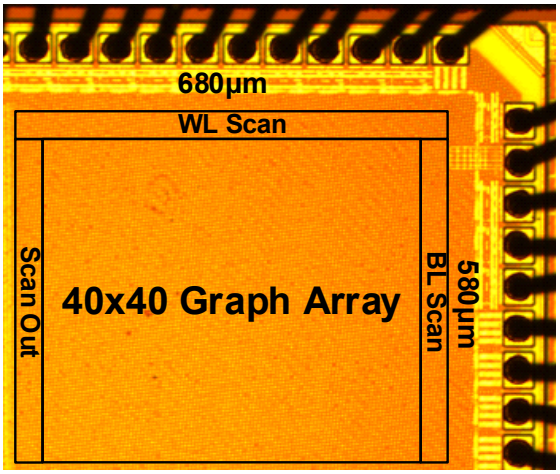
Table 3.1: Comparison Table

Architecture	This Work	FPGA [32]	µProcessor	CPU [33]	GPU [33]
Product	ASIC	Xilinx Virtex	ARM Cortex-M0	Intel Xeon E5630	NVIDIA Tesla K20c
Technology	65nm	20nm	40nm	32nm	28nm
Voltage	1.2V	-	1.1V	0.7-1.35V	-
Peak Power	26.4mW	24.22W	127µW	20W/core	225W
Throughput [MTEPS]	559	731	$5.34*10^{-4}$	0.83	9.0
Energy per Node	0.328pJ*	33nJ	89.1nJ	24.1µJ	25µJ
Normalized Energy	1x	10^5	$2.7x10^5$	$1.19x10^6$	$2.3x10^7$

*55% from SRAM Program (does not include cache access energy)
Energy/Node=Unit Delay*Unit Power
MTEPS = Million Traversed Edges Per Second

Table 3.1 is mainly included for general comparisons since to our knowledge this is the first ASIC for graph traversal. References [32] [33] are comparable in the sense they attempt to map hardware platforms onto optimal A* implementations. Our peak power is quoted when all 156 perimeter vertices are evaluating corresponding to the pulse originating from the center, equating to 183.1µW/vertex. 55% of the power is due to SRAM access storing the pulse information in-situ. During the program there is short circuit current in the two cross coupled inverters of the SRAM. This fact highlights how low the actual compute energy is; each vertex is evaluated by toggling a few gates and two inverters on the edges. Compared to state of the art FPGA [32], µProcessor, CPU, and GPU [33] implementations our core has roughly five orders of magnitude superior energy efficiency. Figure 3.14 shows the die photo of the test chip and the chip summary. The chip is fabricated in 65nmLP CMOS process and is based on a novel time-based architecture. The 40×40 array contains 1,600 vertices and 6,400 edges since each vertex have four neighbors.

The perimeter vertex cells have dummy edges to provide equal loading. The delay resolution on each edge is a 4b binary weighted capacitor bank in conjunction with the analog bias gradient which implements the A* heuristic. At 1.2V supply the peak power is 26.4mW, which corresponds to every cell on the edge evaluating when a pulse is started from the center of the chip. The delay per node is 1.79ns at an edge bias of 0.9V consuming an incredibly low 183.1µW.



Applications	A* shortest path, obstacle avoidance, scientific computation (optics)
Technology	65nm LP CMOS
Architecture	Time-based
# of Vertices	1600
# of Edges	6400
Edge Resolution	4b + Analog Gradient
Voltage	1.2V
Peak Power	26.4mW
Delay per Node	1.79ns @ [V_B=.9V, V_{DD}=1.2V]
Power per Node	183.1µW
Energy per Node	0.238pJ

Figure 3.14: Die Photo and Chip Summary

3.1.4 Applications

3.1.4.1 Collision Avoidance through Voronoi Diagrams

The principle of the Voronoi algorithm is that it segments a plane such that the partitions represent the closest position to a start seed. Visually, at the beginning of the algorithm there are only the start points available. Upon completion, the plane is segmented into regions around the start points. Voronoi diagrams are able to be computed by this ASIC when no bias gradient is applied. In the absence of a bias gradient the velocity of the pulse is constant across the core. This is precisely what k-nearest neighbor classification in

machine learning computes. Voronoi diagrams also describe the bone structure and the way certain groups of cells orient themselves. Computational fluid dynamics meshes are also generated using these principles. Autonomous vehicles can also use these for collision avoidance, which will be described in the next paragraph.

Figure 3.15 highlights an example collision avoidance (CA). CA is useful if there is an incentive to avoid obstacles, such as self-driving cars or drone navigation. In this application the pulse is started simultaneously from the sides of the obstacles. Where the WFs meet, shown in white, signifies the path that maximized the distance between the obstacles. In the bottom left of Figure 3.15 the boundary between obstacles “1” and “2” is highlighted. The white line is added as a visual aid, but it is clear that the WFs have met between the two blockages. The other callout shows how three WFs can meet and be identified. Along the top and side of the main array, the bias voltages for V1 and V2 are equal at 0.9V. The strength of this ASIC is highlighted in this application as each leading cell on the WF can operate autonomous vastly reducing the evaluation time compared to a standard von Neumann machine which services each vertex serially.

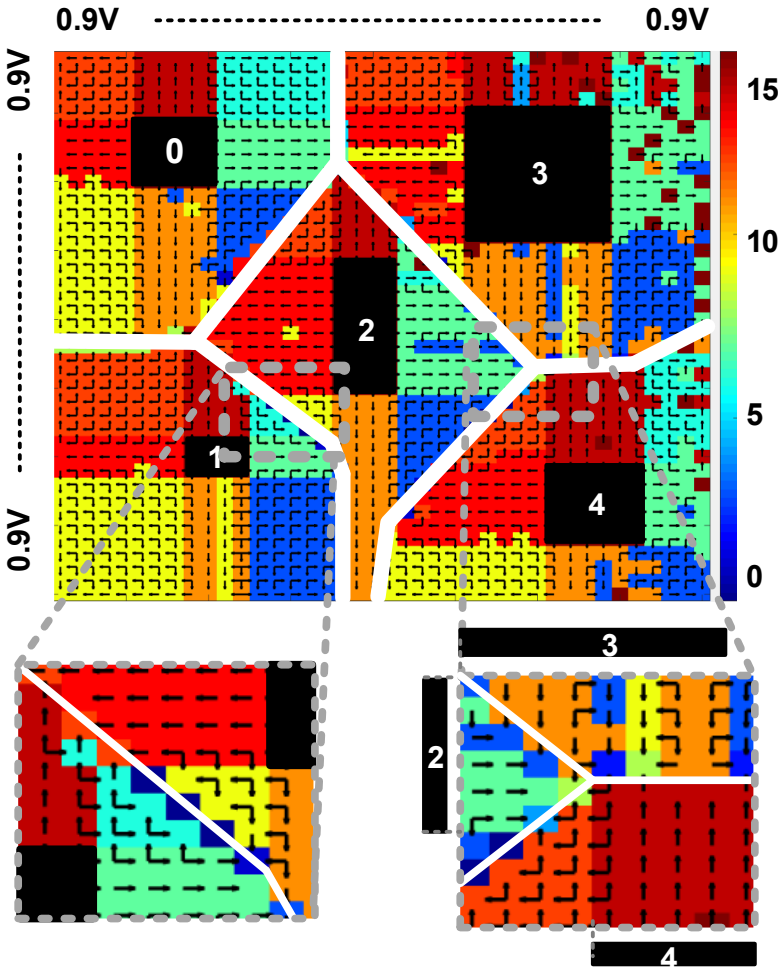


Figure 3.15: Collision avoidance example

3.1.4.2 Shortest Path Planning

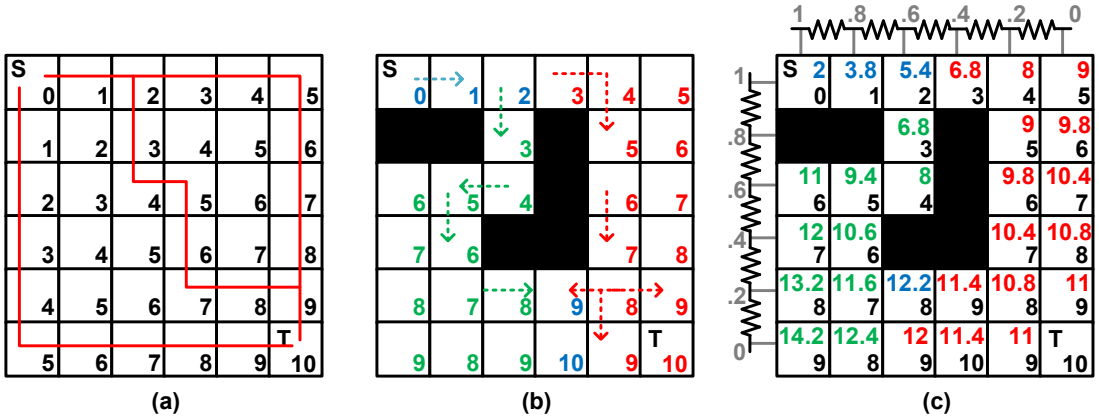


Figure 3.16: Path Planning on grids (a) costs from Dijkstra's (b) Dijkstra's with blockages (c) A* via gradient

The motivating application for this ASIC was SSP. Figure 3.16 solves SSP from the top left (start, S) to the bottom right (target, T). Finding a shortest path in a grid without any blockages is trivial as shown in Figure 3.16(a). Every path the moves in the down-right step will have the shortest path as shown in a selection of the shortest paths shown in red. This is also seen by looking at the distance from S stored in each node, and following an increasing path. However, when blockages are present it becomes a more interesting problem. In the case of Figure 3.16(b), all paths above the blockage in red dominate the paths in green that go below. This is because the lower path is required to move left one unit, instead of the down and right path in the previous slide. In blue is where the paths meet. Additionally, Figure 3.16(c) solves the problem with a gradient in the A* framework. The cost to move between the nodes is no longer a uniform step, but the sum of the grey "voltage" in the row and column. The gradient improves the dominant paths because the equal point has moved one unit to the left. This means the search is more efficient.

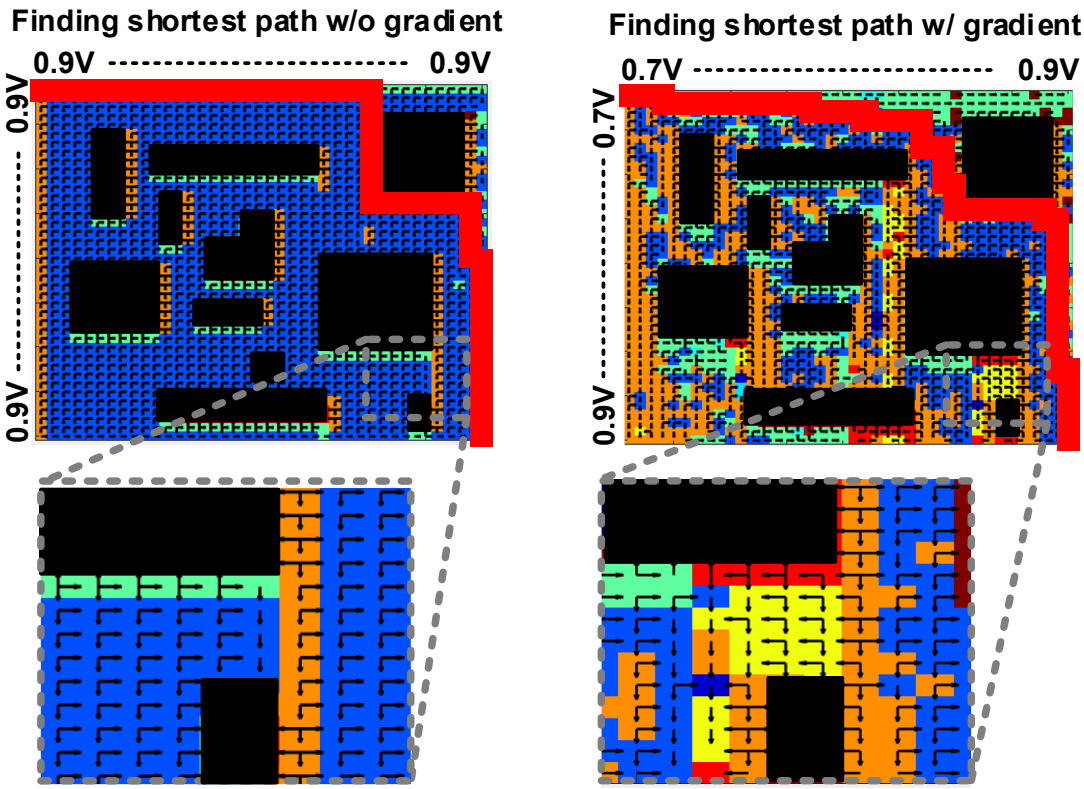


Figure 3.17: Measured results from path planning application without (left) and with (right) the gradient.

In SSP, Figure 3.17 shows two outputs generated from the same map under different conditions. In this application a map is shown with blockages shown as black blocks. The WF is initiated in the upper left and it propagates down and across the core. Figure 3.17(left) does not have a voltage gradient applied and each edge has the same weight. This gives the WF a very regular pattern as it traverses the map. The bottom figure has a voltage gradient applied that is weakest in the top-left corner and strongest in the bottom-right corner. The key difference between the two outputs is shown in lower callouts. Without the gradient, paths above and to the right of the blockage supersede any paths under as shown by the orange strip and no arrows crossing into those paths. With the gradient (right), the orange path is still present, but the "above WF" is so much faster than the "under WF"

that it begins to wrap underneath the blockage shown by the yellow cells. This is precisely the same observation seen in the analytical example in Figure 3.16.

3.1.4.3 Multi-core Scalability

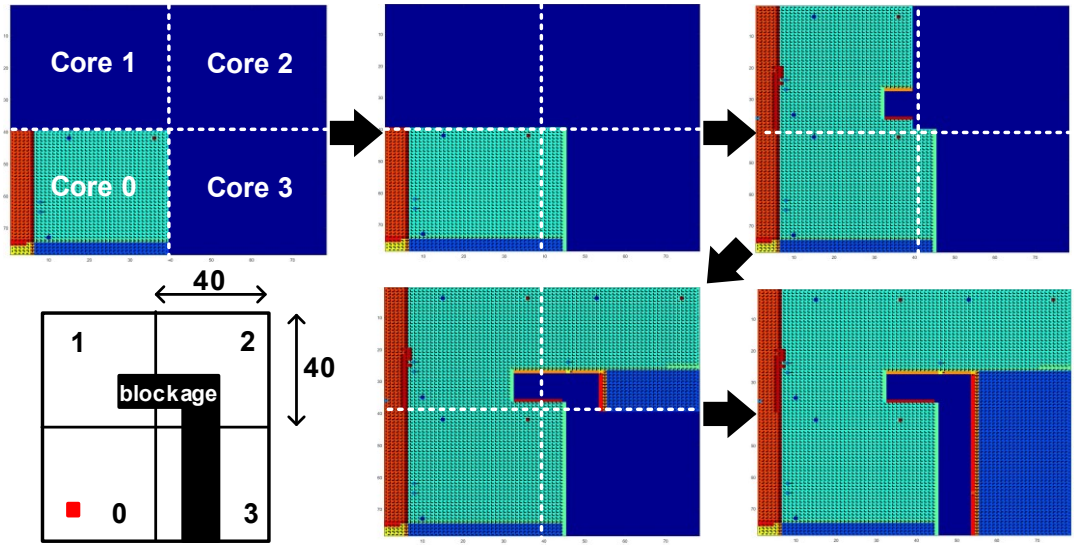


Figure 3.18: Four-core example with time-multiplexed outputs interleaved to shown full map.

This ASIC is not constrained to solving problems that conform to a 40x40 grid. Figure 3.18 highlights the scalability of this core via a four-core example with a single blockage spanning three cores. First, core 0 is evaluated and the WF reaches cores 3 and 1 both at two points, which will be used to start the pulse in subsequent evaluations. A simple rule to pick the vertex cells that have a single input pulse whose direction is opposite the boundary is used to determine the next start points in the following core. Next, core 3 is evaluated but the pulse does not uncover much of the map due to the complete blockage. After this, core 1 is evaluated and it contacts core 2 in two locations, the bottom right corner and directly above the blockage. These two cells are used to start the evaluation for core 2. Finally, it is revealed that core 2 and core 3 share an unexplored boundary, and core 3 is

re-evaluated to fully uncover the obstacle. It is important to note that multi-core still solves the SSP problem in linear time as the grid size expands. It can be safely assumed that any inter-chip communication will incur a large overhead relative to the WF propagation in the single core. One option to mitigate this could be to time-multiplex the core. The difference in the time to load a new map into the array and evaluate it could be compared to the energy and overhead from communicating with another die. This is a study well suited for future exploration including, simple processor to determine candidate start nodes, router framework for communicating between cores, and system design to ensure intelligent allocation of board-level resources.

3.1.4.4 Optics Experiment

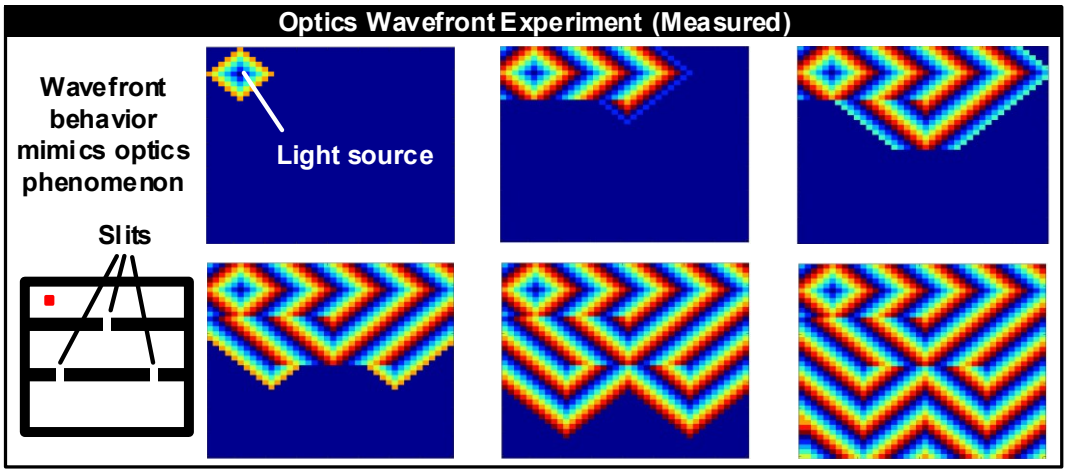


Figure 3.19: Measured optics Wavefront experiment.

Motivated by [29], this ASIC can also model optics experiments in straight-line geometries to illustrate the utility of our hardware architecture. The goal of this "experiment" is to model the propagation of waves under Manhattan geometry. In the physical world of course distances are measured in Euclidean geometries. Shown in Figure 3.19 is a sampling of the core readout at different time points during a single evaluation of

a two-slit experiment. The colors are encoded as the remainder of division between the distance from the start and eight (eight was selected to provide color contrast). The pulse starts to the left and above of the first slit. The WF traverses the first partition and then passes into the second corridor. Next, the WF reaches the middle of the second boundary and then spreads out until it reaches the two-slits and generates two leading WF. In this ASIC, interference is not easily modeled due to the lack of phase in the wave. However, the vertex can latch two inputs that occur at the same time, which could be interpreted as the point at which the waves would interfere. This behavior mimics what is reported in [29] and has interesting consequences for future physical explorations of novel applications with low-power CMOS.

3.1.5 Conclusion

In this section an in-memory computing ASIC graph processor was described. The core consisted of a 40×40 four-neighbor grid with each unit consisting of a vertex and its edges which enabled pulse propagation. Each vertex operates asynchronously which enables this chip to break the von Neumann bottleneck and solve the SSP in linear time as grid size increases. The design details and tradeoffs were discussed for the vertex, edge, and gradient blocks. Measured edge delay was presented as well as a compelling comparison table with conventional platforms used to solve the SSP. Versatile applications were highlighted along with the potential scalability of the ASIC. The time-based, in-memory computing ASIC applies a known algorithm to a well-studied class of problems in a new approach. The very competitive energy efficiency should drive this architecture to be considered for many other types of algorithms.

3.2 3D Graph Traversal ASIC

3.2.1 Introduction

In the previous subchapter a 2D path planning chip was presented. A 2D framework is ideal for navigating in planar applications such as finding the fastest route in streets which are in reality flat. There are inherent limitations in the previous design when it comes to implementing path planning in real world environments; namely the vertical direction. Applications that require having a third dimension include: drone navigation [34], underwater vehicle navigation [35], and unmanned ground vehicles [36]. 3D navigation is so challenging and computationally expensive that [34] didn't even bother to consider the third dimension even though the drone has the ability to fly. The authors referred to the ability to fly as a "hop" which severely limits the utility of the proposed system and could be mitigated by a low-power, fast, simple solution to find shortest paths in 3D environments. In this subchapter a 3D path planning ASIC will be introduced. In section 3.2.2 the architecture details including the array floorplan and vertex operation will be described. Section 3.2.3 will recount the measurement details of the chip. Three applications including: path planning, Voronoi diagrams, and k nearest neighbor classification will be demonstrated in section 3.2.4. Finally, conclusions will be drawn in section 3.2.5.

