Low Energy VLSI Architecture for Interfacing Brain: Measurement and Stimulation

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2018

Tomoki SUGIURA

Publications

Journal Article (Refereed)

[J1] Tomoki Sugiura, Masaharu Imai, Jaehoon Yu, and Yoshinori Takeuchi: "A Low-Energy Application Specific Instruction-Set Processor towards a Low-Computational Lossless Compression Method for Stimuli Position Data of Artificial Vision Systems," Journal of Information Processing, vol.25, pp.210–219, 2017.

International Conference Papers (Refereed)

[I1] Tomoki Sugiura, Arif Ullah Khan, Yoshinori Takeuchi, and Masaharu Imai, "Alpha-Gamma Data Compression Method for Artificial Vision Systems using Visual Cortex Stimulation", in Proc. of 18th Workshop on Synthesis And System Integration of Mixed Information Technologies, pp.56–61, Sapporo, Japan, Oct. 21–22, 2013.

[I2] Tomoki Sugiura, Shoko Nakatsuka, Jaehoon Yu, Yoshinori Takeuchi, and Masaharu Imai, "An Efficient Data Compression Method for Artificial Vision Systems and its Low Energy Implementation using ASIP Technology," in Proc. of 10th IEEE Biomedical Circuits and Systems Conference, pp.81–84, Lausanne, Switzerland, Oct. 22–24, 2014.

[I3] Tomoki Sugiura, Jaehoon Yu, Yoshinori Takeuchi, and Masaharu Imai, "A Low-Energy ASIP with Flexible Exponential Golomb Codec for Lossless Data Compression toward Artificial Vision System", in Proc. of 11th IEEE Biomedical Circuits and Systems Conference, pp.97–100, Atlanta, USA, Oct. 22–24, 2015.

[I4] Tomoki Sugiura, Arif Ullah Khan, Jaehoon Yu, Yoshinori Takeuchi, Seiji Kameda, Takatsugu Kamata, Yuki Hayashida, Tetsuya Yagi, and Masaharu Imai: "A Programmable Controller for Spatio-temporal Pattern Stimulation

of Cortical Visual Prosthesis," in Proc. of 12th IEEE Biomedical Circuits and Systems Conference, pp.432–435, Shanghai, China, Oct. 17–19, 2016.

[I5] Tomoki Sugiura, Jaehoon, Yu, and Yoshinori Takeuchi: "Hardware-Oriented Algorithm for Phase Synchronization Analysis of Biomedical Signals," in Proc. of 13th IEEE Biomedical Circuits and Systems Conference, pp.97–100, Turin, Italy, Oct. 19–21, 2017.

International Conference

[C1] Tomoki Sugiura, "A Low-Energy ASIP for Lossless Data Compression toward Cortical Vision Prosthesis," The 6th MEI3 Center International Symposium, Osaka, Japan, Jan. 18, 2016.

[C2] Tomoki Sugiura, Jaehoon Yu, Yoshinori Takeuchi, Yuki Hayashida, Tetsuya Yagi, and Masaharu Imai, "A Low Energy Implementation of Neural Stimulation Microprocessor for Cortical Visual Prosthesis," IEEE BrainCAS2016, Hangzhou, China, 20–21, 2016.

[C3] Tomoki Sugiura, "An Energy Reduction Design of Neural Stimulation Controller for Cortical Visual Prosthesis," The 8th MEI3 Center International Symposium, Osaka, Japan, Mar. 1–2, 2017.

Preface

Owe to miniaturization of integrated circuits, implantable medical devices can be more intelligent than ever. Their users' demands for hardware architecture are significantly severe because their malfunctions directly damage the users' vital and negative effects caused by their daily use degrade the users' quality of life. Like cochlear implants, visual prostheses, and deep brain stimulations, new medical treatments to cure or adverse diseases by stimulating target nerves have emerged, which the conventional methods such as surgery and pharmacological approach cannot cope with. Energy efficiency is one of the most concerning topics not only for hardware development by medical engineers but also for the clinical use of neural stimulation devices by medical experts. Low energy hardware architecture of the neural stimulation devices can increase their longevity, which mitigates the user's physical and mental burden caused by the replacement of its secondary battery. Also, regarding the neural acquisition, the number of electrodes should be increased more and more to improve the precision of diagnosis and prediction of neural diseases for neural stimulation. Therefore, the brain interface for medical use is required to cope with both the energy efficiency and performance.

This thesis presents a VLSI architecture for low energy brain interface for neural signal measurement and stimulation. First, high throughput and low hardware cost biomedical signal processor for phase synchrony analysis is described. Next, a low computational data compression method for neural stimuli data and its hardware implementation are proposed. Then, a system-on-a-chip (SoC) for neural stimulation control with high temporal resolution and high flexible stimuli configuration is developed.

Adaptive stimulation devices, also known as closed-loop stimulators, can increase the efficiency of treatment and reduce burdens on both the users and the healthcare workers. The closed-loop stimulators of the deep brain stimulations observe biomarkers of the target organ and decide to stimulate or not, and then the closed-loop systems reduce the energy consumption of whole of the stimulator. Speaking about the state of the art of the closed-loop stimulators, their signal processing stage requires bulky external computers to deal with the significant amount of recorded data, or they face low processing throughput due to limited computation resource inside the body. In chapter 3, this thesis proposes a high

efficient phase synchrony calculation method for measuring biomedical signals. The mean phase coherence is one of the well-known indicators for phase synchronization of the two signals, and its computation includes trigonometric functions that require a large computational cost in hardware implementation. The proposed method uses only linear algebra instead of trigonometric functions and keeps the computational precision high thanks to equation expansions without any approximation. The proposed method can reduce computational amounts by reusing calculation results. This thesis also proposes a hardware implementation of the proposed signal processing method. From the evaluation, the proposed implementation increased throughput by 2.6 times and reduced hardware cost, which is defined by "throughput per gate count," by 5.3 times. (*Related publication: [1]*)

Energy consumption in a wireless communication component, which is roughly proportional to the amount of data transmission in communication, is dominant of the whole energy consumption of the neural stimulation devices. Especially in visual prostheses, the internal stimulator has to update the information about stimuli position of electrodes frequently to synchronize the visual perception to the surrounding condition in real time. The data transmission includes information to control stimulation such as amplitude, duration, frequency, and position of stimulation, which are generated by an external body component from the surrounding information of the user. The data of stimuli position updates more frequently than the other information, and hence this thesis proposes a data compression method for the stimuli position data in chapter 4 to reduce the amount of data transmission in wireless communication. The proposed method is lossless compression because the neural stimulation devices have to guarantee high precision and accuracy for stimulation. This thesis directs the proposed method to a low computational algorithm, which can suppress extra energy consumed by compression. The stimuli position data is analyzed with a statistical approach, and the proposed method takes advantage of spatial bias in the distribution of bitstreams and temporal redundancy in the stimuli position data to compress them effectively. Also, this thesis proposes a hardware implementation of the proposed compression method as an application specific instruction-set processor (ASIP). Evaluation results show that the proposed compression method reduced the stimuli position data by 77 %, and the proposed implementation reduced energy consumption by 62 % and 75 % in a decompression stage for one frame and in the total of wireless communication with the Bluetooth low energy standard, respectively. (*Related publication: [2]*)

For higher precision for neural prostheses, the neural stimulation devices require the flexibility of stimulation strategies, high temporal resolution, and low energy consumption. An embedded processor enables the neural stimulation devices to change their stimulation manner easily by rewriting the program. However, the embedded processor has to manage many tasks for stimulation control and cannot process a loaded program punctually because of external interrupts from peripherals. In chapter 5, this thesis proposes a highly flexible neural stimulation controller with the high temporal resolution. The controller includes the

ASIP as a microcontroller and a dedicated circuit to control stimulation, which is connected to a signal converter to generate stimuli current discharges. Also, the stimulation device that contains the proposed SoC and the signal converter is presented. Evaluation results with a field-programmable gate array (FPGA) show that the proposed stimulation controller manipulates stimulations in microseconds order and changes stimuli parameters with rewriting the program in external flash memory. The demonstrated stimulation device has a comparable performance with the state-of-the-art architectures regarding the number of stimuli channels and amplitude resolution and better flexibility of stimuli frequency and duration. (*Related publication: [3]*)

This research contributes to the reduction of energy consumption for the implantable neural stimulators, which leads to miniaturization and longevity extension of the devices. Continuous progress in electrophysiology, neuroscience, and biomedical engineering will support to reveal methods of treatment for incurable diseases.

Acknowledgments

The author would like to express his sincere gratitude and appreciation of all persons and institutions who have supported and encouraged the research that is presented, and writing of this thesis.

Heartfelt appreciation is expressed to Prof. Masanori Hashimoto, Osaka University, for invaluable scientific guidance. The reviewers of this thesis are sincerely thanked for investing their time into proofreading and evaluation of the scientific content, Prof. Takao Onoye, Osaka University, Prof. Tetsuya Yagi, Osaka University, Professor Emeritus Masaharu Imai, Osaka University, and Associate Professor Yoshinori Takeuchi, Osaka University.

The author is grateful to the Specially Appointed Associate Professor Hirotsugu Okuno, Osaka Institute of Technology, for the use of Silicon Retina Camera and Insight© for experimentations in chapter 4. Also, the author is grateful to the Specially Appointed Associate Professor Seiji Kameda, Osaka University, for his permission granted to use some of his material.

The support and discussion gratefully encouraged the author with Associated Professor Yuki Hayashida, Assistant Professor Naofumi Suematsu, and members of neural interface group of Biosystems and Devices Area, Osaka University.

The author gratefully acknowledges the valuable daily advice by Assistant Professor Jaehoon Yu and the support of all members of Integrated System Design Laboratory at Osaka University. Daily discussion with doctoral candidates in the laboratory stimulated and encouraged the author.

Finally, the author would like to deeply thank all my friends and family.

Contents

List of Figures

List of Tables

Abbreviations

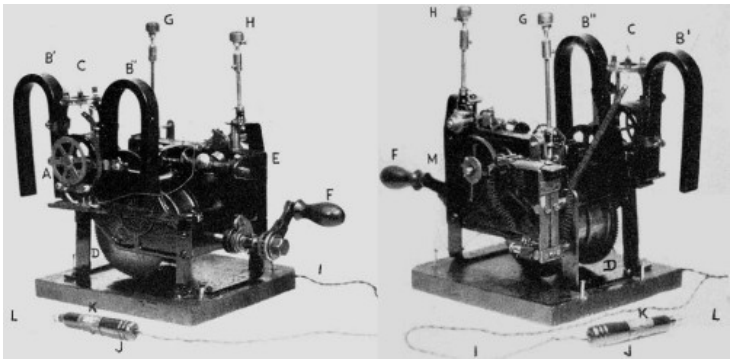
α-EGC	α exponential Golomb coding
AHC	adaptive Huffman coding
ASIP	application specific instruction-set processor
BAN	Body area network
BLE	Bluetooth Low Energy
CI	cochlear implant
CORDIC	coordinate rotation digital computer
CR	compression ratio
DBS	deep brain stimulation
DDA	discrete distance approximation
ECoG	electrocorticography
EEG	electroencephalography
EGC	exponential Golomb coding
FIR	finite impulse response
FPGA	field-programmable gate array
GPR	general-purpose register
IMD	implantable medical device
MPC	mean phase coherence
OCP	on-chip programmer
QoL	quality of life
RISC	reduced instruction set computer
RMSE	root mean square error
RSQRT	reciprocal of square root
RTL	register-transfer level
SHC	static Huffman coding
SoC	system-on-a-chip
SPI	serial-parallel interface
SQRT	square root
UART	universal asynchronous receiver/transmitter
UWB	ultra wide band

Chapter 1

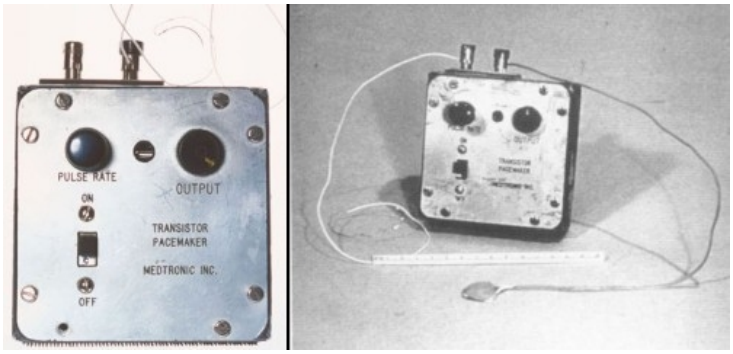
Introduction

The art of medicine has a long history from ancient Egypt. The primitive medicine was developed based on experiments transmitted via oral instruction and the domain of witch doctors and medicine men. This ancient medicine has a magical aspect, and the ancient medical staff regarded diseases and pains as the invasion by devils. Then, the medical knowledge had been gathered and compiled gradually and again by many people like Hippocrates in ancient Greek. However, although the medical staff had been admired as experts from a long time ago, it is said that the progress of medicine began in sixteen century because medical institute had been founded. At the beginning of twenty century, the rapid progress of science like chemistry, physiology, pharmacology, and so on had involved medicine significantly. The progress of science has produced many kinds of interdisciplinary, and biomedical engineering has become one of the most attractive research fields. According to [4], the term *biomedical engineering* means a discipline for applying electrical, chemical, optical, mechanical, and other engineering principles to understand, modify, or control biological (i.e., human and animal) systems.

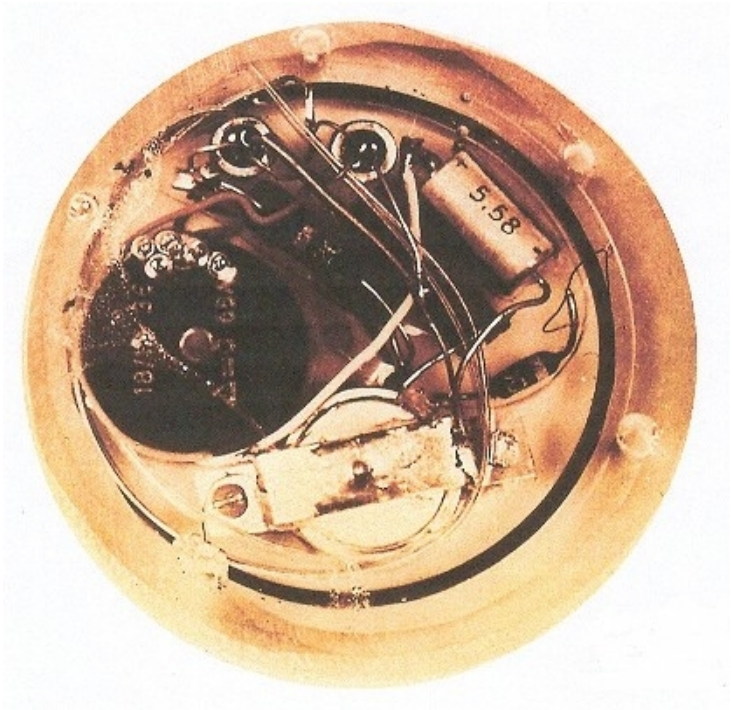
In biomedical engineering, the electricity has a close implication for medicine, but methodologies on how to observe them were established in early twenty century. The early roots of electrophysiology was a remarkable discovery by Italian scientist Galvani in the mid-seventeen century. He discovered that the living tissue of frog muscles exhibits electrical potential. After Galvani, the first medical use of bioelectrical signals is related to the heart. In 1903, William Einthoven invented the first electrocardiograph, which is a start of cardiovascular medicine and biomedical signal measurement of the human [5]. Two physicians, Lidwell and Hyman, were devoted to the development of the first pacemaker separately, and, in 1932, Hyman demonstrated the "artificial pacemaker" illustrated in Fig. 1.1 (a), which was packaged in a bulky box. At the end of the 1950s, the two notable inventions of the pacemaker emerged; the first battery-operated and wearable pacemaker in 1957 and the first implantable pacemaker in 1958. American electrical engineer Bakken produced the first battery-operated wearable pacemaker, which got rid of a drawback of the existing mains-powered pacemaker regarding the power supply. Figure 1.1 (b) depicts the first battery-operated wearable



(a) Hyman's artificial pacemaker (Courtesy of [5]).



(b) Bakken's pacemaker with leads (Courtesy of [5]).



(c) Elmqvist's implantable pacemaker (Courtesy of [5]).

Figure 1.1: Progress of pacemakers.

pacemaker, which includes two transistors, a 9.4 V mercury battery, and an on-off switch. His pacemaker, which was improved and released later, emitted 2 ms square wave, variable in amplitude from 1 to 20 mA into 1000 Ω lead. On the other hand, in 1958 in Sweden, the two experts the surgeon Senning and the inventor Elmqvist collaborated for implantation for the first time. The first implantable pacemaker illustrated in Fig. 1.1 (c) was encapsulated in epoxy resin, its diameter and thickness were 55 mm and 16 mm, respectively, and it contained two silicon transistors. After the first surgery, the pacemaker was functional for a few hours.

These events happened at the dawn of the life science domain, and the latest pacemaker 'Micra' produced by Medtronic Inc. has twelve-year battery longevity, much shorter leads, and 0.8 cm^3 volume [6] thanks to the outstanding progress of the biomedical engineering. This incredible progress has been mainly owing to assiduous researches of electronics, especially integrated circuits. In 1965, Moore released his observation – the complexity of minimum component costs has increased at a rate of roughly a factor of two per year [7] and then adjusted the pace to a double every two years in 1975 [8]. This phenomenon rather than observation results has been called "Moore's law," and the semiconductor industry roughly caught up within the second half of twenty century. Though his law was taking a downturn at the beginning of the twenty-first century, the computational power of integrated circuits have increased more and more, and medicine has benefited significantly from this inflation. Improvement of the integrated circuits brings miniaturization of computers. For example, the latest pacemakers allow their users to diagnose the users' condition continuously and tell a summary of the result to medical staffs via wireless data transmission. Consequently, an attending physician of the users only has to examine them once or twice a year, and the user only has to replace their pacemaker once in ten to twelve years, which is provided by the improvement of miniaturization and integration of the integrated circuits.

Thanks to downsizing and low energy operation of integrated circuits, people can set computers on their skins or even inside bodies as implantable medical devices (IMDs) to keep their lives. The aim of the IMDs is for their user to rebuild body functions and improve longevity. For achieving a better quality of life (QoL), demands of the IMDs have been increasing, which are summarized as follows: Long-term use, minimally inversive, and high reliability. When the IMDs need to be repaired, emergent surgery must be required to exchange the IMD, which forces the users to degrade their QoL. Therefore, the users demand that the IMDs work as long as possible and tell the results from self-diagnosis functions. The IMDs also should be minimally invasive concerning their heat dissipation. They often lie on target organs that are in charge of the users' physical functions. The organs may be damaged if the heat dissipation from the IMDs is excessive. According to work of Kim [9], the power dissipation of the system should be lower than 35 mW to keep temperature increase less than 1 °C. Analysis of heat dissipation from the IMDs [10, 11] and development of neural probes including thermal sensors are still hot topics [12, 13].

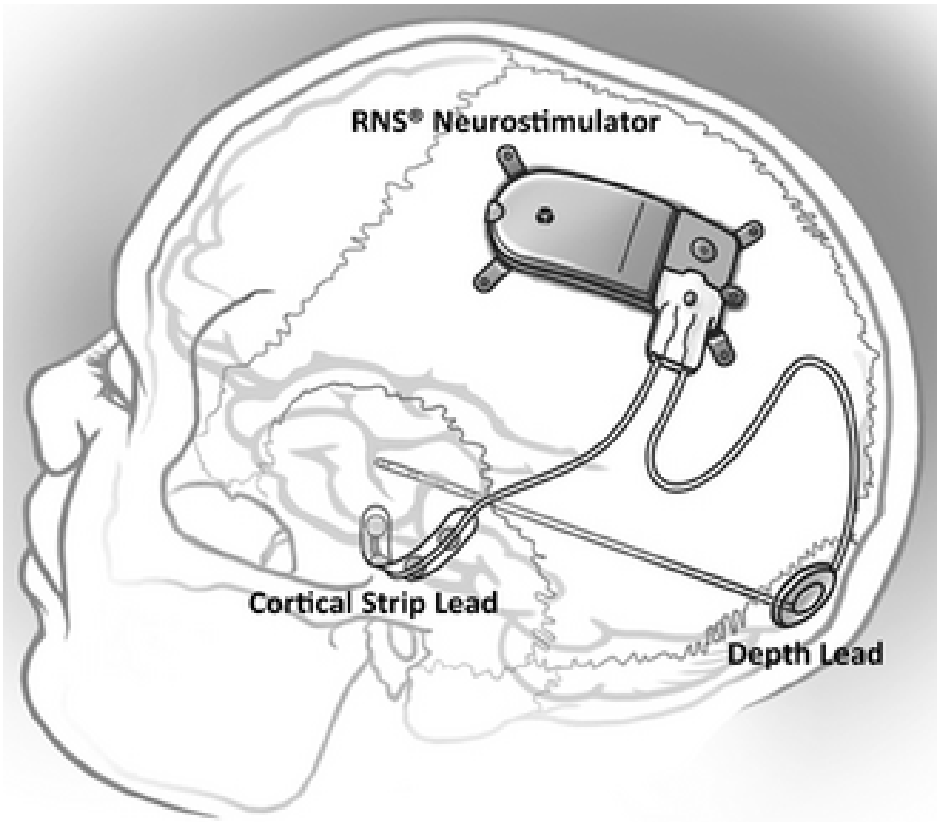


Figure 1.2: Example of neural stimulation device: RNS® Neurostimulator, NeuroPace depth lead, and NeuroPace cortical strip lead (Courtesy of [14]).

The progress of the electronics and increase in computational power of the integrated circuits enable to observe and analyze the more complicated nervous system in the body than the heartbeat, which is the central nervous system. The research field of neural prostheses is one of the most prominent application targets of the IMDs. Neural prosthesis requires more severe restrictions because errors

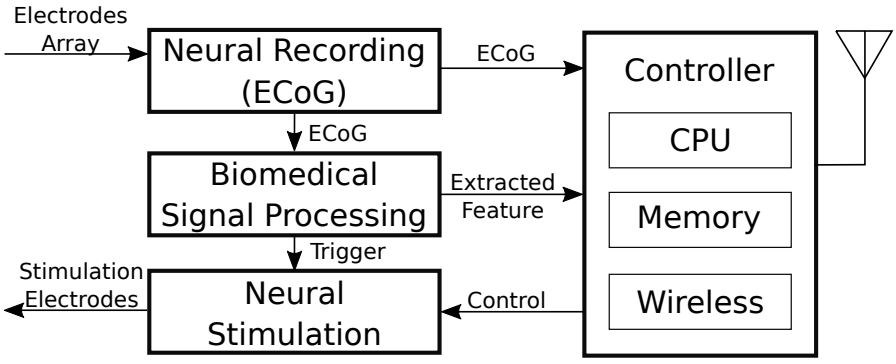


Figure 1.3: Illustration of the closed-loop neural stimulation device.

caused by IMDs affect critical damage to their users. Figure 1.2 depicts an example of neural stimulation devices, and Fig. 1.3 describes a block diagram of the system in Fig. 1.2. This device is called deep brain stimulation (DBS), which aims to stop epilepsy seizure by stimulating the part of the brain of the seizure focus of a recipient by a penetrating lead, which is called depth lead in Fig. 1.2. To detect the abnormal activity of the brain, this device continuously records and analyze the user's neural condition by measuring electrocorticography (ECoG) with electrodes, called cortical strip lead in Fig. 1.2. Energy consumption of the neural prosthesis devices has a key factor to satisfy these requires. There is little space for the neural stimulation device in the brain, and the volume of a battery often occupies the whole size of the system dominantly. Also, the surgery for replacement compromises the recipients' QoL crucially because it forces physical restraint in a long day and extra risks to failure. Low energy consumption architecture of the IMDs enables to extend their operating time and reduce demanded battery capacity, which leads the miniaturization of the whole volume of the IMDs.

The aim of this research is to achieve energy reduction by improving the hardware architecture of neural prosthesis device. Generally speaking, internal body components have more severe restrictions than external body components, and hence this research focuses on energy reduction of internal body components. This thesis mainly discusses energy consumption of wireless communication unit and signal processing unit, which are dominant energy consumers in internal body components. Figure 1.4 shows contributions this research has. This figure shows the entire structure of neural stimulation device, and this thesis focuses on the architecture of an internal body component. The internal body component consists of a controller, a neural stimulator, a neural signal recorder, D/A and A/D converters, and neural interface such as electrodes. For the reduction of energy consumption, this thesis investigates the trade-off between energy efficiency and performance in stimulation control and biomedical signal processing. This thesis proposes architectures for a low-energy and highly flexible neural stimulation controller and high throughput and hardware-efficient neural signal processor. These proposals contribute to the improvement of the neural analysis in precision and resolution and the energy reduction in the internal body component. Also, the proposed neural stimulation device enables high-frequency and high-complex stimulation, which will lead to the practical neural prostheses in daily life driven by high-spatiotemporal stimulus control, especially visual prosthesis devices.

