

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# CMOS VLSI Correlator Design for Radio-Astronomical Signal Processing

A thesis presented in partial fulfilment of the requirements for the  
degree of

Doctor of Philosophy  
in  
Engineering

at Massey University, Auckland, New Zealand

Stepan Lapshev

2018

## Abstract

Multi-element radio telescopes employ methods of indirect imaging to capture the image of the sky. These methods are in contrast to direct imaging methods whereby the image is constructed from sensor measurements directly and involve extensive signal processing on antenna signals. The Square Kilometre Array, or the SKA, is a future radio telescope of this type that, once built, will become the largest telescope in the world. The unprecedented scale of the SKA requires novel solutions to be developed for its signal processing pipeline one of the most resource-consuming parts of which is the correlator. The SKA uses the FX correlator construction that consists of two parts: the F part that translates antenna signals into frequency domain and the X part that cross-correlates these signals between each other. This research focuses on the integrated circuit design and VLSI implementation issues of the X part of a very large FX correlator in 28 nm and 130 nm CMOS. The correlator's main processing operation is the complex multiply-accumulation (CMAC) for which custom 28 nm CMAC designs are presented and evaluated. Performance of various memories inside the correlator also affects overall efficiency, and input-buffered and output-buffered approaches are considered with the goal of improving upon it. For output-buffered designs, custom memory control circuits have been designed and prototyped in 130 nm that improve upon eDRAM by taking advantage of sequential access patterns. For the input-buffered architecture, a new scheme is proposed that decreases the usage of the input-buffer memory by a third by making use of multiple accumulators in every CMAC. Because cross-correlation is a very data-intensive process, high-performance SerDes I/O is essential to any practical ASIC implementation. On the I/O design, the 28 nm full-rate transmitter delivering 15 Gbps per lane is presented. This design consists of the scrambler, the serialiser, the digital VCO with analog fine-tuning and the SST driver including features of a 4-tap FFE, impedance tuning and amplitude tuning.

## **Acknowledgements**


I would like to acknowledge my supervisor Rezaul Hasan without whose support this work would not have been possible.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of figures</b>	<b>ix</b>
<b>List of tables</b>	<b>x</b>
<b>List of acronyms and abbreviations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Telescopes with multiple antennas	1
1.2 The correlator	2
1.3 Motivation	3
1.4 Thesis outline	4
<b>2 Digital CMAC design</b>	<b>6</b>
2.1 Complex multiplication and CMAC	6
2.2 CMAC multiplier design	8
2.3 Design of digital circuit cells	9
2.3.1 Adder circuits	11
2.3.2 Memory registers	15
2.4 CMAC implementation	17
<b>3 Multiple-accumulator CMACs for the input-buffered FX correlator</b>	<b>21</b>
3.1 Architecture overview	21
3.2 Grouping SIs for multiple-accumulator CMACs	23
3.3 Quantifying memory access improvements	25
3.4 Design and evaluation of multiple-accumulator CMACs	27
3.5 Conclusion	29
<b>4 DET flip-flops based on C-elements</b>	<b>30</b>
4.1 Low-glitch LG_C flip-flop	31
4.2 Implicit-pulsed IP_C flip-flop	33
4.3 Floating-node FN_C flip-flop	34
4.4 Conditional-toggle CT_C and CTF_C flip-flops	37
4.5 Simulation methodology	39
4.6 Simulation results and comparison	44
4.7 Conclusion	46

<b>5</b>	<b>Memory design for the output-buffered FX correlator</b>	<b>49</b>
5.1	Sequential access memory	50
5.2	Design description	52
5.3	Testing setup	54
5.4	Test results	56
5.5	Conclusion	59
<b>6</b>	<b>SerDes I/O design</b>	<b>62</b>
6.1	Analog design	62
6.2	LVDS clock reference receiver	63
6.3	Phased-locked loop design	64
6.4	Serialiser	64
6.5	Scrambler	66
6.6	Predriver	66
6.7	Output driver	70
6.7.1	Feed-forward equaliser	70
6.7.2	Impedance tuning	71
6.7.3	ESD protection	71
6.8	Driver biasing	71
6.9	Conclusion	76
<b>7</b>	<b>Conclusion</b>	<b>77</b>
7.1	Summary	77
7.2	Suggestions for future work	78
	<b>Bibliography</b>	<b>79</b>
<b>A</b>	<b>CMAC multiplier netlists</b>	<b>87</b>
A.1	4-bit CMAC	87
A.2	8-bit CMAC	91
<b>B</b>	<b>List of publications</b>	<b>102</b>