3.2.2 Architecture Details

3.2.2.1 Array Structure

One of the primary challenges associated with developing this concept was projecting a three dimensional volume onto a two dimensional plane. Some of the design

constraints were that it had to be scalable so that the size of the array could grow, compact so that the RC delay between vertex cells was not prohibitively larger, regular so that the routing complexity was manageable. Scaling a design in two dimensions is trivial; the x-axis and y-axis each have a degree of freedom in the layout plane which causes no contention in increasing the size. When a third dimension, z-axis, is desired to be incorporated the problem is much more difficult as there is no free dimension in the silicon to scale in.

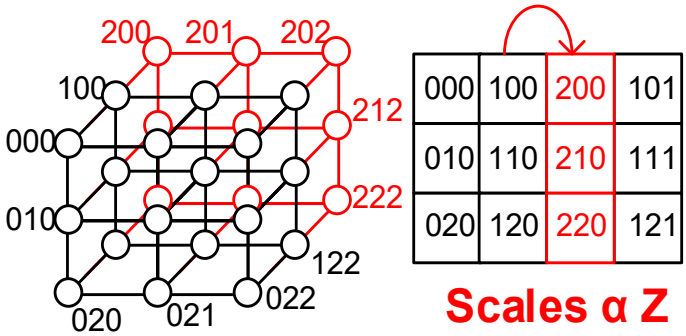


Figure 3.20: Limited Z-axis scalability in prior art.

This limitation is highlighted in Figure 3.20 [37], where a very unique idea is presented to use CMOS devices to model "spin" in a quantum computer. An Ising model is used to define the interactions between spin units, and they claim it can be used for combinatorial optimization, a very expensive problem when the size of the array becomes large. The design appears to map three dimensions into the two dimension plane. Upon careful observation, this is in fact slightly true. They get around the missing dimension problem by nesting, or interleaving the z-axis scaling with x-axis scaling. However, they are significantly restricted by depth of Z. This arises from the fact that the length of the route across x-axis is linearly proportional to the z depth. For example, the length of the

connection from N000→N001 (axis ordering ZYX) in the x-axis is two units, compared to N000→N100 in the z-axis of one unit. As the size of the units in the z-axis increase, the routing length will become a bottleneck. Larger RC will increase power consumption and delay reducing the operating frequency, consume metal routing tracks increasing the size of the cell unsustainably, and most importantly limits the geometry of the spin-mesh they can solve to just two layers.

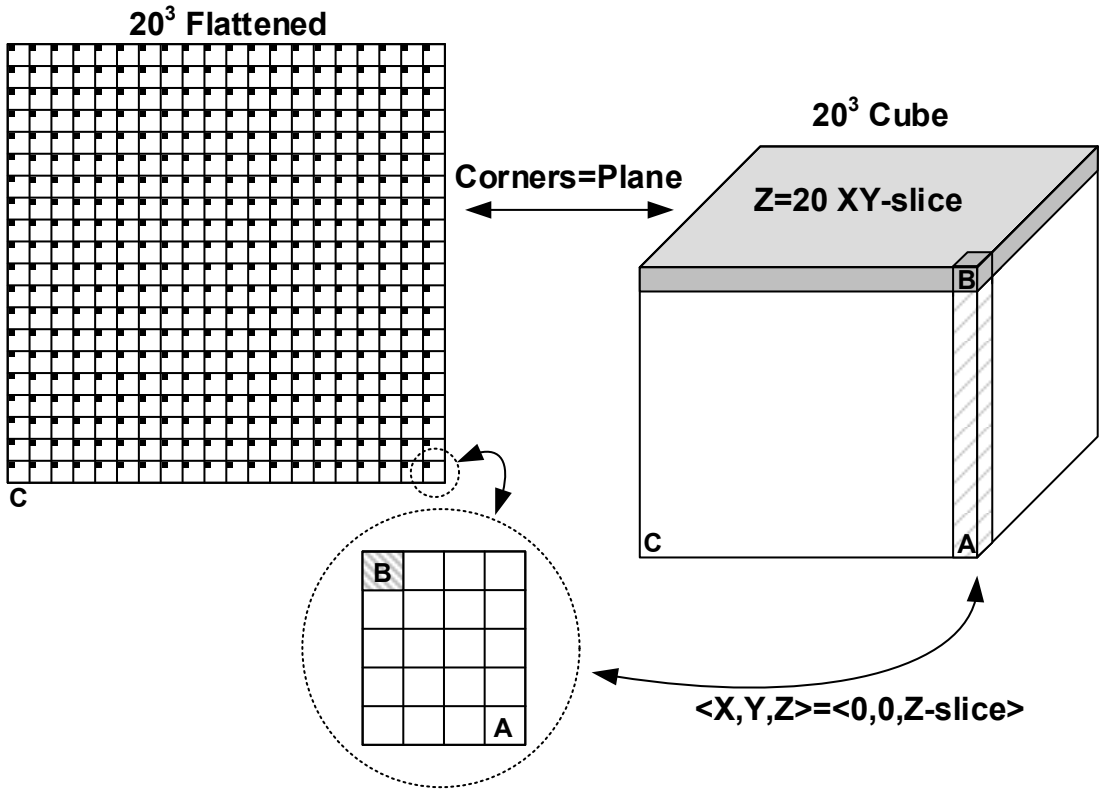


Figure 3.21: Mapping between 3D cube and 2D planar layout.

Figure 3.21 illustrates the solution leveraged by this ASIC. On the right is the 3D volume representation of the cube. Instead of simply interleaving the z-axis in one direction, it is distributed in the y-axis as well. This reduces the routing overhead by $\sqrt{z}$ compared to z. This is seen in the <X,Y>=0,Z-slice rectangle on the lower part of the figure.

This rectangle fits into the global grid in the top left flattened array. Each of the Z-slice locations corresponds to the x,y location in the array. The top face of the cube is the <X,Y,19> slice and is highlighted in the upper left corner of the Z-slices in the flattened array. The points A, B, and C are identified on the three representations in their appropriate locations to aid in understanding the translation between the different structures.

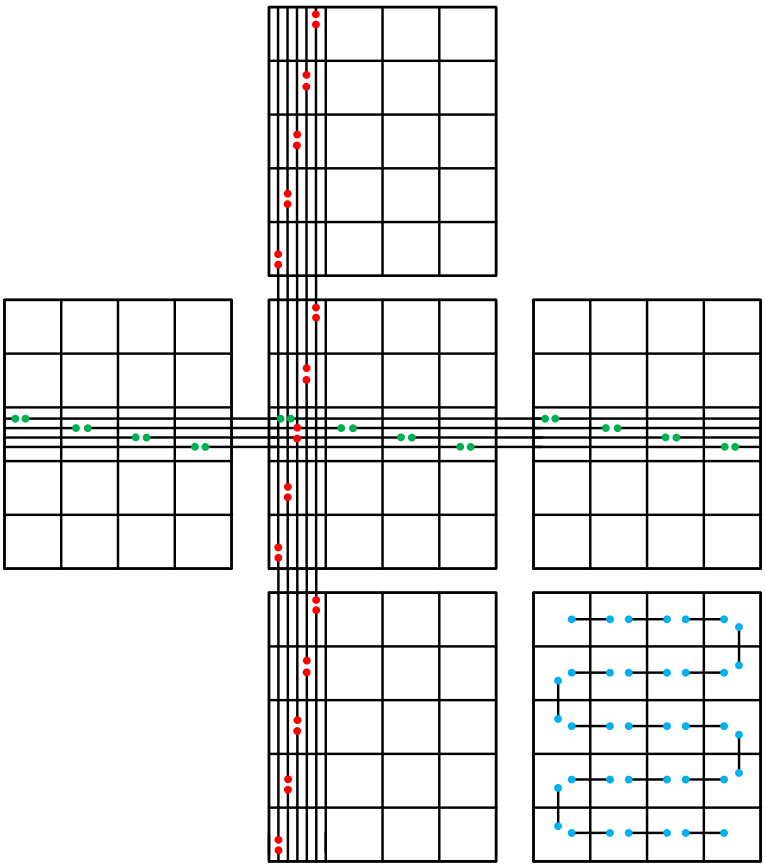


Figure 3.22: Global routing across the three dimensions.

The routing methodology shown in Figure 3.22 had to be developed to connect cells in this compact layout. Each wire connection is bi-directional, but shown as a single route for brevity. The z-axis routing is done locally shown in the bottom right z-slice in blue. The x-axis routes are staggered four wide and the y-axis is staggered five wide. This difference

is due to the unit cell aspect ratio being wider than tall. This stems from the long, narrow layout of a 6T SRAM cell. Each of the X,Y, and Z routes occur in each cell, although in this figure they are split for ease of seeing the pattern. This makes a dense, regular, and automation-friendly routing structure. It can be thought of as automation-friendly because the coordinate location of the cell dictates the location of the tapping vias to the global routes.

3.2.2.2 Vertex Operation

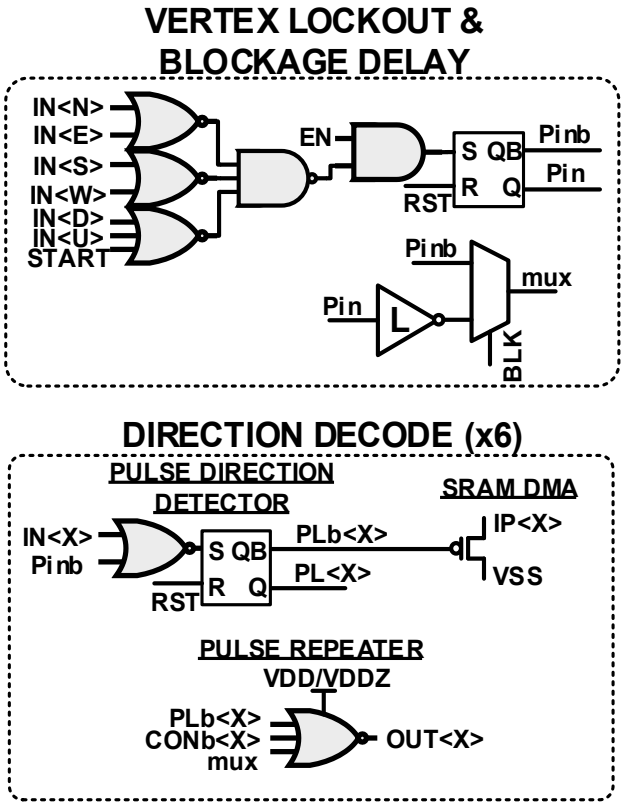


Figure 3.23: Circuit schematic of vertex cell

Figure 3.23 shows the details of the circuit schematic of the vertex cell. Compared to section 3.1, there is no edge cell to reduce design complexity. The key functionality of the vertex is largely similar to the 2D design presented earlier; detect if a pulse arrives,

decode the direction of the first pulse, latch this information in-situ, and propagate the pulse to the connected neighbors. The vertex lockout block consists of three NOR-NAND merging network to determine if a pulse occurred. An improvement over [1] is that the local *START* signal is mixed with the pulse inputs. In the previous work *START* was mixed in the decoder stage (c.f. Figure 3.7), which not only added another logic stage into the critical path but also had to be routed to each direction which increased congestion in the already dense vertex. The global enable, GLB_{EN}, was also mixed in the lockout to also reduce overhead and supervise the logic at an earlier level. The output of the lockout feds the decoders, as well as a new addition in this 3D work, the Blockage Delay circuit though a mux. The idea behind the blockage delay circuit is that during 3D navigation (section 3.2.4) the goal is to find the shortest path from source to target in the presence of blockages. This delay path is enabled within some range of the boundary of the obstacle such that the optimal path will avoid traveling too close to the potentially dangerous blockage while still traveling on the fastest route. This single bit is stored locally in each vertex and is programmed by the user at runtime. The delay from the inverter (denoted L in Figure 3.23) is realized by stacking devices to get a long channel length. In this design the width:length ratio is 1:5 which provides a significant portion of the total vertex time in delay. The polarity of the latch signal will remain the same on both branches of the blockage delay circuit because the pass through branch is connected to the positive output of the SR latch, and the blockage delay branch in fed by the opposite polarity output available from the SR latch. The output of the blockage delay circuit is fed to the pulse propagating circuits described later in this section.

The decoder network is tasked with determining which direction carried the first pulse. Each of the six directions (North, East, South, West, Up, and Down) has a copy of this circuit with the input connected to the corresponding direction. The input pulse and the complement of the lockout latch are passed into a NOR gate which asserts on each direction that does not have an asserted direction. Before the cell is locked, all the inputs are low (active high output of the NOR gate), but the lockout output is low and its complement is then high (dominant active low output of the NOR gate). However, when an input occurs the decoder corresponding to the direction of the input has the values toggle and no change is recorded in the decoder latch, but the directions that did not carry the pulse in the cell are sensitized and latch the decoder cell. The output of the decoder latch drives the in-situ storage which records the one-hot encoded direction which caused the cell to latch. Additionally, the decoder output is fed into the pulse propagation block. Each direction again has its own copy of the decoder output. A three-input NOR gate controls the pulse propagation of each direction. The inputs are: complement of locally stored connection on that direction, blockage delay circuit output, and complement of the decoder latch output. The mux output is mixed at the pulse propagation block and not in the decoder block so that both can work in parallel to reduce the intrinsic vertex delay. Another unique feature of pulse propagation circuit is that the z-axis (Up and Down) NOR gate VDDZ is separate for the core VDD. Recall from section 3.2.2.1 the distance of the z-axis routing was a single unit, whereas the x-axis and y-axis were four and five units respectively. Having the ability to reduce the z-axis VDDZ compared to the core VDD gives the ability to match the short RC delay to the longer RC. The circuit innovations in the 3D version

include: reducing vertex critical path delay, incorporating path heuristics to model blockages, as well as increasing the number of paths from four to six, all which make the 3D version a more refined vertex compared to the previous 2D work.

3.2.3 Measurement Details

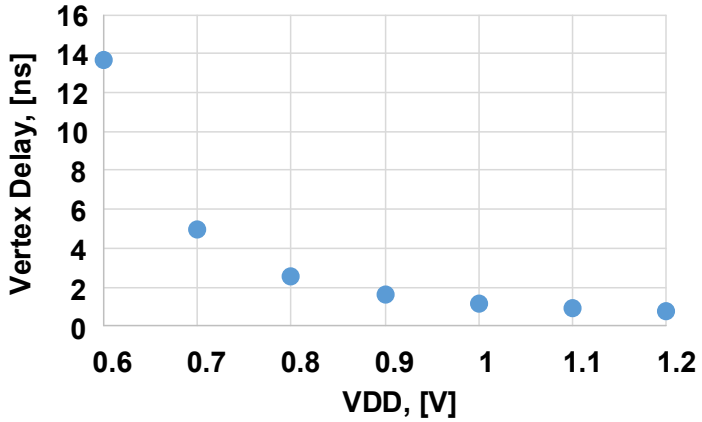


Figure 3.24: Measured average delay of each vertex

Figure 3.24 reports the measured average delay per vertex as supply voltage is changed. This curve was measured by programming a single line path through the core in a zigzag pattern across each z-slice and passed up a level after the pulse travels through a complete path. The chip is enabled for a short period of time and the number of cells that were evaluated in this period is recorded. Dividing the evaluation time by the total number of evaluated cells gives the average delay per cell. This function is essentially creating a single line time-to-digital converter. It was found to be accurate with post-layout simulations within 7.5% difference across supply voltages.

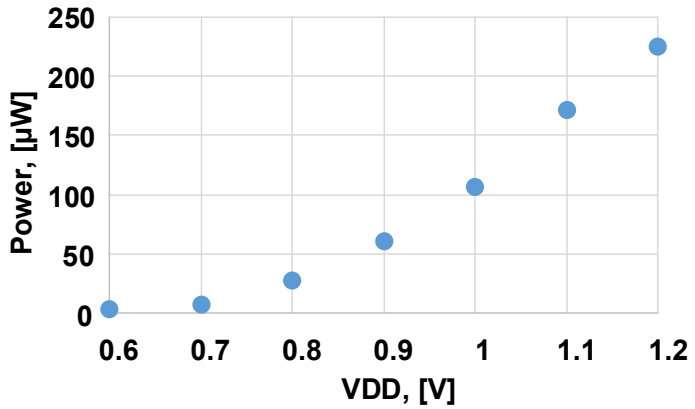
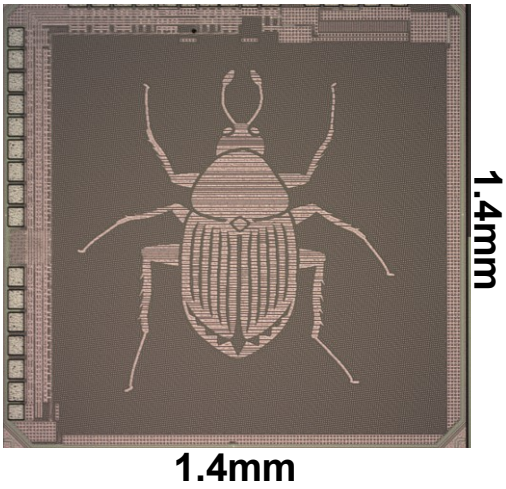


Figure 3.25: Post-Layout Simulation of Power

Figure 3.25 shows the power consumption dependency on supply voltage. Power is quoted as the average power over a single vertex evaluation time. This is because in the asynchronous chip, any number of vertices can be switching simultaneously which increases the power proportional to the number of cells switching. This power was measured in a post-layout simulation because it is not possible to measure the transient switching power of this chip and there is no way to program it to oscillate such that an AC power measurement would be taken over an average time period. Future designs could incorporate such a test structure to enable active power measurements. The highest energy efficiency is achieved at 0.6V for 40.8fJ/node.

Die photo (left) and chip summary table (right) are presented in Figure 3.26. Applications, demonstrated in section 3.2.4, include 3D path planning, Voronoi diagrams, and machine learning. However, this is not an exhaustive list as applications from the 2D chip can be mapped onto the 3D architecture, in addition to new applications due in part to the flexibility of the architecture. The speedup between the 2D and 3D version of this chip is 2.47x at 1.2V. This can be attributed to removing the edge cells and reducing logic levels

on the critical path to increase throughput. Power per node is higher by 23% due to the larger number of connections in the 3D chip.



Applications	3D shortest path Voronoi diagrams Machine Learning
Technology	65nm LP CMOS
Architecture	Time-based
# of Vertices	8,000
# of Edges	48,000
Edge Resolution	1b + Vertical "Gravity"
Voltage	0.6V-1.2V
Peak Power	538.6mW (2400 cells)
Delay per Node	0.723ns @ [V_{DD}=1.2V]
Power per Node	224.4µW
Energy per Node	0.162pJ

Figure 3.26: Die Photo and Chip Summary

The total energy per node is reduced by 46% which is a benefit since there are five times more nodes on this chip. The total die area is 2mm^2, with an active area of 1.33mm^2, and a vertex area of 163µm^2 for an area efficiency of 98.1%.

3.2.4 Applications

3.2.4.1 3D Navigation

Figure 3.27 shows an example of the 3D navigation application for autonomous drones. The goal of the application is to find the shortest path from (0,0,0) to (19,19,19). Blockages are programmed based on the map determined by the application. In this example, there is a blockage that spans the entire horizontal axis near y=10. Along the diagonal, or shortest path from start to finish, the highest point in the obstacle range which forces the shortest path off the diagonal. One of the possible shortest paths has been enumerated in green over the surface plot.

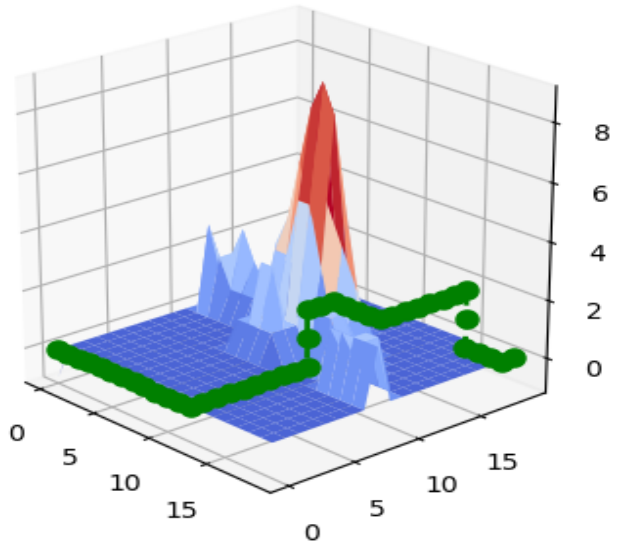


Figure 3.27: Example of the 3D Navigation application.

This path was achieved by simply tracing back the shortest path from the target node to the start node. This trace back can be visualized in Figure 3.28 where z-axis slices are shown. Each point represents a vertex, and the arrows represent the directions responsible for locking out the cell; the fastest way to reach the cell. In the bottom left of each figure is (X,Y)=(0,0) and the top right is (X,Y)=(19,19). The left figure corresponds to the cells in the Z=0 slice. One can see there is no path from (0,0,0) to (19,19,0) which necessitates the path take a route that follows a higher vertical dimension. In the Z=2 slice one can see that there are two paths across the blockage and the green route in Figure 3.27 takes the right most path.

