Palmo: a novel pulsed based signal processing technique for programmable mixed-signal VLSI.

Konstandinos Papathanasiou $K\omega\nu\sigma\tau\alpha\nu\tau i\nu o \Pi\alpha\pi\alpha\theta\alpha\nu\alpha\sigma i o v$



Thesis submitted for the degree of Doctor of Philosophy The University of Edinburgh



1998

Declaration

I declare that this thesis has been completed by myself and that, except where indicated to the contrary, the research documented is entirely my own.

> Konstandinos Papathanasiou Κωνσταντίνος Παπαθανασίου

To KATERINA

For her understanding, love and support throughout all these three years.

J

Acknowledgements

I would like to express my gratitude to the following people for their invaluable assistance during the course of my PhD studies.

- My supervisor Alister Hamilton and Torsten Lehmann, for their remarks, ideas and assistance.
- The Engineering & Physical Science Research Council, for providing equipment and financing the research done in the final year.
- Olivier Chapuis and Thomas Brandtner, for their invaluable help; in particular, for the ideas on the BiCMOS circuits, which invested the project with a new perspective.
- Alexander Astaras, Andy P. Connelly, Alister Hamilton, Torsten Lehmann, Demetris Savvides and Robin Woodburn for proof-reading this thesis.
- Marcus Alphey, Bill Buchan, Mark A Glover, David Mayes and Robin Woodburn for helping me with software and the Cadence monster!
- Alan Murray, David Renshaw, Emma Braithwaite, and all the student members of the *neural* group, for intelligent discussions and their invitation to numerous eating-out expeditions.
- Anna Mavromatis suggesting corrections to my home-page (check her awardwinning web-page: http://www.firstnethou.com/annam).
- Last but by no means, the least, I am grateful to Adrianos and Konstandina, my parents, for their continual support, encouragement and financial assistance throughout my entire education.

Abstract

In this thesis a new signal processing technique is presented. This technique exploits the use of pulses as the signalling mechanism. This $Palmo^1$ signalling method applied to signal processing is novel, combining the advantages of both digital and analogue techniques. Pulsed signals are robust, inherently low-power, easily regenerated, and easily distributed across and between chips. The *Palmo* cells used to perform analogue operations on the pulsed signals are compact, fast, simple and programmable.

To demonstrate the inherent suitability of the technique to programmable analogue implementations, two chips were fabricated and tested on boards containing a standard FPGA for doing the routing. Results from elementary filter implementations and A/D converters are presented to prove the validity of the approach. Finally a current-mode log-domain BiCMOS *Palmo* circuit is analysed, which enables much higher sample frequencies than the voltage domain *Palmo* counterparts.

¹The name *Palmo* is derived from the Hellenic word $\Pi A \Lambda MO\Sigma$ which means pulsebeat, pulse palpitation or series of pulses.

Table of Contents

st of	Abbre	viations and Acronyms	vii
st of	Symbo	ols	ix
st of	Figure	es	x
st of	Tables	3	xiv
ritin	g style		xiv
Intr	oducti	on	1
1.1	The H	istory and Development of Programmable Systems	. 1
	1.1.1	Research in Edinburgh	. 3
	1.1.2	Contribution to Knowledge	. 4
1.2	Thesis	Outline	. 4
1.3	Summ	ary	. 5
Ba	ackgro	ound	6
Sigr	nal Pro	cessing and Field Programmable Analogue Arrays	7
2.1	Signal	Processing Fundamentals	. 7
2.2	Filteri	ng	. 8
	2.2.1	Ideal Filters	. 8
	2.2.2	Practical Ideal Filter Approximations	. 10
	2.2.3	Filter Design Algorithm	. 11
	2.2.4	Classification of Filters	. 13
		Low-pass Filters	. 13
		High-pass Filters	. 14
	2.2.5	Band-pass Filters	. 14
		Wideband	. 14
		Narrowband	. 15
	2.2.6	Bandstop Filters	. 16
	st of st of st of ritin 1.1 1.2 1.3 Ba Sign 2.1 2.2	st of Symbolst st of Figure st of Tables riting style Introducti 1.1 The H 1.1.1 1.1.2 1.2 Thesis 1.3 Summ Backgro Signal Pro 2.1 Signal 2.2 Filteri 2.2.1 2.2.2 2.2.3 2.2.4	st of Symbols st of Figures st of Tables riting style Introduction 1.1 The History and Development of Programmable Systems

			10
	2.2.7	Implementing Filters With Integrators	16
	2.2.8	Sampled Data filters	17
2.3	Sample	ed Data Signal Processing	18
2.4	Digita	Signal Representation	19
2.5	Analog	gue to Digital, Digital to Analogue Conversion Definitions	20
	2.5.1	Properties of Converters	20
	2.5.2	Serial Analogue to Digital Converters	21
	2.5.3	Oversampled $\Sigma - \Delta$ Analogue to Digital Converter \ldots .	23
	2.5.4	Other ADC Implementations	24
		Time-interleaving ADC	24
		Pipeline ADC	24
2.6	Eleme	ntary Signal Processing Functions	24
		Scaling	24
		Integration	25
		Multiplication of two Signals	25
		Comparison	25
		Discussion	25
2.7	Integr	ator and Scaler Implementations	26
	2.7.1	Continuous-Time Integrators	26
	2.7.2	Switched-Capacitor Integrators	28
	2.7.3	Switched Current (SI) Integrators	30
2.8	\mathbf{Field}	Programmable Analogue Arrays	33
	2.8.1	Reconfigurable Analogue Hardware	33
	2.8.2	Present FPAA Implementations	34
	2.8.3	Field Programmable Mixed-Signal Arrays	35
	2.8.4	Discussion	36
	2.8.5	Pulse-Based circuit outline	39
Pa	$lmo~{f Te}$	chniques: The pulse based approach	41
3.1	The u	se of pulses in analogue systems	41
	3.1.1	Advantages of pulsed systems for the implementation of FP- MAs	44
3.2	Typic	al Palmo Cell	45
3.3	Signa	Representation	46
3.4	Palmo	o Voltage Domain Implementation	50
	3.4.1	Palmo Miller Integrator	51
	 2.3 2.4 2.5 2.6 2.7 2.8 <i>Pal</i> 3.1 3.2 3.3 3.4 	2.2.7 2.2.8 2.3 Sample 2.4 Digital 2.5 Analog 2.5.1 2.5.2 2.5.3 2.5.4 2.5.3 2.5.4 2.6 Eleme 3.1 Integr 2.7.1 2.7.2 2.7.3 2.8 Field 2.8.1 2.8.2 2.8.3 2.8.4 2.8.1 2.8.2 2.8.3 2.8.4 2.8.5 <i>Palmo</i> Te 3.1 The u 3.1.1 3.2 Typic 3.3 Signal 3.4 Palmo 3.4.1	2.2.7 Implementing Filters With Integrators 2.2.8 Sampled Data filters 2.3 Sampled Data Signal Processing 2.4 Digital Signal Representation 2.5 Analogue to Digital, Digital to Analogue Conversion Definitions 2.5.1 Properties of Converters 2.5.2 Serial Analogue to Digital Converters 2.5.3 Oversampled $\Sigma - \Delta$ Analogue to Digital Converter 2.5.4 Other ADC Implementations Time-interleaving ADC Pipeline ADC 2.6 Elementary Signal Processing Functions Scaling Integration Multiplication of two Signals Comparison Discussion Outration Signals 2.7.1 Continuous-Time Integrators 2.7.2 Switched-Capacitor Integrators 2.7.3 Switched Current (SI) Integrators 2.8.1 Reconfigurable Analogue Arrays 2.8.2 Present FPAA Implementations 2.8.3 Field Programmable Mixed-Signal Arrays 2.8.4 Discussion 2.8.5 Pulse-Based circuit outline 2.8.4 Discussion 2.8.5 Pulse based approach <

,

.

4.

	3.4.2	Charge-Injection	52
	3.4.3	Current Source Design	53
3.5	Harmo	onic Distortion	54
3.6	Compa	arator	59
	3.6.1	Clamped Comparator Gain	60
3.7	Curren	nt Matching	63
3.8	Filter	Implementation	65
3.9	Conclu	usions	67

II Implementations

Pai	lmo-I f	test-chip	69
4.1	Introd	uction \ldots	69
4.2	Palm	o-I Specifications	69
	4.2.1	Elementary analogue cells	69
	4.2.2	Analogue to signed-PWM conversion	. 70
	4.2.3	Routing	. 72
	4.2.4	Supply voltage	. 72
4.3	Circui	ts used in the PALMO-I Chip	. 72
	4.3.1	Charging and discharging circuits	. 72
4.4	Comp	arator	. 74
	4.4.1	Capacitor array	. 74
	4.4.2	Input Current Mirrors	. 75
4.5	Testin	g The Chip	. 75
	4.5.1	Initial Testing	. 75
	4.5.2	Signed PWM Conversion	. 76
4.6	Furth	er Testing	. 80
	4.6.1	Integrator	. 80
		Integrator Linearity	. 81
	4.6.2	Low-pass Filter Implementations	. 83
4.7	Mixed	l-Signal Systems	. 83
	4.7.1	The technique	. 84
	4.7.2	Mixed-signal FIR implementation	. 85

iii

68

86

5.	Palr	no FPA	AA and prototyping board	88
	5.1	Introdu	uction	88
	5.2	Chip A	Architecture	88
		5.2.1	Analogue Palmo cells	90
		5.2.2	Internal analogue interconnect	90
		5.2.3	Digital logic	90
		5.2.4	Comparator Architecture	91
		5.2.5	Output Buffer	91
		5.2.6	Supply Voltage	91
	5.3	Palmo	FPAA Circuit Details	91
		5.3.1	Typical Cell	92
		5.3.2	Comparator Design	93
			Comparator Design Procedure	94
		5.3.3	Precharge and Evaluate Address Decoder.	95
	•	5.3.4	SRAM implementations	96
		5.3.5	Level shifter	97
	5.4	The p	rototyping Board	98
		5.4.1	System Level considerations	99
		5.4.2	Microcontroller and peripheral chips	99
		5.4.3	Analogue biasing	99
		5.4.4	FPGA	99
		5.4.5	Prototyping area	100
	5.5	Using	the board	100
	5.6	Testin	g The FPAA Chip on the board	101
		5.6.1	Initial Testing	101
		5.6.2	Digital Functionality	102
			Discussion	105
		5.6.3	Testing the Analogue Cells	105
		5.6.4	$\Sigma - \Delta$ Modulator	105
			Discussion	107
	5.7	Conclu	usions \ldots	108
6.	Adv	vanced	implementations: Log-domain BiCMOS Palmo cells	109
	6.1	Bipola	ar Background	110
			The translinear principal	111
	6.2	The L	Log-domain Palmo Cell	111
			Discussion	112

	6.3	Log-do	main cell	3
		6.3.1	The Log-domain Integrator	3
		6.3.2	Current Controlled Comparator	5
	6.4	Voltag	e to Current Converter	8
	6.5	Simula	ted Results	0
		6.5.1	Integrator Linearity	0
		6.5.2	A first-order filter implementation	0
	6.6	Log-do	main Multiplier \ldots \ldots \ldots \ldots \ldots \ldots 12	2
			Log-domain multiplier discussion	3
	6.7	Conclu	nsions	3
7.	Dev	elopm	ents and Conclusions 12	5
	7.1	Introd	uction	5
	7.2	Our A	pproach	5
	7.3	Curren	nt Developments	7
		7.3.1	Using Gate Arrays for Application Specific DSP 12	7
		7.3.2	Texas Instruments TMS-320C6x	8
		7.3.3	Motorola announces the first commercial FPMA 12	8
		7.3.4	Institute for System-Level-Integration	9
	7.4	Future	e Developments	0
		7.4.1	Future Work	1
			Voltage domain CMOS circuits	1
			Log-Domain BiCMOS circuits	2
			Applications	3
		7.4.2	Overall Conclusions	4
Bi	ibliog	graphy	Categories 13	6
B	ibliog	graphy	13	7
A	. Pal	mo FP	AA Addressing Registers 14	7
B	. Pal	mo FP	AA Pin Out 14	9
\mathbf{C}	. Mic	rocon	troller Code 15	1
	C.1	Comn	nands	1
	C.2	Micro	controller Code	1
D	. Boa	ard Do	cumentation 15	7
	D.1	Initial	lisation	7
	D.2	Board	Schematic Diagrams	7

Е.	FPGA Filter Schematics	160
F.	Publications	162
	ISCAS 1996	. 163
	ICECS 1996	. 167
	Colloquium on Analogue Signal Processing 1996	. 171
	IEEE-CAS 1997	. 177
	Submited to IEE Electronics Letters	. 182

List of Abbreviations and Acronyms

- **AC** Alternating Currents.
- ADC Analogue to Digital Conversion.
- AIC Analogue Interfacing Chip.
- ALE Address Latch Enable.
- AMD Advanced Micro Devices (Corporate name).
- **AN** Manufacturer's Application Note.
- **AND** Boolean AND operation (+).
- ANN Artifficiall Neural Networks.
- **ASCII** American Standard Code for Information Interchange: standard US-UK and international character set.
- **BAUD** From the name of Emile Baudot, inventor of the Telex code: now a measure of communications capacity.
 - BCD Binary Coded Decimal number.
- BUS Basic Utility System: used for digital system interconnect.
- BiCMOS Bipolar Complementary Metal Oxide Semiconductor.
 - **CAD** Computer Aided Design (tool).
 - **CCC** Current Controlled Comparator.
 - CM Current Mode.
 - CMOS Complementary Metal Oxide Semiconductor.
 - **CMRR** Common Mode Rejection Ratio.
 - **CNN** A programmable Cellular Neural Network universal machine.
 - DAC Digital to Analogue Converter.
 - **DC** Direct Current.
 - **DSP** Digital Signal Processing (or Digital Signal Processor if referring to a semiconductor device).
- EEPROM Electrically Erasable-Programmable Read-Only Memory.
 - **FIR** Finite Impulse Response.
 - FPAA Field Programmable Analogue Array.
 - FPGA Field Programmable Gate Array.
 - FPMA Field Programmable Mixed Signal Array.
 - **FSF** Filter Scaling Factor.
 - HD Harmonic Distortion.
 - **IBM** International Business Machines (Corporate name).
 - **IC** Integrated Circuit.
 - **IEE** Institute of Electrical Engineers (UK).
 - **IEEE** Institute of Electrical and Electronics Engineers (US).
 - **IIR** Infinite Impulse Response.
 - KCL Kirchoff's Current Law.

- LC Inductor-Capacitor circuit, as in filters.
- **LED** Light Emitting Diode.
- MAX Maxim Semiconductors (Corporate name).

MC Micro-Controller.

Hz HertZ: Unit of frequency.

MIT Massachusetts Institute of Technology.

MOS Metal Oxide Semiconductor.

MSB Most Significant Bit.

NAND Boolean function $(y = \overline{a+b})$.

NMOS N (Negative) channel Metal Oxide Semiconductor.

NN Neural Networks.

NPN Negative-Positive-Negative (transistor).

OPAMP OPerational AMPlifier.

OR Boolean function (\cdot) .

- **OTA** Operational Transconductance Amplifier.
- **OTA-C** Operational Transconductance Amplifier Capacitor, used for filter design.
 - **PC** Personal Computer.

PCB Printed Circuit Board.

PFM Pulse Frequency Modulation.

PGA Pin Grid Array (semiconductor package).

PMOS P (Positive) Channel Metal Oxide Semiconductor.

PNP Positive-Negative-Positive (transistor).

PWM Pulse Width Modulation.

RC Radio Communication.