## List of figures

1.1	Diagram of two antennas receiving two signals. Although both antennas receive the sum of the two signals, the phase difference between these signals is different for different antennas.	2
2.1	A complex multiplier can calculate the cross-correlation without performing the complex conjugate operation explicitly. This can be achieved by simply relabelling the ports for one input and relabelling the outputs as explained by (2.3). In this case, the real and imaginary parts of the $G$ input are swapped.	7
2.2	CMAC functional diagram following (2.2) and including accumulators.	7
2.3	Example of how a signed $4b \times 12b$ multiplication can be performed without sign extensions. The two multiplication operands are $-6$ , which is $1010$ in 2's complement binary, and $-1212$ , which is $101101000100$ . The result is truncated to 16 bits.	8
2.4	An example of the adder tree design for the signed 8-bit multiplication using the Wallace tree (left and centre columns) and the tree method from [25] that is used in this work (right column). The Wallace tree uses 38 FAs and 15 HAs while the design method uses 39 FAs and 7 HAs. Symbol "  " indicates the addition of two bits with a 1, which is a special case of a HA, rather than a FA.	10
2.5	Adder tree diagrams for the 4-bit CMAC for the real (left tree) and imaginary (right tree) parts of the result of the cross-correlation operation. Red and blue colour denote partial product bits from different multiplications.	11
2.6	Adder tree diagrams for the 8-bit CMAC for the real (left tree) and imaginary (right tree) parts of the result of the cross-correlation operation. Red and blue colour denote partial product bits from different multiplications. The vertical lines for the output of the tree denote the placements of 4-bit and 5-bit CLA sections.	12
2.7	Transistor-level schematic diagram of the FA circuit that has been used throughout most of this work. The sum output is $S$ and the carry-out output is $Co$ .	13
2.8	The conventional static FA circuit [32]. This is one of the structures that is used in the provided standard cell library.	14
2.9	A pass-transistor implementation of a FA. This is one of the structures that is used in the provided standard cell library.	14
2.10	The HA schematic diagrams of (a) the circuit that is used in the work and (b) the circuit from the standard cell library.	14
2.11	The layout implementation of the FA circuit from Figure 2.7.	15

2.12	Schematic diagram of the LM DET flip-flop as used inside the CMACs' accumulators.	16
2.13	Layout view of the LM DET flip-flop from Figure 2.12 as used inside the CMACs' accumulators.	16
2.14	The schematic diagram of the pulse generator circuit of the pulsed-latch flip-flop that is shared among several latches.	17
2.15	Schematic diagram of the reset version of the pulsed latch.	17
2.16	Layout of the 4-bit CMAC.	18
2.17	Layout of the 8-bit CMAC.	19
2.18	Layout for the imaginary part of the 4-bit CMAC.	20
2.19	Layout for the imaginary part of the 8-bit CMAC.	20
3.1	Simplified diagram of the internal structure of one processing unit of Architecture 2 in [18].	22
3.2	The example of how a correlation triangle is split into SIs for the case of $w = 4$ . Numbers represent the four signal sets. (a) shows how a full integration is first split into sections. (b) shows the final pattern of SIs after auto-correlation sections are paired together into full SIs.	22
3.3	Architecture of one multiple-accumulator CMAC with $a$ accumulators. The "Accumulator Select" circuit chooses which accumulators are read from and written into in the current processing cycle.	23
3.4	The illustration showing how (a) SIs for the case of $w = 5$ can be grouped into (b) groups of 3 following the rules in Section 3.2. The arrow indicates the swapping of SIs that is performed before grouping.	24
3.5	Plot of the improvement ratio $I$ over the one-accumulator array against $w$ for two cases of $a$ .	26
3.6	Comparison of the schematic diagrams of the two data storage circuits employed in the four CMAC designs: (a) standard Latch-MUX DET FF from Section 2.3.2 for one-accumulator designs and (b) custom 3-bit variant of the same circuit for three-accumulator designs. The 3-bit circuit is only twice the size of the 1-bit circuit. E1 through E4 are the locally-generated enable signals.	27
4.1	Transistor-level implementations of a C-element that are used in this work: (a) the weak-feedback and (b) the symmetric [45] implementations.	31
4.2	Gate-level schematic of the new LG_C DET flip-flop using (a) non-inverting and (b) inverting C-elements.	31
4.3	Operational waveforms showing the behaviour of the LG_C flip-flop.	32
4.4	Schematic diagram of a generic Latch-MUX flip-flop.	32
4.5	Operational waveforms for a generic Latch-MUX flip-flop.	32
4.6	Proposed transistor-level design of the LG_C flip-flop based on weak-feedback C-elements shown in Figure 4.1a.	33
4.7	Transistor-level schematic diagram of the implicit-pulsed IP_C DET flip-flop.	34
4.8	Gate-level schematic diagram of the implicit-pulsed IP_C DET flip-flop.	34
4.9	Logic waveforms showing the behaviour of the IP_C DET flip-flop.	35