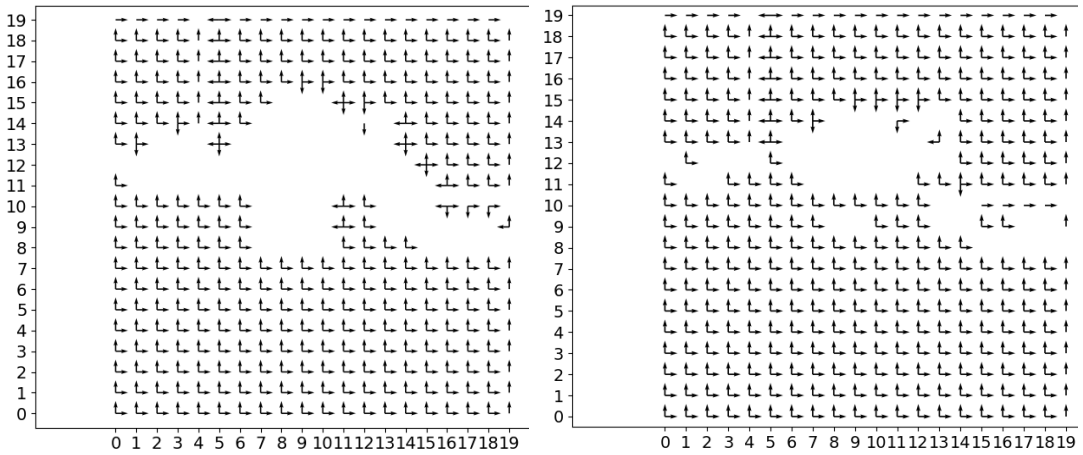


Figure 3.28: Readout of the chip of the Z=0 (left) and Z=2 (right) plane

3.2.4.2 Voronoi Diagrams

As described in section 3.1.4.1, the Voronoi algorithm segments a volume into regions that are closest to the various seed nodes. Visually, at the beginning of the algorithm there are only the start points available, seen as the main black points in the 3D plot in Figure 3.29. The core is enabled and wavefronts propagate from all start nodes in the six directions. Upon completion, the plane is segmented into regions around the start points. This is visualized by the simulated Voronoi diagram that looks like two stacked emeralds. This diagram is not generated from the core, but is analytically calculated in software. Three planes slice the figure at A:(X,Y,15), B:(10,Y,Z), and C:(X,Y,0). Each of the corresponding planes has been shown from the readout of the core. One can see the projection of the simulated Voronoi diagram is projected onto the different planar slices, and is visible in each of the three readout planes. Software could be developed to search plane slices from the core to determine the precise full 3D reconstruction from the core. For this exploration one can visually confirm this is possible and the correct functionality

of the core is confirmed. This can be used for collision avoidance as seen in section 3.1.4.1, or k-Nearest Neighbor classification as will be detailed in the following section.

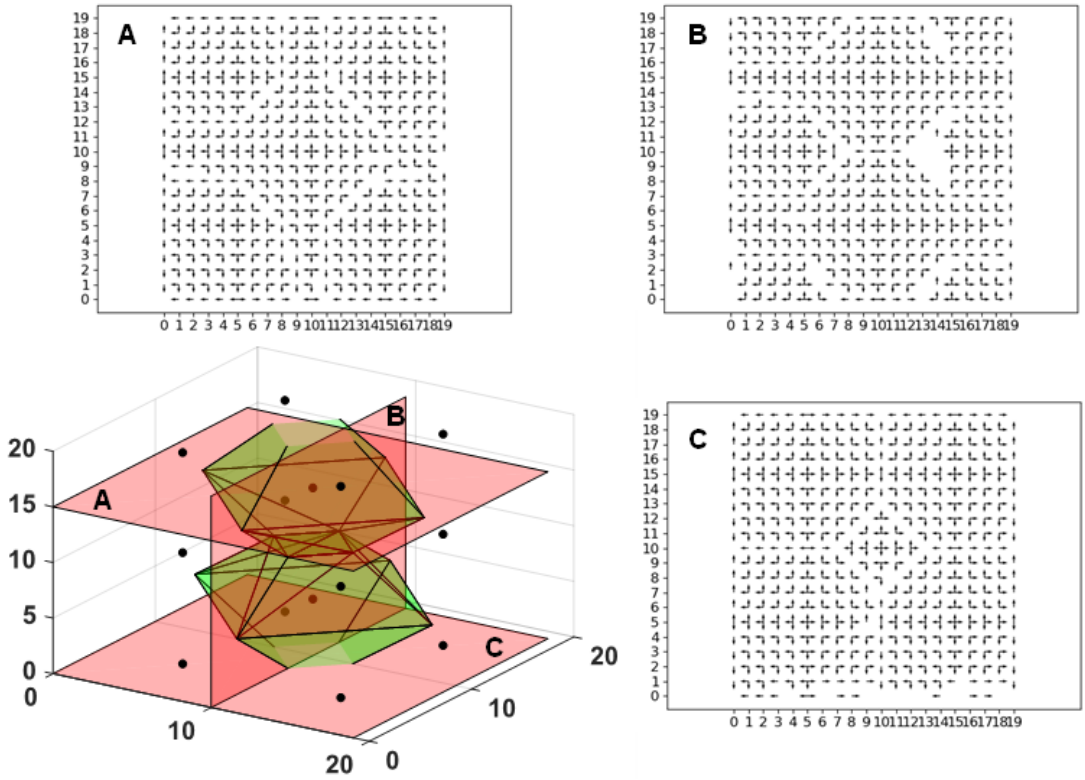


Figure 3.29: 3D Voronoi diagram via 2D reconstruction

3.2.4.3 k-Nearest Neighbor Classification

An application of Voronoi diagrams is k-nearest neighbor (kNN) classification when k=1. This is because the output of the Voronoi algorithm is a segment or partition that contains all the points closest to the start seed. In kNN the goal of the algorithm is determine the class of a new observation given a population of examples that are members of certain classes [38]. Conceptually the new point is compared to all the existing points and the distance between all the features is computed. The majority class of the *k nearest neighbors* is assigned to the new observation. However, the 3D core is not limited to k=1.

Instead of evaluating the seed points, the new point is evaluated. The distances to each of the points in the population is computed by tracing back the shortest path to the new point and then post-processing can be used to determine the size of k.

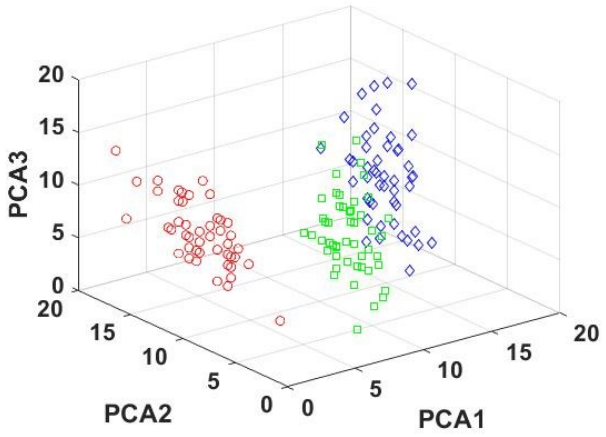


Figure 3.30: Fisher's Iris dataset prepared for the chip

To demonstrate this functionality the canonical Fisher's Iris [39] dataset is used. Originally there are four data attributes, sepal length, sepal width, petal length, and petal width for the 150 iris specimens. There are three target classes, or species of iris flowers; setosa, versicolor, and virginica. Due to the fact that the chip only has three dimensions, Principle Component Analysis [40], or PCA, was used to find the three dominant components of the data, reducing the dimensionality from four to three. Next, the three principle components were scaled onto the range of [0,19] and quantized such that each observation could be given an exact location in the grid, seen in Figure 3.30. The core was loaded with the first start point, evaluated, read out, and offline shortest paths were reconstructed by directly tracing back the shortest route to the start node, and the 10 nearest neighbors were used to determine the accuracy of the core on the algorithm. A software model of the kNN algorithm was used to determine the correct answers. The software

model correctly classified 142/150 examples. The core also classified 142/150 correctly, matching the software result exactly. Figure 3.31 shows the average distance of the 10 neighbors from the start point from the core (top) and software (bottom) colored by flower index. The mean path over all 150 specimens was 3.675 in the core and 2.64 in software. The disparity comes from the fact that the core used Manhattan distance to compute the distance and the software uses the Euclidean distance. The ideal ratio between these two distance metrics is 1.41, or $\sqrt{2}$, from the Pythagorean theorem: $\sqrt{a^2 + b^2} = c$, where in an idealized case $a=b=1$.

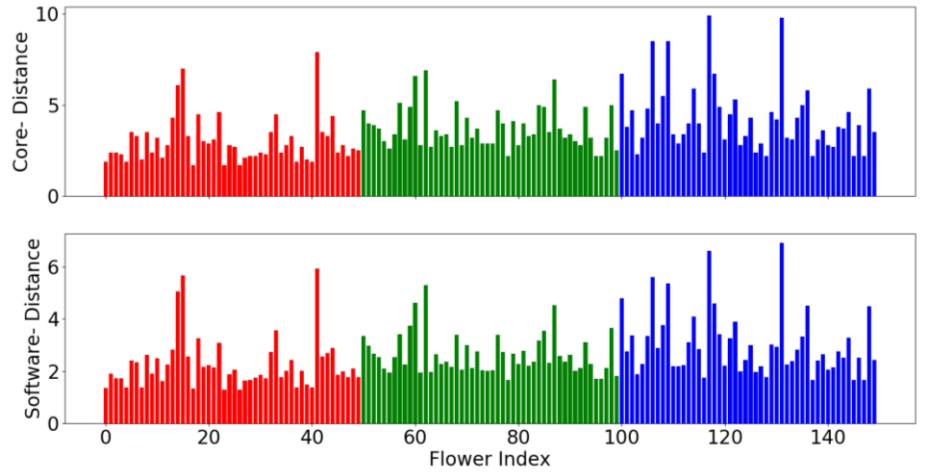


Figure 3.31: Measured kNN distance for k=10

It should be noted that this is not a primary application of the core, nor is it suggested this is an efficient implementation of the algorithm. One of the main limitations is the number of features allowed is restricted to three, each axis of the chip. As shown from the Fisher Iris dataset there were four dimensions given and it would have been ideal to include all for generating a better model. Additionally, computing the Euclidean distance is moderately challenging as the equation requires a square root to be computed. However,

the Manhattan distance, used in the chip, only requires simple subtraction of each dimension, and addition to compute the distance which is significantly less computationally intensive in a conventional von Neumann digital system. The purpose is to motivate the exploration of future algorithms that could be served by this 3D graph traversal chip and highlight the flexibility.

3.2.5 Conclusion

A 3D time-based path planning application specific processor was presented in 65nmLP CMOS. A significant innovation was transforming the 3D cubic architecture to a 2D planar layout that is scalable and efficient to implement. Previous work interleaved the z-axis in a single dimension, whereas in this work the z-axis was interleaved in both dimensions, which reduces the interconnect length by a factor of a square root. The operation of the vertex cell was described. Measurement results were presented along with the chip summary table. 3D path planning, Voronoi diagrams, and k-nearest neighbor applications were demonstrated in the chip. An increased core size to study more interesting maps, addition of the z-dimension, new applications presented, and improved energy efficiency compared to the 2D version of this chip make this a more versatile solution for flying drones.

Chapter 4. Time-based Biosignal Recording System

4.1 Introduction

In-vivo recordings from microelectronic electrode arrays are becoming clinically useful due to their promise of providing the capability to record from hundreds of neurons simultaneously [41] [42] [43]. This capability can provide neuroscientists and clinicians with the essential tools to study neurodegenerative disorders such as Alzheimer's disease. Since neural signal voltages are inherently small compared to full swing-inputs required by conventional ADCs, this poses challenging design constraints from the circuit designer's perspective. These challenges include; low input referred noise, ability to block DC offsets from the electrode-tissue interface, large dynamic range, tunable filters to identify clinically relevant signals, consume minimal power, and occupy as little area as possible [41]. All of these constraints incur trade-offs between area, power, and recording quality. In the leading edge neural recording systems, silicon shank electrodes are outfitted with a capable 1356 channels [43]. Shown in Figure 4.1 (above), each channel requires a custom pixel amplifier which proves a small gain to the neural signals, and multiplexers which send the analog voltages through the shank to a series of high gain amplifiers in the base before it is digitized. While this is an impressive effort, it fundamentally suffers from transmitting analog voltages through the shank. Even though they reduce the aliasing due to the lack of low pass filters at the pixel by using an integrator, the in-band noise is still increased. Crosstalk between channels also degrades the performance. In addition, longer shanks require larger driving strength which reduces the available area in the shank for recording sites [44]. Moreover, high performance ADCs are designed to be optimal for a

given sampling frequency in time multiplexed applications and incorporating additional channels requires a redesign [44]. If the signals could be digitized at the source in Figure 4.1 (below), this would solve the issues caused by analog voltage transmission, enable full use of the electrodes in the shank, and dedicate the shank base to perform more complicated digital filtering.

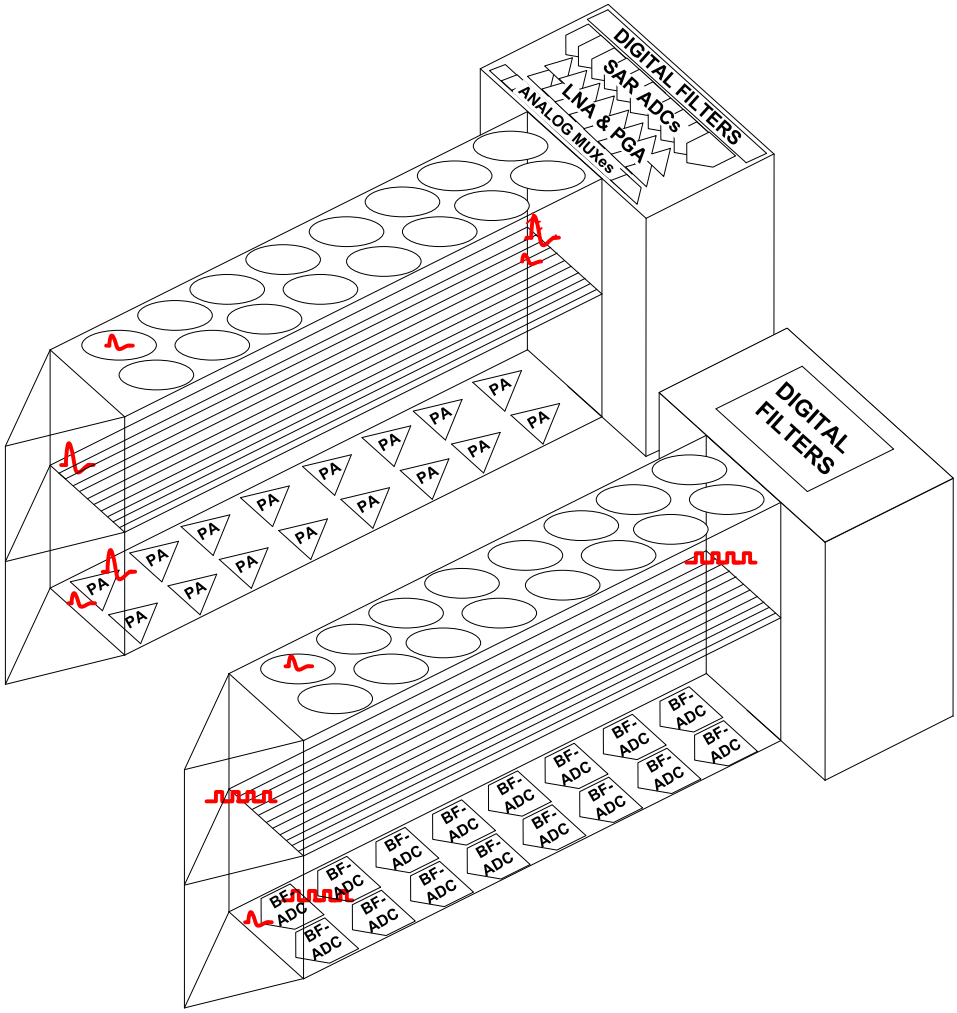


Figure 4.1: (Above) Conventional shank-based neural recording system [3]. (Below) Envisioned BFADC system.

The Beat Frequency Analog-to-Digital Converter (BFADC) is optimized explicitly for ultra-high density neural recording and can sense changes in signals down to 0.01% [45]. The basic principle of BFADC is to measure the frequency difference, or beat frequency, between two identical oscillator circuits, driven by a differential signal pair. By making the two oscillating frequencies similar to each other using trimming circuits, it is possible to obtain an extremely high built-in amplification gain that is inversely proportional to the beat frequency. Furthermore, this digital-intensive approach is amenable to technology scaling and low voltage operation, unlike conventional approaches based on sophisticated analog amplifiers containing large passive devices and requiring matched components to minimize offset. The 65nm test chip presented in this chapter requires a petite 0.0094mm^2/channel for AC coupling, low-gain analog amplification, filtering, and digitization. By focusing on low-area, low-power, digital-intensive circuits, the BFADC could digitize neural signals directly at the electrode source without increasing the footprint of the electrode shank [42] [46].

This chapter begins with the BFADC concept in section 4.2. Section 4.3 details the test chip implementation with benchtop results following in section 4.4. *In-vivo* physiological recordings from a mouse cerebellum highlighting the utility of the BFADC are presented in section 4.5. Conclusions are drawn in section 4.6.

4.2 Beat Frequency ADC

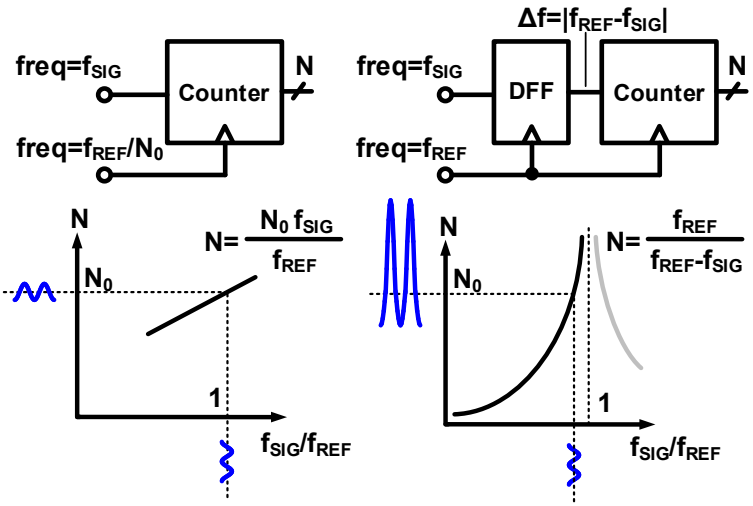
Figure 4.2 compares the schematic and gain characteristics of the conventional linear VCO based quantizer and the proposed BF based quantizer. The voltage input from

external and reference electrodes drive two identical VCOs which generate clock frequencies f_{SIG} and f_{REF} that are linearly proportional to the electrode voltages.

$$f_{SIG} = K_{VCO}V_{SIG} \quad \textbf{4.1}$$

$$f_{REF} = K_{VCO}V_{REF} \quad \textbf{4.2}$$

In linear VCO based ADC, the number of cycles N in a fixed sampling period N_0/f_{REF} is counted. Here, N_0 is the nominal count which is chosen based on the target sampling frequency of the ADC. This gain corresponds to the slope of the straight line in Figure 4.2 (left, middle). Since the slope is proportional to the nominal count N_0, the only way to increase the sensitivity to f_{SIG} is by counting for a longer sampling period which degrades ADC performance.



N_0	Linear Quantizer	BF Quantizer (This work)	
	Gain*	Gain*	f_{SIG}/f_{REF}
50	50	$49^2 = 2401$	0.980
100	100	$99^2 = 9801$	0.990
200	200	$199^2 = 39601$	0.995

*Gain=slope=dN/d(f_{SIG}/f_{REF}))

Figure 4.2: Comparison between linear VCO-based quantizer and BF-quantizer.

Detection of sub-mV neural signals in this scheme necessitates a sophisticated high-gain low-noise AFE [45]. In contrast, the BF quantizer compares f_{SIG} to a reference that has a similar f_{REF} frequency. A standard D-flip-flop circuit is used to generate a beat frequency clock with a frequency of $\Delta f=|f_{REF}-f_{SIG}|$ [7]. The beat frequency is then converted to a digital count N by measuring the number of f_{REF} cycles that fits in a single beat frequency period. The BF count N can be expressed as:

$$N = \left\lfloor \frac{f_{REF}}{f_{REF}-f_{SIG}} \right\rfloor \quad \textbf{4.3}$$

To illustrate the BF quantizer operation further, let us consider the case where f_{SIG} is lower than f_{REF} by 1%. That is, $f_{SIG}=0.99f_{REF}$. This can be easily achieved in a real chip using trimming capacitors. The BF count N in this case will be 100 since it takes 100 cycles for the faster f_{REF} clock to overtake the f_{SIG} clock. If the count drops to 99, then this corresponds to a frequency difference of 1.010101...% between f_{REF} and f_{SIG} (i.e. $f_{SIG}=0.98989\ldots f_{REF}$) which translates into an f_{SIG} change of only ~0.01%. The same change in the output count (i.e. 100→99) would have required a 1% frequency change for the linear VCO scheme. This analysis indicates that the sensitivity of the BF quantizer is about 100 times higher than that of a linear VCO quantizer for a nominal count of 100. In other words, the beat frequency operation effectively amplifies small voltage differences by the built-in non-linear relationship in (4.3). It's worth noting that the quantization error and sampling time of BFADC depend on the beat frequency. For instance, a smaller frequency difference between f_{REF} and f_{SIG} increases the quantizer gain and thereby reduces the quantization error, at the expense of a longer sampling period. The irregular sampling period can be circumvented by enabling the oscillators with a fixed frequency clock. This

ensures that BF counts are generated at a fixed interval. The lower quantization error is the reason why BFADC achieves an extremely high gain for the dynamic range of interest.