- RLC Resistor Inductor Capacitor.
- **RS-232** Recommended Standard 232: for serial communications (by the Electronic Industries Association).
 - SC Switched Capacitor.
 - **SI** Switched Current.
 - **SLI** System Level Integration (systems on a chip).

SRAM Static Read Only Memory.

TH Track & Hold switched current circuit.

THD Total Harmonic Distortion.

TMS Texas Micro Systems (Corporate name).

UPS Uninterpretable Power Supply/System.

VCCS Voltage Controlled Current Sources.

VLSI Very Large System Integration.

XOR Exclusive OR Boolean function .

s-PWM Signed-Pulse Width Modulated signal.

C	Capacitor (F) .
C_{ox}	MOS gate-oxide capacitance per unit area.
$\tilde{f_c}$	Is the cut-off frequency.
f_n	Is the normalised at 1rad/sec cut-off frequency.
I_C	Is the collector current if a bipolar transistor.
$\tilde{I_S}$	Is the saturation current of a bipolar transistor.
${ ilde L}$	MOS transistor channel length (m) or Inductor (H) .
Minus	Palmo integrator inverting input.
\boldsymbol{n}	$n = 1, 2, 3, \dots$
Plus	Palmo integrator non inverting input.
R	Resistance (Ω) .
t	Time.
V_{AF}	The early voltage of a bipolar transistor.
V_{BE}	The Base Emitter voltage.
V_{th}	Thermal voltage: $V_t = kT/q \simeq 26mV$ at $300^{\circ}K$.
V_T	MOS threshold voltage (typically 0.8V in both processes
_	used).
W	MOS transistor channel width.
x_i	Is the input i .
y_i	Is the output <i>i</i> 1.
Z	Complex resistance (Ω) .
z^{-1}	Digital Delay.
$oldsymbol{eta}$	For bipolar transistors the ??gain??
$oldsymbol{eta}$	For MOS transistors: $\beta = \mu_p C_{ox}(W/L)$
ΔT	Time difference, often expressing the width of a pulse.
Φ	The hellenic letter "Phi".
$\Phi \& \overline{\Phi}$	Sampling clock and complement.
μ	Carrier surface mobility.
σ	Statistical variance.
ξ	The hellenic letter "xi".
$\xi_+ \& \xi$	Palmo charging current control inputs (those derive from the
	Plus & Minus cell inputs by the use of digital logic blocks).
Signal	Definition Quantity Subscript Example

Signal Definition	Quantity	Subscript	Example
Total instantaneous value	Lowercase	Uppercase	q_A
DC value	Uppercase	Uppercase	Q_A
AC value	Lowercase	Lowercase	q_a

List of Figures

•

1–1	The Anikythera Calendar Computer—Dated before 80 BC	2
2 - 1	Ideal filter $H(\omega)$ and $h(t)$.	9
2–2	Magnitude response of a practical filter realisation.	10
2–3	Typical L–C low-pass ladder filter.	11
2–4	Wideband band-pass filter realisation by cascading a low-pass and a high-pass filter.	15
2-5	Narrowband gain reduction	15
2–6	Wideband bandstop filter implementation, using a low-pass and a high-pass filter with an adder.	15
2–7	Integrator based implementation of a low-pass LC filter.	16
2–8	Sampling a sin-wave input.	18
2–9	Block diagram of a single-slope ADC	21
2–10	Block diagram of a double-slope ADC.	22
2–11	First and second order $\Sigma - \Delta$ modulator	23
2-12	OTA-C integrators: block diagram and circuit details	26
2–13	SC resistance simulation.	28
2–14	Switched Capacitor Integrator	29
2–15	Elementary Track and Hold circuit.	31
2–16	Track & Hold integrator	32
2–17	⁷ Pulse-Based FPMA implementation A) Device Array B) Analogue circuit block diagram C) Integrator details	39
3–1	Stochastic signal Multiplication and Addition.	41
3–2	Pulse-Stream Neural Networks.	43

3–3	Typical Palmo Cell	45
3–4	Different PWM generation schemes	47
3–5	A) Scaling by the use of the Ramp, B) Saturated Ramp (double and single sided).	48
3–6	The Palmo signalling mechanism.	49
3–7	Palmo typical voltage domain Cell.	50
3–8	MOS switch implementations with reduced clock-feedthrough	53
3–9	Current source circuit	53
3–10	Harmonic Distortion due to comparator delays A) Comparator in- verting stage B) $\cos(x)$ C) Minimum pulse effect $f(x)$ D) Output distorted signal (signals reconstructed for clarity)	55
3–11	Harmonic distortion due to current differences	57
3-12	2 Total Harmonic Distortion	59
3–13	3 A) Clamped Comparator, B) Clamped Comparator with positive feedback to increase the gain.	60
3-14	4 Clamped Comparator small signal analysis	61
3–15	5 Output stage gain calculation	63
3–10	6 Filter implementation using differential integrators (a) of a RLC low-pass filter (b)	65
3–1'	7 Frequency response of the z-domain transfer function and Palmo Filter	65
4–1	Block diagram of PALMO-I cells (signals noted with a '*' are global).	71
4–2	Switched current sources	73
4–3	Comparator schematic diagram, all the NMOS bulks are connected to ground.	74
4-4	Palmo Chip Photograph	76
4–5	Original PWM conversion linearity results	78
4–6	Improved linearity results, by the use of the <i>minimum-pulse</i> cancel- lation	78
4–7	Minimum Pulse Generator	79

4–8 I	Integrating a signed input: the pulsed based approach 80
4-9]	Integrator programmability and linearity
4–10 v	VLSI results from First Palmo Chip: 1st, 2nd and 3rd order filters at cut-off frequencies of 1kHz and 2kHz
4-11	Mixed-signal technique
4-12	Palmo mixed-signal 24 tap FIR filter implementation results 85
5-1	Block diagram of the PALMO-FPAA chip
5-2	Typical cell circuit diagram
5-3	Palmo FPAA comparator schematic
5–4	Address Decoder
5–5	SRAM implementations: A) Standard quasi-static cell, B) DynamicSRAM cell.96
5–6	Input voltage level shifter
5-7	Prototyping board block (and floor) diagram
5–8	Prototyping board and host laptop
5–9	Palmo Chip Photograph
5–10	Schematic diagram of the FPGA cell used to configure the analogue FPAA chips
5–11	Addressing in the FPGA ($address = 80h$)
5–12	Loading "AAh" to the FPAA
5–13	PWM output by the use of an externally generated ramp 104
5–14	Palmo Sigma-Delta modulator
5–15	Sigma-Delta modulator results
6–1	Temperature dependency of Is for Vb=700mV of the BiCMOS pro- cess used.
6–2	Palmo cell and typical waveform diagram.
6–3	Log-domain integrator.
6–4	Current comparator

-

Ŀ

• .

6–5	Voltage-Current-Converter
6–6	Linearity of the Palmo cell
6–7	Frequency response of a simple Palmo filter, fs=1MHz
6–8	Log-domain multiplier
7–1	Typical Palmo FPMA implementation
7–2	1996–1997 World-wide Electronics Market-place (source: ICE status
	1997)
7–3	Dedicated minimum pulse cancellation
A-1	FPAA address register
A-2	FPAA interconnect registers
A–3	Typical cell Capacitor and DAC registers
E1	Second order <i>Palmo</i> filter implementation—FPGA schematic. It includes digital logic to drive the <i>Palmo</i> inputs, <i>signed PWM</i> and <i>Ramp</i> generation
E-2	Tap input configuration
E-3	signed PWM output generation
E-4	Ramp generating cell

List of Tables

2 - 1	Modern filter approximations
2-2	Filter transformations. (Low-Pass parameters derive from tables) $.13$
2–3	SC and SI characteristics
4–1	Summary of the Palmo-I characteristics
5 - 1	Clamped Comparator Implementation Parameters
5-2	Summary of the Palmo FPAA characteristics
6–1	Summary of the BiCMOS chip
7-1	Market-growth tendencies in the foreseeable future
7–2	Characteristics of Analogue, Palmo and Digital signal processing
	implementations
B-1	Palmo FPAA Pin out part I
B-2	Palmo FPAA Pin out part II

Writing style

I use "we" as the personal pronoun throughout this thesis, as I think this eases both reading and writing.

Chapter 1

Introduction

1.1 The History and Development of Programmable Systems

The development and continued improvement of programmable machines is one of the most astonishing achievements of the human mind. Though the word "*Computer*" brings to mind the desktop PC or hand-held calculator, in fact programmable, computing systems are used in almost every modern electronic device.

The abacus was probably the first attempt at mechanised calculation. It was used by many civilisations thousands of years ago. Archimedes had constructed gear-based, mechanical computers at about 250BC [1,2]. The earliest evidence of mechanical computers comes from 80 BC [3,4]. A device was found in a wreckage of a ship (figure 1–1), carrying treasures from the island of Rhodes and the Greek cities on the coast of Asia Minor, to mainland Greece. This device included over thirty gears, representing the movements of planets and was used to calculate the phase of the moon and the location of the stars, possibly for navigation purposes.

A slow course of development of programmable systems took off from these early mechanical computers, calling at *Pascal's* adding machine, *Babbage's* analytical engine (which never got off the drawing board), *Boole's* algebra, the invention of electricity and the vacuum tube. This led to computers (The *Colossus*, MARK-I and ENIAC) and the invention of the transistor in the 1940s. However, the vision



Figure 1-1: The Anikythera Calendar Computer-Dated before 80 BC.

for expansion was still limited and the general manager of IBM characteristically stated: "There is a global market for 15 to 20 computers"!

Early computing machines were either analogue or digital. However, the development of the first *Central Processing Unit* $(CPU)^1$ made modern, massively produced-easily programmable digital systems very cheap. The rapid growth of the digital market, during the eighties and the decreasing use of analogue circuits, led some to pronounce analogue systems "*Dead*"!

However in the nineties new applications emerged: battery operated mobile systems such as mobile telephones, car injection systems, portable computers,

¹The *Intel* four bit 4004 which is the predecessor of modern CPUs. In fact code written for the 4004 can be directly compiled and run on the most advanced *Intel* CPUs

hearing aids and others, which gave a boost to the analogue semiconductor industry. Recent market developments indicate that programmable analogue circuits could have an impact on the semiconductor industry, similar to that of the digital systems during the eighties.

1.1.1 Research in Edinburgh

The author's initial research plans were to apply techniques, which were developed for artificial neural networks [5,6,7,8] (in particular stochastic neural networks [9, 10]), to real world applications [11]. The early literature survey outlined the problems of interfacing a neural network to the environment [12,13,14,15,16,17]. The uneven distribution of signal energy [18,19] in the frequency domain, common in most practical signals, makes an interfacing technique used for one application useless in some other.

It was realised that in order to interface analogue inputs to a neural network, the signals must be modified to facilitate the extraction of useful information. The use of the *wavelet* [20,21] decomposition was investigated, hoping this way to provide the means for interfacing analogue signals to neural networks. The basic concept behind this transformation is to divide the signal spectrum into its subspectra, or subbands, and then to treat those subbands individually[18,20, 21]. A neural network would then be trained to respond to specific inputs and generalise for the purpose at hand [22].

The author consequently looked into other analogue *wavelet* implementations [23, 24] and standard analogue filtering techniques. The *wavelet* transformation can be performed by the use of a ladder filter [18,20,21]. However, the transfer function of the filter varies according to specific applications. No analogue approach provides satisfactory solution to filter programmability (table 2–3). Therefore, it seemed that digital systems are preferable to *wavelet* implementations [25,26,27,28].

Then the similarities of the neural network arithmetics and analogue filtering requirements suddenly became evident. It was realised that by the use of certain techniques used in modern neural network implementations, it is possible to combine analogue functionality with digital signals. Consequently the circuits will benefit from both analogue and digital worlds in a mixed-signal approach [29].

1.1.2 Contribution to Knowledge

This thesis presents an entirely novel strategy for implementing programmable mixed-signal hardware [29], pioneered by the author. The novelty of the technique lies in the use of pulses, to encode in time analogue values, such as a voltage or a current, by modulating the width or the frequency of the resultant pulse(s) [29,30,11]. In that way, all the I/O signals encode analogue information, even though they are digital in nature. The approach will be demonstrated to operate in both analogue filter implementations [31,30], similar to conventional sampled-data techniques, and mixed-signal algorithms performing DSP specific tasks [11,32]. CMOS circuits were manufactured to demonstrate the functionality of the technique , while the limitations and performance implications [33] of the approach are analysed here. The first ever sampled-data, log-domain BiCMOS circuit is proposed [34], using this novel signalling scheme, and offering a unique set of advantages to our approach. Finally, recommendations for future research and subsystem improvements are proposed, since the territory we have exposed is as yet only a little explored and ripe for future investigation.

1.2 Thesis Outline

This thesis consists of two parts. The first part contains the background which is essential to the understanding of our work. Chapter 2 gives a brief introduction to signal processing, a general filter design algorithm and the principals behind analogue-to-digital-to-analogue conversion. Furthermore, conventional approaches to signal processing are presented, to signify their limitations. Finally an introduction to programmable analogue hardware is outlined.

Chapter 3 gives the principals of the new technique. Different signalling mechanisms are presented, the significance of the ramp is highlighted, and a first ap-

Introduction

proach to our circuits is analysed. The chapter concludes with an example of a *Palmo* filter implementation.

The second part of the thesis presents practical implementations of the new programmable analogue cell. Chapter 4 is about our first test chip, which was designed by the author in order to demonstrate the *Palmo* approach. The circuits are analysed, the limitations and our solutions to problems encountered are presented. Results from some simple filter implementations are also produced.

In Chapter 5 our second programmable chip is presented. The design considerations are highlighted, in order to facilitate the understanding of the constrains of a programmable analogue chip. The details of a testing board and initial test results are presented in this chapter as well.

Chapter 6 demonstrates a BiCMOS *Palmo* approach to programmable mixedsignal hardware. A chip using log-domain principals is currently at the final design stages, we believe that it will perform much better than our old circuits.

Finally, Chapter 7 discusses the issues raised in this thesis, presents the current semiconductor developments as well as the future trends, and draws conclusions on the perspectives of this work.

1.3 Summary

This chapter presented a short flashback on the programmable device history and discussed our motivations behind the implementation of mixed-signal systems. In line with this point, the aim of the thesis was presented. It concluded by giving a brief outline of our work which will be presented in the following chapters.

Part I

•

Background

Chapter 2

Signal Processing and Field Programmable Analogue Arrays

2.1 Signal Processing Fundamentals

Signal Processing is an area of science and engineering that has developed rapidly over the past years. The evolution of computers and the need to interface them to the environment, has given rise to the need for even faster and more efficient signal manipulation. In this chapter some basic concepts and definitions will be presented, useful for our further discussion.

A *signal* is defined as a physical value that varies with time, space, frequency or any other independent variable or variables. Therefore a signal can be defined by the use of a mathematical function of one or multiple independent variables. For example the functions:

$$s_1(t) = 4 \cdot t + 5$$

 $s_2(\omega \cdot t) = sin^2(\omega \cdot t)$

express the signal s_1 as a function of the independent variable "time" (t) and s_2 as a function of $(\omega \cdot t)$.

From all the natural *signals* it is possible to focus on speech, to demonstrate a signal which provides information as a function of a single independent variable namely time. Other signals such as an image are two dimensional signals. The two independent variables in that case are the spatial co-ordinates. There are many other *signal* classifications for example continuous and discrete, deterministic and random, periodic and aperiodic, symmetric and asymmetric.

A system can be defined as a physical device (or even a software realisation) that performs an operation to a signal. For example the ear is a system which transforms the uni-dimensional sound to a vector of synapse pulses, which stimulate our brain. Every signal is associated with a *signal source*, the *system* which generates it. In our example speech is generated by the vibration of the vocal chords and consequently is modified by the shape of the vocal cavity, amongst other things. The operations performed to *signals* by *systems* is usually referred as *signal processing*.

