WIRELESS NEURAL RECORDING AND STIMULATION SOCs FOR MONITORING AND TREATMENT OF INTRACTABLE EPILEPSY

by

Karim Abdelhalim

A thesis submitted in conformity with the requirements for the degree of degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

Copyright © 2013 by Karim Abdelhalim
Abstract

Wireless Neural Recording and Stimulation SoCs for Monitoring and Treatment of Intractable Epilepsy

Karim Abdelhalim

degree of Doctor of Philosophy

Graduate Department of Electrical and Computer Engineering

University of Toronto

2013

This dissertation presents the system architecture and implementation of two wireless systems-on-chip (SoCs) for diagnostics and treatment of neurological disorders. It also validates the SoCs as an electronic implant for preoperative monitoring and treatment of intractable epilepsy.

The first prototype SoC is a neural recording interface intended for wireless monitoring of intractable epilepsy. The 0.13µm CMOS SoC has 64 recording channels, 64 programmable FIR filters and an integrated 915MHz FSK PLL-based wireless transmitter. Each channel contains a low-noise amplifier and a modified 8-bit SAR ADC that can provide analog-digital multiplication by modifying the ADC sampling phase. It is used in conjunction with 12-bit digital adders and registers to implement 64 16-tap FIR filters with a minimal area and power overhead. In vivo measurement results from freely moving rodents demonstrate its utility in preoperative monitoring epileptic seizures.

Treatment of intractable epilepsy by responsive neurostimulation requires seizure detection capabilities. Next, a low-power VLSI processor architecture for early seizure detection is described. It the magnitude, phase and phase synchronization of two neural signals - all precursors of a seizure. The processor is utilized in an implantable responsive neural stimulator application. The architecture uses three CORDIC processing cores that require shift-and-add operations but no multiplication. The efficacy of the processor in epileptic seizure detection is validated on human EEG data and yields comparable performance to software-based algorithms.

The second prototype SoC is a closed-loop 64-channel neural stimulator that includes the aforementioned seizure detector processor and is used for preventive seizure abortion. It constitutes a neural vector analyzer that monitors the magnitude, phase and
phase synchronization of neural signals to enable seizure detection. In a closed loop, abnormal phase synchrony triggers the programmable-waveform biphasic neural stimulator. To implement these functionalities, the 0.13µm CMOS SoC integrates 64 amplifiers with switched-capacitor (SC) bandpass filters, 64 MADCs, 64 16-tap FIR filters, a processor, 64 biphasic stimulators and a wireless transmitter. The SoC is validated in the detection and abortion of seizures in freely moving rodents on-line and in early seizure detection in humans off-line. The results demonstrate its utility in treatment of intractable epilepsy.
Acknowledgements

I would like thank my supervisor, Professor Roman Genov for his guidance and support throughout this doctoral thesis. I am thankful for his great insight, guidance, recommendations, inspiration as well as his support and confidence in funding a large ambitious project and ensuring everything is available such as collaborators, equipment and fabrication resources. I would like to thank my defense committee: Professor Chan Carusone, Professor Gulak, Professor Ng and my external examiner, Professor Ghovanloo for their feedback that helped improve this thesis. I would also like to thank our collaborators at the Toronto Western Hospital, Demitre Serletis and Dr. Carlen as well as our collaborators at the Hospital for Sick Children, Prof. Perez Velazquez and Larysa Kokarovtseva for assisting us in vitro and in vivo testing. I would also like to thank CMC for fabrication access to IBM 130nm CMOS and providing us with very generous die space, Jaro for CAD support and NSERC for providing funding for both for myself as well as funding the project.

I would like to thank all my fellow graduate students and friends at the University of Toronto. These include Hamed Mazhab Jafari, Amer Samarah, Massimo Tarulli, Ricardo Aroca, Yannis Sarkas, Andreea Balteanu, Mario Milicevic, Maysam Zargham, Derek Ho, Trevor Caldwell, Bert Leesti, Alex Tomkins, Ruslana Shulyzki, Kentaro Yamamoto, Dustin Dunwell, Mike Bichan, Alireza Nilchi, Farzaneh Shahrokhi, Shahriar Shahramian, Arezu Bagheri, Arshya Feyzi, Hossein Kassiri, Nima Soltani, Rituraj Singh and Maysam Nazari.

My dad, mom and grandma for always supporting me throughout the years and motivating me to finish. My brothers and sister including Amir, Tarek and Vanessa for their love and support. Finally I would like to thank my finance Shohreh Ghetmiri who provided lots of patience, love and support and who I also met during my PHD at the University of Toronto.
Contents

List of Tables x

List of Figures xii

1 Background and Contributions 1

1.1 Objective and Chapter Overview ........................................ 1

1.2 Intractable Epilepsy Treatment by Neurostimulation ................. 3

1.2.1 Epilepsy ................................................................. 3

1.2.2 Preoperative Epilepsy Monitoring .................................... 4

1.2.3 Existing Large-form-factor Neurostimulators for Intractable Epilepsy Treatment .................................................. 5

1.2.4 Seizure Detection Algorithms ....................................... 7

1.2.5 Neural Synchrony in the Brain ...................................... 7

1.3 Integrated Circuits for Neural Implants ................................. 9

1.3.1 Integrated Circuits for Electrophysiological Recording .......... 10

Flicker Noise Reduction in AC-coupled Amplifiers ...................... 14

High-Resistance Feedback Circuits for Offset Correction ............ 14

1.3.2 Integrated Circuits for Neural Signal Processing ................ 15

FIR Filtering ....................................................................... 15

Analog Multiplication ....................................................... 17

Seizure Detection .................................................................. 17

Spectral and Vector Analysis .............................................. 18


<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.3 Integrated Circuits for Closed-loop Electrical Neurostimulation</td>
<td>19</td>
</tr>
<tr>
<td>1.3.4 Integrated Circuits for Wireless Neural Data Transmission</td>
<td>22</td>
</tr>
<tr>
<td>1.3.5 State-of-the-art System-on-Chip (SoC) Neural Implants</td>
<td>24</td>
</tr>
<tr>
<td>1.3.6 CMOS Technology Choice and Scaling</td>
<td>26</td>
</tr>
<tr>
<td>1.4 Summary of Novel Contributions</td>
<td>27</td>
</tr>
<tr>
<td>1.4.1 System-level Contributions</td>
<td>27</td>
</tr>
<tr>
<td>1.4.2 Circuit-level Contributions</td>
<td>28</td>
</tr>
<tr>
<td>1.4.3 Application-level Contributions</td>
<td>29</td>
</tr>
<tr>
<td>1.5 Thesis Organization</td>
<td>29</td>
</tr>
<tr>
<td>2 915-MHz FSK/OOK Wireless Neural Recording SoC with 64 Mixed-Signal</td>
<td>33</td>
</tr>
<tr>
<td><strong>FIR Filters</strong></td>
<td></td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>33</td>
</tr>
<tr>
<td>2.2 System Architecture and VLSI Implementation</td>
<td>35</td>
</tr>
<tr>
<td>2.3 VLSI Implementation</td>
<td>36</td>
</tr>
<tr>
<td>2.3.1 Mixed-Signal FIR Filter</td>
<td>36</td>
</tr>
<tr>
<td>2.3.2 Neural Recording Amplifier</td>
<td>40</td>
</tr>
<tr>
<td>2.3.3 Multiplying SAR ADC</td>
<td>43</td>
</tr>
<tr>
<td>2.3.4 Fully Integrated 915MHz FSK/OOK Transmitter</td>
<td>46</td>
</tr>
<tr>
<td>PFD, Charge Pump, Loop Filter and VCO</td>
<td>47</td>
</tr>
<tr>
<td>Frequency Divider and Power Amplifier</td>
<td>49</td>
</tr>
<tr>
<td>2.4 Experimental Results</td>
<td>50</td>
</tr>
<tr>
<td>2.4.1 Analog Front-end</td>
<td>50</td>
</tr>
<tr>
<td>2.4.2 ADC and Mixed-signal FIR Filter</td>
<td>56</td>
</tr>
<tr>
<td>Area and Power Comparison</td>
<td>56</td>
</tr>
<tr>
<td>Precision Comparison</td>
<td>58</td>
</tr>
<tr>
<td>Scalability</td>
<td>60</td>
</tr>
<tr>
<td>2.4.3 915MHz FSK/OOK Transmitter</td>
<td>60</td>
</tr>
<tr>
<td>2.4.4 Application Example: Noise Suppression and Spike Detection</td>
<td>66</td>
</tr>
</tbody>
</table>
4.3.2 FIR Filtering Circuits .................................................. 112
Multiplying SAR ADC .................................................. 112
Neural Signal Filtering Modes ........................................... 114
4.3.3 Phase, Magnitude, and Phase Synchronization Processor .... 114
4.3.4 UWB Transmitter (Designed by Hamed Mazhab Jafari) .... 115
4.3.5 Neural Stimulator .................................................. 116
DAC-Sharing Architecture .................................................. 116
DAC and Duty Cycle Controller ......................................... 119
Voltage-to-Current Converter ............................................ 119
Current Driver ............................................................ 120
Stimulation Artifact ....................................................... 122
4.4 Electrical Experimental Results ...................................... 122
4.4.1 Analog Front-end and Switched-Capacitor BPF ............ 124
4.4.2 ADC and Full Recording Path ..................................... 125
4.4.3 Neural Vector Analysis ............................................. 126
4.4.4 Phase Synchronization ............................................. 131
4.4.5 UWB Transmitter .................................................. 131
4.4.6 Neural Stimulator .................................................. 135
4.4.7 Offline Early Seizure Detection in Humans ..................... 135
4.5 Discussion ............................................................. 140
4.5.1 Resource Utilization ............................................... 140
4.5.2 Comparison to the State of the Art .............................. 141
4.6 Summary of Novel Contributions .................................. 142
4.7 Conclusion ............................................................ 144
5 In Vivo Validation for Epileptic Treatment ............................ 145
5.1 Introduction ............................................................ 145
5.2 Closed-loop Early Seizure Detector SoC ......................... 146
5.2.1 Early Seizure Detection Algorithm ............................ 146
5.3 Validation of the Prototype In vivo ........................................ 147
  5.3.1 In Vivo Validation of Seizure Detection Algorithm based on
        GBL Model ........................................................................ 147
  5.3.2 In vivo Validation based on Kainic Acid Model ................. 154
5.4 Experimental Validation in Epilepsy Control ......................... 155
  5.4.1 Online Early Seizure Detection and Control in Rodents In Vivo . 155
        Early Seizure Detection in Rats ........................................... 155
        Seizure Control and Abortion in Rats ................................. 156
5.5 Conclusions ............................................................................ 157

6 Conclusions and Future Work .................................................. 162
  6.1 Contributions and Relevant Publications ......................... 162
  6.2 Future Work ................................................................. 164
    6.2.1 Improvements to SoCs ............................................. 164
    6.2.2 Wireless Inductive Power and Receiver IC .................. 165
    6.2.3 Future In Vivo Testing ............................................... 166

Appendix ...................................................................................... 168

A Supplementary Hardware and Software Documentation ............. 168
  A.1 Board Design ............................................................ 168
  A.2 Simulation Results ....................................................... 169
  A.3 FIR Coefficients for Neural Vector Analysis .................... 172
  A.4 Phase Synchrony and Magnitude Processor: VERILOG RTL Code . 173

References ................................................................................. 222
List of Tables

1.1 Integrated State-of-the-Art Neural Recording and/or Stimulation SoCs . . 25

2.1 Transistor sizes of the two OTAs and the feedback element (F.E.) of the
neural amplifier and the comparator of the ADC . . . . . . . . . . . . . . 42
2.2 Transistor sizes of the VCO and power amplifier . . . . . . . . . . . . . 50
2.3 Comparison of Published FIR filters . . . . . . . . . . . . . . . . . . . . . 57
2.4 Comparison of Different Implementations for 64 16-tap FIR filters . . . 57
2.5 Comparison of ISM-band FSK transmitters . . . . . . . . . . . . . . . 65
2.6 Summary of experimental results . . . . . . . . . . . . . . . . . . . . . . 69
2.7 Integrated Wireless Neural Recording SoCs . . . . . . . . . . . . . . . 70

3.1 EEG Database Used in Verification of the Phase-Synchronization Pro-
cessor. Seizure Types: Simple Partial (SP), Complex Partial (CP), General-
ized Tonic-Clonic (GTC). Seizure Location: Hippocampal (H), Neocortical (NC).
Electrode Types: depth (d), strip (s), grid (g) . . . . . . . . . . . . . . 91
3.2 Performance comparison of bivariate seizure prediction algorithms im-
ploented in software with the current work . . . . . . . . . . . . . . . . 95

4.1 Transistor sizes of the two OTAs and the feedback element (F.E.) for
the neural recording amplifier and the neural stimulator . . . . . . . . 110
4.2 Comparison Neural Recording Amplifiers . . . . . . . . . . . . . . . . 125
4.3 Summary of the experimental results . . . . . . . . . . . . . . . . . . . . . 139
4.4 State-of-the-art neural recording and/or stimulation SoCs . . . . . . . . . 143
5.1 Electrode Coordinates . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
List of Figures

1.1 Example of a wireless closed-loop neural recording and stimulation SoC for intractable epilepsy treatment. ............................. 2
1.2 (a) Cyberonics vagus nerve stimulator (images from Cyberonics). (b) Medtronic deep brain stimulator (images from Medtronic). (c) Neuropace RNS neurostimulator (images from Neuropace). ............... 6
1.3 Various neural disorders and their phase synchrony frequency bands. . . 9
1.4 (a) Utah electrode array. (b) University of Michigan-style electrode probe (image from Neuronexus). ................................. 11
1.5 Equivalent circuit model of the electrode-tissue interface, from [1]. . . 11
1.6 Conventional AC-coupled neural amplifier. .............................. 13
1.7 Chopper-stabilized AC-coupled neural amplifiers. ........................ 13
1.8 High impedance pseudo-resistors utilizing (a) a diode-connected MOSFET, and (b) a sub-threshold biased MOSFET. ......................... 14
1.9 Hardware implementation for (a) direct-form FIR filter (b) transpose-form FIR filter. ......................................................... 16
1.10 Example of a neural vector analyzer and phase synchrony computation. 20
1.11 Failure rates from a manufacturer for different technology nodes. The figure is from [2] .............................................................. 27
1.12 Visualization of the thesis organization. ................................. 32
2.1 System diagram of the envisioned implantable wireless neural recording system. ................................................................. 34
2.2 (a) Simplified system diagram of the wireless neural recording SoC. (b) Micrograph of the 4mm by 3mm 0.13\mu m CMOS SoC.

2.3 (a) Conventional 16-tap transposed FIR structure. (b) Mixed-signal 16-tap transposed FIR filter structure.

2.4 (a) Schematic of 64 implemented mixed-signal 16-tap symmetric transposed FIR filters. (b) Control signals for FIR filter.

2.5 Schematic of the analog front-end.

2.6 (a) Simulated resistance across the amplifier output range for the proposed feedback device and for two series PMOS devices. Both cases have $V_{GS}$ of 0.1V. (b) Simulated highpass frequency pole across the amplifier output range for the proposed feedback device and for two series PMOS devices, based on a 100fF feedback capacitor.

2.7 (a) Sampling phase of multiplying SAR ADC. (b) Hold phase of multiplying SAR ADC.

2.8 Schematic of the in-channel multiplying SAR ADC and comparator.

2.9 (a) Block diagram of the fully integrated FSK/OOK wireless transmitter. (b) Schematic of the phase-frequency detector, charge pump and loop filter. (c) Schematic of the static-CMOS divide by 64. (d) Schematic of the output driver and power amplifier.

2.10 Experimentally measured (a) amplitude response of the analog front-end showing tuneability of the high-pass pole, (b) frequency response of eight different channels and (c) input-referred noise of eight different channels.

2.11 (a) Experimentally measured THD of the analog front-end. (b) Experimentally measured high-pass filter frequency response for three different values of $I_{HP}$ on two different channels on the chip. Experimentally measured output waveforms for a 1mV$_{pk-pk}$ 1Hz sinusoid input comparing (c) conventional and (d) proposed feedback elements performance.
2.12 (a) Experimentally measured FFT of an in-channel ADC output for 1kHz sinusoid input at 54kS/s. (b) Experimentally measured ENOB vs. input frequency for an in-channel ADC. 53

2.13 Ideal and experimentally measured frequency responses of various programmable FIR filter configurations at 7.1kSps. 54

2.14 (a) Experimentally measured output spectrum of the FIR filter. (b) Experimentally measured SNDR vs. input amplitude with a 100Hz sinusoid input to the FIR filter and to the FIR filter including the analog front-end. 55

2.15 SIMULINK simulation comparing the dynamic performance between an 8-bit ADC with a 16-tap 16-bit digital FIR filter, the proposed 16-tap 8-bit multiplying ADC mixed-signal FIR filter and a 7-bit ADC followed by a 16-tap 16-bit digital FIR filter. 59

2.16 Scalability of the proposed FIR filter length with respect to (a) sample rate, (b) area and (c) power dissipation. 61

2.17 Experimentally measured (a) spectrum of FSK-modulated data at a 1m distance, (b) transmitted and received data from 1m distance at 1.2Mbps using FSK modulation, (c) OOK-modulated data into 50Ω termination resistor and (d) transmitted and received data from a 1m distance at 10kb/s using OOK modulation. 62

2.18 (a) Experimentally measured phase noise of the RF transmitter. (b) Experimentally measured output spectrum showing the reference spur of the RF transmitter. 63

2.19 Experimental setup for the wireless transmitter. 64

2.20 Experimentally measured data wirelessly received from the chip at 1m distance (including the full signal path with analog front-end, FIR filter and FSK RF TX). 65
2.21 Experimentally measured (a) TX output power and (b) DC power for the RF TX when connected to a 50Ω termination resistance for different power-levels. ................................. 65

2.22 Example of experimentally measured signal band separation using full recording channel (with LP FIR filter). ................................. 67

2.23 Example of experimentally measured spike detection using full recording channel (with BP FIR filter). ................................. 67

2.24 Area and power breakdown of the SoC. ................................. 71

2.25 (a) Online on-chip in vivo recordings before seizure activity triggered by GBL injection, recorded using on-chip BP FIR filter.  (b) Online on-chip in vivo recording showing seizure activity after GBL injection recorded using on-chip BP FIR filter. ................................. 72

3.1 Closed-loop neural recording and stimulation system for epileptic seizure detection and control. ................................. 76

3.2 Top-level block diagram of two analog neural recording channels and the digital phase-synchronization epileptic early seizure detector. .... 77

3.3 VLSI architecture of the phase-synchronization processor. .... 82

3.4 Simulated PLV for different values of N (length of the FIR filters) when the frequency of one input, \( V_{I,N1} \), is held constant at 110Hz and the frequency of the other input, \( V_{I,N0} \), is swept from 70Hz to 150Hz. .... 83

3.5 (a) Simulated error of the PLV for different processor resolutions when compared with an ideal processor resolution.  (b) Offline human seizure input for a University of Toronto seizure patient.  (c) Simulated PLV between two inputs in the 25Hz to 35Hz frequency band computed for a processor resolution of 7-bits and 10-bits. .... 84

3.6 Micrograph of phase synchronization processor. .... 85

3.7 Experimentally measured FIR on-chip FIR filters to generate real and imaginary components. ................................. 86
3.8 (a) Experimentally measured magnitude of a sinusoidal input computed by the phase-synchronization processor. (b) The corresponding relative percent error. .................................................. 87

3.9 Experimentally measured phase difference between two input sinusoids. (b) The corresponding relative percent error. .................................................. 88

3.10 Experimentally computed and simulated PLV when the frequency of one input, $V_{IN1}$, is held constant at 110Hz and the frequency of the other input, $V_{IN0}$, is swept from 60Hz to 160Hz. ......................... 90

3.11 Experimentally measured power dissipation at different operating frequencies. .................................................. 90

3.12 Early seizure detection results example for patient 2. (a), (b) Two input intracranial EEG signals recorded on electrodes 13 and 23, respectively. (c), (d) Magnitude of the signals shown in (a), (b), respectively, after they are bandpass filtered in the 15Hz to 25Hz frequency range, computed by the prototyped phase-synchronization processor. (e) Integrated PLV between the two inputs in the 15Hz to 25Hz frequency band computed by the prototyped phase-synchronization processor. ........... 93

3.13 Early seizure detection results example for patient 3. (a), (b). Two input intracranial EEG signals recorded on EEG electrodes 1 and 2, respectively. (c), (d) Magnitude of the signals shown in (a), (b), respectively, after they are bandpass filtered in the 15Hz to 25Hz frequency range, computed by the prototyped phase-synchronization processor. (e) Integrated PLV between the two inputs in the 15Hz to 25Hz frequency band computed by the prototyped phase-synchronization processor. ........... 94

3.14 Early seizure detection results for patient 2, showing (a) true positive rate vs. false positive rate plot with the seizure occurrence period held at 30min, and (b) true positive rate vs. seizure occurrence period plot. For the above detectors, MAG refers to magnitude, PLV refers to phase locking value, and RND refers to random. .......................... 96
3.15 Experimentally measured early seizure detection results example for patient 1. (a), (b). Two input intracranial EEG signals recorded on EEG electrodes 41 and 44, respectively. (c) Experimentally measured magnitude in the 15-25Hz range of the signal shown in (a). (d) Experimentally measured magnitude in the 15-25Hz range of the signal shown in (b). Experimentally measured PLV between signals (a) and (b).

4.1 System diagram of the envisioned implantable closed-loop wireless SoC.
4.2 Functional diagram of the SoC.
4.3 Detailed system block diagram of the SoC.
4.4 Neural recording channel including LNA, SC BPF and SC LPF.
4.5 Noise and area tradeoff of introducing chopping in front and inside the OTA.
4.6 Summary of noise contributions with the chopper enabled and disabled.
4.7 Schematic of the multiplying SAR ADC (MADC).
4.8 Integrated circuit implementation of eight channels of neural signal filtering for different modes of operation. (a) Raw ADC mode, (b) General purpose filtering FIR mode and (c) I/Q extraction mode.
4.9 Block diagram of the phase, magnitude and phase synchronization processor.
4.10 Schematic of the UWB transmitter.
4.11 MADC reuse in the neural stimulation mode.
4.12 (a) Schematic of neural stimulator. (b-c) Example simulation results showing output bits from the SAR ADC logic being used to generate a stimulator current pulse.
4.13 Micrograph of the 4mm × 3mm 0.13μm CMOS SoC.
4.14 (a) Layout and (b) micrograph of the recording and stimulation channel.
4.15 Experimentally measured (a) input-referred noise, (b) frequency response of neural recording amplifier and (c) total-harmonic distortion (THD). (d) Experimentally measured output waveforms for a 20Hz $1\text{mV}_{\text{pk-pk}}$ input and (e) a 20Hz $20\mu\text{V}_{\text{pk-pk}}$ input. ........................................ 123
4.16 Experimentally measured frequency response of the neural recording amplifier and the switched-capacitor bandpass filter. ................................. 124
4.17 Experimentally measured FFT of an in-channel ADC output with a 640Hz input and sampled at 28kS/s. .......................................................... 126
4.18 Experimentally measured (a) INL and (b) DNL of an in-channel ADC, sampled at 28kS/s. ................................................................. 127
4.19 (a) Experimentally measured output FFT of an in-channel analog frontend, SC BPF and ADC with a 65Hz $700\mu\text{V}$ input sampled at 14kS/s. (b) Experimentally measured output FFT of an in-channel analog frontend, SC BPF, SC LPF, RC LPF and FIR LPF with a 65Hz $700\mu\text{V}$ input sampled at 7.2kS/s. .................................................. 128
4.20 (a) A low-amplitude sinusoidal input into neural recording amplifier. (b) Experimentally measured I/Q separation of signal in (a) including full signal path of input (a). (c) A high-amplitude sinusoid input into neural recording amplifier. (d) Experimentally measured I/Q separation including full signal path of input (c). (e) Experimentally measured frequency response of full signal path for I and Q. .............................. 129
4.21 (a) Experimentally measured phase difference computation including full signal path (100$\mu\text{V}$ inputs). (b) Experimentally measured phase difference error for different input amplitudes between two neural recording amplifier. (c) Experimentally measured magnitude computation including full signal path and compared with ideal case. (d) Experimentally measured magnitude computation of a low frequency step introduced at the input of the neural recording amplifier. .......................... 130
4.22 (a) Experimentally measured PLV computation between two different channels for three different frequency bands. (b) Experimentally measured PLV computed between two channels, where one frequency is constant for one channel and the other frequency is increasing for the other channel.

4.23 Comparison between simulated and experimentally measured PLV computed between two channels, where one frequency is constant for one channel and the other frequency is increasing for the other channel.

4.24 (a) Wireless TX digital Manchester-encoded data and modulated UWB pulses. (b) Experimentally measured Output spectrum of UWB pulse into 50Ω load.

4.25 (a) Wireless TX digital Manchester-encoded data. (b) Experimentally wirelessly received data from a 5cm distance @ 10Mb/s in the 3.1-10.4GHz band.

4.26 (a) Experimentally measured output current for different output voltages. (b) Output current vs. DAC input for positive and negative current cases. (c), (d) Experimentally measured neural stimulating current into a saline solution for different duty cycle and amplitude control settings. (e) Biphasic stimulator set to 450μA output current and connected to a 1kΩ load.

4.27 Offline human seizure detection for a University of Toronto seizure patient.

4.28 (a) Power dissipation breakdown of the SoC (neural stimulator excluded). (b) Area breakdown of the SoC. The remaining 5mm² are utilized for I/O, ESD, test blocks, routing, decoupling capacitors and biasing.

5.1 *In vivo* test setup.

5.2 In-vivo testing on freely moving rat showing detection of a non-convulsive seizure triggered by a GBL injection, at t=0.
5.3 Example of magnitude and phase synchrony extraction on a freely moving rat injected by GHB. Comparison of offline SIMULINK model with online realtime SoC. .......................................................... 150
5.4 Example of an early seizure detection example on a freely moving rat, 
\textit{in vivo}. Seizure triggered by a kainic acid injection. ....................... 151
5.5 Example of an early seizure detection example on a freely moving rat, 
\textit{in vivo}. Seizure triggered by a kainic acid injection. ....................... 152
5.6 Example of an early seizure detection example on a freely moving rat, 
\textit{in vivo}. Seizure triggered by a kainic acid injection. ....................... 153
5.7 Successful abortion of a seizure in a rat using the on-chip neural stimu-
lator. ............................................................................................................. 156
5.8 Example of closed-loop neural stimulation aborting a seizure for a freely 
moving rat \textit{in vivo}. Seizure triggered by a kainic acid injection. ............ 158
5.9 Example of closed-loop neural stimulation aborting a seizure for a freely 
moving rat \textit{in vivo}. Seizure triggered by a kainic acid injection. ............ 159
5.10 Example of closed-loop neural stimulation aborting a seizure for a freely 
moving rat \textit{in vivo}. Seizure triggered by a kainic acid injection. ............ 160
6.1 Micrograph of the 2.5mm by 1mm 0.13\textmu m CMOS wireless power and 
data receiver IC. ............................................................................................. 166
A.1 The PCB designed for characterizing the two SoC prototypes. .......... 169
A.2 Simulation results showing AFE and signal processing. .................. 170
A.3 Simulation results showing stimulator and RF TX. ......................... 171
Chapter 1

Background and Contributions

1.1 Objective and Chapter Overview

This work focuses on a novel implantable technology that wirelessly monitors on-line the abnormal electrical activity of epileptic seizures in the brain and dynamically aborts them by means of electrical neurostimulation. The form-factor novelty of the presented approach is derived from the fact that the recording and processing of neurological signals needed to detect epileptic seizures accurately typically requires bulky electrophysiological equipment. Furthermore, significant computing resources are also required to provide on-line real-time early seizure detection information which can then be correlated in time with external electrical stimuli. Design of such instruments has been a significant research direction around the world, but to date, a closed-loop electrical neurostimulator utilizing a high efficacy seizure detector and implemented as a monolithic system-on-chip (SoC) has not been engineered.

A simplified block diagram of the proposed wireless neural stimulator with a closed-loop feedback is depicted in Figure 1.1. The bidirectional system is connected to the brain through a set of electrodes and would be implanted under the skull. Energy is
delivered wirelessly through inductive coils and data are to be sent by an RF wireless transmitter. The key building blocks for the implantable SoC are neural recording, signal processing for early seizure detection, neural stimulation for seizure abortion and a wireless interface. Applying an electrical stimulus in response to abnormal neuroelectrical activity in a closed loop is utilized to treat intractable epilepsy [3].

This thesis focuses on the design and validation of two wireless systems-on-chip (SoCs) for diagnostics and treatment of intractable epilepsy. The first prototype SoC is a neural recording and signal processing interface for wireless preoperative monitoring of intractable epilepsy. The second prototype SoC operates as a neural vector analyzer to compute the magnitude, phase and phase synchrony of neural signals. This SoC detects epileptic seizures and triggers an on-chip neural stimulator in a closed loop to abort a seizure. This thesis also demonstrates the intended utility of these SoCs in monitoring and treatment of intractable epilepsy in humans and rodents, respectively.
In this introductory chapter, the treatment of intractable epilepsy by neurostimulation is discussed in Section 1.2. Previously reported integrated circuits (ICs) for neural recording, neural signal processing, neural stimulation and neural wireless transmitters are extensively reviewed in Section 1.3. Lastly, the thesis contributions and the thesis organization are provided in Section 1.4.

1.2 Intractable Epilepsy Treatment by Neurostimulation

1.2.1 Epilepsy

Approximately 50 million people worldwide are epileptic. There are over 15,000 new cases of epilepsy in Canada annually. Epilepsy is the third most common neurological disorder following stroke and Alzheimer’s disease, but it imposes higher costs on society than stroke does. Approximately 30 percent of epileptic patients suffer intractable epilepsy in that the symptoms do not respond to medication. Half of these individuals will ultimately be deemed candidates for surgical resection of the source of the seizures (i.e., the seizure focus), with its own risks. Preoperative epilepsy monitoring with implanted electrode grids are done to localize the focus, and to ensure that the resection can be accomplished without producing neurological deficits. Resection of the focus in well assessed patients portends an 80 percent chance of being rendered seizure free. The remainder of all the patients can only be treated with vagus nerve stimulation and deep brain stimulation, both open-loop and of limited benefit.

None of these therapies considers the nonlinear dynamics of dysfunctional brain activity exhibiting themselves in changes in neural synchrony, or utilizes an automated therapeutic feedback approach. Epileptic seizure onsets are often characterized by specific precursors including increased fluctuations in phase synchrony and subsequent increased brain synchronization [4]. There are many studies that have investigated the
higher synchrony associated with epileptiform activities.

1.2.2 Preoperative Epilepsy Monitoring

The focus of a seizure can be identified from a variety of tests of both function (EEG, MEG, fMRI and PET scans) and structure (MRI and CT scans). These methods are complementary and often have limited efficacy and applicability. Cortical surface electrical recordings known as electrocorticography (ECoG) provide better signal-to-noise ratio than scalp surface electroencephalography (EEG), and higher bandwidth, up to 500 Hz of signal content, as well as increased spatial resolution. ECoG is currently the gold standard procedure for seizure localization in most patients with intractable epilepsy who are candidates for resective surgery. It involves implanting ECoG electrode arrays that may take the form of a matrix of contacts (grid), or a linear array of contacts (strip and depth electrodes), typically embedded in a thin sheet of silicone and placed either over the cerebral cortex (surface electrodes), or within the brain substance itself (depth electrodes). Another benefit of ECoG is that it allows for brain function assessment prior to surgery. This is done by neural stimulation through the same electrodes in order to understand the function of the part of the brain that is to be removed. Unlike grid and strip electrodes, depth electrodes are used to record activity from deeper structures such as the hippocampus.

The current ECoG preoperative epilepsy monitoring and brain function assessment procedure requires that wires from the implanted electrode array connect to electrophysiological equipment. These include racks of neural amplifiers and stimulators, computers with digital visualization, spike and seizure detection software, and nurse alarm systems. As a result, patients have to be tethered while staying in a neuro-critical care unit and then in an epilepsy monitoring unit for up to several weeks making this approach cumbersome, unsafe and costly. The patients skin has to be opened during
the entire procedure which increases the risk of infection. Also, not infrequently, during seizures, patients pull the implanted electrodes out, which terminates the recording session and may require another surgery for reimplantation if enough data was not obtained the first time. In addition, sometimes during motor seizures, there are marked movement artifacts, which will be greatly diminished by using a wireless system.

1.2.3 Existing Large-form-factor Neurostimulators for Intractable Epilepsy Treatment

Commercial neurostimulation systems for medically refractory epilepsy treatment are available from Cyberonics, Medtronic, and Neuropace as shown in Figure 1.2(a)-(c). Cyberonics offers a device implantable in the chest cavity that continuously stimulates the vagus nerve (VNS), without any monitoring of the brain as shown in Figure 1.2(a). Fewer than 10 percent of patients implanted with this device become seizure-free [5]). Starting from 2010, Medtronic offers a programmable deep brain stimulation (DBS) system. It is an open loop system that delivers stimuli to the anterior nucleus of the thalamus at scheduled intervals on up to eight electrodes as shown in Figure 1.2(b). Clinical trials showed that 13 percent of patients were rendered seizure-free [6].

Neuropace, the only FDA-approved closed-loop neurostimulator out of the three, is in the clinical trials stage with its responsive neural stimulator (RNS) product which has up to eight electrodes as shown in Figure 1.2(c). Smaller studies of this device resulted in 13 percent of subjects being rendered seizure-free [7], [8]. A larger study reported 55 percent of subjects having more than 50 percent reductions in the number of seizures, and 7 percent of subjects rendered seizure-free [9].