Conventional designs rely on sophisticated amplifiers combined with a large-dynamic-range ADC to prevent the output signal from being saturated due to common-mode noise in the signal and reference voltages. This approach incurs a large area overhead and requires significant design effort. Interestingly, the non-linear relationship of BF quantizer inherently suppresses common-mode noise effects. This is because the neural signal component in f_{REF}-f_{SIG} is amplified by the inverse relationship in (4.3) while common-mode noise contained in f_{SIG} or f_{REF} is not amplified. This unique property allows the BFADC to extract neural signals as small as 100μV from a noisy environment.

4.3 Test Chip Organization

The simplified schematic of the proposed neural recording IC consisting of a simple AFE (TIA gain of 5), passive filters, oscillator, and BFADC is shown in Figure 4.3.

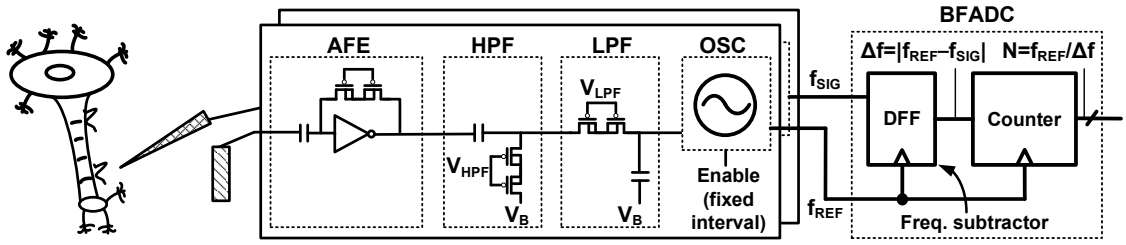


Figure 4.3: Schematic representation of the implemented neural recording BFADC test chip.

The test chip was implemented in 65nm LP CMOS to validate the proposed low-area, all-digital neural recording system. Each subsystem in the recording chain will be detailed in the following subsections.

4.3.1 Analog Front End Circuit

The first stage of the AFE is an AC-coupled digital-inverter based Trans-impedance amplifier (TIA). By applying resistive junction feedback to the inverter, the operating point is fixed at the trip point and any perturbation on the input will be amplified at the output due to the steep slope. The feedback resistor is implemented as a pseudo-resistor by shorting the drains and connecting the body and gate to V_{DD} which puts the devices in cut-off to give the channel a large (MΩ) resistance. The primary drawback of using this configuration is the static current. However, this can be reduced to an acceptably low level by decreasing the supply voltage. In addition, simulation results have shown that very aggressive gate widths near minimum size still give good amplification performance. The trade off with using a smaller device is that the device noise is proportional to the g_m of the device which favors using larger devices. Since the BFADC has such high intrinsic gain, the extremely low area, and fully-digital implementation makes the junction feedback amplifier a pragmatic choice. The output of the amplifier is AC coupled by the first stage of the band pass filter (BPF) which inhibits the DC offset of the TIA from setting the bias of the VCO. The pseudo-resistors are controlled by an off-chip gate bias, which enables tuning the pass band to physiologically relevant signals [41]. In this implementation, the bias voltages are shared to further reduce the implementation overhead.

4.3.2 *Current Controlled Oscillator*

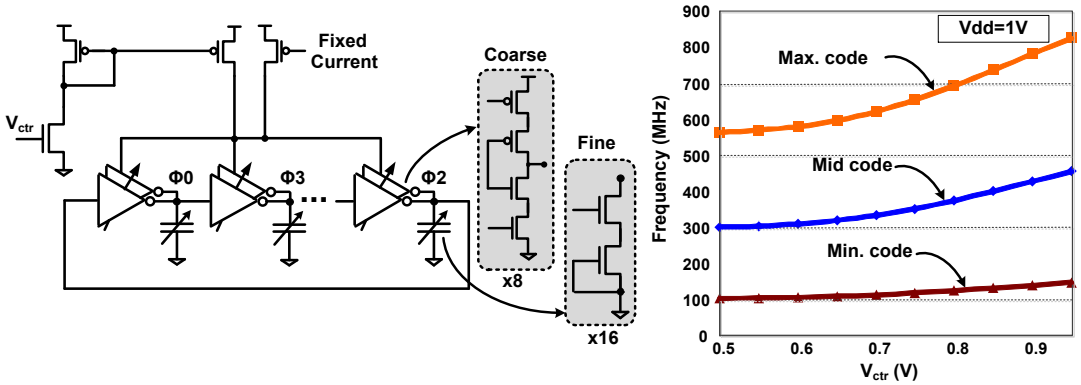


Figure 4.4: (Left) Schematic of current controlled oscillator and (Right) parasitic-extracted simulated range [48].

A major improvement in this design over previous works [47] [45] is that the VCO is implemented as a current controlled oscillator (CCO) shown in Figure 4.4(left) [48]. This work leverages a digital-intensive voltage to current conversion circuit which helps reduce the entire channel area making this configuration attractive for high channel count neural recordings. This replaces the need for a large unity gain buffer to drive the VCO. The frequency tuning is controlled through 3bits of course tuning by enabling parallel driver inverters. This is used to set the operating point, or nominal count, for the BFADC. Fine tuning via 4bits of capacitive loading compensates for process variations between different CCOs. The HPF (section 4.3.1) also sets the operating point for the CCO which is DC coupled through another area-efficient pseudo-resistor and connected to V_{ctr} in Figure 4.3 (left). The frequency-voltage transfer characteristic for three different course tuning points is seen in Figure 4.4.

4.4 Measurement Results

Test chip measurements were performed in three scenarios: bench testing, *in-vitro* saline tank simulation, and *in-vivo* mouse electrophysiology described in section 4.5. Power was supplied from a 9V battery and regulated with discrete voltage regulator ICs. In the controlled bench test, input signals were supplied from an Agilent 33520A function generator. An input of $1mV_{pp}$ at 416.6Hz was applied in which the full ADC chain provided a SNDR of 20.9dB at supply voltage of 0.8V. This is slightly less than previously reported BFADC designs [47][45] due to the new buffer-less CCO having lower K_{VCO}. Additionally, previous works required two references to reconstruct the input signal where this work employs a single reference. In this design the goal was to aggressively cut area while retaining enough performance to remain functional. Figure 4.5 (upper) shows the measured SNDR as a function of the input voltage. All measurements were recorded at a given operating point with no tuning between inputs to simulate an actual use case. Figure 4.5 (lower) shows the analytical relationship between the BF quantizer gain and the default count.

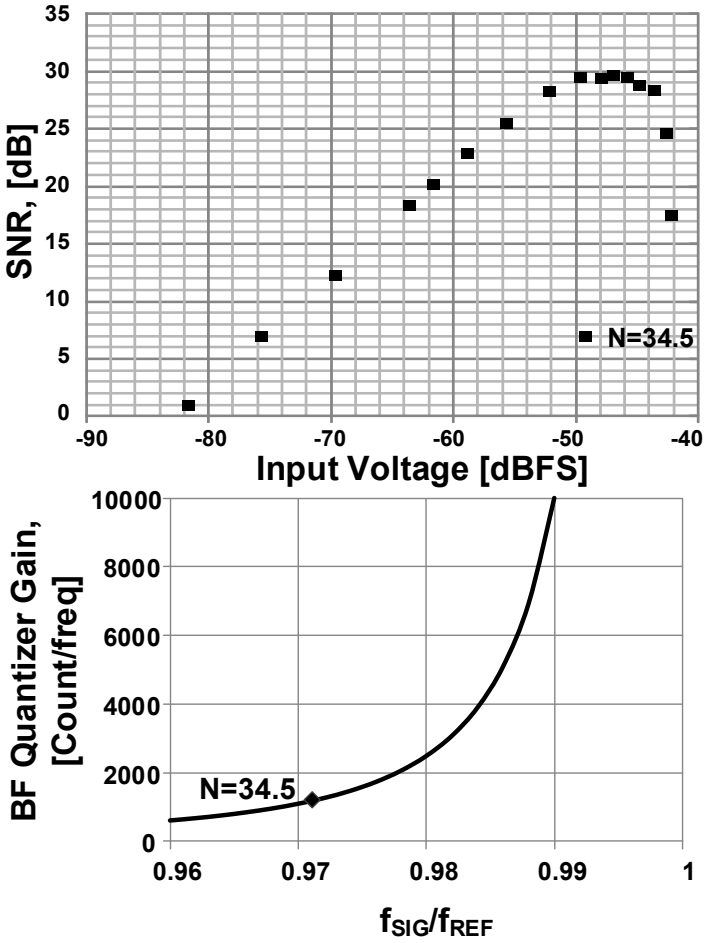


Figure 4.5: (Above) Measured SNDR vs. input amplitude. (Below) BF quantizer gain plot.

Table 4.1: Performance Comparison

Parameters	This Work	[54]JSSC'17	[51]JSSC'16	[53]CICC'15	[45]CICC'15	[52]TCAS-I'15
ADC Type	Beat Freq.	VCO	CT-$\Delta\sum$	VCO-$\Delta\sum$	1-Step BF	Incr.-$\Delta\sum$
Process/Supply	65nm/0.8V	40nm/1.2V	130nm/1.2V	130nm/1.2v	65nm/1.2V	180nm/1.2V
Bandwidth	4.5kHz	200Hz	15MHz	1.7MHz	1.2KHz	4kHz
Sampling Rate	50kHz	3kHz	500MHz	250MHz	50kHz	8kHz
In_{0db} [dBFS]*	-84	-75	-80	-75	-86	-85
$SNDR_{1mVpp}$ [dB]**	20.9	35	20	14	22	22
$ENOB_{1mVpp}$ [b]**	3.17	5.52	3.03	2.03	3.36	3.36
Power	52uW	7uW	20mW	910uW	34uW	34.8uW
FoM @ F_{in} [pJ/Conv]***	683 @ 900Hz	380 @ 3Hz	81.4 @ 4.15MHz	66.6 @ 500kHz	1252 @ 300Hz	424 @ 175Hz
Chip Area [mm^2]	0.046	2.16	1.3	0.04	0.096	0.0564
Area/Ch [mm^2] (Relative)	0.0094 (1x)	0.135 (14.5x)	1.3 (138x)	0.04 (4.3x)	0.078 (8.3x)	0.0564 (5.9x)
Experiment	*In-vivo*	*In-vitro*	-	-	-	-

Input Amplitude at SNDR=0dB, 0dBFS=1.2V* ****Reported at V_{in}=1mV_{pp} ***FoM =Power/(2*BW*2^{ENOB})

The power consumption of the total system (excluding scan and pad I/O power) at 0.8V is 52µW. The input referred noise of our entire recording chain is calculated to be 5.3µVrms by the histogram method given in [49]. The area required for a single channel to be digitized includes the AC coupling capacitor, AFE, BPF, CCO, and BFADC is 0.0094mm^2 as seen in Figure 4.6.

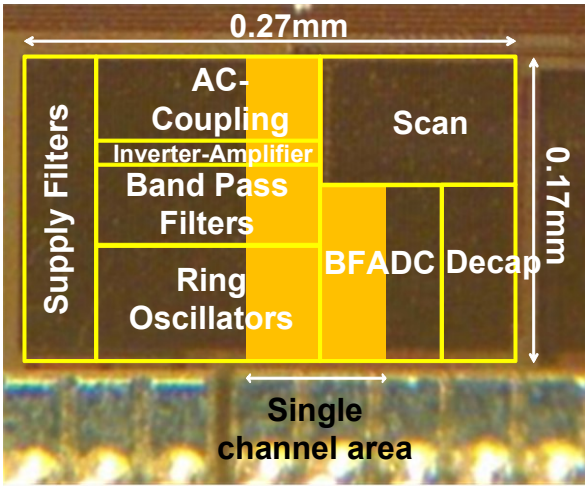


Figure 4.6: Die photo of the test chip in 65nm LP CMOS. Box highlighted in orange represents the area of a single channel.

The reference electrode AFE area can be amortized over many channels and does not require its own quantizer. *In-vitro* recordings were performed in a saline solution with a shielded beaker to reduce external noise [50]. The saline environment simulates the charge transfer mechanism that occurs in the brain. Measured SNDR for a 1mV$_{pp}$ sine wave at 800Hz was 4.5dB *in-vitro*. Table 4.1 compares this work to state of the art time-domain ADCs. The Walden FOM is the same used in [45] [51] [52]. Our reported FOM is higher than [51] [53] due to their larger bandwidth, but otherwise in line with or outperforms state of the art. The work in [54] reported higher SNDR but it has 22.5x lower bandwidth which

could be due to the long required sample period since the architecture is a linear multi-phase VCO. This restricts the efficacy to only local field potentials (LFPs), whereas with the BFADC can record LFPs and spikes.

4.5 Results of *In-vivo* Experiment

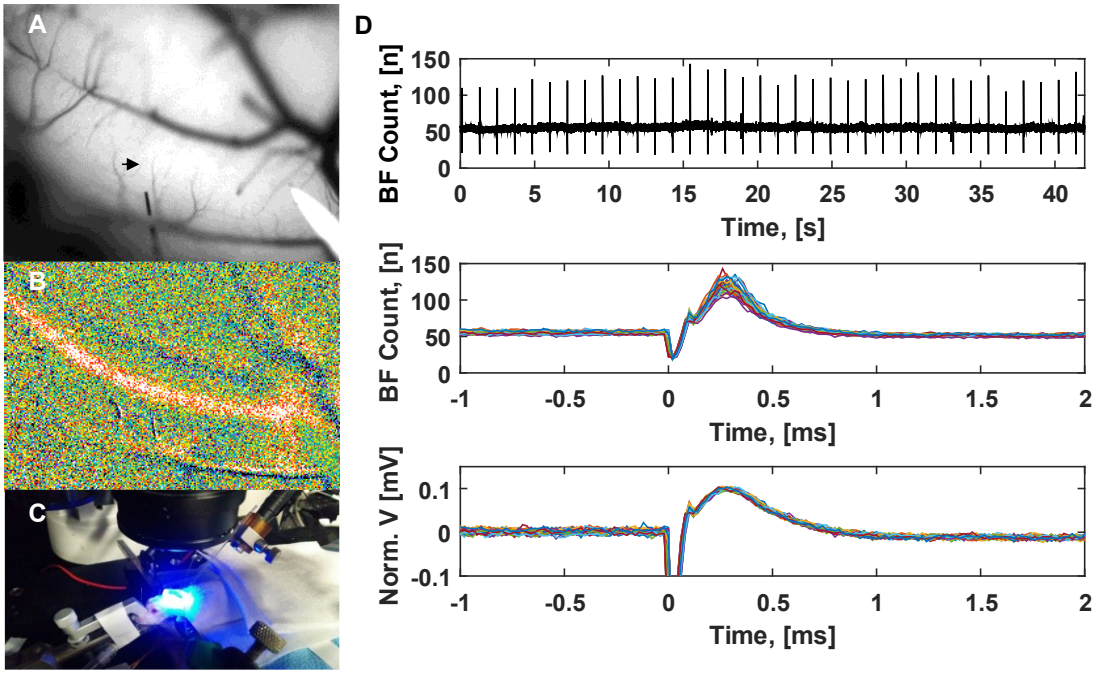


Figure 4.7: Results from the in-vivo recording experiment.

All animal handling procedures were approved by the Institutional Animal Care and Use Committee of the University of Minnesota. Electrophysiology was performed using the neural recording BFADC in an anesthetized (WT)/FVB mouse for recording from the Purkinje fibers (PF) in the cerebellum shown in Figure 4.7a. Figure 4.7b shows the microscope image placement of the stimulating Tungsten microelectrode and glass micropipette recording electrode. Activity-dependent optical imaging was used to determine the location of the PF, indicated by the arrow, using flavoprotein autofluorescence in Figure 4.7c [55]. PF can have two sets of pre-synaptic and post-

synaptic activations from stimulation, manifested here as positive (depolarization) and negative (hyperpolarization) signal swings. Figure 4.7d (top) shows one section of the experiment recorded with the BFADC. Stimulation was applied at 1Hz and evoked potentials were observed and overlaid in the middle plot. PF have two sets of activations corresponding to the notch and the main peak seen in the traces. No digital filtering or off-line processing was applied to the data other than (4.3) in Figure 4.7d (bottom). The BFADC gives a relative measurement, so extracellular potentials were estimated by applying a $1mV_{pp}$ input, applying (4.3), and then linearly scaling to match a $1mV_{pp}$ digitized output. A benefit of the non-linear quantization is the BFADC does not saturate during stimulation artifacts manifested as the large hyperpolarization. The difference in the relative magnitudes of the peaks in the middle and bottom plots highlights the non-linear quantization in the BF counts. The plots confirm the ability of the BFADC to record both sets of PF activations.

4.6 Conclusion

In this chapter the implementation of a fully-digital, low area neural recording system based on the BFADC was presented. It was able to achieve 20.9dB SNDR for a 1mVpp signal while occupying a meager $0.0094mm^2$. The high sensitivity to small signal changes obviates the need for sophisticated high gain amplifiers and massive filter circuits which makes the BFADC architecture well suited for neural recording as evidenced by the *in-vivo* experiment.

Chapter 5. Time-based Photoplethysmography Machine Learning Algorithms

5.1 BiometricNet: Deep Learning based Biometric Identification using Wrist-Worn PPG

5.1.1 Introduction

Biometrics as defined by the International Organization for Standardization (ISO) as the "automated recognition of individuals based on their behavioral and biological characteristics" where common characteristics include fingerprints, iris, DNA, voice and gait analysis [56]. State-of-the-art work related to biometrics have also focused on one-dimensional physiological signals, i.e. electrocardiograms (ECG) [57], electroencephalograms (EEG) [58], phonocardiogram (PCG) and lastly photoplethysmography (PPG) [59]. If physiological signals can correlate to the clinical condition of a subject, this can pave the way for personalization and identification of individuals. Such signals also have the distinct advantage of enabling continuous authentication systems since the measurements modalities are pervasive and automatic. ECG is more established and robust for identification than PPG. The primary drawback is that it requires multiple electrodes placed on the chest making it inefficient in terms of wearability for daily life usage. PPG signals are obtained in a less intrusive manner from smartwatches that users voluntarily wear. PPG signals are obtained from pulse oximeters which emit light on the skin and measure the change of light intensity, which is either transmitted or reflected through the skin. The periodicity of the reflected/transmitted light

corresponds to the cardiac rhythm, often used for heart rate estimation as will be seen in section 5.2. PPG signals can be acquired from various positions such as earlobes, fingertips or wrist, with the latter considered as a convenient position for unobtrusive daily use. The primary limitation to PPG based biometric identification is the acquisition is vulnerable to motion artifacts (MA) in normal daily living conditions. This correspondingly distorts the signal fidelity and inhibits identification quality.

PPG-based biometric identification has been a studied, but prior art has used signals collected in clinical settings which are less prone to MA. This renders these approaches unsuitable for usage in daily life, given the fact that motion artifacts are ever present during daily activity of the user [58]. Earlier efforts on PPG-based identification utilized data collected from the finger and analyzed using frequency domain analysis (Fourier analysis), correlation, peak detection, feature extraction and classification, employing fuzzy-logic [58] [59] and Linear Discriminant Analysis [60]. These studies yielded high accuracies, approximately between 90-95%. Additional approaches employed predictive learning methods, but relied on hand crafted features, sometimes as many as up to 40 features, used in conjunction with a kNN classifier [61]. On this backdrop, a recent work has focused on a two-stage procedure involving clustering (using 11 hand-crafted features) and deep learning techniques (Restricted Boltzmann Machines and Deep Belief Networks) [62]. Contrary to prior work, it was evaluated on PPG signals collected from the wrist using green light under ambulant conditions. Results have been promising which paves the way for future research involving no feature extraction and entirely relying on the neural network for subject identification. This paradigm shift poses research challenges on the

use-case of such a system involving deployment, model update based on the changing cardiac and physical conditions of the subjects and hardware security and privacy which are yet to be explored on a system level.

In this section, the key contribution lies in the fact that a completely data-driven approach based on convolution neural network (CNN) in conjunction with long and short term memory (LSTM) is used for modelling the underlying temporal sequence in the biological data of each subject. This relieves the limitations of data processing and heuristics involved in popularly used classification schemes. The proposed framework, *BiometricNET*, has been motivated by the fundamentals of Deep Neural Networks (DNN). It eliminates hand-engineered features because the first layer acts as an automated feature extractor. The section is further structured as follows: section 5.1.2 introduces the dataset used in this chapter, as well as 5.2 and 5.3 because it has become the benchmark of the community for comparing PPG algorithms. Section 5.1.3 describes the problem formulation using our learning-based approach. The proposed methodology highlighting the DNN fundamentals and the developed network architecture, *BiometricNET* have been detailed in section 5.1.4. The results have been presented in section 5.1.5 and the conclusions have been drawn in section 5.1.6.