Finally the idea of a *filter* is introduced. This is a system which is used to reduce noise or interference to an input *signal*. A filter is characterised by the type of operation it performs to the *signal*, it could be linear or nonlinear and so forth.

2.2 Filtering

One of the most important signal processing applications is filtering. Filters are used in virtually every modern electronic system available in the market. In this section we will give an introduction to continuous and sampled-data filtering, we will present a filter synthesis technique and we will demonstrate the use of active components to implement LC ladder filters.

2.2.1 Ideal Filters

The word filter stands for a device that discriminates between its inputs according to their attributes. This function of the filtering device can be used for cleaning out impurities or to obtain an output within certain specifications. Therefore an air filter on a aeroplane cleans the air, which is recycled in the cabin, from dust and other small particles. A petrol filter stops impurities from entering the engine. A pair of sun glasses stops the dangerous ultraviolet radiation from damaging our eyes.

In that sense a linear time-invariant system discriminates between the various frequency components of the input signal. The output in the frequency domain $(Y(\omega))$ is defined by the impulse response $H(\omega)$, giving $Y(\omega) = H(\omega) \cdot X(\omega)$, where $X(\omega)$ is the input. Such a system acts as a heightening function or a spectral shaping function and can be called a frequency shaping filter.



Figure 2–1: Ideal filter $H(\omega)$ and h(t).

In our discussion we will focus on low-pass filters in order to demonstrate a technique for filter design. In the following sections it will be clarified that every filter can be realised from a low-pass prototype, by the use of the appropriate transformations [35,36,37,38,39]. An ideal low-pass filter would in principle enable all low-frequency components of an input signal to be transmitted, however it should stop any component with a frequency higher than ω_0 . The impulse response $(H(\omega))$ of an ideal low-pass filter is given by the equation

$$H(\omega) = \begin{cases} 1, & |\omega| < \omega_0 \\ 0, & otherwise \end{cases}$$
(2.1)

The impulse response on the frequency domain $(H(\omega))$ is shown in figure 2-1a. The impulse response on the time domain of this filter is given by the inverse Fourier transformation of equation (2.1) [40,41].

$$h(t) = \omega_0 \cdot \frac{\sin \omega_0 t}{t}$$
(2.2)

The plot of equation (2.2) in time is shown in figure 2–1b. It is clear that h(t) is not zero for t < 0, or in other words, the ideal filter is *noncasual* therefore unrealisables in practice, because its response is dependent on future inputs [35].

2.2.2 Practical Ideal Filter Approximations



Figure 2–2: Magnitude response of a practical filter realisation.

In practice filter specifications can be relaxed from the strict ideal and such a non-ideal system can be realisable in "real-world" applications. Therefore a small attenuation in both the *pass-band* and the *stop-band* ($\delta_{p(ass)}$, $\delta_{s(top)}$ in figure 2– 2) is tolerable. Moreover there is usually no need for a "brick-wall" response, instead a *transition region* or *transition band* is affordable. This transition region is denoted by the two edge frequencies ($\omega_{p(ass)}$ and $\omega_{s(top)}$). The width of the *passband* is called also the *bandwidth*, for a low-pass filter the bandwidth is ω_p . Usually the magnitude of $H(\omega)$ is plotted in decibels (dB) to accommodate the large dynamic range of the graph of the frequency response. Decibels are expressed as the $-20log_{10}H(\omega)$. An *octave* is defined as the ratio between two frequencies which is equal to 2.

Modern network theory has provided many different filter approximations, a brief comparison in between them is given in table 2–1.

LC ladder networks can be used to implement those filter approximations [39, 36,38,37]. A typical Nth order LC low-pass filter is shown in figure 2–3, the capacitance and inductance values are normalised for a cut-off frequency of 1rad/Sec, for every filter approximations (*Butterworth, Chebyshev,* etc). Those parameter values, for a normalised low-pass filter, are given in the relevant bibliography. By



Figure 2-3: Typical L-C low-pass ladder filter.

Category	Advantages	Disadvantages
Butterworth	Flattest passband	Medium transition rate, medium distortion
Chebyshev	Fast transition ratio	High distortion, high passband ripple
Elliptic	Fastest transition rate	High distortion, high passband ripple
Bessel	Least distortion	Slow transition rate
Gaussian	Low distortion	Medium transition rate

applying the appropriate transformations to those normalised filters, it is possible to implement any linear filter structure.

2.2.3 Filter Design Algorithm

As it was mentioned above, filter design is a simple, automated task. An algorithmic approach to continuous-time or sampled data active filter design, is presented and analysed in this section.

1. **Specifications:** At first the designer must define the specifications of the filter. The type (low-pass, high-pass, bandpass or bandstop), the attenuation

 (δ_p, δ_s) , the transition region (ω_p, ω_s) and the passband should be defined according to the filtering task.

- 2. Filter circuit: The second step is the selection of the filter approximation (Butterworth, Chebyshev, Bessel, elliptic, etc) and order, to meet the specifications. The designer has a wide variety of filter types to choose from, the decision will be influenced by factors such as complexity of the implementation, device constrains, frequency range, degree of selectivity, level, size and cost.
- 3. **Transformations:** The appropriate transformation (table 2–2) is applied, in order to obtain the desired filter from the normalised low-pass ladder circuits.
- 4. Denormalisation: Scaling to the desired cut-off frequency is performed by the use of the equations (2.3) and (2.4). Note that sampled-data filters are usually easier to implement if the *impedance scaling factor* Z = 1.
- 5. Integrator circuit: Transforming the LC ladder filter to the equivalent integrator based filter (as in figure 2-7), is the next step. The integrator scaling factors K are derived from the equation K = 1/X, where X is one of the LC filter denormalised active components (L or C).
- 6. Sampled-Data filters: For an sampled-data filter implementation some further scaling (K') is used:

$$K' = \frac{K}{f_s} = \frac{1}{X \cdot f_s}$$

where again X is one of the LC filter denormalised active components (L or C) and f_s is the sampling frequency, which should be significantly bigger than the maximum input frequency.

7. Analysis: The filter performance is analysed, with respect to tolerances, in order to verify that the specifications are still met by the filter circuit.

8. Filter Construction. The filter is constructed and the output can consequently be measured.

2.2.4 Classification of Filters

There are four filter classes namely *low-pass*, *high-pass*, *band-pass* and *band-stop*. These filters can derive from the normalised LC ladder circuits of a *low-pass* filter for a given approximation.



Table 2–2. Filter transformations. (Low-Pass parameters derive from tables)

Low-pass Filters

Low-pass filters filter high-frequency components, while they enable the transmission of low-frequency signals. The *cut-off frequency*, passband & stopband attenuation and slope in the transition area specify a low-pass filter. Filter coefficients have to be scaled, in order to denormalise the filter and to meet the requirements of the *cut-off frequency* and the *load resistance*. To do this the *Frequency Scaling Factor (FSF)* is calculated:

$$FSF = \frac{f_n}{f_c} \tag{2.3}$$

where f_c is the desired cut-off frequency and f_n the normalised one in the tables. Since in most of the cases the cut-off frequency is given in Hertz (Hz) and the tables are normalised at 1rad/sec,

$$FSF = \frac{1rad/sec}{2\pi f_c rad/sec}$$

where f_c is the desired cut-off frequency in Hertz.

By the use of FSF and the *impedance scaling factor* (Z) all the parameters of an LC filter are scaled according to the following equations:

$$R' = R \cdot Z$$

$$C' = \frac{C}{FSF \cdot Z}$$

$$L' = \frac{L \cdot Z}{FSF}$$
(2.4)

High-pass Filters

High-pass filters enable the transmission of high-frequency signals, while cancelling low-frequency components. High pass filters can be easily derived from low-pass equivalent filters by applying the appropriate transformation shown in table 2–2.

2.2.5 Band-pass Filters

Wideband It is possible to implement wideband band-pass filters by the use of a lowpass and a highpass filter (figure 2-4), provided that the two filters have equal *source* and *terminating* resistors and that the cut-off frequencies are separated by at least one octave.







Figure 2–5: Narrowband gain reduction

Narrowband If the lower and upper 3dB limits of the bandpass filter (f_1, f_2) are very close, some reduction in the gain might occur (figure 2-5). If this effect is taken into account the filter might not meet the passband ripple specifications. In that case the narrowband approach is used:



Figure 2–6: Wideband bandstop filter implementation, using a low-pass and a high-pass filter with an adder.

The filter is designed based on a low-pass filter with a cut off frequency (f_0) equal to the geometric mean of f_1 , f_2 and the desired ripple and cut-off slope. By the use of

$$f_0 = \sqrt{f_1 \cdot f_2}$$

It is possible to transform the low-pass filter into the desired band-pass one by applying the appropriate transformation shown in table 2–2.

2.2.6 Bandstop Filters

Wideband It is possible to implement a bandstop filter by using a low-pass, a high-pass filter and an adder (figure 2–6), provided that the source and load resistors of the two filters and the adder match each other and that the cutoff frequencies are separated by at least one octave.

Narrowband In the case were the lower and higher limits of the bandstop filter are so close that it is not possible to implement the filter by the circuit of figure 2–6, the band-stop transformation shown in table 2–2 is used.

2.2.7 Implementing Filters With Integrators



Figure 2–7: Integrator based implementation of a low-pass LC filter.

The LC components of a ladder filter can be implemented by the use of differential integrators. Take a typical LC low-pass filter of figure 2–3. The current and voltage equations for any C_k or L_k $(1 < k \le n)$ are given by the equations:

$$v_{c_k}(s) = \frac{1}{C_k \cdot s} \left[i_{L_{k-1}}(s) - i_{L_k}(s) \right]$$
$$v_{L_k}(s) = L_k \cdot s i_{L_k}$$

or

$$i_{L_{k}} = \frac{1}{L_{k} \cdot s} \left[v_{C_{k-1}}(s) - v_{C_{k}}(s) \right]$$

It is possible to implement the function of the inductor or the capacitor by a differential integrator as shown in figure 2–7.

The integrator based filter implementation can be easily derived from the LC diagram, experienced designers would be able to construct the integrator circuit without even having to use the transfer equations of the LC circuit. Note that the input current for the capacitor C_1 comes from R_{in} therefore the connectivity of the first integrator is slightly modified because:

$$v_{C_1}(s) = \frac{1}{C_1 \cdot s} \left[\frac{v_{in} - v_{C_1}}{R_{in}} - i_{L_1}(s) \right]$$

The same applies at the output, where

$$v_{C_n}(s) = \frac{1}{C_n \cdot s} \left[i_{L_{n-1}}(s) - i_{out}(s) \right] = \frac{1}{C_n \cdot s} \left[i_{L_{n-1}}(s) - \frac{vout(s)}{R_l} \right]$$

2.2.8 Sampled Data filters

As it was mentioned above, filters can be implemented by the use of integrators, those integrators can be sampled-data analogue or digital cells. The integrator expressed as a function of s is suitable for continuous systems only, therefore an equivalent circuit operating in the z domain should be used for sampled-data implementations [42,43,44]. Take the system which has a transfer function given by the following equation:

$$H(z) = K \cdot \frac{1}{1 - z^{-1}} \tag{2.5}$$

which can be rewritten as:

$$H(z) = K \cdot \frac{z^{1/2}}{z^{1/2} - z^{-1/2}}$$

To obtain the response for the physical frequencies ω we substitute $z = e^{-jT}$, where T is the sampling frequency.

$$H(z) = K \cdot \frac{e^{jT/2}}{j2\sin(\omega T/2)}$$

Observe that this is an integrator in the $sin(\omega)$ domain. However if $\omega T \ll 1$ then $sin(\omega T/2) \simeq \omega T/2$ therefore (2.5) can be written as:

$$H(z) \simeq \frac{K}{T} \cdot \frac{e^{jT/2}}{j\omega}$$
(2.6)
The result of (2.6) is actually an integrator on the $s = j\omega$ domain, with a scaling factor K/T and some phase lead $(e^{jT/2})$, where T is the sampling frequency. The phase lead is not significant, partly because it does not change the magnitude of the output, which is important for filtering and partly because it is possible to use integrators in feedback-loops. In such an *integrator loop* one of them would add some phase lead $(e^{j\omega})$ and the other would add some phase lag $(e^{-j\omega})$, cancelling any phase shift. The integrator which gives phase lag is given by the equation:

$$H(z) = K \cdot \frac{z^{-1}}{1 - z^{-1}}$$

The approximation $\omega T \ll 1$ used in (2.6) is very critical. If the filter is not clocked at a much higher frequency than the signal frequency (in some cases ten times the maximum input frequency) significant errors will occur, unless this effect is taken into account (*exact design* technique [42]). Such an integrator approximation in the z domain is also called a *Miller* integrator.

2.3 Sampled Data Signal Processing



Figure 2–8: Sampling a sin-wave input.

In this section the *discrete* (or sampled) *signals* and the *sampling* principal is presented. Let us consider the sin-wave signal (sin(t)) of (figure 2-8), it can be realized as a series of discrete samples:

$$s_3(n) = \sin(nT)$$

where n = 1, 2, 3, ... and $T = \frac{1}{10} \cdot \pi$. $s_3(n)$ is a discrete version of the continuous time signal sin(t).

Sampling is the function of transforming a continuous signal to a discrete one. We limit our discussion to *periodic sampling*, which is the one used most frequently in practice. It is given by the relation

$$x(n) = x_a(nT), \quad -\infty < n < \infty \tag{2.7}$$

If we assume that an analogue signal can be represented as a sum of sinusoids of different amplitudes, frequencies and phases (also known as the *Fourier* theorem):

$$x_a(t) = \sum_{i=1}^N A_i \cos(2\pi F_i t + \theta_i)$$

and we ensure that F_{max} does not exceed some predetermined value, then we can select the sampling rate so that $F_s > 2 \cdot F_{max}$ (Nyquist frequency). In this case all the sinusoidal components in the analogue signal are mapped into corresponding discrete-time frequency components. Therefore the analogue signal can be reconstructed without distortion from the sample values using the interpolation given by the sampling theorem [35,36].

2.4 Digital Signal Representation

It is possible to realise sampled data as analogue values (ex. a voltage or a current) but also as digital values. The operations performed to digital representations of signals is usually called *Digital Signal Processing* [35,36].

There are five commonly used digital value codes namely Natural Binary, Binary Coded Decimal, Offset Binary, Two's Compliment Binary and Sign plus Magnitude. The latter is the one mostly used in our work.

Sign plus Magnitude Is the coding technique were the MSB contains the information about the polarity of the number (ex. "1" indicates a positive number), while the magnitude of the number is defined by all the other bits. This coding provides a double zero (a "positive" and a "negative" one). Note also that

it is very simple to invert or rectify such a code (by changing the sign bit), since positive and negative values are symmetrical in magnitude.

2.5 Analogue to Digital, Digital to Analogue Conversion Definitions

A mixed signal analogue-digital system, typically consists of an Analogue to Digital Converter (ADC), a digital circuit and a Digital to Analogue Converter (DAC). The interfacing to the analogue word is done through the two converters, while the desired algorithm is performed by the digital circuit [42,45,46]. Input level shifting, analogue sampling and antialiasing filtering might be needed at the input, while some smoothing must be performed at the output. In general the ADC will convert a sampled analogue signal to a digital word representation. Often more than one analogue inputs are multiplexed at the input of the ADC. The DAC process regenerates analogue output values, from the input digital representation. In this section some conversion principals and some basic ADC schemes will be presented, which will be used in our future discussion.