In this thesis, a neural stimulation device including a microprocessor is demonstrated. The advantage of the proposed architecture is to be able to calculate the values *in situ* and to change its behavior easily by rewriting the program in the memory. While the ideal model for retinotopic mapping has been proposed, the research about the relationship between stimulation to the visual nerves and the characteristics of evoked phosphene has been currently ongoing. The proposed architecture can change the entire or partial stimulation strategies more easily than the architecture without the microprocessor. Also, the implementations with a mi-

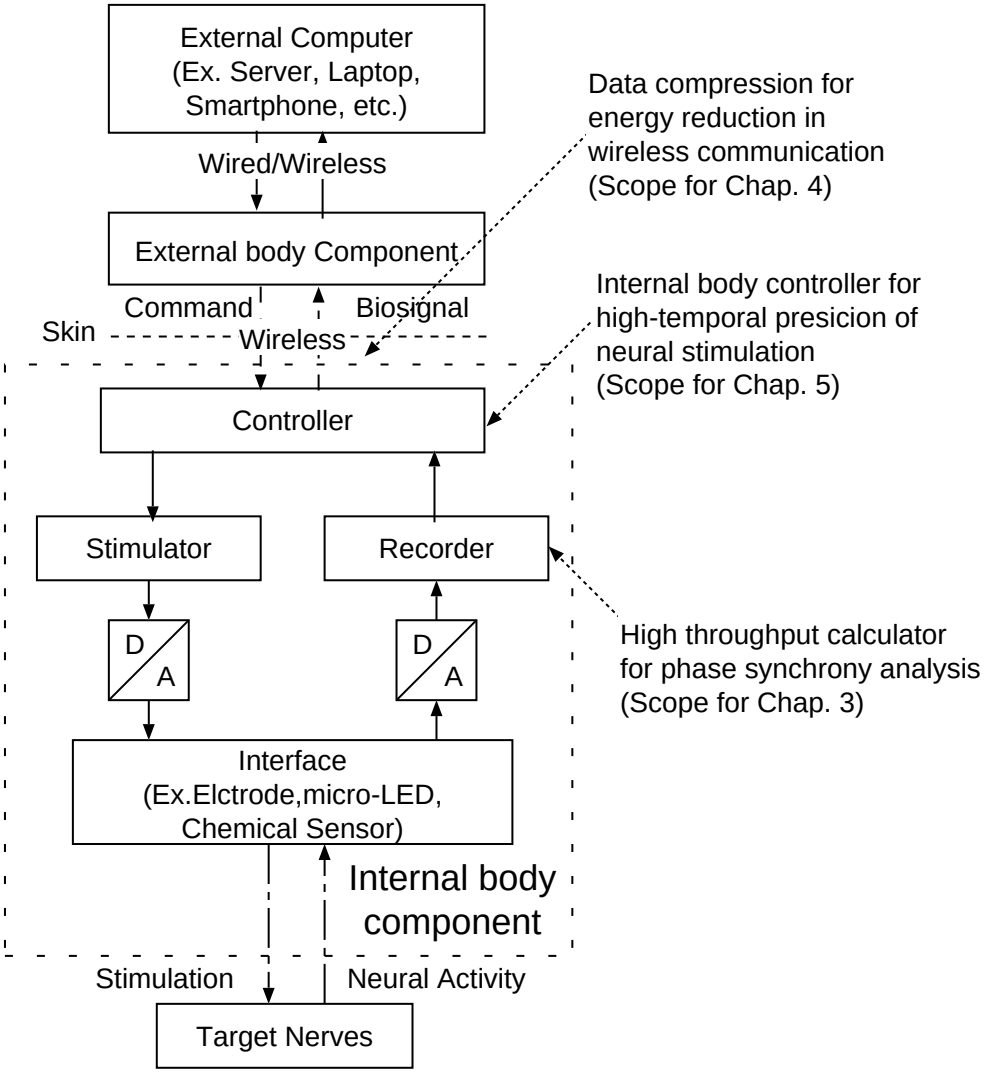


Figure 1.4: Contribution to low-energy and high-flexibility implantable neural stimulator.

croprocessor can reduce both the energy consumption by wireless communication and the latency from analysis to adjust stimulation because the microprocessor can process the closed-loop inside the body and it does not have to send the characteristics of recorded data to the external body component for extra calculation. While power dissipation and the circuit area of a digital core in the implementations with a microprocessor becomes larger than these without the microprocessor, the proposed implementation has potential to reduce total energy consumption in the device by closed-loop stimuli adjustment. At the current state of visual prosthesis research, the proposed architecture is reasonable to cope with both of the flexibility of stimulation and low energy architecture, and the proposed architecture may replace the hard-wired implementation when that research progresses enough to manipulate evoked phosphene. The point of the difficulty in the architecture is how to reduce energy consumption and shrink processing time in the microprocessor. This thesis tackles these problems to improve the energy efficiency by adding dedicated circuits and dedicated instruction-set to the microprocessor.

This thesis organizes the rest as follows: Chapter 2 discusses related work. Chapter 3 explains a high throughput and high hardware-efficient signal processor for on-line phase synchronization analysis. Chapter 4 explains a low computational data compression method for neural stimulation data to reduce energy consumption in the wireless communication component. The chapter also demonstrates an implementation of the proposed data compression method as ASIP. Chapter 5 describes a hardware implementation of low energy and high flexibility controller for neural stimulation as a system-on-a-chip, which includes the proposed ASIP in chapter 4. Finally, chapter 6 concludes this thesis and describes future work.

Chapter 2

Related Work

Chapter 1 describes an overview of the close interaction between of integrated circuits and medical applications. This chapter introduces integrated circuits for neural stimulation and neural recording as related works. Focusing on miniaturization of systems, this chapter gives a comprehensive introduction of cochlear implants (CIs), visual prostheses, and deep brain stimulations (DBSs).

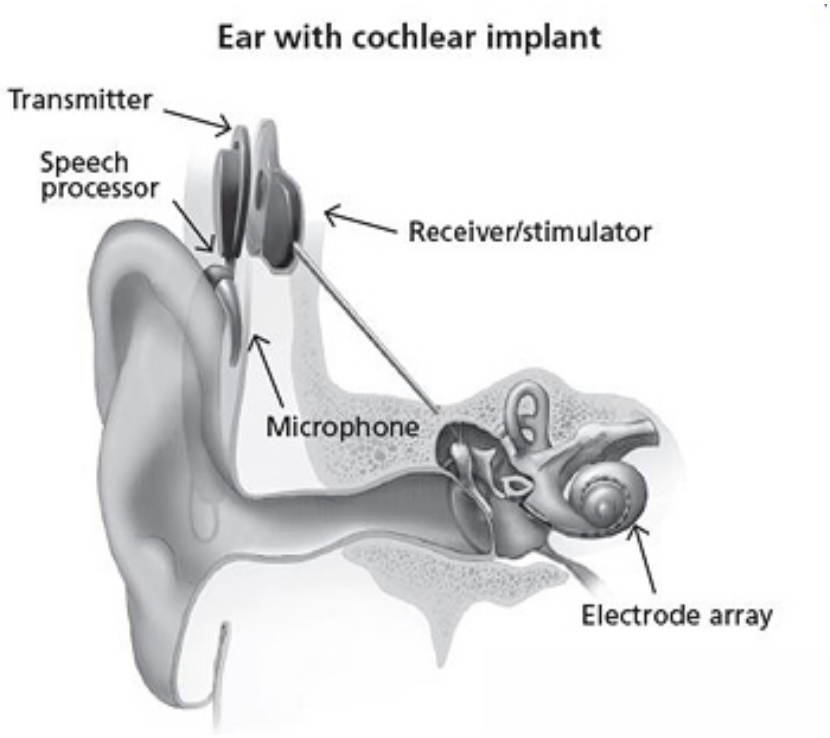


Figure 2.1: Illustration of a cochlear implant (Courtesy of [15]).

2.1 Cochlear Implants

CI has become the most popular device for neural stimulation. The CI partially restores hearing in auditory disorders by electrical stimulation to their auditory nerves via an electrode array implanted in the cochlea. Figure 2.1 illustrates an example of CI. The CI consists of two components: An external processor and an internal unit with electrodes. The external processor is composed of a sound input unit, such as a microphone and an analog front-end, a digital speech processor, and a wireless communication unit. The internal unit is composed of a wireless communication unit, a stimulation generator, and electrodes that are inserted in the scala tympani of a cochlear. The RF link between the internal and external wireless communication units also provide telemetry power supply to the whole of the internal unit. The feedback information from the internal unit to the external unit is used to observe the status of the electrodes.

In 1957, Djoumo and Eyriès performed the first direct electrical stimulation to the human auditory system [16, 17]. The first insertion of the CI was held by House and Doyle in 1961 [18]. The electrodes of first version CI were inserted into the scala tympani through the round window membrane. However, the implantation test had to stop due to the insufficient biocompatibility of the electrodes. In 1966, Simmons implanted single-wire electrodes into the modiolus of a volunteer, and he conducted basic studies about stimulation to cochlear[19]. His experiments were held with a chronically-implanted human subject for the first time.

At the beginning of the CI development, many experts asserted that the CI could not restore any useful hearing. However, through studies by the University of California at San Francisco, Michelson and Merzenich [20–22], people currently recognize the CI as an available treatment for deafness. Work [21] reported the limitation that the stimulation frequency for discriminative hearing is limited below 400–600 Hz while the stimulation of the spectrum from below 25 to above 10 thousand Hz generates hearing. A review by Bilger [23] in 1977 summarized the clinical utility of the CI is reported. This work claimed that the CI improved the quality of life for subjects compared with the scenes without it, and then people regard the CI feasible not only technically but also ethically. The improvement of the CI has been remarkable on today, and a review by Wilson *et al.*in 2008 reported that more than a quarter of the CI users marked perfect score in tests of sentence recognition [24].

2.2 Visual Prosthesis

The visual prosthesis is a recovery method with supports of the implantable medical devices (IMDs) to provide pseudo vision generated by external information for people who lost their sight partially or entirely. According to WHO's report about vision impairment in 2017 [25], the number of people with vision impairment was

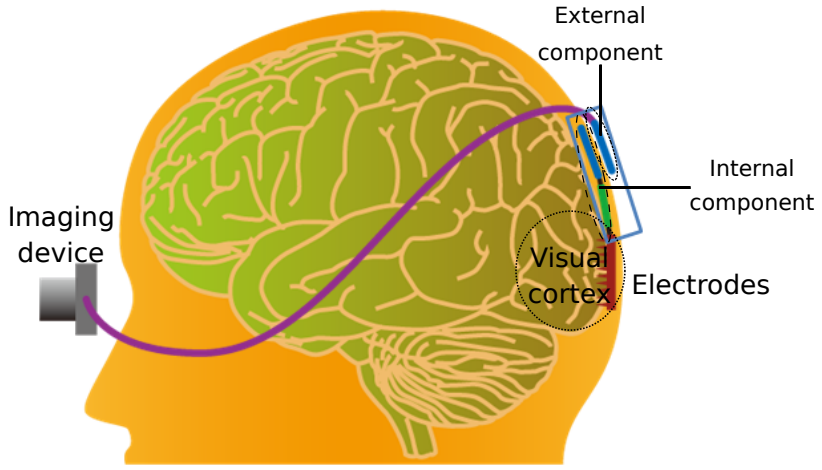


Figure 2.2: Illustration of a cortical visual prosthesis device (Courtesy of Seiji Kameda).

36 million, and 80% of them can recover by surgery. The primary target of the visual prosthesis is people whose vision impairment cannot cure by operations, which is caused by such as age-related macular degeneration, glaucoma, and diabetic retinopathy. From WHO's report about diabetes in 2017 [26], the number of people with diabetes was 422 million, and diabetes occupied 2.6% of the primary cause of blindness.

A principle of the visual prosthesis is that implanted electrodes stimulate target of optic nerves according to external information to emit visual perception to the brain. Figure 2.2 illustrates an example of cortical visual prosthesis device, and visual prosthesis system roughly consists of four components: an imaging device, an image converter, a stimulator, and a electrodes array. The imaging device obtains external body information as an image. The image converter converts the information from the imaging device to stimulation parameters; amplitude, duration, frequency, and position of stimulation. According to the stimulation parameters, the stimulator sends control signals to multi-electrode arrays via a D/A converter.

This research classifies visual prostheses into four types by stimulation target: Retina, optic nerve, lateral geniculate body, and visual cortex. The retinal prosthesis aims to recover functions of photoreceptor cells to use existing functions of remained optic nerves. The retinal prosthesis is classified by where the electrodes locate: On the inner surface of the retina (epiretinal) [27–29], between the retina and choroid (subretinal) [30–32], and between the choroid and the sclera (suprachoroidal) [33]. One of the advantages of the retinal prosthesis is the availability of photo-diodes arrays, which mimics the role of the photoreceptor. The photo-diode array converts light into an eye to electrical stimulation. This method allows surgeries to be safer and simpler thanks to miniaturization of the internal

eye components. The significant disadvantage of these methods is the difficulty of preparing sufficient energy for stimulation. To overcome this challenge, external energy supply was studied [30], and photovoltaic subretinal prosthesis whose photodiodes receive both power and data through pulsed near-infrared illumination (880–915 nm) converted by the outer video goggles was also studied [31]. Moreover, mechanical stability is one of the critical points for the feasibility of the retinal prosthesis because visual impairment people have the similar number of saccades as ordinarily visible people; between 100,000 and 150,000 eye movements per day [34]. Although the epiretinal methods require minimal power for stimulation owes to proximity to the retina, the epiretinal methods have a weakness at that point. In contrast, subretinal methods have enough stability against the saccades because the electrode array is located between the two layers, retina and choroid. It is a considerable idea to replace lost retina functions with the device so that the epiretinal methods can take advantage of the neural processing by mid-retinal layer and retinal ganglion cells more than the epiretinal method. However, there is a potential risk to impede blood supply from the choroid to the surviving retina, and hence this method has to minimize retinal damage by miniaturization of electrode arrays.

Visual cortical prosthesis started from a publication by Button in 1962 [35]. The visual cortex has adequate space for implanted devices and a large area for stimulation, and then many kinds of research have been investigated. The visual perception is evoked by stimulating the surface of the brain by tiled electrodes or inside of the brain via penetrating electrodes. On the other hand, it is unclear whether the perception processed by natural visual nerves is comparable to that one generated by the cortical stimulations, and hence studies about a relationship between stimulation and evoked visual perception has been ongoing.

According to work of Ghovanloo *et al*. [36], requirements for the visual prosthesis device comparing these of CIs are summarized as follows:

- More than 1000 stimulation sites for being able to read text with large fonts,
- flexibility to stimulation strategy to activate appropriately and efficiently 1.2 million nerves in optic nerves,
- low power dissipation less than 100 μW to extend the lifetime of batteries and high power requirements more than one mW to wireless power link,
- a large number of data transmissions to catch up with 60Hz human eye frequency, and
- extremely size limitation for the inner body component.

Also, there are some studies for redundancy reduction of input images to make up for lack of stimulation sites [37, 38]. The study [38] can shrink scene redundancy by retargeting objects that are devoted to more attention in human vision without changing the image size.

The visual prosthesis devices have been in clinical use. Argus II [39, 40], which applies the epiretinal method, has been on the market, and its clinical research [41] has been released. Experimental results show that the recipients have kept their vision ability restored by Argus II in five-years observation. Alpha–IMS [32], which applies the subretinal method, also has been in the clinical test stage.

2.3 Deep Brain Stimulation

DBS is a device-based therapy for such as epilepsy seizure resulted in Perkinson's disease, which has fewer side-effects and higher efficiency in drug-resistant patients than that based on drug or surgical. The primitive DBS device, also known as an open-loop stimulator, had only neural stimulator inside of the body, and the recipients or their attendants pushed a button of an external transmitter to generate stimulation when a seizure happened. The open-loop stimulators need setting manually by specialists who track the user's clinical condition and programming in a trial-and-error based manner. Although the open-loop stimulator is successful in commercials and clinicians, the adjustment of the stimulation parameters based on the on-going neurophysiological variations in the brain is required to reduce adverse effects by the stimulation. To overcome these problems, closed-loop stimulation device has been researched to adjust stimulation parameters according to the condition of target nerves. The closed-loop stimulation device consists of two components: a neural stimulator and a brain signal processor. The closed-loop stimulation device emits stimulation automatically when the brain signal processor detects or predicts a symptom of the seizure. Therefore, the closed-loop stimulation device can reduce the burden of the recipients. Table 2.1 summarizes the state-of-the-art of closed-loop neural interface research.

The closed-loop stimulator can reduce the energy consumption of the whole system. As DBS from reports by Little *et al*. [42] and Wu *et al*. [43], the closed-loop stimulator reduced energy consumption by more than 50% compared with the open-loop stimulator. The closed-loop control allows the devices to increase the energy efficiency of stimulation, which occupies dominant of the whole energy consumption in the internal body component. Therefore, the closed-loop enables to mitigate the users' burden and increase their quality of life (QoL).

Table 2.1: State-of-the-Art Closed-loop Neural Interface

Specification	[44]	[45]	[46]	[47]	[48]
Year	2016	2014	2016	2013	2017
Recording Target	N/A	EEG	Dopamin	EEG	EEG/ECoG
# Channels (Stim./Rec.)	8/8	1/8	N/A	64/64	24/24
Size	3.06×2.53	2.76×4.88	3.3×3.2	4.0×3.0	2.0×2.0
Technology [μm]	TSMC 0.18	TSMC 0.18	AMS 0.35	IBM 0.13	IBM 0.13
Supply Voltage [V]	1.0	1.8	2.5	1.2, 3.3	1.2
Power supply	Inductive	Inductive		No	Inductive
Data transmission (Tx/Rx)	LSK/ASK	OOK/OOK	FSK/–	UWB/–	FSK/–
Data rate [bps] (Tx/Rx)	2M/100K	4M/4M		10M/–	$\geq$ 200k
Stimulation current range [A]	82.5μ–229μ	30μ		10μ–1m	10μ–1m
Stimulation amplitude resolution [bit]	5	-	6	8	8
Stimulation frequency [Hz]	60–220	-	20–60	N/A	N/A
Stimulation pulse width range [s]	40μ–440μ	-	2.1m	N/A	N/A
Recording resolution [Samples/bit]	–	10	–	10	8
Sampling rate [Samples/s/channel]	–	62.5	–	$\leq$100k	3k
Closed-loop stimulation	No	Yes	Yes	No	Yes
Delay [s]	-	0.8	-	-	-
Power consumption of digital core	365 [μW]	1.298 [mW]	90 [μW]	1.4-1.5 [mW]	897 [μW]

Chapter 3

High Throughput Architecture for Phase Synchronization Analysis

This thesis discusses an architecture of neural signal recording and processing, which has a key role in the precision of closed-loop stimulation. The neural recorders are divided into two parts: Analog front end and digital signal processor, and a well-known feature extraction method in the digital signal processor is focused on, which is phase synchrony analysis. For precision and safety of stimulation, the closed-loop control requires a simultaneous pursuit of high-speed performance and low energy consumption for the digital signal processor. Therefore, this chapter discusses how to improve the throughput of the phase synchrony analyzer and how to decrease hardware implementation cost. In this chapter, a hardware-oriented phase synchrony analysis algorithm is demonstrated, which has lower computational complexity and enables faster processing than a straightforward conventional method. A hardware implementation which enables high throughput for feature extraction is also proposed, and then the proposed implementation is compared with one of the conventional methods regarding hardware efficiency.

3.1 Motivation and Objective

Electroencephalography (EEG) and electrocorticography (ECoG) are electrical signals that represent neural activities in the superficial layers of the cortex. EEG is recorded from the scalp, and ECoG is recorded by subdural grid electrodes on the cortical surface. The recent advances in microelectrode technology to acquire electrical signals in biomedical application enables the analysis of EEG and ECoG to not only understand the relationship between neural activity and cognitive behavior but also reveal the pathomechanisms in brain diseases such as seizure of epilepsy and Alzheimer's disease [49–51].

Owe to the outstanding progress of neuroscience, especially prediction with EEG and ECoG [52–56], wearable and implantable devices have been able to

diagnose numerous disease. Low energy consumption design of these devices provides the improvement of ease for carrying and surgery for implantation while excess energy consumption impedes portability. In these devices, the trade-off between processing speed and energy consumption is always the biggest issue. For practical applications, many researchers are conducting studies to find out how to realize both computational efficiency and low energy consumption in embedded systems, and hardware acceleration is indispensable for this purpose. However, most of the algorithms used in these applications are not initially designed for hardware implementation. Therefore, it is often difficult to break existing trade-off between the execution time and the energy consumption in hardware accelerator design.

In this thesis, we mainly focus on one of phase synchronization analyses, which is mean phase coherence (MPC). MPC is a method proposed by Mormann [57] for analyzing coordinated activation of distributed brain regions and enables to detect and predict seizures of epilepsy [56]. The derivation of MPC mainly consists of three processing: Band-pass filtering, Hilbert transform, and MPC calculation. The band-pass filtering extracts desired frequency band of interest, the Hilbert transform computes instantaneous phase of each wave, and the MPC metric quantifies phase-locking based on the difference between phase time courses of two waves. The filtering and the Hilbert transform can be efficiently implemented with finite impulse response (FIR) filters, but calculating MPC metric is computationally expensive in both software and hardware due to the use of trigonometric functions. Computers cannot calculate identical values of trigonometric functions like sinusoid because the width of registers is limited and finite combinations of four arithmetic operations cannot represent these functions. Therefore, calculation of the trigonometric functions uses approximate values that are calculated with approximate equations like Taylor expansion or are stored in look-up tables. To calculate MPC with trigonometric functions, Abdelhalim *et al*. [58] proposed the implementation that contains coordinate rotation digital computer (CORDIC), which Volder presented in 1959 in work [59], but the CORDIC requires a significant number of execution cycles and impedes processing throughput. Romaine *et al*. [60, 61] proposed discrete distance approximation (DDA) which can approximate phase synchronization values without trigonometric functions. However, the DDA sacrifices calculation accuracy of the phase synchronization. Also, Sylmarie *et al*. [62] proposed a simplified calculation of phase synchrony for ECoG to decimalize multi-bit input signals to binarize waveform with a statistically determined threshold. This method enables to reduce the complexity of calculator logics, but it has some drawbacks: This method sacrifices calculation accuracy and its threshold is predetermined value while that depends on characteristics of input signals and changed frequently.

For the increase in MPC calculation efficiency, this thesis proposes a novel MPC calculation method and its hardware implementation, in which its calculation result is identical to that of the original MPC. The processing target of the

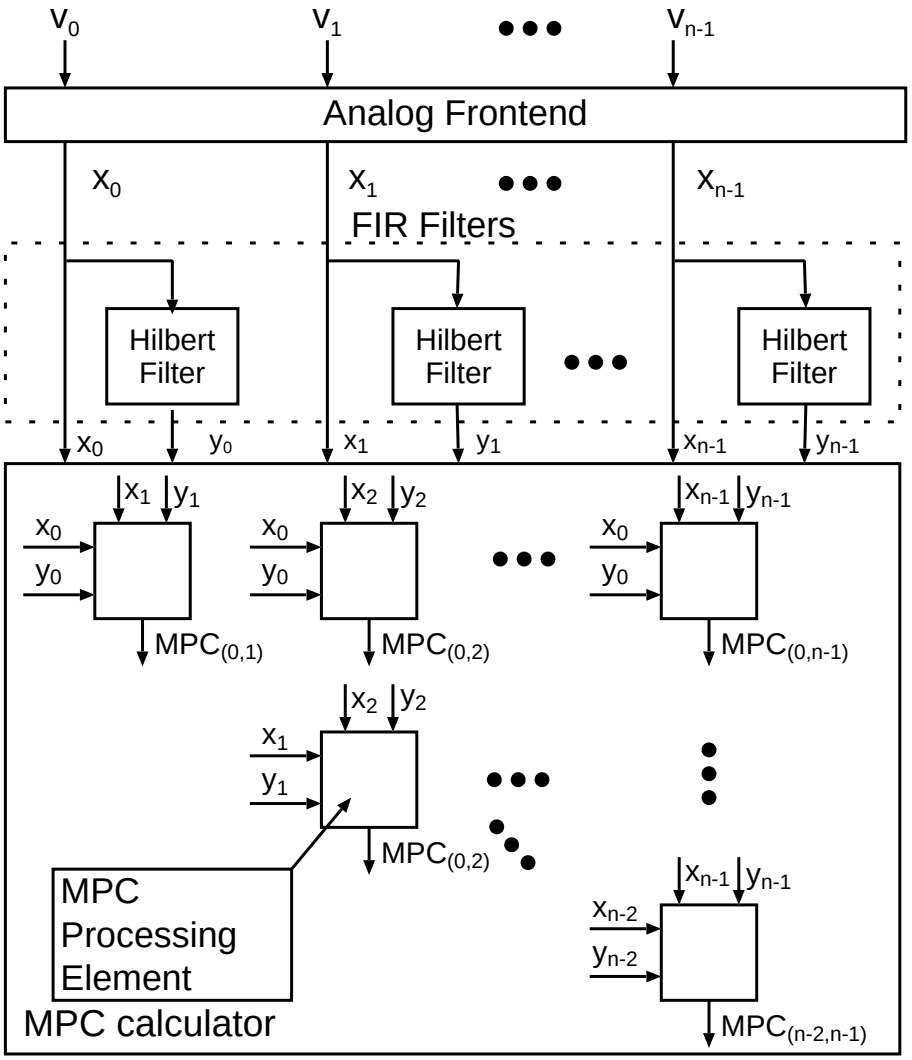


Figure 3.1: Overview of phase synchronization analyzer.

proposed method is electrical signals such as ECoG. Also, the proposed method is hardware-oriented and realizes the MPC calculation without trigonometric functions, so that it can accelerate its process. This contribution will provide higher accuracy of diagnosis and prediction of the critical situation in neural diseases with the clinical closed-loop stimulator such as deep brain stimulation (DBS).