5.1.2 IEEE 2015 Signal Processing Cup Dataset

In this work, a completely personalized approach using DNN for robust biometric identification is adopted. For this exploration, as well as the applications presented in section 5.2 and section 5.3, the algorithm is evaluated on the IEEE SPC 2015 database comprising of PPG signals of 5-minute duration, from 20 healthy subjects, age ranging 18

to 58 [63]. Each subject's data contains two channels of PPG, recorded from the wrist (dorsal) using a pulse oximeter with green LED (wavelength: 515 nm); tri-axial accelerometer signals also recorded from the wrist, and a channel of ECG recorded from the chest using wet ECG electrodes, all recorded simultaneously. The ECG is not required for biometric identification because the class labels are the subject index, not related to the ECG. All signals were sampled at 125 Hz and transmitted to a computer through Bluetooth. PPG window (frame) lengths considered for this exploration was 8s (sliding by 2s), like ECG-HR computation. The subjects performed three types of activities. First, Type1 (T1), performed by subjects 1-12, involving walking or running on a treadmill with the following speeds in order: 1–2 km/h for 0.5 min, 6–8 km/h for 1 min, 12–15 km/h for 1 min, 6–8 km/h for 1 min, 12–15 km/h for 1 min, and 1–2 km/h for 0.5 min. The subjects used their hand (with wristband) to pull clothes, wipe sweat on forehead, and push buttons on the treadmill. Second, Type2 (T2), performed by subjects 13, 14, 15, 18 and 20, involved in forearm/upper arm exercises (e.g. shake hands, stretch, push, running, jump, and push-ups). Last, Type3 (T3), performed by subjects 13, 15, 16, 17, 18 and 19, involving intense arm movements (e.g. boxing). Hence, a total of 20 subjects and 23 records in aggregate, since subjects 13, 15 and 18 were involved in both T2 and T3. For *BiometricNET* only T1 subjects have been considered. This is due to the limited recordings from all subjects which makes it challenging to train learning models. It is feasible to compare small datasets as long as the motion types are consistent between subjects.

5.1.3 Problem Formulation

The problem is formulated as a binary classification task (one vs all) wherein a given subject is identified against a group of subjects based on the individual's PPG. For training the network, there are significantly unbalanced class distributions, given that there are 12 subjects, the imbalance is 11:1. During training, the errors of the target class are weighted accordingly to allow the network to learn the underlying data distribution instead of resorting to predict the dominant zero-class. The benefit to this approach is that the network can learn subject-specific features from the PPG. However, the main drawback is that the network will need to be trained for each new user. In a commercial setting this would be very costly, since data would have to be collected during an initial enrollment period, networks would have to be trained, and weights downloaded back to the device. The training cost could be minimized by fine-tuning [64]. However, this formulation is natural, because it would not be feasible to implement this problem as multi-class classification since training examples of every possible customer would not be available during the initial training period. Furthermore, it is not practical to have a class for each subject since that would lead to an unbounded number of classes.

To provide the reader some background on prior work a brief description on [62] will be presented, since it is one of the most recent investigations using wrist-worn, 'green' PPG under motion, which proposes a two-step process. The first phase groups the subjects based on 11 extracted features. These clusters are formed based on attributes such as gender or physical condition, but the authors also mention different forms of motion would fall under different clusters. This means that a single person could end up in more than one

cluster depending if the given PPG was taken at rest, walking, biking, or any other ambulatory motion. The second stage utilizes a deep fully connected neural network to classify the subject within the cluster. Using user-defined features is susceptible to human bias and limits the ability of the network to learn fully from the data. Additionally in this approach, a model is required for each group which still proposes significant overhead. During this two-step approach, if the subject is not correctly filtered into the correct segment initially, there is no possibility to get the correct identification. To overcome these shortcomings, *BiometricNet* is an entirely data-driven personalized deep learning approach.

5.1.4 BiometricNet Framework

An overview of the data processing pipeline is presented in Figure 5.1. The PPG data samples are pre-processed with a band-pass 4th order Butterworth filter having the cut-off frequencies 0.1 – 18 Hz, which primarily restricts the high frequency noise component and drifts from the signal of interest. The signal is further normalized to zero mean and unit variance.

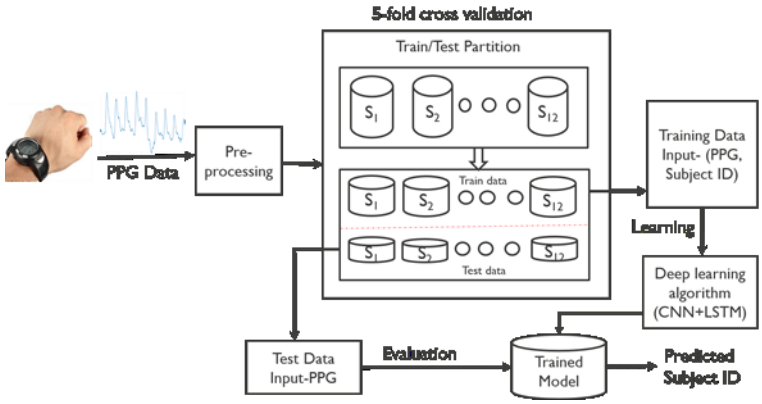


Figure 5.1: Overview of the proposed methodology for biometric identification

5.1.4.1 Deep Neural Network: CNN + LSTM

DNN enables feature extraction directly from data, thus enabling the learning of task-adapted feature representations [65]. The taxonomy of deep neural models mainly includes Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Stacked Auto-encoders (SAE). CNNs are characterized by an initial layer of convolutional filter layers that consists of a set of weights which are passed over the input. This is followed by non-linearity activation functions, such as rectified linear units. The activations are sub-sampled (also known as pooling). After the CNN layers, a fully connected layer is employed to perform the classification. LeNet [19] was one of the very first CNNs which helped propel the field of Deep Learning to prominence. The structure of the CNN enables it to integrate the information from the different filters at various levels of abstraction. The stacking of multiple convolutional layers helps to achieve automatic feature extraction, where denser layers capture more complex or differentiating features.

Analyzing time series data often necessitates inferring the sequential/time-dependent information. This is where RNNs prove to be an effective choice since they are able to incorporate contextual information from past inputs, having the advantage of being robust to localized distortions in the input sequence along the time. One key limitation of deep RNN structures is the long chain problem (vanishing gradient) wherein information from previous computations rapidly attenuates as it progresses through the data flow [66]. This arises from the chain rule as the gradients are backpropagated through multiple layers. As the number of layers increases, the number of multiplicative terms grows. Large gradients, greater than one, will explode rapidly to infinity, and small gradients, less than

one, will quickly approach zero. This is where a variant of RNN, LSTM comes in handy, in which the scalar valued hidden neuron is replaced with the LSTM memory block [66]. The LSTM memory block is inspired by a computer memory cell where context-dependent input, output, and forget gates control what is written to, read from, and kept in the cell in each time-step. It also breaks the multiplication chain as the gradients are added instead of multiplied. In this way, it is easier for the network to store a given input over many time steps, in effect helping LSTM layers to capture temporal properties.

5.1.4.2 BiometricNet Architecture

In order to realize the data driven approach the network is allowed to learn the discriminatory features, accomplished by using a two layer unidimensional CNN. The output from the 1-D CNN is then fed into two LSTM layers and finally, the LSTM output is passed through a dense layer with SoftMax activation function for classification, constituting *BiometricNET*. The 1-D CNN serves the purpose of a feature extractor. The input is convolved with the filters to generate points in the temporal-feature domain. Corresponding layers use these features to convolve with additional filters to generate the final features from the time-series PPG input. A drawback of using the CNN is that the generated features are not completely phase invariant. Depending on the time of occurrence of the heart beat relative to the beginning of the sample, the relevant features will be slightly shifted. The use of LSTM, which is primarily instrumental in capturing the temporal dependency in the sequence of historical local trends of the underlying cardiac activity inherent in the PPG signals, further helps to recover from the phase offset.

5.1.4.3 Implementation Details

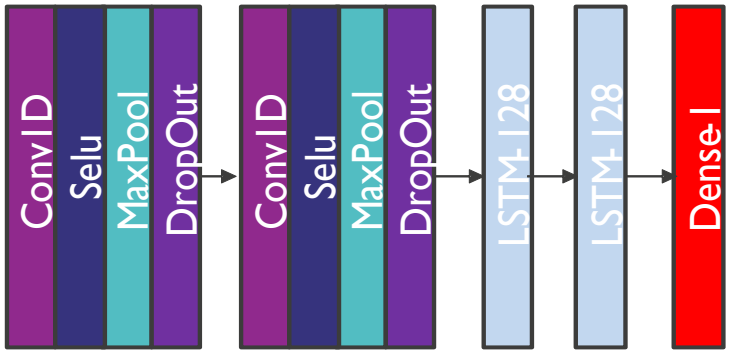


Figure 5.2: *BiometricNet* topology

The PPG identification network was trained on an Nvidia Tesla K80 GPU and is modeled in Keras 2.0.4 version [67] configured to use theano [68] version 0.0.9 as the backend. Each CNN layer consists of a 1-D CNN operation with Scaled Exponential Linear Unit (SELU) activation [69], whereas each LSTM layer used hyperbolic tangent (tanh) function. The max-pooling layers used a pool size of 4, and a dropout layer with rate 0.1. Root Mean Square Propagation (RMSProp) is used as the optimizer with the default hyperparameters which is recommended for training recurrent networks [70]. As described earlier, the class loss is weighted to offset the class imbalance. Training batch size is set to 25 which minimizes training time while keeping sensitivity to individual inputs. Hyperparameters such as filter length and number of filters, layers, and LSTM units were optimized using a heuristic grid search method, details of which are presented in the following section. Figure 5.2, illustrates the proposed *BiometricNET* topology for this exploration.

5.1.5 Results

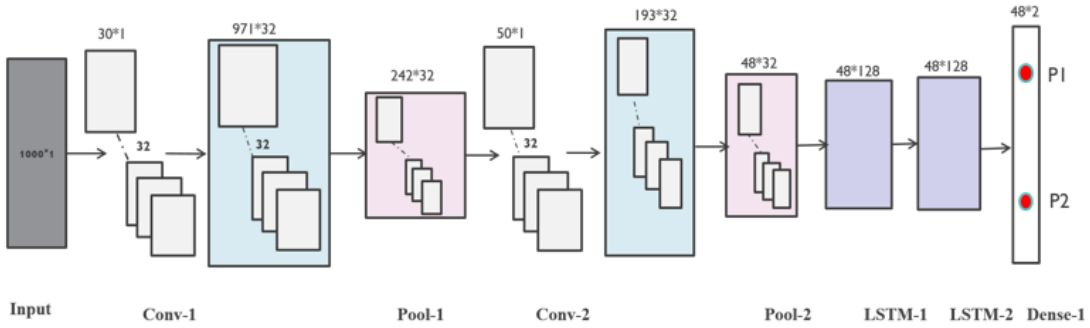


Figure 5.3: Details of the filter sizes used in *BiometricNet* found from the grid search

The details of the hyperparameter grid search are presented in Table 5.1 with the final model annotated in Figure 5.3. The performance of the model, in conjunction with a given set of hyperparameters, has been evaluated by standard classification metrics, described briefly below. The results in Table 5.1 represent the average outcomes of the various metrics as calculated over the 12 subjects, wherein C_1 and C_2 represents each CNN layer; *sF* and *nF* represents the filter size and number of filters respectively.

a) *Accuracy (A)* - ratio of correctly classified observation to the total observations.

b) *F1- Score (F1)*- weighted average of precision and recall.

c) *Recall (R)* - ratio of correctly predicted positive observations to the all observations in actual class.

d) *Precision (P)* - ratio of correctly predicted positive observations to the total predicted positive observations.

Table 5.1: Parameter Grid Search Results

Network Type	Network Configuration				A	F1	R	P
	sF		*nF*					
	C_1	C_2	C_1	C_2				
1	15	20	15	15	0.95	0.77	0.75	0.82
	15	30	15	15	0.95	0.80	0.77	0.85
	15	40	15	15	0.95	0.80	0.77	0.85
	15	**50**	15	15	0.96	0.82	0.81	0.85
2	30	20	15	15	0.95	0.81	0.78	0.86
	30	30	15	15	0.95	0.82	0.79	0.86
	30	40	15	15	0.96	0.82	0.80	0.89
	30	**50**	15	15	0.96	0.83	0.80	0.89
3	30	20	32	15	0.95	0.81	0.78	0.86
	30	30	32	32	0.96	0.84	0.81	0.88
	30	40	32	32	0.96	0.84	0.81	0.89
	30	**50**	**32**	**32**	**0.96**	**0.86**	**0.84**	**0.89**

It can be observed from Table 5.1 that in all three network types, configuration with filter size (*sF*) 50 in second layer (C_2) performs comparatively better (highlighted in bold). Furthermore, among the selected best configuration in three network types, the network with a filter length (*nF*) of 32 and *sF* of 30 and 50 (marked in bold), achieved the best performance demonstrating an improvement of approximately 2% and 4% in F1 score and recall value respectively compared to the other two networks highlighted. Hence, one can conclude that the configuration 30-50-32-32 provides the best performance for our investigated problem, demonstrating an average accuracy, F1 score, recall, and precision of 0.96, 0.86, 0.84 and 0.89 respectively. Increasing the *sF* and *nF* values, beyond this did not improve the overall performance and are not shown here for brevity. It is worth mentioning that a similar exhaustive exploration was also done with the LSTM size and the selected 128 units for both layers provided the best performance (details have not been shown again for sake of brevity) in conjunction with the demonstrated CNN architecture.

The dense layer used at the end has a softmax activation function and has 2 neurons. One can visually see the difference in the weights pertaining to the dense layer for classification of two example subjects (1 and 5) have been shown in Figure 5.4. This can be interpreted as the brighter units are feature locations that more likely describe that individual. This combined with a high feature level at that location will encourage the classification to favor that individual.

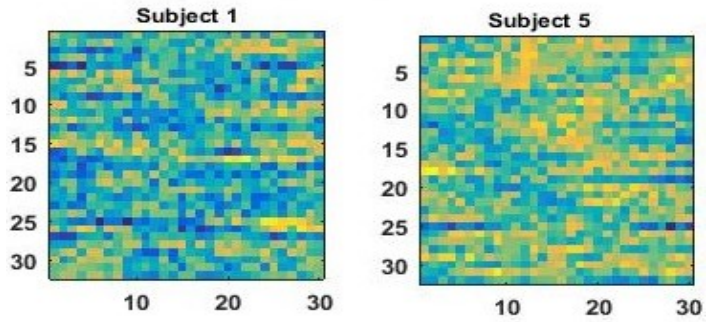


Figure 5.4: Difference in dense layer weights between subject 1 and 5

The evaluation results over 12 subjects have been presented in Table 5.2. It can be observed that subject 4 and subject 9 have the best performance in terms of average accuracy (98%) and precision (98%). Five-fold cross-validation is utilized to increase the robustness of the model and to prevent overfitting. For real-time execution on an embedded device, driven by application requirements, a complexity analysis of the proposed architecture was performed. A single evaluation of the four layer network requires approximately 452K MACs (multiply-and-accumulate dot products). Current trends in architecture development hold promise and make this number achievable [71].

Table 5.2: Performance of *BiometricNet* on SPC Dataset

Subject	Average (5-CV) Accuracy	F1 Score	Recall	Precision
S1	0.95	0.75	0.68	0.86
S2	0.96	0.83	0.82	0.83
S3	0.94	0.73	0.76	0.70
S4	0.98	0.92	0.87	0.98
S5	0.96	0.83	0.90	0.79
S6	0.96	0.88	0.83	0.93
S7	0.94	0.73	0.67	0.82
S8	0.98	0.94	0.94	0.94
S9	0.98	0.94	0.91	0.98
S10	0.98	0.95	0.95	0.97
S11	0.97	0.90	0.88	0.93
S12	0.98	0.94	0.91	0.97
Mean	**0.96**	**0.86**	**0.84**	**0.89**

5.1.6 Conclusion

This section presents a paradigm shift away from hand crafted features for biometric identification and reports the efficacy of a personalized, data-driven, approach using deep learning on wearable PPG signals collected in an environment affected by motion artifacts. The *BiometricNET* topology, using two layers each of CNN and LSTM yields best accuracy of 98% and an average accuracy of 96% on all 12 subjects. In view of the present work, future explorations would focus on evaluating the network on other databases having wrist PPG signals acquired amidst motion. Other avenues of this work will focus on energy-efficient execution of the algorithm on wearable devices in real-time on a microcontroller or android for mobile platform or hardware solutions (ASIC, SoC) using the schemes proposed in [71] [72].

5.2 CorNET: Deep Learning framework for PPG based Heart Rate Estimation and Biometric Identification in Ambulant Environment

5.2.1 Introduction

Cardiovascular monitoring using wearable sensors has primarily focused on processing electrocardiography (ECG) for key applications – heart rate (*HR*) monitoring, disease detection/prognosis, sports, biometric identification (*BId*), etc. [73] [74]. ECG is limited in terms of its placement (requires ground and reference sensors proximal to chest) for signal fidelity, making it inefficient in terms of wearability in ambulant daily living conditions as mentioned in section 5.1.1. The primary limitation is the low signal-to-noise ratio due to motion artifacts (MA) during ambulatory conditions. MA are caused by several factors – physical activity, ambient light leaking through the widening gap between sensor and the skin surface during motion and change in blood flow due to movements. This causes the spectral component of the MA to overpower the heart-beat related PPG component [63] as seen in Figure 5.5.

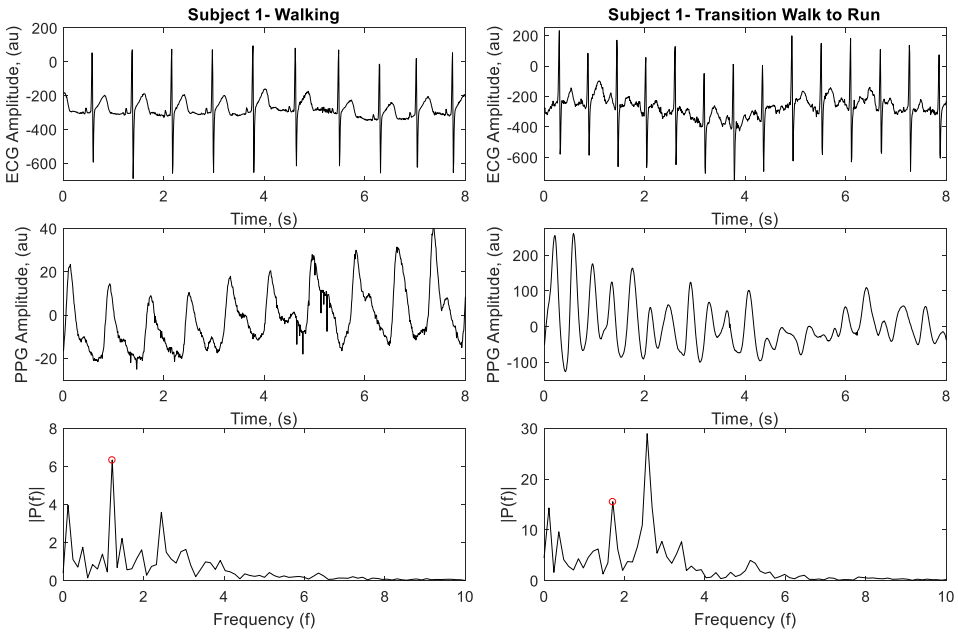


Figure 5.5: Raw ECG, PPG signal and spectrum while walking (left) and during transition from walking to running (right) respectively. The highest PPG spectral peak does not coincide with true HR (encircled) during intense motion.

A host of signal processing techniques have been proposed to remove/attenuate MA using adaptive filtering [75] [76], Kalman filtering [77], wiener filtering [78], independent component analysis [79], empirical mode decomposition (EMD) [80], spectral subtraction [81] [82] and feature-engineering based learning algorithms [83] [84]. Such methods have often used a motion reference signal from an external sensor (e.g. accelerometer), for detecting and removing MA resulting from motion. The majority of this research was propelled by the IEEE Signal Processing Cup (SPC) 2015, focusing on HR estimation from wrist-worn, two green light illuminated PPG channels, collected during vigorous physical activities [63]. Although successful, the reported improvements in performance are usually accompanied by heuristic thresholds or a large number of expertly-tuned free parameters which could prevent generalization of the developed methodologies.

In this section, the *BiometricNet* architecture, presented in section 5.1, has been leveraged for heart rate estimation. This is accomplished by using a single unit in the dense layer with linear activation function which predicts a real-valued output, instead of generating a binary classification with two units in *BiometricNet*. This framework is called herein, *CorNET*. The framework was evaluated on the widely used SPC database and achieves a mean absolute error of 0.48±0.19 beats per minute (BPM) for *HR* estimation on 23 PPG records collected during various physical activities

This section is structured as follows: section 5.2.2 introduces the prior art and section 5.2.3 lays out the problem formulation using a learning-based approach. The proposed methodology, highlighting the DNN fundamentals and the developed network architecture, *CorNET*, have been detailed in section 5.2.4. The results, comparison with state-of-the-art approaches and complexity analysis for *CorNET* have been presented in section 5.2.5. Finally, the section is concluded and future research avenues have been discussed in section 5.2.6.