2.5.1 Properties of Converters

The performance of the converters is critical to the overall operation of the implementation. Careful characterisation of the converters is needed to specify the system. To facilitate the characterisation process the properties of a converter are presented in the following paragraphs:

Resolution is the number of bits that the converter is able to distinguish.

Accuracy is the difference between the desired mapping of the analogue to the digital signal and the mapping that was achieved.

Quantising Error is the maximum deviation from a straight line transfer function of a perfect ADC. The analogue to digital transformation quantises the analogue input into a finite number of output codes. Offset Error is the output of a DAC when the input is zero, or the required input to an ADC to output zero.

Linearity Errors indicate the departure from a linear transfer curve for either an ADC or a DAC. Linearity errors do not include quantising, accuracy or offset errors.

Conversion Time is the speed at which an ADC or a DAC can make repetitive data conversions. This time depends in the internal architecture of the ADC or DAC circuits.

2.5.2 Serial Analogue to Digital Converters



Figure 2–9: Block diagram of a single-slope ADC.

The serial ADC performs serial operations until the conversion is complete [44, Appendix7-1][46, pp.638-651]. A typical serial single-slope ADC is shown in figure 2-9. The circuit consists of a ramp-generating integrator, a comparator, two counters and an AND gate. The operation of the circuit is controlled by the input clock (f = 1/T) and the counter which resets the integrator. At the beginning of every cycle the ramp is zeroed. The inputs of the comparator force the output to high. The *Ref* input is integrated in time (generating the ramp). The output of the comparator is ANDed with the input clock, thus generating a series of clock pulses which are counted by the second counter generating the digital word at the output. When the ramp becomes equal to the *Input* the comparator goes low, thus the output of the AND gate is inhibited. The number of the resultant pulses

is proportional to the size of the input value. The output of the counter can be converted to the desired digital coding format.

The single-slope presented here can have many different implementations, while the principal of operation remains the same. The advantage of such a converter is that is is simple and therefore it can be easily implemented. However the conversion time is long (2^nT) , where *n* is the resolution in bits), it is unipolar and the ADC is subject to errors in the ramp -due to integrator nonlinearities-, comparator offsets and delays.



Figure 2–10: Block diagram of a double-slope ADC.

Figure 2-10 shows a typical double-slope ADC. The operation of this circuit is similar to the operation of the single-slope ADC, however there is a difference at the input of the ramp generating integrator. The counter changes the state of a D Flip-Flop, this controls the switch which directs the reference input to either the inverting or the noninverting input of the integrator. Thus the signal which is generated by the integrator (shown at the waveform diagram in figure 2-10) is a two sided ramp. This generates an output pulse which is proportional to the size of the input signal and it is centred around T_0 .

It is this symmetrical ramp generation that cancels any inaccuracies due to comparator delays and ramp inaccuracies. However the time needed for a conversion is double the time needed by the single slope circuit $(2^{n+1}T)$, where n is the resolution) and the input signal must be unipolar.

2.5.3 Oversampled $\Sigma - \Delta$ Analogue to Digital Converter



Figure 2–11: First and second order $\Sigma - \Delta$ modulator.

Oversampled ADCs offer enhanced resolution with a trade-off in the conversion time. A clocked negative feedback loop is used to produce a coarse estimate that oscillates about the true value of the input. A digital filter is used at the output to average this coarse and achieve a finer approximation. In that way precision ADC is performed [46, pp.668–669].

A $\Sigma - \Delta$ modulator is an oversampling ADC with a coarse estimate of one bit. A practical oversampling first and second order $\Sigma - \Delta$ modulator are shown in figure 2–11. The input to such a modulator is integrated in time, causing the comparator to switch state when it crosses zero. At this moment the DAC subtracts 1/2 from the integrated value thus forcing it down. The averaged comparator output gives the digital representation of the input. The accuracy of the DAC is not critical to the operation of a $\Sigma - \Delta$ modulator and therefore nonlinearity errors will not exist. In fact the signal to noise ratio of first and second order $\Sigma - \Delta$ modulators can be as big as 58dB and 92dB respectively [46, pp.668–669].

The comparator output is filtered by a digital low-pass filter, the input of the filter is a single bit, therefore implementing such a filter is a simple task, it can be done by a lookup-table without the need for any multipliers. The output of the filter is a multibit representation of the input value.

2.5.4 Other ADC Implementations

There are other faster but more complicated and less area efficient ADC circuits which perform parallel ADC. In this thesis parallel architectures will not be used, however two techniques for reducing the conversion time of a given will be mentioned:

Time-interleaving ADC Is one way to achieve a fast ADC by using multiple slow ADC circuits. In that way M ADC converters demanding N clock cycles to perform a conversion can be used, every ADC starts the conversion at N/M intervals, thus N clock cycles later it outputs a digital word. However the overall output of the M ADCs is produced every N/M clock cycles. Shorter conversion time is consequently achieved.

Pipeline ADC In that approach more stages with smaller resolution (bits per stage) can be used to increase the frequency of operation, because for serial ADC the conversion time is proportional to the accuracy of the converter.

2.6 Elementary Signal Processing Functions

As was shown in the previous sections an integrator, a scaler and a comparator are all that is needed to construct Filters, ADCs and DACs. Therefore even though it is possible to realise a system which performs any mathematical operation to the input signal, in practice only a few functions are actually used in most "real-world" signal processing systems. This is very important for realising analogue electronic systems, because such circuits have by definition a limited range of capabilities to emulate the behaviour of a signal processing algorithm.

Scaling Scaling is the function mostly used by practical signal processing circuits. We define *scaling* as the product of an input signal with a constant number

which we call K factor:

$$Y_{scale}(t) = K \cdot X_{in}(t)$$

The constant number K could be varying in an adaptable system. If this is the case the frequency with which K changes must be significantly smaller than the maximum frequency of the input signal (F_{max}) . Therefore we can assume the K factor constant over a short period of operation.

Integration Integration in time of a signal is expressed by the following equation:

$$Y_{int} = \int X_{in}(t) dt$$

Multiplication of two Signals We define *multiplication* of two signals a different function than *scaling*. Though *scaling* is an essential function in virtually every system, *multiplication* is rarely used. The product of two signals is expressed by the following equation:

$$Y_{mult}(t) = X_1 \cdot X_2(t)$$

Comparison The function of a comparator is given by the equation:

$$Y_{comp}(t) = \begin{cases} 1, & X_{plus}(t) - X_{minus} > 0 \\ 0, & X_{plus}(t) - X_{minus} < 0 \end{cases}$$

The comparator is the most important system for doing any analogue to digital conversion.

Discussion

The integrator and scaling systems are very important for filtering and analogue signal processing in general. It is these circuits that we will implement using pulse based techniques, though we will demonstrate techniques for doing multiplication and comparison during this process. It is obvious that by the use of an integrator it is possible to realise many alternative operations in a sampled data system. By integrating only two continuous samples we implement an *adder-subtracter* or a *differator*. Furthermore if we integrate only one sample we realise an elementary memory cell or a *delay* (z^{-1}) .

2.7 Integrator and Scaler Implementations

In this section we will present conventional integrators and scalers. A continuoustime OTA-C integrator, a Switched Capacitor (SC) and a Switch Current (SI) integrator will be analysed, in order to highlight the inherent limitations of those implementations, which led to the introduction of pulse-based circuits.

2.7.1 Continuous-Time Integrators



Figure 2–12: OTA-C integrators: block diagram and circuit details.

Though L-C components were used in early signal processing analogue circuits, such as electronic filters, they are sensitive to variations in the component values and suffer from power loss between source and load [47,43,44]. For this reason the implementation of LC circuits by the use of active-RC components, which emulate the function of the inductor and the capacitor was introduced. The first analogue systems implemented in silicon were undoubtedly active RC circuits based on operational amplifiers to implement the *scaler* and the *integrator*. These circuits though still much in use today, suffer from drift and the uncertainty of the absolute capacitance and resistance values in VLSI; thus special processing options are needed to implement integrated active RC integrators-scalers. At present other active integrators circuits, are used for building continuous-time integrators and scalers, based on *Operational Transconductance Amplifier* (OTAs—figure 2–12) cells [48].

The basic building block is the OTA which is a Linear Voltage Controlled Current Source (VCCS). By adding a capacitor C at the output it is possible to realise an integrator with a controlled gain. Take the circuit shown in figure 2-12, if M_1 - M_2 are matched ($\beta_1 = \beta_2 = \beta$ and $V_{T1} = V_{T2} = V_T$) and the bulk effect is neglected, then the drain currents I_{d_1} and I_{d_2} are given by the equations:

$$i_{D_1} = \frac{\beta}{2} \cdot (\nu_{GS_1} - V_T)^2, \quad i_{D_2} = \frac{\beta}{2} \cdot (\nu_{GS_2} - V_T)^2$$
 (2.8)

where $\nu_{GS_i} = \nu_{IN_i} - V_s$. Note that $I_s = i_{D_1} + i_{D_2}$. The two currents can be also expressed as a function of:

$$i_{D_1} = I_D + i_d = \frac{I_s}{2} + \sqrt{\frac{\beta I_s}{4}} \Delta \nu_p \sqrt{1 - \frac{\beta}{4I_s} (\Delta \nu_m)^2}$$
$$i_{D_2} = I_D - i_d = \frac{I_s}{2} - \sqrt{\frac{\beta I_s}{4}} \Delta \nu_m \sqrt{1 - \frac{\beta}{4I_s} (\Delta \nu_m)^2}$$

Therefore

$$i_{OUT} = i_{D_1} - i_{D_2} = 2i_d \simeq \sqrt{\beta I_s} \Delta \nu_{IN} = G_m \Delta \nu_{IN}$$
(2.9)

where $\nu_{IN} = \nu_p - \nu_m$. It is obvious from (2.9) that the OTA circuit is a VCCS. When the output of (2.9) is integrated in time through the capacitor C

$$\nu_{out} = \frac{1}{C} \int i_{OUT} \, dt = \frac{G_m}{C} \int \Delta \nu_{IN} \, dt$$

Even though this circuit still suffers from the need to know the exact value of the capacitance C, it is possible to trimmer a ladder filter to the exact gain values we require, because the gain is electronically controllable.

Bipolar OTA filter circuits have reached frequencies of operation as high as 100MHz [48]. Yet OTA-C circuits are difficult to implement, interconnect and it is possible to alter the gain only over a limited range [49]. Usually such circuits are tailored for a specific application in the high frequency range.

2.7.2 Switched-Capacitor Integrators

The SC technique is based on the principle that a periodically switched capacitor can behave as a resistor, as long as the sampling frequency is 7 to 10 times faster than the maximum signal frequency. The idea of implementing a resistor with a switched capacitor was first published by J. Maxwell in 1891 [50] and a few circuits using SC techniques were designed in the thirties. Yet it was in mid sixties when researchers realised the importance of the SC circuits and gave to the technique its name [47,51].



Figure 2–13: SC resistance simulation.

The principle of SC circuits is shown in figure 2–13. During the phase Φ_1 of the non overlapping clocks (Φ_1, Φ_2) the capacitor C is charged to $Q_{in} = CV_1$. During Φ_2 the charge which is flowing at the output is $Q_{out} = -CV_2$. Therefore the flow of the charge is given by the equation $Q = C(V_1 - V_2)$ [50,44,46]. If this event takes place f_s times per second the current which flows through the capacitor is ∇_{abc} .

$$I_{SC} = f_s(V_1 - V_2) \Rightarrow R_{eq} = \frac{V_1 - V_2}{I_{SC}} = \frac{1}{f_s C}$$

SC is a well established respectable technique, much used by industry. Since the first practical implementation of SC circuits in 1967 more than 30 years of improvements and research have enhanced the capabilities of the initial circuits [52, 53].



Figure 2–14: Switched Capacitor Integrator.

A basic SC integrator is based on the idea of simulating a resistor R_{eq} with a capacitor switched at f_s is shown in figure 2-14. Consider the timing diagram of figure 2-13; the voltage $V_{out_{n-1}}$ at the moment t = (n-1)T is $-V_{C2}$ when Φ_1 is high, at the same time the capacitor C_1 is charged to the input $V_{in_{n-1}}$. During the next Φ_2 clock at $t = (n - \frac{1}{2})T$ the two capacitors are connected in series. In this case the voltage difference in V_{C1} is zero (because of the OPAMP virtual ground), therefore C_1 is discharged. The current which discharges C_1 comes through C_2 (because of the OPAMP infinite input resistance), therefore the charge $\Delta Q = C_1 V_{in}(z)$ is transferred to C_2 . By expressing that in the z domain we get:

$$H(z) = \frac{V_{out}(z)}{V_{in}(z)} = \frac{C_1}{C_2} \frac{z^{-1}}{1 - z^{-1}}$$
(2.10)

it is obvious that the circuit of figure 2–14 is an non inverting lossy integrator. In order to obtain an inverting SC integrator we must exchange Φ_1 and Φ_2 as it is indicated in figure 2–14, in that case

$$H(z) = -\frac{C_1}{C_2} \cdot \frac{1}{1 - z^{-1}}$$
(2.11)

The development of switched capacitor circuits (triggered by [54]) was ideally suited to MOS technology using double polysilicon linear capacitors. The advantages of this technique are summarised here

• The gain is controllable by an accurately controlled capacitor ratio. rather than by absolute values which have big variations in VLSI implementations.

- It is suitable to MOS implementation. Because it uses three components: capacitors, switches and OPAMPs which are easily implemented in a double polysilicon MOS process. Furthermore there is no need for special processing options required for the implementation of accurate big value resistors.
- It is possible to operate over a big range of frequencies, by just changing the sampling frequency f_s .
- They are smaller in size (because the capacitors are small) compared to active RC circuits. Therefore it is cheaper to produce SC filters.
- The same design process for both LC or active RC ladder filters can be used thus avoiding the need to redesign from scratch.

These reasons made SC the dominant technique for implementing filters in silicon. However the market demanded even higher frequencies of operation resulting in higher levels of *clock feedthrough* noise [53]. Furthermore the dynamic range of a SC filter depends to the power supply level; the signal to noise ratio decreases at lower voltage supplies, limiting the operation of a SC integrator. The use of supply voltages as low as 1V, in low-power systems emerged in this decade, pose significant constrains to the use of SC circuits.

2.7.3 Switched Current (SI) Integrators

A new technique emerged in 1988 [55,56,57], to address the problems in analogue signal processing due to the reduction of the power supply voltage (to 3V, 2.2V or even around 1V today). This technique uses currents to represent the input values instead of voltages [57,58,47,42]. Those circuits can operate in lower voltage supplies with a big dynamic range. This area of analogue electronics is called *Current Mode* (CM) design [59,60,61,59,62].

An other definition of CM is *compressed voltage operation*. That is because internal node voltages are still used, however in this case the voltage on the capacitors is "compressed" by the use of the V-I characteristic of the MOS transistor, where the voltage at the *gate* of the transistor is a function of the square-root of the drain current. The opposite effect takes place at the output where the voltage is expanded in order to regenerate the output current (for bipolar transistors the compressing is exponential). This is the reason why CM circuits can operate with lower voltage supplies.

Here we will demonstrate the use of sampled data Switched Current (SI) cells, suitable for sampled-data signal processing [58,42,63]. The basic Track and Hold (TH) Switched Current circuit is shown in figure 2–15 [64,63,56, 65]. When the switch Φ_1 is closed the output I_{out} of the current mirror combined by the transistors M_1 , M_2 tracks the input current I_{in} . When Φ_1 opens



Figure 2–15: Elementary Track and Hold circuit.

the parasitic capacitor of the output transistor M_2 is charged to the voltage which forces $I_{out} = -I_{in0}$, where I_{in0} is the input current at the moment Φ_1 opened. The current source I_A is used to enable bipolar I/O currents, since both transistors M_1 and M_2 can only sink currents. Note also that the output current is inverted with regard to the input current.