All of the above commercially-available systems have a large form factor (not a single microchip), have few recording or stimulation channels and have no or primitive signal processing.
Large-form-factor neurostimulators developed in academia that may potentially be adapted to treat epilepsy include Neurochip-2 at University of Washington [10] and HermesD at Stanford University [11]. These designs utilize many basic electronic components but are not implemented as a single advanced integrated circuit. This results in a prohibitively large size (for full implantation in either rodents or humans) and limited functionality (e.g., Neurochip-2 measures 55x35x20mm$^3$, weights 145g, has only three recording and stimulation channels and performs no signal processing as, for example, needed for early seizure detection).

Figure 1.2: (a) Cyberonics vagus nerve stimulator (images from Cyberonics). (b) Medtronic deep brain stimulator (images from Medtronic). (c) Neuropace RNS neurostimulator (images from Neuropace).
1.2.4 Seizure Detection Algorithms

Over the last two decades, extensive research has been conducted on prediction and early detection of seizures using a variety of different methods. Univariate algorithms, which involve computations on a single input, have been used to predict seizures. Such methods include computing wavelet transforms [12], energy of signal bands [13], correlation dimensions [14] and computing the Lyapunov exponent [15]. For example, the Neuropace responsive neural stimulator uses three seizure detections methods: computing the total area under a curve, the line-length and the half-wave of a neural signal [3]. These univariate algorithms lack spatial specificity because they only rely on one neural input.

This issue can be addressed through bivariate or multivariate algorithms which operate on two or more inputs, respectively [16]. Seizure prediction and detection algorithms that use bivariate or multivariate measures to quantify synchronization among two or more neural signals have been shown to yield superior accuracy. These more computationally intensive techniques that use inputs from multiple recording sites such as computing the phase synchronization, the correlation index and the similarity index between neural signals, have been used to develop more accurate seizure prediction and detection algorithms [17], [18], [19], [20], [21], [22], [23], but to date have only been implemented in software. Computing phase synchronization has been shown to outperform other multivariate algorithms [24]. It relies on the changes in neural synchrony in the brain.

1.2.5 Neural Synchrony in the Brain

Neuroelectrical activity in the brain generates electrophysiological signals in multiple frequency bands such as alpha (8-12Hz), beta (13-30Hz) and gamma (40-80Hz). Through oscillations in these bands, information is communicated among various re-
regions in the brain. These oscillations have been linked to a wide range of cognitive and perceptual processes including sleep states and memory [25]. Simple tasks, such as motor or sensory tasks also often require the coordination or synchronization of many brain areas [25]. On the other hand, abnormal coordinated interactions in various regions of the brain are widely recognized as key indicators of pathological brain states in numerous neurological disorders such as epilepsy, schizophrenia, Parkinson’s disease, autism and traumatic brain injury [26]. Figure 1.3 outlines the frequency bands where abnormal phase synchrony is exhibited by various neurological disorders and the percentage of the population who suffer from each disease [27], [28]. It has been shown that before and during a seizure the amount of synchrony between these oscillations among neurons located in different regions of the brain changes significantly [17]. Thus, the amount of synchrony between multiple neural signals is a strong indicator in predicting or detecting seizures [17], [18].

A number of methods exist that quantify the level of frequency-specific synchronization between two neuroelectric signals, including wavelets and coherence. Estimation of phase locking has emerged as a popular leading method of quantifying neural synchronization. Its effectiveness comes from the fact that it relies on the phase information of a neural signal, separately from its magnitude [29], and there is substantial evidence for the fundamental importance of phase relations in establishing communication patterns between neurons. This method is effective in measuring neural synchrony from many types of recordings: deep-brain, intracranial, MEG, scalp EEG, as well as in animal models. Notwithstanding certain technical and analytical limitations in the assessment of synchrony patterns from brain signals (extensively treated in [30]), the quantification of the coordinated activity amongst brain circuits using this type of measurement is revealing fundamental information from the normal and the pathological brain. To quantify the level of synchrony between two neural signals, a phase lock-
1.3 Integrated Circuits for Neural Implants

Very-large-scale integration (VLSI) technology enables miniature (e.g., less than 1 cm³) single-chip implementations of neurostimulators. Heat dissipation within a neurostimulator resulting in one degree Celsius temperature increase is deemed safe for the cortex tissue. This corresponds to an overall system power density of 15-80 mW/cm² [34] and is dependent on the heat conductivity of the encapsulation materials. To be deemed safe for implantation, the overall power density of a miniature neurostimulator must be
reduced to satisfy this constraint. Low power dissipation is also critical for extending battery life. In one study, the Cyberonics VNS had a mean battery life that only ranged from 8 to 12 years [35]. Minimizing the power dissipation directly decreases the cost of the therapy as additional surgical operations are required for battery replacement.

1.3.1 Integrated Circuits for Electrophysiological Recording

Monitoring neuro-electrical activity in the brain such as neural oscillations is a vitally important method in diagnostics and evaluation of various neurological disorders such as epilepsy. As mentioned in Section 1.2.2, intracranial electrical recordings known as electrocorticography (ECoG) record local field potentials (LFPs) on the cortex surface and provide better signal-to-noise ratio, higher bandwidth, up to 500Hz of signal content, as well as increased spatial resolution compared to other techniques for monitoring neuro-electrical activity. Other invasive methods include neural recording from micro-electrodes, such as the commercially available Utah electrode array and the University of Michigan-style probe as shown in Figure 1.4(a) and (b), respectively. They allow for recording of wider-bandwidth single neuron activity (neural spikes) in addition to LFPs. These tiny electrodes are able to interface with a highly dense network of neurons. The high number of electrodes enable the most accurate representation of neural activity, and allows better control over the location of the stimulation. The Utah electrode array (UEA) consists of an array of 1 to 1.5mm long silicon electrodes with a pitch of 400µm. The University of Michigan-style probe [36] is another microelectrode array and it utilizes multiple connection sites along each electrode shank to allow for recording from different depth in the neural tissue.

The first-order-model for the tissue-electrode interface is shown in Figure 1.5 [1]. Impedance $Z_{CPE}$ represents the interface capacitance of the electrode. For commonly used electrodes for neurostimulator the capacitance is typically in the order of 0.16nF.
CHAPTER 1. BACKGROUND AND CONTRIBUTIONS

Figure 1.4: (a) Utah electrode array. (b) University of Michigan-style electrode probe (image from Neuronexus).

Figure 1.5: Equivalent circuit model of the electrode-tissue interface, from [1].

to 16nF. Resistor, $R_{ct}$ is the transfer resistance due to the oxidation and reduction currents that flow across the electrode-tissue interface and is the order of 100MOhm to 10GOhm. The series resistance, $R_S$, and the tissue resistance, $R_{tissue}$, when combined are typically 10kOhm. Lastly, voltage sources, $V_{sig}$ and $V_{OS}$ model the low amplitude neural signal and the DC offset at the electrode tissue interface, respectively.

Typically for ECoG recordings, the signals are in the range of 10µV to 5mV with frequencies below 500Hz. For microelectrodes, signals include frequencies up to 7kHz [37]. For an electrode with a resistance of 1MΩ and a recording bandwidth of 5kHz, the thermal noise contribution from this electrode will be given as

$$V_{RMS} = \sqrt{4kTR_D} = 9.3\mu V_{RMS} \quad (1.1)$$

with a temperature of 37 degrees Celsius (body temperature), and $k$ as Boltzmann’s constant. Furthermore there is background noise within the extra-cellular region, resulting
in a total electrode noise and extra-cellular noise to be approximately $20\mu V$ for a $1M\Omega$ electrode with a 5kHz bandwidth [38].

As neural potentials recorded through implanted electrodes typically have amplitudes of $10\mu V$ to 5 mV with a bandwidth between 0.5Hz and 7000Hz, circuit noise at these low frequencies should be minimized. The large DC voltage offset due to the interaction of electrodes and the tissue and will saturate a neural recording amplifier and thus needs to be eliminated. This results in the requirement of a low frequency high-pass filter (below 1Hz). The requirement of low flicker noise and a low frequency high-pass pole results in very large transistors and large integrated passives. The requirements for an integrated neural recording front-end include a compact AC-coupled amplifier with a gain of approximately 60dB, a bandwidth from below 1Hz to up to 5kHz and an integrated noise of below $10\mu V$. With neural signals as high as 5mV, this requires a dynamic range for the ADC of 8-10-bits. For implantable devices, to avoid neural tissue damage due tissue heating, the power density of the IC should be below 0.8mW/mm$^2$ [39], [34]. Minimizing power dissipation also minimizes the form factor of the device and ensures wireless power harvesting is feasible. The challenge in achieving a large number of recording channels (more than 32) and a low-noise front-end lends itself to high power dissipation and large silicon area.

As many as 100 or more recording sites in the brain can be monitored simultaneously. Through multiple recording channels coordinated synchronization between various regions of the brain can be monitored. To cover a wide range of neural activity in different regions of the brain and study the communication between multiple neurons, a large number of recording channels is required. As well, with state-of-the-art electrode arrays exceeding 64 recording sites, a large number of channels is required to interface with these electrode arrays.

The conventional neural amplifier architecture is shown in Figure 1.6. The gain
CHAPTER 1. BACKGROUND AND CONTRIBUTIONS

Figure 1.6: Conventional AC-coupled neural amplifier.

![Diagram of conventional AC-coupled neural amplifier]

Figure 1.7: Chopper-stabilized AC-coupled neural amplifiers.

![Diagram of chopper-stabilized AC-coupled neural amplifiers]

is set by ratio of the input capacitor to the feedback capacitor and the highpass pole is set by the feedback resistance and capacitance. Examples of this architecture were demonstrated in [40] which integrates 96-channels, with an in-channel ADC of 10-bit resolution. It requires a large 25mm\(^2\) die area in 0.13\(\mu\)m CMOS and dissipates 6.5mW. Other stand-alone multi-channel neural recording interfaces were reported in [41], [42], [43], [44] including previous work from our research group [45], demonstrate channel counts ranging between 16 and 256 channels. This conventional neural amplifier utilizes small input capacitors and has a high-input-impedance, but suffers from high flicker noise, resulting in the requirement of large input transistors.
Flicker Noise Reduction in AC-coupled Amplifiers

As neural signals exist in the sub Hz to 10kHz frequency band flicker noise becomes a dominate noise source. A few designs introduce chopping in the neural recording amplifier to remove flicker noise. Conventionally, placing chopping in front of the input pair as shown in Figure 1.7(a), as is done conventionally [46], [47], and [48] results in noise multiplication due to a switched capacitor resistor created by the parasitic capacitance of the input pair [46]. Conventionally, this is overcome by utilizing very large capacitors at the input of the amplifier [46] at the expense of large silicon area. Compared with neural amplifiers with no chopper-stabilization, the conventional chopper designs result in higher power dissipation, lower bandwidth (below 200Hz) and large chip area, making them not directly suitable for high channel counts. Another approach is to place the chopper at the front of the AC-coupled capacitors in the neural recording amplifier as shown in Figure 1.7(b), as demonstrated by [49]. This approach, can utilize small input capacitors but will suffer from a reduced input impedance and limited head-room due to the DC-offset of the biomedical tissue (which can be in the order of several 100mV). An area-efficient (suitable for large channel counts) and high input impedance chopper-stabilized neural amplifier has not yet been reported.

High-Resistance Feedback Circuits for Offset Correction

To implement a low-frequency highpass pole (below 1Hz), the feedback resistor in Figure 1.6 must be in the order of a few GOhm. Implementing such a passive resistor in
a standard CMOS technology would require too much silicon area. The conventional approach has been to utilize a diode-connected MOSFET as shown in Figure 1.8(a) or subthreshold-biased MOSFET as shown in Figure 1.8(b). The diode connected MOSFET [41] cannot be tuned, has non-linearity at high output swings, but maintains a stable DC operating bias point for the OTA. The subthreshold biased MOSFET [50], [51], [52] can be tuned for different frequency bands, but suffers from non-linearity at high output swings and the DC biasing can drift if the resistance is set too high. A high resistance feedback element that is linear, area-efficient, tunable and has a stable DC operating point is required to maintain a stable low frequency highpass pole.

1.3.2 Integrated Circuits for Neural Signal Processing

FIR Filtering

The power and area constraints imposed by an implantable device have restricted on-chip signal processing to simple algorithms, such as spike detection or to computing on a few recording channels [41], [53], [54], [55]. Adding advanced, customizable on-chip signal processing algorithms such as FIR filters capable of operating on many recording channels can significantly expand the functionality of an implant. An advantage of FIR filters over IIR filters is that they can have a linear phase. This becomes important when analyzing the phase of neural signals as it will not distort the phase. The finite-impulse response (FIR) filter has a difference equation of the form

\[ y[n] = \sum_{k=0}^{M-1} b_k \times x[n - k], \]  

(1.2)

where M is the length of the filter, b are the filter coefficients and n is the current digital sample. This equation is mapped to the direct-form hardware implementation shown in Figure 1.9(a). The equivalent transpose-form FIR filter can be derived and its hardware
implementation is shown in Figure 1.9(b). The digital multipliers result in the highest complexity for the FIR filters, especially with a large number of taps.

FIR filters have many applications in neural SoCs. When recording from a large number of channels, the wireless data transmission rate can be reduced by data compression. Noise and interference can be suppressed. Precise bandpass filtering of neural signals such as LFPs or neural spikes can be implemented. Other advanced tasks such as feature extraction and data classification enable a closed-loop system implementation such as that with locally triggered neural stimulation or drug delivery. Lastly, on-chip signal processing can be utilized to power down the RF transmitter and further reduce the power dissipation of the implantable device.

Several techniques to reduce the hardware complexity of FIR filters in VLSI have been demonstrated [56–58]. A digital approach is to use memory-based structures such as distributed arithmetic (DA) or look-up tables (LUT) [58]. For a 16-tap FIR filter with 8-bit resolution the area in 90nm CMOS is 200x200µm² and 180x180µm² using the DA implementation and the LUT implementation, respectively. Many analog FIR filters have also been demonstrated [59–61]. A 16-tap FIR filter in 45nm CMOS was demonstrated which uses binary weighted transconductance stages [59]. This implementation utilizes 380x380µm² of silicon area. A combined SAR ADC/FIR filter in
0.13$\mu$m CMOS with 12-taps was demonstrated [60] that utilizes 1100x600$\mu$m$^2$. A 16-tap, 7-bit analog FIR filter based on floating gates was demonstrated in 0.35$\mu$m and utilizes 400x400$\mu$m$^2$ [61].

Integrating a large number of N-tap FIR filters (32 or more) onto a single SoC requires significant hardware resources. One technique to reduce the area is to simplify the multiplication operation in the FIR filter.

**Analog Multiplication**

The multiplication operation is the most hardware intensive operation in the FIR filter. Each N-tap FIR filter requires N-multipliers. If the FIR multiplication operation can be performed within (or before) the ADC in a system, significant area and power dissipation can be saved. ADCs which perform multiplication within the conversion cycle with little extra area and power overhead have been demonstrated. A multiplying dual-slope ADC [62], algorithmic ADC [63], delta-sigma ADC [64], pipelined ADC [65] have been reported in literature. Analog multipliers have also been demonstrated [66], [67], but these require additional circuits to be placed before the ADC and thus have additional area and power overhead.

**Seizure Detection**

Existing VLSI systems that perform signal processing on neural signals typically employ univariate seizure detection algorithms. These involve computations on a single input, such as computing spike thresholds [68], correlation integrals [69], autoregressive parameters [70], extracting energy bands, RMS, maximum-minimum, line-length and nonlinear energy [71] and analog wavelet filtering [12]. More computationally intensive techniques that use inputs from multiple recording sites such as computing the phase-synchronization yield a higher efficacy seizure detector [24] than the current
state-of-the-art VLSI approaches. To compute phase synchronization, the phase needs to be extracted from the neural signals. This requires the implementation of a neural vector analyzer.

**Spectral and Vector Analysis**

More recently, researchers have also started performing neural recording and signal processing that extracts spectral information such as the magnitude or energy that exists in specific frequency bands [48] and [72], operating similarly to an analog AM radio receiver. The real-time processing of spectral energy bands can help establish a closed-loop mechanism. However, these implementations ignore information that exists in the phase of each signal and particularly in the phase difference between signals. A neural recording SoC which processes both the magnitude and phase (a neural vector analyzer) or the amount of synchronization between two or more neural inputs, has not yet been demonstrated.

To implement a neural vector analyzer, the in-phase (I) and quadrature-phase (Q) vectors of the neural signal need to be extracted. In applications related to radio frequency (RF) communication RC/CR polyphase filters are used to implement accurate I and Q generation [73, 74]. The frequencies at which the I and Q generation occurs are related to the RC time constant, and thus for frequencies in the neural band (1-7kHz), this yields very large passives. First-order or higher-order OTA-based analog filters can also be used to implement an all-pass filter to achieve the 90 degree phase shift required for I and Q generation [75]. For neural recording applications, this method also requires very large passive devices resulting in large silicon area.

A more area-efficient method is the Hilbert transform, \( \hat{f}(t) \) of function \( f(t) \) and is defined as

\[
\hat{f}(t) = \frac{1}{\pi} P \sum_{-\infty}^{\infty} \frac{f(\tau)}{t - \tau} d\tau,
\]  

(1.3)
and implements a 90 degree phase shifter. The conventional method is to implement a
digital FIR or IIR filter to approximate the Hilbert transform [76]. Analog methods can
be used to compute the Hilbert transform by utilizing a switched capacitor-based IIR
filter [77, 78]. A low frequency analog Hilbert transform to implement 90-degree phase
shift was demonstrated [79] in 0.35\(\mu\)m CMOS but utilizes large silicon area (1.4mm\(^2\))
due to the large RC passives. Utilizing an area-efficient FIR filter would provide the
most optimum implementation for low frequency I/Q extraction.

Extraction of the I/Q vectors enables the implementation of a neural vector analyzer.
This is required to simultaneously analyze both the magnitude and phase of neural sig-
nals as shown in Figure 1.10. After amplification and bandpass filtering of the neural
signals, the magnitude and phase in specific neural bands are analyzed and computed.
The bivariate phase synchronization can then be processed between the two neural sig-
nals. In epilepsy, variation in the phase synchrony is typically associated with seizures.

### 1.3.3 Integrated Circuits for Closed-loop Electrical Neurostimulation

For neural stimulation, similar to neural recording, a large number of channels ensures
interfacing with state-of-the-art electrodes as well as wide coverage and control of the
neural tissue. The recommended stimulation by clinicians to ensure full charge con-
trol and charge balance is a biphasic current stimulation. Studies show that acceptable
stimulus signal specifications are as follows: frequency of 1 to 200Hz; current of 0.1
to 1.2mA; pulse width of 40 to 1000\(\mu\)s; burst duration <5s; and number of bursts of
1-5 [80]. It is important to have control of amplitude, frequency and duty cycle as
stimulation efficacy varies from patient to patient.

Electrical stimulation has shown positive results in reducing the frequency of seizures
in such patients with refractory epilepsy [81], [82]. Typically, the stimulation pulses are applied continuously and periodically, which can result in suboptimal treatment efficacy, shorten the battery life, increase the size of the device and increase the cost of the therapy as additional surgical operations are required for battery replacement [82]. Automated identification of optimal time instances when an electrical stimulus should be applied can help address these issues [7], [12]. In many cases, seizures can be detected prior to the clinical onset of the seizure. It has been widely hypothesized that anticipation of ictal events (i.e., seizures) is critically important for a proper control of seizures [17], [4]. This is based on the assumption and some evidence [83], [84] that it is easier to stop a seizure by electrical stimulation before or early in its development.
than when it has already fully developed.

For example, in epilepsy as explained in Section 1.2.1, fluctuations in phase synchrony in narrow frequency bands are among the best features used for early detection of seizures [24]. Detecting the seizure before it develops is critical in a closed-loop implantable device for automated seizure control. Closed-loop operation yields two major advantages. Firstly, the chances of aborting a seizure are much higher if responsive electrical stimulation takes place before one fully develops [24]. Secondly, closed-loop neural stimulators minimize the overall power dissipation of an implantable device. Neural stimulators must provide large biphasic currents to invoke a neural response and exhibit a much higher power dissipation than neural recording and signal processing units. By only triggering the neural stimulator when a seizure is detected, the power dissipation of a closed-loop implantable device can be significantly reduced. This lengthens the battery life and minimizes tissue damage resulting from the heat generated by the neural stimulator. Thus, to improve the performance of a closed-loop implantable device, a phase synchronization estimation algorithm with high performance (e.g., high true-positive rate and low false-positive rate) is required to be implemented on an integrated circuit within the implant.

The ability to integrate neural recording and neural stimulation channels on the same chip allows for a closed-loop feedback of the neural tissue in which a stimulation is sent to the tissue based on a response from neural activity. Closed-loop SoCs have been demonstrated [85], [52] and [52]. The recent work in [85] integrates 4 neural recording and 10 neural stimulation channels. The closed-loop SoC in [52] integrates 4 neural recording and 4 neural stimulation channels. The work described in [86] integrates 8 neural recording channels with 64 neural stimulation channels. These three closed-loop SoCs have simple signal processing capabilities where they can observe or separate neural spikes or detect energy. The work in [52] and [85] also integrates
a wireless transmitter but suffer from a limited number of recording and stimulation channels. The work in [86] increases the number of stimulation channels, but lacks a wireless transmitter. The signal processing capabilities of these designs are not able to detect seizures and the number of recording and stimulation channels are low when compared to current state-of-the-art (higher than 32). Our research group previously developed and tested in vitro a 128-channel fully differential neural recording and stimulation interface [87] and 256-channel fully differential neural recording and 64-channel biphasic neural stimulation interface [51]. Both these systems lack signal processing required for true closed-loop operation. Integrating a large number of stimulation channels, N, requires N-DACs and N-duty cycle controllers and a large digital memory to store amplitude and duty cycle information leading to designs with large die area. For example, a 100-channel stand alone neural stimulator with in-channel DACs and duty cycle controllers reported in [88] utilizes 25mm$^2$ of die area. Thus, integrating a large number of neural recording and stimulation channels (higher than 32) with advanced signal-processing such as seizure detection has not yet been demonstrated in literature.

### 1.3.4 Integrated Circuits for Wireless Neural Data Transmission

Integrating a wireless transmitter on the IC enables freedom of movement for a patient, reduces artifacts and noise in the recorded data and reduces the possibility of an infection. The requirements of a large number of recording channels (64 or higher) and a wide signal bandwidth (7kHz) translate to a high output data rate (>1Mbps) of the transmitter. As discussed earlier, tissue heating and available inductive power budget severely limit the transmitter output power and date rate.

Design requirements for some commercial implantable medical transmitters are limited power dissipation of 6mW (due to peak current limitations in small implantable batteries), minimum external passive components due to a small form factor (less than
10mm$^2$), reasonable data rates (higher than 20kb/s), high reliability and a 2 meter transmission range [89]. Losses due to the small form factor of the antennas, body tissue absorption and fading are approximately 40-45dB [89]. The FCC allocates the Medical Implant Communication Service (MICS) frequency band (402-406MHz) for wireless implantable medical devices. This band has a 300kHz wide maximum bandwidth and an output power of -16dBm [89]. The industrial, scientific and medical (ISM) radio bands are also frequently used for implantable medical transmitters. These include the 902MHz to 928MHz, the 2.4GHz to 2.4835GHz and the 5.725GHz to 5.875GHz frequency bands and have transmission ranges as high as 10 meters and an allowable output power of 0.5mW [90]. Lower frequencies require larger antennas, whereas higher frequencies have higher losses due to tissue absorption. The optimum frequency band for wireless implantable transmitters was previously determined to be approximately 900MHz [90].

Existing neural SoC designs overcome the bandwidth and power density limitations by transmitting only neural spikes on most or all channels [41], [53] and by utilizing simple low-power transmitter architectures [41], [42] and [91]. Spike detection results in a loss of vital information such as neural oscillations in the gamma band present in patients with epilepsy, schizophrenia, Alzheimer’s disease and other neurological disorders. Simple wireless transmitter architectures for neural recording have shortcomings and are typically not fully integrated. Open-loop VCO designs [41], [42], [55] suffer from temperature sensitivity and frequency drift. A closed-loop transmitter architecture addresses these issues by ensuring the output frequency is locked to a precise reference crystal oscillator. Closed-loop wireless transmitters for neural SoCs were implemented in [53] and [92]. The wireless transmitter in [53] suffers from high power dissipation and requires off-chip components, limiting its ability to operate as an implantable device. The design in [92] consists of only a single neural recording channel.
and also requires off-chip components. Lastly, the SoC in [44] integrates 128 neural recording channels, column-parallel ADCs and FIR filtering with a basic UWB wireless transmitter, resulting in a wireless link that is susceptible to interference and noise making it impractical for a chronically implantable device. This is typically overcome by transmitting data at much higher data rates (>10MB/s). A closed-loop fully integrated wireless transmitter with a large number of recording channels has not yet been demonstrated in literature.

### 1.3.5 State-of-the-art System-on-Chip (SoC) Neural Implants

The current state of the art in neural recording and/or stimulation systems-on-chip (SoCs) is summarized in Table 1.1.

SoCs with closed-loop operation have been demonstrated which combine neural recording, neural signal processing blocks and neural stimulation on a single die [54], [85], [55]. For these designs, the neural recording functionality has been limited to a few recording channels and the signal processing functionality has been limited to simple infinite impulse response (IIR) or FIR filtering. The work demonstrated in [96] and our previous work in [87] and [51] integrate a large number of neural recording and neural stimulation channels. However, these systems lack signal processing required for high-efficacy closed-loop operation, such as neural stimulation triggered as a result of abnormal phase synchrony detection.

Integration of 96 recording channels and in-channel ADCs in 0.13μm CMOS is reported in [95]. The large area and power dissipation constrain the system and make further integration of signal processing, neural stimulation and an RF transmitter difficult. The SoC reported in [42] integrates 32-channels and a wireless transmitter. This design utilizes a large die area (16mm²) without any signal processing or neural stimulation. As well, the wireless transmitter utilized an open-loop, not-fully integrated
Table 1.1: Integrated State-of-the-Art Neural Recording and/or Stimulation SoCs.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Diss. (mW)</td>
<td>0.47</td>
<td>0.375</td>
<td>0.0771</td>
<td>7.05</td>
<td>0.27</td>
<td>0.005</td>
<td>6.0</td>
<td>-</td>
<td>0.230</td>
<td>6.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CMOS Tech. (μm)</td>
<td>0.18</td>
<td>0.35</td>
<td>0.18</td>
<td>0.5</td>
<td>0.18</td>
<td>0.8</td>
<td>0.35</td>
<td>0.18</td>
<td>0.090</td>
<td>0.13</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Area (mm)²</td>
<td>4</td>
<td>10.9</td>
<td>6.25</td>
<td>16.4</td>
<td>4</td>
<td>18</td>
<td>65</td>
<td>25</td>
<td>14.82</td>
<td>25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Supply (V)</td>
<td>1.8</td>
<td>1.5</td>
<td>1.0</td>
<td>3.0</td>
<td>1.8</td>
<td>1.8</td>
<td>3.3</td>
<td>1.8</td>
<td>1.0</td>
<td>1.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># of Record. Chan.</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>32</td>
<td>8</td>
<td>4</td>
<td>128</td>
<td>8</td>
<td>16</td>
<td>96</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Power/chan. (μW)</td>
<td>61</td>
<td>32.8</td>
<td>4</td>
<td>75</td>
<td>9</td>
<td>4.5</td>
<td>11</td>
<td>2.5</td>
<td>14.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gain (dB)</td>
<td>54</td>
<td>51.8-65.7</td>
<td>72</td>
<td>67.8</td>
<td>-</td>
<td>-</td>
<td>57-60</td>
<td>52</td>
<td>40dB</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Noise (μVrms)</td>
<td>-</td>
<td>3.1</td>
<td>1.3</td>
<td>4.4</td>
<td>-</td>
<td>1</td>
<td>4.9</td>
<td>0.91</td>
<td>-</td>
<td>2.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bandwidth (Hz)</td>
<td>-</td>
<td>12k</td>
<td>100</td>
<td>10k</td>
<td>-</td>
<td>100</td>
<td>5k</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CMRR (dB)</td>
<td>-</td>
<td>58</td>
<td>60</td>
<td>134</td>
<td>-</td>
<td>80</td>
<td>90</td>
<td>90</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADC SNDR/res.</td>
<td>35.5</td>
<td>54.7</td>
<td>65</td>
<td>PMW-based</td>
<td>44</td>
<td>-</td>
<td>10-bit</td>
<td>9-bit</td>
<td>10-bit</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Analog Filters</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>FIR/IIR Filters</td>
<td>2x FIR</td>
<td>1x IIR</td>
<td>7x FIR</td>
<td>-</td>
<td>1x FIR</td>
<td>-</td>
<td>1x FIR</td>
<td>64x FIR</td>
<td>1x FIR</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Processor</td>
<td>-</td>
<td>-</td>
<td>Feature Ext.</td>
<td>-</td>
<td>-</td>
<td>Analog</td>
<td>-</td>
<td>Feature Ext.</td>
<td>GP RISC</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Univariate Oper.</td>
<td>Mag</td>
<td>4xSpike</td>
<td>1xMag</td>
<td>-</td>
<td>-</td>
<td>4xMag</td>
<td>Spike</td>
<td>SVM</td>
<td>spectral</td>
<td>cross-corr.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bivariate Oper.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mag. Floor (μVrms)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Closed-loop</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Neural Stimulation</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td># of Stim. Chan.</td>
<td>10</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DAC Res. (bits)</td>
<td>6</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Compliance (V)</td>
<td>-</td>
<td>3.2</td>
<td>-</td>
<td>-</td>
<td>1.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Current Range (mA)</td>
<td>0.1</td>
<td>0-0.0945</td>
<td>-</td>
<td>-</td>
<td>0-0.135</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Duty Cycle Tuning</td>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>6-bit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Wireless Telemetry</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Modulation</td>
<td>FSK</td>
<td>FSK/OOK</td>
<td>FSK</td>
<td>FSK</td>
<td>FSK</td>
<td>-</td>
<td>UWB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Date-rate (Mb/s)</td>
<td>0.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>90</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TX Pow. Diss. (mW)</td>
<td>0.005</td>
<td>0.2</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>1.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
approach making it susceptible to interference, drift and temperature. The SoC reported in [44] integrates 128 neural recording channels with FIR filtering and a UWB transmitter. Similarly, this design has a large die area (65mm²) and power dissipation without integrating any closed-loop capabilities and advanced signal processing.

Neural recording SoCs with multiple channels and advanced signal processing to assist in seizure detection have been demonstrated [48], [94], [93]. A 0.18µm eight-channel EEG acquisition SoC with an on-chip classification processor for seizure detection is reported in [93]. A 90nm SoC which integrates a RISC processor, a FIR filter and other digital signal processing blocks with 16 neural recording channels is demonstrated in [94]. These previously published designs that utilize advanced signal processing [93], [94], [72], [48] operate in open-loop (no stimulation), have few recording channels (less than 16) and have no wireless communication capabilities.

### 1.3.6 CMOS Technology Choice and Scaling

The majority of the designs listed in Table 1.1 are implemented in 0.18µm or older CMOS technology nodes. The designs [94] and [95] are implemented in a 90nm and 0.13µm CMOS technology respectively. The analog frontend of an implantable SoC requires a large number of analog recording channels (higher than 32). Each amplifier requires a low input-referred noise, low-leakage devices and large passives to implement low frequency poles (below 1Hz) and high supply voltages that cannot be achieved through technology scaling to state-of-the-art nodes. Neural stimulation typically has currents as high as 1.2mA to invoke a neural response. Electrode impedances are in the order of several kOhms. This requires a supply voltage of at least 3.3V for proper operation. For these reasons, a more mature CMOS technology is required for implantable devices.

A key requirement for implantable medical devices is reliability as many of these
Figure 1.11: Failure rates from a manufacturer for different technology nodes. The figure is from [2]

devices are life sustaining devices [97] and thus require a mature CMOS process to significantly reduce the risk of failure. Implantable devices require very low failure rates with proper operation of 10 or more years. To meet these constraints a CMOS technology node older than 0.13 \( \mu \)m is required [2] to have reliability higher than 10 years as shown in Figure 1.11.

1.4 Summary of Novel Contributions

For this dissertation, contributions were made at the system-level, the circuit-level and the application-level. The key contributions were submitted to three IEEE journals (one accepted, one being revised and one under review), and presented at six IEEE conferences. The key novelties and contributions in this thesis are listed below.

1.4.1 System-level Contributions

- A wireless SoC for monitoring of epilepsy is demonstrated. The SoC integrates 64 neural recording amplifiers, 64 16-tap mixed-signal FIR filters and a fully integrated PLL-based FSK/OOK ISM-band transmitter. Relevant publications
include a presentation at *IEEE 2011 European Solid-State Circuits Conference (ESSCIRC).*

- A CORDIC-based processor that implements a seizure detector based on computing the magnitude, phase and phase synchrony of neural signals is demonstrated. The processor is benchmarked with offline human seizure data and yields comparable performance to offline software-based seizure detectors. Relevant publications include a manuscript accepted to *IEEE Transactions on Biomedical Circuits and Systems (Volume 5, issue 5)*, and presentations at *IEEE 2010 Biomedical Circuits and Systems Conference (BIOCAS)* and *IEEE 2011 Conference on Neural Engineering.*

- A wireless closed-loop SoC for monitoring and treatment of epilepsy is demonstrated. This SoC integrates 64 neural recording amplifiers, 64 16-tap FIR filters, a CORDIC-based processor for seizure detection, 64 neurostimulators and a UWB transmitter. This work implements the first neural vector analyzer and the first closed-loop neural SoC utilizing phase synchrony. Relevant publications include a presentation at *IEEE 2012 European Solid-State Circuits Conference (ESSCIRC).*

### 1.4.2 Circuit-level Contributions

- A compact, low-power and programmable mixed-signal FIR filter implementation that utilizes the SAR ADC as the multiplier is demonstrated. This mixed-signal FIR filter approach eliminates the need for 512 conventional 8-bit digital multipliers by efficiently implementing the multiplication within the SAR ADC conversion cycle. It was also extended to implement a compact I/Q generator to perform vector processing on low-frequency neural signals.
• A compact chopper-stabilized neural amplifier with a low-distortion high-pass filter is demonstrated. This work was presented in the *IEEE 2011 International Symposium on Circuits and Systems (ISCAS)*.