3.2 Calculation of Mean Phase Coherence

This section explains the definition of the MPC and the proposed MPC calculation method without trigonometric functions. The phase synchronization analysis can be conducted by either of the following two methods: using the analytic signal or using wavelet transform [63]. The difference between the two methods is

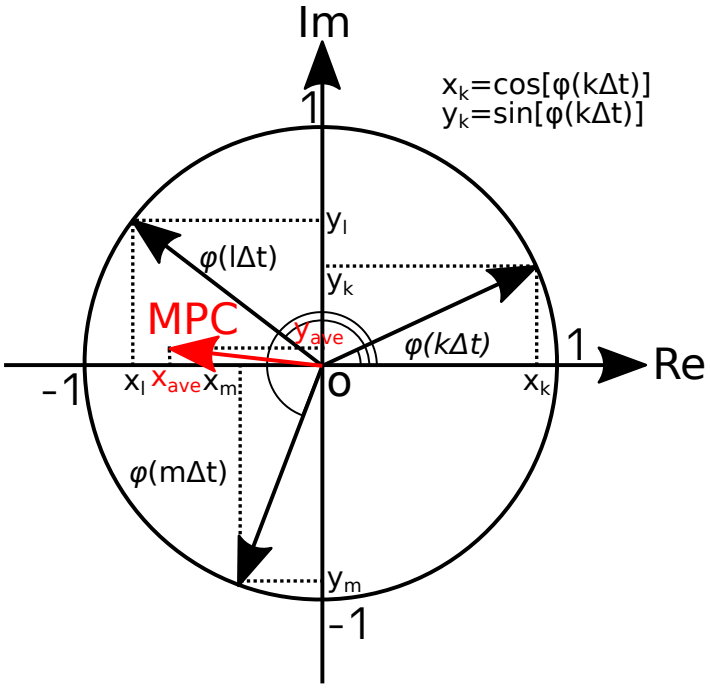


Figure 3.2: Definition of MPC.

minor [64], and the analytic signal approach requires less calculation amount than the wavelet approach. Therefore, this thesis studies the analytic signal approach.

Mean Phase Coherence

With the analytic signal approach, the MPC indicates the degree of the phase synchronization between the two channels represented by the circular variance of difference signal phase. Fig. 3.1 shows the target of phase synchronization analyzer. The analyzer obtains biomedical signals v_k, like EEG and ECoG, from multichannel electrodes, and then a digital block gets digitalized signal x_k from an analog front end. Through the FIR filters, an MPC calculator computes each pair of the signals. This study focuses on the MPC calculator in the bottom of Fig. 3.1. From [65], the analytic signal $Z_x(t)$ of a target signal $x(t)$ is given by the following equation:

$$Z_x(t) = x(t) + jy(t) = A_x(t)e^{j\phi_x(t)}, \tag{3.1}$$

where the function $y(t)$ is the Hilbert transform of $x(t)$, and the instantaneous amplitude $A_x(t)$ and the instantaneous phase $\phi_x(t)$ of the signal $x(t)$ are thus uniquely defined from (3.1). Also, $\phi_x(t)$ in $0 \leq \phi < 2\pi$ is defined by the following equation:

$$\phi_x(t) = \arctan\left(\frac{y(t)}{x(t)}\right). \tag{3.2}$$

From [57], the MPC of an angular distribution, which is denoted as R in the literature, is defined as:

$$\mathrm{MPC} = \left| \frac{1}{N} \sum_{i=0}^{N-1} e^{j[\phi_1(t_i) - \phi_0(t_i)]} \right|, \tag{3.3}$$

where N is the number of samples in a window. The use of Euler's formula turns the above formula into

$$\mathrm{MPC} = \left[\left(\frac{1}{N} \sum_{i=0}^{N-1} \sin[\varphi(i\Delta t)] \right)^2 + \left(\frac{1}{N} \sum_{i=0}^{N-1} \cos[\varphi(i\Delta t)] \right)^2 \right]^{\frac{1}{2}}. \tag{3.4}$$

An MPC calculation method based on (3.4) is referred as a conventional method in the following. From the trigonometric addition theorem, $\varphi(t)$ is calculated by

$$\varphi(t) = \phi_1(t) - \phi_0(t) = \arctan\left(\frac{y_0(t)x_1(t) - x_0(t)y_1(t)}{x_0(t)x_1(t) + y_0(t)y_1(t)} \right). \tag{3.5}$$

Figure 3.2 indicates the relationship between each vector of input signals and MPC value. (3.4) calculates the average of real values and imaginary values of the input signals separately. As shown in Fig. 3.2, MPC represents length of the vector, or Euclid norm, whose coordinate consists of the average of the real and imaginary value of the input signals.

Generally speaking, calculation of the trigonometric functions and division needs a large amount of computation in hardware implementation, and then it results in a significant amount of the execution cycles and the circuit area. In (3.4), sine, cosine, and arctangent are used, and the number of execution times of each function is equal to the number of samples. The trigonometric functions are calculated with look-up tables, polynomial approximation, or CORDIC. Calculation using look-up table is efficient when the targetted input range is limited. However, the input range of transformed signals by Hilbert transform is not limited, and hence the desired look-up table will significantly occupy a considerable amount of the circuit area. Polynomial approximation like Taylor expansion can calculate trigonometric functions with four basic arithmetic operations. For example, the Maclaurin expansion of arctangent of x in $-1 < x < 1$, sine of θ, and cosine of θ are represented by the following equation respectively:

$$\arctan(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} \tag{3.6}$$

$$= x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots, \tag{3.7}$$

$$\sin(\theta) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \tag{3.8}$$

$$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots, \tag{3.9}$$

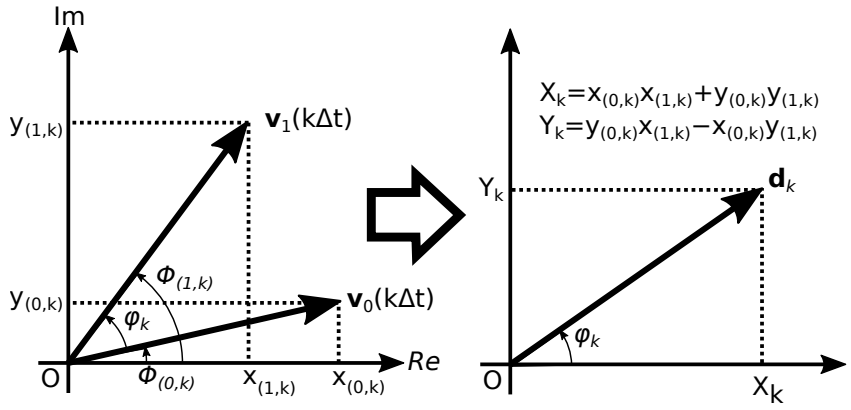


Figure 3.3: Conversion of a vector in (3.12).

$$\cos(\theta) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n} x^{2n} \tag{3.10}$$

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots . \tag{3.11}$$

Note that MPC calculation includes sine, cosine, and arctangent function and each approximate calculation is consists of the considerable number of additions and multiplications. Sine and cosine calculation have to wait to finish the processing of arctangent. While the calculation times of division can make zero by substituting for multiplications of constant values, such a large amount of calculation amount impedes throughput of MPC implementation. As proposed in [58], CORDIC enables to perform trigonometric calculations with adders, shifters, and a look-up table. On the other hand, the convergence of CORDIC operation for k-bit value requires k time iteration, hence CORDIC consumes k execution cycles to calculate the k-bit value and this occurs a bottleneck of throughput in MPC calculation. Also, straightforward k-bit division consumes k cycles and hardware-acceleration of division requires a large number of gates. To avoid these high-cost implementations, this thesis provides an MPC calculation method without the trigonometric functions and division from (3.4) to reduce hardware implementation cost.

Proposed Method

Highly accurate calculations for the trigonometric function require a significant number of execution cycles implemented as either software or hardware. With the trigonometric addition theorem and inner product calculations, equations equal to the MPC is expanded, and then the proposed method achieves an MPC calculation without the trigonometric functions. For MPC from (3.4), the equation is

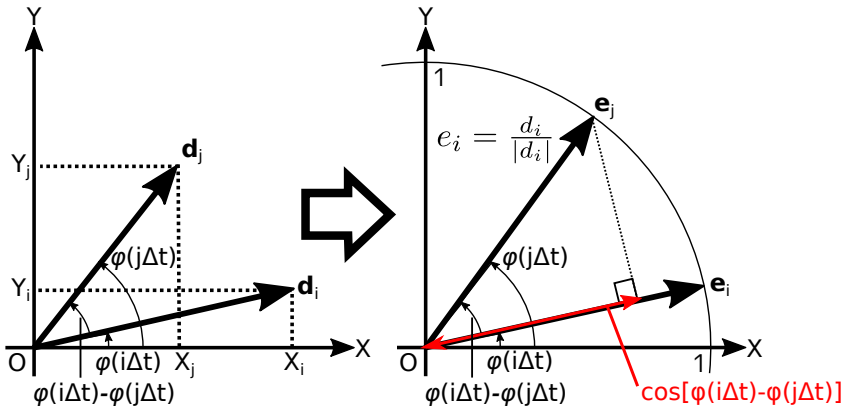


Figure 3.4: Conversion from cosine of $\varphi(n\Delta t) - \varphi(m\Delta t)$ to the inner product of unit vectors $\mathbf{e}$ in (3.15).

formulated and rewritten by the trigonometric addition theorem as follows:

$$\begin{aligned}\mathrm{MPC} =& \frac{1}{N}\left[\sum_{i=0}^{N-1}\Big(\sin^2[\varphi(i\Delta t)] + \cos^2[\varphi(i\Delta t)]\Big)\right. \\ &\left. + 2\sum_{n=1}^{N-1}\sum_{m=0}^{n-1}\Big(\sin[\varphi(m\Delta t)]\sin[\varphi(n\Delta t)] + \cos[\varphi(m\Delta t)]\cos[\varphi(n\Delta t)]\Big)\right]^{\frac{1}{2}} \\ =& \frac{1}{N}\left[N + 2\sum_{n=1}^{N-1}\sum_{m=0}^{n-1}\cos[\varphi(n\Delta t) - \varphi(m\Delta t)]\right]^{\frac{1}{2}}. \end{aligned} \tag{3.12}$$

In (3.12), the calculation of the second cosine term can be replaced by the following equation:

$$\cos[\varphi(n\Delta t) - \varphi(m\Delta t)] = \frac{\mathbf{d}_n \cdot \mathbf{d}_m}{|\mathbf{d}_n||\mathbf{d}_m|}, \tag{3.13}$$

where

$$\mathbf{d}_k = (x_{(0,k)}x_{(1,k)} + y_{(0,k)}y_{(1,k)}, y_{(0,k)}x_{(1,k)} - x_{(0,k)}y_{(1,k)}), \tag{3.14}$$

which is represented in the orthogonal coordinate system and the angular of $\mathbf{d}_k$ is $\varphi(k\Delta t)$ in the polar coordinate system. Here, Fig. 3.3 explains the conversion from the two input signals $\mathbf{v}_0$ and $\mathbf{v}_1$ to a new vector $\mathbf{d}_k$.

Then, it is possible to rewrite (3.4) as follows:

$$\mathrm{MPC} = \frac{1}{N}\left[N + 2\sum_{n=1}^{N-1}\sum_{m=0}^{n-1}\mathbf{e}_n \cdot \mathbf{e}_m\right]^{\frac{1}{2}} \tag{3.15}$$

$$\because \frac{\mathbf{d}_n \cdot \mathbf{d}_m}{|\mathbf{d}_n||\mathbf{d}_m|} = \mathbf{e}_n \cdot \mathbf{e}_m. \tag{3.16}$$

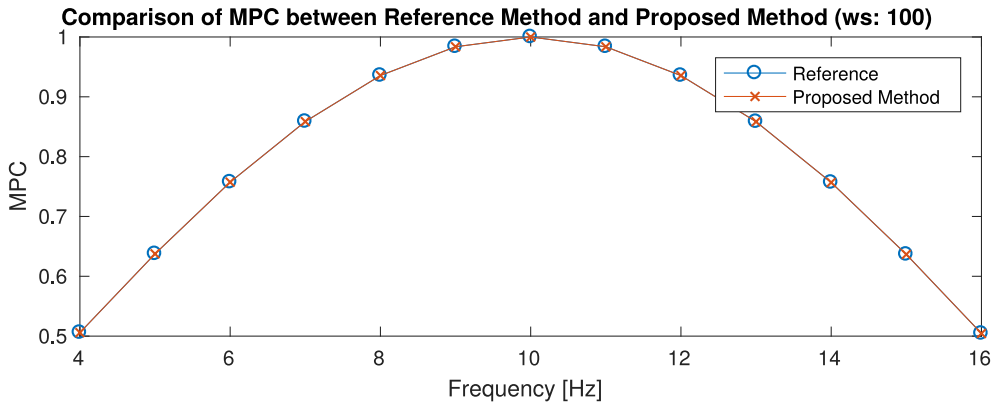


Figure 3.5: Comparison of MPC accuracy of proposed method and reference.

Figure 3.4 illustrates the conversion from cosine to the inner product of the two vectors $\mathbf{e}$. One of the advantages to use unit vectors $\mathbf{e}$ is the reduction of calculation amount. MPC calculation with vector $\mathbf{d}$ requires inner product, the norm of vectors, and one division for each combination. While reuse of calculation results reduces calculation amount, this method needs a large amount of memory to store results. On the other hand, unit vectors $\mathbf{e}$ allows MPC calculation to reduce memory because (3.15) requires storing only the unit vectors $\mathbf{e}$.

From (3.15), the number of the inner products calculations can be reduced as follows:

$$\sum_{n=1}^{N-1}\sum_{m=0}^{n-1}\mathbf{e}_n\cdot\mathbf{e}_m = \mathbf{e}_1\cdot\mathbf{e}_0 + \mathbf{e}_2\cdot(\mathbf{e}_0+\mathbf{e}_1)+\cdots + \mathbf{e}_{N-1}\cdot(\mathbf{e}_0+\cdots+\mathbf{e}_{N-2}) \tag{3.17}$$

$$= \sum_{n=1}^{N-1}\left(\mathbf{e}_n\cdot\sum_{i=0}^{n-1}\mathbf{e}_i\right) \tag{3.18}$$

Finally, an MPC calculation equation is proposed as the following equation,

$$\mathrm{MPC} = \frac{1}{N}\left[N + 2\sum_{n=1}^{N-1}\left(\mathbf{e}_n\cdot\sum_{i=0}^{n-1}\mathbf{e}_i\right)\right]^{\frac{1}{2}}. \tag{3.19}$$

The proposed equation and the MPC definition equation (3.4) are mathematically identical. Calculation results are coincident with these by (3.4) thanks to no approximation.

For validation of calculation precision, the proposed method was evaluated concerning the calculation accuracy in software simulation compared with a straightforward implementation of (3.4). The accuracy is evaluated in MATLAB environment version 2017a, and results are shown in Fig. 3.5. In this evaluation, the MPC between two channels is computed with one input held constant at 10 Hz

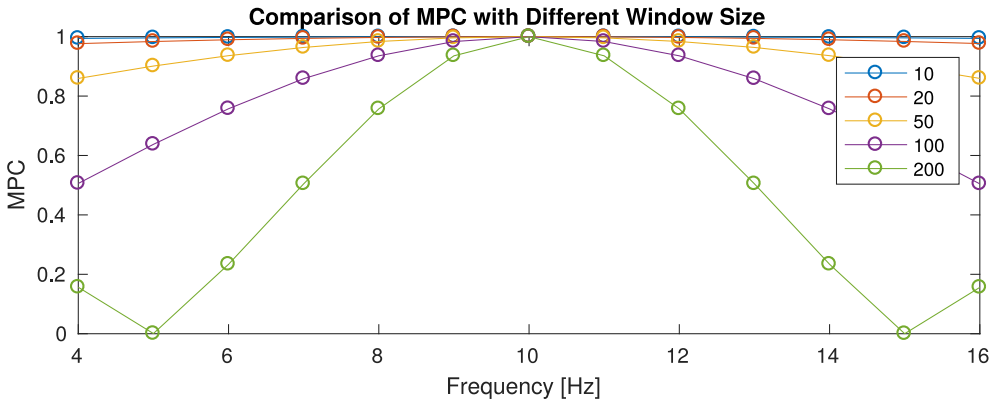


Figure 3.6: Comparison of MPC with different window size.

sinusoid, while the other input is swept from 4 Hz to 16 Hz. The graph shown in Fig. 3.5 demonstrates that the proposed method has comparable accuracy with reference values calculated by (3.4). This graph demonstrates that the line of the proposed method and that of the reference method is overlapped closely enough, and root mean square error (RMSE) of them is kept less than 1.1×10^{-16}. The graph in Fig. 3.6 illustrates that the MPC value is at unity when the two signals are the same frequency with different window size. The proposed method cannot express the difference of the frequency of the two input signals with small window size like 10, and the large window size can express the difference in frequency as that of MPC value.

Comparison of Computational Amount

The computational amount of each equation is listed in Table 3.1, where N denotes the number of samples in a window. Regarding normalization of the vectors, multiplying orthogonal coordinate components by reciprocal of square root (RSQRT) is applied instead of dividing them by square root (SQRT) because the latency by divider is longer than multiplier in general. Also, the square root of X is equal to multiplying the X by its RSQRT so that the SQRT calculator can be eliminated for calculation of (3.15) and (3.19). Shown in Table 3.1, (3.15) and (3.19) can calculate the MPC without the trigonometric functions. Moreover, the computational amount of (3.19) is proportional to N although (3.15) is proportional to the square of N.

3.3 Hardware Implementation

In this section, register-transfer level (RTL) level hardware architecture of the proposed MPC calculation method is introduced. Figure 3.7 represents a block diagram of the MPC processing element, where the black boxes and white boxes indicate registers and functional units, respectively. The proposed implementation

Table 3.1: Comparison of computational amount

Function	(3.4)	(3.12)	(3.15)	(3.19)
sin	N	0	0	0
cos	N	$\frac{N}{2}(N-1)$	0	0
atan	N	N	0	0
Mul.	3	$4N+1$	$5N^2-5N+1$	11N-9
Div.	N	N	0	0
SQRT	1	1	0	0
RSQRT	0	0	$\frac{1}{2}N^2-\frac{1}{2}N+1$	N

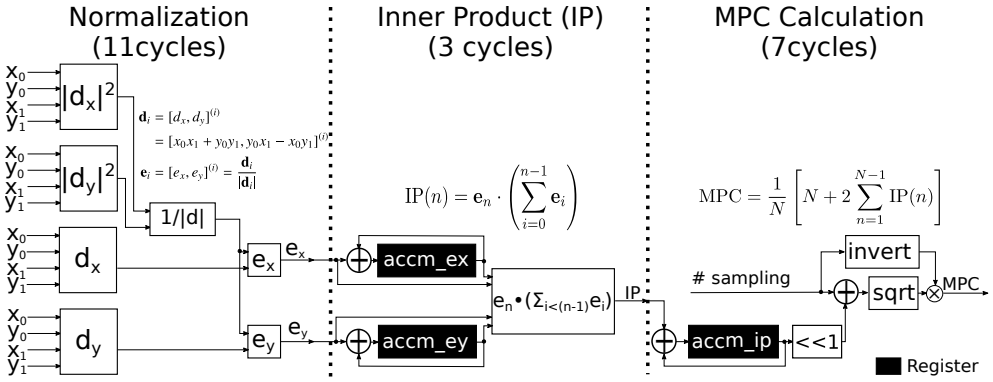
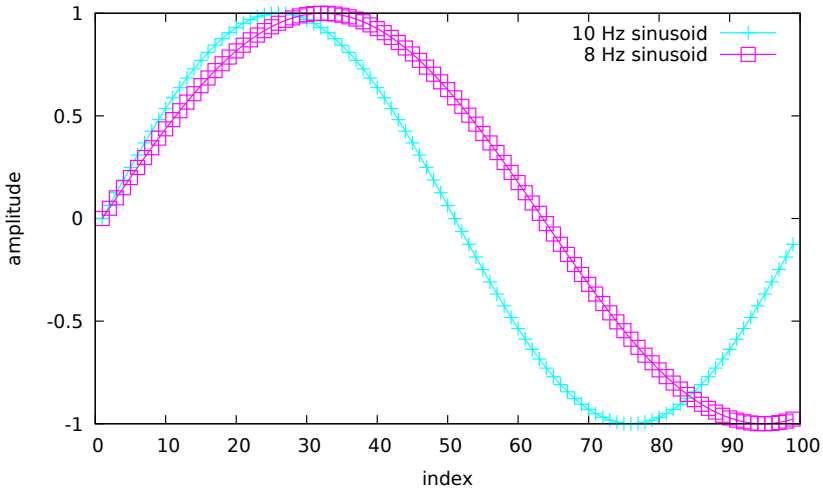


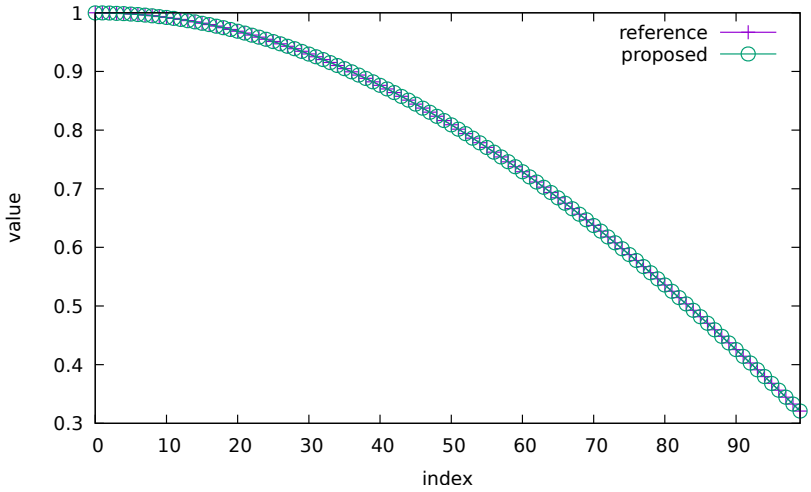
Figure 3.7: Block diagram of MPC processing element.

is divided into three stages of calculation: A normalization stage, an inner product stage, and MPC calculation stage. In the proposed implementation, calculation of both RSQRT and SQRT, which is needed in the normalization stage and MPC calculation stage, respectively, adapt Goldschmidt's approach [66] shown in Algorithm 1. Compared with the Newton-Raphson's approach, the Goldschmidt's approach changes two calculation modes of RSQRT and SQRT more simply. Moreover, its implementation consists of less arithmetic function units. In the proposed implementation, it takes five cycles to converge the calculation results by the implementation of the Goldschmidt's approach. The normalization stage uses four adders and three multipliers, the inner product stage uses four adders and one multiplier, the MPC calculating stage uses two adders and one multiplier, and the calculator of Goldschmidt's approach uses two adders and four multipliers. The latency of the implementation is 21 execution cycles: eleven for normalization, three for updating the inner product, and seven for calculating of MPC. The pipeline interval is five cycles because the calculation of either RSQRT or SQRT requires five cycles.

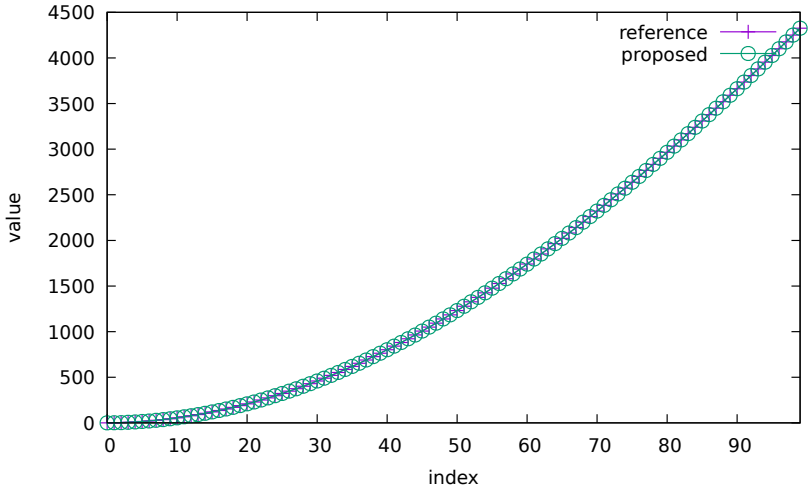
Comparisons of the reference values and the implementation are shown in Fig. 3.8. Figure 3.9 (a) shows the two input signals for evaluation, where one



(a) Input signals; 100 samples of 10 Hz and 8 Hz sinusoid

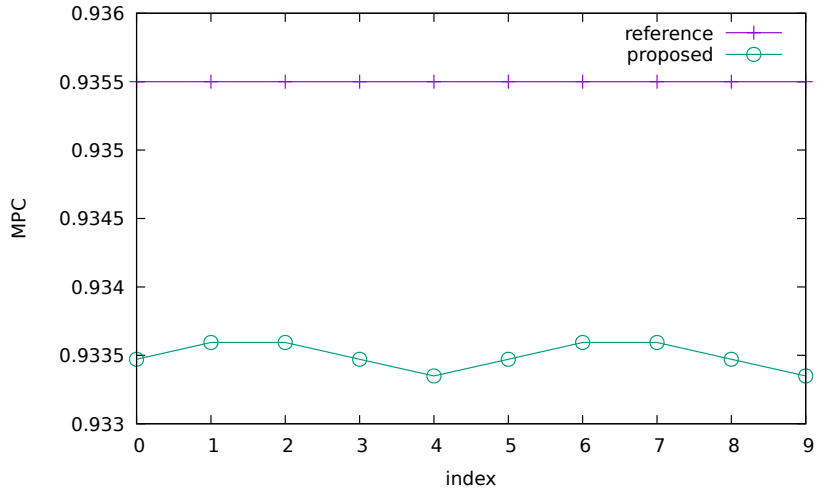


(b) Comparison of ex_i



(c) Comparison of *acc_ip*

Figure 3.8: Evaluation of implementation.