4.10	Transistor-level diagram of the improved floating-node FN_C DET flip-flop that uses 5 C-elements including weak devices for the inner C-elements.	35
4.11	Gate-level schematic of the FN_C DET flip-flop shown in Figure 4.8 with 2 inner 3-input weak C-elements exhibiting a floating-node behaviour.	36
4.12	Simulated signal levels of the implemented FN_C flip-flop. Floating states are denoted as “~”.	36
4.13	Transistor-level schematic diagram of the conditional-toggle CT_C DET flip-flop.	38
4.14	Simulated signal levels of the CT_C flip-flop implemented in the GF 28HPP technology.	38
4.15	Transistor-level schematic diagram of the improved conditional-toggle CTF_C DET flip-flop.	38
4.17	Transistor-level schematic diagrams of the six previous DET flip-flop designs that are considered in this work for comparison with the new DET flip-flops. All circuits include input, output and clock buffering. The flip-flops are (a) LM [37], (b) EP [38], (c) LM_C [46], (d) TSP [48], (e) CP [34] and (f) IP [43].	41
4.18	Illustration of the procedure for measuring the worst-case minimum D–Q delay. This plot is for a particular Monte Carlo point of the LM flip-flop. The curves are for the four cases of CK and Q transitions. The worst-case minimum D–Q delay is marked on the plot and also on the y-axis as $t_{dq}$ .	43
4.19	Plot of the CK–Q delay versus the supply voltage for new flip-flops and two previous LM and EP designs.	47
5.1	Architecture of the SAM memory.	50
5.2	Schematic diagrams of the sense amplifier: (a) the general diagram and (b) its transistor-level circuit. $V_b$ is the voltage on the bitline and $V_p$ is the precharge voltage equal to $0.5 V_{DD}$ .	51
5.3	Top-level architecture of the fabricated chip.	52
5.4	The schematic diagram of the Clock Generator circuit.	52
5.5	The schematic diagram of the Charge Pump circuit.	53
5.6	The layout of the designed memory prototype.	54
5.7	The chip photo of the fabricated memory prototype.	55
5.8	Schematic of the test circuit with the fabricated chip.	55
5.9	Oscilloscope traces of (CH1, blue) the power supply voltage and (CH2, red) the output voltage of the charge pump during operation at high clock frequencies.	57
5.10	Oscilloscope traces of (CH1, blue) the VCO control voltage and (CH2, red) the output voltage of the charge pump when the VCO control voltage is varied manually. Note the different voltage scales for the two traces.	57
5.11	Oscilloscope traces of (CH1, blue) the VCO control voltage and (CH2, red) the output voltage of the charge pump when the VCO control voltage is generated externally to be a low-frequency saw-tooth wave.	58