• An area and power-efficient stimulator for integrated neural recording and stimulation arrays is demonstrated. This work utilizes sharing of the ADC and DAC during the neural recording and stimulation phases to perform amplitude and duty cycle control with minimal area and power overhead. It was presented in the *IEEE 2010 International Symposium on Circuits and Systems (ISCAS)*.

### 1.4.3 Application-level Contributions

• *In vivo* experiments demonstrate online seizure detection and seizure abortion on rodents. The experimental results validate the first reported SoC to perform closed-loop seizure detection based on phase synchrony.

### 1.5 Thesis Organization

Based on the survey of current state-of-the-art neural recording and/or stimulation SoCs as summarized in Table 1.1 researchers have yet to demonstrate a closed-loop SoC that utilizes advanced signal processing (such as high-efficacy seizure detection algorithms). Furthermore, integrating a large number of neural recording and/or stimulation channels (higher than 32) with advanced on-chip signal processing to enable closed-loop operation and a wireless data transmitter (for implantable operation) poses many challenges. These include the limited power density constraints to prevent tissue damage, the total power dissipation constraints to ensure power harvesting is feasible, a prohibitively large silicon area and integration challenges resulting in an SoC combining
sensors with $\mu V$ signals with noisy DSP, stimulation and RF transmitter circuitry.

This thesis highlights the techniques used to overcome the challenges with integrating a large number of neural recording and/or stimulation channels, advanced bivariate signal processing and a wireless transmitter while ensuring the area and power dissipation are minimized. Two 0.13 $\mu$m CMOS prototypes are demonstrated and validated through extensive in vivo animal testing to demonstrate its utility in monitoring and treatment of intractable epilepsy. Figure 1.12 outlines the organization of the thesis. This figure shows an outline of the implantable SoC that is described in this thesis and it highlights the Chapters that correspond to the design and electrical testing of the prototype SoCs (Chapters 2, 3 and 4), the animal testing conducted with the prototype SoCs (Chapter 5) and future work that further extends beyond the scope of this thesis (Chapter 6). The details of each Chapter are described below.

- Chapter 2 presents a 64-channel SoC with a scalable neural recording channel that utilizes an in-channel ADC. With minimal extra hardware and power overhead the architecture makes use of multiplication within the SAR ADC. This removes all digital multipliers from the system and enables the integration of 64 programmable 16-tap FIR filters with minimal compromise on the power dissipation and area of the SoC. The system also has a fully integrated closed-loop wireless transmitter with no off-chip passive elements. The prototype IC fabricated in an IBM 0.13 $\mu$m CMOS technology utilizes 12 mm$^2$.

- Chapter 3 presents a novel low-power VLSI architecture that implements an advanced seizure detection algorithm based on computing the magnitude, phase and phase synchrony of neural signals in different frequency bands. This chapter describes the system-level architecture, RTL implementation and experimental results.
Chapter 4 presents a closed-loop neural recording and stimulation SoC with on-chip seizure detection and wireless telemetry. The SoC integrates three CORDIC processors, a 3.1 to 10.4GHz UWB transmitter, 64 fully differential LNAs with in-channel high-Q switched capacitor bandpass filters, 64 fully configurable neural stimulators and 64 16-tap FIR filters onto a 12mm² IBM 0.13μm CMOS die. This implements a wireless closed-loop 64-channel seizure detector SoC based on phase synchrony. It is the first reported on-chip neural vector analyzer with the ability to process not only the magnitude, but also the phase of the neural signals.

Chapter 5 validates the closed-loop SoC in epilepsy treatment. The experiments are conducted on live rodents, in vivo using two models for epilepsy. The in vivo results are used to cross-validate the SoC with the Matlab SIMULINK model, to validate the SoC as an early seizure detector, and to validate the SoC in closed-loop to abort seizures.

Chapter 6 highlights future work related to the neural interface SoCs described in this thesis. A wireless inductive power and a wireless receiver IC were fabricated. This IC utilizes inductive powering as an energy source. It will provide an energy source to the ICs mentioned in Chapter 2 and Chapter 4 and will allow for wireless experiments on freely moving live animals.
Figure 1.12: Visualization of the thesis organization.
Chapter 2

915-MHz FSK/OOK Wireless Neural Recording SoC with 64 Mixed-Signal FIR Filters

2.1 Introduction

A conceptual view of a fully integrated wireless implantable microsystem for preoperative epilepsy monitoring is shown in Figure 2.1. Two chips are combined in one encapsulated package which connects to an RF antenna, inductive coils, an electrode array and a battery. The first system-on-chip (SoC) (chip 1, presented in this chapter) provides multi-channel neural recording for LFPs and neural spike signals, signal processing and a low-power RF wireless link to the baseband receiver located outside the human body. The second SoC (chip 3, discussed in Chapter 6) includes a wireless inductive power link and a data command link to deliver energy and commands, respectively, to the first SoC. A wireless inductive power link can deliver 86 percent power transmission efficiency across a 1cm distance at 5MHz [98] and can provide 50mW of
Figure 2.1: System diagram of the envisioned implantable wireless neural recording system.

received power [99]. This can be used to power the SoC directly, such as the 7mW SoC described in [100]. It can also provide energy to charge a microbattery such as a 200mg, 10mm \( \times \) 10mm battery that can handle peak currents as high as 12mA [101] or a larger rechargeable 4.5g 18mm \( \times \) 31mm microbattery that can handle peak currents as high as 1.5A [102].

The wireless interface enables the freedom of movement for a patient, reduces artifacts and noise in the recording data and reduces the risk of an infection. The implantability requirement imposes the constraints of a small overall form factor for the device and low power dissipation. Low power dissipation is important in order to avoid thermal damage to the tissue and to minimize the power transmitted to the device inductively. By integrating the neural recording, signal processing and RF transmission functions into a single SoC, both the size and power dissipation of the device can be
significantly reduced.

In this chapter, we present an implantable SoC (chip 1 in Figure 2.1) that densely integrates 64 neural recording amplifiers and ADCs, 64 programmable 16-tap FIR filters, an on-chip digital controller and a fully-integrated closed-loop wireless transmitter on a 4mm×3mm 0.13µm CMOS die. The SoC performs versatile signal filtering individually configurable for each channel to extract information of interest such as beta, gamma or other frequency bands, local-field potentials (LFP) and spikes. It also performs interference suppression, spike detection, precise bandwidth control and a variety of other filtering tasks. A mixed-signal FIR filter implementation eliminates the need for 512 conventional 8-bit digital multipliers by efficiently implementing the multiplication within the ADC conversion cycle. This results in over an order of magnitude savings in the power-area product. In summary, by minimizing both the area and power dissipation, an SoC that successfully integrates a large number of neural recording channels with advanced signal processing and a robust fully integrated wireless link onto a single chip is demonstrated.

The rest of the chapter is organized as follows. Section 2.2 discusses the VLSI architecture of the neural recording SoC. Section 2.3 presents the circuit implementation of the key functional blocks in the SoC. Section 2.4 presents experimental results of the individual blocks, the full system and in vivo characterization in rodents.

### 2.2 System Architecture and VLSI Implementation

The full system VLSI architecture with 64 neural recording amplifiers is shown in Figure 2.2(a). The SoC connects to a 64-electrode ECoG grid or a microelectrode array to record neural signals with frequencies as low as 0.1Hz and as high as 7kHz and with signal amplitudes between 10µV and a few mV. This requires the analog front-end to
have an integrated noise below $10\mu V$ and an ADC resolution of 8 bits.

Each channel contains a programmable mixed-signal 16-tap FIR filter. It has two modes of operation. In the first mode, the amplified signal is digitized by an in-channel ADC. In the second mode, the neural amplifier output is digitized and filtered using an in-channel mixed-signal FIR filter. The FIR filtering mode reuses the same in-channel ADC as an analog-to-digital multiplier to perform the bulk of the FIR computations with a minimal area and power overhead. The digital data are modulated using either Manchester encoded frequency-shift keying (FSK) or on-off keying (OOK) and are sent to the 915MHz (unlicensed ISM-band) PLL-based transmitter. On-chip shift registers are programmed externally and store coefficients for the FIR filters and program modes for the RF transmitter. A single off-chip 14.32-MHz crystal oscillator acts as the reference clock for the RF transmitter and is also used to generate all on-chip clocks for the ADC, FIR filters and data modulation.

Figure 2.2(b) shows the micrograph of the $4 \times 3\text{mm}^2$ wireless neural recording SoC fabricated in a standard 1P8M 0.13$\mu$m CMOS technology. An array of 8x8 neural recording channels and in-channel 8-bit ADCs is organized on a 300$\mu$m pitch grid to be flip-chip bonded to a 8x8 trace carrier or directly to an electrode array. Digital logic to implement the add-and-delay functionality of 64 8-bit FIR filters are split above and below the 64-channel array. The fully integrated RF transmitter and digital control logic are located at the top of the chip.

### 2.3 VLSI Implementation

#### 2.3.1 Mixed-Signal FIR Filter

In the FIR filtering mode a 16-tap transposed FIR filter architecture depicted in Figure 2.3(a) is employed. Conventionally this architecture requires 16 8-bit digital mul-
tipliers for each recording channel. This number can be reduced by half when there is symmetry in the filter coefficients $|M_i| = |M_{15-i}|$ for $i=0,1,...,15$ so only eight filter coefficient absolute values are utilized in the multiplication. We use a bank of eight adjacent channel ADCs to implement this computation in parallel as depicted in Figure 2.3(b). As shown in Figure 2.4(a) the outputs of eight LNAs are multiplexed and each output is sampled by eight ADCs in parallel. Each ADC output is fed to two simple digital sign multipliers in the corresponding two slices of the 16-tap add-and-delay
Figure 2.3: (a) Conventional 16-tap transposed FIR structure. (b) Mixed-signal 16-tap transposed FIR filter structure.

line. Eight add-and-delay lines corresponding to the eight input channels are clocked cyclicly by the on-chip digital controller. The simplified control signals for the ADC and FIR filters are shown in Figure 2.4(b). The SAR logic within the ADC operates at 625kHz and is divided by 11 to sample its input at 56.8kS/s. The ADC sampling clock is further divided by 8 to generate 7.2kHz for the FIR filters. A 3-bit counter output, SELECT, is synchronized with the ADC sampling clock and cycles the output of the neural recording amplifier and its corresponding FIR filter clock.

This architecture is tiled eight times for a total effective number of 64 FIR filters. In this mixed-signal FIR implementation the 512 digital multipliers (8 for each of 64 channels) as depicted in Figure 2.3(a) (for 1-channel) are eliminated from the system, while the ADCs are clocked 8 times faster \((8f_s)\) than the in-channel sampling rate \((f_s)\). The scalable implementation allows for in-channel analog-to-digital conversion of raw recorded data on 64-channels at a ADC sample rate of \(f_s\) per channel, FIR filtering on 8-channels at a ADC sample rate of \(f_s\) per channel or FIR filtering on 64-channels at a ADC sample rate of \(8f_s\) per channel. The implemented 64-channel mixed-signal
Figure 2.4: (a) Schematic of 64 implemented mixed-signal 16-tap symmetric transposed FIR filters. (b) Control signals for FIR filter.
FIR approach adds a 20 percent area and 2X power overhead when compared to an SoC with an in-channel ADC without FIR filters or signal processing. Our approach yields 9 times less area and 1.5 times less power dissipation when compared with a fully digital FIR filter implementation with 512 multipliers for a savings in the power-area product of a factor of 13 (more details are described in Section 2.4.2). The digital switching noise of digital multipliers is also eliminated.

2.3.2 Neural Recording Amplifier

Each LNA in Figure 2.2(a) and Figure 2.4(a) implements a neural recording amplifier. The schematics of the neural recording amplifier and the two OTAs are shown in Figure 2.5. Amplification and filtering are performed over two stages with a total gain of 54dB set by the input to feedback capacitor ratio. The first stage uses a fully differential architecture to ensure a high CMRR, which is important for dense analog-digital SoC implementations. The first stage uses a fully differential folded-cascode OTA with a gain of 60dB. To minimize the flicker noise of the input differential pair M1/M2, large PMOS devices are utilized. To minimize the thermal noise, the input-pair transistors are biased in the sub-threshold region to maximize their $g_m$, while transistors M3/M4 and M5/M6 are biased in the saturation region to minimize their $g_m$. Common-mode feedback (CMFB) is implemented by modulating the tail current of the first stage of the folded-cascode OTA using a continuous-time CMFB amplifier while maintaining the common-mode voltage at 0.6V. The OTA requires a total of 2.4$\mu$A of current which includes 1.6$\mu$A into the input pair, and 0.8$\mu$A into the cascode transistors. The second stage OTA requires a high output swing and thus is implemented using a 2-stage OTA. The single-ended OTA has 60dB of gain and utilizes PMOS input devices. A total of 350nA of current is consumed by the OTA, with 100nA in the input stage and 250nA in the second stage. All OTAs utilize standard thin-oxide transistors.
The first stage has a small output swing and uses two long thick-oxide NMOS pseudo-resistors as feedback elements. Along with $C_2$ they yield a high-pass pole in the sub-Hertz range. It is important to be able to tune to the bandwidth allowing attenuation of the low frequency drift or DC offsets and also to shift the high-pass pole to correct for process variation. The pseudoresistor in the first stage can be made tunable by setting the gate of the MOS transistor to a programmable DC voltage [50] and operating the device in subthreshold. Setting the $V_{GS}$ of the feedback device too low leads to leakage issues and bias drifts often saturating the OTA. The pseudoresistor, as implemented in the first stage, exhibits distortion if large output swings were to be present. Thus it is not suitable for use in the second stage [41], [103].

These issues are addressed by implementing the feedback resistor in the second stage by dynamically adjusting the gate voltages of two MOS transistors using two source-followers. This ensures a constant $V_{GS}$ on the subthreshold feedback devices for
Table 2.1: Transistor sizes of the two OTAs and the feedback element (F.E.) of the neural amplifier and the comparator of the ADC

<table>
<thead>
<tr>
<th>Transistor (OTA1)</th>
<th>W/L (µm)</th>
<th>Transistor (OTA2)</th>
<th>W/L (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{1,2}$</td>
<td>25 $\times$ 2/1.5</td>
<td>$M_{1,2}$</td>
<td>4 $\times$ 0.5/5</td>
</tr>
<tr>
<td>$M_{3,4}$</td>
<td>4 $\times$ 0.5/15</td>
<td>$M_{3,4}$</td>
<td>2 $\times$ 0.5/10</td>
</tr>
<tr>
<td>$M_{5,6}$</td>
<td>4 $\times$ 1/10</td>
<td>$M_5$</td>
<td>2 $\times$ 0.8/1</td>
</tr>
<tr>
<td>$M_{7,8}$</td>
<td>4 $\times$ 0.6/1</td>
<td>$M_6$</td>
<td>2 $\times$ 1.0/3.8</td>
</tr>
<tr>
<td>$M_{9,10}$</td>
<td>4 $\times$ 0.6/1</td>
<td>$M_7$</td>
<td>5 $\times$ 0.8/3</td>
</tr>
<tr>
<td>$M_{11}$</td>
<td>4 $\times$ 1/2</td>
<td>$M_8$</td>
<td>2 $\times$ 2.5/1</td>
</tr>
<tr>
<td>$M_{12}$</td>
<td>1 $\times$ 1/2</td>
<td>$C_f$</td>
<td>1pF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transistor (F.E.)</th>
<th>W/L (µm)</th>
<th>Transistor (Comparator)</th>
<th>W/L (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>1 $\times$ 0.5/15</td>
<td>$M_1$</td>
<td>1 $\times$ 0.5/2</td>
</tr>
<tr>
<td>$M_2$</td>
<td>1 $\times$ 0.5/15</td>
<td>$M_2$</td>
<td>4 $\times$ 2.5/1</td>
</tr>
<tr>
<td>$M_3$</td>
<td>2 $\times$ 2.5/0.5</td>
<td>$M_3$</td>
<td>2 $\times$ 0.5/8</td>
</tr>
<tr>
<td>$M_4$</td>
<td>3 $\times$ 1.0/0.5</td>
<td>$M_4$</td>
<td>4 $\times$ 2.5/1</td>
</tr>
<tr>
<td>$M_5$</td>
<td></td>
<td>$M_5$</td>
<td>2 $\times$ 0.3/5</td>
</tr>
<tr>
<td>$M_6$</td>
<td></td>
<td>$M_6$</td>
<td>2 $\times$ 2.5/1</td>
</tr>
<tr>
<td>$M_7$</td>
<td></td>
<td>$M_7$</td>
<td>1 $\times$ 1/0.24</td>
</tr>
<tr>
<td>$C_{OFF}$</td>
<td></td>
<td></td>
<td>170fF</td>
</tr>
</tbody>
</table>

large output swings as shown in Figure 2.5 (top, right). The resistance can be reduced by increasing the bias current in the source followers and thus increasing the $|V_{GS}|$ of the MOS devices. The area is minimized by utilizing low-threshold devices to allow $|V_{GS}|$ tunability between approximately 0.1V and 0.3V without requiring a wide device. Utilizing both NMOS and PMOS devices ensures at least one device has the proper $|V_{GS}|$ to implement a large resistance across the full output swing of 0.6V. Additionally, compared to the conventional approach [50], this feedback resistance achieves more accurate biasing by guaranteeing a controlled resistance (i.e., with a $|V_{GS}| > 0V$) on the devices in the feedback. A 10-15dB improvement in distortion at sub-Hz frequencies was observed experimentally when using the implemented approach in Figure 2.5 (top, right) over the conventional subthreshold transistor when used in the second stage. This is illustrated by the simulation result shown in Figure 2.6(a) and (b). In this simulation one node of the feedback device is tied to the CM node of 0.6V, while the output node is swept from 0.3V to 0.9V representing the output range of the amplifier. The proposed
feedback device had a $V_{GS}$ of 0.15V and was compared with two PMOS devices in series with a $V_{GS}$ of 0.15V. The proposed feedback device had its high-pass pole shift from 3.4Hz to 20Hz which would introduce distortion at low frequencies (below 20Hz) when the output is low as a result of large output swings. The case with two series PMOS devices had its high-pass pole shift above 6kHz when the output is at full swing resulting in much higher distortion than the proposed case.

All transistor sizes for the neural recording amplifier are listed in Table 2.1.

### 2.3.3 Multiplying SAR ADC

Each channel in the 8x8 array has an 8-bit ADC to provide the most power-efficient and scalable architecture. The in-channel architecture relaxes the settling limitations of the neural recording amplifier and maintains a fixed sample rate per ADC. As introduced in Section 4.2, each channel can be configured in one of the two modes of operation: raw data analog-to-digital conversion, and FIR filtering which is implemented as shown.
CHAPTER 2. WIRELESS NEURAL RECORDING SoC WITH 64 MIXED-SIGNAL FIR FILTERS

Figure 2.7: (a) Sampling phase of multiplying SAR ADC. (b) Hold phase of multiplying SAR ADC.

in Figure 2.4(a). The in-channel ADC is an 8-bit capacitive charge redistribution SAR ADC. The same ADC circuits are reused in both modes. In raw data analog-to-digital conversion mode, the output of the neural amplifier is sent directly to the ADC. The FIR filtering mode requires the ADC to perform analog-digital multiplication as part of the conversion cycle. The successive approximation register (SAR) ADC architecture can be modified to implement such computation with a small area and power overhead.

Analog multiplication within the ADC is introduced by modifying the digital SAR logic to perform multiplication during the sampling phase as shown in Figure 2.7(a). During the sampling phase, the capacitors in the array are disabled or enabled which is equivalent to performing a multiplication by 0 and 1, respectively. The hold phase is similar to that in a conventional SAR ADC as shown in Figure 2.7(b).

The SAR ADC architecture is shown in Figure 2.8 and utilizes a split-capacitor
Figure 2.8: Schematic of the in-channel multiplying SAR ADC and comparator.
array to minimize area and power dissipation. Each unit capacitor is implemented using a (metal-insulator-metal) MIM capacitor with a unit size of 100fF. The split-capacitor value of 70fF was determined through post-layout simulations. The comparator shown in Figure 2.8 utilizes two stages with offset correction followed by a level-shifter and digital buffers. The ADC has an input range of 0.6V and operates from a 1.2V supply. Transistor sizes are listed in Table 4.1.

The ADC mixed-signal multiplication requires only a few extra digital gates in the SAR logic, as shown in Figure 2.8. The bits (bit0, bit1, ..., bit7) are the outputs from the conventional SAR digital controller logic. Multiplication coefficients (m0, m1, ..., m7) are loaded into an in-channel shift register. In the raw data analog-to-digital conversion mode, all the multiplication bits are set to high. In the FIR mode they are programmed to the value of the binary bits of the coefficient $M$. During the sample phase (S=1), each bit at the output of the multiplication logic as shown in Figure 2.8 represents a one-bit binary multiplication by 0 or 1. After the sampling phase (S=0), the ADC operates as a conventional SAR ADC, logically equivalent to the charge-sharing switches being controlled by bits (bit0, bit1,..., bit7). An output register synchronizes the ADC output data. Two sign coefficients (not shown) are also programmed to implement two’s complement multiplication. The outputs are sent to the add-and-delay logic of the FIR filter, as shown in Figure 2.4(a). The ADC dissipates $1.8\mu W$ at 57kS/s from a 1.2V supply.

### 2.3.4 Fully Integrated 915MHz FSK/OOK Transmitter

The digital data are serialized and encoded on-chip and transmitted wirelessly using either on-off keying (OOK) or frequency-shift keying (FSK) modulation at up to 1.5Mb/s. Each data packet is 16-bit, consisting of 10 bits of digitized neural data and a 6-bit channel address. This results in sending 64 channels of FIR-filtered neural data at a data rate of 1.5kS/s per channel or at higher data rates on fewer channels. The closed-loop PLL
transmitter prevents the VCO frequency from drifting allowing for a robust wireless channel without the addition of any off-chip passive components. The full wireless transmitter schematic is shown in Figure 2.9(a). The output power (4 bits), VCO center frequency (4 bits), FSK modulation index (3 bits), PLL loop filter bandwidth (6 bits) and modulation type (1 bit) are programmed by loading an on-chip shift register.

**PFD, Charge Pump, Loop Filter and VCO**

The schematics of the phase-frequency divider (PFD), charge pump and loop filter are shown in Figure 2.9(b). The PFD is implemented using a conventional three-state digital circuit [104]. The charge pump provides a current of $10\mu$A and uses a regulated current mirror to minimize mismatch in the current sources [105]. Thick-oxide devices are used at the output to maximize the output impedance. The on-chip programmable RC loop filter has 3-bit tuning for both the capacitors and the resistor to allow for various PLL bandwidths and 50 degree phase margin in the PLL loop. The capacitor $C_1$ can be programmed from 50pF to 200pF, $C_2$ from 2.5pF to 10pF, and $R$ from 25k$\Omega$ to 200k$\Omega$. For FSK modulation, the bandwidth of the PLL was set to 50kHz.

The voltage-controlled oscillator (VCO) was implemented using a cross-coupled architecture [106] with a 12.6nH on-chip spiral inductor in the thick copper top-level metal. A large varactor implemented using a P+/N-junction diode was connected to the control voltage of the VCO to lock the PLL to 916.3MHz. Additional programmable MIM capacitors (labeled as 3.1pF in Figure 2.9) were added to the tank to allow 3-bit binary tuning of the VCO center frequency to account for process variation. Two OTA buffers and digital inverters within the PLL loop convert the differential VCO output to a rail-to-rail level that is sent to the frequency divider. The differential VCO output ($VCO_m$ and $VCO_p$) is sent to the power amplifier.

To prevent low frequency signals in the data to be filtered out by the PLL, the data
Figure 2.9: (a) Block diagram of the fully integrated FSK/OOK wireless transmitter. (b) Schematic of the phase-frequency detector, charge pump and loop filter. (c) Schematic of the static-CMOS divide by 64. (d) Schematic of the output driver and power amplifier.
is modulated with a clock operating at twice the data rate to implement Manchester-encoded FSK. The Manchester-encoded data is connected to the smaller NMOS VCO varactors as shown in Figure 2.9(b). They consist of binary-weighted transistors M4, M5 and have 3-bit programming to adjust the FSK modulation index. The bandwidth of the PLL must be significantly lower than the data bit-rate to prevent the data from being filtered out by the PLL [107]. Transistor sizes for the VCO are shown in Table 2.2.

**Frequency Divider and Power Amplifier**

The VCO output frequency is divided by 64 and the output of the divider is then fed back to the PFD. As shown in Figure 2.9(c), the divider uses six cascaded divide-by-2/3 stages. The division ratio of each stage is set to 2 which locks the division ratio of the six cascaded dividers to 64. Static CMOS logic is used operating at up to 1.5GHz in the slowest corner (in post-layout simulation), allowing for enough margin to function with the supply voltages as low as 0.8V (experimentally confirmed).

The differential output of the VCO is buffered by an OTA with resistive degeneration and sent to the power amplifier as shown in Figure 2.9(d). The interface between the VCO buffer and the power amplifier is AC-coupled which allows the input of the power amplifier to be adjusted for class-A or AB operation. The power amplifier utilizes a 12.6nH on-chip spiral inductor and a capacitive matching network to match to a 50-Ohm antenna. The output transistor size is programmed by enabling or disabling output transistor fingers with binary weighted widths through the cascode transistors by signals \( P_0 \) to \( P_3 \). This allows the output power to be adjusted from -20dBm to 0dBm in 16 levels in FSK mode. For OOK modulation, the PLL is locked to the carrier frequency and the cascode transistors of the power amplifier are turned on or off. For OOK modulation, the PLL bandwidth can be made larger as it does not have to be limited as in the FSK mode. Transistor sizes of the power amplifier are listed in Table 2.2.
Table 2.2: Transistor sizes of the VCO and power amplifier

<table>
<thead>
<tr>
<th>Transistor</th>
<th>W/L (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{0,1}$</td>
<td>$15 \times 2/0.12$</td>
</tr>
<tr>
<td>$M_{2,3}$</td>
<td>$45 \times 2/0.12$</td>
</tr>
<tr>
<td>$M_{4,5}$</td>
<td>$3T$ binary weighted bank, $(2, 4, 8) \times 2.5/0.12$</td>
</tr>
<tr>
<td>$M_{6}$</td>
<td>$40 \times 2/0.12$</td>
</tr>
<tr>
<td>$M_{7}$</td>
<td>$20 \times 2/0.12$</td>
</tr>
<tr>
<td>$M_{8}$</td>
<td>$10 \times 2/0.12$</td>
</tr>
<tr>
<td>$M_{9}$</td>
<td>$6 \times 2/0.12$</td>
</tr>
</tbody>
</table>

2.4 Experimental Results

The simulation results and the PCB test board is described in the Appendix.

2.4.1 Analog Front-end

The experimentally measured amplitude frequency response, total harmonic distortion (THD) and the input referred noise of the neural recording channel are shown in Figure 2.10. Figure 2.10(a) shows the high-pass corner frequency being adjusted from approximately 0.1Hz to 1Hz by programming the bias current of the source follower and adjusting the $V_{GS}$ of M3 and M4 in Figure 2.5 (top, right). Figure 2.10(b) shows the frequency response of eight different channels across the chip showing a voltage gain spread between 53.3dB and 54dB. Figure 2.10(c) shows the experimentally measured input-referred noise from eight different channels across the chip. The integrated noise between 10Hz and 5kHz is between $6.0\mu V_{rms}$ and $6.9\mu V_{rms}$ (at room temperature), resulting in an NEF of approximately 7.2. The experimentally measured common-mode rejection ratio (CMRR) across eight channels between 1Hz and 1kHz is at least 75dB.

The THD was measured with an amplifier input amplitude of 1.4mV (output of 0.7V) for different input frequencies as shown in Figure 2.11(a). A distortion of approximately 0.27 percent (-51.3dB) at frequencies above 100Hz was observed and a worst-case distortion of 3.2 percent (-29.9dB) at 0.2Hz. Tunability of the high-pass pole
Figure 2.10: Experimentally measured (a) amplitude response of the analog front-end showing tuneability of the high-pass pole, (b) frequency response of eight different channels and (c) input-referred noise of eight different channels.
Figure 2.11: (a) Experimentally measured THD of the analog front-end. (b) Experimentally measured high-pass filter frequency response for three different values of $I_{HP}$ on two different channels on the chip. Experimentally measured output waveforms for a $1\text{mV}_{\text{pk-pk}}$ 1Hz sinusoid input comparing (c) conventional and (d) proposed feedback elements performance.

is shown in Figure 2.11(b) for two different neural recording amplifiers. The source follower current, $I_{HP}$, was set to 25nA, 50nA and 125nA. The 3dB high-pass cutoff frequencies between the two channels are in close agreement. The proposed feedback element yielded up to a 14dB improvement in linearity at low frequencies as compared to the conventional pseudoresistor feedback [50]. This linearity improvement is also visible in the time domain as shown in Figures 2.11(c),(d). Figure 2.11(c) shows the distorted output of the neural amplifier at a frequency of 1Hz when utilizing the conventional feedback element. The less distorted output of the neural amplifier with the implemented feedback element is shown in Figure 2.11(d).
Figure 2.12: (a) Experimentally measured FFT of an in-channel ADC output for 1kHz sinusoid input at 54kS/s. (b) Experimentally measured ENOB vs. input frequency for an in-channel ADC.
Figure 2.13: Ideal and experimentally measured frequency responses of various programmable FIR filter configurations at 7.1kSps.
Figure 2.14: (a) Experimentally measured output spectrum of the FIR filter. (b) Experimentally measured SNDR vs. input amplitude with a 100Hz sinusoid input to the FIR filter and to the FIR filter including the analog front-end.
CHAPTER 2. WIRELESS NEURAL RECORDING SoC WITH 64 MIXED-SIGNAL FIR FILTERS

2.4.2 ADC and Mixed-signal FIR Filter

The experimentally measured ADC performance is shown in Figures 2.12(a) and (b) with an ENOB of 7.8 and close to 7 bits over the full Nyquist bandwidth when sampled at 56kS/s. The performance was consistent across all on-chip ADCs. The figure-of-merit of the ADC is 140fJ/step.

Four different 16-tap FIR filter configurations were programmed on-chip and clocked at 7.1kS/s (two different low-pass, one high-pass, and one band-pass filter). The results shown in Figures 2.13(a) through (d) demonstrate a good match between the experimental result and the ideal filter response in Matlab. The FFT of a FIR filter output (programmed as a low-pass filter) sampled at 7.2kS/s is shown in Figure 2.14(a). It yields an SNDR of 46.5dB and an SFDR of 56.9dB. Both the SNDR for the stand-alone FIR filter and the SNDR for the FIR filter including the analog front-end is shown in Figure 2.14(b). The SNDR for different input amplitude levels for a 100Hz sinusoid is shown for the two cases. A 6dB drop in SNDR (46.5dB vs. 40dB) is observed when the analog front-end is utilized due to the additional in-band flicker noise, thermal noise and distortion of the amplifier.

Area and Power Comparison

A comparison of the mixed-signal FIR filter of this Chapter with previously published 12-16 tap analog and digital FIR filters with 7-8 bit resolution is listed in Table 2.3. The area-optimized digital FIR filters [58] implemented in a 90nm CMOS technology utilize the distributed arithmetic (DA) or look-up table (LUT) implementations to reduce the area of the FIR filter. This approach yields an area much smaller than conventional FIR filters using digital multipliers [58]. The analog FIR filters [59] and [61] require more area than the digital approaches in [58] but include a built-in ADC. The FIR filter approach covered in this Chapter has an overall equivalent area as the optimized digital
Table 2.3: Comparison of Published FIR filters.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>FIR Filter Type</th>
<th>No. of Taps</th>
<th>Res. (bits)</th>
<th>CMOS Tech.</th>
<th>Area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[58] (no ADC)</td>
<td>Digital LUT</td>
<td>16</td>
<td>8</td>
<td>90nm</td>
<td>0.033</td>
</tr>
<tr>
<td>[58] (no ADC)</td>
<td>Digital DA</td>
<td>16</td>
<td>8</td>
<td>90nm</td>
<td>0.04</td>
</tr>
<tr>
<td>[61] (includes ADC)</td>
<td>Analog</td>
<td>16</td>
<td>7</td>
<td>350nm</td>
<td>0.13</td>
</tr>
<tr>
<td>[60] (no ADC)</td>
<td>Analog</td>
<td>12</td>
<td>7</td>
<td>130nm</td>
<td>0.68</td>
</tr>
<tr>
<td>[59] (includes ADC)</td>
<td>Analog</td>
<td>16</td>
<td>8</td>
<td>45nm</td>
<td>0.14</td>
</tr>
<tr>
<td>This Work (no ADC)</td>
<td>Mixed</td>
<td>16</td>
<td>8</td>
<td>130nm</td>
<td>0.02</td>
</tr>
<tr>
<td>This Work (includes ADC)</td>
<td>Mixed</td>
<td>16</td>
<td>8</td>
<td>130nm</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 2.4: Comparison of Different Implementations for 64 16-tap FIR filters.