A SI Track & Hold integrator (figure 2-16 is composed by the use of two TH circuits, the second being a PMOS equivalent to the circuit shown in figure 2-15. At the falling edge of the first clock Φ_1 , the transistor M_2 "stores" the input current plus I_f which is fed back from M_4 . At the next clock Φ_2 the new output current of the second TH circuit I_f is feed back to the input. Therefore the output of the circuit in figure 2-16 is given by the equation.

$$H(z) = \frac{I_{out}}{I_{in}} = K \frac{z^{-1}}{1 - z^{-1}}$$
(2.12)

where K is the scaling factor which depends to the ratio of the input to output transistors $K = r_1/r_5$, if $r_1 = w_1/l_1 = r_2 = r_3 = r_4$. The extra transistor M_5 is needed because the Track & Hold circuit can drive only one output, in contrast





Figure 2–16: Track & Hold integrator.

to SC circuits which have a bigger "fan-out". On the other hand currents are easier distributed and added at the input, compared to SC circuits which offer limited "fan-in" and drive capabilities [66]. Equation (2.12) is identical to the noninverting SC integrator, it is possible to generate an inverting SI integrator (by modifying the circuit of figure 2–16), therefore all the techniques used in SC filter implementations can be directly applied to SI circuits [67,48,68,49].

In the SI circuit presented here noise due to clock feedtrough problems is more critical than in SC circuits, were it generates offsets. In this case mismatch at the gate voltage of two transistors generates *Harmonic Distortion* at the output.

$$THD \propto \frac{\Delta V_1}{I(V_{GS} - V_1)}\hat{i}$$

where $V_1 = V_T + V_{cf}$ and V_{cf} is the voltage mismatch due to clock feedthrough [67, 69,63]. Compared to SC where distortion is due to OPAMP inaccuracies, it seems that SI techniques lack in accuracy.

2.8 Field Programmable Analogue Arrays

2.8.1 Reconfigurable Analogue Hardware

Early analogue computers were the first examples of programmable analogue hardware [70,71,72]. Long before the introduction of digital computers analogue systems were used for *Bode plots* [71], or for the control of electric motors and servomechanisms [70]. Some of them included programmable features and controllable error correction [71]. However the development of digital electronics rapidly diminished the use of analogue electronics to stand-alone filtering or ADC-DAC systems used in digital computer interfacing circuits.

The development of reconfigurable accurate, low-cost, rapid-prototyping analogue techniques was always desirable in the electronic market. In the past several commercial analogue IC circuits (mostly filters) were designed in such a way to enable some sort of programmability. Continuous time circuits (for example the MAX274 & MAX275 [73]) use external resistance and capacitors to implement many filter configurations. User-friendly software was developed in order to calculate the filter coefficients and map them to the circuit [73]. Switched capacitor circuits offer an even larger programmability [51], by changing the sampling frequency [74,75].

All these circuits are tailored for filtering, nevertheless some early universal analogue circuits were developed, that could be reconfigured for different applications. One of the earliest ones being the GAP-01 by Precision Monolithics (developed at the early 80s) which uses on-board analogue switches and external routing for reconfiguring the device. Digitally controlled potentiometers are another example of accurate, reconfigurable analogue devices, incorporating switches for defining the resistor ratio. The *Cellular Neural Network* (CNN) universal machine is an interesting "analogic" computer [76,77,78]. Finally the programmable Analogue Interfacing Chip (AIC) by Texas Instruments is worth mentioning, to conclude these early implementations.

2.8.2 Present FPAA Implementations

During this decade the concept of FPAAs was introduced, as a monolithic device with an array of analogue reconfigurable cells connected together by analogue local and global interconnect [79,80]. The device includes SRAM cells for storing the configuration parameters. Some high-level software tools are closely related to FPAAs; these CAD tools help the end-user to represent the analogue circuit at an abstracted level. This design is then transformed to the configuration bits which are usually shifted into the FPAA device.

It is evident that FPAAs are very similar to their digital counterparts the FP-GAs [81]. In fact the ultimate goal is to implement analogue circuits which might revolutionise analogue design as much as FPGAs revolutionised digital design. It should be noted that the *Field Programmable* factor of the FPAA implementation is critical for their commercial success. By saying that we mean that the device should be as programmable as possible. The requirement for externally connected components is in direct contradiction to the FPAA concept.

The first circuit which combined all the elements of an FPAA was performing tasks of neural-computation, by the use of transmission gates to connect the circuit resources [82]. Other papers at the beginning of this decade addressed the implementation of FPAAs [83,78]. It is possible to categorise them in three big categories SC [84,85,86] SI [87] and continuous time [88,89,90,91] implementations (sometimes low-power [89]). Most of them use transmission gates to program the local interconnect, while integrators, scalers or multipurpose analogue cells were implemented by the use of standard SC or CM techniques.

Today some commercially available products have entered the market. Starting with IMP which launched several chips [92,93] targeting automotive, data acquisition, medical instrumentation, distant measurement, reconfigurable test equipment, process control and sensor signal conditioning applications. These chips integrate some elementary building blocks such as filters, multiplexes, adders, input and output amplifiers based on SC circuits, together with programmable local interconnect. Motorola has launched its own version of a FPAA, licensed from Pilkintron Microelectronics [85]. This device is based on SC circuits and it can implement a big range of analogue signal processing tasks. A CAD tool is also available to facilitate the design of standard circuit filter, comparator, and other OPAMP based circuits.

Finally a new Analogue Signal Processor chip entered the market. It uses logdomain multipliers in order to perform multiplication [88,94]. Log-domain multipliers use logarithmic properties to achieve multiplication by adding the logarithm of the input currents (figure 6-8):

$$A \times B = e^{\ln(A) + \ln(B)}$$

where bipolar transistors are used to obtain the logarithm of the input currents, the voltage addition is done by an OPAMP based adder and the output current is derived from other bipolar transistors.

2.8.3 Field Programmable Mixed-Signal Arrays

The next step on the development of *Field Programmable* circuits is to combine analogue and digital cells on the same chip, integrated into a mixed-signal array [80]. The mixed-signal cell must combine both an FPGA and the analogue parts into a closely related entity, with analogue signals transformed to digital words and DACs regenerating analogue outputs [95]. There is a big potential market for those devices, market research firms (including ICE) indicate that mixed-signal ICs represent a 25% of \$4.0 billion digital IC market.

At the moment there is no commercially available mixed-signal IC. Nevertheless Actel has filed a patent which describes an FPGA with ADC and DAC cells [96], AMD describes two types of mixed-signal cells where an analogue front-end (or back-end) circuit is combined with a programmable logic array [97,98]. Recently *Motorola* announced a mixed-signal version of its FPAA [99]. It is believed that both FPAAs and FPMAs are in their infancy and that their progress awaits a "killer application" to boost their production, rather than being a solution to an unknown problem. Several factors indicate that progress and development will continue, namely:

- The assembly of intellectual property on field programmable analogue or mixed signal arrays, by many companies (expressed by an increasing number of patent filings).
- The fact that the FPGA market approaches saturation, therefore other alternative field programmable device markets are explored to expand the financial basis.
- The demand for fast accurate efficient prototyping of analogue circuits by companies which cannot afford the cost of custom-made analogue devices.
- The tendency in the IC market to integrate complete systems on single devices.
- The boom of the analogue market by the introduction of new small, battery operated products (mobile phones, advanced music systems, laptop computers etc.)

2.8.4 Discussion

Even though the market looks promising, conventional approaches have failed to address the problems of practical FPMA implementations, because SC or SI techniques enforce numerous restrictions on the operation of mixed-signal circuits. We believe that new sampled data circuits should be used for implementing FPMAs.

In table 2–3 a brief outline of standard SC and SI features critical to the implementation of FPMAs is given. From this comparison it is clear that SC circuits are better suited to the implementation of FPAAs, because the gain can be easily controlled by modifying the sampling frequency, though voltage signals are more sensitive to noise than currents [100].

	Switched Capacitor	Switched Current
Fan-Out	Big fan-out depending on the	Fan-Out is one per output stage
	design of the OPAMP and the ca-	therefore n output stages are
	pacitance of the load.	needed for n outputs.
Fan-In	A switched capacitor circuit is	Big fan-in since it is very easy to
	needed per input therefore the	add currents at the input node
	fan-in is one per input stage.	(subject to reaching the limits of
		operation).
Intercon-	Limited because of coupling	Better because currents are not
nection	between analogue and digital	so sensitive to noise and because
	lines, because of the line capacit-	it is easier to switch currents.
	ance, and because of the use of	
	analogue switches.	
Gain	Can be easily controlled by chan-	Because of the need to control
Control	ging the sampling frequency	the ratio of transistors, the cir-
	88 I 8 I V	cuit needed for programmable
		gain is big and clumsy.
Clock-	Bad, because clock-feedthrough	Very bad, clock feedtrough ef-
Feed-	effects corrupt the input voltage	fects give high order effects res-
through	samples resulting in voltage	ulting in harmonic distortion (as
noise	offsets	well as offsets).
Features	Sampled analogue cells, Ana-	Sampled analogue cells, Ana-
	logue I/O (voltage), digitally	logue I/O (current), transistor
	controlled gain	W/L controlled gain

Table 2–3. SC and SI characteristics.

Present SC based FPMA circuits integrate some analogue FPAA and digital FPGA cells on the same chip [99]. There is limited interconnection between the analogue and the digital cells. This is due to the fact that voltage domain analogue values are very sensitive to noise coupled from the digital signals. In these circuits extra A/D and D/A circuits are needed per cell, in order to achieve such a closely related entity with feedback from both the analogue and the digital cells [96, 97,98,95]. In that way the mixed-signal virtues of the FPMA circuit could be used to achieve an optimum operation of the overall device. For example some feedback from the analogue cells could be used in a negative feedback loop or an adaptable algorithm implemented by the digital cells of a FPMA device. However the analogue and digital interfacing cells would have to be of a limited accuracy, in order to reduce the size needed by the converters and integrate many cells on a single chip [95,97].

Finally the software which is used to translate the analogue schematic design to the FPMA configuration bit series, should be friendly to the user based on a straight-forward window environment, without too many constraints which would be difficult to remember. This is particularly difficult in SC circuits, since many constraints are imposed by the limited interconnect and the drive capability of the OPAMPS. Indeed the software development cost is frequently as large as the design cost of the device.

In our first attempts to understand these constraints and adopt some approach to FPMA implementation, we realised that standard SC [52] or SI [58,67,42] cells which have been used in industry, in other FPAA implementations, are not suited to the implementation of FPMAs [53,68]. A different signalling technique should be used in order to exploit the mixed-signal virtues of the cells. Because of the artificial neural information background [5,7,6,101] which was available in our group, we realised that an alternative method for representing the analogue inputs by the use of pulses, would be better suited for the FPMA implementation [29].

In that way all the inputs to an analogue cell are digital signals, modulating analogue values in time. These inputs are used to charge the integrating capacitors, the output derives from a comparator which compares the voltage on the integrating capacitor to a ramp [29,31,30]. Therefore every cell includes an A/D and D/A converter, in consequence all the cell I/O (apart from a few global reference or bias voltages) is digital, while all the internal manipulation is analogue. The signals can be distributed by conventional FPGA cells, while some of the digital functionality can be used in a mixed-signal implementation [11,32]. In that way there is no need to develop new FPMA software, it is possible to use commercially available software, by extending the FPGA libraries to include analogue cell interconnect.



2.8.5 Pulse-Based circuit outline

Figure 2–17: Pulse-Based FPMA implementation A) Device Array B) Analogue circuit block diagram C) Integrator details.

In this section we will briefly introduce and analyse the operation of a typical pulse-based FPMA implementation; more detailed discussion will appear in the following chapters. In a typical pulse-based FPMA (figure 2–17A) an array of analogue and digital programmable blocks is integrated on the same chip (in our designs the digital cells will be provided by an external FPGA). Digital lines are used for the interconnection of both analogue and digital cells, the analogue

values are represented by pulses, which encode in time the analogue information, by modulating a digital pulse.

Figure 2-17B shows the block diagram of a typical analogue cell. It has one or multiple PWM inputs, those inputs are integrated in time, the resultant integrated value is scaled by a constant number (K) and the overall value is compared to a ramp signal. The output of the comparator is a PWM signal, which can be directed to other analogue cells (usually some extra processing is performed at the PWM output signal—in our circuits a XOR gate is used for that.) The typical analogue cell has also some digital control lines which control the gain (K) and configure the internal circuit.

Figure 2-17C shows the implementation details of the our typical integrator and scaler. The PWM input signals (figure 2–17B) are initially processed by some digital logic blocks (external to our circuits). Those logic blocks multiplex in time the inputs (or multiply the K factor to an integer value). The two resultant signals $(\xi_+ \text{ and } \xi_-)$ control the two switches (shown in figure 2–17C), which either dump or remove some charge on the integrating capacitor (C_{int}) . The overall K factor can be controlled electronically, by changing the integrating current (I_{int}) and the capacitor value (C_{int}) .

Analogue arrays like the one analysed above, will be used throughout this thesis, in conjunction to standard digital programmable logic, to produce analogue sampled-data, mixed-signal and ADC-DAC systems. A lossy integrator, such as the one described above (figure 2–17C), needs to be sampled at frequencies extending over ten times the maximum input signal frequency (in our CMOS circuits the maximum sampling frequency was about 1MHz while in the BiCMOS implementations it can reach 20MHz). Alternatively an exact design technique or a mixed-signal algorithm can be used; this enables sampling at the Nyquist frequency ($f_N = f_s = 2 \cdot f_{max}$).

Chapter 3

Palmo Techniques: The pulse based approach

3.1 The use of pulses in analogue systems

In our approach to the design of Field Programmable Mixed Signal arrays, we realised that novel ideas should be used in order to interface an analogue FPAA to a digital FPGA. Our approach involved the use of digital pulses, to encode analogue input values in time [29]. Though this is a novel approach to signal processing it has already been used in other analogue and mixed-signal implementations [102].



Figure 3-1: Stochastic signal Multiplication and Addition.

Stochastic computers, which were proposed during the seventies [71, Chapter 20], is one example of using pulsed signals, in order to implement mixed signal systems. Stochastic computers incorporated the use of digital I/O with analogue circuits for the generation of uncorrelated outputs. The principal behind stochastic

arithmetic is that if two pulse signal sources are statistically uncorrelated, then it is possible to realise a *multiplier* and an *adder* by the use of an *AND* and an *OR* gate respectively (figure 3-1). Take for example the uncorrelated signals (*A* and *B*) of figure 3-1. In stochastic systems the pulsed signals series represents the analogue value of the probability that the signal is high during a fixed sample period T_s . A signal represented by the maximum frequency of pulses represents the value of "1.0". Therefore in figure 3-1 A = 0.5 and B = 0.1. By ANDing the two signals we obtain an output of $A \cdot B \simeq 0.05$, while the output of the OR gate is $A + B \simeq 0.6$. It is clear that the signals used in stochastic arithmetic should be uncorrelated and random, otherwise the output of the AND gate would be 0.1 and that of the OR gate 0.5

Stochastic computers use compact circuits for performing arithmetic functions. Nevertheless the complexity of the circuits needed to generate uncorrelated outputs [103] and the big number of pulses needed to minimise random effects and generate accurate results, posed a considerable limitation to the use of stochastic computers and especially to their maximum frequency of operation.