(d) Comparison of *MPC*

Figure 3.8: Evaluation of implementation. (cont.)

is 10 Hz sinusoid and the other is 8 Hz sinusoid. Like literature of [58] and [60], the sinusoid waves are used for the evaluation of accuracy because the feature of input signals is ideal and it is convenient of visualization of experimental results. Figures 3.9 (b) and 3.9 (c) show comparisons between intermediate values of the proposed implementation and those of the reference method. These graphs show that the proposed implementation has comparable accuracy with the reference method because lines produced by the proposed method match closely to those of the reference method. Figure 3.8 (d) depicts the comparison between the

Algorithm 1: Goldschmidt's square root and reciprocal of square root approximation

Input: Fixed-point input value:In, Number of loop: N, Mode flag: M
Output: Fixed-point output value: Out

if M == *'reciprocal'* **then**
  $Z_0 := 1$;
else
  $Z_0 :=$ In;
end
repeat
  $D_i := 1.5 - 0.5X_i$;
  $X_{i+1} := X_i D_i^2$;
  $Z_{i+1} := Z_i D_i$;
until N;
Out := Z_{N-1};

Table 3.2: Comparison of Implementation of Circuit Specification

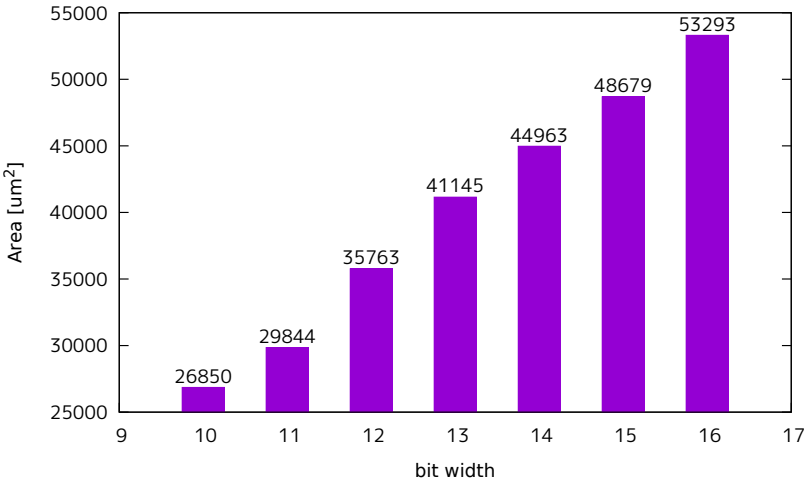
Specification	Abdelhalim [58]	Proposed	Diff. [%]
1.2V CMOS Technology	0.13μm	45nm [67]	–
Latency [cycle]	54	21	–61.12
Throughput [Samples/cycle]	5.5×10^{-2}	2.0×10^{-1}	+263.6
Gate count [gate]	41 336	28 389	–31.4
Throughput/Gate count	1.3×10^{-6}	7.0×10^{-6}	+529.9

MPC value of the proposed implementation and those of the reference method. This graph shows that the proposed implementation can calculate the MPC with small errors, and the RMSE is less than 0.15. From results, the proposed method has comparable accuracy for MPC calculation compared with the conventional method. Table 3.2 summarizes comparison results of simulation and implementation. In 16-bit precision, gate count of the proposed implementation was 28389, and the power dissipation was 1.6 mW. For evaluation of implementation efficiency, the efficiency is defined by a value that throughput divided by gate count, and then the larger number indicates the high efficiency of hardware implementation. Therefore, the proposed implementation is about 5.3 times better than Abdelhalim's implementation. In contrast to Abdelhalim's method using CORDIC [58], the proposed method can achieve high-performance thanks to the elimination of trigonometric functions.

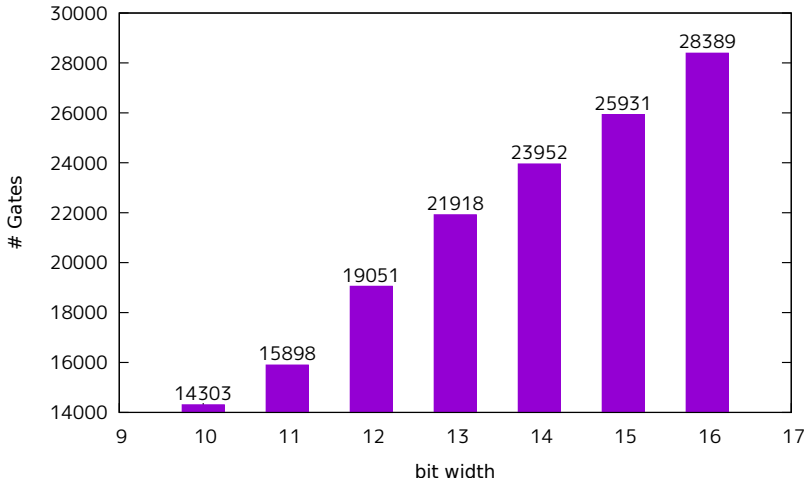
3.4 Discussion

From the evaluation results, this section discusses the hardware implementation efficiency of the proposed method. Have mentioned above, the closed-loop stimuli controls require a simultaneous pursuit of high-speed performance and low hardware cost for its signal processing. The proposed method can calculate the MPC with high throughput and low hardware cost, which is demonstrated with the metric of "throughput per gate count."

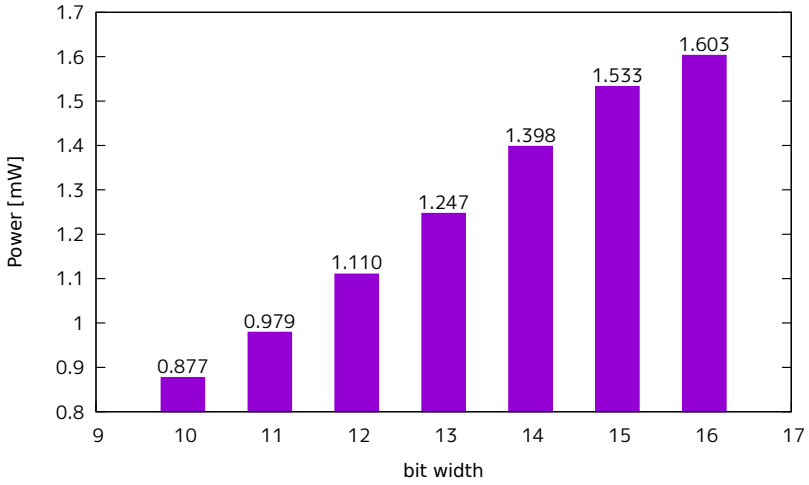
The proposed implementation can calculate the MPC with only addition, multiplication, RSQRT, and SQRT, and it needs 21 execution cycles per one execution. In Abdelhalim's work [58], CORDIC is utilized for calculation of trigonometric function and root square, and it needs 54 execution cycles per one execution. The CORDIC can calculate trigonometric functions and SQRT with enough accuracy, however, it needs a large number of execution cycles until completing calculation. In contrast, the proposed method replaces the trigonometric functions with inner product calculation, which requires additions, multiplications, and RSQRT. The addition and multiplication can be completed in only one cycle. The RSQRT is calculated by Goldschmidt's approximation [66], which can calculate its result with enough accuracy in smaller execution cycles than the CORDIC.



(a) Area of Circuit.



(b) Number of gate counts.



(c) Power dissipation.

Figure 3.9: Relevance of bit width in proposed circuit and results of logical synthesis.

Therefore, the proposed implementation increases the throughput of MPC calculation by 263 % compared with the Abdelhalim's work [58]. This improvement will allow the implantable neural analyzer to miniaturize its volume and accelerate its processing time, and then these will lead more promotion to neuroscience study and clinical use. The proposed implementation also achieves higher hardware implementation efficiency than the Abdelharlim's work. Note that the hardware implementation efficiency defined above does not depend on their operating frequency or technology process. The proposed method increases the hardware implementation efficiency by 529 % compared with the conventional implementation.

The bars in Fig. 3.9 depict the relevance of bit width in the proposed implementation and characteristics. In the Abdelhalim's work [58], the signal processor has 10-bit MPC and 16-bit input signals are used. Each graph demonstrates the characteristics from 10-bit to 16-bit precision. These figures show that the proposed implementation in 10-bit precision was 26860 μm^2, 14303 gates, and 0.8 mW power dissipation. Hence, the hardware implementation efficiency is 10.8 times better with the same output precision compared with the Abdelhalim's work [58]. Mentioned above, this thesis applies the implementation in 16-bit precision according to the bit width of the input signals.

3.5 Conclusion

In this chapter, a hardware-oriented MPC calculation method was proposed. For improvement of stimulation control, the biomedical signal processing has been required a simultaneous pursuit of high-speed performance and low-cost implementation. MPC calculation is one of the well-known methods for feature extraction of biomedical signals, and it has been utilized for detection and prediction of the epilepsy seizure. The proposed method attains high computational efficiency by substituting trigonometric functions with the computation of linear algebra. In the comparison between software implementations, the proposed method achieved about 20 % reduction in processing time compared with the straightforward implementation of definition equation of MPC. This chapter also proposed a hardware implementation that aimed at a simultaneous pursuit of high throughput processing and implementation efficiency. From evaluation results, the proposed implementation improved throughput of the MPC calculation by 2.6 times and the hardware implementation efficiency by 5.3 times, respectively, compared with the conventional implementation. This improvement in hardware implementation will provide the neural analysis devices with more miniaturization and processing power.

Chapter 4

Low Computational Data Compression Architecture for Neural Stimulation

This chapter describes a data compression method for visual stimulation position data with a low computational amount. First, this chapter explains why neural prosthesis systems need low-computational data compression method, and analyzes characteristics of target data statistically, which is called as stimuli position data. Next, a data compression method for the target data is proposed and evaluated by comparing with well-known compression methods. Moreover, the proposed method is implemented with an application specific instruction-set processor, and it is evaluated regarding energy consumption in wireless communication.

4.1 Motivation and Objective

Regardless of where electrodes stimulate, stimulation components in visual prosthesis systems require low-power and low-energy design. The system requires miniaturization because the space to insert the systems is highly limited. For example, regarding the retinal prosthesis, the diameter of an eyeball is from 22 mm to 24 mm, and the stimuli sites are penetrated on or inserted into the retina tissue. For avoiding damages to the users, the volume of stimulation components should be minimized. The amount of power transmission with inductive coupling is limited (see [68] to find the state-of-the-art methodologies), and the secondary battery has concerns about safety and risk of surgery for replacement. Reduction of energy consumption enables to decrease the volume of the entire systems and mitigate extra heat dissipation.

Figure 4.1 illustrates the target visual prosthesis system of this thesis. The target system consists of two part: Outside body module and internal body module. The external body module has an imaging device, an image converter circuit, and a wireless communication unit, and the internal body module has a wireless

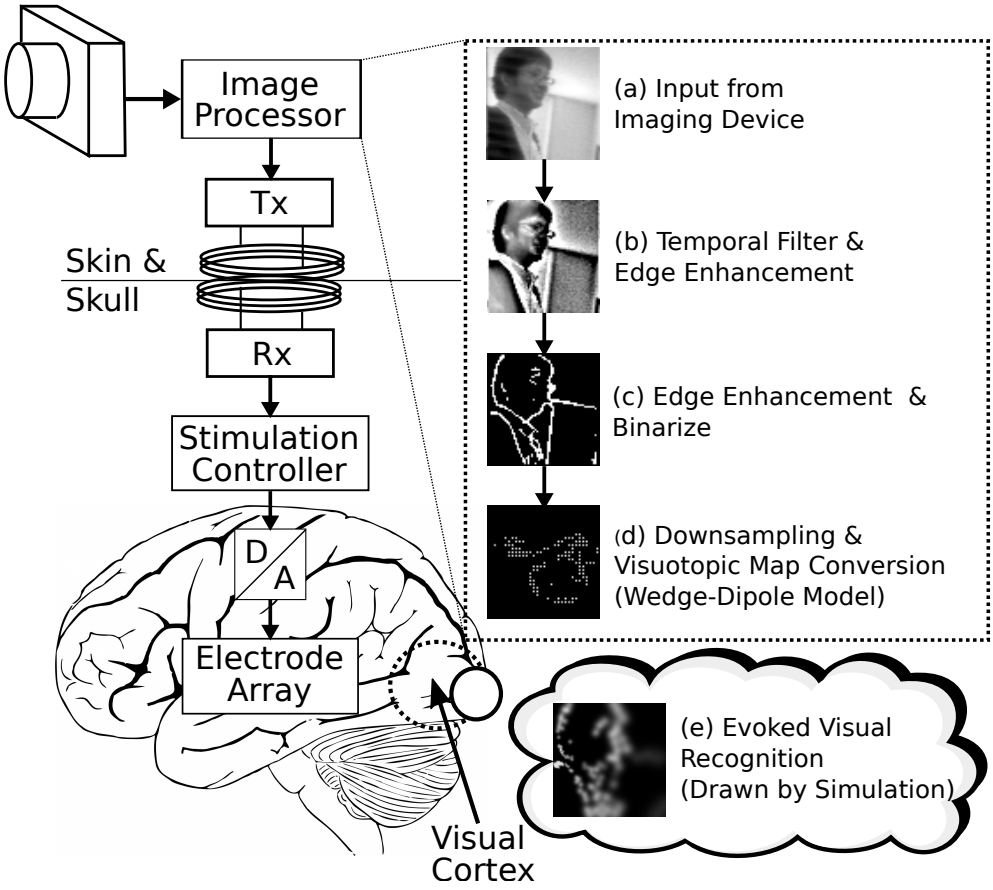


Figure 4.1: Overview of target visual prosthesis system.

communication unit, a stimulation controller, a signal converter for stimulation, and an array of stimuli electrodes.

For reducing the entire energy consumption of the systems, this study aims to reduce the amount of data transmission in wireless communication. Since energy consumption in wireless communication is roughly proportional to the amount of data transmission, and it is dominant of whole energy consumption of the systems. Improvement of visual recognition requires higher spatial and temporal resolution of stimulation, which increases the amount of data transmission. According to [36], the number of stimulation sites reaches more than a thousand in practical use. Raising transmission speed of the wireless communication unit is one possible way to resolve this concern, but this will cause the increase in heat dissipation at the internal body component and damage the optic nerves. The rise of energy density by high-speed transmission also will damage the tissue of skull and scalp, which compromises the user's quality of life significantly. Therefore, the visual prosthesis system for clinical use requires an alternative solution to manage the trade-off between transmission amount and effects for the user.

This thesis proposes a data compression method for neural stimulation data

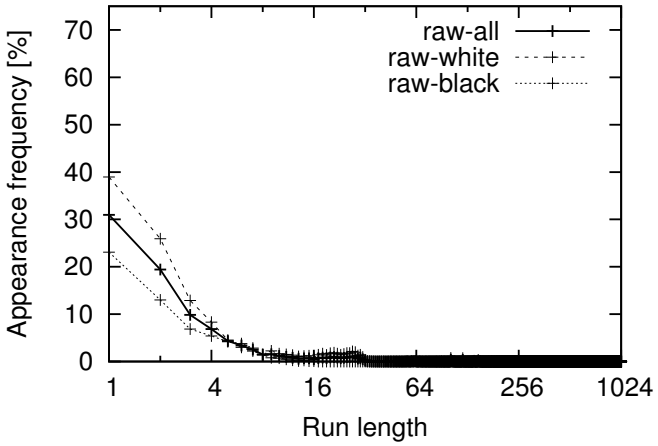
to reduce data transmission. Compression methods are classified into two types, lossless and lossy. In general, lossy compression methods can reduce the amount of data more than lossless compression methods. However, the lossy compression methods also lose some information in the target data in exchange for less data amount. The systems have to stimulate target organs according to the stimulation position data faithfully. Moreover, the energy to use for compression is limited, and hence the proposed compression method should concern the trade-off between computational amount and energy consumption. In this thesis, the proposed compression aims a simultaneous pursuit of a low computational algorithm and a low energy implementation. The proposed compression method and implementation will lead to miniaturization and low energy implementation of the neural stimulation devices. Because the visual prosthesis requires a large amount of transmission for desirable spatiotemporal resolution in evoked visual recognition, the proposed compression method will achieve significant energy reduction in the internal body component.

4.2 Data Analysis

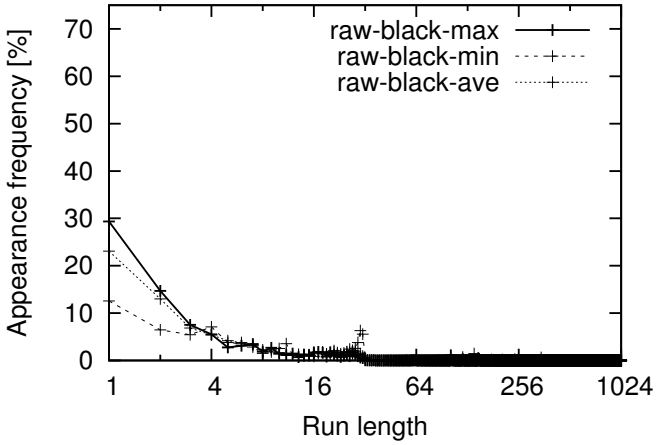
This section explains an overview of target visual prosthesis device firstly and the statistical analysis of data for electrodes activation called the stimuli position data. Then, the proposed lossless data compression method is explained, and it is compared with the well-known compression methods such as Huffman coding concerning compression efficiency.

4.2.1 Target Visual Prosthesis Device

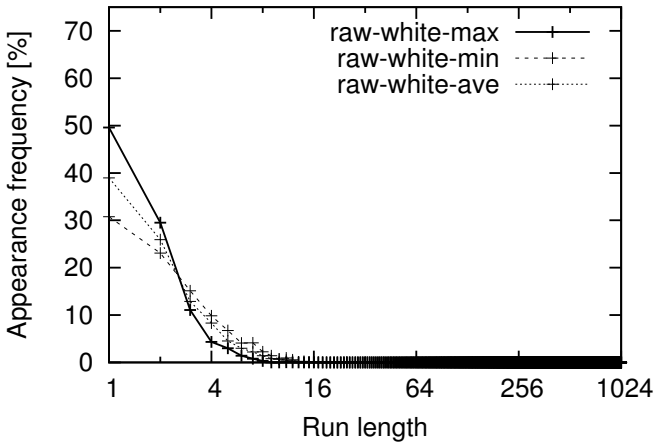
In transmissions from the external body component to the internal body component, transmitted data contain parameters for stimulation control: Amplitude, polarity, duration, position, and repetition of stimulation. This chapter focuses on the stimuli position data because it occupies a large part of the traffic in wireless communication. Figure 4.1(a) is an input image from the imaging device, and Fig. 4.1(e) is an output visual perception. By the image conversion circuit, the input data of the obtained image is processed for edge enhancement, and an image in Fig. 4.1(b) is generated. After that, edge extraction from binary image data of Fig. 4.1(b) generates an image in Fig. 4.1(c). Next, the image convert circuit transforms and quantizes the data mentioned above to stimulation parameters: Amplitude, duration, frequency, polarity, repetition, and position of stimulation. Figure 4.1(d) illustrates a stimuli position data. In this thesis, the image conversion circuit converts the data according to Wedge-Dipole mapping model [69], which is reported by Fehervari in [70]. Then, the internal body module receives the stimulation parameters from the external body module, and the stimulation controller and the signal converter emit stimulation. Finally, the user can receive stimulation as pseudo vision like Fig. 4.1(e).



(a) Appearance frequency of the run-length in raw stimuli position data.



(b) Appearance frequency of the run-length of black dots in raw stimuli position data.



(c) Appearance frequency of the run-length of white dots in raw stimuli position data.

Figure 4.2: Analysis results of the appearance frequency of the run-length of white and black dots in raw stimuli position data and difference data.

Figure 4.1(d) shows an example of an image representing stimulus position data as a binary bitmap. The stimulus data is composed of 32×32, and 1024 bit in total. Each pixel is related to the coordinate of electrodes, and the value of each pixel represents whether the electrode at the same coordinate provides a stimulus or not. If the data is '1', represented as a white dot in Fig. 4.1(d), the electrode emits stimulation. If the data is '0', represented as a black dot in Fig. 4.1(d), the electrode does not.

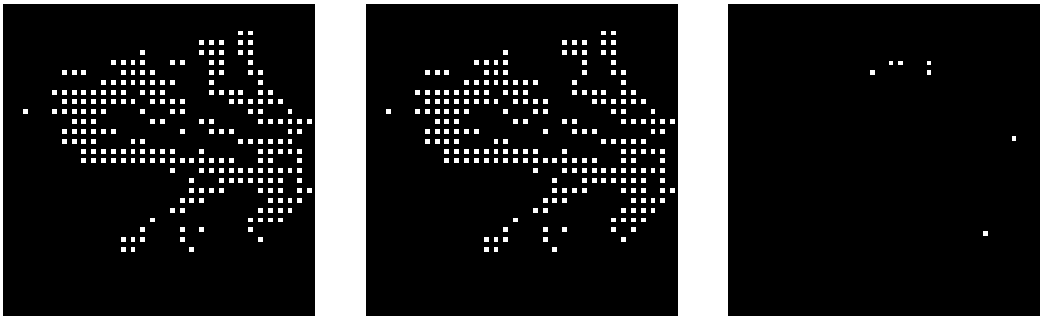
4.2.2 Analysis of Stimuli Position Data

In Fig. 4.1(d), black dots occupy much space, and most of all these dots are successive in long length in the horizontal direction. On the other hand, some white dots are successive in short length and isolated. Therefore, we use the run-length encoding to express stimulus data, and we will analyze the distribution of run-length data to identify the encoding method that marks higher compression efficiency. The dataset used in this analysis includes the following objects in the input image: walking people, running cars, bicycles, and motorbikes on the road. Forty-five samples of scenes in various situations such as on narrow roads, in the cross of arterial roads, and at the station are captured and used.

Fig. 4.2 shows the appearance frequency of the run-length of white dots and black dots in the stimuli position data. The x-axes of Figs. 4.2 (a), 4.2 (b), and 4.2 (c) represent the run-length, and the y-axes represent the appearance frequency of the run-length. In Fig. 4.2 (a), 'raw-all' indicates the average of the whole run-length from all the datasets and 'raw-white' and 'raw-black' indicates the run-length of white dots and black dots, respectively. Figures 4.2 (b) and 4.2 (c) show the variance of the appearance frequency of the run-length of each dot. In these figures, '-max' and '-min' indicate the samples that score the highest and lowest frequencies of run-length whose length equal to one, respectively, and '-ave' indicates the average of appearance frequency from all the datasets. Fig. 4.2 shows that the probability distribution of 0's run-length data is different from that of 1's run-length data. In other words, 1's run-length is shorter than 0's run-length. A run-length whose length is one occupies 30% of 1's run-length, while it occupies 22% of 0's run-length. Furthermore, the run-lengths whose length is less than or equal to eight occupy about 90% in 1's run-length, but on the contrary, they occupy about 65% of 0's run-length. From these analysis results, the 0's run-length and the 1's run-length have a similar bias to small length in appearance frequency, and utilizing this characteristic is helpful to make a compression algorithm efficient.

4.2.3 Difference Data of Successive Stimuli Position Data

Fig. 4.3 (a) and Fig. 4.3 (b) are successive frames of stimuli position data. Fig. 4.3 (c) shows the difference between Fig. 4.3 (a) and Fig. 4.3 (b) by calculating exclusive-



(a) Current Stimuli position data

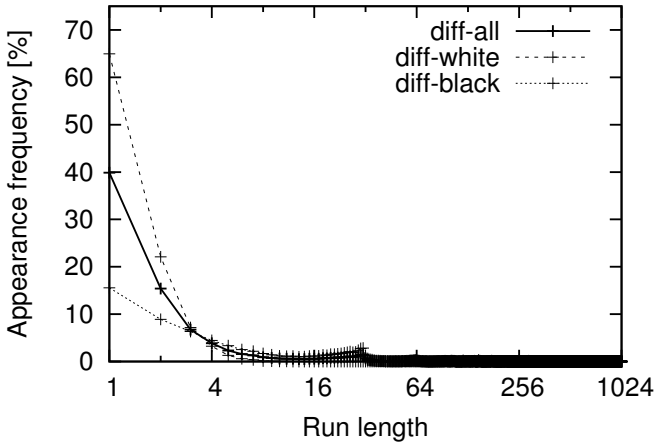
(b) Previous frame of (a)

(c) Difference data between (a) and (b)

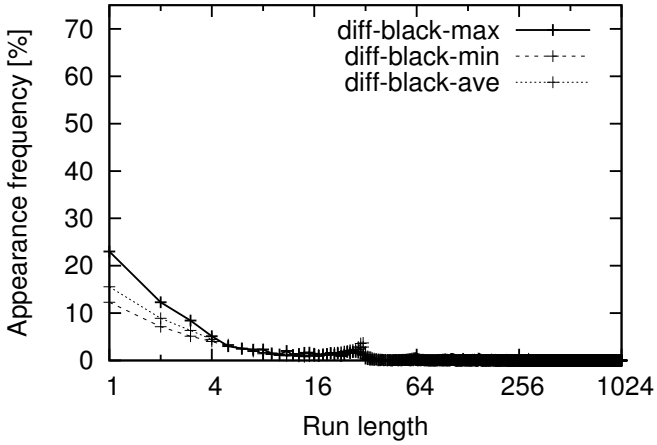
Figure 4.3: Examples of continuous stimuli position data and difference data.

or operations. In Fig. 4.3 (c), white dots represent the change of the stimulus status from the previous frame as '1' in data, and black dots represent the same status in the previous frame as '0' in data. There are more successive black dots and less isolated white dots in Fig. 4.3 (c) than in Fig. 4.3 (a). In the rest of this chapter, the stimuli position data is called raw data in distinction from difference data.