<table>
<thead>
<tr>
<th>Spec.</th>
<th>64x8 Digital Multipliers</th>
<th>*64x1 Digital Multipliers</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Diss. of Mult.</td>
<td>512×P_{Mult}</td>
<td>64×8×P_{Mult}</td>
<td>0</td>
</tr>
<tr>
<td>Power Diss. of ADC</td>
<td>64×P_{ADC}</td>
<td>64×P_{ADC}</td>
<td>64×8×P_{ADC}</td>
</tr>
<tr>
<td>Power Diss. of Add./Reg.</td>
<td>64×P_{Add,reg}</td>
<td>64×P_{Add,reg}</td>
<td>64×P_{Add,reg}</td>
</tr>
<tr>
<td>Area of Mult.</td>
<td>512×A_{Mult.}</td>
<td>64×A_{Mult.}</td>
<td>0</td>
</tr>
<tr>
<td>Area of Add./Reg.</td>
<td>64×A_{Add,reg}</td>
<td>64×A_{Add,reg}</td>
<td>64×A_{Add,reg}</td>
</tr>
<tr>
<td>Power Diss.</td>
<td>512×P_{Mult}+64×P_{ADC}</td>
<td>64×8×P_{Mult}+64×P_{ADC}</td>
<td>64×8×P_{ADC}</td>
</tr>
<tr>
<td></td>
<td>+64×P_{Add,reg}</td>
<td>+64×P_{Add,reg}</td>
<td>+64×P_{Add,reg}</td>
</tr>
<tr>
<td>Est. Power Diss. (mW)</td>
<td>0.560</td>
<td>0.56</td>
<td>0.400</td>
</tr>
<tr>
<td>Total Area</td>
<td>512×A_{Mult}+64×A_{Add,reg}</td>
<td>64×A_{Mult}+64×A_{Add,reg}</td>
<td>64×A_{Add,reg}</td>
</tr>
<tr>
<td>Est. Area (mm²)</td>
<td>13.2</td>
<td>2.95</td>
<td>1.5</td>
</tr>
<tr>
<td>Power×Area</td>
<td>7430</td>
<td>1700</td>
<td>600</td>
</tr>
</tbody>
</table>

Note: \( A_{Mult.} = 0.023mm^2, A_{Add,reg} = 0.023mm^2, 8×P_{ADC} = 1.6\mu W, P_{Add,reg} = 4.6\mu W, P_{Mult} = 0.5\mu W \)

*Additional area and power dissipation required for memory.

approaches in [58], but this work also includes an ADC and is implemented in an older technology node.

An analysis comparing the FIR filter approach of this Chapter with the conventional digital FIR filter implementations is described in Table 2.4. Compared with an approach of 512 digital multipliers (8 per 16-tap FIR filter), our approach yields a 13 times lower power-area product. Compared with an approach of 64 digital multipliers (one per 16-tap FIR filter), where the multiplier is clocked 8× higher, our approach yields a 3 times lower power-area product. This approach requires additional memory logic to store the incremental multiplications.

The neural SoC in [108] contains signal processing which includes a digital 22-tap FIR filter in 0.18\mu m CMOS. The area utilized by this FIR filter is 0.65mm². This
implementation requires 12 times the area of the FIR filter approach described in this thesis, after assuming a 50 percent scaling due to process and after scaling the filter from 22 taps to 16 taps. The 22-tap FIR filter (scaled to 16-taps) dissipates $7\mu W$ per channel. Our approach dissipates $6.2\mu W$ which also includes the power dissipated by the ADC.

**Precision Comparison**

The dynamic performance of the 16-tap mixed-signal FIR filter was compared with conventional digital approaches as shown in Figure 2.15. For the comparison a 16-tap bandpass FIR filter was selected and Matlab SIMULINK was used for simulation. The conventional approach is to use an 8-bit ADC followed by an 8-bit digital FIR filter. In our proposed mixed-signal FIR filter, the bank of multiplying ADCs scale the input with a digital coefficient and thus will degrade the dynamic performance. Compared with the conventional approach, the proposed mixed-signal FIR filter yields a worst-case drop of 15dB in dynamic performance when the input is at the minimum loading for the ADC as shown in Figure 2.15. At the maximum input range of the ADC a difference in dynamic performance of approximately 5dB occurs. A comparison is also made with a case in which a 7-bit ADC is followed by an 8-bit FIR filter. In this case, the dynamic performance closely matches the proposed mixed-signal FIR filter as shown in Figure 2.15. Overall, the proposed mixed-signal FIR filter has a precision equivalent to a 1-bit drop over the conventional digital approach. In this design, the input-referred noise of the neural recording channel is approximately $5\mu V_{rms}$ and the voltage gain is 500V/V. This corresponds to a noise floor of approximately -40dB below the full scale of the ADC. In terms of precision, this limit will reduce the benefit of using a higher resolution digital FIR filter over the proposed approach.
Figure 2.15: SIMULINK simulation comparing the dynamic performance between an 8-bit ADC with a 16-tap 16-bit digital FIR filter, the proposed 16-tap 8-bit multiplying ADC mixed-signal FIR filter and a 7-bit ADC followed by a 16-tap 16-bit digital FIR filter.
Scalability

For this design, 16-taps was selected for the length of each FIR filter. Each channel requires eight parallel MADCs. As mentioned in Section 2.3 each MADC yields two multiplies in a transposed symmetric FIR filter implementation. For this mixed-signal FIR filter architecture, it is possible to scale the design by utilizing more parallel MADCs per channel to increase the length of the FIR filter and improve the FIR filters selectivity. Increasing the FIR filter length by utilizing more parallel MADCs per channel impacts the sample rate, area and power dissipation. Maintaining the same ADC sample rate per channel decreases the FIR filter sample rate due to the cyclic timing of the architecture. The area and power dissipation also increase as more digital adders and registers need to be integrated onto the SoC. Based on experimental results and area utilization in this prototype SoC, the predicted sample rate, area and power dissipation for 64 FIR filters with respect to the number of taps for the FIR filter is shown in Figure 2.16(a), (b) and (c), respectively.

2.4.3 915MHz FSK/OOK Transmitter

Experimental results for the wireless transmitter are shown in Figure 2.17. Figure 2.17(a) shows the output spectrum of the FSK transmitter with 1.2Mbps Manchester-encoded input data. Figure 2.17(b) shows transmitted and received data at a distance of 1m. Figure 2.17(c) shows OOK-modulated data on an oscilloscope and Figure 2.17(d) shows transmitted and received OOK data from a distance of 1m. The wireless link demonstrated robust performance at distances up to 10m.

The phase noise and output spectrum of the full transmitter at the output of the power amplifier with a 50 Ohm load are shown in Figures 2.18(a) and (b) respectively. The phase noise at 1MHz offset is -123dBc/Hz, the integrated jitter over a 10MHz bandwidth is 1.06 degrees and the reference spur (14.33MHz) is -62dB below the fun-
Figure 2.16: Scalability of the proposed FIR filter length with respect to (a) sample rate, (b) area and (c) power dissipation.

Implantable antennas for wireless neural recording operating between 400MHz to 1.5GHz have been described [111], [112] and [113]. The 1.5GHz patch antenna de-
Figure 2.17: Experimentally measured (a) spectrum of FSK-modulated data at a 1m distance, (b) transmitted and received data from 1m distance at 1.2Mbps using FSK modulation, (c) OOK-modulated data into 50Ω termination resistor and (d) transmitted and received data from a 1m distance at 10kb/s using OOK modulation.
Figure 2.18: (a) Experimentally measured phase noise of the RF transmitter. (b) Experimentally measured output spectrum showing the reference spur of the RF transmitter.
scribed in [111] has dimensions of 25x25x0.5mm. The 915MHz ISM-band partially-folded planar antenna has dimensions of 26x19x4.4mm. An implantable transmitter integrates a 2.4GHz omnidirectional antenna and a 0.18µm transmitter IC with dimensions of 8x9x2mm. As described in Chapter 1 the losses due to the small form factor of these antennas, body tissue absorption at 900MHz and fading associated for wireless biomedical implants are approximately 40-45dB [89]. Experimental measurements for the FSK transmitter and receiver have a BER of less than $10^{-5}$ for a 1m distance with a -20dBm output power. At a 10cm distance, the free-space loss decreases by 20dB and by increasing the output power of the power amplifier to 0dBm, this can account for another 20dB loss, equaling the 40dB loss due to the antennas, body tissue absorption and fading.

The full transmitter dissipates 3.7mW and 6.6mW in FSK-mode for an output power of -20dBm and 0dBm, respectively and 5mW for 0dBm OOK modulation. The experimentally measured output power and DC power of the transmitter with a 1.0V and 1.2V supply for different power amplifier program settings are shown in Figures 4.25(a) and (b).

A comparison to other ISM-band FSK RF transmitters is displayed in Table 2.5.
Chapter 2. Wireless Neural Recording SoC with 64 Mixed-Signal FIR Filters

Figure 2.20: Experimentally measured data wirelessly received from the chip at 1m distance (including the full signal path with analog front-end, FIR filter and FSK RF TX).

Figure 2.21: Experimentally measured (a) TX output power and (b) DC power for the RF TX when connected to a 50Ω termination resistance for different power-levels.

Table 2.5: Comparison of ISM-band FSK transmitters.

<table>
<thead>
<tr>
<th>Spec.</th>
<th>[114]</th>
<th>[109]</th>
<th>[110]</th>
<th>[115]</th>
<th>THIS WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS Tech.</td>
<td>0.18μm</td>
<td>0.18μm</td>
<td>0.13μm</td>
<td>0.25μm</td>
<td>0.13μm</td>
</tr>
<tr>
<td>Supply (V)</td>
<td>1.5</td>
<td>1.8</td>
<td>1.5</td>
<td>1.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>402</td>
<td>900</td>
<td>915</td>
<td>433</td>
<td>915</td>
</tr>
<tr>
<td>Data Rate (Mb/s)</td>
<td>1.5</td>
<td>1.0</td>
<td>0.05</td>
<td>0.1</td>
<td>1.5</td>
</tr>
<tr>
<td>Phase Noise @10kHz Offset</td>
<td>-88.6</td>
<td>-</td>
<td>-</td>
<td>-75</td>
<td>-91.5</td>
</tr>
<tr>
<td>Phase Noise @100kHz Offset</td>
<td>-85.8</td>
<td>-</td>
<td>-97</td>
<td>-</td>
<td>-93.8</td>
</tr>
<tr>
<td>Phase Noise @1MHz Offset</td>
<td>-110.1</td>
<td>-107.9</td>
<td>-</td>
<td>-</td>
<td>-122.8</td>
</tr>
<tr>
<td>Maximum Output Power (dBm)</td>
<td>-9</td>
<td>-14.5</td>
<td>-6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DC Power (mW)</td>
<td>19.5</td>
<td>20.7</td>
<td>2.7</td>
<td>5.0</td>
<td>6.6</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>1.6</td>
<td>3.0</td>
<td>1.5</td>
<td>4.8</td>
<td>0.42</td>
</tr>
</tbody>
</table>
2.4.4 Application Example: Noise Suppression and Spike Detection

The full neural recording channel with the FIR filter programmed to suppress high frequency noise was experimentally characterized as shown in Figures 2.22(a) and (b). In Figure 2.22(a), the input to the amplifier consists of a 20Hz sinusoid with an amplitude of 250\(\mu\)V and an additional high frequency noise at 800Hz with a larger amplitude of 500\(\mu\)V. The FIR filter was programmed for low-pass filtering. It removes the 800Hz interference, resulting in the output depicted in Figure 2.22(b) showing the amplified and FIR filtered 20Hz signal.

The functionality of the analog front-end, ADC and FIR filters was verified with pre-recorded neural activity of epileptic seizure events. Seizure-like events were induced in an intact hippocampus from Wilstar rats by immersing it in a low-Mg2+ solution. The signals were recorded with a conventional bench-top low-noise amplifier and then programmed onto an arbitrary waveform generator with the original signal as shown in Figure 2.23(a). The input signal was then contaminated with 60Hz interference with a much larger amplitude than that of the pre-recorded signal as shown in Figure 2.23(b) and applied to one of the 64 channels on the chip. Threshold-based spike-detection for these two cases would yield a high false-positives rate. The FIR filter was programmed with the bandpass filter response shown in Figure 2.13(d). Figure 2.23(c) shows the output after the FIR filtering with the neural activity clearly visible and the 60Hz interference removed resulting in threshold-based spike detection having fewer false positives (depicted with the dashed line).

2.4.5 Power Dissipation and Area Breakdown

The power dissipation and area breakdown of the full neural recording SoC are summarized in Figure 2.24. The neural recording SoC dissipates 0.8mW for the analog front-end, ADCs and biasing (includes all LNA, ADC, RF bias currents), 0.53mW for
Figure 2.22: Example of experimentally measured signal band separation using full recording channel (with LP FIR filter).

Figure 2.23: Example of experimentally measured spike detection using full recording channel (with BP FIR filter).
the digital controller and the adders/registers to implement the FIR filters operating at a sampling rate of 7.1kS/s. The RF transmitter dissipates 3.7mW in FSK mode when operating the power amplifier with a -20dBm output power. The total system power dissipation is 5.03mW from a 1.2V supply. When using a 1.0V supply for the digital and RF transmitter, the total power dissipation is reduced to 4.17mW at the expense of requiring a second supply voltage. The FIR filters can be used with additional logic to detect channels with little neural activity and the RF transmitter can be shut off to further reduce the power dissipation. In terms of the integration area, the 64-channel analog front-end and ADCs utilize the majority of the die area. The 64 FIR filters add only an additional twenty percent area overhead to the SoC.

A summary of the experimental results is given in Table 2.6. A comparison with other wireless neural recording SoCs is given in Table 2.7. This work demonstrates the highest level of integration among recently published state-of-the-art designs by combining 64 recording channels, 64 16-tap FIR filters and a fully integrated closed-loop wireless transmitter without a significant compromise on area or power dissipation.

2.4.6 In Vivo Neural Recording and FIR Filtering

The SoC was validated in on-line in vivo experiments in a population of four freely moving rats. Three depth electrodes were implanted into the hippocampus of each rat with two electrodes connected to the inputs of two channels of the neural recording SoC and one electrode acting as a reference electrode for the SoC. The amplifier was programmed for a bandwidth of 0.1Hz to 5kHz. The FIR filter was programmed to implement a bandpass response between 5Hz and 300Hz to further remove low frequency drift and high frequency noise. The pharmacological agent gamma-butyrolactone (GBL) was used to invoke a non-convulsive epileptic seizure in a rat.
Table 2.6: Summary of experimental results

<table>
<thead>
<tr>
<th>System:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>1P8M 0.13µm</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>1.2V</td>
</tr>
<tr>
<td>Die Dimensions</td>
<td>4.0mm × 3.0mm</td>
</tr>
<tr>
<td>Area per Channel</td>
<td>300µm × 300µm</td>
</tr>
<tr>
<td>Number of Recording Channels</td>
<td>64</td>
</tr>
<tr>
<td>On-Chip Memory</td>
<td>1kb</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>5.03mW</td>
</tr>
<tr>
<td>Power Density</td>
<td>0.41mW/mm²</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recording Channel:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Recording Channels</td>
<td>64</td>
</tr>
<tr>
<td>Gain</td>
<td>54-60dB</td>
</tr>
<tr>
<td>Input-Referred Noise</td>
<td>6.5µV</td>
</tr>
<tr>
<td>CMRR</td>
<td>75dB</td>
</tr>
<tr>
<td>THD @ 1kHz</td>
<td>-51dB</td>
</tr>
<tr>
<td>Power Dissipation of LNA</td>
<td>4.5µW</td>
</tr>
<tr>
<td>NEF of LNA</td>
<td>7.2</td>
</tr>
<tr>
<td>ADC SNDR</td>
<td>48.5dB</td>
</tr>
<tr>
<td>ADC ENOB</td>
<td>7.8-bits</td>
</tr>
<tr>
<td>LNA+FIR SNDR</td>
<td>40dB</td>
</tr>
<tr>
<td>ADC Sample Rate</td>
<td>57kS/s</td>
</tr>
<tr>
<td>Power Dissipation of ADC</td>
<td>1.8µW</td>
</tr>
<tr>
<td>ADC Figure-of-Merit</td>
<td>140fJ/step</td>
</tr>
<tr>
<td>Power Diss. (LNA+ADC)</td>
<td>6.3µW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signal Processing:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FIR filters</td>
<td>64</td>
</tr>
<tr>
<td>FIR filter type</td>
<td>Symmetric 16-tap</td>
</tr>
<tr>
<td>Coefficient Resolution</td>
<td>8-bit signed</td>
</tr>
<tr>
<td>Output Resolution</td>
<td>12-bits</td>
</tr>
<tr>
<td>FIR SNDR</td>
<td>46.5dB</td>
</tr>
<tr>
<td>FIR SFDR</td>
<td>56.9dB</td>
</tr>
<tr>
<td>FIR Sample Rate (64-channels)</td>
<td>7.1kS/s</td>
</tr>
<tr>
<td>FIR Sample Rate (8-channels)</td>
<td>57kS/s</td>
</tr>
<tr>
<td>Power Diss. (LNA+ADC+FIR)</td>
<td>14.5µW/Channel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RF Transmitter:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Frequency</td>
<td>915MHz</td>
</tr>
<tr>
<td>Reference Frequency</td>
<td>14.3MHz</td>
</tr>
<tr>
<td>Closed-loop Modulation</td>
<td>FSK and OOK</td>
</tr>
<tr>
<td>Data Rate</td>
<td>1.5Mbps</td>
</tr>
<tr>
<td>Phase Noise</td>
<td>-123dB/Hz @ 1MHz</td>
</tr>
<tr>
<td>Largest Spur</td>
<td>-62dB</td>
</tr>
<tr>
<td>Output Power</td>
<td>-20dBm-0dBm</td>
</tr>
<tr>
<td>DC Power</td>
<td>3.7mW to 6.6mW</td>
</tr>
<tr>
<td>Energy Efficiency</td>
<td>2.5-4.4 (nJ/bit)</td>
</tr>
</tbody>
</table>
Table 2.7: Integrated Wireless Neural Recording SoCs

<table>
<thead>
<tr>
<th>Spec.</th>
<th>[41]</th>
<th>[53]</th>
<th>[42]</th>
<th>[44]</th>
<th>[95]</th>
<th>[55]</th>
<th>[54]</th>
<th>[92]</th>
<th>THIS WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Diss. (mW)</td>
<td>13.5</td>
<td>17.2</td>
<td>7.05</td>
<td>6.0</td>
<td>6.5</td>
<td>0.375</td>
<td>0.27</td>
<td>0.5</td>
<td>5.03</td>
</tr>
<tr>
<td>CMOS Tech. (µm)</td>
<td>0.5</td>
<td>0.35</td>
<td>0.5</td>
<td>0.35</td>
<td>0.13</td>
<td>0.35</td>
<td>0.18</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Area (mm)^2</td>
<td>27.3</td>
<td>8.4</td>
<td>16.3</td>
<td>63.4</td>
<td>25</td>
<td>10.9</td>
<td>2.7</td>
<td>2.5</td>
<td>12</td>
</tr>
<tr>
<td>Supply (V)</td>
<td>3.3</td>
<td>3.0</td>
<td>3.0</td>
<td>3.3</td>
<td>1.2</td>
<td>3.0</td>
<td>1.8</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td># of Recording Chan.</td>
<td>100</td>
<td>16</td>
<td>32</td>
<td>128</td>
<td>96</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>Power/channel (µW)</td>
<td>38</td>
<td>9</td>
<td>75</td>
<td>11</td>
<td>-</td>
<td>32.8</td>
<td>9</td>
<td>75</td>
<td>6.3</td>
</tr>
<tr>
<td>Gain (dB)</td>
<td>60</td>
<td>60</td>
<td>68-78</td>
<td>57-60</td>
<td>-</td>
<td>51.8-65.7</td>
<td>-</td>
<td>38.3</td>
<td>54-60</td>
</tr>
<tr>
<td>Noise (µV_rms)</td>
<td>5.1</td>
<td>3.1</td>
<td>9.3</td>
<td>4.9</td>
<td>2.2</td>
<td>3.1</td>
<td>-</td>
<td>2.0</td>
<td>6.5</td>
</tr>
<tr>
<td>ADC</td>
<td>1xSAR</td>
<td>1xSAR</td>
<td>PWM</td>
<td>1xSAR</td>
<td>96xSAR</td>
<td>1xSAR</td>
<td>1xLog.</td>
<td>1xSAR</td>
<td>64xSAR</td>
</tr>
<tr>
<td>ADC Resolution (bits)</td>
<td>9b</td>
<td>8b</td>
<td>N/A</td>
<td>6-9b</td>
<td>10b</td>
<td>10b</td>
<td>8b</td>
<td>8b</td>
<td>8b</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>Spike</td>
<td>Spike</td>
<td>None</td>
<td>Spike</td>
<td>None</td>
<td>1xIIR</td>
<td>1x22tap-FIR</td>
<td>None</td>
<td>64x16tap-FIR</td>
</tr>
<tr>
<td>Programmable</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RF Transmitter</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RF Carrier (GHz)</td>
<td>0.433</td>
<td>0.433</td>
<td>0.915</td>
<td>3.1</td>
<td>-</td>
<td>0.433</td>
<td>-</td>
<td>0.450</td>
<td>0.915</td>
</tr>
<tr>
<td>Modulation</td>
<td>FSK</td>
<td>FSK</td>
<td>FSK/OOK</td>
<td>UWB</td>
<td>-</td>
<td>FSK</td>
<td>-</td>
<td>FSK</td>
<td>FSK/OOK</td>
</tr>
<tr>
<td>Data rate (Mb/s)</td>
<td>0.330</td>
<td>1.25</td>
<td>0.708</td>
<td>90</td>
<td>-</td>
<td>N/A</td>
<td>-</td>
<td>0.100</td>
<td>1.5</td>
</tr>
<tr>
<td>PLL</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fully Int. TX</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RF P_out(dBm)</td>
<td>N/A</td>
<td>0</td>
<td>-22</td>
<td>N/A</td>
<td>-</td>
<td>N/A</td>
<td>-</td>
<td>-16</td>
<td>-20 to 0</td>
</tr>
<tr>
<td>RF P_DC(mW)</td>
<td>4.0</td>
<td>16.6</td>
<td>3.3</td>
<td>1.6</td>
<td>-</td>
<td>0.2</td>
<td>-</td>
<td>0.4</td>
<td>3.7 to 6.6</td>
</tr>
</tbody>
</table>
Before injecting the rat with the drug, no seizure activity is recorded as shown in Figure 5.3(a). Ten minutes after injecting the rat with GBL, seizure activity is clearly recorded on two channels in Figure 5.3(b) as expected for this epilepsy model.

### 2.5 Summary of Novel Contributions

- A wireless SoC for monitoring of epilepsy is demonstrated. The SoC integrates 64 neural recording amplifiers, 64 16-tap mixed-signal FIR filters and a fully integrated PLL-based FSK/OOK ISM-band transmitter. This work demonstrates the highest level of integration among recently published state-of-the-art wireless neural recording SoCs.

- This work demonstrates the first neural recording SoC with a fully integrated, PLL-based transmitter.

- A compact, low-power and programmable mixed-signal FIR filter implementa-
Figure 2.25: (a) Online on-chip in vivo recordings before seizure activity triggered by GBL injection, recorded using on-chip BP FIR filter. (b) Online on-chip in vivo recording showing seizure activity after GBL injection recorded using on-chip BP FIR filter.
tion that utilizes the SAR ADC as the multiplier is demonstrated. This mixed-signal FIR filter approach eliminates the requirement of 512 conventional 8-bit digital multipliers by efficiently implementing the multiplication within the SAR ADC conversion cycle.

- The feedback element in the neural recording amplifier utilizes a tunable low-distortion subthreshold MOS-resistor to reject DC offsets and set a controlled highpass pole. Experimental measurements yield a 10-15dB linearity improvement over the conventional approach.

### 2.6 Conclusion

A 0.13µm CMOS wireless SoC for preoperative epilepsy monitoring is presented. The die integrates 64 fully differential recording amplifiers with 64 SAR ADCs modified to perform in-channel multiplication. The low noise recording amplifiers make use of a novel low-distortion pseudo-resistor to maintain a robust low frequency pole. The multiplying SAR ADCs are utilized to implement a novel mixed-signal FIR filter with minimal area and power overhead. This enables 64 programmable 16-tap mixed-signal FIR filters to perform versatile signal filtering within each channel. A fully integrated PLL-based FSK/OOK transmitter wirelessly sends the digitized neural recorded data at up to 1.5Mb/s with an output power as high as 0dBm. This is the first fully integrated, closed-loop wireless transmitter to be combined with a neural recording SoC. The total power dissipation is 5.1mW from a 1.2V supply. The system was also characterized in vivo. Compared with previous neural recording SoCs, the presented design integrates a large number of recording channels, signal processing and a closed-loop wireless transmitter without a significant compromise on area or power dissipation. This makes the presented SoC a viable option for an envisioned chronically implantable brain activity
monitoring device, and specifically for preoperative epilepsy monitoring.
Chapter 3

A Phase Synchronization and Magnitude Processor VLSI Architecture

3.1 Introduction

Chapter 2 presented a wireless SoC for preoperative epilepsy monitoring with 64 mixed-signal FIR filters. This chapter presents a VLSI architecture that efficiently performs computationally intensive phase-locking value (PLV) and a neural vector analysis for the purpose of early seizure detection. The VLSI architecture utilizes the 16-tap FIR filter structure from Chapter 2 and extends it to include a phase synchrony processor to implement an early seizure detector. This chapter focuses on the details of the digital signal processor (DSP) that will be utilized in the closed-loop neurostimulation SoC presented in Chapter 4.

The seizure detector processor VLSI architecture employs the COrdinate Rotation DIgital Computer (CORDIC) algorithm [116]. The algorithm offers a hardware-
efficient approach to computing trigonometric and vector functions, as it requires only shift-and-add operations for vector rotations. The VLSI architecture is to be integrated with neural recording and stimulation circuits [117] to implement a multi-channel implantable closed-loop microsystem as shown in Figure 3.1. The phase synchronization processor combines three CORDIC processor cores, which operate on vectors to compute the magnitude, phase and the phase-synchronization of two signals. The rest of the chapter is organized as follows. Section 3.2 discusses the phase-synchronization algorithm. Section 3.3 presents the VLSI architecture of the processor. Section 3.4 describes its VLSI implementation. Section 3.5 contains simulation and experimental results of early seizure detection by the phase-synchronization processor, in human EEG recordings.

3.2 Phase-Synchronization Algorithm

For two oscillations, $V_0$ and $V_1$, when their instantaneous phase difference is locked to a constant value, synchronization is present between the two signals. A number of methods exist that quantify the level of frequency-specific synchronization between two neuroelectric signals, including mutual information and Shannon entropy [20]. Estimation
of phase locking has emerged as a popular leading method of quantifying neural synchronization. Its effectiveness comes from the fact that it relies on the phase information of a neural signal, separately from its magnitude. Thus, to quantify the amount of phase locking between two neural signals requires the computation of the phase difference followed by the computation of a phase locking index. First, the Hilbert transform is applied to both signals $V_0$ and $V_1$ to compute their real and imaginary components,

$$ V_0 = Re(V_0) + j Im(V_0), \quad V_1 = Re(V_1) + j Im(V_1) $$

where $V_0$ and $V_1$ are two sinusoidal continuous or discrete-time signals. The Hilbert transform is conventionally performed over the full band of frequencies in the neural spectrum, and thus, a narrow-band bandpass filter should be applied before the Hilbert transform to isolate the signal band of interest [20].

The magnitude in the extracted frequency band can be computed as

$$ MAG(V_k) = \sqrt{(Re(V_k))^2 + (Im(V_k))^2} $$
where $k=0,1$. Next, the instantaneous phases are computed for each channel

$$\phi_k = \arctan \frac{Im(V_k)}{Re(V_k)}$$ (3.3)

and if phase-synchronization exists between the two channels in the same frequency band then the difference in phase is equal to a constant

$$\Delta \phi = \phi_1 - \phi_0 = \text{constant}.$$ (3.4)

Numerous statistical tools exist that quantify the level of phase-synchronization between two signals such as entropy index, mutual information index and mean phase coherence [17]. The hardware-efficient mean phase coherence in [17] was selected, which uses a phase locking value (PLV) between 0 and 1 to evaluate the amount of phase-synchronization. The algorithm defines PLV as

$$PLV = \frac{1}{N} \sqrt{\left( \sum_{i=0}^{N-1} \sin(\Delta \phi_i) \right)^2 + \left( \sum_{i=0}^{N-1} \cos(\Delta \phi_i) \right)^2}$$ (3.5)

where $N$ is the length of the moving-average FIR filters and $\Delta \phi_i$ is the instantaneous phase difference between the $i$-th samples of the two signals.

In summary, the PLV computation requires the Hilbert transform, arctan, addition, sine and cosine, moving-average filtering, square-root, and lastly, the PLV magnitude. The arctan, sine/cosine and magnitude operators will be computed using the CORDIC algorithm while the moving-average filtering will be computed using digital FIR filtering. Both the magnitude value in (2) and the PLV value in (5) are used in this work for early detection of epileptic seizures.
3.3 VLSI Architecture

The architecture of the feedforward path of the system in Figure 3.1 for two channels is presented in Figure 3.2 and contains both analog and digital components. After low-noise amplification of the neural signals by a low-power neural amplifier, narrow-band filter extracts the signal in the frequency band of interest which is then digitized by a low-power medium resolution analog-to-digital converter (ADC).

Next, each digitized signal is fed to a set of two 10-bit finite impulse response (FIR) filters. One FIR filter is configured to perform the Hilbert transform to shift the signal by 90 degrees, while the other FIR filter is an all-pass filter to ensure the digital delays of the two FIR filters are matched. To further save power, this FIR filtering can also be efficiently performed in the mixed-signal VLSI domain by incorporating it within the ADC [118]. This is why the front-end FIR filtering is not considered as a part of the phase-synchronization processor. The rest of the computation is efficiently performed in the digital domain by using the CORDIC algorithm. Sampling more than one pair of analog channel outputs within a single seizure prediction time window yields multivariate signal processing.

3.3.1 CORDIC Algorithm

Next, the phase locking value is computed using the CORDIC algorithm. The CORDIC algorithm has been demonstrated in a large number of applications, such as matrix computations (QRD and eigenvalue estimation), image processing (DCT) and digital communications (FFT, DDS) [116]. The CORDIC algorithm operates on a vector of complex numbers by multiplying it by powers of two removing the requirement of complex multipliers and utilizing only adders, shifters and memory retrieval operations [116]. Using an iterative approach, CORDIC provides a high-accuracy, low-power and low-
area computational algorithm at the cost of reduced speed. Two modes were implemented in CORDIC: the rotational mode which is used for computing sine and cosine, and the vectoring mode which is used to compute the magnitude and the phase. The two modes only differ in the directions of rotation [116].

The CORDIC algorithm involves iterations on three difference equations as follows

\begin{align}
\text{for } n = 1 : 16 \\
x[n + 1] &= x[n] + y[n]2^{-(n-1)} \\
y[n + 1] &= y[n] - \text{sign}(y[n])x[n]2^{-(n-1)} \\
z[n + 1] &= z[n] + \text{sign}(y[n])\arctan(2^{-(n-1)})
\end{align}

To compute the magnitude and the phase using CORDIC, the initial values must first be set. For the CORDIC equations, the initial \(x[1]\) and \(y[1]\) would represent the real and imaginary components of the signal, respectively, with \(y[1]\) set to 0. Over the next 16 clock cycles the procedure that computes magnitude and phase is repeated while \(y\) converges to 0. The final value \(x[16]\) represents a scaled magnitude and the final value \(z[16]\) represents the phase. A look-up table that stores 16 arctan values was used in the phase computation. Computing sine and cosine is similar except we initialize \(x[1]\) to the CORDIC aggregate constant \(K\), set \(y[1]\) to 0 and set \(z[1]\) to the angle we want to compute. Also \(z\) converges to 0 instead of \(y\) and the final output, \(x[16]\), represents the cosine of the angle, while \(z[16]\) represents the sine of the angle.

### 3.3.2 CORDIC-based Processor VLSI Architecture

The VLSI architecture of the 10-bit phase-synchronization and magnitude processor is shown in Figure 3.3. It uses three pipelined CORDIC cores and two moving-average
FIR filters. The pipelined architecture allows the supply voltage to be lowered to minimize power dissipation by using a lower frequency clock while maintaining a constant throughput. The first core receives the two digitized vectored signals, preprocesses them by extracting the quadrant of the angle and then simultaneously computes both the angle between 0 and 90 degrees and the magnitude using a 16-bit CORDIC core configured in the vectoring mode. The angles are re-adjusted using the stored quadrant information to output an angle between 0 and 360 degrees. The difference between the two computed angles is transferred to the next stage.

The sine and cosine of the angle difference are computed using a 16-bit CORDIC core configured in the rotational mode. The computed sine and cosine as well as the negative flags are transferred to the two 32-tap moving-average FIR filters. Higher sensitivity for the PLV algorithm can be achieved by increasing the length of the FIR filters at a cost in area and complexity. Lastly, the PLV is computed by extracting the magnitude of the FIR averaged sine and cosine outputs using a 16-bit CORDIC core configured in the vectoring mode. An output multiplexer can be configured to output the instantaneous magnitude and phase of each channel, as well as the phase difference and the PLV between two channels. Each CORDIC core requires 18 clock cycles which include one clock cycle for pre-processing the angles, 16-clock cycles to perform the CORDIC algorithm and one clock cycle to output the data and post-process the angles for a total latency of 54 clock cycles to compute the PLV algorithm. The simulation results in Figure 3.4 show how the PLV sensitivity improves with increasing the length of the moving-average FIR filters.