Most Analogue to Digital converters are another example of analogue pulsebased systems. In Pulse Width and Pulsed Frequency Modulators the analogue input (usually a voltage) is compared to a ramp and the width or the frequency of the resultant pulses define the magnitude of the input [46,43,44]. $\Sigma - \Delta$ circuits is another example of A/D conversion, where the frequency of the resultant pulse stream is used by the digital circuit to measure an analogue input value with a big dynamic range (section 2.5.3). Take the first order $\Sigma - \Delta$ converter of figure 2–11: if the input value of $x_{in}(t)$ is 1/5 and the initial condition of the integrator is zero, then the output of the comparator for the first 20 cycles samples will be: "0", "0", "1", "0", "0", "0", "0", "1", "0", "0", "0", "0", "1", "0", "0", "0", "0", "1", "0", "0". The average of the resultant pulse series approaches 1/5.

Finally we will mention the field of pulse-based artificial neural networks, because this was the source of our inspiration for the implementation of pulsebased integrators and scalers for analogue and mixed-signals field programmable devices [5,7,104,102]. Neural network implementations use pulse-based signals to generate the output (y) of the function

$$y = f(\sum_{k=1}^N w_k \cdot x_k)$$

where x_k are the inputs, w_k are the synapse weights and f(x) is the neuron activation function [5]. In analogue implementations the inputs are pulses and they define which current sources $(I_k = w_k)$ will charge a capacitor for the duration ΔT_{x_k} (figure 3-2). The charge which is accumulated on the capacitor over a cycle



Figure 3–2: Pulse-Stream Neural Networks.

defines the sum of the input products $w_k \cdot x_k$. This voltage is then compared to a ramp in order to generate the pulsed output [7,6]. By changing the ramp to a non-linear one it is possible to realise non-linear activation functions f(x) [7]. Other neural network implementations incorporate stochastic pulse series [9,10, 103], in order to implement circuits similar to the stochastic computers mentioned before. In fact almost every modern artificial neural network implementation uses some form of pulsed input/output scheme [102].

The biggest disadvantage of pulse-based systems on the other hand comes from the nature of their signal representation. The information is encoded in time, therefore the faster the circuit operates the smaller the signal to noise ratio becomes. Though this is true for any analogue design, in the pulse-based case the increase of the operating frequency significantly reduces the dynamic range of the output [102]. It is estimated that a pulse-based system will be two to four times slower than an SC or SI circuit for a given process. Nevertheless new circuit design ideas for a sampled data log-domain integrator, which will be presented in chapter 6, are ideally suited to the pulsed signalling mechanism, combining the advantages of our circuits with a sample frequency much higher than conventional SC or SI techniques.

3.1.1 Advantages of pulsed systems for the implementation of FPMAs

Pulse-based signals offer distinct advantages over conventional voltage or current mode circuits for the implementation of Field Programmable hardware.

- Compact A/D and D/A converters The complexity of A/D and D/A converters in a pulse-based system is significantly reduced compared to a multibit conventional mixed signal circuit, because the signal is coded by the use of only one digital line. This makes the use of pulsed systems ideal for the implementation of compact mixed signal circuits with two converters per cell.
- Simple Cells The circuits used to implement the analogue part of the mixed signal array are simple to design, layout and test.
- Small Cells Compared to SC or SI implementations our *Palmo* circuits are much more compact for a given resolution. Because the gain of the integrator is proportional to the product of a capacitor ratio times a current ratio, bigger programmability can be achieved in smaller area.
- Analogue and Digital Cell interface In our circuits all the I/O signals are digital and the analogue operations are performed at the main core of the cells. It is therefore possible to isolate the analogue signals from the digital control lines with guard-rings and physical distance, thus minimising the need for careful design of the interconnect topology.

- Limited Noise An added advantage of our circuits is that it is possible to realise switching techniques which do not affect the voltage on the integrating capacitors, hence reducing the clock-feedtrough noise to virtually zero.
- FPGA routing Since all the I/O signals are digital and therefore robust and easily redistributed across chips, an FPGA can be used for the routing, enabling extended programmability and a big fanout.

3.2 Typical Palmo Cell



Figure 3-3: Typical Palmo Cell.

Having realised that pulse-based signals offer significant advantages to the implementation of FPMAs, the main parameters of the pulse-based cells were introduced. Ideally these cells perform analogue operations to digital inputs and generate pulsed outputs [5,102]. It is these cells that were implemented by the use of either voltage or current mode circuits. We named this technique *Palmo* signal processing, from the hellenic word " $\Pi \alpha \lambda \mu os$ " which means pulsebeat, pulse palpitation or series of pulses, to signify the importance of the use of pulses as the signalling mechanism. The elementary functions were implemented are those of *scaling* and *integration*, since it is possible to perform most linear signal processing functions by the use of integrators and scalers, as was mentioned in the previous chapter. A typical block diagram of our circuits is shown in figure 3-3 [29,31]. The input pulses are integrated over time by the use of an analogue integrator. This is usually done by charging a capacitor with a constant current. In the current mode circuits the voltage on the capacitor is compressed because of the transfer function of the input transistors, while in the voltage domain it is linear.

This integrated value is then compared to a ramp, in order to regenerate the pulsed output. The ramp is generated from a step signal by the use of an identical integrator. This ramp can be global for the whole chip or local for every cell. The advantage of using such a ramp generation is that it is possible to accurately control the overall gain (K) of the circuit. This is equal to the ratio of the individual gains of the integrator (K_{int}) and the ramp (K_{ramp})

$$K = \frac{K_{int}}{K_{ramp}} \tag{3.1}$$

Because ratios of capacitors and currents in analogue VLSI can be accurately matched, as opposed to their absolute values which suffer from a big variation [105, 106,107] the control over the gain is improved. It is these integrators that we have implemented by the use of analogue VLSI in our *Palmo* cells.

3.3 Signal Representation

After having considered the implementation of a typical Palmo cell the signalling mechanism needs to be defined in order to design the actual circuits. There are two main pulse modulation techniques with many variations, namely *Pulse Frequency Modulation* (PFM) and *Pulse Width Modulation* (PWM) [102].

PFM is a pulse based technique used in many applications such as modems and mobile phones. According to this modulation the frequency of a pulse series represents the size of the signal. In an ideal PFM scheme the width of every pulse is the same; in practice some small variations may occur. PWM on the other hand modulates the width of a pulse depending on the input signal while the frequency of the pulsed output remains constant.

PFM is not appealing for implementing our *Palmo* signal processing cells. This is mainly due to two factors: first a varying number of pulses is needed to define a typical *Palmo* signal and in some cases it is not possible to have equal sampling intervals, which makes the use of such a PFM technique not suitable for sampleddata signal processing; furthermore because the number of pulses which are needed to represent a signal is big, the maximum frequency of operation is limited to small values. Second, a big number of input pulses will generate significant noise on the integrating capacitors, because of coupling between the input pulses and the analogue integrating cells.



Figure 3-4: Different PWM generation schemes.

PWM on the other hand is better suited to sampled-data signal processing because the sampling rate is well defined and because only one pulse represents a signal value. In addition it is possible to run the circuits with a higher sampling frequency than PFM based circuits. Having decided upon the pulse modulation technique, the actual signal representation has to be defined. There are many different ways to generate a PWM signal, depending on the ramp signal used. In figure 3-4 an integrated value *Int* is compared to a *Ramp* generating the *Out*- put. The first option 3-4A, uses a negative going double-sided ramp, the resultant output pulse being a centred pulse with a width proportional to the size of Int. Figure 3-4B shows a positive going double-sided ramp, resulting in a centred absence of pulse. The overall time which the two output pulses are high is proportional to the Int value as well. The signals in figure 3-4C,D show the PWM output of a system using a single sided ramp. In this case the pulses are not centred and in the case of the up going single sided ramp the output is inversely proportional to the Int value.

The use of double sided ramps offers the advantage of cancelling the effects of any comparator delays, as long as the up and down going delays of the comparator are matched. Furthermore, the centred pulse of figure 3–4A introduces less noise than the other pulse signalling schemes. In this scheme, the power supply variations due to the current spikes generated by the transition of the comparators, are smaller, since in most cases the pulses occur at different moments. This is in contrast to all the other signal coding schemes where the transitions occur simultaneously. However the scaling factor of the Palmo cells is controlled by the ratio



Figure 3-5: A) Scaling by the use of the Ramp, B) Saturated Ramp (double and single sided).

of the gains of the two integrators generating the Ramp and Int (equation 3.5). In other words scaling can be achieved by controlling the slope of the ramp (figure 3-5A). In order to exploit the full dynamic range of the circuit we would like the integrator, which integrates the input pulses, to reach the maximum available values. In that case the ramp would saturate at the maximum value defined by the supply voltage. In order to generate a double sided ramp a delay is needed (figure 3-5). The circuit for making this delay proportional to the slope of the ramp is very complicated while any timing mismatches will generate harmonic distortion at the output. Therefore we used a single sided ramp in our Palmo implementations.

Finally a sign-magnitude coding is used in our circuits [29]. The magnitude of the signal is represented by the duration of the pulse, while the sign is determined by whether the pulse occurred in the positive or negative cycle of a global *sign* clock (figure 3–6).

Therefore a positive signal of a value 'A' is



Figure 3–6: The *Palmo* signalling mechanism.

represented by a pulse which is 'high' for ΔT_{pulse} during the positive cycle of the sign clock, and a negative input value of '-A' by a pulse which is 'high' for ΔT_{pulse} during the negative cycle of the sign clock. The sign is defined by a global sign clock and the magnitude by the width of the pulse.

This representation has the advantages that without initiating any significant delay, a zero signal value results in the absence of any pulses (in either the 'high' or the 'low' period of the sign clock) and that, apart from the global *sign* clock, there is only one data line for each signed pulse signal. This reduces the amount of interconnect required.



Figure 3–7: Palmo typical voltage domain Cell.

3.4 Palmo Voltage Domain Implementation

Consider the circuit of Figure 3-7. The non-inverting and inverting signed-PWM inputs (*plus* and *minus* respectively) of a typical palmo cell, are directed by external digital logic to either the ξ_+ or the ξ_- switch depending on the *sign* bit and other control signals. If ξ_+ is closed then charge is dumped onto the capacitor C_{int} for the duration of the pulse ΔT . If the ξ_- switch is closed, charge is removed from C_{int} .

The charge accumulated on the integrating capacitor V_{Cint} is then compared with a ramp (sixth trace in figure 3-7 — V_{Cr}). The switches (ξ_+ and ξ_-) of the ramp generating circuit are driven directly by the signals *Rup* and *Rdown*, which are usually globally generated. The comparison of V_{int} and V_{Cr} results in a pulse signal, which encodes the PWM information. However an exclusive OR (XOR) gate should be used at the output, to ensure the regeneration of the signed pulse representation described earlier.

By allowing signals arriving at the *plus* and *minus* inputs to be continuously

integrated, the resultant output signal, *out*, is a scaled, pulse width modulated representation of the integrated signal.

3.4.1 Palmo Miller Integrator

In a Miller Integrator the input at the *plus* node is delayed by one clock period to the output. The input at the *minus* is inverted and delayed by one-half clock period to the output. This initiates the need for a delay clock (*B*). The function of the proposed *Palmo* analogue cell (figure 3–7) is defined by the digital logic block which drives the ξ_+ and ξ_- switches. The appropriate digital logic to generate the signals ξ_+ and ξ_- in order to implement a Miller integrator are given by the following equations

$$\xi_{+} = P \cdot S \cdot \overline{B} + M \cdot \overline{S} \cdot B$$
$$\xi_{-} = P \cdot \overline{S} \cdot \overline{B} + M \cdot S \cdot B$$

where S is the sign clock, P is the plust input, M is the minus input, and B is produced by the delay.

By the use of this digital logic the charge accumulated on the integrating capacitor during one cycle (C_{int}) is given by equation 3.2.

$$\Delta Q_{Cint}(z) = I_{int} \left(\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2} \right)$$
(3.2)

The voltage (V_{Cint}) on the integrating capacitor at the end of an integrating cycle is given by the following equations:

$$V_{Cint}(z) = V_{Cint} z^{-1} + \frac{\Delta Q_{Cint}(z)}{C_{int}}$$
$$= \frac{I_{Cint}}{C_{int}} \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}}$$

The voltage (V_{Cint}) accumulated on the integrating capacitor (figure 3-7) is compared with the ramp (V_{Cr}) in order to regenerate the pulsed output. When the voltage V_{Cr} becomes equal to the voltage on C_{int} , the comparator output will change state, defining the end of the pulse-width output. In this time the voltage on the ramp capacitor is

$$V_{Cr}(t) = \frac{I_r}{C_r} \Delta T_{out}$$



and $V_{Cint} = V_{Cr}$. thus:

$$\Delta T_{out} = \frac{C_r I_{int}}{C_{int} I_r} \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}}$$
(3.3)

In the well established S–C active RC filter implementation a switched-capacitor replaces R [52,44,42,47]. The transfer function of the S–C Miller integrator (figure 2–14) with its output sampled on Φ is:

$$V_{out}(z) = \frac{C_u}{C_I} \left(\frac{V_1 z^{-1} - V_2 z^{-1/2}}{1 - z^{-1}} \right)$$
(3.4)

It is noticeable from equations 3.3 and 3.4 that the proposed *Palmo* basic building block and the S–C Miller integrator have identical transfer functions with:

$$K = \frac{C_u}{C_I} = \frac{C_r}{C_{int}} \cdot \frac{I_{int}}{I_r}$$
(3.5)

Because of this similarity existing S–C synthesis techniques and tools can be applied to the *Palmo* realisation [29,31]. On the other hand it is very important to note that scaling (K) is a function of the ratio of two capacitances multiplied by the ratio of two currents, resulting in greater dynamic range of filter coefficients, compared to conventional S–C (or S–I) techniques [53,67,57]. Since the ratio of capacitors in equation (3.5), can be modified by switching between the elements of a capacitor array, and the ratio of the currents can be electrically modified, with sufficient accuracy [108,106,109,69]; it is realised that the scale factor K is fully programmable and insensitive to absolute values [110,105,111][8, Appendix C].

3.4.2 Charge-Injection

Charge injection is a very well known phenomenon in which some part of the charge stored in the channel of a MOS transistor used as a switch, is discharged on the associated capacitors [100,42] (figure 3–8A). This effect in conjunction with coupling from the control clock (Φ), known as clock feedthrough [42], is responsible for errors in the integrated voltage. It also causes offsets and poses limitations to the overall circuit performance.

This effect is proportional to the capacitor ratio C_{Switch}/C , the control clock slope and voltage swing. There is a way of cancelling charge injection, by generating an accurate complementary clock ($\bar{\Phi}$, figure 3–8B) [100,42]. However, because



Figure 3–8: MOS switch implementations with reduced clock-feedthrough.

in practice it is very difficult to control the slope and the delays of digital logic cells, in order to generate identical inverted clocks, precise cancellation of the charge is impossible to achieve.

This discussion makes evident that the use of switches to control the current of the current source I_{int} in figure 3-7 will introduce some charge injection on the integrating capacitor.

3.4.3 Current Source Design



Figure 3–9: Current source circuit

As seen from Figure 3-7, the circuit building blocks required to implement the *Palmo* filter tap are simple and common-place. The critical structure in the implementation is the charge dump/remove circuit.
By using a standard switching arrangement we would require relatively large currents (10's of μA) and careful consideration of switching noise. In conventional techniques, mentioned in the previous section, a current from a current source is switched on and off using a transistor (or arrangement of transistors) as a switch [42,57,56]. During the switching transition a large voltage swing results in *charge injection* into the data-holding capacitor thus corrupting the data.