5.12 Voltage traces of (CH1, blue) the VCO control voltage and (CH2, red) the “E” signal which indicates correctness of operation as reported by the built-in self-test circuits.	58
5.13 Correct traces of (CH1, blue) the “CK_out” and (CH2, red) the “mem_state” signals which are respectively one eighth and one fourth of the internal clock frequency.	59
5.14 Voltage traces of (CH1) the “CK_out” and (CH2) the correctness “E” signals as reported by the chip at a low clock frequency. The memory operates correctly.	60
5.15 Voltage traces of (CH1) the “CK_out” and (CH2) the correctness “E” signals at 773 MHz of internal VCO frequency. The “CK_out” trace does not appear to be square wave because of the 100 MHz bandwidth limit of the oscilloscope.	60
5.16 Voltage traces of (CH1) the “CK_out” and (CH2) the correctness “E” signals when the VCO is at about 1.2 GHz. The memory works correctly. The “CK_out” trace does not appear to be a square wave because of the 100 MHz bandwidth limit of the oscilloscope.	61
5.17 Voltage traces of (CH1, blue) the “CK_out” and (CH2, red) the correctness “E” signals as reported by the chip when the VCO is at above 1.2 GHz. This is the clock frequency for which the memory begins to fail. The “CK_out” trace does not appear to be a square wave because of the 100 MHz bandwidth limit of the oscilloscope.	61
6.1 Block-level diagram of the transmitter circuit.	62
6.2 The layout of the self-biased folded cascode differential amplifier.	63
6.3 Diagram of the LVDS clock receiver.	63
6.4 The layout of the designed clock reference input amplifier.	65
6.5 Schematic diagram of the implemented LC-tank oscillator.	66
6.6 The layout of the 3-to-1 serialiser.	67
6.7 The layout of the 11-to-1 serialiser.	68
6.8 The layout of the 66-to-1 serialiser.	69
6.9 The layout of the predriver circuit.	70
6.10 The schematic diagram of the designed SST driver.	71
6.11 The layout of the implemented SST driver. The image shows one half of the pseudo-differential driver design.	72
6.12 The layout of the transmission gates that can be used to regulate the driver’s impedance.	73
6.13 The layout of the bias generator circuit.	74
6.14 The layout view of the replica biasing transistors within the biasing circuit.	75

## List of tables

- |     |  |    |
|-----|--|----|
| 3.1 | Summary of the design information and simulation results of the 4-bit and 8-bit CMACs with one and three accumulators. | 28 |
| 4.1 | Simulation results of the new and previously reported DET flip-flops.  | 45 |

## List of acronyms and abbreviations

AC	Alternating current
ASIC	Application-specific integrated circuit
CLA	Carry-lookahead adder
CMAC	Complex multiplier-accumulator
CML	Current-mode logic
CMOS	Complementary metal-oxide semiconductor
CSA	Carry-save adder
CV	Coefficient of variation
DAC	Digital-to-analog converter
DC	Direct current
DET	Dual-edge-triggered
DFT	Discrete Fourier Transform
DRAM	Dynamic random-access memory
DTSCR	Diode-triggered silicon controlled rectifier
EDA	Electronic design automation
eDRAM	Embedded dynamic random-access memory
ESD	Electrostatic discharge
FA	Full adder
FF	Flip-flop
FFE	Feed-forward equaliser
GF	GlobalFoundries
HA	Half adder
IC	Integrated circuit
IDDQ	Leakage current
I/O	Input/output
IP	Intellectual property

*List of tables*

LFSR	Linear-feedback shift register
LVDS	Low-voltage differential signalling
MC	Monte Carlo
MOSFET	Metal-oxide-semiconductor field-effect transistor
MSB	Most significant bit
MUX	Multiplexer
PDP	Power-delay product
PFA	Partial full adder
PLL	Phase-locked loop
PRNG	Pseudorandom number generator
PVT	Process, voltage and temperature
RAM	Random-access memory
RSD	Relative standard deviation
SAM	Sequential-access memory
SCR	Silicon controlled rectifier
SD	Standard deviation
SerDes	Serialiser/deserialiser
SET	Single-edge-triggered
SI	Sub-integration
SST	Source-series termination
VCO	Voltage-controlled oscillator
VLSI	Very-large-scale integration