A resolution of 10-bits was selected for the processor. Figure 3.5(a) shows the difference between results using offline human seizure data obtained from an ideal resolution processor and a finite resolution processor quantified in standard deviations. In terms of seizure detection performance there was little benefit of increasing the processor reso-
solution above 9 bits. As the processor resolution drops below 7 bits, the accuracy of the PLV algorithm degrades. This is demonstrated in Figure 3.5(c) which compares seizure detection performance between a 7-bit and 10-bit processor. As the seizure in Figure 3.5(b) approaches, the lower resolution 7-bit processor is not able to detect the small variations of synchrony which occur 30-40 seconds before the seizure. For resolutions above 9 bits the performance is nearly identical.

3.4 VLSI Implementation

The processor was designed and synthesized using a standard 8-metal 0.13\(\mu\)m CMOS technology. The synthesizable RTL code of the full processor is provided in the Appendix. It contains a total of 41,366 gates and occupies an area of 0.178\(mm^2\). The first magnitude/phase CORDIC core occupies 20.6 percent of the area, the second sine/cosine CORDIC core uses 12.8 percent, the FIR moving-average filters occupy 57 percent, the third magnitude CORDIC core utilizes 9 percent and pre-processing and the output MUX occupy 1 percent of the total core area. Accuracy and sensitivity of the PLV com-
Figure 3.4: Simulated PLV for different values of $N$ (length of the FIR filters) when the frequency of one input, $V_{IN1}$, is held constant at 110Hz and the frequency of the other input, $V_{IN0}$, is swept from 70Hz to 150Hz.
Figure 3.5: (a) Simulated error of the PLV for different processor resolutions when compared with an ideal processor resolution. (b) Offline human seizure input for a University of Toronto seizure patient. (c) Simulated PLV between two inputs in the 25Hz to 35Hz frequency band computed for a processor resolution of 7-bits and 10-bits.
putation can be traded for area by reducing the length of the moving-average FIR filters. A 4 times increase in the length of FIR filters yielded an overall layout area increase of 1.8 times and an overall power dissipation increase of 1.7 times.

The univariate magnitude and phase-estimation operations and the bivariate phase difference and PLV-estimation operations are all computed simultaneously for every sample and are time-multiplexed through a 10-bit output. The synthesized design can operate at frequencies above 100MHz, which is beyond the requirements of the intended application. This margin allows the ability to further reduce power dissipation by lowering the supply voltage.

### 3.5 Results

#### 3.5.1 Experimental Circuit-Level Results

The phase-synchronization processor was prototyped in a standard 0.13µm CMOS technology and characterized experimentally. The processor is shown in Figure 3.6. Two analog signals were digitized, sent through FIR filters to obtain the Hilbert transform and its delayed version, and fed to the processor as shown in Figure 3.7. The experimentally measured magnitude extraction results are shown in Figure 3.8, when a
A sinusoid is applied as an input to the processor. The maximum error is below 3.5 percent with respect to the full scale when the input is between 0 mV and 600 mV. For a neural amplifier gain of 2,000 V/V, this corresponds to a neural signal between 0 µV and 300 µV.

Next, two sinusoid inputs were set to 110 Hz, with one sinusoid having its phase locked while adjusting the phase of the other sinusoid. The measured phase difference between this pair of inputs is shown in Figure 3.9. The maximum error is approximately 1.5 percent.

The measured PLV between a pair of inputs is shown in Figure 3.10. The average PLV between the two inputs is computed with one input held constant at 110 Hz, while the other input frequency is swept from 60 Hz to 160 Hz. As expected, the computed PLV is near unity when the two signals have the same frequency. When the second signal has its frequency set to 60 Hz or 160 Hz, the PLV drops to 0.45 and 0.4 respectively.
Figure 3.8: (a) Experimentally measured magnitude of a sinusoidal input computed by the phase-synchronization processor. (b) The corresponding relative percent error.
The phase locking sensitivity can be further improved by increasing the length of the moving-average FIR filters. The experimentally measured PLV result was compared with the SIMULINK model with the bandpass filter removed. The PLV result computed by the SoC and by the simulink model are in close agreement when the frequencies of the two inputs are within 20 percent or exhibiting a PLV above 0.7. As the two frequencies further deviate, the error between the simulated and experimental PLV is less than 0.1.

The experimentally measured power dissipation of the processor for a supply voltage of 0.85 and 1.15V operating at various clock frequencies is shown in Figure 3.11. For a 0.85V supply at 2.5MHz and 10MHz the processor dissipates 102\(\mu\)W and 412\(\mu\)W, respectively. For a 1.15V supply at 2.5MHz and 10MHz the processor dissipates 231\(\mu\)W and 897\(\mu\)W, respectively. For only one pair of channels, operating at 1.7kS/s the pro-

Figure 3.9: Experimentally measured phase difference between two input sinusoids. (b) The corresponding relative percent error.
cessor dissipates 3.6µW from a 0.85V supply.

On-chip signal processing offers better energy efficiency when compared with off-chip signal processing connected through a wireless link. The seizure detector processing dissipates only 3.6µW/channel and has an energy efficiency of 210pJ/bit. For processing off-chip this requires an RF wireless transmitter such described in Chapter 2 has a power dissipation of 3.7mW which yields an energy efficiency of 2.5nJ/bit. It is important to note that a certain output power (100µW-1mW) is required for a wireless transmitter to transmit the data and overcome losses due to tissue absorption and antenna loss. A simple ASK wireless receiver to program and trigger the on-chip neural stimulators such as described in [119], has a power dissipation of 350µW. Overall the on-chip signal processing requires 100X less power dissipation compared with wireless off-chip processing.

### 3.5.2 Simulated Human EEG Results

A Verilog-AMS model of the phase-synchronization processor is too computationally complex to be used in early seizure detection simulations. Instead, a Simulink model which had its resolution and accuracy set to match the performance of the RTL-level implementation of the synthesized processor was utilized.

The efficacy of the phase-synchronization processor in early seizure detection was verified and validated on an EEG database from the seizure prediction project at the University of Freiburg [120]. It consists of 222 hours of intracranial EEG recording of three patients with a total of 30 seizures analyzed and labeled by certified epileptologists as summarized in Table 3.1. The data was acquired by a Neurofile NT digital video EEG system with 128 channels, 512 Hz sampling rate, and a 16-bit analog-to-digital converter via implanted depth, strip and grid electrodes.

For each patient, three intracranial electrodes located in the proximity of an epileptic
Figure 3.10: Experimentally computed and simulated PLV when the frequency of one input, $V_{IN1}$, is held constant at 110Hz and the frequency of the other input, $V_{IN0}$, is swept from 60Hz to 160Hz.

Figure 3.11: Experimentally measured power dissipation at different operating frequencies.
Table 3.1: EEG Database Used in Verification of the Phase-Synchronization Processor. Seizure Types: Simple Partial (SP), Complex Partial (CP), Generalized Tonic-Clonic (GTC). Seizure Location: Hippocampal (H), Neocortical (NC). Electrode Types: depth (d), strip (s), grid (g).

<table>
<thead>
<tr>
<th>Patient</th>
<th>Sex</th>
<th>Age</th>
<th>Type</th>
<th>Location</th>
<th>Electrodes</th>
<th>Outcome</th>
<th>No Seizures</th>
<th>EEG (hrs)</th>
<th>No. Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>30</td>
<td>SP, CP, GTC</td>
<td>H</td>
<td>D, S</td>
<td>I</td>
<td>10</td>
<td>98</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>17</td>
<td>SP, CP, GTC</td>
<td>NC</td>
<td>G, S</td>
<td>I</td>
<td>10</td>
<td>30</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>10</td>
<td>SP, CP, GTC</td>
<td>H</td>
<td>D, S</td>
<td>I</td>
<td>10</td>
<td>102</td>
<td>22</td>
</tr>
</tbody>
</table>

focus were used. For each electrode type, a sub-set of contact pairs was selected for the PLV computation as follows. First, only the contact pairs of the same electrode type were used as inputs to the PLV processor to suppress common-mode noise. Next, a reference electrode was chosen for each electrode type as the contact furthest away from the epileptic focus. Then, for each contact pair, the signal spectrum was separated into five frequency bands via high-Q bandpass filters: delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), a sub-band of beta (15-25 Hz) and gamma (30-48 Hz). Finally, a sub-set of contact pairs across all frequency bands was selected by pairing each contact with the reference and observing the channel magnitudes and the PLV within three hours of a clinical seizure onset.

The sensitivity, defined as the true positive rate ($TPR$), of the phase-synchronization algorithm was evaluated as a function of seizure occurrence period ($SOP$), seizure prediction horizon ($SPH$), false positive rate ($FPR$) and maximum false positive rate ($FPR_{max}$) [21], [22], [23]. The $SOP$ is defined as the period of time in which only one seizure is to be expected. It was varied between 5 and 40 minutes to account for the uncertainty in the seizure frequency. The $SPH$ was set to between 16 seconds
and 5 minutes to allow enough time for the therapeutic intervention, such as electrical stimulation or a warning signal to the patient.

Three early seizure detection methods were used in the analysis of the EEG database: magnitude, PLV and combined magnitude with PLV detectors. The magnitude detector is active when the amplitudes of both input channels within a frequency band of interest cross their corresponding thresholds. The PLV detector is triggered when the phase locking value, integrated over an adjustable period of time, drops below a certain threshold. The combined magnitude and PLV detector output is formed by AND-ing the outputs of the individual detectors. Each threshold detector, once triggered, remains active for the duration equal to the sum of $SOP$ and $SPH$ time periods. All early seizure detection methods were compared against the random detector.

The experimental results showed highest PLV activity in the 15-25Hz frequency band. Figure 3.12 presents an example of an early seizure detection result computed by the phase-synchronization processor in the 15-25Hz frequency band for patient 2. The clinical seizure onset is characterized by an increase in the magnitude and a marked fluctuation (a drop in this case) in the PLV before the seizure. The phase locking value reaches 1 during the seizure reflecting the synchronized firing of neurons. Similar early seizure detector outputs generated by the processor are observed in the 15-25Hz frequency band of patient 3 as shown in Figure 3.13. The magnitude and PLV thresholds are adjusted for each patient to achieve the highest sensitivity in early detection of seizures.

Figures 3.14(a) and 3.14(b) show the variation of the sensitivity with $FPR$ and $SOP$, respectively, for patient 2 in the University of Freiburg data set. Each plot compares the sensitivity of the three detectors in comparison to a random detector. The sensitivity plot in Figure 3.14(a) shows that the best detector, the PLV detector, achieves 66 percent sensitivity with FPR of 0.65 FP/hr, $SOP$ of 30 minutes, and $SPH$ of less
Figure 3.12: Early seizure detection results example for patient 2. (a), (b) Two input intracranial EEG signals recorded on electrodes 13 and 23, respectively. (c), (d) Magnitude of the signals shown in (a), (b), respectively, after they are bandpass filtered in the 15Hz to 25Hz frequency range, computed by the prototyped phase-synchronization processor. (e) Integrated PLV between the two inputs in the 15Hz to 25Hz frequency band computed by the prototyped phase-synchronization processor.
Figure 3.13: Early seizure detection results example for patient 3. (a), (b). Two input intracranial EEG signals recorded on EEG electrodes 1 and 2, respectively. (c), (d) Magnitude of the signals shown in (a), (b), respectively, after they are bandpass filtered in the 15Hz to 25Hz frequency range, computed by the prototyped phase-synchronization processor. (e) Integrated PLV between the two inputs in the 15Hz to 25Hz frequency band computed by the prototyped phase-synchronization processor.

The sensitivity plot in Figure 3.14(b) shows that when the $FPR_{max}$ is increased to 1.2/hr, the true positive rate reaches 100 percent for a seizure occurrence period greater than 10 minutes. By increasing the $FPR_{max}$ to 1.2 FP/hr-2.0 FP/hr depending on a patient, $TPR$ approaches 100 percent for all patients. The $FPR_{max}$ could not always be made constant by varying the thresholds; therefore, the average $FPR_{max}$ rates are shown for each early detection method in Figure 3.14(b).
Table 3.2: Performance comparison of bivariate seizure prediction algorithms implemented in software with the current work.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Detection/Prediction Method</th>
<th>TPR vs. FPR (%)</th>
<th>FPR=0/hr</th>
<th>FPR=1/hr</th>
<th>SOP=10 min</th>
<th>SOP=40 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>[21]*</td>
<td>Phase Locking Value</td>
<td>20</td>
<td>100</td>
<td>20</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>[22]*</td>
<td>Similarity Index</td>
<td>30</td>
<td>40</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>[121]*</td>
<td>Correlation Dimension</td>
<td>15</td>
<td>40</td>
<td>15</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>This Work**</td>
<td>Phase Locking Value</td>
<td>33</td>
<td>66</td>
<td>33</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

*- Software-based, **-VLSI-based, TPR-true positive rate, FPR-false positive rate, SOP-seizure occurrence period.

The sensitivity results in early seizure detector are comparable to previously reported software-based bivariate prediction algorithms [21], [22], [23] as summarized in Table 3.2 (with the exception of [21] which results in 100 percent TPR at FPR=1/hr versus 100 percent TPR at FPR=1.2/hr performance yielded by the presented phasesynchronization processor architecture). All of the listed algorithms have been tested on the same benchmark data set from the University of Freiburg, but the presented algorithm is the only one that has been implemented as a low-power implantable integrated circuit.
Figure 3.14: Early seizure detection results for patient 2, showing (a) true positive rate vs. false positive rate plot with the seizure occurrence period held at 30min, and (b) true positive rate vs. seizure occurrence period plot. For the above detectors, MAG refers to magnitude, PLV refers to phase locking value, and RND refers to random.
3.5.3 Experimental Human EEG Results

Two-electrode intracranial EEG data from patient 1 from University of Freiburg database was loaded onto a dual-channel Tektronix AFG3252 arbitrary waveform generator and fed into the phase-synchronization processor chip. A bandpass filter filtered the inputs to 15-25Hz frequency band. Two 8-bit ADCs and four 16-tap FIR filters (two all-pass and two Hilbert filters) digitized and applied a 90 degree phase shift, respectively, to the two signals. The processor simultaneously computed both magnitudes, the phase difference and the PLV at 1.7kS/s dissipating 3.6µW from a 0.85V supply.

Two-electrode intracranial EEG seizure recordings from patient 1 (electrode-40 and electrode-44) are shown in Figure 3.15(a) and (b). Figure 3.15(c) shows the experimentally measured magnitude from electrode-40, while Figure 3.15(d) shows the experimentally measured magnitude from electrode-44. For both inputs, the experimentally measured magnitude was observed to increase during the seizure. Lastly, Figure 3.15(e) shows the experimentally measured PLV between the two inputs. The PLV was observed to drop 15 seconds before the onset of the seizure displaying an early detection, then increasing during the seizure. Seizures from other patients were also input into the processor. The magnitude was observed to increase before and during the seizure while a pronounced fluctuation of PLV is observed before and during the seizure.

3.6 Summary of Novel Contributions

- A bivariate advanced seizure detector CORDIC-based processor is demonstrated. The phase synchrony processor is benchmarked with offline human seizure data and yields comparable performance to offline software-based seizure detectors.
Figure 3.15: Experimentally measured early seizure detection results example for patient 1. (a), (b). Two input intracranial EEG signals recorded on EEG electrodes 41 and 44, respectively. (c) Experimentally measured magnitude in the 15-25Hz range of the signal shown in (a). (d) Experimentally measured magnitude in the 15-25Hz range of the signal shown in (b). Experimentally measured PLV between signals (a) and (b).
3.7 Conclusions

A compact low-power signal processing VLSI architecture has been presented that computes the phase locking value on two neuroelectrical signals and the instantaneous magnitude on individual neural inputs. The signal processor is used in conjunction with a neural recording front-end and operates in real time on frequency bands in the neural spectrum. The processor occupies 0.178 mm$^2$ area and dissipates 3.6 $\mu W$ operating on a pair of inputs sampled at 1.7kS/s from a 0.85V supply. Results from pre-recorded human intracranial EEG data demonstrate the effectiveness of the processor in early seizure detection.
Chapter 4

64-Channel Wireless Neural Vector Analyzer and Phase Synchrony-Triggered Stimulator SoC

4.1 Introduction

A conceptual view of a fully integrated wireless closed-loop neurostimulation microsystem for intractable epilepsy treatment is shown in Figure 4.1. In this chapter we present a 64-channel neural recording and 64-channel neural stimulation SoC (Chip 1 in Figure 4.1) that operates in a closed loop using a high-efficacy signal detection algorithm. The work from Chapter 2 including the array of 64 FIR filters, and the neural recording amplifiers were combined with the signal processing functionality described in Chapter 3 to yield a neural vector analyzer with an on-chip phase synchrony processor. Further integration of on-chip neural stimulators and a UWB transmitter yields the full wireless closed-loop phase-synchrony triggered SoC.

This chapter demonstrates the first on-chip neural vector analyzer with the abil-
Figure 4.1: System diagram of the envisioned implantable closed-loop wireless SoC.

ity to process not only the magnitude, but also the phase of neural signals. It is also the first SoC to demonstrate real-time on-chip computation of the phase synchrony between neural signals and to demonstrate on-chip closed-loop stimulation based on bivariate signal processing. The SoC integrates 64 neural recording channels with in-channel SC bandpass filters and ADCs, 64 biphasic current stimulation channels, 64 programmable mixed-signal 16-tap FIR filters, a tri-core coordinate rotation digital computer (CORDIC) processor and a 3.1-10.4GHz ultra wideband (UWB) wireless transmitter onto a compact 4x3mm\(^2\) 0.13\(\mu\)m CMOS die. The SoC extracts narrow frequency bands in the neural spectrum (below 5kHz) and computes the magnitude and phase on 64 individual channels. It also computes the phase difference and phase synchrony between channels in up to 32 pairs. To minimize the area and power dissipation
and to enable implantability, the successive approximation register (SAR) ADC is re-used as an analog-digital multiplier for FIR filtering and as a DAC and digital controller for the neural stimulator.

The rest of the chapter is organized as follows. Section 4.2 discusses the VLSI architecture of the 64-channel neural recording and stimulation SoC. Section 4.3 presents the circuit implementation of the key functional blocks in the SoC. Section 4.4 presents electrical experimental results from the individual blocks and the full system. Section 5.4 presents in vivo online animal and offline human results validating the SoC in epilepsy treatment.
CHAPTER 4. WIRELESS NEURAL VECTOR ANALYZER AND PHASE SYNCHRONY-TRIGGERED STIMULATOR SOC

Figure 4.3: Detailed system block diagram of the SoC.
4.2 System VLSI Architecture

4.2.1 Top-level VLSI Architecture

The functional diagram of the SoC is depicted in Figure 4.2. The loop consists of the neural vector analyzer in the feed-forward path and the phase synchrony-triggered neural stimulator in the feedback. Pairs of neural signals, \( V_0 \) and \( V_1 \), are amplified and then filtered within a specific pass-band to monitor the abnormal phase synchrony of interest in that band. The band of interest can be adjusted by programming the center frequency of the bandpass filter.

To monitor the phase synchrony, the in-phase (I) and quadrature-phase (Q) components are then computed by the allpass and Hilbert filters, respectively. The Hilbert filter introduces a 90 degree phase shift and a time delay, while the allpass filter introduces an equal time delay. This results in a 90 degree phase difference between the two components. These components comprise each neural signal, \( V_0 \) and \( V_1 \), as follows

\[
V_0 = Re(V_0) + jIm(V_0), \quad V_1 = Re(V_1) + jIm(V_1),
\]

where \( Re(\cdot) \) is the I component and \( Im(\cdot) \) is the Q component.

The allpass and Hilbert filtering are conventionally performed over the full band of frequencies in the signal spectrum. In the proposed implementation, a narrow-band bandpass filter is employed before the Hilbert transform to isolate the narrow frequency band of interest where abnormal oscillations are monitored. This enables precise computation of the magnitude and phase at the narrow band center frequency. The algorithm details were given in detail in Chapter 3.

These I and Q components are then used to compute the magnitude (MAG) and phase (\( \phi \)) of the signals, and their pairwise phase differences (\( \Delta \phi \)). The magnitude and
instantaneous phase in the extracted frequency band are then computed as

\[ |V_k| = \sqrt{\text{Re}(V_k)^2 + \text{Im}(V_k)^2}, \phi_k = \arctan \frac{\text{Im}(V_k)}{\text{Re}(V_k)} \] (4.2)

where \( k = 0, 1 \). The vectored outputs are serially wirelessly transmitted. This implements the function of a wireless neural vector analyzer.

Phase locking value (PLV) is a common metric of phase synchrony and is computed in the feedback loop of the SoC. In epilepsy, seizure precursors are abnormally large fluctuations in PLV which can be detected by thresholding. The biphasic current-mode stimulator is triggered when the PLV indicates abnormal phase synchrony levels. In epilepsy this is done to abort an upcoming seizure. The hardware-efficient mean phase coherence algorithm [17] was utilized here to compute PLV ranging between 0 and 1 as

\[ PLV = \frac{1}{N} \sqrt{\left( \sum_{i=0}^{N-1} \sin(\Delta \phi_i) \right)^2 + \left( \sum_{i=0}^{N-1} \cos(\Delta \phi_i) \right)^2}, \] (4.3)

where \( N \) is the length of the sample observation window and \( \Delta \phi_i \) is the instantaneous phase difference between the \( i \)-th samples of the two signals.

### 4.2.2 Detailed VLSI Architecture

A detailed block diagram of the SoC is shown in Figure 4.3. There are 64 ADCs organized in a scalable manner, one per neural input, for raw neural data monitoring (not shown). In the neural vector analysis and phase synchrony computation mode, the ADCs are re-configured as multiplying ADCs (MADCs). These MADCs perform the multiplications which are the most computationally intensive operations in allpass and Hilbert FIR filters. A bank of eight MADCs combined with a 16-tap folded add-and-delay line yields a 16-tap transposed symmetric mixed-signal FIR filter. MADCs are
time-multiplexed among eight neural inputs and eight add-and-delay lines to comprise eight FIR filters [122]. In this design, two sets of eight FIR filters, eight allpass and eight Hilbert filters are implemented to compute I and Q components for eight inputs, respectively. The phase, magnitude and phase synchronization processor utilizes three digital CORDIC cores. By performing all FIR multiplications within the ADC, and utilizing vector processing in the CORDIC-based processor, the computationally intensive digital multiplication operation is avoided entirely. The computed PLV value is compared with the programmed neural stimulation criteria. When triggered, the 64-channel neural stimulator generates individually programmable biphasic currents. The duty cycle and amplitude can be programmed and stored in the on-chip memory. An UWB wireless transmitter sends raw ADC data, neural vector parameters and PLV data off-chip.

The SoC also contains 1kb of on-chip registers that store the FIR coefficients, stimulation current amplitude and stimulation current duty cycle. The 0.13\(\mu\)m CMOS SoC utilizes 1 million transistors and operates from a 1.2V supply for the neural recording, signal processing and wireless transmission and a 3.3V supply for the neural stimulation. An on-chip controller generates all clocks used by various blocks within the SoC.

### 4.3 VLSI Circuit Implementation

#### 4.3.1 Neural Recording

**Low-noise Amplifier (LNA)**

The first stage of the amplifier utilizes two diode-connected MOSFETs which implement two MOS-bipolar pseudo-resistive devices [41]. In the second stage, to ensure
Figure 4.4: Neural recording channel including LNA, SC BPF and SC LPF.
a more constant $|V_{GS}|$ as the output swings and to allow for tunability, two source followers act as floating DC voltage sources to track the dynamic output as shown in Figure 4.4. The NMOS and PMOS feedback devices are implemented using thick oxide and the source followers are implemented using low-threshold thin-oxide devices. Transistor sizes of the feedback element are listed in Table 4.1.

The first stage utilizes a fully differential folded-cascode OTA (OTA1 in Figure 4.4) with a 60dB open-loop gain. To minimize flicker noise of devices M3/M4 and M5/M6 the first chopping circuit is placed at the folded node. Placing chopping in front of the input pair, as is done conventionally [46], would have resulted in noise multiplication due to a switched capacitor resistor created by the parasitic capacitance of the input pair [46]. Conventionally, this is overcome by utilizing very large capacitors at the input of the amplifier [46] at the expense of large silicon area. The trade-offs of different chopping topologies are described in Section 1.3.1 of Chapter 1.

The low impedance at the folded-cascode node allows for high-frequency chopping at that node. We choose to chop at that node. The chopping harmonics are filtered out by the bandpass filter of the second stage of the low-noise amplifier. To minimize flicker noise of the input pair M1/M2, large PMOS devices are selected. To minimize thermal noise, transistors M1 and M2 are biased in the sub-threshold regime as needed to maximize their $g_m$. Transistors M3/M4 and M5/M6 are biased in saturation to minimize their $g_m$. Common-mode feedback (CMFB) is implemented by modulating the tail current of the first stage of the folded-cascode OTA using a continuous-time CMFB amplifier as described in [123]. The OTA requires a total of $2.6\mu\text{A}$ of current which includes $1.6\mu\text{A}$ into the input pair, $0.8\mu\text{A}$ into the cascode transistors and $0.2\mu\text{A}$ for the CMFB amplifier. The second stage utilizes a two-stage OTA topology. This OTA has PMOS input devices, has an open-loop gain of 60dB and a phase margin of 70 degrees. It draws 350nA of current from a 1.2V supply. Transistor sizes for both OTAs are listed
A noise simulation (at 27 degrees Celsius) summarizing the input-referred noise of a AC-coupled capacitive feedback amplifier with the gain set to 50 and sweeping the capacitor sizes yields a trade-off of area and noise as shown in Figure 4.5. This depicts the input-referred noise with no chopping, chopping in front of the OTA and chopping inside the OTA at the folded node. For a large number of channels (greater than 32), placing the chopping in front of the OTA requires large passives to reduce the effect of noise multiplication. Setting the input capacitor between 10pF and 20pF allows for high channel integration and for this case placing the chopping inside the OTA provides the best area-noise tradeoff.

When chopping is disabled, 30 percent of the total input-referred integrated noise in the frequency band from 10Hz to 5kHz is contributed by the flicker noise of M3 and M4 as shown in Figure 4.6(a). When chopping is enabled, the majority of the total input-referred integrated noise is due to thermal noise as depicted in Figure 4.6(b). Flicker noise is contributed mainly by the input differential pair. This design allocates 25 percent of the total noise contribution to the flicker noise of the input differential pair.
which can be further reduced by increasing the size of the input devices at the expense of silicon area.

**Switched-Capacitor Bandpass Filter**

The second-order bi-quad switched-capacitor (SC) bandpass filter (BPF) selects the neural band of interest. The circuit architecture as shown in Figure 4.4 was selected as it minimizes the capacitance ratios resulting in a smaller area. The quality factor (Q) and center frequency, respectively, are

\[
Q = \frac{C_2}{C_1}, \quad f_{BP} = \frac{1}{2\pi} \frac{C_3}{C_2} f_{SC}
\]  

with \(C_1=770\text{fF}, C_2=2.5\text{pF}\) and \(C_3=200\text{fF}\). This yields a Q of 3.2 and a center frequency 78.5 times lower than the switched-capacitor clock frequency, \(f_{SC}\). A 2kHz clock yields a center frequency of approximately 25Hz.
Figure 4.6: Summary of noise contributions with the chopper enabled and disabled.

The two OTAs are implemented using the two-stage topology, each requiring 350nA of current with capacitors implemented as dense dual-MIM capacitors. Adjusting the SC clock frequency, \( f_{SC} \), enables linear tuning of the center frequency of the bandpass filter. This allows for filtering in different neural frequency bands.

**Switched-Capacitor Lowpass Filters**

The SC BPF discussed above introduces harmonics below the Nyquist rate of the ADC. To remove these harmonics a switched-capacitor lowpass filter (LPF) is utilized as shown in Figure 4.4 in order to reduce the area of the on-chip passives required for the low-frequency filtering. The LPF is a SC LPF which utilizes a clock frequency that is 16 times higher than that of the switched-capacitor BPF filter, and attenuates the BPF harmonics. The lowpass pole frequency is

\[
f_{LP} = \frac{f_{SC}}{2\pi \times 10}
\]  

and yields

\[
\frac{f_{BP}}{f_{LP}} = 0.8.
\]
This sets the lowpass SC filter cutoff frequency to 1.25 times the center frequency of the SC bandpass filter.

### 4.3.2 FIR Filtering Circuits

**Multiplying SAR ADC**

A successive approximation register (SAR) ADC is often used to digitize the amplified neural signals since it provides high energy efficiency for medium resolutions and sample rates. The implemented ADC is an 8-bit charge redistribution SAR multiplying ADC (MADC) as shown in Figure 4.7 and detailed in [122]. The MDAC utilizes a split-capacitor array to minimize the overall area of the binary capacitor array. It multiplies two digital values at the cost of a small overhead of three two-input logic gates per bit as shown in Figure 4.8. The ADC dissipates 1.8\( \mu \)W at 56kS/s from a 1.2V supply. Each channel contains a 22-bit register to store the 8-bit multiplication coefficient, two 1-bit sign coefficients for the FIR filter, and 12-bits for neural stimulation mode.
Figure 4.8: Integrated circuit implementation of eight channels of neural signal filtering for different modes of operation. (a) Raw ADC mode, (b) General purpose filtering FIR mode and (c) I/Q extraction mode.
Neural Signal Filtering Modes

The SoC is divided into eight modules each consisting of eight channels. A module in different configuration modes is shown in Figures 4.8(a)-(c). In the raw-ADC mode, the analog output is digitized directly by the ADC as shown in Figure 4.8(a). In the FIR filtering mode, an OTA buffer drives the eight parallel MADCs of the mixed-signal FIR filter. Each MADC is a SAR ADC with a multiplying DAC (MDAC). In this mode, 64 recording channels use the 64 FIR filters as depicted for eight channels in Figure 4.8(b). In I/Q vector mode, the OTA buffer drives two sets of eight parallel MADCs. One set of MADCs implements an FIR filter programmed as an allpass filter (I-extraction). The other set implements an FIR filter programmed for the Hilbert transform (Q-extraction). In this mode, 32 recording channels utilize 64 FIR filters (32 allpass and 32 Hilbert FIR filters) as shown for eight channels in Figure 4.8(c). Both the I and Q signals are sent to the on-chip CORDIC processor to compute the magnitude, phase and phase synchronization.

Eight or 16 recording channels share the tuneable RC LPF as shown in Figures 4.8(b) and (c), respectively. The RC LPF removes the higher frequency harmonics of the in-channel SC LPF described in Section 4.3.1 and switching harmonics of the multiplexer. The two-stage OTA buffer draws a current of 25µA. The value of R is tuneable between 365kΩ to 1.5MΩ and the value of C is 25pF yielding a lowpass pole between 4.2kHz and 17.4kHz. For example, for a SC BPF center frequency of 65Hz, this requires a SC BPF clock frequency of 5kHz and a SC LPF clock frequency of 80kHz (16×f_{SC}). The 80kHz harmonics are then removed by this RC lowpass filter.

4.3.3 Phase, Magnitude, and Phase Synchronization Processor

The block diagram of the digital signal processor that computes phase, magnitude and phase synchronization is shown on the right hand side of Figure 4.9 and detailed in
Chapter 3. The processor receives vectored inputs, I and Q, for each of the two channels. The 10-bit processor is fully synthesized utilizing three CORDIC cores to simultaneously compute the magnitude, phase, phase difference and the phase-locking value per sample. An iterative CORDIC-based architecture was selected as it does not require any digital multiplication, but only adders, bit shifters and memory retrieval operations [116]. This minimizes the area and power dissipation while trading-off speed and latency. It utilizes 41k gates, requires 18 clock cycles per operation, occupies an area of 0.178 mm² and has a total latency of 54 clock cycles. It dissipates 200 µW from a 1.2V supply at 2MHz at 1.7kS/channel and scales linearly with the sampling rate. A digital threshold can be programmed to trigger a stimulation pulse if the magnitude, phase, phase-difference or PLV goes above or below a chosen threshold.

4.3.4 UWB Transmitter (Designed by Hamed Mazhab Jafari)

The block diagram of the all-digital pulsed UWB transmitter is shown in Figure 4.10. The input data are modulated using on-off keying (OOK) modulation at up to 10Mb/s.
UWB pulses are generated on the rising edge of the input data. The proposed digital UWB transmitter achieves both power efficiency and spectral compliance in a minimal chip area by combining a delay line architecture with a capacitively coupled output combiner [124–126]. The input data are passed through a delay line and a delayed version of the input data are passed through three pulse generators. The pulse generators form a first-order Gaussian pulse at the rising edge of the input data. Each pulse generator forms pulses which are in-phase but have opposite-sign DC components. By capacitively combining the three paths outputs, the opposite-sign DC signals are canceled and the zero-DC double differentiated Gaussian pulse propagates to the single-ended antenna. The width of the output pulse depends on the delay in the delay lines. The delay cells in all the paths are implemented as current-starved inverters to allow for tuning of the UWB pulse width for two different bands (0-1GHz or 3.1-10.4GHz).

### 4.3.5 Neural Stimulator

**DAC-Sharing Architecture**

Any of the 64-channels can be individually configured as a neural recorder or a neural stimulator. Incorporating neural recording and neural stimulation capabilities into each channel requires both an ADC for the recording phase and a DAC and digital controller for the stimulation phase.
The SAR ADC described in Section 4.3.2 requires a capacitor-array DAC and digital SAR logic which take up a significant amount of area. The DAC as well as the digital logic SAR controller can be reused for neural stimulation since the neural recording and stimulation circuits do not operate at the same time on one input. Here the SAR MADC is reconfigured such that the MDAC and the digital logic of the SAR MADC are reused to set the stimulation current amplitude and its duty cycle, respectively, as shown in Figure 4.11. By sharing the MDAC and digital logic between the recording and stimulation phases, a more compact implementation is realized.