5.2.2 Prior Art

State-of-the-art research using wrist-PPG has primarily focused on using green light [63], having a shorter wavelength (in comparison to red/infra-red), as the illuminating source, since they provide the distinct advantage of producing large intensity variations to cardiac modulation and yields a better signal-to-noise ratio (SNR) [85]. Moreover, reflective system, with LED and photodetector (PD) on the same side is the preferred mode (in comparison to transmitive) due to user comfort [86]. An illustration of the effect of MA on *HR* estimation has been shown in Figure 5.5, with an example PPG segment collected

during walking and transition from walking to running. The spectral information, reflected in the peak due to MA is higher compared to the *HR* peak (encircled in bottom plot). It is evident from Figure 5.5, that the widely used spectral information has high variance and suffers from saturation effects from the MA. The TROIKA framework [63], based on signal decomposition, sparsity-based spectral estimation and peak tracking, was successful in estimating HR every 2 seconds (s) from 8s of MA-affected PPG windows and introduced the SPC database, reporting 2.34 BPM error on 12 subjects. It was followed up by an improvement, formulating a multiple measurement vector model which computed the spectra of PPG and acceleration jointly, reporting 1.28 BPM on the same dataset [87]. Recently, an approach based on Wiener filtering and phase vocoder (WFPV) has produced comparative results with 1.02 and 1.97 BPM on 12 and all 23 PPG recordings respectively [78].

Further developments based on short-time Fourier transform [88], adaptive normalized least mean square (NLMS) filters [89], singular value decomposition [76], and wavelet decomposition [90] using both PPG channels and/or accelerometer data (as motion reference) have been successfully used. Random Forest, in conjunction with features, were used to detect beat vs inter-beat samples, allowing HR estimation with 2.86% classification error [83]. Another recent approach has used probabilistic methods, feature extraction and a 3-layer multi-layer perceptron with 22 neurons, reporting 2.81 BPM on the 23 PPG recordings of the complete SPC [84]. One of the most recent works uses Wiener Filtered PPG subtracted from the accelerometer spectrum to remove MA [91]. A crest factor, or the ratio of the peak value to the RMS of the sample window, is used to signal transitions

between a Finite State Machine which uses different thresholds based on the motion category the user is in. The algorithm has a strong bias towards SPC because the distribution of the time the algorithm spends in each state across the different activities is unbalanced. The algorithms based on spectral processing, inevitably require heuristic thresholds and custom post-processing steps, whereas the learning algorithms have relied on data-driven feature engineering.

5.2.3 Problem Formulation

In this chapter, the *HR* prediction is made from an ECG-assisted supervised framework for *HR* estimation from PPG data. During the training phase, the relationship between each PPG window (frame) and the *HR* computed from corresponding ECG frame (considered as ground truth) are learned by the network. Once trained, the model predicts the *HR* for new, unseen, examples of PPG data. The predictions are verified against the ground truth *HR* and the error magnitude is averaged over the number of observations and reported in beats per minute (BPM). A personalized use-case scenario is formulated in the next section, for our proposed framework. The basic premise of this use-case rests on the fact that biological signals are influenced by various physiological factors – age, sex, height, weight/body-mass-index, etc. and most importantly the cardiac condition of a given subject. Therefore, *CorNET* is developed to build a model which learns the underlying subject-specific data distribution. *CorNET* needs to learn all possible variations inherent within the data. For this, cross-validation over the training set is employed. Cross-validation breaks the data into training and testing groups. The training data is used to learn the weights of the model, and the test data is only shown to the model after training in the

evaluation phase. After this, a new partition of the data is sampled and the process repeats. This is a standard way to ensure the model is robust and generalizes as it has the opportunity to learn and be evaluated on each sample in the dataset. The reader is referred to section 5.1.2, IEEE 2015 Signal Processing Cup Dataset for a description of the dataset used. In this study activities T1, T2, and T3 are evaluated. With the problem formulated the next section will describe the *CorNET* approach to generating accurate *HR* estimation.

5.2.4 CorNET Framework

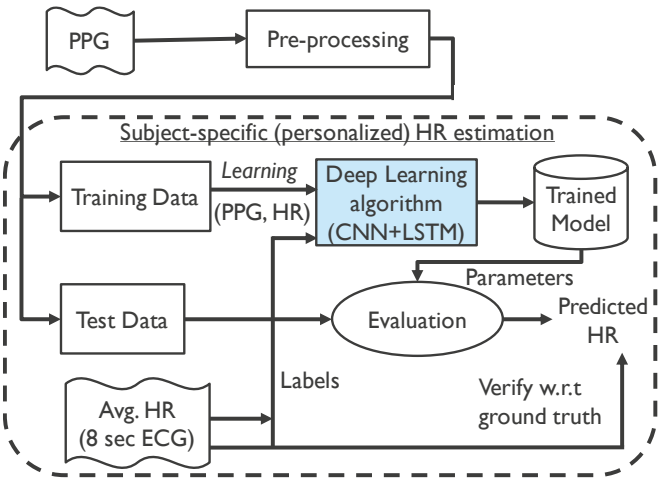


Figure 5.6: HR Estimation methodology and dataflow

An overview of the framework for *HR* estimation is shown in Figure 5.6, illustrating the ECG-HR assisted training and validation. It should be noted that this concept is similar to Figure 5.1. The main difference is that HR estimation is formulated as a regression problem, whereas biometric identification is developed as a classification problem. Figure 5.7 clearly shows the similarity between the *HR* estimation network and the *BId* network. The same feature extraction engine, two CNN and two LSTM layers, can be used to generate relevant features from PPG for two different tasks. One could speculate that the structure could have an intrinsic relationship to the PPG signal. At a deeper level, the CNN

can be thought of as looking for 1-D sequences that correspond to unique patterns in the PPG. The LSTM can then integrate and store these activations over time windows, making the network robust to phase shifts in the incoming signal. The combination of these two techniques yields satisfactory results in both applications and time will tell if this architecture gains traction in the field.

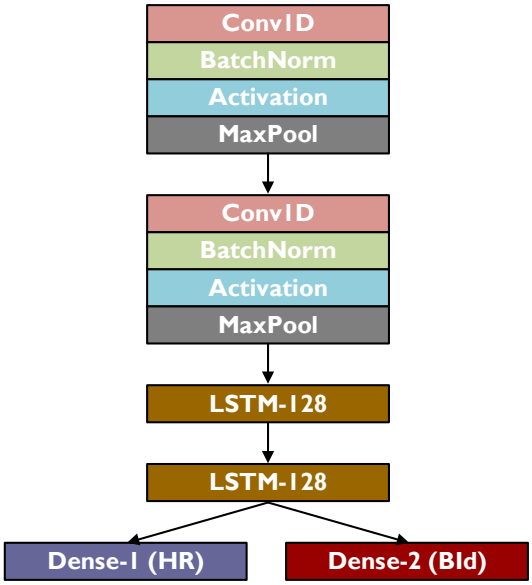


Figure 5.7: *CorNET* (HR) and *BiometricNET* (BId) share the same feature extraction engine

5.2.5 Results and Analysis

The performance of *CorNET* has been evaluated with respect to standard metrics [63], which have been briefly described here. The metric considered is absolute error (*AE*), computed as the absolute difference between true HR_E (from ECG) and estimated HR_P (from PPG), i being the window, as in (**5.1**).

$$AE_i = abs(HR_{E_i} - HR_{P_i}) \quad \textbf{5.1}$$

Correspondingly, the mean absolute error (*MAE*) and the standard deviation of the absolute error (*SDAE*) over all processed windows, are computed and compared with state-of-the-art work, as summarized in Table 5.3. The group metrics have been computed for the first 12 subjects, performing T1, records 14-23, involved in T2 and T3 and finally the results for all 23 records, enabling a comparative evaluation with [63], [87], [80] and [78]. *CorNET* achieves a *MAE*±*SDAE* of 0.52±0.19, 0.44±0.18 and 0.48±0.19 for records 1-12, 14-23 and 1-23 respectively, reflecting the improvement compared to state-of-the-art methodologies.

The correlation between HR_E and HR_P over all records is shown in Figure 5.8(a), having a correlation coefficient of 0.998. Furthermore, a comparison of HR_E and HR_P for subject 9 and subject 1 have been shown in Figure 5.8(b) and Figure 5.8(c) respectively. This shows the best and the worst performance of *CorNET* over the first 12 subjects of the experimental cohort. Shown in Table 5.3, it is interesting to note that *MAE* for records 14-23 is less compared to 1-12, although the former involves random and intense arm movements, as described in section 5.1.2. On closer observation, it is evident that higher *MAE* (in records 1-12) is particularly dominated by record 1 (1.37), with the first sets of 6-8 and 12-15 km/h, incurring maximal error (cf. Figure 5.8(c)). An activity-wise (T1) analysis of *MAE*, shown in Table 5.4, highlights the high *MAE* (> 1 BPM) for these two phases, where the trained model potentially overfits on the test data samples.

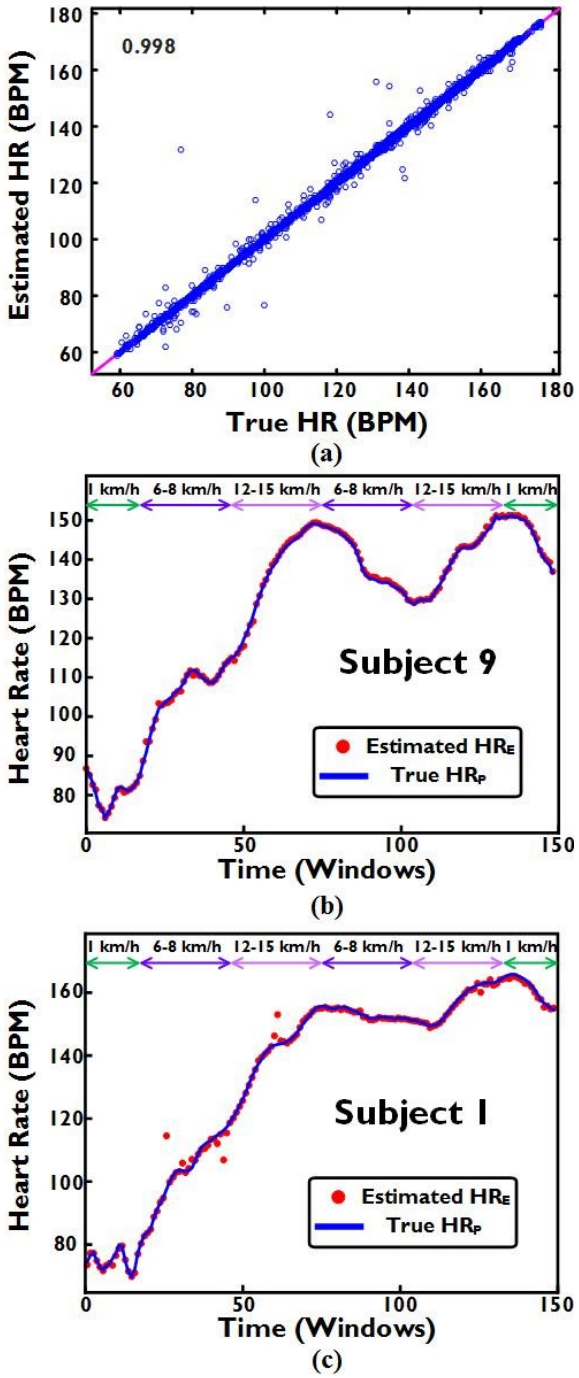


Figure 5.8: Results from the *HR* estimation problem

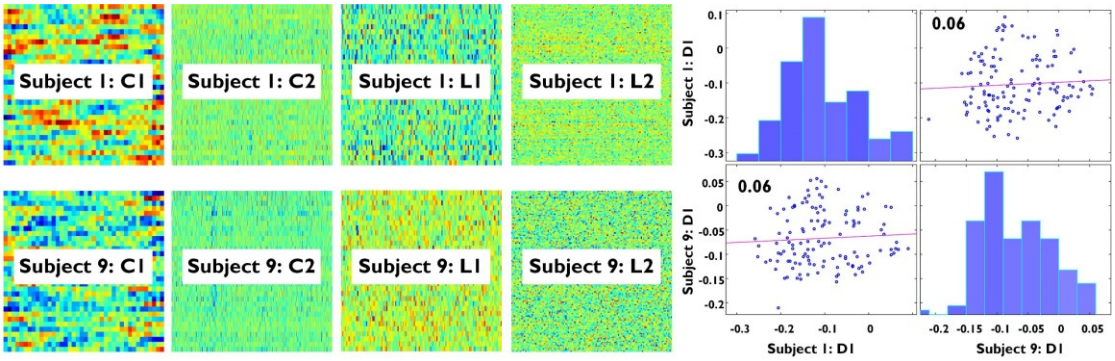


Figure 5.9: Weights of *CorNET* for subject 1 and 9

The weight matrix at the output of each layer for records 1 and 9 have been shown in Figure 5.9, highlighting the differing subject-dependent patterns as the input PPG time series propagates through the network layers. The differences in the distribution and uncorrelated (*viz.* 0.06) behavior of the last layer leads up to different HR_P's for the two example subjects. The low correlation opens further exploration of a general model.

Table 5.3: Performance Comparison for *HR* estimation

Metric	Record	Troika [63]	Joss [87]	EEMD [80]	WFPV [78]	CorNET
	1	2.29±2.18	1.33±1.19	1.70	1.25±1.15	1.37±0.87
	2	2.19±2.37	1.75±1.66	0.84	1.41±1.30	0.60±0.30
	3	2.00±1.50	1.47±1.27	0.56	0.71±0.59	0.39±0.18
	4	2.15±2.00	1.48±1.41	1.15	0.97±0.88	0.57±0.29
	5	2.01±1.22	0.69±0.51	0.77	0.75±0.57	0.39±0.02
	6	2.76±2.51	1.32±1.09	1.06	0.92±0.75	0.33±0.09
	7	1.67±1.27	0.71±0.54	0.63	0.65±0.50	0.35±0.07
	8	1.93±1.47	0.56±0.47	0.53	0.97±0.83	0.34±0.19
	9	1.86±1.28	0.49±0.41	0.52	0.55±0.48	0.27±0.03
	10	4.70±2.49	3.81±2.43	2.56	2.06±1.29	0.67±0.14
	11	1.72±1.29	0.78±0.51	1.05	1.03±0.68	0.51±0.09
	12	2.84±2.30	1.04±0.81	0.91	0.99±0.70	0.41±0.06
	13	-	-	-	3.54±4.08	-
	14	6.63±8.76	8.07±10.9	-	9.59±12.2	0.31±0.08
	15	1.94±2.56	1.61±2.01	-	2.57±3.16	0.06±0.03
	16	1.35±1.04	3.10±2.69	-	2.25±1.87	0.94±0.50
	17	7.82±4.88	7.01±4.49	-	3.01±1.99	0.91±0.39
	18	2.46±2.00	2.99±2.52	-	2.73±2.29	0.65±0.36
	19	1.73±1.28	1.67±1.23	-	1.57±1.15	0.36±0.04
	20	3.33±3.90	2.80±3.46	-	2.10±2.41	0.17±0.12
	21	3.41±2.43	1.88±1.32	-	3.44±2.45	0.57±0.19
	22	2.69±2.12	0.92±0.74	-	1.61±1.26	0.42±0.04
	23	0.51±0.59	0.49±0.57	-	0.75±0.88	0.05±0.01
Record 1-12 (T1)						
MAE± SDAE		2.34±2.47	1.28±2.61	1.02±1.79	1.02±1.25	0.52±0.19
Record 14-23 (T2 and T3)						
MAE± SDAE		3.19±3.61	3.05±3.35	-	2.95±3.71	0.44±0.18
Record 1-23 (T1, T2, T3)						
MAE± SDAE		-	-	-	1.97±2.48	0.48±0.19

Table 5.4: Activity Dependent Performance for Subject 1

Subject1: Activity (T1)	
Activity (km/h)	MAE±SDAE
1 - 2	0.92±0.38
6 - 8	4.14±3.35
12 - 15	0.76±0.22
6 - 8	0.62±0.37
12 - 15	1.08±0.55
1 - 2	0.72±0.36
Mean	**1.37±0.87**

5.2.6 Conclusion

In this section, a paradigm shift for performing HR estimation via a personalized data-driven approach using DNN on wearable PPG signals was described. *CorNET* moves away from using heuristics/thresholds, post-processing steps, auxiliary sensor data and extraction of hand crafted features in exchange for a fully automated method. The proposed CorNET topology, using two layers of CNN and LSTM is customized in conjunction with a dense layer for HR estimation through regression. It yields an average error of 0.48±0.19 BPM for HR on all 23 PPG recordings. The results could be considered favorable since to the best of knowledge, these represent the best accuracy in comparison to state-of-the-art methods on the given application area. The present exploration focusses on estimating average HR, predicting a new value every 2s using an 8s PPG frame, thereby missing out on instantaneous information. In section 5.3, the *BioTranslator* model will be introduced for investigation towards heart rate variability measures, which could provide insights into functioning of the sympathetic nervous system and help in disease prognosis (e.g. myocardial infarction).

5.3 BioTranslator: Inferring R-Peaks from Ambulatory Wrist-Worn PPG Signal

5.3.1 Introduction

Heart Rate Variability (HRV) serves as a biomarker of autonomic activity and is of significant clinical interest. Given a fixed average heart rate (HR), the time between individual heart beats may not be constant since it is actively modulated by the Autonomic Nervous system, which gives rise to HRV [92]. Instantaneous heart rate (IHR) is commonly derived from electrocardiography (ECG) traces by measuring the time between two consecutive R-peaks. The key enabler of this can be traces to the commercial development of wrist-worn sensing modalities – smart watches and fitness bands/trackers. Photoplethysmography (PPG) sensors have also gained prominence [93]. PPG is a low-cost, non-invasive, optical technique used to detect blood volume changes in the microvascular tissue bed, measured from skin surface. PPG signals are obtained from pulse oximeters, which emit light (using a light emitting diode) on the skin and measure (using a photodiode) the miniature variations in reflected or transmitted light intensity. The periodicity of the reflected/transmitted light corresponds to the cardiac rhythm, often used for HR estimation [94]. PPG presents a popular alternative to ECG, since it can be placed in various locations such as earlobes, fingertips or wrist making them convenient for ambulant usage. The wrist is considered a favorable position for unobtrusive daily usage; however the PPG data is vulnerable to motion artifacts (MA), which distorts signal fidelity, inhibiting robust estimation of vital parameters [95]. MA is generally caused due to a gap that develops between sensor and skin surface as well as change in blood flow due to

movements. This causes the heart-beat frequency spectrum to be suppressed by the MA spectrum, which has been mitigated by a number of research initiatives, mainly propagated by the 2015 IEEE Signal Processing Cup (SPC) [63]. Time-frequency domain signal processing and machine learning techniques have been used to attenuate MA and estimate HR from wrist-PPG data acquired during physical activity. Such techniques have often relied on motion reference signal, e.g. accelerometers for determining the MA spectral component and use it with adaptive filtering [76], Kalman filtering [77], wiener filtering [78], independent component analysis [79], empirical mode decomposition [80], spectral subtraction [87] and learning algorithms [96] to determine HR. However, majority of these techniques have successfully, computed average HR every two seconds (s) using an 8s sliding window of PPG data, which causes a loss in the instantaneous information. MA pose a challenge in extracting IHR measures during motion.

5.3.2 Prior Work in HRV Prediction

Studies performed in stationary conditions using time-invariant analysis showed that pulse rate variability (PRV) is a proxy of HRV. The difference between the two signals is the time taken by pumped blood (pulse) to travel from heart through arteries and capillaries to the measurement site. This is referred to as pulse transit time (PTT), varying with posture and blood pressure [97]. Two recent studies have successfully reported IHR estimation from wrist PPG, collected during motion. Operating on single channel PPG data, deppG employs an adaptive non-harmonic model with short-time Fourier transform (STFT) to obtain the spectrogram [98]. Using short-time cepstrum transform (STCT), a nonlinear data adaptive mask is designed, which is multiplied with the spectrogram to

obtain the required spectral information. Further, the IHR information is sharpened by applying a synchrosqueezing transform using the phase information of STFT. On the other hand, the study in [99] operates on two PPG channels and tri-axial accelerometer data to generate an initial estimate of average HR using a bank of six adaptive filters in conjunction with STFT. This is followed by a spectral masking technique to determine IHR using Empirical Mode Decomposition (EMD) on the PPG signals and estimates the instantaneous frequency and amplitude using the Hilbert transform. The average HR acts as the center point for time and spectral masking of instantaneous frequencies from the Hilbert spectrum. Both studies highlight the challenges involved in capturing the instantaneous spectral information due to the time-varying frequency/amplitude and non-sinusoidal oscillatory characteristics of PPG morphology. These promising results, pave the way for further exploration.