The circuit techniques used here [104,31] overcome this problem. Instead of switching the current from the current source on and off, this circuit switches the actual current source on and off. This virtually eliminates charge injection. The details of this are shown in Figure 3–9. When ξ is high, transistor M1 is on, while M2 is off. This enables the voltage established on the gate of M3 by current Ir/Iint to be transferred to the gate of transistor M4, thus discharging the capacitor with a constant current Ir/Iint. When ξ is low, transistor M1 is off, while M2 is on. The voltage on the gate of M4 is now Vss, switching the current source off. This virtually eliminates charge injection as shown in simulations and verified by chip results. The circuit shown in figure 3–9 was implemented, incorporating standard transistor inverters. The current sources were biased with a small gate-source voltage (V_{gs}) in order to source a small current of 5nA; yet no switching noise was discernible on a 1pF capacitor.

3.5 Harmonic Distortion

A very important parameter of a signal processing system is the Total Harmonic Distortion (THD). Because it offers a way to calculate the dynamic range within which the THD of the circuit is below a maximum acceptable level. In our voltage domain circuits the causes of harmonic distortion are mainly two [33]. These are delays to the output pulse and differences between the charging and discharging currents.

The non-ideal effects of the input-offset voltage of a typical comparator would generate offsets at the output of the *Palmo* circuit. While propagation delays could result in Harmonic Distortion.



Figure 3-10: Harmonic Distortion due to comparator delays A) Comparator inverting stage B) cos(x) C) Minimum pulse effect f(x) D) Output distorted signal (signals reconstructed for clarity).

Comparator delays are due to the parasitic capacitances C_{G_p} and C_L which need to be charged (or discharged) when the comparator changes stage (figure 3– 10A). Because of our *signed PWM* signal representation and the use of a single sided ramp, these inevitable delays will introduce a ΔT_{ν} at the PWM output. If a double sided ramp was used, then this delay (ΔT_{ν}) will be the difference between raise and fall times of the comparator. Therefore the ideal cosine output (figure 3–10B) with a magnitude ΔT_{μ} will be distorted, because of the signal f(x)(figure 3–10C), resulting in the signal shown in figure 3–10D, which is given by the equation:

$$y(t) = \Delta T_{\mu} \cos(\omega t) + f(t) \tag{3.6}$$

Where ΔT_{μ} is the magnitude of the desired cosine output¹, and f(x) a step function given by the equation

$$f(\omega t) = \begin{cases} \Delta T_{\nu}, & -\frac{\pi}{2} \le \omega t + 2\kappa \pi \le \frac{\pi}{2}, \text{ where } \kappa \in \mathcal{Z} \\ -\Delta T_{\nu}, & \text{otherwise} \end{cases}$$

¹Our systems are sampled-data ones, therefore the output is discrete, however for simplicity, we will consider them continuous during this THD analysis.

The Fourier cosine series of the output signal (3.6) (figure 3-10D) is equal to the sum of the Fourier series of the two signals shown in figures 3-10B and 3-10C namely $\Delta T_{\mu} \cos(\omega t)$ and f(t), because the signal is an even function of ωt the resultant Fourier series is a Fourier cosine series thus

$$\mathcal{F} \{ y(t) \} = \mathcal{F} \{ \Delta T_{\mu} \cos(\omega t) \} + \mathcal{F} \{ \Delta T_{\nu} f(t) \} \iff$$
$$\Delta T_{out} = \Delta T_{\mu} \cos(x) + (\alpha_0 + \alpha_1 \cos(x) + \alpha_2 \cos(2x) + ...)$$

Where $x = \omega t$,

$$\begin{aligned} \alpha_0 &= \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} f(x) \, dx \\ &= \frac{\Delta T_{\nu}}{\pi} \left(\int_0^{\frac{\pi}{2}} 1 \, dx + \int_{\frac{\pi}{2}}^{\pi} (-1) \, dx \right) \\ &= 0 \end{aligned}$$

And

$$\begin{aligned} \alpha_n &= \frac{2}{\pi} \int_0^{\pi} f(x) \cos(nx) \, dx = \\ &= \frac{2}{\pi} \left[\int_0^{\frac{\pi}{2}} \Delta T_{\nu} \cos(nx) \, dx + \int_{\frac{\pi}{2}}^{\pi} (-\Delta T_{\nu}) \cos(nx) \, dx \right] \\ &= \frac{2\Delta T_{\nu}}{\pi n} \left[\sin(nx) \mid_0^{\frac{\pi}{2}} - \sin(nx) \mid_{\frac{\pi}{2}}^{\pi} \right] \\ &= \begin{cases} 0, & n = 2k \ \forall k \in \mathcal{N} \\ (-1)^{(n-1)/2} \cdot \frac{4 \cdot \Delta T_{\nu}}{\pi \cdot n}, & n = 2k + 1 \ \forall k \in \mathcal{N} \end{cases} \end{aligned}$$

The Harmonic Distortion, due to the propagation delay of the comparator (HD_i) , for $\Delta T_{\nu}/\Delta T_{\mu} \ll 1$, is given by the equation:

$$HD_i \simeq \frac{\sqrt{\alpha_3^2 + \alpha_5^2 + \alpha_7^2 + \dots}}{\Delta T_\mu + \alpha_1} = \frac{\sqrt{(1/3)^2 + (1/5)^5 + (1/7)^2 + \dots}}{\frac{\pi \Delta T_\mu}{4 \Delta T_\nu} + 1}$$
(3.7)

Where ΔT_{μ} is the magnitude of the signed PWM cosine input and ΔT_{ν} is the size of the comparator delay.

As shown in equation (3.3) capacitor or current mismatches generate K mismatches or offsets. However these mismatches do not generate distortion. On the other hand differences between the charging and discharging currents I_p and I_n in figure 3-11, will generate harmonic distortion; because of the different magnitudes



Figure 3-11: Harmonic distortion due to current differences.

 ΔT_p and ΔT_n at the output g(x) which is given by the equation:

$$g(\omega t) = \begin{cases} \Delta T_p \cos(\omega t), & -\frac{\pi}{2} \le \omega t + 2\kappa \pi \le \frac{\pi}{2}, & where \ \kappa \in \mathcal{Z} \\ \Delta T_n \cos(\omega t), & otherwise \end{cases}$$
(3.8)

This effect will generate a second THD component given by the *Fourier* transformation of (3.8), resulting the following cosine series:

$$\mathcal{F}\{g(t)\} = \alpha_0 + \alpha_1 \cos(x) + \alpha_2 \cos(2x) + \alpha_3 \cos(3x) + \dots$$

where $x = \omega t$ and the Fourier coefficients $\alpha_0, \alpha_1, \ldots$ are given by the equations:

$$\begin{aligned} \alpha_0 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} g(x) \, dx \\ &= \frac{1}{\pi} \left(\Delta T_p \int_0^{\frac{\pi}{2}} \cos(x) \, dx + \Delta T_n \int_{\frac{\pi}{2}}^{\pi} \cos(x) \, dx \right) \\ &= \frac{\Delta T_p - \Delta T_n}{\pi} \end{aligned}$$

Also

$$\alpha_1 = \frac{2}{\pi} \int_0^{\pi} g(x) \cos(x) dx$$

= $\frac{2}{\pi} \left(\Delta T_p \int_0^{\frac{\pi}{2}} \cos^2(x) dx + \Delta T_n \int_{\frac{\pi}{2}}^{\pi} \cos^2(x) dx \right)$
= $\Delta T_p/2 + \Delta T_n/2$

And

$$\alpha_n = \frac{2}{\pi} \int_0^{\pi} g(x) \cos(nx) dx$$
$$= \frac{2}{\pi} \left(\Delta T_p \int_0^{\frac{\pi}{2}} \cos(x) \cos(nx) dx + \Delta T_n \int_{\frac{\pi}{2}}^{\pi} \cos(x) \cos(nx) dx \right)$$

Because

$$\int \cos(x) \cos(nx) \, dx = \frac{\sin\left((n+1)x\right)}{2(n+1)} + \frac{\sin\left((n-1)x\right)}{2(n-1)} \quad n > 1$$

 α_n is derived:

$$\alpha_{n} = \frac{2}{\pi} \left[\Delta T_{p} \cdot \frac{\sin\left((n+1)x\right)}{2(n+1)} \Big|_{0}^{\frac{\pi}{2}} + \Delta T_{p} \cdot \frac{\sin\left((n-1)x\right)}{2(n-1)} \Big|_{0}^{\frac{\pi}{2}} + \Delta T_{n} \cdot \frac{\sin\left((n+1)x\right)}{2(n-1)} \Big|_{0}^{\frac{\pi}{2}} \right] \leftrightarrow \alpha_{n} = \begin{cases} 0, & n = 2k+1 \quad \forall k \in \mathcal{N} \\ \frac{2}{(2k+1)(2k-1)\pi} (\Delta T_{p} - \Delta T_{n})(-1)^{k+1}, & n = 2k \quad \forall k \in \mathcal{N} \end{cases}$$

Therefore if $2 \cdot |\Delta T_p - \Delta T_n| / (\Delta T_p + \Delta T_n) \ll 1$

$$HD_{ii} \simeq 4 \cdot \frac{\sqrt{(\frac{1}{3})^2 + (\frac{1}{15})^2 + (\frac{1}{35})^2 + \dots}}{\frac{\Delta T_n + \Delta T_p}{|\Delta T_n - \Delta T_p|}}$$
(3.9)

Where ΔT_p , ΔT_n are the currents driven by the transistors Mn, Mp. For small values of HD_i and HD_{ii} the Total Harmonic Distortion of the *Palmo* circuit (THD) can be approximated by adding (3.7) and (3.9), thus $THD = HD_i + HD_{ii}$.

For improved accuracy the THD can be calculated from the following equation:

$$THD = \frac{\sqrt{HD_{ii2}^2 + HD_{i3}^2 + HD_{ii4}^2 + HD_{i5}^2 + \dots}}{\sqrt{P_{AC}}}$$

for $\Delta T_{\nu}/\Delta T_{\mu} \ll 1$ and $2 \cdot |\Delta T_p - \Delta T_n|/(\Delta T_p + \Delta T_n) \ll 1$ the THD can be approximated by the equation:

$$THD \simeq \frac{\sqrt{\left(\left(\frac{1}{3}\right)^2 + \left(\frac{1}{5}\right)^2\right) \left(\frac{4}{\pi} \frac{\Delta T_{\nu}}{\Delta T_{\mu}}\right)^2 + \left(\left(\frac{1}{3}\right)^2 + \left(\frac{1}{15}\right)^2\right) \left(\frac{4 \cdot |\Delta T_p - \Delta T_n|}{\Delta T_p + \Delta T_n}\right)^2}{\sqrt{2}}}{\sqrt{2}}$$
(3.10)

Which is in fact much smaller than the approximated one given by:

$$THD = HD_i + HD_{ii}$$

The comparator delays ΔT_{ν} are independent (to a first approximation) to the frequency of operation, however its size increases proportionally to the output (ΔT_{μ}) at high sampling frequencies. Current matching on the other hand depends heavily to the size of the currents. Therefore at different operating frequencies,

÷



Figure 3–12: Total Harmonic Distortion.

different integrating currents are used and HD_{ii} will vary. Figure 3–12 shows plots of HD_i , HD_{ii} and Total Harmonic Distortion ($THD = HD_i + HD_{ii}$) calculated using equations 3.7 and 3.9 and measured parameters from a test chip. On initial inspection, a THD of 2% for the Palmo filter compares favourably with a figure of 2.5% for early switched-current circuits [57] but not so favourably with a figure of 0.4% [85] quoted for a more mature S--C FPAA cell. In our graph, distortion introduced by the comparator is dominant at high frequencies where the comparator delays become comparable to the sampling rate. Conversely, distortion due to current matching dominates at low frequencies due to the use of small and therefore less well matched integrating currents.

3.6 Comparator

It is evident that in order to operate the Palmo circuit at high frequencies we need to take special consideration of the comparator. In our second chip a *clamped comparator* (figure 3–13A) was used to reduce the delays and improve the matching between the rising and falling times [46]. This architecture reduces the voltage swing of the differential stage by keeping it clamped. Closer examination of the comparator shows that only the drains of the output transistors M_5 and M_6 have a large voltage swing. Therefore the propagation delay of the differential



Figure 3-13: A) Clamped Comparator, B) Clamped Comparator with positive feedback to increase the gain.

stage is considerably reduced. Unfortunately there is the trade-off of reduced gain in comparison to a standard comparator [46, p. 511]. In fact the gain of the differential stage is almost equal to the ratio of the transconductances g_{m1}/g_{m2} . In order to overcome the low gain problem of the clamped comparator, small positive feedback can be used [46, p. 512], given by transistors M_{10} and M_{11} in figure 3–13B). This increases the gain without having significant impact on the propagation delay. The amount of positive feedback should be less than unity, otherwise the circuit will stop acting like a linear stage. In order for the circuit to operate, some overall negative feedback is needed, which is provided by the source connections of transistors M_1 and M_2 .

Furthermore the differential stage of a *clamped comparator* is a very symmetrical circuit. Therefore if the difference between the channel mobility of n-mos and p-mos channels (μ_n and μ_p) is taken into consideration, and assuming good matching of transistors M_9 and M_6 , the raising and falling delays should be better matched, compared to a standard comparator implementation.

3.6.1 Clamped Comparator Gain

As it was mentioned above the *clamped comparator* seems ideal for the implementation of our *Palmo* chips, because it is fast and the rise and fall delays are matched. Nevertheless the knowledge of the comparator gain (A_v) is needed for



Figure 3-14: Clamped Comparator small signal analysis.

61

the design of such a comparator. This gain is going to be calculated in this section. The overall gain is the product of the gain of the differential stage and the output stage.

The differential stage small signal equivalent circuit of the clamped comparator shown in figure 3–13B is demonstrated in figure 3–14A (assuming that the bulk effect is zero). In this circuit V_{dd} , V_{gs7} and V_{ss} are constant, therefore these voltages are considered to be an AC ground and the effect of the transistor M_7 is zeroed. The input voltage $V_p - V_n$ is a symmetrical AC input with a DC offset. Therefore the AC input can be expressed by two symmetrical inputs $\pm \nu_{in}/2$ where $\nu_{in} =$ $\nu_p - \nu_n$. The AC output is: $\nu_{do} = \nu_{d1} - \nu_{d2}$, while $\nu_{gs11} = \nu_{gs3} = \nu_{ds3}$ and $\nu_{gs11} = \nu_{gs4} = \nu_{ds4}$ as shown in figure 3–14B. The final figure 3–14C can easily be derived by taking into account fact that transistors the $M_{3,4,10,11}$ are PMOS therefore $\nu_{gs3} = -\nu_{sg3} = -\nu_{d1}$, $\nu_{gs4} = -\nu_{sg4} = -\nu_{d2}$ and that the lower input voltage source needs to be inverted.

By applying Kirchoff's Current Law (KCL) to the upper node of the circuit shown in figure 3–14C, assuming the transistor pairs M_1-M_2 , M_3-M_4 , $M_{10}-M_{11}$ are the same we obtain

$$(g_{ds1} + g_{ds10} + g_{m3} + g_{ds3})\nu_{d1} = -g_{m10}\nu_{d2} - g_{m1}\frac{\nu_{in}}{2}$$
(3.11)

By applying KCL to the lower node of the circuit shown in figure 3–14C, we get

$$(g_{ds1} + g_{ds10} + g_{m3} + g_{ds3})\nu_{d2} = -g_{m10}\nu_{d1} + g_{m1}\frac{\nu_{in}}{2}$$
(3.12)

Subtracting (3.12) from (3.11) we obtain

$$(g_{ds1} + g_{ds10} + g_{m3} + g_{ds3}) (\nu_{d1} - \nu_{d2}) = g_{m10}(\nu_{d1} - \nu_{d2}) - g_{m1}\nu_{in} \iff$$
$$A_{V_d} = \frac{\nu_{d1} - \nu_{d2}}{\nu_{in}} = -\frac{g_{m1}}{g_{ds1} + g_{ds10} + g_{m3} + g_{ds3} - g_{m10}}$$

Because for a typical transistor $g_m \gg g_{ds}$ the gain of the differential stage of a clamped comparator with positive feedback yields:

$$A_{V_d} \simeq -\frac{g_{m1}}{g_{m3} - g_{m10}} \tag{3.13}$$



Figure 3–15: Output stage gain calculation.