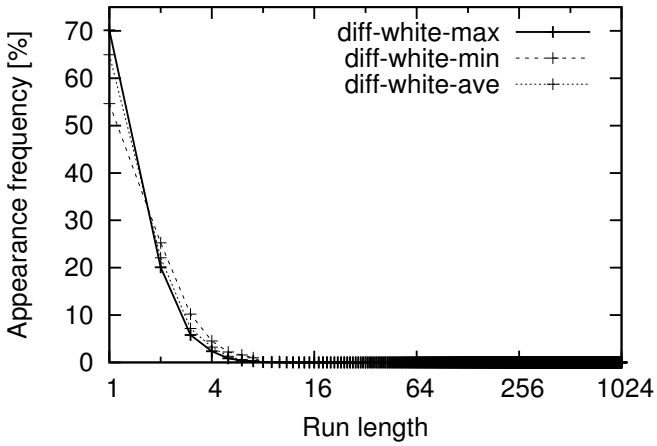
Fig. 4.4 shows the appearance frequency of the run-length of the modified point and the stable point of the difference data. As in Fig. 4.2, the x-axes of Fig. 4.4 (a), 4.4 (b), and 4.4 (c) represent the run-length, and the y-axes of these figures represent the appearance frequency of the run-length in the sample. In Fig. 4.4 (a), 'diff-all' indicates the average of the whole run-length from all the datasets and 'diff-white' and 'diff-black' indicate the run-length of white dots and black dots, respectively. Fig. 4.4 (b) and Fig. 4.4 (c) show the variance of the appearance frequency of the run-length of each dot. In these figures, '-max' and '-min' indicate the samples that score the highest and lowest frequencies of run-length whose length is equal to one, respectively, and '-ave' indicates the average of appearance frequency from difference data generated from all the datasets. Fig. 4.4 (a) shows that the frequency of the occurrence of 1's runs is extremely biased to be short on the difference data. For the length of 1's run, the appearance frequency of run whose length is one occupies about 64%, which is nearly twice as much as that in the raw data. For the length of 0's run, the frequency of occurrence of run whose length is less or equal to eight occupies 40%, which is significantly different from that in the raw data of 65%. In the difference data, the frequency of occurrence of 0's run whose length is more than 32 occupies 20%, while they occupy 3% of the raw data. From the analysis results of the difference data, the 0's run-length and the 1's run-length have more distorted bias in appearance frequency than the stimuli position data. Therefore, the difference data provides more biased distribution, and separating the compression method for the 1's run-length from the one for the 0's run-length is helpful to compress the difference data more efficiently.



(a) Variance of appearance frequency of the run-length in difference data



(b) Variance of appearance frequency of the run-length of black dots in difference data



(c) Variance of appearance frequency of the run-length of white dots in difference data

Figure 4.4: Appearance frequency of run-length of white and black dots in difference data.

Table 4.1: Codeword table of Elias α coding

Input	Codeword	Code length
1	1	1
2	01	2
3	001	3
4	0001	4
⋮	⋮	
7	0000001	7
8	00000001	8
⋮	⋮	

4.3 Data Compression Method

From the above analysis, this study revealed characteristics of the stimuli position data, which has strong correlation in both spatial and temporal domain, especially in the difference data. Then, a data compression method that takes advantage of these characteristics is proposed, which is called α exponential Golomb coding (α-EGC) in this thesis. The proposed method is effective for binary data that has different distribution between 1's run and 0's run to apply different compression methods according to each characteristic of bias.

The α-EGC compresses the stimuli position data in the following steps:

1. Generating the difference data from the two successive frames of stimuli position data,
2. counting the run of successive bits in the difference data with raster scanning, and
3. encoding the length of runs into a bit stream
   (a) with the offset using the exponential Golomb coding (EGC) [71] if the run consists of unchanged bits, and
   (b) with the offset using the Elias alpha coding [72] if the run consists of changed bits.

The α-EGC encodes the length of runs with different encoding methods. For the length of 1's runs, the proposed method uses the Elias α coding, also called unary coding. Algorithm 2 shows how the Elias α coding encodes an input integer value, and Table 4.1 explains conversions from integer values to codewords. Fig. 4.4 (a) shows that 94% of whole 1's run in the difference data is less or equal to four. The Elias α coding is suitable to encode small values because this coding can encode

Table 4.2: Codeword table of exponential Golomb coding

	$C_{EGC}(N)$			
Input	$k = 0$	$k = 1$	...	$k = 3$
0	0	00		0000
1	100	01		0001
2	101	1000		0010
3	11000	1001		0011
4	11001	1010		0100
5	11010	1011		0101
6	11011	110000		0110
7	1110000	110001		0111
⋮	⋮	⋮	⋮	⋮
127	111111100000000	11111100000001		111100000111
⋮	⋮	⋮	⋮	⋮

Algorithm 2: Elias alpha (α) coding

Input: inVal
Output: codeword
1 codeword := '0'*(inVal - 1) & '1'

numeric values from one to three without overhead for encoding comparing with the other entropy coding methods such as Elias gamma coding. Therefore, the Elias α coding encodes the length of 1's run in difference data. For the length of 0's runs, the proposed method uses the EGC. Algorithm 3 shows how the EGC encodes an integer value. 'tmp' and 'dLen' are temporary variables. Table 4.2 shows conversions from integer values to codewords of the EGC.

The distribution of 0's run in the difference data depends on the objects of the input data. In Algorithm 3, the EGC has a parameter k that determines distributions of their code length. Small k value makes small input values convert to short codewords, and large k value makes large input values convert short codewords. Therefore, the EGC can encode the 0's run in the difference data into small data with suitable k. Also, the α-EGC uses an offset value because the smallest length of both 0's runs and 1's runs are not 0 but 1, while the Elias α coding and the EGC begin to encode from 0.

The α-EGC is compared with conventional compression methods to evaluate its compression efficiency. The conventional methods include static Huffman coding (SHC), adaptive Huffman coding (AHC), Elias gamma coding (Gamma) [72], Golomb coding [73], and the original EGC. In this evaluation, CR is used as an indicator of the effectiveness of compression methods, and the value of the CR is

Algorithm 3: Exponential Golomb coding

Input: inVal, k
Output: codeword
1 tmp := inVal + 2^k;
2 dLen := the number of binary digits without leading zeros of (tmp - 1);
3 codeword := '1'*(dLen - k) & '0' & inVal[dLen .. 0];

Table 4.3: Compression ratio

sample	SHC	AHC	Gamma	Golomb (m=7)	EGC (k=2)	Proposed (α-EGC) (k=3)
bicy+car	1.55	1.29	2.17	2.00	2.36	2.48
bicycle	2.79	2.98	5.73	4.03	5.78	6.62
bike	1.60	1.35	2.55	2.13	2.54	2.74
bike+car	2.28	2.06	3.25	3.98	3.98	4.55
car	1.69	1.48	2.77	2.32	2.77	3.01
ped+car	2.07	1.70	3.00	2.82	2.98	3.31
pedestrian	2.46	2.54	4.55	3.44	4.65	5.20
face	2.40	2.34	4.45	3.36	4.46	5.08
average	1.12	2.07	3.68	3.00	3.68	4.16

defined as follows:

$$\mathrm{CR} = \frac{D_r[\mathrm{bit}]}{D_c[\mathrm{bit}]}, \tag{4.1}$$

where D_c is compressed data size. D_r is raw data size, and it is 1024. The larger CR indicates that the target compression method can compress data better and the resulting data size is smaller than the other methods.

In the comparison, the parameter k in the α-EGC is three and the offset is one, which makes CR the highest. This comparison experiment was carried out for 45 sets of sample data, which are classified according to objects and their combination: pedestrians, bicycles, bikes, cars, and a person's face captured at a close range. The experimental results are summarized in Table 4.3. The α-EGC achieved the best CR compared with the other methods. Therefore, the α-EGC can compress the stimuli position data more efficiently than the conventional compression methods.

Table 4.4 shows the relationship between the values of k and CR in α-EGC evaluated with the same samples used in Table 4.3. In this table, the results written in bold indicate the maximum CR compared with CR with other values of k. This result shows that the optimum value of the k in α-EGC depends on the objects in the input data.

Table 4.4: Relevance between values of k and CR in α-EGC

	α-EGC						
sample	k=0	k=1	k=2	k=3	k=4	k=5	k=6
bicy+car	2.45	2.54	**2.57**	2.48	2.35	2.21	1.99
bicycle	5.85	6.20	6.43	6.62	6.78	**6.92**	6.54
bike	2.63	2.76	**2.80**	2.74	2.59	2.47	2.23
bike+car	4.06	4.31	4.48	**4.55**	4.52	4.48	4.15
car	2.86	3.00	**3.06**	3.01	2.88	2.76	2.49
ped+car	3.09	3.25	**3.34**	3.31	3.20	3.10	2.82
pedestrian	4.65	4.92	5.11	**5.20**	5.19	5.18	4.80
face	4.57	4.83	4.98	**5.08**	**5.08**	5.07	4.66
Average	3.77	3.98	4.11	**4.16**	4.13	4.09	3.81

4.4 Implementation of Compression and Decompression

This section explains how the proposed compression method α-EGC is implemented with an ASIP, which has dedicated functional units and an instruction-set of compression and decompression. To enhance the configuration ability of the stimulation, the inner controller circuits of artificial vision systems need to include a processor, and the ASIP can satisfy the demand for miniaturization of the internal components more than the implementation with a processor and dedicated circuits connected via a system bus. A 16-bit reduced instruction set computer (RISC) processor is selected as the base processor of the proposed ASIP, where the instruction-set of the base processor is shown in Table 4.5.

The proposed α-EGC compression procedure is described in Algorithm 4. Both stages of encoding and decoding in the α-EGC occupy a large part of the execution cycles in compression and decompression, respectively, with the base RISC processor. The breakdowns of execution cycles of compression and decompression processes on the base RISC processor are shown in Table 4.6 and Table 4.7, respectively. In the compression process, run-length encoding is mostly dominant and occupies 42% in all the execution cycles, and the α-EGC encoding also occupies 28%. In the decompression process, the α-EGC decoding accounts for approximately half of the whole execution cycles, which occupies 45%, and the run-length decoding occupies 37%. These analyses show that it is highly efficient to implement codecs of the α-EGC and run-length coding to the base RISC processor.

From the results of Tables 4.6 and 4.7, this section designs a dedicated instruction-set for the proposed ASIP and its circuit. The proposed ASIP has the dedicated instruction-set for the run-length coding and the α-EGC coding to reduce execution cycles efficiently, and the proposed ASIP has a codec for these coding meth-

Algorithm 4: Compression procedure

```
   Input: stimuli position data
   Output: encoded bit stream
 1 Initialize buffers;
 2 repeat
 3     Load stimuli position data to buffer register;
 4     Generate difference data from the previous frame;
 5     repeat
 6         if curBit is different from preBit then
 7             if curBit == '0' then Exponential Golomb coding ;
 8             else α-coding ;
 9             Store encoded data to memory;
10             curBit := preBit;
11         end
12         else
13             Run length coding;
14         end
15     until the tail of memory;
16 until the tail of buffer register;
```

ods to reduce execution cycles of bit operation, while the base RISC processor needs several instructions. The dedicated instruction-set is listed in Table 4.8. The proposed ASIP also has special purpose registers to save the number of working general-purpose register (GPR) and reduce memory access to the data memory, which needs several extra cycles due to waiting for loading or storing the data. The special purpose registers in Table 4.9 are introduced.

Table 4.5: Instruction set of base RISC processor

Operation class	Operations
Arithmetic Ops.	ADD, SUB, SLA, SRA
Logical Ops.	AND, OR, XOR, NOT, SLL, SRL
Comparison Ops.	SLT, SLTU, SEQ, SNEQ
Immediate Ops.	ADDI, SLLI, SRLI, LHI, LLI
Bit Ops.	SETB, CLRB, TSTB
Load/Store Ops.	LHB, LLB, SHB, SLB, LD, ST
Branch Ops.	BRZ, BRNZ
Jump Ops.	JP, JPL, JPR, JPRL
Interrupt Ops.	TRAP, RETI
Special Ops.	NOP, SLEEP

Table 4.6: Breakdown of execution cycles of compression on base RISC processor

Process	Execution cycles	Percentage [%]
1. Initialization & End process	24	0.06
2. Load & Generate difference data	9 338	25.23
3. Run length encoding	15 679	42.36
4. α-EGC encoding	10 448	28.23
5. Store codeword	272	0.73
All	37 016	100.00

Table 4.7: Breakdown of execution cycles of decompression on base RISC processor

Process	Execution cycles	Percentage [%]
1. Initialization & End process	26	0.03
2. Load encoded data	32	0.39
3. α-EGC decoding	3 694	45.33
4. Run length decoding	3 031	37.19
5. Restore & Store decoded data	1 364	16.74
All	8 149	100.00

Table 4.8: Dedicated instructions

Instruction	Used in Compression	Used in Decompression
EGINIT(EGc INITialize)	Y	Y
RLEN(Run Length ENcoding)	Y	-
ALEN(Alpha ENcoding)	Y	-
EGENC(EGc ENCode)	Y	-
PACK(PACKing)	Y	-
STDATA(STore encoded DATA)	Y	-
EGFLASH(EGc FLASH)	Y	-
LDDATA(LoaD encoded DATA)	-	Y
ALDEC(ALpha DECoding)	-	Y
EGDEC(EGc DECoding)	-	Y
UNPACK(UNPACKing)	-	Y
INITRLD (INITialize Run Length Decoding)	-	Y
RLDEC(Run Length DECoding)	-	Y
LDRL(LoaD Run Length)	-	Y

Table 4.9: Special purpose registers for codecs

name	width [bit]	description
bufpnt	6	index where data remain in codebuf
codebuf	48	buffer register for encode data
codelen	8	code length of codeword from codec
codewd	33	codeword from codec
membuf	48	codeword snippets in one word
	(16x3)	(use in encoding only)
param	16	parameter for coding
		(including k for EGC and offset)
memacc	2	control signal for the times of memory access
rlpntr	4	pointer of tail of decoded run length data
rlbuf	16	buffer register for decoded run length data

Compression

The compression procedure is divided roughly into five stages:

- Initialization and end process stage,
- loading and generating difference data stage,
- encoding with run-length coding stage,
- encoding with α-EGC stage, and
- storing codeword stage.

In the initialization stage, instruction EGCINT initializes the special purpose registers; clears the bufpnt register, the codebuf register, the memacc register, and the codewd register, and sets the k of EGC and offset of coding to the param register. In the stage of the run-length decoding, instruction RLDEC and LDRL are used. Instruction RLDEC expands the numerical run in one of the operands to the bit stream to the rlbuf register. After execution, instruction RLDEC subtracts the value that is equal to the length of the bit stream instruction which expands from the numerical run. If the rlbuf register is full, instruction RLDEC writes the status to the GPR indicated by an operand, and then instruction LDRL is used to load the bit stream in the rlbuf register to the GPR indicated by an operand. In the stage of the α-EGC encoding, instruction ALEN and EGENC encode the value of the run to each codeword individually, and the codeword and its code length are stored in the codewd register and the codelen register, respectively. Instruction PACK is used at every step after execution of ALEN or EGENC. This instruction loads the codeword data from the codewd register and stores the data to the tail of data in the codebuf register referring the bufpnt register, and then the instruction also updates the bupnt register, the memacc register, and the membuf register. In the stage of storing the codeword, instruction STDATA stores the encoded data in the codebuf register to the data memory. This instruction refers the memacc register that stores the number of times for storing the data in the membuf register. When the STDATA instruction is carried out, the data in the membuf register is stored in 16 bits, and the membuf register is shifted to MSB and the value of the memacc register is subtracted by one. If the value of the memacc register is zero, the STDATA instruction is not carried out, which is equal to the NOP instruction. In the stage of the end process, the instruction EGFLASH stores the remains of the codewords in the codebuf register to the data memory.

Decompression

The decompression procedure is divided roughly into five stages.

- Initialization and end process stage,

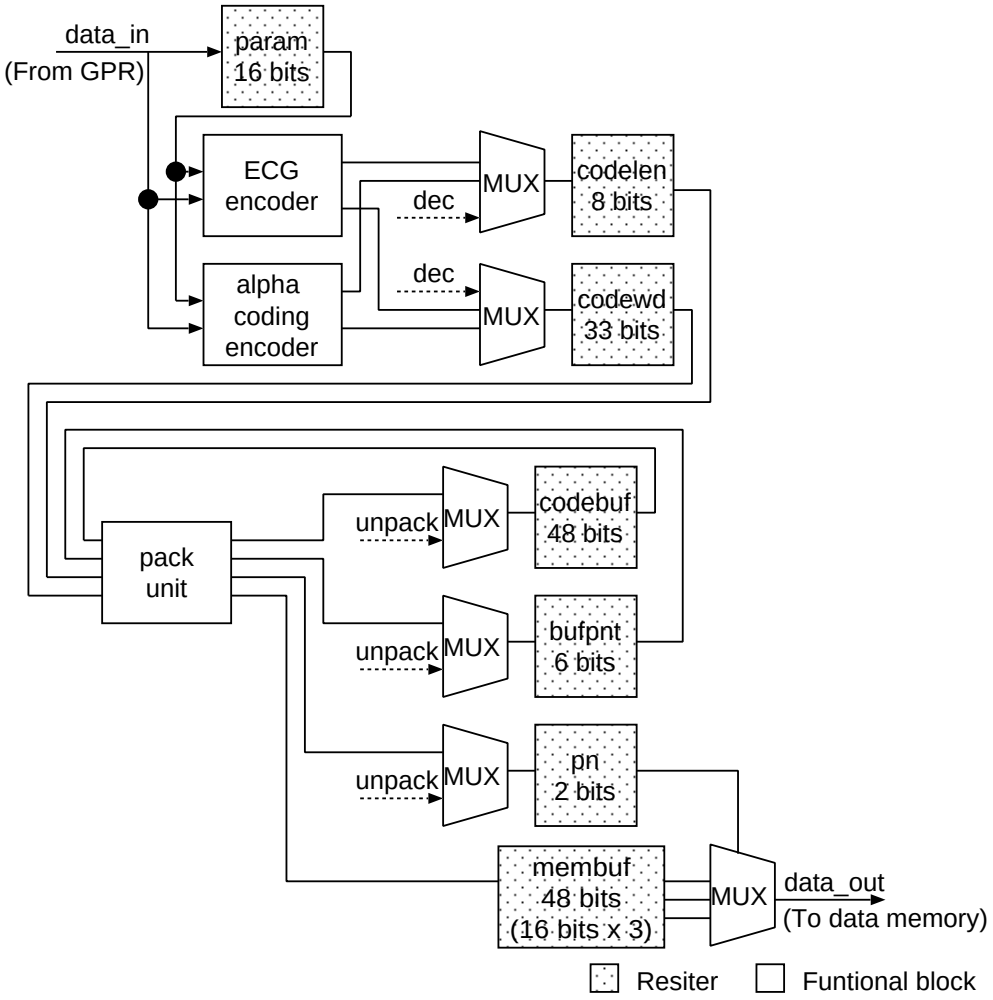


Figure 4.5: Circuits and registers dedicated for compression.

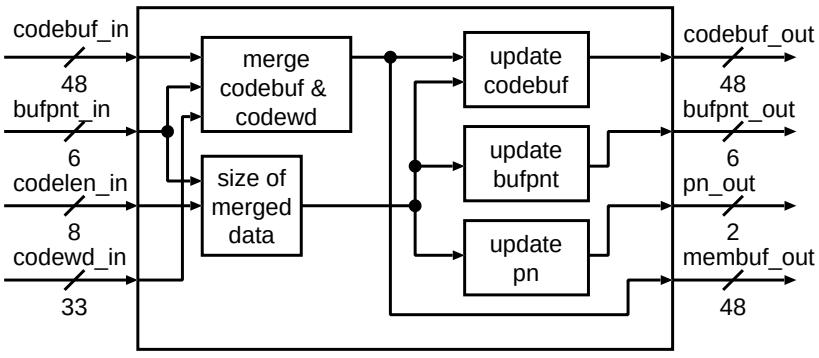


Figure 4.6: Circuit for packing codebuf register and updating bufpnt register, memacc register, and membuf register in encoding.

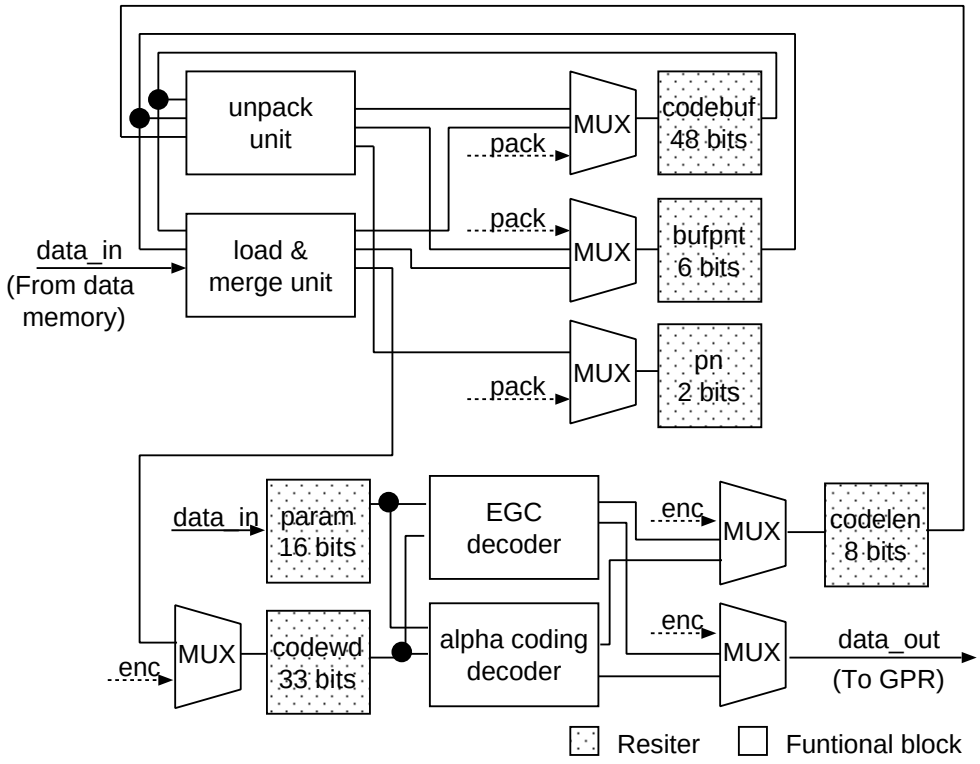


Figure 4.7: Circuits and registers dedicated for decompression.

- loading codeword stage,
- decoding with α-EGC stage,
- decoding with run-length coding stage, and
- restoring difference data and storing decoded data stage.

The dedicated instruction-set is used in the initialization stage, loading, and decoding the codeword. In the initialization, instruction EGCINT is also used as in the compression, and the instruction INITRLD is used to clear registers in the run-length decoder. In the stage of loading the codeword, instruction LDDATA loads encoded data to the codebuf register from the data memory, which requires the address of the data memory as an operand. In the stage of α-EGC decoding, instruction ALDEC and EGDEC decode the codeword in the codebuf register to the numerical value of the run of bit stream, and then store the data to the GPR indicated as an operand. After decoding by ALDEC or EGDEC, instruction UNPACK is used and loads the encoded data from the codebuf register and stores one codeword to the codewd register in parallel with decoding the header of the codeword to get the code length of the codeword.

In the stage of the run-length decoding, instruction RLDEC and LDRL are used. Instruction RLDEC expands the numerical run in one of the operands to

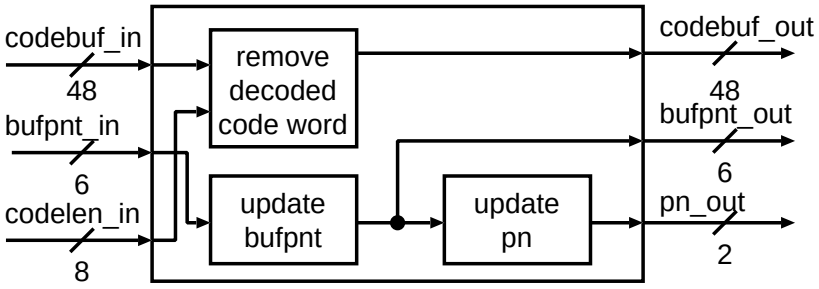


Figure 4.8: Circuit for detaching decoded codeword from codebuf register and updating bufpnt register and memacc register in decoding.

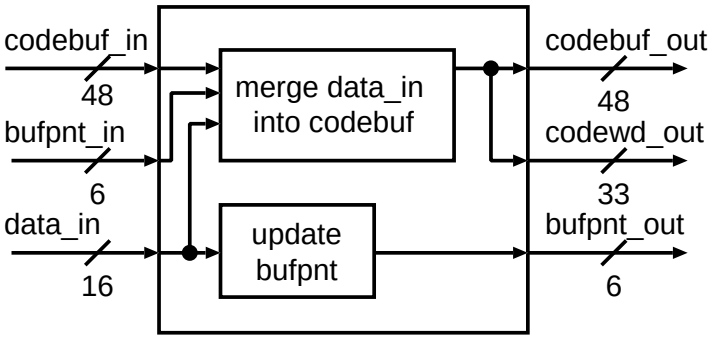


Figure 4.9: Circuit for merging data from data memory to codebuf register and updating bufpnt register in decoding.

bit stream to the rlbuf register. After execution, instruction RLDEC subtracts the value that is equal to the length of the bit stream instruction which expands from the numerical run. If the rlbuf register is full, instruction RLDEC writes the status to the GPR indicated by an operand, and then instruction LDRL is used to load the bit stream in the rlbuf register to the GPR indicated by an operand.

Figure 4.5 illustrates components realizing the dedicated instruction-set for compression, and Fig. 4.6 illustrates the block diagram of 'pack unit' in Fig. 4.5. In Fig. 4.5, 'data_in' indicates input from one of the GPR indicated by an operand, and 'data_out' indicates output to the access unit of the data memory in the base RISC processor. Fig. 4.7 also illustrates components realizing the dedicated instruction-set for decompression, and Fig. 4.8 and Fig. 4.9 illustrate the block diagrams of 'unpack unit' and 'load & merge unit', respectively. In Fig. 4.7, 'data_in' indicates output from access unit of the data memory, and 'data_out' indicates input to the one of the GPR. In both Fig. 4.5 and Fig. 4.7, shadowed boxes indicate registers that are initialized by instruction ECGINIT.