In the neural stimulation mode, the MDAC of the SAR MADC operates as a voltage DAC and the SAR logic of the MADC is reconfigured to implement duty cycle modulation. This novel neurostimulator architecture is area and power-efficient as hardware resources are shared between modes. Both are combined with a V-I converter and an output current driver to implement the current-mode neural stimulator as shown in Figure 4.11. The neural stimulator is a biphasic current stimulator with an 8-bit in-channel DAC, a 4-bit duty cycle controller and the corresponding 12-bit shift register memory. Biphasic current-mode stimulation is preferred over voltage-mode stimulation as it offers direct control over the charge delivered to the tissue and ensures charge balancing.
when sinking and sourcing current to and from the neural tissue. Currents in the range of 10\(\mu\)A - 1mA can be generated and are sufficient to invoke a neural response [80].

The detailed schematic is shown in Figure 4.12(a) with transistor sizes listed in Table 4.1. The main components of the stimulator are an 8-bit MDAC capacitor array, low-power SAR digital logic, the voltage-to-current converter and the high-impedance biphasic current driver.
DAC and Duty Cycle Controller

The DAC and duty cycle controller are reconfigured from the SAR ADC architecture. Each bit is toggled beginning with the most significant bit. There are eight pulses in a shift register which toggle each bit of the DAC. To adjust the duty cycle, these pulses are combined together to form a wider or narrower current pulse. Two conversion cycles are required for a biphasic current pulse generation. During one conversion cycle up to eight different pulses can be combined to form the positive phase pulse and similarly up to eight different pulses can be combined to form the negative phase pulse. This requires 4-bit storage for adjusting the duty cycle. The approach to combine pulses to modulate the duty cycle, is illustrated in an example in Figures 4.12(b) and (c), where five bits are combined to generate the positive biphasic current pulse. To minimize both area and power, all digital logic utilizes 1.2V thin-oxide devices with minimum length (0.12µm). The signal is boosted to 3.3V to drive the rest of the stimulator circuitry.

Voltage-to-Current Converter

The capacitive DAC output voltage ranges from 0.3V to 0.9V (which matches the input range of the ADC). The DAC voltage is forced across an on-chip resistor $R$ of 5.5kΩ by OTA6 as shown Figure 4.12. The current gets multiplied by 10 by setting the current mirror width ratios of M2:M4 and M5:M6 to 1:10. The common reference voltage of the DAC is 0.3V, and thus the generated current is

$$I_{DAC} = 10 \left( V_{DAC} - 0.3V \right) / R \tag{4.7}$$

which ranges between 5µA and 1.09mA. This selection of $R$ can be adjusted for the intended application or electrode impedance. The on-chip resistor value can vary by 10 percent over process corners. With a large channel count, to adjust for this, the currents
for each channel can be read off and each channel can be digitally calibrated. The OTA6 in the voltage-to-current converter uses thick oxide 3.3V devices and a standard two-stage architecture.

**Current Driver**

For the current driver it is important to ensure that the positive and negative current pulses are matched closely to minimize excess charge at the electrode-tissue interface. It is also important to have a high impedance at the electrode node to ensure the current is less sensitive to the load variations. As shown in Figure 4.12(a), during the UP pulse, the current from the voltage-to-current converter is mirrored to the output transistor M4. This current is inversely proportional to the electrode voltage at the drain of M4. To match currents between the UP and DOWN pulses, OTA7 ensures the current during the DOWN pulse is also inversely proportional to the electrode voltage at the drain of M7. OTA7 was implemented using a five-transistor differential amplifier with thick-oxide devices.

A ratio of 1:10 was selected to implement the current mirrors to reduce area and power dissipation. Capacitor $C_c$ was added to stabilize the feedback loop with OTA7 across all operating conditions of the current driver. For this feedback loop, capacitor $C_c$ sets the dominant pole and thus lowers the unity gain frequency from 33MHz to 1.7MHz while improving the phase margin from -30 degrees to 74 degrees. An additional inversion due to the common source transistor M5 implements Miller compensation. For lower stimulation currents, the 2nd pole frequency is lowered and stability degrades. The worst-case operating condition (10 $\mu$A) has a phase margin of 54 degrees.

OTA7 can also be configured to remove any excess charge after the stimulation by applying the reference voltage to the electrode [127].
Figure 4.13: Micrograph of the 4mm × 3mm 0.13µm CMOS SoC.

Figure 4.14: (a) Layout and (b) micrograph of the recording and stimulation channel.
Stimulation Artifact

During neural stimulation the large currents that are generated in the neural tissue typically cause the recording amplifiers to saturate. Due to the low frequency (below 1Hz) high-pass filter response of the recording amplifier, a significant amount of time is required for the amplifier output to settle back to its normal range. This makes simultaneous recording/processing and stimulation impossible. Our approach is to invoke the neural stimulator for a predetermined short period of time after a seizure is detected to abort it. To ensure the recording channel can return to normal signal operating range as fast as possible, the gate of the feedback PMOS reset devices on both stages of the recording amplifier can be temporarily set to 0V right after the stimulation burst to temporarily reduce the settling time of the amplifier [87].

4.4 Electrical Experimental Results

The micrograph of the SoC implemented in a standard 1P8M IBM 0.13µm CMOS technology is shown in Figure 4.13. The SoC occupies an area of 4mm × 3mm. There is a total of 1kb of on-chip memory and approximately 1 million transistors. The SoC operates from a 1.2V supply for the neural recording, digital signal processing and RF transmitter and a 3.3V supply for the neural stimulator. The micrograph and layout of the full channel including the neural recording amplifier, SC bandpass filter, biphasic current driver, multiplying SAR ADC/DAC and a 22-bit memory are depicted in Figures 4.14(a) and (b). The full channel occupies 300µm × 300µm of area. Each channel has a bondpad to be flip-chip bonded directly to a microelectrode array.
Figure 4.15: Experimentally measured (a) input-referred noise, (b) frequency response of neural recording amplifier and (c) total-harmonic distortion (THD). (d) Experimentally measured output waveforms for a 20Hz 1mV_{pk–pk} input and (e) a 20Hz 20μV_{pk–pk} input.
Figure 4.16: Experimentally measured frequency response of the neural recording amplifier and the switched-capacitor bandpass filter.

4.4.1 Analog Front-end and Switched-Capacitor BPF

The experimentally measured input-referred noise, amplitude frequency response and total harmonic distortion (THD) of the neural recording channel are shown in Figures 4.15(a)-(c). The amplifier has an integrated input-referred noise of 4.7μV and 3.7μV (at room temperature) for the frequency bands of 10Hz to 5kHz and 1Hz to 100Hz, respectively. The NEF of the fully differential neural recording amplifier is 4.4 for the 5kHz bandwidth. The experimentally measured CMRR at 10Hz and 1kHz is 75.4dB and 71.5dB, respectively. The frequency response is consistent across multiple channels on the chip. Both the highpass and lowpass poles can be adjusted, with the maximum bandwidth of 0.01Hz to 10kHz. The experimentally measured THD is -50dB when the output is 700mV between 100Hz and 5kHz and drops to -35dB at 1Hz due to the non-linearity introduced by the feedback device of the second stage. The output waveforms for a 1mV_{pk−pk} 20Hz sinusoid and a 20μV_{pk−pk} 20Hz sinusoid inputs are shown in Figures 4.15(d) and (e), respectively. For the 1mV_{pk−pk} input (0.5V_{pk−pk} output), the distortion is not visible, with a THD below -50dB. Figure 4.15(e) depicts


Table 4.2: Comparison Neural Recording Amplifiers

<table>
<thead>
<tr>
<th>Spec.</th>
<th>[41]</th>
<th>[50]</th>
<th>[95]</th>
<th>[128]</th>
<th>THIS WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Diss.</td>
<td>42.2 µW</td>
<td>43 µW</td>
<td>15 µW</td>
<td>5.04 µW</td>
<td>3.8 µW</td>
</tr>
<tr>
<td>Tech.</td>
<td>0.5 µm</td>
<td>0.13 µm</td>
<td>0.35 µm</td>
<td>0.065 µm</td>
<td>0.13 µm</td>
</tr>
<tr>
<td>Supply (V)</td>
<td>3.3</td>
<td>1.2</td>
<td>3.3</td>
<td>0.5</td>
<td>1.2</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>0.16</td>
<td>0.2</td>
<td>0.04</td>
<td>0.013</td>
<td>0.022</td>
</tr>
<tr>
<td>CMRR (dB)</td>
<td>-</td>
<td>75</td>
<td>75.4</td>
<td>75</td>
<td>75.4</td>
</tr>
<tr>
<td>Noise (LFP)</td>
<td>5.1 µV</td>
<td>14 µV</td>
<td>7 µV</td>
<td>4.3 µV</td>
<td>3.7 µV</td>
</tr>
<tr>
<td>Noise (Spike)</td>
<td>-</td>
<td>2.2 µV</td>
<td>-</td>
<td>4.9 µV</td>
<td>4.7 µV</td>
</tr>
<tr>
<td>NEF</td>
<td>5.1</td>
<td>5.0</td>
<td>4.6</td>
<td>6</td>
<td>4.4</td>
</tr>
<tr>
<td>Fully Integrated</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

The output for a low-SNR signal of 20 µVpk−pk. The amplifier noise and high frequency ripple due to the chopper switching can be observed. A comparison to recently reported neural recording amplifiers is given in Table 4.2.

The experimentally measured frequency response of the neural recording amplifier and SC BPF is shown in Figure 4.16. Four different bands are shown when tuning the SC clock between 625 Hz and 5 kHz that yield the bandpass filter center frequency of 8 Hz, 16 Hz, 32 Hz and 64 Hz, respectively. The SC filter provides an additional 2 dB of gain to the analog front-end.

### 4.4.2 ADC and Full Recording Path

The experimentally measured FFT of an in-channel ADC is shown in Figure 4.17. The ENOB and SFDR are 7.6 bits and 62.4 dB, respectively. The ADC achieves close to 7 bits over the full Nyquist bandwidth with consistent performance across multiple channels on the chip. The experimentally measured INL and DNL are 0.6 and 0.7, respectively, as shown in Figure 4.18. The FFT of the full recording path output including the analog front-end, SC bandpass filter and the ADC is displayed in Figure 4.19(a). The 5 kHz harmonic clock of the SC BPF (CLK<sub>SC</sub>) is visible in the spectrum and limits the SFDR to 38 dB. As shown in Figure 4.19(b), when enabling the SC and RC lowpass filters and programming the FIR filter as a lowpass filter, the SFDR improves to 51 dB. A
larger second harmonic is introduced due to mismatch of utilizing eight different ADCs in the FIR filtering mode.

### 4.4.3 Neural Vector Analysis

The SoC was characterized to demonstrate the computation of magnitude, phase and phase synchronization. Figure 4.20 depicts experimental results for the full signal path from the LNA to the SC filter (set to 30Hz center frequency) then processed by two 16-tap FIR filters to generate the real and imaginary components. Figures 4.20(a) and (b) show the I/Q extraction for a low-amplitude (30µV) sinusoid signal. Figures 4.20(c) and (d) show the case for a larger amplitude input (500µV) sinusoid signal. Both cases yield an accurate 90 degree phase shift. The frequency response of the I/Q extraction measured starting from the input of the LNA is depicted in Figure 4.20(e) when the SC filter is centered at 30Hz. Due to the approximate response of the 16-tap FIR filters, exact gain matching between I and Q is not achieved with a difference of approximately 2dB, as expected in Matlab simulations. A higher order FIR filter achieves more accu-
rate gain matching at the expense of area and power dissipation.

Experimental results for the phase difference computation between two neural recording channels are shown in Figure 4.21(a). The two input sinusoid signals have an amplitude of 100µV and a frequency of 30Hz. The phase difference was varied using a Tektronix arbitrary waveform generator and the computed phase was plotted against the phase displayed on the source. In Figure 4.21(b), the two signals were placed at a 180 degree phase shift and the amplitudes of both signals were swept. At low SNR (10µV-inputs), the phase varied by 60 degrees over three standard deviations from the mean. At higher SNR (500µV-inputs), the phase varied by five degrees over three standard deviations from the mean.

The magnitude was computed for a single channel as shown in Figures 4.21(c) and (d). Figure 4.21(c) displays the extracted magnitude for different signal amplitudes from 12.5µV to 400µV and compares with the ideal magnitude. Figure 4.21(d) depicts the computed magnitude in response to a slow step-wise amplitude increase of only 25µV into the amplifier. The minimum noise floor for magnitude extraction is 8µVrms.
Figure 4.19: (a) Experimentally measured output FFT of an in-channel analog front-end, SC BPF and ADC with a 65Hz 700µV input sampled at 14kS/s. (b) Experimentally measured output FFT of an in-channel analog front-end, SC BPF, SC LPF, RC LPF and FIR LPF with a 65Hz 700µV input sampled at 7.2kS/s.
Figure 4.20: (a) A low-amplitude sinusoidal input into neural recording amplifier. (b) Experimentally measured I/Q separation of signal in (a) including full signal path of input (a). (c) A high-amplitude sinusoid input into neural recording amplifier. (d) Experimentally measured I/Q separation including full signal path of input (c). (e) Experimentally measured frequency response of full signal path for I and Q.
Figure 4.21: (a) Experimentally measured phase difference computation including full signal path (100µV inputs). (b) Experimentally measured phase difference error for different input amplitudes between two neural recording amplifier. (c) Experimentally measured magnitude computation including full signal path and compared with ideal case. (d) Experimentally measured magnitude computation of a low frequency step introduced at the input of the neural recording amplifier.
4.4.4 Phase Synchronization

The phase-locking value (PLV) was computed on-chip between two different inputs for three different frequency bands. The experimental results are shown in Figure 4.22(a). In this case, a 100µV sinusoid input was held constant at 8Hz, 30Hz and 120Hz with the SC BPF center frequency set to these frequencies. The other 100µV sinusoid signal frequency was swept and the PLV was computed. The PLV output is 1 as expected when both frequencies are locked. Any deviation away from the center frequency yields a lower phase synchronization as expected. In Figure 4.22(b), one 100µV sinusoid was set to 30Hz and the other 100µV input had its frequency linearly swept from 45Hz down to 15Hz. The PLV tracks the change in the frequency and outputs a 1 when both frequencies are 30Hz.

A comparison between the simulated and measured PLV is shown in Figure 4.23. The PLV result computed by the SoC and by the simulink model are in close agreement when the frequencies of the two inputs are within 20 percent or exhibiting a PLV above 0.7. As the two signal frequencies deviate by more than 20 percent, the error between the computed PLV and the Simulink PLV is approximately 0.1. Non-linearities between the two channels such as phase mismatch, gain mismatch and distortion in the analog front-end were not accounted for in the Simulink model.

4.4.5 UWB Transmitter

The experimentally measured on-chip Manchester-encoded input data and the corresponding transmitted UWB data are shown in Figure 4.24(a) transmitting at a data rate of 1Mb/s. The measured output spectrum and corresponding FCC mask for the UWB transmitter operating in the lower-frequency DC to 1GHz UWB frequency band is depicted in Figure 4.24(b). Figure 4.25(a) shows transmitted 10MB/s Manchester encoded-data. Figure 4.25(b) depicts the wirelessly received UWB modulated data uti-
Figure 4.22: (a) Experimentally measured PLV computation between two different channels for three different frequency bands. (b) Experimentally measured PLV computed between two channels, where one frequency is constant for one channel and the other frequency is increasing for the other channel.
**Figure 4.23:** Comparison between simulated and experimentally measured PLV computed between two channels, where one frequency is constant for one channel and the other frequency is increasing for the other channel.

**Figure 4.24:** (a) Wireless TX digital Manchester-encoded data and modulated UWB pulses. (b) Experimentally measured Output spectrum of UWB pulse into 50Ω load.
Figure 4.25: (a) Wireless TX digital Manchester-encoded data. (b) Experimentally wirelessly received data from a 5cm distance @ 10Mb/s in the 3.1-10.4GHz band.
lizing the 3.1-10.4GHz UWB frequency band across a 5cm distance.

### 4.4.6 Neural Stimulator

The neural stimulator reuses the same MDAC as the SAR MADC and exhibits an INL and DNL below 1 LSB as depicted in Figure 4.18. The neural stimulator was characterized experimentally using both a resistive load and a saline solution. In Figure 4.26(a) the DAC was set to output 225μA and the electrode voltage was swept from 0.3 to 2.7V. The results show a close match between the positive and negative current pulses. The UP and DOWN currents are accurately matched when the electrode voltage swings as high as 2.0V. If a lower current is used (<100μA), the currents are matched when the electrode voltage swings as high as 2.5V. When the output voltage is high the UP current transistor in Figure 4.12(a) enters the triode region and no longer sustains the proper current. Similarly, when the output voltage is low the DOWN current transistor, M6 enters the triode region and can no longer sustain the proper current. When the output voltage is high, OTA7 can no longer maintain the proper loop gain. By disabling OTA7 at high output voltage, the swing can be extended at the expense of lower current matching. The output current for different DAC configurations is depicted in Figure 4.26(b). The UP and DOWN currents up to 1mA are closely matched. Figures 4.26(c) and (d) show various amplitude and duty cycle configurations for the neural stimulator while generating a biphasic current pulse into a saline solution. Lastly, the voltage across a 1kΩ resistor is shown on the oscilloscope plot in Figure 4.26(e) when the biphasic neural stimulator is programmed for a 450μA current.

### 4.4.7 Offline Early Seizure Detection in Humans

Results of offline early seizure detection in human ECoG data from a University of Toronto patient are shown in Figure 4.27. For spontaneous seizures in humans the SC
Figure 4.26: (a) Experimentally measured output current for different output voltages. (b) Output current vs. DAC input for positive and negative current cases. (c), (d) Experimentally measured neural stimulating current into a saline solution for different duty cycle and amplitude control settings. (e) Biphasic stimulator set to 450μA output current and connected to a 1kΩ load.
CHAPTER 4. WIRELESS NEURAL VECTOR ANALYZER AND PHASE SYNCHRONY-TRIGGERED STIMULATOR SoC

Figure 4.27: Offline human seizure detection for a University of Toronto seizure patient.

BPF center frequency was set to 16Hz to capture abnormal synchrony at that frequency. The two 16Hz magnitude outputs both increase early in and during the seizure. The synchrony in the 16Hz frequency band between the two inputs also increases early in and during the seizure. This trend was observed in multiple patients.
Figure 4.28: (a) Power dissipation breakdown of the SoC (neural stimulator excluded). (b) Area breakdown of the SoC. The remaining 5mm\(^2\) are utilized for I/O, ESD, test blocks, routing, decoupling capacitors and biasing.
### Table 4.3: Summary of the experimental results

<table>
<thead>
<tr>
<th>System:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>IBM 0.13μm</td>
</tr>
<tr>
<td>Supply Voltage (Rec.)</td>
<td>1.2V</td>
</tr>
<tr>
<td>Supply Voltage (Stim.)</td>
<td>3.3V</td>
</tr>
<tr>
<td>Die Dimensions</td>
<td>4.0mm × 3.0mm</td>
</tr>
<tr>
<td>Area per Channel</td>
<td>300μm × 300μm</td>
</tr>
<tr>
<td>Number of Recording Channels</td>
<td>64</td>
</tr>
<tr>
<td>Number of Stimulation Channels</td>
<td>64</td>
</tr>
<tr>
<td>On-Chip Memory</td>
<td>1kb</td>
</tr>
<tr>
<td>Number of Transistors</td>
<td>1 Million</td>
</tr>
<tr>
<td>Power Dissipation (Rec.)</td>
<td>1.4mW</td>
</tr>
<tr>
<td>Power Density (Rec.)</td>
<td>0.11mW/mm²</td>
</tr>
<tr>
<td>Power Dissipation (Stim.)</td>
<td>1.5mW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recording Channel:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
<td>54-60dB</td>
</tr>
<tr>
<td>Input-Ref. Noise (1Hz-5kHz)</td>
<td>5.1μV</td>
</tr>
<tr>
<td>Input-Ref. Noise (1Hz-100Hz)</td>
<td>3.5μV</td>
</tr>
<tr>
<td>LNA NEF</td>
<td>4.4</td>
</tr>
<tr>
<td>CMRR</td>
<td>75dB</td>
</tr>
<tr>
<td>THD @ 1kHz</td>
<td>0.3%, 0.9V output</td>
</tr>
<tr>
<td>ADC SNDR</td>
<td>47.5</td>
</tr>
<tr>
<td>ADC SFDR</td>
<td>62.5</td>
</tr>
<tr>
<td>ADC ENOB</td>
<td>7.6-bits</td>
</tr>
<tr>
<td>ADC INL</td>
<td>0.7LSB</td>
</tr>
<tr>
<td>ADC DNL</td>
<td>0.6LSB</td>
</tr>
<tr>
<td>ADC Sample Rate</td>
<td>up to 100kS/s</td>
</tr>
<tr>
<td>Power Diss. (LNA+SCF+ADC)</td>
<td>10μW</td>
</tr>
</tbody>
</table>

**Signal Processing:**

- **Analog Filters**: 64x high-Q SC BPF
- **Number of FIR filters**: 64
- **FIR filter type**: Symmetric 16-tap
- **Coefficient Resolution**: 8-bit signed
- **Processor**: 3xCORDIC
- **Univariate Operations**: 64xMag,Phase
- **Bivariate Operations**: 32xPhase diff,PLV
- **Power Dissipation**: 0.26mW (64-chan. 30Hz band)

**Neural Stimulation:**

- **Type**: Biphasic current
- **Voltage Compliance**: 2V
- **Current Matching**: 1%
- **Current Range**: 10μA-1.0mA
- **DAC res.**: 8-bits
- **Duty Cycle res.**: 4-bits

**Wireless TX:**

- **Modulation**: UWB
- **Freq. Band**: 0-1GHz/3.1-10.4GHz
- **Data Rate**: 10Mb/s
- **Power diss.**: 100μW
4.5 Discussion

4.5.1 Resource Utilization

A summary of experimental results is given in Table 4.3. The power dissipation and area breakdown of the entire SoC are shown in Figures 4.28(a) and (b), respectively. The SoC operating in the neural vector analysis mode with the UWB TX on dissipates 1.4mW for 64-channels or 21.6µW per channel when running the FIR filters at 7.2kS/s from a 1.2V supply. The analog feed-forward path (including biasing) from the LNA to the output of the ADCs dissipates 886µW or 13.8µW per channel. The digital adders and registers for the FIR filters, digital controller and the CORDIC processor dissipate 400µW or 6.25µW per channel.

The neural stimulator (not included in Figure 4.28(a)) dissipates 1.5mW from a 3.3V supply for all 64-channels or 23µW per channel. This includes the DAC, control logic, V-I converter and output current driver operating with a 1kHz 50µA biphasic current at minimum duty cycle. The neural stimulator is not expected to be active during routine operation of the SoC and thus its power dissipation is not as critical when compared with the recording and signal processing functionalities. For a 10mW budget delivered directly by a wireless power unit, this would allow for 64 channels with 1kHz continuous stimulation with a maximum current amplitude of 380µA per channel at minimum duty cycle or 50µA per channel at 50 percent duty cycle. For 50 percent duty cycle and maximum current across all 64-channels, the SoC dissipates 250mW. In this case, the stimulation pulses can be delivered in very short bursts to minimize tissue heating. For such high power requirements, the wireless power unit could be used to charge a battery such as the rechargeable 4.5g 18mm×31mm battery that can handle peak currents as high as 1.5A [102] or a smaller 200mg, 10mm×10mm battery that can handle peak currents as high as 12mA [101].
Power dissipation can be reduced from 21.6\(\mu\)W to 18\(\mu\)W per channel by operating the digital FIR adder-and-delay logic and the CORDIC processor from a lower 0.8V supply at the expense of requiring another voltage regulator. It can also be further minimized by using low-power logic (requiring LP devices available in this process) for all on-chip digital blocks. The neural stimulator can operate at a lower supply voltage (2.5V) to further minimize power dissipation at the expense of voltage compliance.

In terms of integration area (excluding routing, IO pads, decoupling capacitors, etc.), the 64 neural recording amplifiers, 64 ADCs and 64 16-tap 12-bit digital adders and registers for the FIR filters each utilize approximately 1/4 of the total active area as shown in Figure 4.28(b). The area allocation is well balanced. The large area required by the conventional digital multipliers for implementing 64 16-tap FIR filters is saved as the multiplication is performed within the ADC conversion cycle. Adding 64 channels of neural stimulation only requires additional 6 percent of area, as the DAC and duty-cycle controller are reused from the ADC. The SC BPF, SC LPF and RC LPF and buffer stage utilize a combined 20 percent of the area. The phase-synchrony processor requires only 3.5 percent of the total area due to using a CORDIC-based processor. This architecture minimizes the area and power dissipation while trading off speed. The place-and-route and timing was optimized for area and power dissipation as the operating frequency was far below the limits for this CMOS process.

### 4.5.2 Comparison to the State of the Art

A comparison with other neural recording and/or stimulation SoCs is given in Table 4.4. This work demonstrates the highest degree of integration among recently published state-of-the-art SoCs by combining 64 recording channels with SC bandpass filtering, 64 stimulation channels, 64 multiplying SAR ADCs, 64 16-tap FIR filters, a tri-core CORDIC processor and a UWB transmitter. It is the first published SoC to compute
and monitor neural signals phase and phase synchronization.

It also computes the magnitude or energy in a frequency band of up to 5kHz across 64-channels. The work in [94], [93], [48] and [72] demonstrate 16, 8, 18 and 4 channels, respectively. These designs also lack a neural stimulator and a wireless transmitter. The work described in [72] was also combined with a Medtronic DBS to implement a closed-loop neurostimulator utilizing multiple ICs [129] and not a single SoC. Even with the high degree of integration the presented design demonstrates comparable power dissipation and area. The SoC also demonstrates seizure detection and seizure abortion in vivo in freely moving rats.

### 4.6 Summary of Novel Contributions

- A wireless closed-loop SoC for monitoring and treatment of epilepsy is demonstrated. This SoC integrates 64 neural recording amplifiers, 64 16-tap FIR filters, a CORDIC-based processor for seizure detection, 64 neurostimulators and a UWB transmitter. This work implements the first neural vector analyzer and the first closed-loop SoC utilizing phase synchrony. It also demonstrates the highest degree of integration among recently published state-of-the-art neural interface SoCs.

- The compact, low-power and programmable mixed-signal FIR filter implementation from Chapter 2, was used in conjunction with SC analog filters to implement and area and power-efficient I/Q generator. This enables vector processing on low-frequency neural signals.

- A compact chopper-stabilized neural amplifier with a low-distortion high-pass filter is demonstrated. A trade-off between area and noise is made to ensure scalability for a large number of recording channels.
### Table 4.4: State-of-the-art neural recording and/or stimulation SoCs

<table>
<thead>
<tr>
<th>Spec.</th>
<th>[85]</th>
<th>[93]</th>
<th>[94]</th>
<th>[42]</th>
<th>[44]</th>
<th>[72]</th>
<th>[55]</th>
<th>[48]</th>
<th>[54]</th>
<th>[130]</th>
<th>Chapter 2</th>
<th>THIS WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pwr. Diss.(mW)</td>
<td>0.47</td>
<td>-</td>
<td>0.253</td>
<td>7.05</td>
<td>6.0</td>
<td>0.005</td>
<td>0.375</td>
<td>0.077</td>
<td>0.27</td>
<td>6.5</td>
<td>5.03</td>
<td>1.4/1.5</td>
</tr>
<tr>
<td>Tech.(µm)</td>
<td>0.18</td>
<td>0.18</td>
<td>0.090</td>
<td>0.5</td>
<td>0.35</td>
<td>0.8</td>
<td>0.35</td>
<td>0.18</td>
<td>0.18</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Area(mm)²</td>
<td>4</td>
<td>25</td>
<td>14.8</td>
<td>18.4</td>
<td>6.5</td>
<td>18</td>
<td>10.9</td>
<td>6.25</td>
<td>4.0</td>
<td>25</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Supply(V)</td>
<td>1.8</td>
<td>1.8</td>
<td>1.0</td>
<td>3.0</td>
<td>3.3</td>
<td>1.8</td>
<td>1.5</td>
<td>1.0</td>
<td>1.8</td>
<td>1.2</td>
<td>1.2</td>
<td>1.2/3.3</td>
</tr>
<tr>
<td># Record Chan.</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>128</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>96</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Pwr/chan.(µW)</td>
<td>61</td>
<td>2.5</td>
<td>10.3</td>
<td>75</td>
<td>11</td>
<td>4.5</td>
<td>32.8</td>
<td>4</td>
<td>9</td>
<td>68</td>
<td>6.3</td>
<td>10</td>
</tr>
<tr>
<td>Gain(dB)</td>
<td>54</td>
<td>52</td>
<td>40</td>
<td>67.8</td>
<td>57-60</td>
<td>-</td>
<td>52-66</td>
<td>72</td>
<td>-</td>
<td>56</td>
<td>54-60</td>
<td>54-60</td>
</tr>
<tr>
<td>Noise(µVrms)</td>
<td>-</td>
<td>0.91</td>
<td>-</td>
<td>4.4</td>
<td>4.9</td>
<td>1</td>
<td>3.1</td>
<td>1.3</td>
<td>-</td>
<td>2.2</td>
<td>6.5</td>
<td>5.1</td>
</tr>
<tr>
<td>$f_{L3dB}$ (Hz)</td>
<td>0.6</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>-</td>
<td>280</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>$f_{H3dB}$ (Hz)</td>
<td>6k</td>
<td>100</td>
<td>-</td>
<td>10k</td>
<td>5k</td>
<td>100</td>
<td>12k</td>
<td>100</td>
<td>-</td>
<td>10k</td>
<td>5k</td>
<td>5k</td>
</tr>
<tr>
<td>CMRR(db)</td>
<td>-</td>
<td>90</td>
<td>-</td>
<td>134</td>
<td>90</td>
<td>80</td>
<td>58</td>
<td>60</td>
<td>-</td>
<td>75</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>ADC res./SNDR (dB)</td>
<td>35.5</td>
<td>10bit</td>
<td>9bit</td>
<td>PWM</td>
<td>9bit</td>
<td>-</td>
<td>54.7</td>
<td>65</td>
<td>44</td>
<td>60.3</td>
<td>48.5</td>
<td>47.5</td>
</tr>
<tr>
<td>SC Filter</td>
<td>-</td>
<td>LPF</td>
<td>LPF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>BPF</td>
<td>BPF</td>
<td>BPF</td>
<td>-</td>
<td>BPF</td>
<td></td>
</tr>
<tr>
<td>Signal Proc.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FIR Filters</td>
<td>2</td>
<td>64x</td>
<td>2x</td>
<td>-</td>
<td>1x</td>
<td>-</td>
<td>-</td>
<td>7x48T</td>
<td>1x22T</td>
<td>-</td>
<td>64x16T</td>
<td>64x16T</td>
</tr>
<tr>
<td>IIR Filters</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1x1T</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>Log</td>
<td>Classification</td>
<td>GP</td>
<td>-</td>
<td>Feature</td>
<td>-</td>
<td>Analog</td>
<td>-</td>
<td>Feature</td>
<td>-</td>
<td>Trig. CORDIC</td>
<td></td>
</tr>
<tr>
<td>Univariate</td>
<td>4xMag</td>
<td>SVM</td>
<td>16xMag,</td>
<td>-</td>
<td>1xSpike</td>
<td>4x</td>
<td>4x</td>
<td>1x</td>
<td>-</td>
<td>64x</td>
<td>64xMag</td>
<td></td>
</tr>
<tr>
<td>Bivariate</td>
<td>-</td>
<td>8xSpike</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>32x∆ϕ</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>32xPLV</td>
<td></td>
</tr>
<tr>
<td>Mag. Floor</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2µV</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8µV</td>
<td></td>
</tr>
<tr>
<td>Closed-loop</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Stimulation</td>
<td># Stim. Chan.</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>DAC Res.(bits)</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Compliance(V)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.2</td>
<td>-</td>
<td>1.8</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Current(mA)</td>
<td>0.1 or 4.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.01-0.1</td>
<td>-</td>
<td>0.01-1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duty Cycle Res.</td>
<td>6-bit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>6-bit</td>
<td>-</td>
<td>-</td>
<td>4-bit</td>
<td></td>
</tr>
<tr>
<td>Wireless TX</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modulation</td>
<td>FSK</td>
<td>-</td>
<td>-</td>
<td>FSK</td>
<td>UWB</td>
<td>-</td>
<td>FSK</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>FSK</td>
<td>UWB</td>
</tr>
<tr>
<td>Date-rate(Mb/s)</td>
<td>0.8</td>
<td>-</td>
<td>-</td>
<td>90</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.5</td>
<td>10</td>
</tr>
<tr>
<td>TX Pwr.(mW)</td>
<td>0.005</td>
<td>-</td>
<td>3.3</td>
<td>1.6</td>
<td>0.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.7</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>
• An area and power-efficient stimulator for integrated neural recording and stimulation arrays is demonstrated. This work utilizes sharing of the ADC and DAC during the neural recording and stimulation phases to perform amplitude and duty cycle control with little area and power overhead.

4.7 Conclusion

A 0.13µm CMOS wireless closed-loop neural stimulation SoC for the treatment of intractable epilepsy is presented. The 12mm² die integrates 64 fully differential recording amplifiers with in-channel bandpass filtering with 64 multiplying SAR ADCs, 64 FIR filters, 64 neural stimulators, a tri-core CORDIC processor, 1kB of memory and a UWB RF transmitter. A novel low-noise and compact chopper-stabilized recording amplifier, a novel DAC-sharing stimulator architecture and duty cycle controller and a novel area and power-efficient technique to generate low-frequency I/Q components was demonstrated. It demonstrates the first SoC implementation for a neural vector analyzer and computes phase synchrony between inputs to trigger a closed-loop biphasic current stimulation. The total power dissipation is 1.4mW from a 1.2V supply in the recording mode and 1.5mW from a 3.3V supply in the stimulation mode. The system was also characterized in vivo and demonstrates seizure detection and abortion in freely moving rodents and offline seizure detection in humans.
Chapter 5

In Vivo Validation for Epileptic Treatment

5.1 Introduction

In this chapter we present *in vivo* results validating the utility of the closed-loop SoC presented in Chapter 4 in treating intractable epilepsy. The SoC (described in Chapter 4) operates as a neural vector analyzer to process the magnitude and phase of neural inputs and to compute the phase synchrony between a pair of neural inputs. An example of this is described in detail in Figure 1.10. The magnitude, phase and the phase synchrony is used to trigger the neural stimulator to abort or cancel a seizure. This chapter demonstrates the *in vivo* validation of the closed-loop SoC for epilepsy treatment in rats (epilepsy rat models).