In the following sections, a supervised Deep Neural Network (DNN) framework, BioTranslator will be described. A model is trained to learn the underlying relationship between ECG signal and its corresponding PPG signal. When tested on new PPG test input, the trained model predicts a time series signal which is referred to as predicted ECG ($Pred_{ECG}$). The corresponding R-peaks from $Pred_{ECG}$ are extracted, using state-of-the-art Pan Tompkin's algorithm [100] and verified against ground-truth ECG ($True_{ECG}$). The framework was evaluated on 12 SPC subjects and identifies 92.8% of R-peaks, besides producing 51ms of mean absolute error when compared to ECG-derived IHR. Section 5.3.3 describes the problem formulation. The BioTranslator architecture has been detailed in

section 0. The results and analysis have been presented in section 0 and the conclusions have been drawn in section 0.

5.3.3 Problem Formulation

The idea of signal translation has been motivated by Google's Neural Machine Translation (GNMT) systems for automated language translation [101]. GNMT comprises deep long short-term memory (LSTM) networks with eight encoder and decoder layers for a sequence-to-sequence learning framework. Deriving inspiration from GNMT, an encoder-decoder framework is used for translating time series data. In this particular work emphasis is focused on PPG to ECG, but as noted in section 0 the framework can be applied to other time series signals. The translation makes it easier to capture the instantaneous information from the ECG signal with distinct R-peaks rather than MA corrupted PPG signal. It is realized using a convolution-deconvolution neural network (CNN-DCNN) topology, interleaved with pooling and unpooling layers respectively. Contrary to convolutional layers, which connect multiple input activations within a filter window to a single activation, deconvolutional layers associate a single input activation with multiple outputs. The output of a deconvolutional layer is an enlarged and dense activation map, allowing the extraction of richer information ($Pred_{ECG}$ with distinct R-peaks). The modelling of the inherent relationship between ECG and corresponding PPG signal rests on the fact that both are corollaries of cardiac activity. An overview of the supervised methodology enabling signal translation is shown in Figure 5.10.

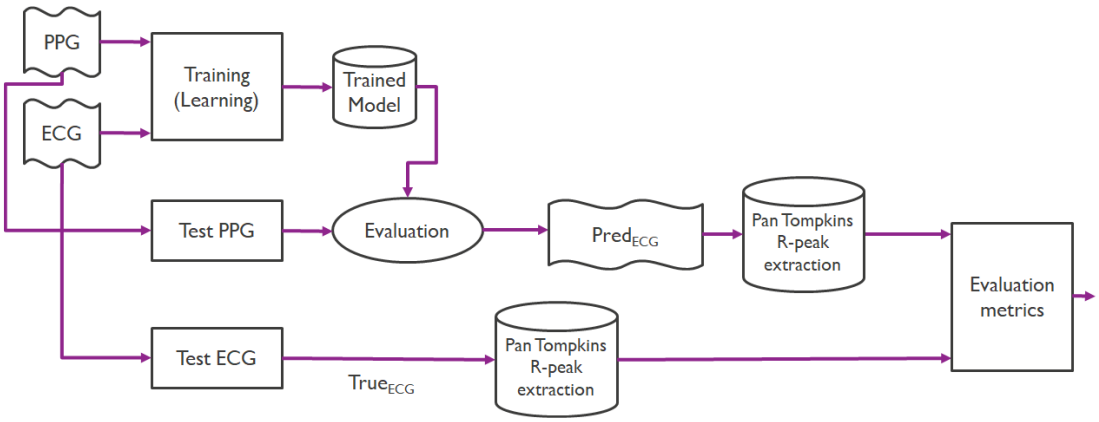


Figure 5.10: Overview of the novel signal transformation methodology for estimating R-peaks.

It is worth noting that there is no intention to verify the clinical fidelity of $Pred_{ECG}$ with respect to $True_{ECG}$. The primary focus at this time is to develop a mechanism which allows R-peak extraction from the $Pred_{ECG}$ morphology, enabling IHR estimation.

The framework is evaluated on 12 healthy subjects, age ranging 18 to 35 from the SPC database [63]. Each subject's data contains simultaneous recordings of - two channels of PPG from the wrist (dorsal) using a pulse oximeter with green LED (wavelength: 515 nm); tri-axial accelerometer signals from the wrist, and a channel of ECG from the chest using wet ECG electrodes. The ECG signal acts as the ground-truth for PPG-based IHR estimation. All signals were sampled at 125 Hz and transmitted to a computer using Bluetooth. The subjects were involved in walking/running on a treadmill with speeds in order: 1–2 km/h for 0.5 min, 6–8 km/h for 1 min, 12–15 km/h for 1 min, 6–8 km/h for 1 min, 12–15 km/h for 1 min, and 1–2 km/h for 0.5 min. The hierarchical structure of CNN-DCNN layers, capturing different levels of shape details is described in section 5.3.4.2.

5.3.4 BioTranslation Framework

5.3.4.1 Deep Neural Network Essentials

DNNs in general enable learning of task-adapted feature representations from the data [65] and have been successfully employed in a wide range of applications. Conventionally, CNNs are characterized by an initial layer of convolutional filters (set of weights which slides over the input), followed by a non-linearity (activation functions) and sub-sampling (pooling). These layers are stacked to achieve automatic feature extraction. Downstream layers capture more complex or differentiating features. Depth in the network integrates information from different filters to obtain various levels of abstraction, thereby reducing the size of activations. DCNN's consists of deconvolution operations combined with unpooling, which enlarges the activation maps. Pooling is designed to filter noisy activations by abstracting activations in a receptive field with a single representative value. Unpooling layers perform the reverse operation and reconstructs the original size of the activations. The output of an unpooling layer is an enlarged, yet sparse activation map. The deconvolution layers rectify sparse activations through convolution-like operations with multiple learned filters. Hence, a hierarchical structure of DCNN layers are used to capture different levels of shape details [102]. Deconvolution networks have been often used for semantic segmentation algorithms, where an input image is transformed through convolution into a lower-dimensional feature representation. Correspondingly, the deconvolution reconstructs the image from the features for class-specific object segmentation [103].

5.3.4.2 Experimental Setup Details

The PPG data samples are pre-processed with band-pass 4th order Butterworth filter having cut-off frequencies 0.1-16 Hz. This separates the high frequency noise component and DC drifts from the signal of interest. The filtered signal is further normalized to zero mean and unit variance and passed through the network. The BioTranslator topology is illustrated in Figure 5.11, comprising – (1) encoder-stage: two stacked CNN layers, interleaved with pooling layers; (2) decoder-stage: two DCNN layers interleaved with two unpooling layers. Each convolution operation in both stages comprises of a 1D CNN operation with Scaled Exponential Linear Unit (SELU) activation [69]. It uses 32 filters each of size 20 with the last layer having a single filter of size 30 to construct the ECG-like trace and match the target 125 sample length. Max-pooling and upsampling by 2 implements dimensionality scaling. Dropout is employed at a rate of 0.1, which ensures generalization, thereby preventing the model from overfitting to a given set of trained data samples. The input at the encoder has 250 samples, whereas the output of the decoder produces a time-series of 125 samples. The shorter output of the decoder incorporates the effect of PTT associated with PPG described in section 5.3.3. If the input and output were of the same length, some of the synthetic ECG signal would not have physical representation in the input PPG signal.

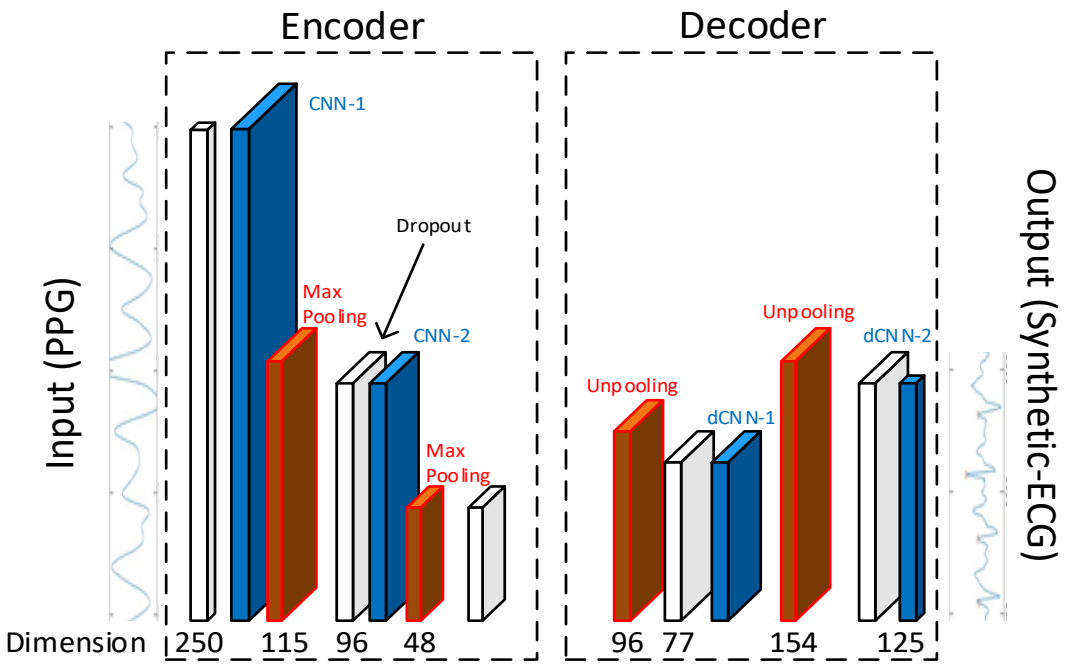


Figure 5.11: BioTranslator network topology

A data driven personalized approach has been adopted for the proposed framework, since biological signals are influenced by physiological factors – age, sex, height, weight/body-mass-index, etc. and importantly the cardiac condition of an individual. The *BioTranslator* framework is trained for each subject, and a five-fold cross-validation, with shuffling disabled, is performed to evaluate the trained model. Disabling shuffling ensures that any data leakage during training is limited only to the edges of the folds. The network was trained on an Nvidia Tesla K80 GPU and is modeled in Keras 2.0.4 version [67] configured to use theano [68] version 0.0.9 as backend. A batch size of 50 was set to balance training time and sensitivity to individual inputs. Hyperparameters such as filter length, number of filters, and number of layers were optimized using a heuristic grid search method.

5.3.5 Results and Analysis

Our approach towards IHR prediction goes against conventional methods, which typically return a real valued output over a shortened window. In this work, our goal was to translate the PPG to an equivalent ECG-like time series with distinct R-peaks. Since a rich time series is predicted, it is possible to use standard R peak detection algorithms such as Pan-Tompkins, enabling IHR computation. R-R intervals are minimally post-processed by comparing adjacent samples and any windows that differ by more than 150ms (heuristically determined) are replaced by the previous value. The method for generating the evaluation metrics is based on comparing the distance between the correctly predicted R peaks (obtained from $Pred_{ECG}$) and actual peaks ($True_{ECG}$). Dynamic time warping (DTW) is employed to align the two time-series of R indices, even with different lengths by replicating values to make the smallest Euclidean distance between each set of points [104]. These replicated values are useful for determining the false positives (FP) and false negatives (FN) related to prediction. FP occurs if the predicted R peak does not correspond to an actual peak. This manifests when the DTW algorithm inserts a duplicate TRUE index to align the false prediction. Similarly, FN occurs if there is no predicted R peak, corresponding to an actual R peak, which results in a duplicate predicted peak. Finally, true positive (TP) occurs when absolute distance between the two peaks is less than a threshold; set to be less than the minimum RR interval. For the SPC dataset the minimum RR is 42 samples (180 BPM at 125 Hz sampling frequency), hence a threshold of 30 samples is selected to restrict the likelihood of comparing two differing sets of R peaks.

The quality of the R-index predictions are quantified by the following metrics, which are summarized in Table 5.5: (i) accuracy – ratio of TP and total actual R peaks; (ii) sensitivity – accounts for the miss rate, computed as the ratio of TP to the sum of TP and FN; (iii) precision – false prediction rate, computed as the ratio of TP to the sum of TP and FP and (iv) mean absolute error (MAE) – mean of the absolute differences between true positive predicted R indices and actual R indices, in samples (n) and time (s), as in (5.2) and (5.3).

$$MAE[n] = \frac{1}{TP}\sum_{i=1}^{TP}|R_{TP}(i) - R_{ECG}(i)| \quad \textbf{5.2}$$

$$MAE[s] = \frac{MAE[n]}{F_S} \quad \textbf{5.3}$$

The performance of *BioTranslator* is in line with state-of-the-art methodology, achieving an average MAE of 51 ± 6.34ms on correctly predicted peaks. The algorithm deppG [98], achieves 50ms MAE when compared to reference IHR on the SPC dataset, but has a standard deviation of 16.2ms. This shows that our framework has a more consistent performance. The study [99] reports a strong performance of 28ms of IHR error, however they utilize 2s of PPG data to generate each prediction and employ a complex post-processing particle filter to ensure compliance to physiological limits such as spurious R peak removal. *BioTranslator* performance is evaluated in terms of standard time domain HRV metrics [105] – (i) SDNN: standard deviation of the normal-to-normal (NN) R-peak interval; (ii) meanNN: mean time duration of R-peak intervals and (iii) RMSSD: root mean square of successive RR intervals. Table 5.6: Time Domain Metrics illustrates these metrics with respect to the R-peak indices extracted from $Pred_{ECG}$, $True_{ECG}$ and manual

annotations ($Manual_{ECG}$) on $True_{ECG}$ (obtained from [99]). There is a close agreement between $Pred_{ECG}$, $True_{ECG}$ and $Manual_{ECG}$ for the measures SDNN and meanNN in comparison to RMSSD. The higher difference in RMSSD could be attributed to the limitations of Pan Tomkins based R-peak extraction on the backdrop of ambulant noise and artifacts. An illustration of the raw PPG, $True_{ECG}$ and $Pred_{ECG}$ traces is shown in Figure 5.12.

Table 5.5: R Peak Prediction Results on SPC Dataset

Subject	1	2	3	4	5	6	7	8	9	10	11	12	**Mean**
True R	669	655	676	650	653	630	641	637	645	692	689	677	659.5
TP	613	590	633	621	602	602	610	598	627	653	656	619	618.7
FP	48	64	11	29	25	45	32	13	29	14	20	38	30.7
FN	56	65	44	29	52	28	31	39	18	39	33	58	41
Accuracy(%)	91.6	90.1	93.6	95.5	92.2	95.6	95.2	93.9	97.2	94.4	95.2	91.4	93.8
Sensitivity(%)	91.6	90.1	93.5	95.5	92.1	95.6	95.2	95.2	97.2	94.4	95.2	91.4	93.8
Precision (%)	91.7	89	97.7	94.5	94.6	92.3	94.7	97.6	93.8	96.4	96.3	93.2	94.3
Avg. R error [n]	7.05	7.85	5.25	6.63	7.23	5.97	5.68	5.43	6.99	7.82	6.89	7.74	6.71
Avg. R error [ms]	53	59	40	51	54	45	45	43	51	59	53	60	**53**

Table 5.6: Time Domain Metrics

Sub.	**SDNN**			**MeanNN**		
	$Pred_{ECG}$	**$True_{ECG}$**	**Man_{ECG}**	**$Pred_{ECG}$**	**$True_{ECG}$**	**Man_{ECG}**
1	159.78	126.51	122.22	593.44	440.79	449.62
2	93.64	125.15	102.46	419.37	453.13	495.61
3	86.28	53.22	81.37	422.95	434.56	455.79
4	83.53	96.42	91.44	425.38	454.87	451.35
5	78.13	112.05	61.72	417.23	445.17	422.55
6	179.30	121.74	106.89	604.79	460.03	458.22
7	124.74	92.19	74.09	463.42	453.40	445.37
8	85.55	71.57	85.77	455.45	456.67	480.48
9	66	60.18	97.15	438.55	452.65	475.92
10	107.32	39.35	37.59	469.07	419.33	373.94
11	102.33	85.44	-	417.61	419.06	-
12	128.09	108.01	72.88	517.64	434.454	423.77
Median	**97.98**	**94.30**	**85.77**	**446.99**	**448.91**	**451.35**

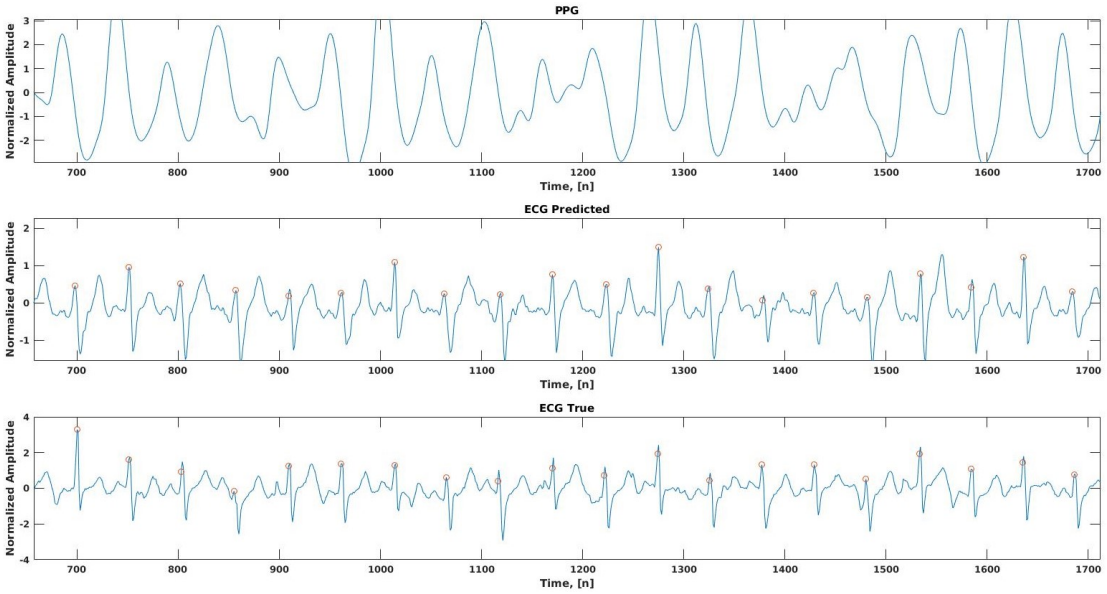


Figure 5.12: An illustration of PPG, predicted and true ECG, with the R-peaks (obtained through Pan Tompkins) marked over them.

Table 5.7: *BioTranslator* Complexity Analysis

Layer	Metadata	MACs	Trainable Parameters
CNN-1	*sF*=20, *nF*=32	147.2 k	672
CNN-2	*sF*=20, *nF*=32	1.95 M	20512
dCNN-1	*sF*=20, *nF*=32	156 M	20512
dCNN-2	*sF*=30, *nF*=1	119 k	961

Low-complexity processing on the sensor node is key to obtain energy efficient, long-term operability of battery-operated sensors [106]. This enables continuous/real-time monitoring for remote CVD applications. A complexity analysis of *BioTranslator*, shown in Table 5.7: *BioTranslator* Complexity Analysis, illustrates the number of multiply-and-accumulate (MAC) operations and trainable parameters for each convolution layer. Recent developments in CNN architecture [72] [5], make it achievable for an accelerator ASIC to perform online inference on real-time sensor data.

5.3.6 Conclusion

A first of its kind exploration is presented where a high-dimensional (information rich) signal is predicted from a low-dimensional signal. In this case, PPG signal represents an optically derived signal, affected by MA, collected in pervasive settings in comparison to ECG, which on the contrary represents a galvanic signal and has a higher SNR. In view of this work, future explorations would focus on evaluating the framework on additional signal pairs incorporating varying levels of MA and other applications. An energy efficient implementation of the network on an embedded platform (android for mobile platform or ASIC) would enable real-time information extraction. The *BioTranslator* framework could potentially be applied to other physiological pairs such as Electrocorticograph & Electroencephalogram [107].

Chapter 6. Conclusion

In Chapter 2, circuits for machine learning have were proposed. The kernel of the application, the dot product, is implemented in the time domain by modulating the delay of each unit as the multiplication, and the addition is realized intrinsically by connecting delay units serially. The first design presented used a digitally controlled oscillator, where the delay of each stage was realized by connecting different capacitor loads, proportional to the output of the multiplication, to the output of an inverter. The DCO is connected to a counter which is sampled and compared to other counters to determine the dominant neuron. Additionally, brain-inspired leakage is realized by resetting the least significant bits of the counter periodically. Local lateral inhibition, the phenomenon that makes boundaries of colors significant in our brains, is implemented by resetting neighboring counters after a threshold is reached. The second design foregoes the counter to increase energy efficiency by only requiring a single-shot evolution with a 2 bit time-to-digital converter. The multiplication is realized by modulating the number of inverter delays are enabled in each stage. An accuracy-efficiency scalable technique called Dynamic Threshold Error Correction is presented to combat ambiguity in the output by increasing the phase detector resolution. Both of these circuits demonstrate impressive energy efficiency and present novel solutions to address one of the most challenging applications gaining rapid adoption.