Table 4.10: Comparison of execution cycles in compression

Process	Base	Proposed	Diff. [%]
1. Initialization & End process	24	47	+95.83
2. Load & Generate difference data	9 338	1 282	-86.27
3. Run length encoding	15 679	2 148	-86.30
4. α-EGC encoding	10 448	1 326	-87.30
5. Store codeword	272	272	±0.00
All	37 016	5 075	-86.30

Table 4.11: Comparison of execution cycles in decompression

Process	Base	Proposed	Diff. [%]
1. Initialize & End process	26	215	+826.92
2. Load encoded data	32	234	+731.25
3. α-EGC decoding	3 694	64	-98.27
4. Run length decoding	3 031	1 372	-54.73
5. Restore & Store decoded data	1 364	792	-41.94
All	8 149	2 678	-67.14

4.5 Evaluation

This section explains the comparison of the proposed ASIP and the base processor regarding the following points: area, execution cycles, power consumption, and energy consumption. The area and power consumption based on the switching probabilities of each gate were estimated using Design Compiler from Synopsys, Inc., with the TSMC 0.18 μm typical library. The execution cycles and the switching probabilities of each gate were based on the simulation results from ModelSim by Mentor Graphics Corp. The energy consumption E is defined as follows.

$$E = P\frac{N}{f},$$

where P is power [W], N is the number of execution cycles, and f is operation frequency [Hz].

The comparisons of execution cycles for compression and decompression with the proposed processor and the base RISC processor are summarized in Table 4.10 and Table 4.11, respectively. For compression, the proposed processor reduced the number of execution cycles by 86% and 87% for run-length encoding and the α-EGC encoding, respectively, and reduced the total number of execution cycles by 86%. For decompression, the proposed processor reduced the number of execution cycles by 98% and 54% for the α-EGC decoding and run-length decoding. As shown in Table 4.12, the power consumption of the proposed processor decreased

Table 4.12: Comparison with base processor and proposed processor

		Base	Proposed	Diff. [%]
Area [μm^2]		53 102	122 764	+131.19
Max Freq. [MHz]		303	222	-26.70
Power [μW/MHz]	Compression	37.6	33.3	-11.44
	Decompression	41.5	47.6	+14.42
Energy [nJ]	Compression	1 391	169	-87.86
	Decompression	338	128	-62.43

Table 4.13: Characteristics of low-power wireless communications

Specification	[74]	[75]
Standard	BLE	BAN
V_s [V]	2.35–3.3	0.74
P_r [mW]	15.3	3.66
T [kbps]	1 000	971.4
e_r [nJ/bit]	15.3	4.1

by 11% and increased 14% for compression and decompression, respectively, and the proposed processor reduced energy consumption by 87% and 62% for compression and decompression, respectively. Therefore, these results show that the addition of the dedicated instructions is useful for reducing both processing time and energy consumption.

The evaluation results also show that the proposed ASIP can perform decompression regarding real-time processing. It can decompress data in 303 μs per one frame at 10 MHz. Compared with the base RISC processor, the proposed processor reduced execution times in decompression, which can achieve the equivalent performance of the base RISC processor at lower operation frequency.

4.6 Discussion

From the evaluation results, we estimate a reduction of energy consumption in wireless communication thanks to the proposed implementation. Some examples of low-power wireless communication implementation and their parameters are listed in Table 4.13 (see [76] to find more details). In Table 4.13, V_s is the supply voltage, P_r is power consumption of receivers in activation time, T is data rate in wireless communication, and e_r is energy consumption per one bit in receiving time. As communication standards, Bluetooth Low Energy (BLE) [74] and body area network (BAN) [75], which is also called IEEE 802.15.6 standard, are well-known low-power communication standards, and BLE has been already used for

Table 4.14: Comparison of estimated energy consumption in receiver

	w/o compression	Base RISC	Proposed ASIP
n [bit]	1 024	246.15	246.15
E_{d} [nJ]	0	339	108
E_{r}(BLE [74]) [nJ]	15 667	4 104	3 894
(Diff. [%])	(± 0.00)	(-73.81)	(-75.15)
E_{r}(BAN [75]) [nJ]	4 198	1 347	1 137
(Diff. [%])	(± 0.00)	(-67.92)	(-72.92)

Table 4.15: Comparison with proposed compression method and other methods in energy consumption

	w/o compression	Gamma	EGC	Proposed
P_{d} [μW/MHz]	0	46.42	46.52	47.62
E_{d} [nJ]	0	117	100	128
n [bit]	1 024	278.26	278.26	246.15
E_{r} (BLE [74]) [nJ]	15 667	4 374	4 358	3 894
(Diff. [%])	(± 0.00)	(-72.08)	(-72.19)	(-75.15)
E_{r} (BAN [75]) [nJ]	4 198	1 258	1 241	1 137
(Diff. [%])	(± 0.00)	(-70.04)	(-70.44)	(-72.93)

various purposes. The energy consumption of receiving one stimuli position data in wireless communication E_{r} is defined as follows:

$$E_{\mathrm{r}} = ne_{\mathrm{r}} + E_{\mathrm{d}},$$

where n [bit] is the amount of received traffic data per one frame of stimuli position data and E_{d} [nJ] is the energy consumption in compression processing. Estimation results are summarized in Table 4.14 that compares the proposed implementation with the base RISC processor and Table 4.15 that compares the proposed implementation of the α-EGC and other simpler methods: Elias gamma coding (represented as Gamma in this table) and EGC. These tables show that the α-EGC can reduce the energy consumption of a wireless communication receiver more significantly than the other methods. Regarding the receiver of wireless communication, the proposed α-EGC reduced the energy consumption by 11 % and 10 % compared with Elias gamma coding and EGC in BLE, respectively, and by 9.6 % and 8.4 % compared with Elias gamma coding and EGC in Body area network (BAN), respectively. These experimental results show that the increment of the energy consumption by adding data compression processing is negligible for both standards. Regarding the visual prostheses, this method will contribute significantly to the energy reduction of the internal body component because the visual prostheses require a large number of stimulation sites and high frequent

updating for practical pseudo vision. Therefore, the proposed compression and its implementation will help providing the neural stimulation device, especially for the visual prostheses, with low energy consumption design, which will lead these devices to miniaturization and improvement of the QoL.

4.7 Conclusion

This chapter proposed a low computational data compression architecture for neural stimulation towards the visual prosthesis. The proposed method, called the α-EGC, aimed to reduce energy consumption in wireless communication of neural stimulation systems. The proposed method took advantage of bias in target data, stimulation position data of visual cortex, and designed to compress the target data in low computational amount. Evaluation results show that the proposed method reduced the target data size by 77%. This chapter also proposed a hardware implementation of the α-EGC as an ASIP. Simulation results indicate that the proposed implementation reduced energy consumption by 87% and 62% in compression and decompression, respectively, compared with an implementation with a base RISC processor. The proposed compression. From these investigations, the proposed compression method and implementation will contribute to energy reduction in wireless communication of the neural stimulation devices, which lead to sufficiently practical use.

Chapter 5

Low Energy Architecture of Neural Stimulation Controller

Chapter 4 discussed a low computational data compression method to reduce energy consumption in the wireless communication system for the neural prosthesis systems and its implementation as an application specific instruction-set processor (ASIP). This chapter describes a design of neural stimulator architecture including the ASIP as a stimulation controller. The neural stimulator is designed as a system-on-a-chip (SoC). First, requirements for neural stimulation devices to realize practical sensory perception are explained, and this thesis focuses on how to stimulate the target nerves "strictly" by a dedicated hardware and proposes the design of a stimulation controller. Finally, an SoC architecture for a controller of the internal body component, which includes the proposed ASIP as a microprocessor in chapter 4, is proposed and evaluated.

5.1 Motivation and Objective

Electrical neuromodulation treatments have been expected to be effective for drug-resistant neural diseases such as epilepsy, and some neural stimulation devices are available on the market. Approximately 30% of epileptic patients remain either drug-resistant or remain limited adverse effect [77]. Although the neural stimulation devices have possibilities for the contribution of medicine, many challenges remain toward improved safety and quality of life (QoL). Regarding both clinical and research use, the neural stimulation devices require the followings: Flexibility for stimulation strategies, microseconds order temporal resolution, and low energy design.

Deciding appropriate parameters of stimulation, such as strength, duration, phase, and timing, before using neural stimulation devices is extremely difficult because their proper values depend on individual difference. For fixing unexpected behavior and avoiding undesirable effects of stimulation, neural stimulation devices require flexibility in stimulation in both fields of experimental and

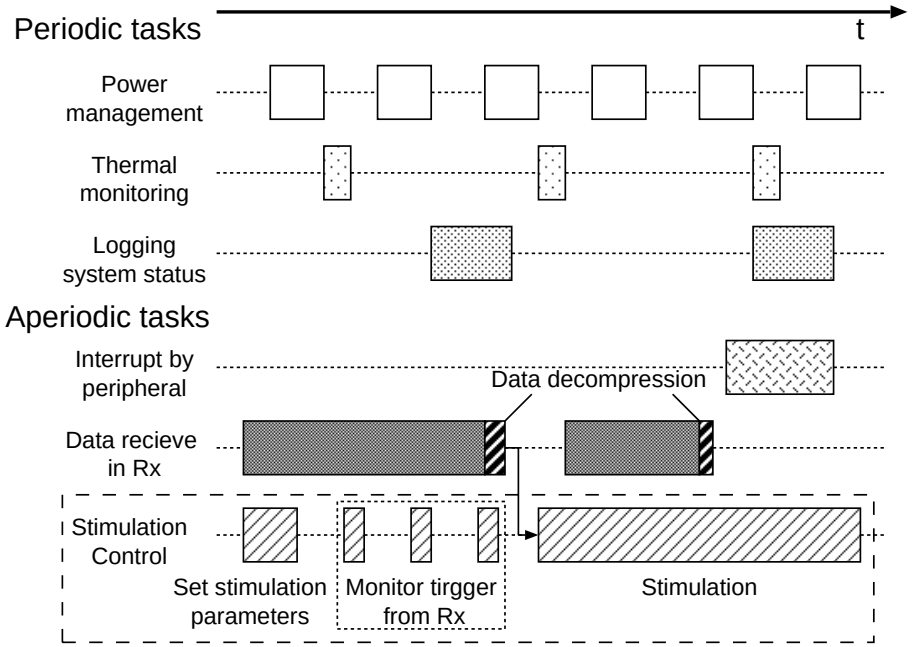


Figure 5.1: Task management of internal body microcontroller.

clinical. To satisfy this requirement, a possible solution is that a component for generating stimulation separates from that for controlling stimulation in architecture level. For ensuring high flexibility of stimulation strategies, the stimulation controller includes more than or equal to one processor, which can adopt a new stimulation strategy by loading new data to referring memories. The processors can perform calculations *in situ*; they may calculate the stimulation parameters by using values from peripheral sensors as closed-loop if there are appropriate control algorithms.

The second requirement, microseconds order for stimulation, is relatively complicated to satisfy. Owe to the notable improvement of the embedded systems, commercial microcontrollers designed for low energy operation has enough performance to control stimulation in the microseconds order if they are devoted to only stimulation control. However, there are many tasks to manage the internal body component as depicted in Fig. 5.1. As shown in this figure, what the microcontrollers have to deal with is classified into two types: Periodic tasks and aperiodic tasks. The periodic tasks contain power management, thermal monitoring caused by power dissipation of the device, logging of system status, and so on. The microcontrollers commonly process these task by interrupts from timers in the SoC, which period is programmed in advance. In contrast, the aperiodic tasks including the stimulation task contained communication from the external SoC modules via wireless such as Bluetooth and ultra wide band (UWB) and wired such as serial-parallel interface (SPI) and universal asynchronous receiver/-transmitter (UART). The difficulty of the stimulation task management is the re-

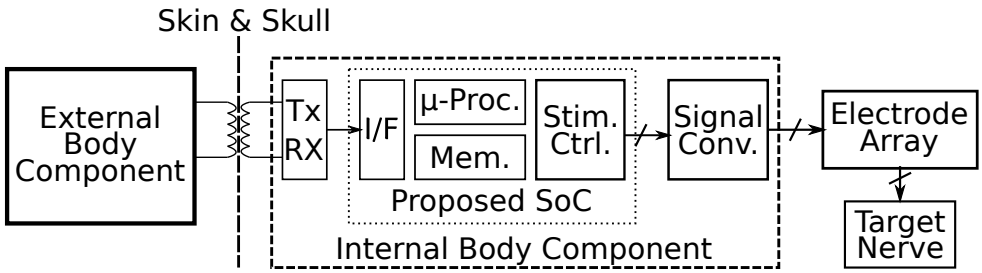


Figure 5.2: Overview of the target neural stimulation device and proposed SoC.

quirements for high temporal resolution and precision. In general, the processors carry out instructions in the instruction memory sequentially. However, they cannot process strictly in time because of pipeline-hazards and external or internal interrupt by peripheral modules such as timers. The stimulation tasks occupy the processing of the microcontroller for a long time to satisfy these requirements, and hence the microcontroller cannot process the other tasks. For ensuring high temporal resolution and precision, the stimulation controller needs to include a circuit dedicated to controlling stimulation strictly in microseconds resolution, which is processing stimulus in parallel the processors.

A space for implanting the neural stimulation devices and the amount of wireless power supply are limited. Low energy design of them enables to reduce the volume of a secondary battery and extend their operation time. The low energy design also expands the longevity of the systems, which can increase their QoL by reducing the number of additional surgeries for device replacement. To reduce the energy consumption, this thesis applies the data compression method discussed in chapter 4, which can reduce the stimuli position data for the visual prosthesis by 77% in average. In the visual prostheses, the neural stimulation devices require both high-frequency and high-configurable stimulation devices because the visual prostheses devices have to complement the gap between the ideal visual field in the external image processor and reported visual recognition by the user. Therefore, the proposed SoC will contribute to the construction of practically visual recognition for the visual prosthesis devices.

This study proposes a high-flexibility, high temporal resolution, and low energy architecture of neural stimulator. The ASIP allows the proposed SoC to change stimulation strategies with only rewriting data in memories, and the dedicated control signal generator enables the proposed SoC to stimulate the target nerves in microseconds order in parallel to the other tasks processed by the processors.

5.2 Architecture Overview

In this section, an overview of the proposed SoC architecture is introduced. The proposed SoC is designed for flexibility regarding stimulation strategies, and then

it applies common-used protocols for communication other modules in the internal body component. Also, the proposed SoC has an on-chip programmer (OCP) connected to a discrete flash memory, which is convenient for replacement of ASIP program.

Figure 5.2 illustrates an overview of a target neural stimulation device that includes the proposed SoC. The target neural stimulation device consists of an external body component and an internal body component. The two components communicate via wireless communication for neural stimulation and recorded data. This study focuses on the internal body component, and it consists of a wireless communication module, the proposed SoC, and a signal converter for stimuli generator. The proposed SoC is connected to the signal converter and the wireless communication module on the board, and the signal converter is connected to an electrodes array for stimulation. The electrodes array penetrated into the target nerve, which is the visual cortex in this study, and it has more than hundreds of electrodes to realize the desired phenomenon in high-spatial resolution. The signal converter was presented in [78] by Hayashida *et al*.as a prototype composed by discrete components and designed by Kameda (ref. [79]), which has high flexibility for stimulation parameters such as stimuli position (64 channels), amplitude (7-bit precision: ± 100, ± 200, ± 300, and ± 400 μ A/phase), and so on. To bring out its performance, the proposed SoC includes the stimulation controller to manipulate the stimuli in microsecond order. Here, the combination of the proposed SoC and the signal converter is evaluated in this thesis.

SoC Interface

For the convenience of development, the internal stimulation controller has to have two types of communication interfaces for existing components: Inter-device and inter-chip. Figure 5.3 illustrates the interface between the proposed SoC architecture and peripheral components. The proposed SoC adopts UART for the inter-device communication. UART is easy to implement and commonly used for asynchronous communication with a computer. The computer can read and write internal the SoC status for debugging use via UART. For the interaction with the external body component, the proposed SoC applies discrete wireless communication SoC and communicates with it via the inter-chip communication.

In this thesis, the inter-chip communication is classified into two types: For the discrete chips and the signal converter. The former is used for the communication between the proposed SoC and the discrete SoC for peripheral modules including the wireless communication and sensors. The communication for the discrete chips require usability, and then the proposed SoC adopts SPI protocol, which is commonly used for inter-chip communication. The SPI protocol enables the proposed SoC to communicate with plural modules through a single interface, which indicates that the proposed SoC has expandability for the whole design of peripheral modules.

The communication with the signal converter is divided into two parts: data

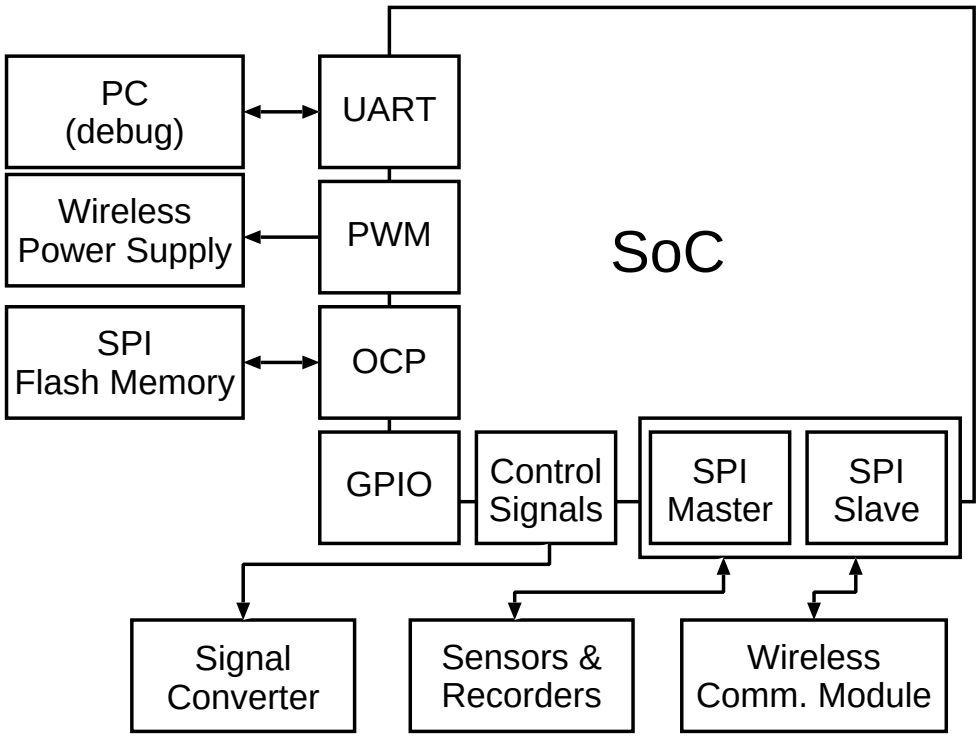


Figure 5.3: Block diagram of proposed SoC interface.

transmission and control stimulation. To configure amplitude and position of stimulation, the proposed SoC communicates with the signal converter via SPI protocol. Because control of stimulation requires high temporal resolution, the proposed SoC has output ports for individual signals for the input port of the signal converter, and the detail is explained after.

Components in SoC

Fig. 5.4 depicts the structure of the stimulation controller. The proposed stimulation controller consists of the following components: an on-chip programmer, an external SPI flash memory, a 16-bit microprocessor, an instruction memory, a data memory, and a control signal generator for the signal converter. The microprocessor connects the instruction memory directly, and the data memory, the internal wireless communication unit, the control signal generator are connected via a system bus. In this study, the ASIP proposed in chapter 3 is utilized as a microprocessor of the SoC, and, for the convenience of reconfiguration, the flash SPI memory stores the compiled binary data.

Then, the following sentences explain how to perform the stimulation controller from reset to generating stimulation. When the inside module is reset, the on-chip programmer loads the compiled binary data in the external flash SPI memory and stores them to the instruction memory and the data memory. After storing data

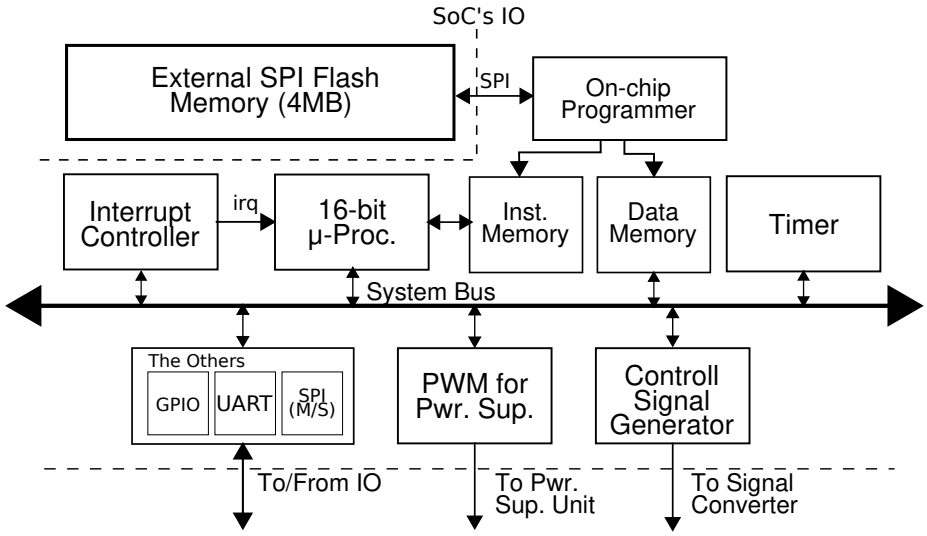


Figure 5.4: Block diagram of inside body component.

to these memories, the on-chip programmer sends the reset signal to the microprocessor, and then the microprocessor starts initialization of peripheral modules. When the internal communication unit receives the stimulation information, the microprocessor stores the stimulation data for the control signal generator in registers.

Control Signal Generator

Figure 5.5 illustrates the details of the control signal generator and its interface. For simplicity of programming, the microprocessor controls the stimulation by

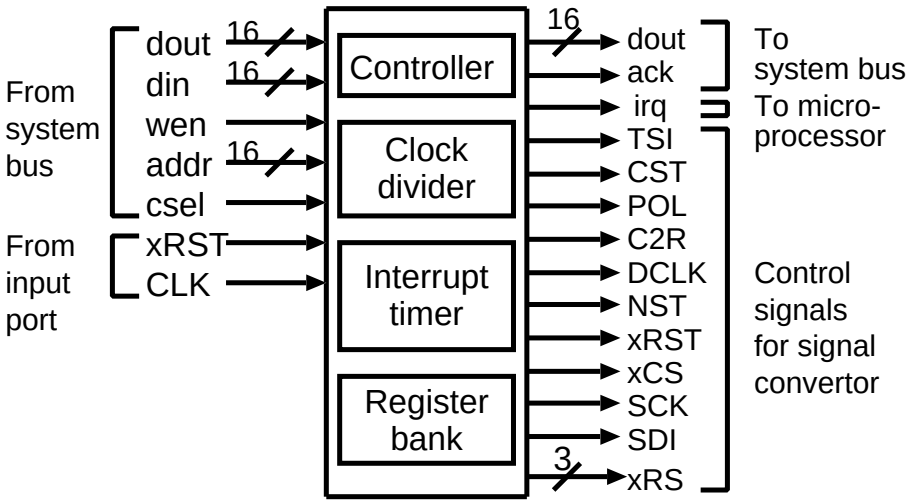
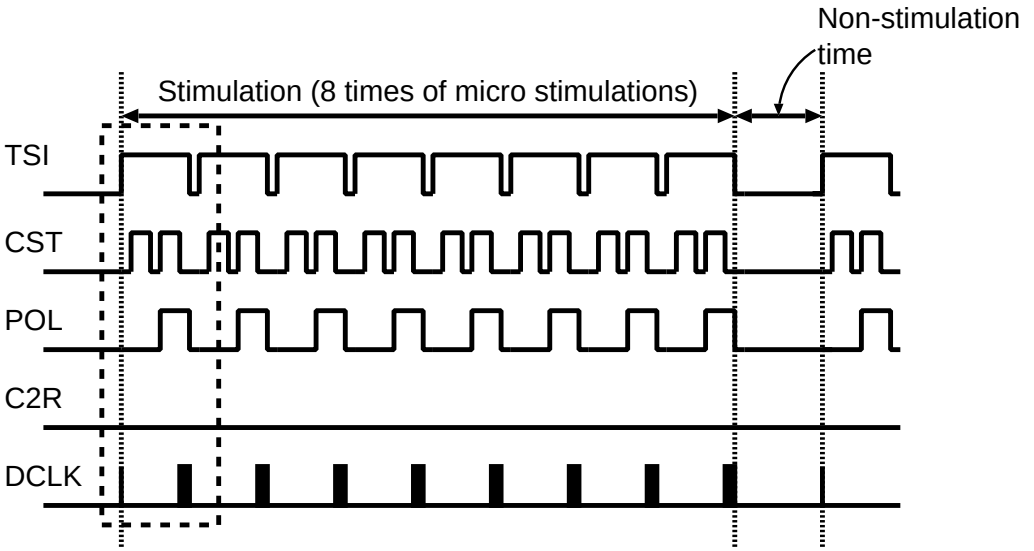


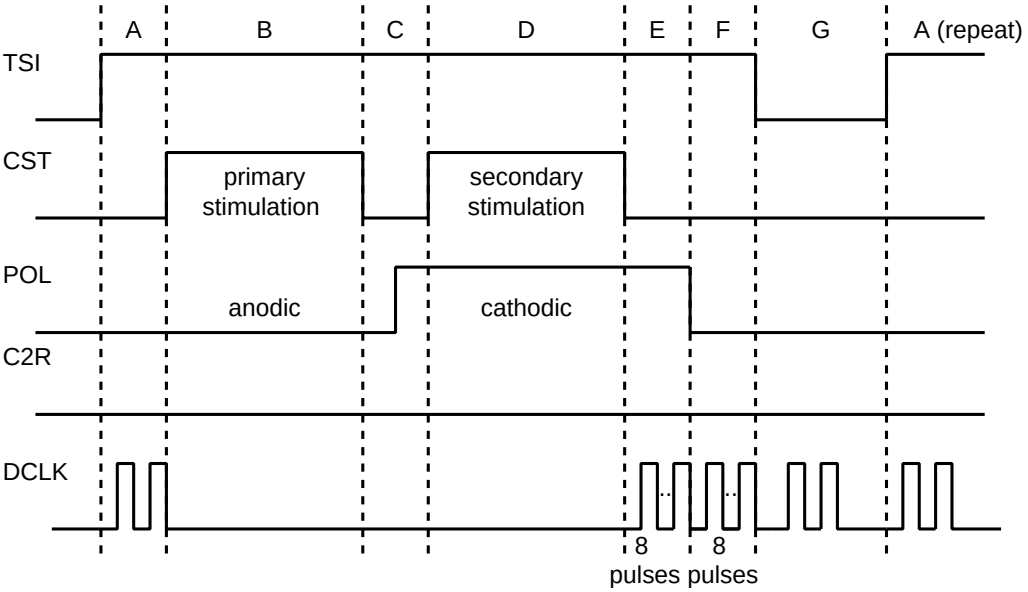
Figure 5.5: Control signal generator.

sending several kinds of commands to the control signal generator, which generates control signals for the signal converter independently from the microprocessor.