The chapter is organized as follows. Section 5.2 describes the closed-loop system and the prototype SoC. Section 5.3 presents *in-vivo* results on rats to validate the closed-loop SoC.
5.2 Closed-loop Early Seizure Detector SoC

5.2.1 Early Seizure Detection Algorithm

The seizure detection algorithm is described in detail in Chapter 3. The efficacy of the phase synchronization processor VLSI architecture in early seizure detection was verified on the intracranial EEG database from the international seizure prediction project from University of Freiburg. A 70 percent detection rate is obtained when the false positives are set to the rate of 0.6 false positives per hour. The detection rate approaches 100 percent when the false positives rate is set to 1.2-2 false positives per hour. This is comparable to other algorithms for early seizure detection (as described in detail in Chapter 3), with the key advantage that our approach utilizes a low-power SoC that computes on line and is implanted, as opposed to existing off-line software programs running on bulky computers.

The closed-loop seizure detection SoC is described in detail in Chapter 4. It consists of 64 neural recording and 64 neural stimulator channels. Two pairs of neural inputs get amplified and then bandpass filtered using a high-Q switched capacitor bandpass filter. The center of the bandpass filter is set to a specific band in the neural spectrum. For our in vivo experiments (described in Section 5.3), the bandpass filter was set to either 4Hz or 8Hz. Next, the two inputs are passed to two pairs of mixed-signal FIR filters, an all-pass and Hilbert FIR filter, to extract the real and imaginary components. The FIR multiplication is performed in the analog domain within the energy efficient SAR ADC. The add-and-delay blocks of the FIR filters are implemented in the digital domain. This FIR implementation minimizes the area and power dissipation as described in Chapter 2. Next, the real and imaginary components are fed to the on-chip 10-bit CORDIC-based phase synchronization and magnitude processor. A threshold detector can be programmed to trigger the neural stimulator in the feedback to form a closed-
loop system. To complete the system all digital data can be transmitter wirelessly using an on-chip 3.1-10.4GHz UWB transmitter.

5.3 Validation of the Prototype \textit{In vivo}

5.3.1 \textit{In Vivo} Validation of Seizure Detection Algorithm based on GBL Model

Two models of absence-like epilepsy were used. Both conditions result in abundant paroxysmal activity, and differ in their nature: one is a chronic model (rats will have seizures all their life), while the other is an acute model (the animals will experience
seizures during the day of the drug injection) and is discussed in Section 5.4.1. The first is a model of atypical absence seizures. The model results from the injection of gamma-hydroxybutyric acid (GHB), and the absence-like seizure activity is considered to represent the clinical condition of typical absence epilepsy. Long Evans rats are administered 100-120mg/kg of gamma-butyrolactone (GBL, which is a precursor of GHB) and their brain activity is monitored 2 to 6 hours thereafter. Typical absence seizures consisting of bilaterally synchronous spike-and-wave discharges associated with facial myoclonus, vibrissal twitching, and behavioural arrest, have their onset within 5 minutes of GBL administration. Manipulations of the rats are performed according to the protocols approved by the Animal Care Committee of the Hospital for Sick Children.

Four bipolar electrodes (Plastics One) were implanted chronically into specific brain areas of Long Evans rats (50 to 90 days old) using a stereotaxic apparatus. The coordinates of the implanted electrodes are summarized in Table 5.1. Three electrodes connect to the closed-loop SoC. These electrodes connect to different recording channels and provide a reference voltage. The other electrode connects to a differential CyberAmp 380 signal conditioner (Axon Instruments), and an analog to digital converter (MP100) to cross-validate the results. The in-vivo test setup is shown in Figure 5.1. Seizures were identified by examining the rat’s electrical activity recorded by the benchtop amplifier and by observing the rat’s seizure-like symptoms, both by trained clinicians.

The pharmacological agent GBL was injected into the hippocampus of a rat with a dose of 100-120mg/kg when time is equal to 0 seconds to induce non-convulsive seizure-like events as shown in Figure 5.2. Neural activity in the 4Hz frequency band was recorded and processed by the SoC. For approximately 3 minutes after the injection, the two 4Hz magnitude outputs shows minimal activity. At time, t=50s, 172s before the seizure starts, the PLV at 4Hz computed by the chip increases, resulting in an early seizure detection. Approximately 3 minutes after the injection, the magnitude and phase
synchronization in the 4Hz frequency band increases before and during the seizure-like event. After injection of GBL, the processed magnitude and phase synchrony in the 8Hz frequency band both increase as the freely moving rat experiences absence-like seizure activity. The amplified and bandpass filtered (at 8Hz) output of two channels is shown in Figure 5.3[top]. This GBL seizure model was used to validate the neural vector analyzer and the phase synchrony extractor of the SoC \textit{in vivo}. The amplified and bandpass filtered output of two channels were processed online in real-time by the SoC and is shown in Figure 5.3[bottom] (highlighted in blue). As well, the amplified and bandpass filtered output was fed offline into a SIMULINK model of the processor.
Figure 5.3: Example of magnitude and phase synchrony extraction on a freely moving rat injected by GHB. Comparison of offline SIMULINK model with online realtime SoC.
Figure 5.4: Example of an early seizure detection example on a freely moving rat, in vivo. Seizure triggered by a kainic acid injection.
Figure 5.5: Example of an early seizure detection example on a freely moving rat, \textit{in vivo}. Seizure triggered by a kainic acid injection.
Figure 5.6: Example of an early seizure detection example on a freely moving rat, \textit{in vivo}. Seizure triggered by a kainic acid injection.
(described in Chapter 3) and is depicted in Figure 5.3[bottom] (highlighted in red). A few points of interest were marked on both plots for comparison. The two peaks on the magnitudes match closely between the SIMULINK model and the SoC. The phase synchrony has two peaks higher than a threshold. Both these peaks match closely between the SIMULINK model and the SoC.

5.3.2 In vivo Validation based on Kainic Acid Model

In addition to the absence seizure models, a chronic model of a different type of seizure syndrome, that simulates limbic epilepsy in humans, was used. The reason is that most of malignant human seizures are of this type, while absences are considered sort of benign. This model is normally termed in the literature the self-sustained status epilepticus model in rats. It is can be caused by injection of kainic acid. Due to the spontaneous nature of the seizures with this model, it can be used to validate the early seizure detector and neural stimulator.

Four bipolar electrodes (Plastics One) were implanted chronically into specific brain areas of Long Evans rats (5090 days old) using a stereotaxic apparatus. The coordinates of the implanted electrodes are summarized in Table 5.1. Three electrodes connect to the closed-loop SoC. These electrodes connect to different recording channels and provide a reference voltage. The other electrode connects to a differential CyberAmp 380 signal conditioner (Axon Instruments), and an analog to digital converter (MP100) to cross-validate the results. The in vivo test setup is shown in Figure 5.1. Seizures were identified by examining the rat’s electrical activity recorded by the the benthtop amplifier and by observing the rat’s seizure-like symptoms, both by trained clinicians.
5.4 Experimental Validation in Epilepsy Control

The SoC was demonstrated experimentally in both detection and control of epileptic seizures. An online animal epilepsy model (in vivo freely-moving rodents) and offline human ECoG data from epilepsy patients were used to validate the SoC.

5.4.1 Online Early Seizure Detection and Control in Rodents In Vivo

Early Seizure Detection in Rats

Rats were injected with a dose of kainic acid. The administered dose was a 15-20mg/kg intraperitoneal injection. The SoC provided voltage amplification and bandpass filtering of different selected frequency bands. The information in these frequency bands were then processed by the SoC in real-time on the same chip. The magnitude of two neural inputs and the phase synchrony between two neural inputs were computed by the SoC. Three examples of online early seizure detection for the same rat are discussed. In Figure 5.4, the raw recording by the benchtop amplifier after offline highpass filtering is shown. The rat had a seizure lasting 60 seconds starting at a time of 130 seconds after being administered with a kainic acid injection at time t=-5min. The SoC processed the 8Hz frequency band with the analog output of two neural inputs shown in Figure 5.4. The real-time magnitude and synchrony processing is shown in Figure 5.4 in the 8Hz frequency band. The processed magnitude of both inputs increase during the seizure. The synchrony between the two inputs increases above a threshold 10 seconds before the electrical seizure onset, resulting in an early seizure detection. A second example is shown in Figure 5.5 with a seizure lasting 35 seconds starting at a time of 210 seconds. The SoC processed the 8Hz frequency band. The processed magnitude by the SoC for both inputs yielded a negligible increase and could not be used to detect a seizure. The
processed phase synchrony increased above a threshold 10 seconds before the electrical seizure onset. A third example is shown in Figure 5.6 with a seizure lasting 45 seconds starting at a time of 10 seconds. The SoC processed the 4Hz frequency band. In this case, the magnitude increased during the seizure and the synchrony increased above a threshold 4 seconds before the electrical seizure onset. For this rat, the real-time processing of phase synchrony between two inputs could be used to provide early seizure detection of between 4 and 10 seconds before the seizure onset.

**Seizure Control and Abortion in Rats**

For this case, each input can be configured for neural recording and stimulation. Once a seizure was identified electrically and visually through seizure-like symptoms, the neural stimulator was invoked by providing a low-frequency 5Hz, 100μA biphasic current. Once the neural stimulation is invoked, the recording and processing capabilities of the
SoC are disabled (the SoCs amplifiers are saturated) and the benchtop amplifier is used to analyze the results.

Four different seizures were observed. The first example is displayed in Figure 5.7 and shows the neural stimulator aborting a seizure instantaneously after the neural stimulator is invoked. The other three examples demonstrate closed-loop operation with the SoC processing the 8Hz frequency band. The second example is displayed in Figure 5.8. The magnitude and synchrony in the 8Hz frequency band both increase before the seizure by 30-40 seconds. After 10 seconds, the neural stimulator was invoked and the seizure symptoms and electrical activity was aborted after 30 seconds. The third example is displayed in Figure 5.9. Both the magnitude and synchrony in the 8Hz frequency band increase before the seizure by 50 seconds. After 10 seconds, the neural stimulator was invoked and the seizure symptoms and electrical activity was aborted after 3 seconds. The last example is displayed in Figure 5.10. In this case, both the magnitude and synchrony in the 8Hz frequency band gradually increase before the seizure by at least 30 seconds. After 6 seconds, the neural stimulator was invoked and the seizure symptoms and electrical activity was aborted after 2 seconds. For this rat, the real-time processing of phase synchrony between two inputs could be used to provide early seizure detection of between 30 and 50 seconds before the seizure onset. As well a total of 80% of seizures were aborted through neural stimulation.

5.5 Conclusions

Validation of epileptic treatment utilizing a single SoC was demonstrated in vivo on rats. Experimental early seizure detection and closed-loop seizure abortion were both presented in vivo on rats. The SoC can record and process the magnitude, phase and phase synchrony of 64 neural inputs in real time. Similarly, the SoC can operate as a biphasic
Figure 5.8: Example of closed-loop neural stimulation aborting a seizure for a freely moving rat in vivo. Seizure triggered by a kainic acid injection.
CHAPTER 5. *In Vivo* Validation for Epileptic Treatment

Figure 5.9: Example of closed-loop neural stimulation aborting a seizure for a freely moving rat *in vivo*. Seizure triggered by a kainic acid injection.
Figure 5.10: Example of closed-loop neural stimulation aborting a seizure for a freely moving rat in vivo. Seizure triggered by a kainic acid injection.
neural stimulator on up to 64 channels. A significant amount of hardware resources are shared between the recording/processing and stimulation phases to minimize the area and power dissipation of the SoC.
Chapter 6

Conclusions and Future Work

6.1 Contributions and Relevant Publications

This dissertation described the design and characterization of two wireless neural recording and/or stimulation SoCs. It also describes a novel VLSI architecture for a bivariate early seizure detector. Lastly, the SoCs were experimentally validated in vivo using rodents to demonstrate the application of seizure detection and seizure abortion.

- Chapter 2 described a wireless 64-channel neural recording SoC in 0.13\(\mu\)m CMOS. The low noise recording amplifiers make use of a novel low-distortion pseudo-resistor to maintain a robust low frequency pole. A novel mixed-signal FIR filter implementation eliminates the need for 512 conventional 8-bit digital multipliers by efficiently implementing the multiplication within the ADC conversion cycle. This results in over an order of magnitude savings in the power-area product. In summary, by minimizing both the area and power dissipation, an SoC that successfully integrates a large number of neural recording channels with advanced signal processing and a robust fully integrated wireless link onto a single chip is demonstrated. This work was presented in the 2011 IEEE European Solid-State
Chapter 3 introduced the system architecture of a bivariate early seizure detection algorithm that could be implemented in VLSI. Offline human intracranial EEG data demonstrate the effectiveness of the processor in early seizure detection. The VLSI-based seizure detector had performance similar to software-based seizure detectors in recent literature. This work was presented in the 2010 IEEE Biomedical Circuits and Systems conference (BIOMCAS) and 2011 IEEE Conference on Neural Engineering. An expanded journal paper was published in the 2011 (Volume 5, Issue 5), IEEE Transactions on Biomedical Circuits and Systems.

Chapter 4 demonstrated the first reported neural vector analyzer and phase synchrony processor. As well it was the first reported closed-loop neural stimulator SoC based on bivariate signal processing. A novel low-noise and compact chopper-stabilized recording amplifier, a novel DAC-sharing stimulator architecture and duty cycle controller and a novel area and power-efficient technique to generate low-frequency I/Q components was demonstrated. Compared with previous neural recording SoCs, the presented design offers the most processing and functionality of neural recording SoCs published to date while dissipating 1.4mW and 1.5mW across 64-channels in neural recording and stimulation modes, respectively. The full SoC is implemented on a $12mm^2$ 0.13µm CMOS die and utilizes the FIR filtering technique of Chapter 2, and a DAC-sharing technique for the neural stimulator to minimize area and power dissipation. The DAC-shared neural stimulator and chopper-stabilized low-distortion amplifier were presented in 2011 and 2012 ISCAS, respectively. The full SoC was presented in the 2012 IEEE European Solid-State Circuits Conference (ESSCIRC).

Chapter 5 demonstrated the application of this research. In vivo testing on freely...
moving rats validated the SoC as a early seizure detector and as a possible treatment for epilepsy in the future.

6.2 Future Work

6.2.1 Improvements to SoCs

Both SoC prototypes have a large number of debug I/O pins and off-chip biasing to simplify testing. After extensive experimental validation and validation of the SoCs, biasing can be placed on-chip with bandgap references and the debug I/O pins can be removed. As well, integrating the wireless power unit in Figure 6.1 onto the SoC of Chapter 2 and 4, further reduces the number of off-chip components, further decreasing the form-factor of the system and reducing its overall system complexity. More extensive testing to include bit error rate, interference and tissue loss of both wireless transmitters should be performed. Also adding coding to the data such as through the Viterbi algorithm would further make the wireless link more robust.

During neural stimulation, the recording and processing functionality is not functional as the stimulation artifact saturates the recording amplifiers. This results in having to set a time period for the neural stimulator once a seizure is detected. This periodic neural stimulation can often be either too long (leading to higher power dissipation) or too short (seizure is not aborted). Techniques to enable true bidirectional recording and stimulation would significantly improve the functionality of the SoC.

A multi-sensor and multi-channel biomedical IC to record and process a wide variety of signals such as ECG, EMG, EEG, ECoG, spikes and neurotransmitter currents would provide more flexibility and applications. These signals require different bandwidths and dynamic range requirements. Currently in this work, the power dissipation of the ADC, the FIR filters and the phase synchrony processor scale linearly, only with
frequency. One challenge is to provide a power-scalable analog front-end to address these different noise and frequency requirements. The noise is largely dominated by flicker noise for these biomedical signals and thus AC-coupled chopper-modulated amplifier with power scaling needs to be implemented. This has not been implemented to date. The ADC resolution, sample rate and power dissipation should also be scaled to accommodate different dynamic range requirements.

Analog improvements include reduction of flicker noise and reduction of the on-chip capacitors of the analog frontend by utilizing a new amplifier architecture. Digital improvements include increasing the FIR filter length to implement a more accurate Hilbert and allpass FIR filter and increasing the moving-average FIR filter length of the phase synchronization processor further improves the accuracy of the algorithm as described in Chapter 3 at the expense of die area. Adding a support vector machine instead of a threshold decision circuit further improves the seizure detection algorithm. Lastly, integrating data compression algorithms would help reduce the data rate for the wireless transmitter.

6.2.2 Wireless Inductive Power and Receiver IC

A third IC co-designed with Hamed Mazhab Jafari was fabricated in IBM 0.13\(\mu\)m CMOS as shown in Figure 6.1. The 2mm\(^2\) chip contains a wireless data receiver, a wireless data transmitter and a link for wireless inductive power. This chip is to provide a wireless power supply and wireless data commands to the other two neural recording/stimulation SoC chips.

The wireless data receiver uses a 5MHz ASK/OOK modulation capable of 100b/s datarates. Its primary purpose is to receive data commands and program the SoCs described in this thesis (from chapter-2 and chapter-4) while fully implanted inside an animal. The chip also has wireless power circuits, when connected to an off-chip coil
can provide 10mA of current generating 2.5V and 1.2V supply for the SoCs described in this thesis. Additional circuitry was added to provide eight different programmable DAC voltages for biasing. Lastly, the previously designed 915Mhz transmitter (chapter-2) and the UWB receiver (chapter-4) were also integrated on the same die.

### 6.2.3 Future In Vivo Testing

Currently both SoC chips from Chapter 2 and Chapter 4 are validated using a 6 inch by 8 inch PCB board as shown in the Appendix. The larger PCB board can be scaled down to include only an ultra-low power FPGA, biasing, the SoC chip and the new wireless power IC. The overall power dissipation should be below 10mW and the size of the PCB from a preliminary study would be 2cm by 1cm. This microsystem would allow for freely-moving implantable animal experiments. As well, utilizing all 64-channels through a state-of-the-art ECoG or micro electrode array would enable a more comprehensive study of phase synchrony in seizures.

The *in vivo* experiments in this thesis were performed on a population of five rodents. A study with a larger population of rats to benchmark the SoC in early seizure detection, seizure abortion and with different stimulation configurations should be performed. This should be performed with the miniaturized microsystem so that the rat
is moving more freely instead of with tethered wires. Furthermore, a study with non-human primates would provide a better model for epilepsy than with rats.
Appendix A

Supplementary Hardware and Software Documentation

A.1 Board Design

The 6-layer PCB that was used to test the chip was designed using the Altium Designer 6 software. The manufactured board with components is shown in Figure A.1. To minimize cost and simplify the testing process one PCB was shared for both SoCs. The PCB could be reconfigured via FPGA code and jumpers for different SoCs.

A few of the key components on the PCB are listed below:

- Voltage Regulator, 497-8717-1-ND: Regulator for on-chip 1.2V and 3.3V VDD supply.
- Oscillator Crystal, 40MHz, 3.3V XC1437CT-ND: Clock for FPGA.
- crystal, 14.3MHz, 1.8V, 478-4777-1-ND: Reference for on-chip PLL of SoC1.
- crystal, 12MHz, 3.3V XC1426CT-ND: Clock for USB chip.
- EEPROM 544-1379-5-ND: Required for FPGA programming.
A.3. Simulation Results

Simulation results for the analog front-end, signal processing, stimulator and RF TX are displayed in Figure A.2 and...
Figure A.2: Simulation results showing AFE and signal processing.
Figure A.3: Simulation results showing stimulator and RF TX.
A.3 FIR Coefficients for Neural Vector Analysis

The following filter is programmed on to the SoCs registers by the FPGA. The SoC has a coefficient register and a sign register.

Hilbert FIR Filter:
-0.238043547175795
-0.033481791464408
-0.037779442721307
-0.044661742098734
-0.056161703972340
-0.077485349187524
-0.127972540466038
-0.382186122086405
0.382186122086405
0.127972540466038
0.077485349187524
0.056161703972340
0.044661742098734
0.037779442721307
0.033481791464408
0.238043547175795

All-pass FIR Filter:
-0.127379674922053
-0.105961482039361
-0.071984490366035
-0.027307688214898
0.023026274641338
0.071551063197221
0.109985542903530
0.131242367405554
0.131242367405554
0.109985542903530
0.071551063197221
0.023026274641338
-0.027307688214898
-0.071984490366035
-0.105961482039361
-0.127379674922053
A.4 Phase Synchrony and Magnitude Processor: VERILOG RTL Code

The following code as used to generate the processor detailed in Chapter 2 and implemented on the SoC of Chapter 4.

```verilog
//*************************** BY: Karim Abdelhalim *****************************************
//TOP MODULE: Contains the pins for final chip and modules.
//inputs: rst, clock, opcode, number1, number2--> (2 operands)
//output: data, negative_trig_flag
//------------------------------------------------------------------------------
module top2(rst, clk, select_inputs, select_operation, number1, number2, number11, number22, number111, number222, number1111, number2222, sign1, sign2, sign11, sign22, sign111, sign222, sign1111, sign2222, final_out, negative_trig_flag_sine, negative_trig_flag_cosine);
input clk, rst;
input [2:0] select_inputs;
input [2:0] select_operation;
input [15:0] number1;
input [15:0] number2;
input [15:0] number11;
input [15:0] number22;
input [15:0] number111;
input [15:0] number222;
input [15:0] number1111;
input [15:0] number2222;
input sign1;
input sign2;
input sign11;
input sign22;
input sign111;
input sign222;
input sign1111;
input sign2222;
output [9:0] final_out;
output negative_trig_flag_sine;
output negative_trig_flag_cosine;
wire data_valid;
wire data_valid2;
wire data_small;
wire [2:0] select_operation;
wire [2:0] select_inputs;
wire [15:0] number1;
wire [15:0] number2;
wire [15:0] number11;
wire [15:0] number22;
wire [15:0] data1;
wire [15:0] data2;
wire [3:0] count;
wire [3:0] count2;
wire [3:0] count3;
```
wire [15:0] data11;
wire [15:0] data22;
wire [15:0] sine_out;
wire [15:0] cosine_out;
wire [19:0] x0;
wire [19:0] y0;
wire [19:0] z0;
wire [19:0] x02;
wire [19:0] y02;
wire [19:0] z02;
wire [19:0] x00;
wire [19:0] y00;
wire [19:0] z00;
wire [19:0] x;
wire [19:0] y;
wire [19:0] z;
wire [19:0] x2;
wire [19:0] y2;
wire [19:0] z2;
wire [19:0] xx;
wire [19:0] yy;
wire [19:0] zz;
wire [15:0] data_difference;
wire [9:0] final_out;
wire [15:0] mag1;
wire [15:0] mag2;
wire [15:0] phase1;
wire [15:0] phase2;
wire [1:0] quad1;
wire [1:0] quad2;
wire [1:0] quad_sin_cos;
wire ready;
wire ready2;
wire ready_small;
wire negative_trig_flag_sine;
wire negative_trig_flag_cosine;
wire [15:0] output_sum_sin;
wire [15:0] output_sum_cos;
wire [19:0] x0_small;
wire [19:0] x0_small;
wire [19:0] y0_small;
wire [19:0] y0_small;
wire [19:0] x_small;
wire [19:0] y_small;
wire [19:0] z_small;
wire [15:0] number1a;
wire [15:0] number2a;
wire [15:0] number11a;
wire [15:0] number22a;
wire [15:0] phase_synch;
wire sign1a;
wire sign2a;
wire sign11a;
wire sign22a;
wire data_valid_small;
CHAPTER A. SUPPLEMENTARY HARDWARE AND SOFTWARE DOCUMENTATION

front_end front_end_module(select_inputs, number1, number2, number11, number22, number111, number222, number1111, number2222, sign1, sign2, sign11, sign22, sign111, sign222, sign1111, sign2222, number1a, number2a, number11a, number22a, sign1a, sign2a, sign11a, sign22a);
controller_mag_phase cont_instance_mag(rst, clk, number1a, number2a, number11a, number22a, sign1a, sign2a, sign11a, sign22a, ready, x0, y0, x00, y00, quad1, quad2, data_valid);
cordic_core_mag_phase core_instance_mag(rst, clk, ready, x0, y0, z0, x00, y00, z00, x, y, z, xx, yy, zz, count);
output_data_mag_phase output_instance_mag(rst, clk, x, y, z, xx, yy, zz, quad1, quad2, data_valid, phase1, mag1, phase2, mag2, data_difference);
controller_sin_cos controller_instance_sin(rst, clk, {data_difference[14:0],1'b0}, ready2, x02, y02, z02, quad_sin_cos, data_valid2);
cordic_core_sin_cos core_instance_sin(rst, clk, ready2, x02, y02, z02, x2, y2, z2, count2);
output_data_sin_cos output_instance_sin(rst, clk, x2, y2, z2, quad_sin_cos, data_valid2, sine_out, cosine_out, negative_trig_flag_sine, negative_trig_flag_cosine);
output_FIR output_instance_FIR(rst, clk, data_valid, negative_trig_flag_sine, negative_trig_flag_cosine, sine_out, cosine_out, output_sum_sin, output_sum_cos);
controller_mag_phase_small cont_instance_mag_small(rst, clk, output_sum_sin, output_sum_cos, ready_small, x0_small, y0_small, z0_small, data_valid_small);
cordic_core_mag_phase_small core_instance_mag_small(rst, clk, ready_small, x0_small, y0_small, z0_small, x_small, y_small, z_small, count3);
output_data_mag_phase_small output_instance_mag_small(rst, clk, x_small, y_small, z_small, data_valid_small, phase1, mag1, phase2, mag2, phase_synch);
output_mux output_mux_instance(select_operation, phase1, mag1, phase2, mag2, sine_out, cosine_out, data_difference, phase_synch, final_out);
endmodule

//****************************BY: Karim Abdelhalim*******************************/
output [15:0] number1a;
output [15:0] number2a;
output sign1a;
output sign2a;
output sign11a;
output sign22a;
wire [2:0] select_inputs;
wire [15:0] number1;
wire [15:0] number2;
wire [15:0] number11;
wire [15:0] number22;
wire [15:0] number111;
wire [15:0] number222;
wire [15:0] number1111;
wire [15:0] number2222;
reg [15:0] number1a;
reg [15:0] number2a;
reg [15:0] number11a;
reg [15:0] number22a;
reg sign1a;
reg sign2a;
reg sign11a;
reg sign22a;

parameter ab=0, ac=1, ad=2, bc=3, bd=4, cd=5;

always @(select_inputs or number1 or number2 or number11 or number22 or number111 or number222 or number1111 or number2222 or sign1 or sign2 or sign11 or sign22 or sign111 or sign222 or sign1111 or sign2222)
begin
    case (select_inputs)
        ab:
        begin
            number1a=number1;
            number2a=number2;
            sign1a=sign1;
            sign2a=sign2;
            number11a=number11;
            number22a=number22;
            sign11a=sign11;
            sign22a=sign22;
        end
    ac:
    begin
        number1a=number1;
        number2a=number2;
        sign1a=sign1;
        sign2a=sign2;
        number11a=number11;
        number22a=number22;
        sign11a=sign11;
        sign22a=sign22;
    end
end
sign22a=sign222;
end

ad: begin
number1a=number1;
number2a=number2;
sign1a=sign1;
sign2a=sign2;

number11a=number1111;
number22a=number2222;
sign11a=sign1111;
sign22a=sign2222;
end

bc: begin
number1a=number11;
number2a=number22;
sign1a=sign11;
sign2a=sign22;

number11a=number1111;
number22a=number2222;
sign11a=sign1111;
sign22a=sign2222;
end

bd: begin
number1a=number11;
number2a=number22;
sign1a=sign11;
sign2a=sign22;

number11a=number1111;
number22a=number2222;
sign11a=sign1111;
sign22a=sign2222;
end

cd: begin
number1a=number111;
number2a=number222;
sign1a=sign111;
sign2a=sign222;

number11a=number1111;
number22a=number2222;
sign11a=sign1111;
sign22a=sign2222;
end

default: begin
number1a=number1;
number2a=number2;
sign1a=sign1;
sign2a=sign2;
number11a=number11;

number22a=number22;
sign11a=sign11;
sign22a=sign22;
endcase
end
endmodule

module controller_mag_phase(rst, clk, number1, number2, number11, number22, sign1, sign2, sign11, sign22, ready, x0, y0, z0, x00, y00, z00, quad1, quad2, data_valid);
input rst, clk;
input [15:0]number1;
input [15:0]number11;
input [15:0]number2;
input [15:0]number22;
input sign1, sign2, sign11, sign22;
output ready;
output [19:0] x0;
output [19:0] y0;
output [19:0] z0;
output [19:0] x00;
output [19:0] y00;
output [19:0] z00;
output [1:0]quad1;
output [1:0]quad2;
output data_valid;

// *********************************** Variables ********************************************
reg [4:0]count;
reg [4:0]new_count;
wire rst, clk;
wire sign1, sign2, sign11, sign22;
wire [15:0]number1;
wire [15:0]number2;
wire [15:0]number11;
wire [15:0]number22;
wire ready;
reg [19:0] x0;
reg [19:0] y0;
reg [19:0] z0;
reg [19:0] x00;
reg [19:0] y00;
reg [19:0] z00;
reg [3:0]state, next_state;
reg [1:0]quad1;
reg [1:0]quad2;

// *********************************** Parameter Declarations for states ********************************************
parameter off=0, read=1, preprocess=2, process=3, data_output=4, delay0=5, delay1=6, delay2=7, delay3=8; // names of states for FSM

// *********************************** Flip Flop circuitry ********************************************
always @ (posedge clk or negedge rst) 
begin 
  if (!rst) 
    state<=off; 
  else 
    state<=next_state; 
end 

always @ (posedge clk or negedge rst) 
begin 
  if (!rst) 
    count<=0; 
  else 
    count<=new_count; 
end 

// Next State Calculations
always @(state or count) 
begin 
  next_state=state; 
  case (state) 
    off: next_state=delay0; 
    delay0: next_state=delay1; 
    delay1: next_state=delay2; 
    delay2: next_state=delay3; 
    delay3: next_state=read; 
    read: next_state=preprocess; 
    preprocess: next_state=process; 
    process: 
      if (count==20) 
        next_state=data_output; 
      else 
        next_state=process; 
    default: next_state =off; 
  endcase 
end 

// Execute Instruction
always @(state or number1 or number2 or number11 or number22) 
begin 
  x0={2'b00,number1,2'b00}; 
  y0={2'b00,number2,2'b00}; 
  z0=0; 
  x00={2'b00,number11,2'b00}; 
  y00={2'b00,number22,2'b00}; 
  z00=0; 
end 

// Adjust angle and find quadrant
always @(sign1 or sign2 or sign11 or sign22 or state) 
// always @(state) 
begin 
  quad1<=2'b00; 
  quad2<=2'b00;
if \((\text{sign1}==0)\&\&(\text{sign2}==0))
quad1<=2'b00;
else if \((\text{sign1}==1)\&\&(\text{sign2}==0))
quad1<=2'b01;
else if \((\text{sign1}==1)\&\&(\text{sign2}==1))
quad1<=2'b10;
else if \((\text{sign1}==0)\&\&(\text{sign2}==1))
quad1<=2'b11;

if \((\text{sign11}==0)\&\&(\text{sign22}==0))
quad2<=2'b00;
else if \((\text{sign11}==1)\&\&(\text{sign22}==0))
quad2<=2'b01;
else if \((\text{sign11}==1)\&\&(\text{sign22}==1))
quad2<=2'b10;
else if \((\text{sign11}==0)\&\&(\text{sign22}==1))
quad2<=2'b11;
end

// **************************************** Count Calculations ****************************************
always @(count)
begin
  if (count==21)
    new_count=0;
  else new_count=count+1;
end

// **************************************** Some outputs ****************************************
assign ready=\!\!(state==process);
assign data_valid=(state==data_output);
endmodule

// ******************************************BY: Karim Abdelhalim******************************************

// Verilog HDL for "CORDIC2", "controller_mag_phase" "verilog"

module controller_mag_phase_small(rst, clk, number1, number2, ready, x0, y0, z0, data_valid);
input rst, clk;
input [15:0]number1;
input [15:0]number2;
output ready;
output [19:0] x0;
output [19:0] y0;
output [19:0] z0;
output data_valid;

// ******************************************Variables******************************************
reg [4:0]count;
reg [4:0]new_count;
wire rst, clk;
wire sign1, sign2;
wire [15:0]number1;
wire [15:0]number2;
wire ready;
reg [19:0] x0;
reg [19:0] y0;
reg [19:0] z0;
reg [3:0] state, next_state;

// *********************************** Parameter Declarations for states ***********************************
parameter off=0, read=1, preprocess=2, process=3, data_output=4, delay0=5, delay1=6, delay2=7, delay3=8;  //
// names of states for FSM

// *********************************** Flip Flop circuitry ****************************************************
always @(posedge clk or negedge rst)
begin
    if (!rst)
        state<=off;
    else
        state<=next_state;
end

always @(posedge clk or negedge rst)
begin
    if (!rst)
        count<=0;
    else
        count<=new_count;
end