A new class of processors is described in Chapter 3; time-based graph computing application specific integrated circuits. Pulses are propagated through a graph network in the chip and distances between nodes are proportional to the delay of the pulse. In the 2D

chip, a voltage gradient is applied across the chip to shape the pulse which implements a variant of the popular A* algorithm. This enabled a more efficient search trajectory for the pulse. Additional applications including, collision avoidance for self-driving cars, multi-core synthesis, and scientific computing, were demonstrated which hopefully only scratches the surface of this new architecture. A 3D chip is also presented. The key innovation of this core is the 3D to 2D mapping which enables a scalable efficient design. The vertical axis is interleaved in both horizontal directions which causes the size to grow by the square root of the size, compared to previous works which increase linearly. This architecture solves 3D Voronoi diagrams, shortest path planning for autonomous flying drone, and machine learning classification. Solving such complex problems with simple circuits drives these to be some of the lowest power options available for solving the shortest path problem.

Next, a work that builds on the strong pedigree of the Beat Frequency circuit is presented for digitizing neural signals. The BFADC works by comparing the frequency derived from the extracellular potential of the neurons, to a frequency generated by a reference voltage which is tied to the recording subject to cancel out global noise. The BFADC provides a non-linear quantization which enables superior resolution with ultra-fast measurement periods. This chip was demonstrated on an anesthetized mouse to validate the changes to the front end circuits that were explicitly designed for neural recording.

Finally in Chapter 5, algorithm development using deep learning on biomedical signals was described. Using deep learning as the feature extraction engine, complex feature

extractors, heuristically set thresholds, and advanced signal processing techniques were obviated. This new philosophy of using a data-driven approach was new for the field. Three applications using photoplethysmography signals for biometric identification, average heart rate prediction, and heart rate variability estimation were presented. State of the art results were achieved in all three using deep learning architectures that will surely upend the conventional approach to signal processing.

Time-based computing is poised to make up gains lost by the slowing rate of process technology improvements. Time-based circuits use the delay it takes to return a result as the answer of the computation. Variable delay circuits can be constructed of simple digital building blocks and connected to counters or phase detectors to implement time-based architectures. These designs trade off being a general purpose processor for higher performance in tasks they are custom designed to solve; traditionally problems that are poorly served by conventional processors. The designs put forth in this thesis pave the way for future adoption of this exciting, efficient, and simple approach.

Bibliography

[1] L. R. Everson, S. S. Sapatnekar and C. H. Kim, "A 40x40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2-Dimensional Gradient Control," in *International Solid-State Circuits Conference*, San Francisco, CA, 2019.

[2] T. Simonite, "Moore's Law is Dead. Now What?," *MIT Technology Review,* pp. 1-2, 13 May 2016.

[3] J. P. Eckert, "A survey of Digital Computer Memory Systems," *Proceedings of the IRE,* vol. 41, no. 10, pp. 1393-1406, Oct. 1953.

[4] P.A. Merolla, et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science,* vol. 345, no. 6197, pp. 668-673, Aug. 2016.

[5] L. R. Everson, M. Liu, N. Pande and C. H. Kim, "A 104.8TOPS/W One-Shot Time-Based Neuromorphic Chip Employing Dynamic Threshold Error Correction in 65nm," in *2018 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Tainan, 2018, pp. 273-276.

[6] M. Liu, L. R. Everson and C. H. Kim, "A scalable time-based integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition capabilities," in *IEEE Custom Integrated Circuits Conference (CICC)*, Austin, TX, 2017, pp.1-4.

[7] B. Moons, R. Uytterhoeven, W. Dehaene and M. Verhelst, "14.5 Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSIO," in *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2017, pp. 146-147.

[8] H. Valavi, P. J. Ramadge, E. Nestler and N. Verma, "A Mixed-Signal Binarized Convolutional-Neural-Network Accelerator Integrating Dense Weight Storage and Multiplication for Reduced Data Movement," in *2018 IEEE Symposium on VLSI Circuits*, Honolulu, HI, 2018, pp. 141-142.

[9] Y. Chen, T. Krishna, J. Emer and V. Sze, "14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016, pp. 262-263.

[10] A. Biswas and A. P. Chandrakasan, "31.1 Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, San Francisco, CA, 2018, pp. 480-490.

[11] S. K. Gonugondla, M. Kang and N. Shanbhag, "31.2 A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, San Francisco, CA, 2018, pp. 490-492.

[12] C. Mead, "Neuromorphic Electronic Systems," *Proc. of the IEEE,* vol. 78, no. 10, pp. 1629-1636, Oct. 1990.

[13] B. V. Benjamin, et al., "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE,* vol. 102, no. 5, pp. 699-716, May 2014.

[14] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, 2011, pp. 1-4.

[15] L. Ni, et al., "An energy-efficienct digital reram-crossbar-based CNN with bitwise parallelism," *IEEE Journal on Exploratory solid-state computational devices and circuits,* vol. 3, pp. 37-46, Dec. 2017.

[16] M. Kim, et al, "A 68 parallel row access neuromorphic core with 22k multi-level synapses based on logic-compatible embedded flash memory," in *International Electron Devices Meeting (IEDM)*, San Francisco, CA, pp. 1-4, Dec. 2018.

[17] M. Z. Straayer and M. H. Perrott, "A multi-path gated ring oscillator TDC with first-order noise shaping," *IEEE Journal of solid-state circuits,* vol. 44, no. 4, pp. 1089-1099, April 2009.

[18] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transations on Neural Networks,* vol. 14, no. 6, pp. 1569-1572, Nov. 2003.

[19] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. pp. 2278-2324, Nov. 1998.

[20] K.J. Lee, et al., "14.2 A 502GOPS and 0.948mW Dual-Mode ADAS SoC with RNN-FIS Engine for Intention Prediction in Automotive Black-Box System," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016, pp. 256-257.

[21] P. Knag, C. Liu and Z. Zhang, "A 1.4mm2 141mW 898GOPS Sparse Neuromorphic Processor in 40nm CMOS," in *2016 IEEE Symposium on VLSI Circuits*, Honolulu, HI, 2016, pp. 1-2.

[22] S. Park, K. Bong, D. Shin, J. Lee, S. Choi and H. Yoo, "4.6 A1.93TOPS/W scalable deep learning/inference processor with tetra-parallel MIMD architecture for big-data applications," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, 2015, pp. 1-3.

[23] J. Lu, S. Young, I. Arel and J. Holleman, "30.10 A 1TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13µm CMOS," in *2014 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2014, pp. 504-505.

[24] S. Henzler, Time-to-Digital Converters, New York, New York: Springer, 2010.

[25] Y. Tsividis, Operation and Modeling of the MOS Transistor, Oxford: Oxford University Press, 1999.

[26] D. Miyashita, S. Kousai, T. Suzuki and J. Deguchi, "Time-domain neural network: A 48.5 TSOp/s/W neuromorphic chip optimized for deep learning and CMOS technology," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Toyama, 2016, pp. 25-28.

[27] S. Bang, et al., "14.7 A 288uW programmable deep-learning processor with 270kB on-chip wieght storage using non-uniform memory hierarchy for mobile intelligence," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2017, pp. 250-251.

[28] E. H. Lee and S. S. Wong, "24.2 A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016, pp. 418-419.

[29] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Transactions on Electronic Computers,* Vols. EC-10, no. 3, pp. 346-365, Sept. 1961.

[30] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[31] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the hueristic determination of minimum cost paths," *IEEE Trans. on systems science and cybernetics,* vol. 4, no. 2, pp. 100-107, 1968.

[32] S. Zhou, C. Chelmis and V. K. Prasanna, "High-Throughput and Energy-Efficient Graph Processing on FPGA," in *IEEE International Symp. on Field-Programmable Custom Computing Machines*, pp. 103-110, 2016.

[33] Y. Zhou and J. Zeng, "Massively Paralllel A* search on a GPU," in *Conference on Ariticial Intelligence (AAAI)*, pp.1248-1254, 2015.

[34] V. Honkote, et al., "2.4 A Distributed Autonomous and Collaborative Multi-Robot System Featuring a Low-Power Robot SoC in 22nm CMOS for Integrated Battery-Powered Minibots," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019, pp. 48-50.

[35] B. Wang, W. Luo and B. Nie, "Navigation Planning of Submarine in 3D Space Based on Space Contraction Ant Colony Algorithm," in *2018 17th International*

Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Wuxi, 2018, pp. 262-264.

[36] M. Kim and M. Lee, "UGV optimal path planning algorithm by using 3D survivability map," in *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, 2013, pp. 325-327.

[37] M. Yamaoka, et al., "24.3 20k-spin Ising chip for combinational optimization problem with CMOS annealing," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, 2015, pp. 1-3.

[38] V. Cherkassky and F. Mulier, Learning from Data: Concepts, Theory, and Methods, Wiley-IEEE Press, 2007.

[39] R. A. Fisher, "The use of multiple measurements in axonomic problems," *Annals of Eugenics,* vol. 7, pp. 179-188, 1936.

[40] I. T. Jolliffe, Principle Component Analysis, Springer, 2002.

[41] R. R. Harrison, "The Design of Integrated Circuits to Observe Brain Activity," *Proc. of the IEEE,* vol. 96, no. 7, pp. 1203-1216, July 2008.

[42] C.M. Lopez, et al., "A 966-electrode neural probe with 384 configurable channels in 0.13um SOI CMOS," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016, pp. 392-393.

[43] B.C. Raducau, et al., "Time multiplexed active neural probe with 678 parallel recording sites," in *European Solid-State Device Research Conference (ESSDERC)*, Lausanne, 2016, pp. 385-388.

[44] A. Rodriguez-Perez, M. Delgado-Restituto, F. Medeiro, "A 515 nW, 0-18dB programmable gain analog-to-digital converter for in-channel neural recording interfaces," *IEEE Trans. Briomed. Circuits Syst.,* vol. 8, no. 3, pp. 358-370, June 2014.

[45] S. Kundu, B. Kim and C. H. Kim, "Two-step beat frequency quantizer based ADC with adaptive reference control for low swing bio-potential signals," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, 2015, pp. 1-4.

[46] Intel Corp., "Technology Manufacturing Day - Strategy Overview," 28 Mar. 2017.

[47] B. Kim, S. Kundu, S. Ko and C. H. Kim, "A VCO-based ADC employing a multi-phase noise-shaping beat frequency quantizer for direct sampling of Sub-1mV input signals," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, San Jose, CA, 2014, pp. 1-4.

[48] S. Kundu, "Digital Intensive Mixed Signal Circuits with In-situ Performance Monitors," PhD Thesis, University of Minnesota, Minneapolis, MN, 2016.

[49] W. Kester, "ADC Input Noise: The Good, The Bad, and The Ugly. Is," Feb. 2006. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/adc-input-noise.html. [Accessed Jan. 2017].

[50] G. DeMichele and P. R. Troyk, "Stimulus-resistant neural recording amplifier," in *Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Cancun, Mexico, Sep. 2003, pp. 3329-3332.

[51] C. Briseno-Vidrios, A. Edward, N. Rashidi, J. Silva-Martinez, "A 4 Bit Continuous-Time$\\Sigma \Delta $Modulator With Fully Digital Quantization Noise Reduction Algorithm Employing a 7 Bit Quantizer," *IEEE Journal of Solid-State Circuits,* vol. 51, no. 6, pp. 1398-1409, June 2016.

[52] S. Tao and A. Rusu, "A Power-Efficient Continuous-Time Incremental Sigma-Delta ADC for Neural Recording Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 62, no. 6, pp. 1489-1498, June 2015.

[53] Y. Yoon, et al., "A 0.04-mm20.9-mW 71-dB SNDR distributed modular AS ADC with VCO-based integrator and digital DAC calibration," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, 2015, pp. 1-4.

[54] W. Jiang, V. Hokhikyan, H. Chandrakumar, V. Karkare, D. Marovic, "A 50-mV Linear-Input-Range VCO-Based Neural Recording Front-End with Digital Nonlinearity Correction," *IEEE Journal of Solid-State Circuits,* vol. 52, no. 1, pp. 173-184, Jan. 2017.

[55] J. Barnes, et al., "Abnormalities in the Climbing Fiber-Purkinje Cell Circuitry Contribute to Neuronal Dysfunction in ATXN1[82Q] Mice," *J. Neuroscience,* vol. 31, no. 36, pp. 12778-12789, Sept. 2011.

[56] R. Bonissi, et al., "A preliminary study on continuous authentication methods for photoplethysmographic biometrics," in *2013 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*, Naples, 2013, pp.28-33.

[57] A. Lourenco, H. Silva and A. Fred, "Unveiling the biometric potential of finger-based ecg signals," *Computational intelligence and neuroscience,* vol. 2011, pp. 1-8, 2011.

[58] Y. Gu, Y. Zhang and Y. T. Zhang, "A novel biometric approach in human verification by photoplethysmographic signals," in *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, Brimingham, UK, 2003, pp. 13-14.

[59] Y. Y. Gu and Y. T. Zhang, "Photoplethysmographic authentication through fuzzy logic," in *IEEE EMBS Asian-Pacific Conference on Biomedical Engineering*, Kyoto, Japan, 2003, pp. 136-137.

[60] P. Spachos, J. Gao and D. Hatzinakos, "Feasibility study of photoplethysmographic signals for biometric identification," in *2011 17th International Conference on Digital Signal Processing (DSP)*, Corfu, 2011, pp. 1-5.

[61] A. Kavsaoglu, K. Polat and M. Bozkurt, "A novel feature ranking algorithm for biometric recognition with PPG signals," *Computers in Biology and Medicine,* vol. 49, pp. 1-14, 2014.

[62] V. Jindal, et al., "An adaptive deep learning approach for PPG-based identification," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 2016, pp. 6401-6404.

[63] Z. Zhang, et al., "TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on Biomedical Engineering,* vol. 62, no. 2, pp. 522-531, 2015.

[64] J. Lalor, H. Wu and Y. H, "Improving Machine Learning Ability with Fine-Tuning," 2017.

[65] Y. Bengio, "Learning deep architectures for AI," in *Foundations and trends® in Machine Learning 2.1*, pp. 1-127, 2009.

[66] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[67] "https://keras.io," [Online].

[68] "https://deeplearning.net/software/theano," [Online].

[69] G. Klambauer, et al., "Self-normalizing neural networks," in *arXiv preprint:1706.02515*, 2017.

[70] "http://climin.readthedocs.io/en/latest/rmsprop.html," [Online].

[71] S. Han, H. Mao and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *arXiv preprint arXiv:1510.00149*, 2015.

[72] M. Panwar, et al., "Modified Distributed Arithmetic Based Low Complexity CNN Architecture Design Methodology," in *2017 European Conference on Circuit Theory and Design (ECCTD)*, Catania, 2017, pp. 1-4.

[73] E. B. Mazomenos, et al., "A Low-Complexity ECG Feature Extraction Algorithm for Mobile Healthcare Applications," *IEEE Journal of Biomed. and Health Inf.,* vol. 17, no. 2, pp. 459-469, March 2013.

[74] H. Kim, et al., "A configurable and low-power mixed signal SoC for portable ECG monitoring applications," *IEEE Transations on Biomed. Circuits and Systems,* vol. 8, no. 2, pp. 257-267, April 2014.

[75] R. Yousefi, et al., "A motion-tolerant adaptive algorithm for wearable photoplethysmographic biosensors," *IEEE Journal of Biomed. and Health Inf.,* vol. 18, pp. 670-681, 2014.

[76] M. Mashhadi, et al., "Heart Rate Tracking using Wrist-Type Photoplethysmographic (PPG) Signals during Physical Exercise with Simultaneous Accelerometry," *IEEE Signal Processing Letters,* vol. 23, no. 2, pp. 227-231, Feb. 2016.

[77] B. Lee, et al., "Improved elimination of motion artifacts from a photoplethysmographic signal using a kalman smoother with simultaneous accelerometry," *Phys Meas,* vol. 31, no. 12, pp. 1585-1603, 2010.

[78] A. Temko, et al., "Accurate Heart Rate Monitoring During Physical Exercises Using PPG," *IEEE Transactions on Biomedical Engineering,* vol. 64, no. 9, pp. 2016-2024, 2017.

[79] B. Kim and S. Yoo, "Motion artifact reduction in photoplethysmography using independent component analysis," *IEEE Transactions on Biomedical Engineering,* vol. 53, no. 3, pp. 566-568, March 2006.

[80] E. Khan, et al., "A Robust Heart Rate Monitoring Scheme Using Photoplethysmographic Signals Corrupted by Intense Motion Artifacts," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 3, pp. 550-562, March 2016.

[81] S. Salehizadeh, et al., "Novel Time-Varying Spectral Filtering Algorithm for Reconstruction of Motion Artifact Corrupted Heart Rate Signals During Intense Physical Activities Using a Wearable Photoplethysmogram Sensor," *Sensors,* vol. 16, no. 1, 2016.

[82] B. Sun and Z. Zhang, "Photoplethysmography-based heart rate monitoring using asymmetric least squares spectrum subtraction and bayesian decision theory," *IEEE Sensors,* vol. 15, pp. 7161-7168, 2018.

[83] E. Grisan, et al., "A supervised learning approach for the robust detection of heart beat in plethysmographic data," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, Milan, 2015, pp. 5825-5828.

[84] M. Essalat, M. B. Mashhadi and F. Marvasti, "Supervised heart rate tracking using wrist-type photoplethysmographic (PPG) signals during physical exercise without simultaneous acceleration signals," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washingoton, DC, 2016, pp. 1166-1170.

[85] Y. Meada, et al., "The advantage of green reflected photoplethysmograph," *Journal of Medical Systems,* vol. 35, pp. 829-834, 2011.

[86] Y. Sun and N. Thakor, "Photoplethysmography revisited: from contact to noncontact, from point to imaging," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 3, pp. 463-477, 2016.

[87] Z. Zhang, et al., "Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction," *IEEE Transactions on Biomedical Engineering,* vol. 62, no. 8, pp. 1902-1910, 2015.

[88] D. Zhao, et al., "Sfst: A robust framework for heart rate monitoring from photoplethysmography signals during physical activities," *Biomedical Signal Processing and Control,* vol. 33, pp. 316-342, 2017.

[89] T. Schack, et al., "A new method for heart rate monitoring during physical exercise using photoplethysmographic signals," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*, 2015, pp. 2666-2670.

[90] M. Raghuram, et al., "Evaluation of wavelets for reduction of motion artifacts in photoplethysmographic signals," in *Proc. 10th Int. Conf. Inform. Sci. Signal Process. Appl.*, 2010, pp. 460-463.

[91] H. Chung, H. Lee and J. Lee, "Finite State Machine Framework for Instantaneous Heart Rate Validation using Wearable Photoplethysmography During Intensive Exercise," *IEEE Journal of Biomedical and Health Informatics,* pp. 1-1, 2018.

[92] A. J. Camm, et al., "Heart rate variability. Standards of measurement, physiological interpretation, and clinical use," *European heart journal,* vol. 17, no. 3, pp. 354-381, 1996.

[93] S. Kirk, "Comparison of the Apple Watch, Fitbit Surge, and Actigraph GT9X Link in Measuring Energy Expenditure, Steps, Distance, and Heart Rate (Thesis," Cleveland State University, Cleveland, OH, 2016.

[94] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Phys. Meas.,* vol. 28, pp. 1-39, 2007.

[95] T. Aoyagi and K. Miyasaka, "Pulse oximetry: its invention, contribution to medicine, and future tasks," *Anesthesia and analgesia,* vol. 94, no. 1, p. S1, 2002.

[96] D. Biswas, et al., "CorNET: Deep Learning framework for PPG based Heart Rate Estimation and Biometric Identification in Ambulant Environment," *IEEE Transactions on Biomedical circuits and systems,* pp. 1-1, 2019.

[97] E. Gil, et al., "Photoplethysmography pulse rate variability as a surrogate measurement of heart rate variability during non-stationary conditions," *Physiological Measurements,* vol. 31, no. 9, pp. 1271-1290, 2010.

[98] Cicone Antonio, Wu Hau-Tieng, "How Nonlinear-Type Time-Frequency Analysis Can Help in Sensing Instantaneous Heart Rate and Instantaneous Respiratory Rate from Photoplethysmography in a Reliable Way," *Frontiers in Physiology,* vol. 8, p. 701, 2017.

[99] D. Jarchi, et al., "Towards photoplethysmography-based estimation of instantaneous heart rate during physical activity," *IEEE Transaction on Biomedical Engineering,* vol. 64, no. 9, pp. 2042-2053, 2017.

[100] P. Jiapu and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering,* vol. 32, no. 3, pp. 230-236, 1985.

[101] Y. Wu, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," in *arXiv preprint arXiv:1609.08144*, 2016.

[102] Z. Matthew, et al., "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 International Conference on Computer Vision*, Barcelona, 2011, pp. 2018-2025.

[103] S. Noh, S. Hong and B. Han, "Learning deconvolution network for semantic segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1520-1528.

[104] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *AAAI-94 workshop on knowledge discovery in databases*, 1994, pp. 229-248.

[105] F. Shaffer and J. P. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers in public health,* vol. 5, p. 258, 2017.

[106] D. Biswas, et al., "Low-Complexity Framework for Movement Classification Using Body-Worn Sensors," *IEEE Transactions on Very Large Scale Integrated Systems,* vol. 25, no. 4, pp. 1537-1548, 2017.

[107] L. Everson, et al., "BioTranslator: Inferring R-Peaks from Ambulatory Wrist-Worn PPG Signal," in *Engineering in Medicine and Biology Conference*, Berlin, pp. 1-5, 2019.

[108] L. Everson, et al., "BiometricNet: Deep Learning based biometric identification using wrist-worn PPG," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, 2018, pp. 1-5.