The following describes details of each control signal listed in Table 5.1. When signals TSI and CST are set to high, the signal converter stimulates the target nerve referring to the configuration values in registers in the signal converter, and



(a) Eight sets of microstimulation and non-stimulation interval time



(b) One sequence of microstimulation including anodic primary stimulation and cathodic secondary stimulation.

Figure 5.6: Timing chart of control signals for the signal converter.

Table 5.1: Control signals for signal converter

Signal	Description
TSI	Duration of micro stimulation
CST	Input of micro stimulation
POL	Polarity of micro stimulation
C2R	Connection to reference of GND
DCLK	Control signal for digital block in signal converter
NST	Duration of non-stimulation time
xRST	Reset signal of signal converter
xCS	Enable signal of index input of electrode array
SCK	Clock signal for serial communication
SDI	Data input for serial communication
xRS	Selective signal for parameter registers in signal converter

it stops stimulating when either signal is set to low. When signal POL is set to low or high, the polarity of stimulation is set to cathodic or anodic, respectively. In standard physiological experiments, a biphasic current pulse stimulation is used to avoid charge accumulation in tissue, so that the signal converter is designed to generate a microstimulation that consists of a pair of anodic and cathodic stimulation. When the signal C2R is high, the output of the signal converter connects to the ground. The signal DCLK has two roles: as a state transition signal for the signal converter and as an adjuster for the duration of secondary stimulation that is utilized to release the electric charge from the tissue because of the first stimulation. The stimulation has to have a non-stimulation time to avoid inflammation. When the signal NST is high, the signal converter stops generating stimulation. Figure 5.6 (a) shows a timing chart of an example of stimulation train, whose repetition of anodic-first microstimulation is eight.

A set of stimulation is divided into two sections: sequences of microstimulation and non-stimulation time, and a sequence of microstimulation consists of three parts: primary stimulation, secondary stimulation, and an interval of stimulation. Figure 5.6 (b) depicts a timing chart for the output signals from the stimulation generator. In this figure, sections of B, D, and G indicate the primary stimulation, the secondary stimulation, and the interval of stimulation, respectively. To start or repeat the microstimulations, two pulses are sent via DCLK. To stop the microstimulation, sixteen pulses are sent via DCLK, whose rising edge of the first pulse need to synchronize falling edge of CST to adjust the duration of secondary stimulation in the signal converter. One pulse of CST needs to be anodic, and the other needs to be cathodic to realize biphasic stimulation. Table 5.2 summarizes specifications of the proposed stimulation controller.

The signal converter consists of the registers for stimulation parameters, and the control signal generator configures them via the serial communication before or

Table 5.2: Specification of proposed stimulation controller

Specification	Value
Resolution of stimulation sites	4 096
Resolution of micro stimulation	1 [μs]
Number of repetition of micro stimulation	1 – 255
Duration of primary stimulation	1 – 8192 [μs]
Duration of secondary stimulation	1 – 8192 [μs]
Interval of micro stimulation	1 – 8192 [μs]
Non-stimulation time	1 – 8192 [μs]
Polarity of stimulation	Bipolar
Size of instruction memory	16 [kB]
Size of data memory	16 [kB]

during stimulation. After a start of stimulation, the control signal generator runs continuously even while setting the stimulation parameters by the microprocessor. The signal xRST is used to reset signal of the signal converter. The signals SCK and SDI are used for serial communication to the signal converter, and the signal xCS and xRS are used as selective signals. When the xCS is low, serial bit data on the SDI indicates an index of the electrode array. When one of the signals in the xRS is low, the data on the SDI indicate the parameters for the microstimulation of the electrode array directed by the index.

5.3 Evaluation

This section explains evaluation experiments with an FPGA to validate the output wave signal of the proposed stimulation controller, and it also describes a simulation result for the proposed SoC architecture. The proposed stimulation controller was implemented using Quartus Prime by Altera Corp., and the target board was Altera Cyclone IV E in DE2-115 Development and Education Board. In the experiments, the logic analyzer Logic Pro 16 by Saleae, Inc. was used. Design Compiler from Synopsys, Inc. with a GLOBALFOUNDRIES 0.18μm CMOS library was used to estimate the area and power dissipation.

5.3.1 FPGA Implementation

Figure 5.7 depicts the setup of the experimentations. An SPI flash memory was connected to the FPGA, and the logic analyzer connected with the GPIO of the FPGA. In this evaluation, internal SoC status was logged by a desktop computer via UART protocol. Stimulation parameters were set and changed by the replacement of program in the SPI flash memory.

The compilation results for the FPGA are summarized in Table 5.3. Figures 5.8 (a) and (b) show waveforms recorded by the logic analyzer on signals

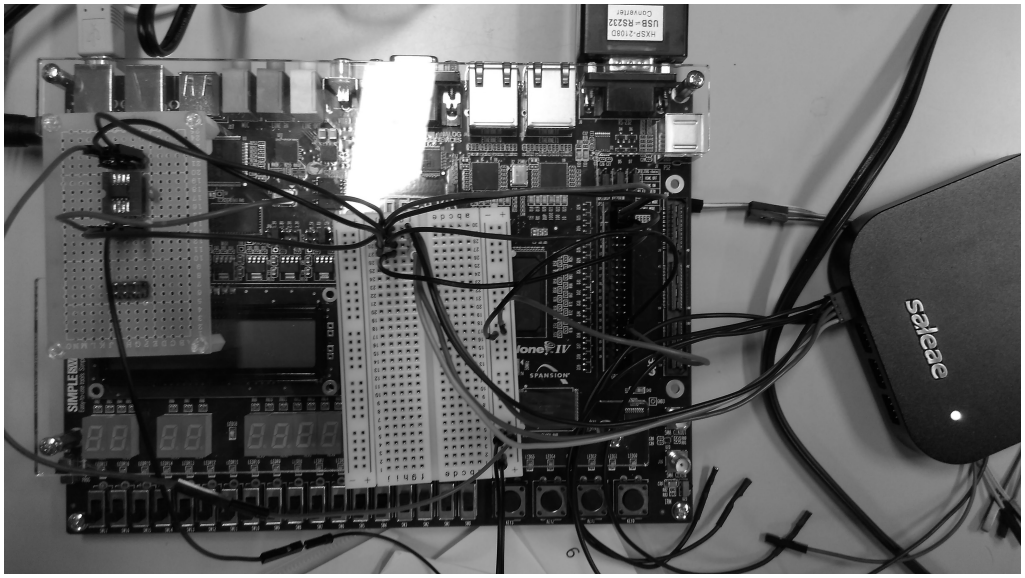


Figure 5.7: Photograph of the experimental setup for functional validation with FPGA.

Table 5.3: Results of Compilation for FPGA

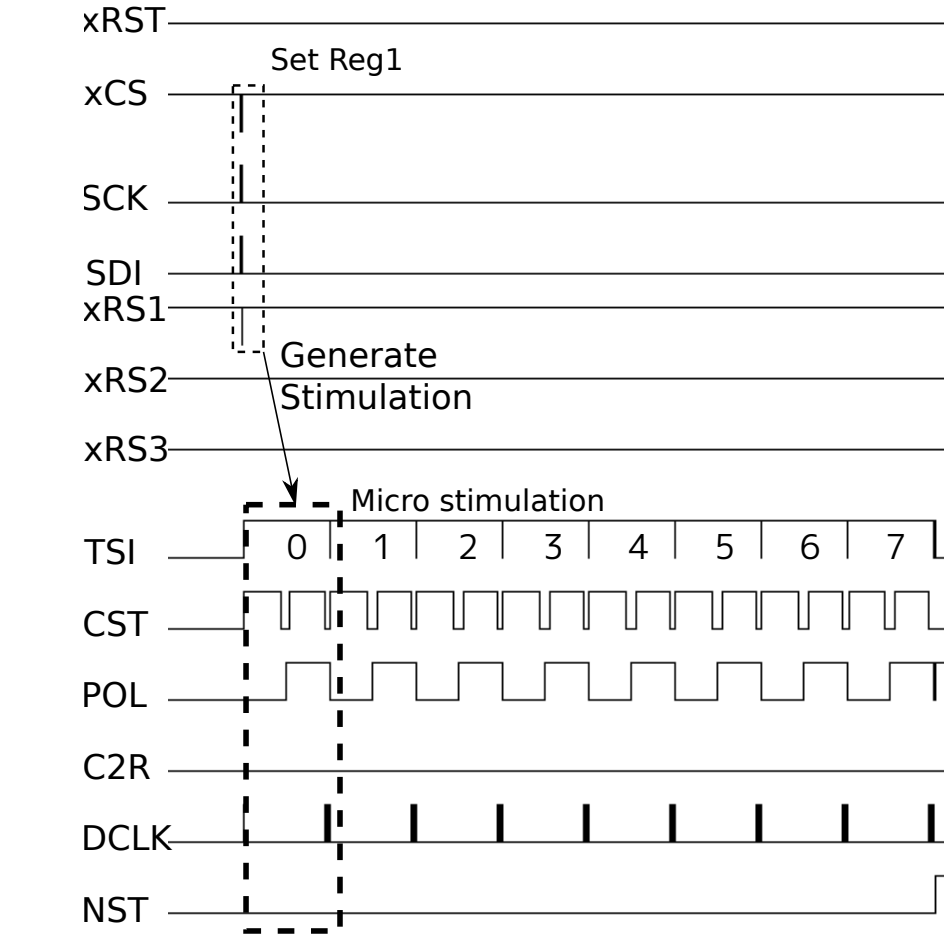
	All	Control signal generator
# of logic elements	14 873	4 209 (28.3%)
# of memory bits [bit]	262 144	0

of the FPGA, and Fig. 5.8 (a) indicates that the FPGA repeated stimulation in eight times as setting. Figure 5.9 shows waveforms for sending the pair of data composed of the index of electrode arrays and the settings of stimulation to each register in the signal converter as specified. Figures 5.9 (a), (b), and (c) illustrate the waveforms of transmission that sends the setting of amplitude, order, and position of the stimulation, respectively.

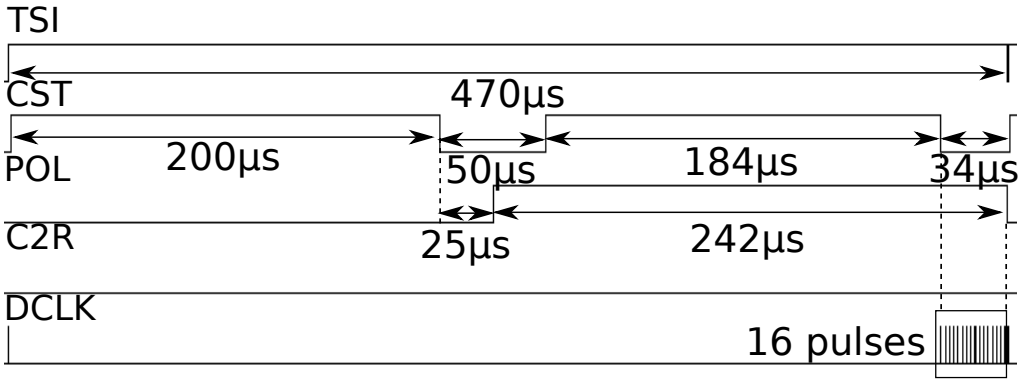
Next, experiments changed the stimulation parameters: Repetition, polarity,

Table 5.4: Results of Compilation for SoC

	All	Control signal generator	ASIP	OCP
Area [μm^2]	56 692 (100.0%)	22 734 (40.1%)	21 680 (38.2%)	1 372 (2.4%)
Power [mW]	1.7 (100.0%)	8.0×10^{-1} (47.1%)	4.0×10^{-1} (23.5%)	4.1×10^{-2} (47.1%)

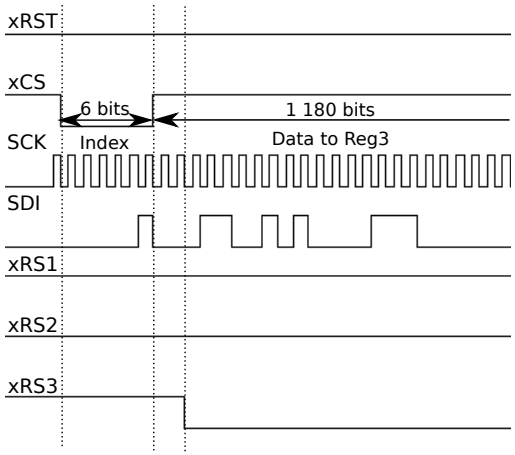


(a) Control signals in setting position of stimulation and eight sets of microstimulation

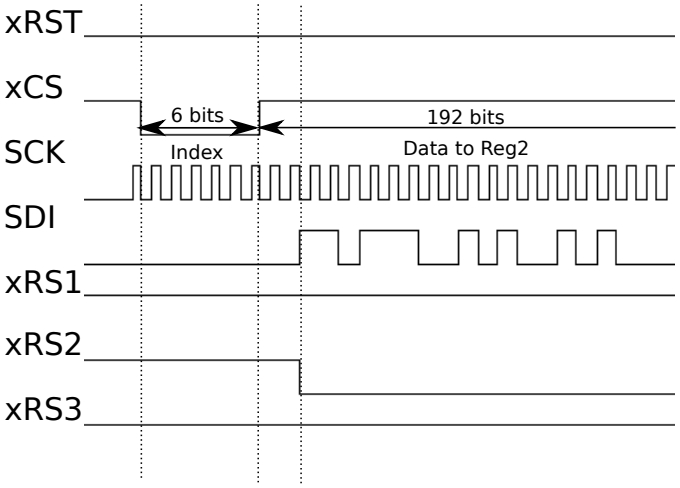


(b) One sequence of microstimulation.

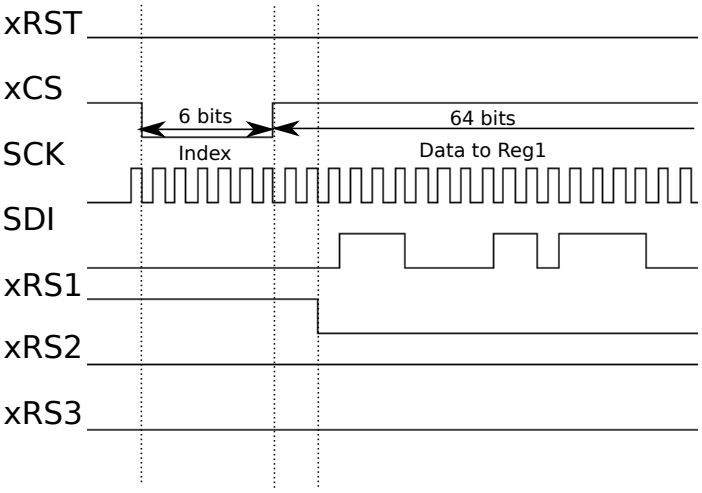
Figure 5.8: Waveforms recorded by logic analyzer from FPGA.



(a) Setting amplitude

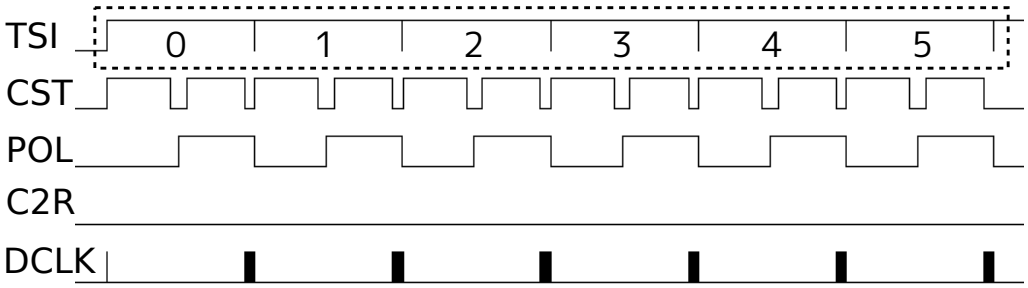


(b) Setting order of stimulation

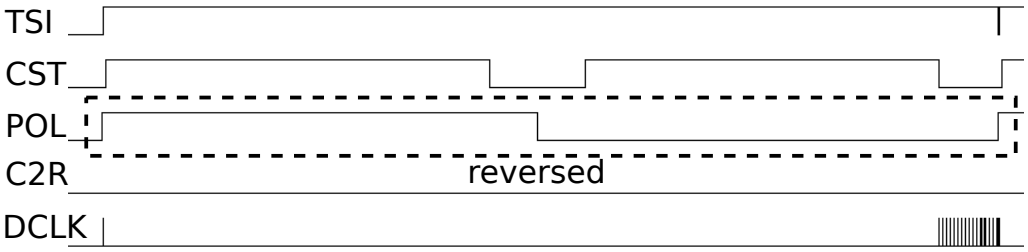


(c) Setting position of stimulation

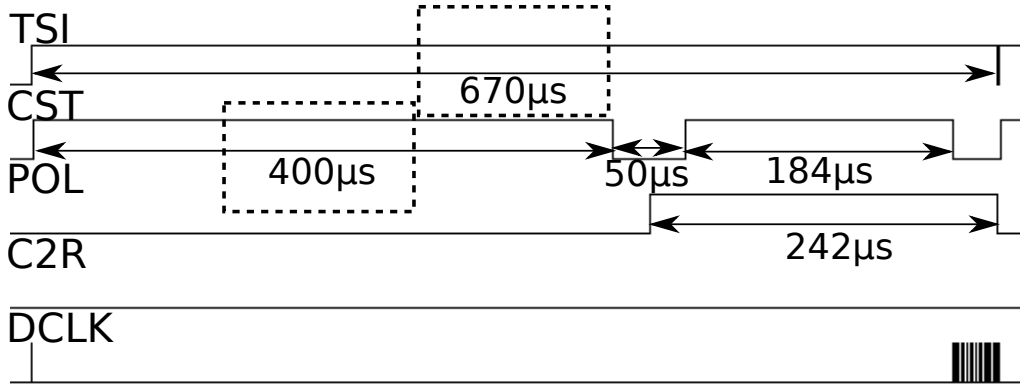
Figure 5.9: Waveform recorded by logic analyzer at the start of setting registers.



(a) Changing repetition of microstimulation to six-time



(b) Changing polarity of stimulation



(c) Changing duration of primary stimulation to 400 μs

Figure 5.10: Waveforms modified configuration from Fig. 5.8 (b).

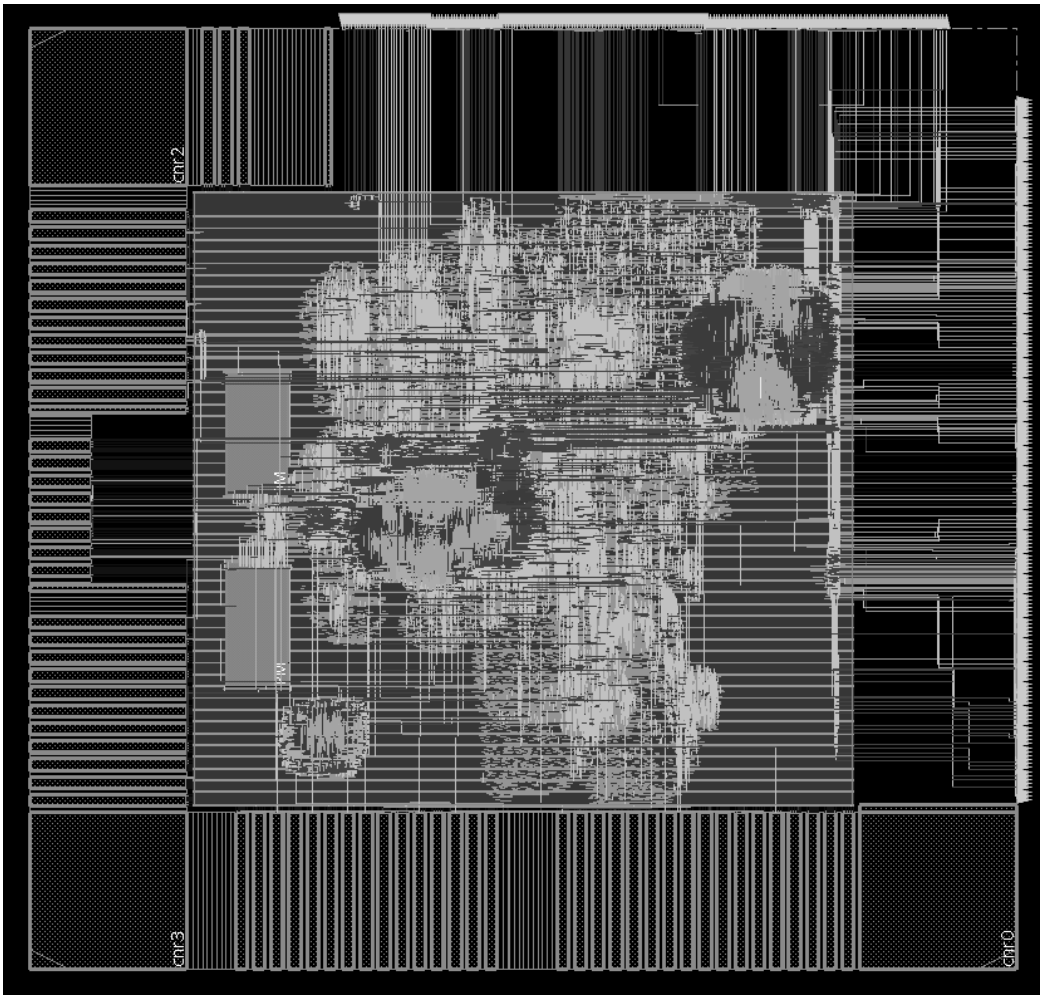


Figure 5.11: SoC layout design.

and duration. Fig. 5.10 (a) shows that a modification that the number of microstimulations is set from eight, which is depicted in Fig. 5.8 (a), to six was reflected correctly. Figures 5.10 (b) and (c) also show the modifications were reflected correctly in polarity, and duration of stimulation, respectively. These results demonstrate that the proposed stimulation controller has high flexibility for the stimulation parameters and the stimulation strategies.

5.3.2 SoC Evaluation

For evaluating the performance in case that the proposed controller is implemented as an SoC, this thesis carried out logical synthesis and designed SoC layout. The logical synthesis results are summarized in of Table 5.4. The design results show that the proposed control signal generator occupies a larger area and consumes an enormous amount of power than the ASIP.

Also, Figure 5.11 depicts a SoC layout screenshot from a placement and routing

Table 5.5: Specification of proposed neural stimulation device set

Specification	Proposed device set
Die size [cm^2]	2.1×3.0 (SoC), 3.5×3.5 (Stim. Chip)
Technology	GF 45 nm (SoC) 0.28 μm (Stim. Chip)
Supply Voltage [V]	1.0, 1.8, 3.3 (SoC) $\pm$ 2.5 (Stim. Chip)
Power supply	Inductive coupling
Data transmission	BLE
Data rate [bps]	2 M
# stimulation channels	64
Stimulation current range [A]	100 μ – 2 m
Stimulation amplitude resolution [bit]	7
Stimulation polarity	Bipolar
Stimulation frequency [Hz]	$\leq$ 2M
Stimulation pulse width range [μs]	1–8192
Area of digital core [μm^2]	56 692
Power consumption of digital core [mW]	1.724

tool. Due to a large number of IO cells, die size of the proposed SoC was 2.1 cm $\times$ 3.0 cm. Finally, Table 5.5 summarizes specifications of the proposed SoC, which includes the specifications of the signal converter proposed in [79]. From these results, the proposed SoC achieved a simultaneous pursuit of high temporal resolution and high flexibility by the combination of an ASIP and a circuit dedicated to stimulation.

5.4 Discussion

This section discusses the flexibility of the proposed device for stimuli parameter by comparing other studies. The evaluation results show that the proposed device has the microsecond-order temporal resolution (up to 2 MHz) and 64-channel stimulation sites. The comparison of characteristics of stimuli parameter is listed in Table 5.6. The number of the stimuli channels and the stimuli amplitude resolution are comparable to the state-of-the-art architectures. The flexibility of the stimuli frequency and stimuli duration, also known as pulse width range, is better than that of the other studies. Regarding power consumption of digital core of the proposed device exceeds that of other studies due to the adoption of the microprocessor.

In this thesis, the proposed neural stimulation device includes the stimulation controller and the microprocessor. Owe to the stimulation controller, the pro-

Table 5.6: Comparison with the conventional neural stimulator

Specification	Proposed	[44]	[46]	[47]	[48]
# stim. channels	64	8	N/A	64	24
Stim. current range [A]	100 μ – 2 m	82.5 μ – 229 μ	N/A	10 μ – 1 m	10 μ – 1 m
Stim. amplitude resolution [bit]	7	5	6	8	8
Stim. frequency [Hz]	$\leq$ 2M	60 – 200	20 – 60	N/A	N/A
Stim. pulse width range [μs]	1–8192	40 – 440	2 100	N/A	N/A
Power consumption of digital core [mW]	1.724	0.365	9.0×10^{-2}	1.4–1.5	0.897

posed device has enough performance to realize a simultaneous pursuit of high-temporal resolution and high-flexible stimulation. This will result in sufficient visual recognition by the visual prosthesis devices. Regarding closed-loop adjustment of stimuli parameters, the microprocessor will play an important role in future to calculate accommodate parameters *in situ* in the skull. This contributes not only to neuroscience and electrophysiology as an experimental prototype of neural stimulation device but also to neural stimulator in clinical use. Concern about heat dissipation by IMD [9] and energy consumption by stimulation [42, 43] will be a big issue in clinical scene. Hence, the microprocessor will be helpful to manage heat and energy in the neural stimulation devices to scavenge kinds of information from sensors and calculate accommodated stimuli parameters for the individual brain. Also, the external memory connected with the proposed SoC makes easy to change stimulation and calibration methodologies by rewriting programs. Therefore, the research results of the proposed device will lead to better QoL to the users of the neural stimulation devices.