// *************************************** Next State Calculations ****************************************
always @(state or count)
begin
    next_state=state;
    case(state)
        off: next_state=delay0;
        delay0: next_state=delay1;
        delay1: next_state=delay2;
        delay2: next_state=delay3;
        delay3: next_state=read;
        read: next_state=preprocess;
        preprocess: next_state=process;
        process:
            if (count==20)
                next_state=data_output;
            else
                next_state=process;
        data_output: next_state=delay0;
        default: next_state =off;
    endcase
end

// ************************************** Execute Instruction **********************************************
always @(state or number1 or number2)
begin
    x0={2'b00, number1, 2'b00};
    y0={2'b00, number2, 2'b00};
    z0=0;
end
always @(count)
begin
  if(count==21)
    new_count=0;
  else
    new_count=count+1;
end

assign ready=˜(state==process);
assign data_valid=(state==data_output);
endmodule

// *********************************** Variables ********************************************
reg [4:0]count;
reg [4:0]new_count;
reg [15:0]pre_angle;
wire [19:0]angle;
wire rst, clk;
wire sign1, sign2;
wire [15:0]number1;
wire ready;
reg [19:0] x0;
reg [19:0] y0;
reg [19:0] z0;
reg [3:0]state, next_state;
reg [1:0]quad;

// *********************************** Parameter Declarations for states ********************************************
parameter off=0, read=1, preprocess=2, process=3, data_output=4, delay0=5, delay1=6, delay2=7, delay3=8; // names of states for FSM

always @(posedge clk or negedge rst)
begin
  if (~rst)
    state<='off;
  else
    state<='next_state;
end

always @(posedge clk or negedge rst)
begin
if (`rst)
  count<=0;
else count<=new_count;
end

//*********************************************Next State Calculations*********************************************
always @ (state or count)
begin
  next_state = state;
  case (state)
  off: next_state = delay0;
  delay0: next_state = delay1;
  delay1: next_state = delay2;
  delay2: next_state = delay3;
  delay3: next_state = read;
  read: next_state = preprocess;
  preprocess: next_state = process;
  process:
    if (count==20)
      next_state = data_output;
    else
      next_state = process;
  data_output: next_state = delay0;
  default: next_state = off;
  endcase
end

//***********************************************ExecuteInstruction***********************************************
always @ (state or angle or number1)
begin
  x0=20'b0000_1001_1011_0111_0100;
  y0=0;
  z0=angle;
end

//******************************************Adjust angle and find quadrant******************************************
always @(number1 or state)
begin
  pre_angle<=0;
  quad<=0;
  begin
    if (number1>16'b100_1011_0110_0101_1) //quad 4
      begin
        pre_angle<=16'b110_0100_1000_0111_1-number1; //2pi-angle
        quad<=2'b11;
      end
    else if (number1>16'b011_0010_0100_0011_1) //quad 3
      begin
        pre_angle<=number1-16'b011_0010_0100_0011_1; //angle-pi
        quad<=2'b10;
      end
    else if (number1>16'b001_1001_0010_0001_1) //quad 2
      begin
        pre_angle<=16'b011_0010_0100_0011_1-number1; //pi-angle
        quad<=2'b01;
      end
  end
end
CHAPTER A. SUPPLEMENTARY HARDWARE AND SOFTWARE DOCUMENTATION

end
else
begin
pre_angle<=number1;
quad<=2'b00;
end
end

// *************************************** Count Calculations **********************************************
always @(count)
begin
if (count==21)
new_count=0;
else new_count=count+1;
end

// *************************************** Some outputs **********************************************
assign angle={1'b0, pre_angle,3'b000};
assign ready=(state==process);
assign data_valid=(state==data_output);
endmodule

//*************************** BY: Karim Abdelhalim*****************************************
//Verilog HDL for "CORDIC", "cordic_core_mag_phase" "verilog"

module cordic_core_mag_phase(rst, clk, ready, x0, y0, z0, x00, y00, z00, x, y, z, xx, yy, zz, count);
input rst, clk, ready;
input [19:0]x0;
input [19:0]y0;
input [19:0]z0;
input [19:0]x00;
input [19:0]y00;
input [19:0]z00;
output [19:0] x;
output [19:0] y;
output [19:0] z;
output [19:0] xx;
output [19:0] yy;
output [19:0] zz;
output [3:0] count;

// *********************************** Variables ********************************************
wire [19:0]x0;
wire [19:0]y0;
wire [19:0]x0;
wire [19:0]x0;
wire [19:0]y0;
wire [19:0]x0;
wire [19:0]x0;
wire [19:0]x0;
wire [19:0]x0;
wirerst, clk;
wire [19:0]arctan[15:0];
reg [19:0]n;
reg [19:0]x;
reg [19:0]y;
reg [19:0]z;
reg [19:0]xx;

```verilog
reg [19:0] yy;
reg [19:0] zz;
reg [19:0] x_new;
reg [19:0] y_new;
reg [19:0] z_new;
reg [19:0] x_newb;
reg [19:0] y_newb;
reg [19:0] z_newb;
reg [19:0] temp1, temp2, temp3, temp4;
reg [19:0] temp11, temp22, temp33, temp44;
reg [3:0] count;
reg [3:0] new_count;

// ******************************** Sequential ***************************************
always @ (posedge clk or negedge rst)
begin
  if (!rst)
    x<=0;
    y<=0;
    z<=0;
    xx<=0;
    yy<=0;
    zz<=0;
  else if (ready)
    x<=x0;
    y<=y0;
    z<=z0;
    xx<=x00;
    yy<=y00;
    zz<=z00;
  else
    x<=x_new;
    y<=y_new;
    z<=z_new;
    xx<=x_newb;
    yy<=y_newb;
    zz<=z_newb;
end

always @ (posedge clk or negedge rst)
begin
  if (!rst)
    count<=0;
  else
    count<=new_count;
end

// ******************************** Combinational **************************************
always @(count or ready)
begin
  if ('rst)
    count<=0;
  else
    count<=new_count;
end
```
begin
  if (ready==0) begin
    if (count==15)
      new_count<=0;
    else new_count<=count+1;
  end
  else new_count<=15;
end

// ******************************** Combinational Cordic Core **************************************
always @({x or y or z or xx or yy or zz or n or temp1 or temp2 or temp3 or temp4 or temp11 or temp22
           or temp33 or temp44})
begin
  if (y[19]==1) begin
    if (x[19]==1) begin
      x_new<=˜((˜x)-temp4);
      y_new<=˜((˜y)+temp3);
    end
    else if (x[19]==0) begin
      x_new=x+temp4;
      y_new<=˜((˜y)-temp1);
    end
    if (z[19]==0) begin
      z_new<=z-n;
    end
    else begin
      z_new<=˜((˜z)+n);
    end
  end
  else if (y[19]==0) begin
    if (x[19]==1) begin
      x_new<=˜((˜x)-temp2);
      y_new<y+temp3;
    end
    else if (x[19]==0) begin
      x_new=x+temp2;
      y_new<y-temp1;
    end
    if (z[19]==0) begin
      z_new=z+n;
    end
    else begin
      z_new<=z+n;
    end
  end
// ******************************** Sine, cosine, multiply core********************
always @(x or y or count or xx or yy or temp1 or temp2 or temp3 or temp4 or temp5 or temp6 or temp7 or temp8 or temp9 or temp10 or temp11 or temp12 or temp13 or temp14 or temp15 or temp16 or temp17 or temp18 or temp19 or temp20 or temp21 or temp22 or temp23 or temp24 or temp25 or temp26 or temp27 or temp28 or temp29 or temp30 or temp31 or temp32 or temp33 or temp34 or temp35 or temp36 or temp37 or temp38 or temp39 or temp40 or temp41 or temp42 or temp43 or temp44)
begin
  temp1<=temp1;
  temp2<=temp2;
end
temp3<=temp3;
temp4<=temp4;

temp11<=temp11;
temp22<=temp22;
temp33<=temp33;
temp44<=temp44;
if (count==15)
begin
  temp1<=0;
temp2<=0;
temp3<=0;
temp4<=0;
temp11<=0;
temp22<=0;
temp33<=0;
temp44<=0;
end
else
begin
  temp1<x>>(count);
temp2<y>>(count);
temp3<(-'x)>>[count]); //approximation maybe add one
temp4<(-'y)>>[count]); //approximation maybe add one

  temp11=x>>(count);
temp22=y>>(count);
temp33=([-xx)>>[count]); //approximation maybe add one
temp44=([-yy)>>[count]); //approximation maybe add one
end

// *********************************** ARCTAN DATA ******************************************
assign arctan[0]=20'b0000_1100_1001_0000_1111;
assign arctan[1]=20'b0000_0111_0110_1011_0001;
assign arctan[2]=20'b0000_0011_1110_1011_0110;
assign arctan[3]=20'b0000_0001_1111_1101_0101;
assign arctan[4]=20'b0000_0000_1111_1111_1010;
assign arctan[5]=20'b0000_0000_0111_1111_1111;
assign arctan[6]=20'b0000_0000_0011_1111_1111;
assign arctan[7]=20'b0000_0000_0001_1111_1111;
assign arctan[8]=20'b0000_0000_0000_1111_1111;
assign arctan[9]=20'b0000_0000_0000_0111_1111;
assign arctan[10]=20'b0000_0000_0000_0011_1111;
assign arctan[11]=20'b0000_0000_0000_0001_1111;
assign arctan[12]=20'b0000_0000_0000_0000_1111;
assign arctan[13]=20'b0000_0000_0000_0000_0111;
assign arctan[14]=20'b0000_0000_0000_0000_0011;
assign arctan[15]=20'b0000_0000_0000_0000_0001;

// *********************************** Outputs Selected Data **********************************
always @(count)
begin
  n<=arctan[count];
module cordic_core_mag_phase_small(rst, clk, ready, x0, y0, z0, x, y, z, count);
input rst, clk, ready;
input [19:0] x0;
input [19:0] y0;
input [19:0] z0;
output [19:0] x;
output [19:0] y;
output [19:0] z;
output [3:0] count;

// *********************************** Variables ********************************************
wire [19:0] x0;
wire [19:0] y0;
wire [19:0] z0;
wire rst, clk;
wire[19:0] arctan[15:0];
reg [19:0] n;
reg [19:0] x;
reg [19:0] y;
reg [19:0] z;
reg [19:0] x_new;
reg [19:0] y_new;
reg [19:0] z_new;
reg [19:0] temp1,temp2, temp3, temp4;
reg [3:0] count;
reg [3:0] new_count;

// ******************************** Sequential ***********************************************
always @([posedge clk or negedge rst])
begina
  if (~rst) begin
    x<=0;
y<=0;
z<=0;
  end
  else if(ready) begin
    x<=x0;
y<=y0;
z<=z0;
  end
  else begin
    x<=x_new;
y<=y_new;
z<=z_new;
  end
end
endmodule
always @(posedge clk or negedge rst)
begin
  if ('rst)
    count<=0;
  else
    count<=new_count;
end

// ******************************** Combinational ***************************************
always @(count or ready)
begin
  if (ready==0)
    begin
      if (count==15)
        new_count<=0;
      else
        new_count<=count+1;
    end
  else
     new_count<=15;
end

// ******************************** Combinational Cordic Core ***************************************
always @(x or y or z or n or temp1 or temp2 or temp3 or temp4)
begin
  if (y[19]==1)
    begin
      if (x[19]==1)
        begin
          x_new<=˜((˜x)-temp4);
          y_new<=˜((˜y)+temp3);
        end
      else //x[19]==0
        begin
          x_new=x+temp4;
          y_new<=˜((˜y)-temp1);
        end
    end
  else if (y[19]==0)
    begin
      if (z[19]==0)
        begin
          z_new=z-n;
        end
      else
        begin
          z_new<=˜((˜z)+n);
        end
    end
  end
end
x_new<=x+temp2;
y_new<=y-temp1;
end
if (z[19]==0)
begin
z_new<=z+n;
end
else
begin
z_new<=˜((˜z)-n);
end
end

// ******************************** Sine, cosine, multiply core ********************
always @(x or y or count or temp1 or temp2 or temp3 or temp4)
begin
    temp1<=temp1;
temp2<=temp2;
temp3<=temp3;
temp4<=temp4;

    if (count==15)
begin
        temp1<=0;
temp2<=0;
temp3<=0;
temp4<=0;
end
else
begin
    temp1<=x>>(count);
temp2<=y>>(count);
temp3<=((˜x)>>(count)); //approximation maybe add one
temp4<=((˜y)>>(count)); //approximation maybe add one
end
end

// *********************************** ARCTAN DATA ******************************************
assign arctan[0]=20'b0000_1100_1001_0000_1111;
assign arctan[1]=20'b0000_0111_0110_1011_0001;
assign arctan[2]=20'b0000_0011_1110_1011_0110;
assign arctan[3]=20'b0000_0001_1111_1101_0101;
assign arctan[4]=20'b0000_0000_1111_1111_1010;
assign arctan[5]=20'b0000_0000_0111_1111_1111;
assign arctan[6]=20'b0000_0000_0011_1111_1111;
assign arctan[7]=20'b0000_0000_0001_1111_1111;
assign arctan[8]=20'b0000_0000_0000_1111_1111;
assign arctan[9]=20'b0000_0000_0000_0111_1111;
assign arctan[10]=20'b0000_0000_0000_0011_1111;
assign arctan[11]=20'b0000_0000_0000_0001_1111;
assign arctan[12]=20'b0000_0000_0000_0000_1111;
assign arctan[13]=20'b0000_0000_0000_0000_0111;
assign arctan[14]=20'b0000_0000_0000_0000_0011;
```verilog
module cordic_core_sin_cos(rst, clk, ready, x0, y0, z0, x, y, z, count);
  input rst, clk, ready;
  input [19:0] x0;
  input [19:0] y0;
  input [19:0] z0;
  output [19:0] x;
  output [19:0] y;
  output [19:0] z;
  output [3:0] count;

  // *********************************** Variables ********************************************
  wire [19:0] x0;
  wire [19:0] y0;
  wire [19:0] z0;
  wire [19:0] arctan[15:0];
  wire rst, clk;
  reg [19:0] n;
  reg [19:0] x;
  reg [19:0] y;
  reg [19:0] z;
  reg [19:0] x_new;
  reg [19:0] y_new;
  reg [19:0] z_new;
  reg [19:0] temp1, temp2, temp3, temp4;
  reg [3:0] count;
  reg [3:0] new_count;

  // ******************************** Sequential **************************************
  always @ (posedge clk or negedge rst)
  begin
    if (!rst)
      begin
        x<='0;
        y<='0;
        z<='0;
      end
    else if (ready)
      begin
        x<='x0;
        y<='y0;
        z<='z0;
      end
  end
endmodule
```

```verilog
assign arctan[15]=20'b0000_0000_0000_0000_0001;

//**************************************************************************** Outputs Selected Data************************************************************************
always @(count)
begin
  n<='arctan[count];
end
endmodule
```

//*****************************************BY: Karim Abdelhalim*****************************************
module cordic_core_sin_cos(rst, clk, ready, x0, y0, z0, x, y, z, count);
  input rst, clk, ready;
  input [19:0] x0;
  input [19:0] y0;
  input [19:0] z0;
  output [19:0] x;
  output [19:0] y;
  output [19:0] z;
  output [3:0] count;

  // *********************************** Variables ********************************************
  wire [19:0] x0;
  wire [19:0] y0;
  wire [19:0] z0;
  wire [19:0] arctan[15:0];
  wire rst, clk;
  reg [19:0] n;
  reg [19:0] x;
  reg [19:0] y;
  reg [19:0] z;
  reg [19:0] x_new;
  reg [19:0] y_new;
  reg [19:0] z_new;
  reg [19:0] temp1, temp2, temp3, temp4;
  reg [3:0] count;
  reg [3:0] new_count;

  // ******************************** Sequential **************************************
  always @ (posedge clk or negedge rst)
  begin
    if (!rst)
      begin
        x<='0;
        y<='0;
        z<='0;
      end
    else if (ready)
      begin
        x<='x0;
        y<='y0;
        z<='z0;
      end
  end
endmodule
else begin
  x<=x_new;
  y<=y_new;
  z<=z_new;
end

always @(posedge clk or negedge rst)
begin
  if (~rst)
    count<=0;
  else
    count<=new_count;
end

always @(count or ready)
begin
  if (~ready==0)
    begin
      if (count==15)
        new_count<=0;
      else
        new_count<=count+1;
    end
  else
    new_count<=15;
end

always @(x or y or z or n or temp1 or temp2 or temp3 or temp4)
begin
  if (z[19]==1'b0)
    begin
      if (x[19]==1'b1)
        begin
          if (y[19]==1'b0)
            begin
              x_new<=˜((˜x)+temp2);  //−x, +y
              y_new<=y-temp3;
            end
          else
            begin
              x_new<=˜((˜x)-temp4);  //−x, −y
              y_new<=˜((˜y)+temp3);
            end
        end
    else if (x[19]==1'b0)
      begin
        if (y[19]==1'b0)
          begin
            if (x[19]==1'b0)
              begin
                x_new<=˜((˜x)+temp2);  //−x, +y
                y_new<=y-temp3;
              end
            else
              begin
                x_new<=˜((˜x)-temp4);  //−x, −y
                y_new<=˜((˜y)+temp3);
              end
          end
      else if (x[19]==1'b0)
        begin
          if (y[19]==1'b0)
            begin
              x_new<=x-temp2;  //+x, +y
              y_new<=y+temp1;
            end
else
begin
   x_new<=x+temp4;  //+x, -y
   y_new<=(~y)-temp1);
end
end
z_new<=z-n;
end
else if(z[19]==1'b1)
begin
   if(x[19]==1'b1)
   begin
      if(y[19]==1'b0)
      begin
         x_new=(~x)-temp2);  //-x, +y
         y_new=y+temp3);
      end
      else
      begin
         x_new=(~x)+temp4);  //-x, +y
         y_new=(~y)-temp3);
      end
   end
   else if (x[19]==1'b0)
   begin
      if(y[19]==1'b0)
      begin
         x_new=x+temp2;  //+x, +y
         y_new=y-temp1;
      end
      else
      begin
         x_new=x-temp4;  //+x, -y
         y_new=(~y)+temp1);
      end
    end
  end
  z_new=(~z)-n);
end

//******************************************************************************
// Sine, cosine core******************************************************************************
always @(count or x or y or temp1 or temp2 or temp3 or temp4)
begin
   temp1<=temp1;
   temp2<=temp2;
   temp3<=temp3;
   temp4<=temp4;
   if (count==15)
   begin
      temp1<=0;
      temp2<=0;
      temp3<=0;
      temp4<=0;
end
end
else
begin
temp1<x>>(count);
temp2<y>>(count);
temp3<<('x)>>{(count)}; //approximation maybe add one
temp4<<('y)>>{(count)}; //approximation maybe add one
end
end

// **************************************** ARCTAN DATA **********************************************
assign arctan[0]=20'b0000_1100_1001_0000_1111;
assign arctan[1]=20'b0000_0111_1110_1011_0110;
assign arctan[2]=20'b0000_0000_1111_1110_1011;
assign arctan[3]=20'b0000_0000_1111_1110_1101;
assign arctan[4]=20'b0000_0000_0111_1111_1111;
assign arctan[5]=20'b0000_0000_0011_1111_1111;
assign arctan[6]=20'b0000_0000_0001_1111_1111;
assign arctan[7]=20'b0000_0000_0000_1111_1111;
assign arctan[8]=20'b0000_0000_0000_0111_1111;
assign arctan[9]=20'b0000_0000_0000_0011_1111;
assign arctan[10]=20'b0000_0000_0000_0001_1111;
assign arctan[11]=20'b0000_0000_0000_0000_1111;
assign arctan[12]=20'b0000_0000_0000_0000_0111;
assign arctan[13]=20'b0000_0000_0000_0000_0011;
assign arctan[14]=20'b0000_0000_0000_0000_0001;
assign arctan[15]=20'b0000_0000_0000_0000_0000;

// **************************************** Outputs Selected Data **********************************************
always @(count)
begin
n<=arctan[count];
end
endmodule

// ****************************************** BY: Karim Abdelhalim **********************************************
module output_data_mag_phase(rst, clk, x, y, z, xx, yy, zz, quad1, quad2, data_valid, data1, data2, data11, data22, data_difference);
input rst, clk;
input [19:0]x;
input [19:0]y;
input [19:0]z;
input [19:0]xx;
input [19:0]yy;
input [19:0]zz;
input [1:0]quad1;
input [1:0]quad2;
input data_valid;
output [15:0]data1;
output [15:0]data2;
output [15:0]data11;
output [15:0]data22;
output [15:0]data_difference;
// ****************************************** Variables **********************************************
wire[19:0]x;
wire[19:0] y;
wire [19:0] z;
wire[19:0] xx;
wire[19:0] yy;
wire [19:0] zz;
wire [1:0] quad1;
wire [1:0] quad2;
wire data_valid;
reg [15:0] data2;
reg [15:0] data1;
reg [15:0] new_data2;
reg [15:0] new_data1;
reg [15:0] data22;
reg [15:0] data11;
reg [15:0] new_data22;
reg [15:0] new_data11;
reg [15:0] data_difference;

// *********************************** Sequential ********************************************
// *********************************** Sequential ********************************************
always @ (posedge clk or negedge rst)
begin
if (~rst)
begin
  data1<=0;
  data2<=0;
  data11<=0;
  data22<=0;
  data_difference<=0;
end
else if (data_valid)
begin
  data1<=new_data1;
  data2<=new_data2;
  data11<=new_data11;
  data22<=new_data22;
  if (((new_data11-new_data1)&(16'b1000_0000_0000_0000))==0)
  data_difference<=new_data11-new_data1;
  else
  data_difference<=new_data11-new_data1+16'b0110_0100_1000_0111;
else
  data1<=data1;
  data2<=data2;
  data11<=data11;
  data22<=data22;
  data_difference<=data_difference;
end
end

// *********************************** Determining Output ********************************************
always@(x or z or xx or zz or quad1 or quad2)
begin
  if (quad1==2'b00)
begin
    new_data1<=z[19:4];
    new_data2<=x[17:2];
end
else if (quad1==2'b01)
    begin
        new_data1<=16'b0011_0010_0100_0011-z[19:4];
        new_data2<=x[17:2];
    end
else if (quad1==2'b10)
    begin
        new_data1<=z[19:4]+16'b0011_0010_0100_0011;
        new_data2<=x[17:2];
    end
else if (quad1==2'b11)
    begin
        new_data1<=16'b0110_0100_1000_0111-z[19:4];
        new_data2<=x[17:2];
    end
else
    begin
        new_data1<=z[19:4];
        new_data2<=x[17:2];
    end
if (quad2==2'b00)
    begin
        new_data11<=zz[19:4];
        new_data22<=xx[17:2];
    end
else if (quad2==2'b01)
    begin
        new_data11<=16'b0011_0010_0100_0011-zz[19:4];
        new_data22<=xx[17:2];
    end
else if (quad2==2'b10)
    begin
        new_data22<=xx[17:2];
    end
else if (quad2==2'b11)
    begin
        new_data11<=16'b0110_0100_1000_0111-zz[19:4];
        new_data22<=xx[17:2];
    end
else
    begin
        new_data11<=zz[19:4];
        new_data22<=xx[17:2];
    end
```verilog
module output_data_mag_phase_small(rst, clk, x, y, z, data_valid, data2);
  input rst, clk;
  input [19:0] x;
  input [19:0] y;
  input [19:0] z;
  input data_valid;
  output [15:0] data2;

  // *********************************** Variables ********************************************
  wire [19:0] x;
  wire [19:0] y;
  wire [19:0] z;
  wire data_valid;
  reg [15:0] data2;
  reg [15:0] new_data2;

  // *********************************** Sequential ********************************************
  always @ (posedge clk or negedge rst)
  begin
    if (!rst)
      data2 <= 0;
    else if (data_valid)
      data2 <= new_data2;
    else
      data2 <= data2;
  end

  // *********************************** Determining Output ***********************************
  always @(x or y or z)
  begin
    new_data2 <= x[17:2];
  end
endmodule
```

-----

```verilog
module output_data_mag_phase(rst, clk, x, y, z, data_valid, data, negative_trig_flag1, negative_trig_flag2);
  input rst, clk;
  input [19:0] x;
  input [19:0] y;
  input [19:0] z;
  input [1:0] quad;
  input data_valid;
  output [15:0] data;
  output [1:0] negative_trig_flag;

  // *********************************** Variables ********************************************
  wire [19:0] x;
  wire [19:0] y;
  wire [19:0] z;
  wire data_valid;
  reg [15:0] data;
  reg [1:0] new_negative_trig_flag;

  // *********************************** Sequential ********************************************
  always @ (posedge clk or negedge rst)
  begin
    if (!rst)
      data <= 0;
    else if (data_valid)
      data <= new_data;
    else
      data <= data;
  end

  // *********************************** Determining Output ***********************************
  always @(x or y or z)
  begin
    new_data <= x[17:2];
  end
endmodule
```

-----

```verilog
module output_data_sin_cos(rst, clk, x, y, z, quad, data_valid, data, data2, negative_trig_flag1, negative_trig_flag2);
  input rst, clk;
  input [19:0] x;
  input [19:0] y;
  input [19:0] z;
  input [1:0] quad;
  input data_valid;
  output [15:0] data;
  output [1:0] negative_trig_flag;

  // *********************************** Variables ********************************************
  wire [19:0] x;
  wire [19:0] y;
  wire [19:0] z;
  wire data_valid;
  reg [15:0] data;
  reg [1:0] new_negative_trig_flag;

  // *********************************** Sequential ********************************************
  always @ (posedge clk or negedge rst)
  begin
    if (!rst)
      data <= 0;
    else if (data_valid)
      data <= new_data;
    else
      data <= data;
  end

  // *********************************** Determining Output ***********************************
  always @(x or y or z)
  begin
    new_data <= x[17:2];
  end
endmodule
```
 CHAPTER A. SUPPLEMENTARY HARDWARE AND SOFTWARE DOCUMENTATION

```
output [15:0] data;
output [15:0] data2;
output negative_trig_flag1;
output negative_trig_flag2;

// *********************************** Variables ********************************************
wire [19:0] x;
wire [19:0] y;
wire [19:0] z;
wire [1:0] quad;
wire data_valid;
reg [15:0] data2;
reg [15:0] data;
reg [15:0] new_data2;
reg [15:0] new_data;
wire negative_trig_flag;
wire negative_trig_flag2;

// *********************************** Sequential *******************************************
always @ (posedge clk or negedge rst)
begin
  if (!rst)
    begin
      data<=0;
data2<=0;
    end
  else if (data_valid)
    begin
      data<=new_data;
data2<=new_data2;
    end
  else
    begin
      data<=data;
data2<=data2;
    end
end

// *********************************** Determining Output ***********************************
always @ (x or y or z or quad)
begin
  new_data<=y[15:0];
  new_data2<=x[15:0];
end
assign negative_trig_flag1=((quad==2’b01)|(quad==2’b11));
assign negative_trig_flag2=((quad==2’b01)|(quad==2’b10));
```

//*************************** BY: Karim Abdelhalim*****************************************
module output_FIR(rst, clk, data_valid, neg_sin, neg_cos, sin_in, cos_in, output_sum_sin, output_sum_cos);
input rst, clk;
input data_valid;
input neg_sin;
input neg_cos;
```
input [15:0] sin_in;
input [15:0] cos_in;
output [15:0] output_sum_sin;
output [15:0] output_sum_cos;
wire neg_sin;
wire neg_cos;
reg [23:0] x_nm31;
reg [23:0] x_nm30;
reg [23:0] x_nm29;
reg [23:0] x_nm28;
reg [23:0] x_nm27;
reg [23:0] x_nm26;
reg [23:0] x_nm25;
reg [23:0] x_nm24;
reg [23:0] x_nm23;
reg [23:0] x_nm22;
reg [23:0] x_nm21;
reg [23:0] x_nm20;
reg [23:0] x_nm19;
reg [23:0] x_nm18;
reg [23:0] x_nm17;
reg [23:0] x_nm16;
reg [23:0] x_nm15;
reg [23:0] x_nm14;
reg [23:0] x_nm13;
reg [23:0] x_nm12;
reg [23:0] x_nm11;
reg [23:0] x_nm10;
reg [23:0] x_nm9;
reg [23:0] x_nm8;
reg [23:0] x_nm7;
reg [23:0] x_nm6;
reg [23:0] x_nm5;
reg [23:0] x_nm4;
reg [23:0] x_nm3;
reg [23:0] x_nm2;
reg [23:0] x_nm1;
reg [23:0] x_nm;
reg [23:0] x2_nm31;
reg [23:0] x2_nm30;
reg [23:0] x2_nm29;
reg [23:0] x2_nm28;
reg [23:0] x2_nm27;
reg [23:0] x2_nm26;
reg [23:0] x2_nm25;
reg [23:0] x2_nm24;
reg [23:0] x2_nm23;
reg [23:0] x2_nm22;
reg [23:0] x2_nm21;
reg [23:0] x2_nm20;
reg [23:0] x2_nm19;
reg [23:0] x2_nm18;
reg [23:0] x2_nm17;
reg [23:0] x_nm16;
reg [23:0] x_nm15;
reg [23:0] x_nm14;
reg [23:0] x_nm13;
reg [23:0] x_nm12;
reg [23:0] x_nm11;
reg [23:0] x_nm10;
reg [23:0] x_nm9;
reg [23:0] x_nm8;
reg [23:0] x_nm7;
reg [23:0] x_nm6;
reg [23:0] x_nm5;
reg [23:0] x_nm4;
reg [23:0] x_nm3;
reg [23:0] x_nm2;
reg [23:0] x_nm1;
reg [23:0] x_nm;
reg [23:0] new_sample1;
reg [23:0] new_sample2;
reg [23:0] sum_sin;
reg [23:0] sum_cos;
reg [15:0] output_sum_sin;
reg [15:0] output_sum_cos;

always @(/posedge clk or negedge rst)
begin
  if ('rst)
  begin
    sum_sin<=0;
    sum_cos<=0;
  end
  else if(data_valid)
  begin
    sum_sin<=x_nm+x_nm1+x_nm2+x_nm3+x_nm4+x_nm5+x_nm6+x_nm7+
                 x_nm8+x_nm9+x_nm10+x_nm11+x_nm12+x_nm13+x_nm14+x_nm15+
                 x_nm16+x_nm17+x_nm18+x_nm19+x_nm20+x_nm21+x_nm22+x_nm23+
                 x_nm24+x_nm25+x_nm26+x_nm27+x_nm28+x_nm29+x_nm30+x_nm31;
    sum_cos<=x2_nm+x2_nm1+x2_nm2+x2_nm3+x2_nm4+x2_nm5+x2_nm6+x2_nm7+
                 x2_nm8+x2_nm9+x2_nm10+x2_nm11+x2_nm12+x2_nm13+x2_nm14+x2_nm15+
                 x2_nm16+x2_nm17+x2_nm18+x2_nm19+x2_nm20+x2_nm21+x2_nm22+x2_nm23+
                 x2_nm24+x2_nm25+x2_nm26+x2_nm27+x2_nm28+x2_nm29+x2_nm30+x2_nm31;
  end
end
//*********************************** Determining Output***********************************
always@ (sin_in or cos_in or neg_sin or neg_cos)
begin
    if (neg_sin==1)
        begin
            new_sample1={7'b1111111,˜sin_in,1'b1};
        end
    else if (neg_sin==0)
        begin
            new_sample1={7'b0000000,sin_in,1'b0};
        end
    if (neg_cos==1)
        begin
            new_sample2={7'b1111111,˜cos_in,1'b1};
        end
    else if (neg_cos==0)
        begin
            new_sample2={7'b0000000,cos_in,1'b0};
        end
end
always@ (sum_sin or sum_cos)
begin
    if (((sum_sin)&(24'b1000_0000_0000_0000_0000_0000))==0)
        output_sum_sin=sum_sin[23:8];
    else
        output_sum_sin=˜sum_sin[23:8];
    if (((sum_cos)&(24'b1000_0000_0000_0000_0000_0000))==0)
        output_sum_cos=sum_cos[23:8];
    else
        output_sum_cos=˜sum_cos[23:8];
end
endmodule

// Verilog HDL for "CORDIC2", "output_mux" *verilog*

//**********************************************************BY: Karim Abdelhalim**********************************************************
module output_mux(select, phase1, mag1, phase2, mag2, sine_out, cosine_out, data_difference, phase_synch, final_out);
input [2:0] select;
input [15:0] phase1;
input [15:0] mag1;
input [15:0] phase2;
input [15:0] mag2;
input [15:0] sine_out;
input [15:0] cosine_out;
input [15:0] data_difference;
input [15:0] phase_synch;
output [9:0] final_out;
wire [2:0]select;
wire [15:0]phase1;
wire [15:0]mag1;
wire [15:0]phase2;
wire [15:0]mag2;
wire [15:0]sine_out;
wire [15:0]cosine_out;
wire [15:0]data_difference;
wire [15:0]phase_synch;
reg [9:0]final_out;

// *********************************** Parameter Declaratoins for states***********************************
parameter sin=0, cos=1, angle1=2, angle2=3, magn1=4, magn2=5, phase_diff=6, synch=7; //names of states for FSM

//Flip Flop circuitry*********************************************************

always @(select or sine_out or cosine_out or phase1 or phase2 or mag1 or mag2 or data_difference or phase_synch or sine_out) begin
  case(select)
    sin: final_out=sine_out[15:6];
    cos: final_out=cosine_out[15:6];
    angle1: final_out=phase1[15:6];
    angle2: final_out=phase2[15:6];
    magn1: final_out=mag1[15:6];
    magn2: final_out=mag2[15:6];
    phase_diff: final_out=data_difference[15:6];
    synch: final_out=phase_synch[15:6];
    default: final_out=sine_out[15:6];
  endcase
end
endmodule
References