5.5 Conclusion

This chapter proposed highly flexible and spatial stimulation controller. For high flexibility to stimulation strategies, this chapter applied the ASIP proposed in chapter 4 as the stimulation controller. Processors including ASIP cannot guarantee microseconds order temporal resolution because of external interrupts from peripheral modules, and hence this chapter designed the dedicated circuit to control stimulation in high temporal resolution. Experimental results showed that the proposed SoC architecture could control stimulations in microseconds order with FPGA implementation. The proposed device has comparable potentials with the state-of-the-art neural stimulation devices, and it will contribute to the alleviation of concerns about heat dissipation and energy consumption by adjusting

the stimuli spatio-temporal parameters for the variation of the individual brain *in situ*.

Chapter 6

Conclusion and Future Work

This chapter concludes the discussion in this thesis and describes future work.

6.1 Conclusion

Passionate researches have been continued in life science domain, and then the knowledge and experience about neurostimulation treatment have been constructed as one clinical application of results of these research. This thesis contributed to the miniaturization and the extended longevity of the neural stimulation devices due to low energy design, which leads the improvement of the quality of life for their users. Also, a closed-loop control of neural stimulations has been required in clinical use to avoid excess stimuli and alleviate physical and mental burden by stimulation adjustment. This thesis discussed how to improve the biomedical signal processing in the measurement part, which provides the user with high precision treatment. The difficulty in the architecture with a microprocessor is how to reduce energy consumption and shrink processing time in the microprocessor. This thesis tackled these problems to improve the energy efficiency by adding dedicated circuits and dedicated instruction-set to the microprocessor.

This thesis described a low power architecture of neural stimulation system aimed at closed-loop control of stimulation. To realize closed-loop stimulation, chapter 3 proposed a high throughput and small hardware implementation for phase synchrony analysis of neural signals. The proposed method achieved mathematically identical yet low computational calculation without trigonometrical function whose hardware implementation cost is significantly high. For increasing throughputs and inhibiting hardware costs, the proposed method used inner products of input neural signals and reused calculation results. Evaluation results show that the proposed method reduced cycle latency by 62 % and improved hardware implementation efficiency, defined as "throughput per gate count," by 5.3 times.

Chapter 4 proposed a low computational data compression architecture for neural stimulation towards the visual prosthesis. The proposed method, called α exponential Golomb coding (α-EGC), aimed to reduce energy consumption in wireless

communication of neural stimulation systems. The proposed method took advantage of statistically biased information in target data, stimulation position data of visual cortex, and designed to compress the target data in low computational amount. Evaluation results showed that the proposed method reduced the size of the target data by 77 %. Chapter 4 also proposed a hardware implementation of the α-EGC as an application specific instruction-set processor (ASIP). Simulation results demonstrated the proposed implementation reduced energy consumption by 87 % and 62 % in compression and decompression, respectively, compared with an implementation with a base reduced instruction set computer processor. Finally, the proposed architecture reduced energy consumption for transmission per one flame of the target data in wireless communication by 75 % and 73 % with Bluetooth low energy and body area network standard, respectively, compared with the architecture without compression scheme.

Chapter 5 proposed a low energy architecture of neural stimulation controller as a system-on-a-chip. The neural stimulation requires low energy consumption, highly temporal stimulation in micro-second order, and high flexibility for stimulation parameters, such as amplitude, duration, polarity, frequency, and repetition. For satisfying this trade-off, the proposed architecture consists of the ASIP proposed in chapter 4 and highly flexible and precise stimulation controller implemented as a dedicated circuit. When a stimulation strategy needs modifications, the proposed architecture could execute another stimulation strategy with only changing programs for the ASIP in the memory, and the ASIP was able to reduce energy consumption in wireless communication as mentioned above. The stimulation controller processed stimulation by sending commands from the ASIP via system-bus, and it could handle stimulation in parallel with the ASIP. This thesis investigated the performance of the neural device with the microprocessor targeting the closed-loop adjustment of stimulation in future, and the proposed SoC has better performance in the spatiotemporal resolution of the stimulation compared with other state-of-the-art work.

6.2 Future Work

This section describes future work of this research focusing on the realization of ultra-low-energy neural stimulators for clinical use and practical visual prosthesis system.

This research described energy efficient implementation of neural stimulator toward visual prosthesis in chapters 4 and 5. One of the limitations of this study is to deal with only stimulation position data of visual prosthesis as the target input data. Therefore, there is room for examining whether the proposed implementation can reduce energy consumption for other kinds of methodologies of neural prosthesis. They adapt utterly different modeling to generate stimulation data compared with the target visual prosthesis while they use two-dimensional stimulation array and similar stimulation parameters. For evaluation of energy

consumption, this research used the wireless communication standard that is used widely in commercials. At the research level, some wireless communication module, for example, one used in [46], achieved less energy consumption and much data transmission in short distance. Regarding energy consumption, more detailed evaluation of trade-off between overhead of data compression and the decrease in wireless communication remains as future work.

While visual prosthesis devices for clinical use such as Argus II [39] and Alpha–IMS [32] have been released, the realization of practical visual prosthesis device has many challenges. Regarding resolution of vision recognition emitted by the visual prosthesis devices, quality of vision for object recognition is not good enough. While the signal processing of this research is limited to analyze phase synchrony, which can predict malfunction of brain activity exposed as seizures of epilepsy, investigation of adjustment methods for stimulation parameters like ocular adaptation and their hardware implementation regarding signal processing remain future work.

Bibliography

[1] T. Sugiura, J. Yu, and Y. Takeuchi, "Hardware-oriented algorithm for phase synchronization analysis of biomedical signals," presented at the 13th IEEE Biomedical Circuits and Systems Conference (BioCAS), Turin, Italy, Oct. 2017, pp. 416–419.

[2] T. Sugiura, M. Imai, J. Yu, and Y. Takeuchi, "A low-energy application specific instruction-set processor towards a low-computational lossless compression method for stimuli position data of artificial vision systems," *Journal of Information Processing*, vol. 25, pp. 210–219, 2017.

[3] T. Sugiura, A. U. Khan, J. Yu, *et al.*, "A programmable controller for spatio-temporal pattern stimulation of cortical visual prosthesis," presented at the 12th IEEE Biomedical Circuits and Systems Conference (BioCAS), Shanghai, China, 2016, pp. 432–435.

[4] J. Enderle and J. Bronzino, *Introduction to biomedical engineering*, 3rd. Elsevier Science, 2011.

[5] O. Aquilina, "A brief history of cardiac pacing," *Images Paediatr Cardiol*, vol. 8, no. 2, pp. 17–81, 2006.

[6] *Micra information for healthcare professionals*, Medtronic Inc., May 2017. [Online]. Available: `http://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/pacemakers/micra-pacing-system.html`.

[7] G. E. Moore, "Cramming more components onto integerated circuits," *Electronics*, vol. 38, no. 8, Apr. 1965.

[8] G. E. Moore, "Progress in digital integrated electronics [technical literaiture, copyright 1975 IEEE. reprinted, with permission. technical digest. international electron devices meeting, IEEE, 1975, pp. 11–13.]," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 36–37, Sep. 2006. DOI: `10.1109/N-SSC.2006.4804410`.

[9] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher, "Thermal impact of an active 3-D microelectrode array implanted in the brain," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 4, pp. 493–501, Dec. 2007. DOI: `10.1109/TNSRE.2007.908429`.

[10] M. M. de Oliveira, P. Wen, and T. Ahfock, "Heat transfer due to electroconvulsive therapy: Influence of anisotropic thermal and electrical skull conductivity," *Computer Methods and Programs in Biomedicine*, vol. 133, no. Supplement C, pp. 71–81, 2016. DOI: 10.1016/j.cmpb.2016.05.022.

[11] N. L. Opie, A. N. Burkitt, H. Meffin, and D. B. Grayden, "Heating of the eye by a retinal prosthesis: Modeling, cadaver and in vivo study," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 339–345, Feb. 2012. DOI: 10.1109/TBME.2011.2171961.

[12] J. Wang, H. Xie, T. Chung, L. L. H. Chan, and S. W. Pang, "Neural probes with integrated temperature sensors for monitoring retina and brain implantation and stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 9, pp. 1663–1673, Sep. 2017. DOI: 10.1109/TNSRE.2016.2634584.

[13] C. Li, P.-M. Wu, Z. Wu, *et al.*, "Brain temperature measurement: A study of *in vitro* accuracy and stability of smart catheter temperature sensors," *Biomedical Microdevices*, vol. 14, no. 1, pp. 109–118, Feb. 2012. DOI: 10.1007/s10544-011-9589-4.

[14] C. N. Heck, D. King-Stephens, A. D. Massey, *et al.*, "Two-year seizure reduction in adults with medically intractable partial onset epilepsy treated with responsive neurostimulation: Final results of the RNS system pivotal trial," *Epilepsia*, vol. 55, no. 3, pp. 432–441, 2014. DOI: 10.1111/epi.12534.

[15] National Institute on Deafness and Other Communication Disorders (NIDCD), *Cochlear implants*, Mar. 2017. [Online]. Available: https://www.nidcd.nih.gov/health/cochlear-implants.

[16] A. Djourno and C. Eyriès, "[auditory prosthesis by means of a distant electrical stimulation of the sensory nerve with the use of an indwelt coiling] (in French)," *La Presse mèdicale*, vol. 65, no. 63, p. 1417, 1957.

[17] M. D. Eisen, "Djourno, Eyries, and the first implanted electrical neural stimulator to restore hearing," *Otology & Neurotology*, vol. 24, no. 3, pp. 500–506, 2003.

[18] M. A and M. M, "The early history of the cochlear implant: A retrospective," *JAMA Otolaryngology-Head & Neck Surgery*, vol. 139, no. 5, pp. 446–453, 2013. DOI: 10.1001/jamaoto.2013.293.

[19] F. B. Simmons, "Permanent intracochlear electrodes in cats, tissue tolerance and cochlear microphonics.," *The Laryngoscope*, vol. 77, no. 2, pp. 171–186, 1967. DOI: 10.1288/00005537-196702000-00003.

[20] R. P. Michelson, "The results of electrical stimulation of the cochlea in human sensory deafness," *Annals of Otology, Rhinology & Laryngology*, vol. 80, no. 6, pp. 914–919, Dec. 1971. DOI: 10.1177/000348947108000618.

[21] M. M. Merzenich, R. P. Michelson, C. R. Pettit, R. A. Schindler, and M. Reid, "Neural encoding of sound sensation evoked by electrical stimulation of the acoustic nerve," *Annals of Otology, Rhinology & Laryngology*, vol. 82, no. 4, pp. 486–503, 1973.

[22] M. M. Merzenich, D. N. Schindler, and M. W. White, "Symposium on cochlear implants. II. feasibility of multichannel scala tympani stimulation," *The Laryngoscope*, vol. 84, no. 11, pp. 1887–1893, 1974. DOI: 10.1002/lary.5540841105.

[23] R. C. Bilger and F. O. Black, "Auditory prostheses in perspective," *Annals of Otology, Rhinology & Laryngology*, vol. 86, no. 3-supplemental, pp. 3–10, May 1977. DOI: 10.1177/00034894770860s301.

[24] B. S. Wilson and M. F. Dorman, "Cochlear implants: A remarkable past and a brilliant future," *Hearing Research*, vol. 242, no. 1-2, pp. 3–21, Aug. 2008. DOI: 10.1016/j.heares.2008.06.005.

[25] W. H. Oraganization. (Oct. 2017). Media centre fact sheets: Vision impairment and blindeness.

[26] ——, (Nov. 2017). Media centre fact sheets: Diabetes.

[27] S. Klauke, M. Goertz, S. Rein, *et al.*, "Stimulation with a wireless intraocular epiretinal implant elicits visual percepts in blind humans," *Investigative Opthalmology & Visual Science*, vol. 52, no. 1, p. 449, Jan. 2011. DOI: 10.1167/iovs.09-4410.

[28] M. Keserü, M. Feucht, N. Bornfeld, *et al.*, "Acute electrical stimulation of the human retina with an epiretinal electrode array," *Acta Ophthalmologica*, vol. 90, no. 1, e1–e8, Nov. 2011. DOI: 10.1111/j.1755-3768.2011.02288.x.

[29] A. E. Hadjinicolaou, R. T. Leung, D. J. Garrett, *et al.*, "Electrical stimulation of retinal ganglion cells with diamond and the development of an all diamond retinal prosthesis," *Biomaterials*, vol. 33, no. 24, pp. 5812–5820, Aug. 2012. DOI: 10.1016/j.biomaterials.2012.04.063.

[30] E. Zrenner, K. U. Bartz-Schmidt, H. Benav, *et al.*, "Subretinal electronic chips allow blind patients to read letters and combine them to words," *Proceedings of the Royal Society B: Biological Sciences*, vol. 278, no. 1711, pp. 1489–1497, Nov. 2010. DOI: 10.1098/rspb.2010.1747.

[31] K. Mathieson, J. Loudin, G. Goetz, *et al.*, "Photovoltaic retinal prosthesis with high pixel density," *Nature Photonics*, vol. 6, no. 6, pp. 391–397, May 2012. DOI: 10.1038/nphoton.2012.104.

[32] K. Stingl, K. U. Bartz-Schmidt, D. Besch, *et al.*, "Subretinal visual implant Alpha IMS - clinical trial interim report," *Vision Research*, vol. 111, pp. 149–160, Part B 2015. DOI: 10.1016/j.visres.2015.03.001.

[33] N. Toshihiko, S. Kiyotaka, T. Takashi, *et al.*, "Performance improvement and functionalization of an electrode array for retinal prosthesis by iridium oxide coating and introduction of smart-wiring technology using CMOS microchips," *Sensors and Actuators A: Physical*, vol. 211, pp. 27–37, 2014. DOI: http://dx.doi.org/10.1016/j.sna.2014.03.001.

[34] B. W. Tatler, N. J. Wade, H. Kwan, J. M. Findlay, and B. M. Velichkovsky, "Yarbus, eye movements, and vision," *Iperception*, vol. 1, no. 1, pp. 7–27, Jul. 2010. DOI: 10.1068/i0382.

[35] J. Button, "Electronics brings light to the blind," *Radio Electron.*, vol. 29, pp. 53–55, 1958.

[36] M. Ghovanloo and K. Najafi, "A modular 32-site wireless neural stimulation microsystem," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2457–2466, Dec. 2004. DOI: 10.1109/JSSC.2004.837026.

[37] H. Li, T. Han, J. Wang, *et al.*, "A real-time image optimization strategy based on global saliency detection for artificial retinal prostheses," *Information Sciences*, vol. 415-416, pp. 1–18, Supplement C 2017. DOI: doi.org/10.1016/j.ins.2017.06.014.

[38] W. I. Al-Atabany, T. Tong, and P. A. Degenaar, "Improved content aware scene retargeting for retinitis pigmentosa patients," *BioMedical Engineering OnLine*, vol. 9, no. 1, p. 52, Sep. 2010. DOI: 10.1186/1475-925X-9-52.

[39] Y. H.-L. Luo and L. da Cruz, "The Argus II retinal prosthesis system," *Progress in Retinal and Eye Research*, vol. 50, pp. 89–107, Supplement C 2016. DOI: 10.1016/j.preteyeres.2015.09.003.

[40] M. S. Humayun, E. de Juan, and G. Dagnelie, "The bionic eye: A quarter century of retinal prosthesis research and development," *Ophthalmology*, vol. 123, no. 10, Suppl, S89–S97, 2016. DOI: 10.1016/j.ophtha.2016.06.044.

[41] L. da Cruz, J. D. Dorn, M. S. Humayun, *et al.*, "Five-year safety and performance results from the Argus II retinal prosthesis system clinical trial," *Ophthalmology*, vol. 123, no. 10, pp. 2248–2254, 2016. DOI: 10.1016/j.ophtha.2016.06.049.

[42] S. Little, A. Pogosyan, S. Neal, *et al.*, "Adaptive deep brain stimulation in advanced parkinson disease," *Annals of Neurology*, vol. 74, no. 3, pp. 449–457, 2013. DOI: 10.1002/ana.23951.

[43] H. Wu, H. Ghekiere, D. Beeckmans, *et al.*, "Conceptualization and validation of an open-source closed-loop deep brain stimulation system in rat," *Scientific Reports*, vol. 4, no. 9921, 2015. DOI: 10.1038/srep09921.

[44] Y. P. Lin, C. Y. Yeh, P. Y. Huang, *et al.*, "A battery-less, implantable neuro-electronic interface for studying the mechanisms of deep brain stimulation in rat models," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 98–112, Feb. 2016. DOI: 10.1109/TBCAS.2015.2403282.

[45] W. M. Chen, H. Chiueh, T. J. Chen, *et al.*, "A fully integrated 8-channel closed-loop neural-prosthetic CMOS SoC for real-time epileptic seizure control," vol. 49, no. 1, pp. 232–247, Jan. 2014. DOI: 10.1109/JSSC.2013.2284346.

[46] B. Bozorgzadeh, D. R. Schuweiler, M. J. Bobak, P. A. Garris, and P. Mohseni, "Neurochemostat: A neural interface SoC with integrated chemometrics for closed-loop regulation of brain dopamine," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 654–667, Jun. 2016. DOI: 10.1109/TBCAS.2015.2453791.

[47] K. Abdelhalim, H. M. Jafari, L. Kokarovtseva, J. L. P. Velazquez, and R. Genov, "64-channel UWB wireless neural vector analyzer SOC with a closed-loop phase synchrony-triggered neurostimulator," vol. 48, no. 10, pp. 2494–2510, Oct. 2013. DOI: 10.1109/JSSC.2013.2272952.

[48] H. Kassiri, S. Tonekaboni, M. T. Salam, *et al.*, "Closed-loop neurostimulators: A survey and a seizure-predicting design example for intractable epilepsy treatment," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1026–1040, Oct. 2017. DOI: 10.1109/TBCAS.2017.2694638.

[49] P. J. Uhlhaas and W. Singer, "Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology," *Neuron*, vol. 52, no. 1, pp. 155–168, Oct. 2006. DOI: 10.1016/j.neuron.2006.09.020.

[50] P. J. Uhlhaas, F. Roux, E. Rodriguez, A. Rotarska-Jagiela, and W. Singer, "Neural synchrony and the development of cortical networks," *Trends in Cognitive Sciences*, vol. 14, no. 2, pp. 72–80, Feb. 2010. DOI: 10.1016/j.tics.2009.12.002.

[51] P. Uhlhaas and W. Singer, "Neuronal dynamics and neuropsychiatric disorders: Toward a translational paradigm for dysfunctional large-scale networks," *Neuron*, vol. 75, no. 6, pp. 963–980, Sep. 2012. DOI: 10.1016/j.neuron.2012.09.004.

[52] F. Varela, J.-P. Lachaux, E. Rodriguez, and J. Martinerie, "The brainweb: Phase synchronization and large-scale integration," *Nature reviews neuroscience*, vol. 2, no. 4, pp. 229–239, 2001.

[53] I. Osorio and Y.-C. Lai, "A phase-synchronization and random-matrix based approach to multichannel time-series analysis with application to epilepsy," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 21, no. 3, p. 033 108, 2011. DOI: 10.1063/1.3615642.

[54] K. Takaura, N. Tsuchiya, and N. Fujii, "Frequency-dependent spatiotemporal profiles of visual responses recorded with subdural ECoG electrodes in awake monkeys: Differences between high- and low-frequency activity," *NeuroImage*, vol. 124, pp. 557–572, 2016.

[55] X. Hu, X. Xu, and X. Tian, "Phase synchronization analysis of multi-channel local field potentials based on high-resolution wavelet transform," presented at the 6th International Conference on Natural Computation (ICNC), vol. 1, Yantai, China, Aug. 2010, pp. 476–479. DOI: 10.1109/ICNC.2010.5583144.

[56] J. R. Williamson, D. W. Bliss, D. W. Browne, and J. T. Narayanan, "Seizure prediction using EEG spatiotemporal correlation structure," *Epilepsy & Behavior*, vol. 25, no. 2, pp. 230–238, Oct. 2012. DOI: 10.1016/j.yebeh.2012.07.007.

[57] F. Mormann, K. Lehnertz, P. David, and C. E. Elger, "Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients," *Physica D: Nonlinear Phenomena*, vol. 144, no. 3, pp. 358–369, 2000.

[58] K. Abdelhalim, V. Smolyakov, and R. Genov, "A phase synchronization and magnitude processor VLSI architecture for adaptive neural stimulation," presented at the IEEE 6th Biomedical Circuits and Systems Conference (BioCAS), Paphos, Cyprus, Nov. 2010, pp. 5–8. DOI: 10.1109/BIOCAS.2010.5709557.

[59] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959. DOI: 10.1109/TEC.1959.5222693.

[60] J. B. Romaine and M. Delgado-Restituto, "Hardware friendly algorithm for the calculation of phase synchronization between neural signals," Atlanta, GA, USA, Oct. 2015, pp. 1–4. DOI: 10.1109/BioCAS.2015.7348347.

[61] J. B. Romaine, M. Delgado-Restituto, J. A. Leñero-Bardallo, and Á. Rodríguez-Vázquez, "Integer-based digital processor for the estimation of phase synchronization between neural signals," presented at the 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), Jun. 2016, pp. 1–4. DOI: 10.1109/PRIME.2016.7519520.

[62] S. Dávila-Montero and A. J. Mason, "An *in-situ* phase-preserving data decimation method for high-channel-count wireless μECoG arrays," presented at the 13th IEEE Biomedical Circuits and Systems Conference, Turin, Italy, Oct. 2017, pp. 144–147.

[63] J.-P. Lachaux, E. Rodriguez, J. Martinerie, F. J. Varela, *et al.*, "Measuring phase synchrony in brain signals," *Human brain mapping*, vol. 8, no. 4, pp. 194–208, 1999.

[64] M. L. V. Quyen, J. Foucher, J.-P. Lachaux, *et al.*, "Comparison of hilbert transform and wavelet methods for the analysis of neuronal synchrony," *Journal of Neuroscience Methods*, vol. 111, no. 2, pp. 83–98, Sep. 2001. DOI: 10.1016/s0165-0270(01)00372-7.

[65] M. Rosenblum and J. Kurths, "Analysing synchronization phenomena from bivariate data by means of the Hilbert transform," *Nonlinear analysis of physiological data*, pp. 91–99, 1998.

[66] R. E. Goldshmidt, "Application of division by convergence," Master's thesis, Massachusetts Institute of Technology, 1964.

[67] *Nangate 45nm open cell library*. [Online]. Available: http://www.nangate.com/?page_id=2325.

[68] Y. Zeng, B. Clerckx, and R. Zhang, "Communications and signals design for wireless power transmission," *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 2264–2290, May 2017. DOI: 10.1109/TCOMM.2017.2676103.

[69] J. R. Polimeni, M. Balasubramanian, and E. L. Schwartz, "Multi-area visuotopic map complexes in macaque striate and extra-striate cortex," *Vision research*, vol. 46, no. 20, pp. 3336–3359, 2006.

[70] T. Fehervari, M. Matsuoka, H. Okuno, and T. Yagi, "Real-time simulation of phosphene images evoked by electrical stimulation of the visual cortex," in *17th International Conference of Neural Information Processing. Theory and Algorithms: Part I*, K. W. Wong, B. S. U. Mendis, and A. Bouzerdoum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, Nov. 2010, pp. 171–178. DOI: 10.1007/978-3-642-17537-4_22.

[71] L. Lei and C. Krishnendu, "On using exponential-Golomb codes and subexponential codes for system-on-a-chip test data compression.," *Journal of Electronic Testing*, vol. 20, no. 6, pp. 667–670, 2004.

[72] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 194–203, Mar. 1975. DOI: 10.1109/TIT.1975.1055349.

[73] S. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, Jul. 1966. DOI: 10.1109/TIT.1966.1053907.

[74] *DA14580 Low Power Bluetooth Smart SoC*, 3.1.0, Dialog Semiconductor, Jan. 2015.

[75] C. Bachmann, G.-J. van Schaik, B. Busze, *et al.*, "10.6 A 0.74V 200 μW multi-standard transceiver digital baseband in 40nm LP-CMOS for 2.4GHz Bluetooth Smart / ZigBee / IEEE 802.15.6 personal area networks," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb. 2014, pp. 186–187. DOI: 10.1109/ISSCC.2014.6757393.

[76] J. Blanckenstein, J. Klaue, and H. Karl, "A survey of low-power transceivers and their applications," *IEEE Circuits and Systems Magazine*, vol. 15, no. 3, pp. 6–17, 2015. DOI: 10.1109/MCAS.2015.2450634.

[77] P. Kwan, S. C. Schachter, and M. J. Brodie, "Drug-resistant epilepsy," *New England Journal of Medicine*, vol. 365, no. 10, pp. 919–926, 2011.

[78] Y. Hayashida, Y. Umehira, K. Takatani, *et al.*, "Cortical neural excitations in rats in vivo with using a prototype of a wireless multi-channel microstimulation system," in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2015, pp. 1642–1645. DOI: 10.1109/EMBC.2015.7318690.

[79] S. Kameda, Y. Hayashida, Y. Tanaka, D. Akita, and T. Yagi, "A multichannel current stimulator chip for spatiotemporal pattern stimulation of neural tissues," presented at the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Chicago, IL, USA, Aug. 2014, pp. 5011–5015. DOI: 10.1109/EMBC.2014.6944750.