Closer examination of the output stage (figure 3–15A) shows that the output of the differential stage $(-\nu_d/2)$ is mirrored through the transistors M_8 and M_9 to M_6 if the transistor transconductance ratio β_8/β_9 is α , the small signal input to transistor M_6 is $-\alpha \cdot \nu_d/2$ Therefore the small signal circuit of figure 3–15B is derived and after the inversion of the lower input voltage we get the final figure 3– 15C. The output gain can be easily calculated by the following equation:

$$A_{V_{out}} = -\frac{1}{2} \cdot \frac{g_{m5} + \alpha \cdot g_{m6}}{g_{ds5} + g_{ds6}}$$
(3.14)

It is noted that if $\alpha = 1$ the above equation equals to the gain of a standard push-pull CMOS inverter.

The overall gain of the comparator is

$$A_V = A_{V_d} \cdot A_{V_{out}} \simeq rac{1}{2} \cdot rac{g_{m1}}{g_{m3} - g_{m10}} \cdot rac{g_{m5} + lpha \cdot g_{m6}}{g_{ds5} + g_{ds6}}$$

3.7 Current Matching

The inaccuracy between the charging and discharging currents $(I_p \text{ and } I_n)$ at the *Palmo* cell is the second source of harmonic distortion. Therefore the accuracy of the two current sources is critical for the operation of the circuit in small frequencies. This is limited by *mismatch* of the sourcing transistors [107,105,112]. To improve the accuracy and signify the design guidelines of these current sources,

we will briefly introduce the factors which influence the parameters of a transistor [107,113,69].

The variance of a parameter P between two rectangular devices (ex. the transistors of a current mirror) is given by modelling the long and short correlation distance variance [69,107].

$$\sigma_P^2 \simeq \frac{A_P^2}{WL} + S_P^2 D_P^2$$

where A_P is the area proportionally constant for parameter P^2 and S_P describes the variation of the parameter P with the distance D_P . Both A_P and S_P are process dependent constants. D_P is a highly non-linear function of device distance (D), device orientation, device context, wafer centre distance and other layout specific quantities; for simplicity we can assume that $D_P = D$ [8, Appendix C]. This shows that the sources of mismatch can be modelled in two categories: local area proportional variations and variations which are proportional to the device distance (D). The physical layout of the matched devices strongly influences the parameter variations. The varying factors can be categorised into the following groups

- Local process variations These are due to random variations on all the parameters which are local and inevitable in every process [110,107]. These are modelled by S_P . It is possible to minimise the effect these variations have at the output, by increasing the product $W \cdot L$.
- Process gradients These are systematic variations which can be of a significant value and are mainly proportional to device distance (D) [108][8, Appendix C]. For example oxide thickness and capacitor values can vary uniformly over the same wafer or device size varies from the centre to the edges of the same wafer. *Centroid layout* can be used to minimise these ef-

²The variations W and L originate from edge roughness. These are one-dimensional variants, therefore it would be reasonable to assume that $\sigma_L^2 \propto 1/W$ and $\sigma_W^2 \propto 1/L$ [8, Appendix C][107].

fects, where devices are placed in such a way so that the centroids (centre of mass) of the distributed devices are common [106,108,114][8, Appendix C].

- Device orientation These are process gradients that vary in different directions. It is important that matching devices are placed symmetrically with respect to gradients in order to minimise the effects of the orientation [106][8, Appendix C]. In practice the device gradients are unknown therefore symmetry over the horizontal or vertical axis and all known heat sources will minimise the effects of space and temperature process gradients.
- Boundary effects Which is due to inaccuracies on the boundary of the device. Ensuring that the boundary conditions on the matched devices are identical and splitting the devices to unit sized ones minimises the influence of these variations [106][8, Appendix C].

all these factors should be taken into account to ensure the matching of the two current sources $(I_p \text{ and } I_n)$ in order to minimise the harmonic distortion components due to current mismatches.

3.8 Filter Implementation



Figure 3–16: Filter implementation using differential integrators (a) of a RLC low-pass filter (b).



Figure 3–17: Frequency response of the z-domain transfer function and Palmo Filter.

As an example we will demonstrate the *Palmo* implementation of a fourth order Butterworth low-pass filter with a cut-off frequency of 1kHz, following the filter design algorithm presented in the previous chapter (section 2.2.3) [29].

From the tables which are available in the bibliography we obtain the fourth order Butterworth normalised Low-pass LC ladder parameters, which are scaled to the required cut off frequency by the use of (2.4), where

$$FSF = \frac{1rad/sec}{2\pi f_c rad/sec} = \frac{1}{62.8k}$$

and Z = 1. This yields the LC circuit of figure 3–16b. The lowpass filter can be implemented by the use of *Palmo Miller* integrators (equation 2.6), the topology of the filter is shown in figure 3–16a. The scaling factors K_i can be calculated by the use of the equation $K'_i = \frac{1}{X \cdot f_s}$ where X stands for L_i or C_i in the LC filter and $T = 100\mu S$ is the sampling frequency. This yields:

$$K_1 = K_4 = 0.821$$

 $K_2 = K_3 = 0.34$

The appropriate digital logic to generate the signals ξ_+ and ξ_- for the *Miller* implementation are given by the following equations

$$\xi_{+} = S \cdot P \cdot \overline{M} + \overline{S} \cdot \overline{P} \cdot M$$
$$\xi_{-} = S \cdot \overline{P} \cdot M + \overline{S} \cdot P \cdot \overline{M}$$

where S is the sign clock, P is the plus input and M is the minus input.

The frequency responses of the z domain transfer function, H(z), and the resultant Palmo Filter Implementation were calculated. These are shown in Figure 3–17. The results from the HSPICE simulation of the Palmo Filter Implementation are very close to the theoretical z domain response.

3.9 Conclusions

In this chapter we presented the principals of our *Palmo mixed-signal* approach. The signalling mechanism and the integrator implementation were clarified. Furthermore the limitations of the *Palmo* circuits, because of *harmonic distortion* and comparator delays, were analysed and some solutions were proposed. Finally we demonstrated an implementation example of a fourth order low-pass filter to explain the applicability of the *Palmo* approach.

In the following part of this thesis we will refer to some VLSI *Palmo* implementations which we designed, in order to demonstrate the validity of pulse-based signal processing.

Part II

Implementations

Chapter 4

Palmo-I test-chip

4.1 Introduction

This chapter describes a chip, PALMO-I, designed to confirm the idea that pulsebased systems can be used in signal processing, and especially in filtering, in other words to demonstrate the validity of the Palmo approach.

4.2 PALMO-I Specifications

Our PALMO-I chip has the following features:

- 1. Three elementary analogue cells.
- 2. Analogue to signed-PWM conversion cell.
- 3. Digital I/O routed directly to the environment.
- 4. A supply voltage of 5V.

4.2.1 Elementary analogue cells

To demonstrate the principles of *Palmo* signal processing, we included three different type of cells on the first test chip. This chip enabled testing of elementary filter

PALMO-I test-chip

structures [29,30,32] while more chips could be easily cascaded in order to generate high-order filters. The block diagram of PALMO-I can be seen in figure 4–1.

The upper cell (A) is a two-input one. The two input structures are driven by $\xi_+ - \xi_-$ and $\xi_{+2} - \xi_{-2}$, while the integration is done on the integrating capacitor. This cell can be used for the implementation of a first tap in a low-pass filter. The alternative is to multiplex the two inputs in time, which reduces the maximum frequency of operation by a factor of two.

Cell (B) is a typical, general-use one-input $(\xi + -\xi_{-})$ cell. In the chip there are two test structures like this. The first uses large-magnitude current sources and the other minimum-magnitude current sources. Unfortunately, testing demonstrated that differences in the input charging currents are responsible not only for the alteration of the scaling factor, but also for harmonic distortion and should therefore be avoided. Because the latter cell has minimum sized devices it is liable to random process errors; thus this cell causes more harmonic distortion at the lower frequencies (where current errors become dominant (3–12), because of current inaccuracies [33]).

4.2.2 Analogue to signed-PWM conversion

It was anticipated that the generation of signed PWM signals, needed by the *Palmo* cells, would be a very difficult task. Therefore a typical cell was modified in order to implement an analogue-to-signed-PWM converter (figure 4–1C). Instead of using an integrator-circuit input to the comparator's positive terminal, the terminal was connected directly to the input during the design process. PWM signals are generated at the output, when appropriate ramp signals are applied. This cell proved extremely useful for testing the technique, because it offers an easily accessible node which can be driven by an external analogue input.





4.2.3 Routing

All the analogue bias and digital control signals were routed directly to the output. This was achieved in a 40 pin package, because of the small number of cells. In that way the complexity of the circuit was minimised, while there was no significant increase in the manufacturing cost.

4.2.4 Supply voltage

The process used in our implementation offered the possibility of using two power supplies: an analogue one at (maximum) 14V and a digital at (maximum) 7V. We used separated supplies for the analogue and the digital cells. However because of the risk of *snap-back* break-down [115,116] in some of our minimum-sized transistors, and because of the interfacing complexity, we decided to use only 5V analogue and digital supplies.

4.3 Circuits used in the PALMO-I Chip

This section presents the circuits needed to implement the *Palmo* cells used in the first chip namely:

- 1. Charging and discharging current sources
- 2. Comparator
- 3. Capacitor array
- 4. Input current mirrors

4.3.1 Charging and discharging circuits

As was mentioned in section 3.4.3, current source design is critical for the reduction of clock-feedthrough noise. The circuit used for the implementation of



Figure 4–2: Switched current sources.

these sources uses the switching arrangement mentioned in the previous chapter (figure 4-2). The signals ξ_+ and ξ_- for both the ramp generation and the integration are routed directly to input pads; there is a pad for each integration signal, while there are two global pads for the ramp control signals (ξ_+ and ξ_-). External digital logic can be used to control these signals in order to implement a *Miller* integrator for filter design. The inverters shown in figure 4-2 are standard, two-transistor, CMOS inverters. The reset switch shown in the same figure is used to reset C_{int} to the voltage V_{ref} .

Two versions of current sources were laid out one with minimum-width transistors and a scaled version with 5 times larger devices. This was done in order to demonstrate the significance of the size variations in the circuit. It was because of these small devices that we identified the second source of harmonic distortion in our circuits [8, Appendix C][106,110,33].



Figure 4–3: Comparator schematic diagram, all the NMOS bulks are connected to ground.

4.4 Comparator

Because of time constrains, the comparator used in PALMO-I, and shown in figure 4–3, was an existing cell designed originally for the implementation of a neural network. The cell is inadequate for inclusion in a *Palmo* device: it is very slow and, due to the fact that all the NMOS devices have their bulk terminals tied to ground, including those of the differential pair, it has limited *Common Mode Rejection Ratio* (CMRR). In fact the comparator implementation posed the greatest obstacle to successful operation of our first chip.

4.4.1 Capacitor array

For the implementation of the integrating capacitors, linear, double-poly-silicon capacitors were used. The capacitor ratio was fixed at the design time and it was

$$\frac{C_{int}}{C_{ramp}} = \frac{1}{4}$$

The value of the capacitor C_{int} was calculated to be 0.75pF but process variations mean this may vary by 10%.

At first the fixed capacitor ratio did not seem a disadvantage of our circuit. However, for the implementation of elementary low-pass filters, the capacitor ratio posed a significant limitation to filter design, because it became evident that the capacitor ratio which would have been preferred is the inverted $C_{int}/C_{ramp} =$ 4. The minimum time constant of the *Palmo* circuits expressed in terms of the integrating capacitance and the minimum source current, enabled the operation of the circuit with a minimum sampling frequency of about 1kHz.

4.4.2 Input Current Mirrors

The currents needed to charge the integrating capacitors and generate the ramp are all provided by external current sources. The input currents are generated externally; on the chip these external currents are then divided by 100, to reach more appropriate values (μA). This is done by the use of two input current mirrors with a ratio of 1/10.

4.5 Testing The Chip

This section describes the basic characterisation tests on the first Palmo chip. In addition it discusses the initial comparator problems, which were overcome by the use of *minimum pulse* generation. As was previously mentioned, some digital functionality is needed for the operation of our chip. Our early testing structures based on standard digital-logic chips were later improved considerably, with the use of a reconfigurable FPGA.

4.5.1 Initial Testing

The *Palmo* chip was fabricated using a EUROCHIP 2.4 μm double-poly-silicon process. A photomicrograph of the chip is shown in figure 4–4. After fabrication a series of tests were performed to verify the operation of the chip.



Figure 4-4: Palmo Chip Photograph.

- Power-up To test for hard shortcircuits.
- Comparator test The comparators were biased and the ramp was reset continuously. By varying the voltages at the analogue input and the voltage reference (V_{ref}) pin, the comparator of the fourth cell (PWM converter) changed state, thus indicating operation to a first approximation.
- *PWM Conversion* The circuit to test the PWM conversion was set up to generate the first *Palmo* signals.

4.5.2 Signed PWM Conversion

The circuit shown in figure 4–1C was used for the implementation of analogue-tosigned PWM conversion. The *Analogue Input* voltage was varied from an external voltage source in order to generate signed pulses. A sample frequency of 10kHz was used and the width of the resultant pulses was measured by the use of a digital storage scope; the sign of the pulses was defined by the level of the sign clock. Figure 4–5 shows the size of the resultant pulses for different voltage inputs and ramp generating currents.

In this plot (figure 4-5) it is easy to identify a noticeable offset around zero. This is true both for the positive and the negative signals. The offset is due to comparator delays. As was mentioned in section 3.3, a single-slope ramp was used in the voltage-domain *Palmo* implementations. Therefore any delays generated offsets which were proportional to the frequency of operation. A fast sampling rate would mean that smaller time intervals would be allocated for the PWM representation of the signal. The delays however are not proportional to the frequency of operation (at least to a first approximation). Therefore the smaller the time interval the greatest the significance of the delays at the output. As can be seen in figure 4-5, the delay of the comparator varies with the ramp-generating current: the smaller the current the longer it takes for the comparator to respond to a voltage difference. The delay was measured to be $10\mu s$ for the rising edge and $11\mu s$ for the falling edge at a current of $I_r = 500nA$.

At first it was believed that the problem was due to comparator biasing. The comparator needs to be biased with a voltage V_{bias} to supply a $8\mu A$ current at the drain of the transistor M_7 (figure 4–3). This biasing voltage however is generated from a current mirror biased from an external current supply. All the input currents are divided by a factor of 100 to accommodate the need for driving large currents across chips. Those input current mirrors were designed to provide more that $50\mu A$ at $V_{gs} = 10V$. However in the chip we were limited to a 5V supply due to the fact that some of our transistors would break down due to the *snap-back* effect [115,116]. Therefore the maximum input current that could be achieved was also limited to $2\mu A$. To correct this problem we used the microfabrication facility, available at the University of Edinburgh (MIAC), to shortcircuit the current mirror and directly connect the transistors M_7 and M_6 to a pad. This pad was then biased by a voltage source to the appropriate voltage needed to source $8\mu A$ at the output of transistor M_7 .

The measurements were repeated after this correction was done. Results



Figure 4-5: Original PWM conversion linearity results.



Figure 4-6: Improved linearity results, by the use of the *minimum-pulse* cancellation

PALMO-I test-chip

showed a significant improvement. The delays were $3\mu s$ for the rising edge and $4\mu s$ for the falling edge with a current of $I_r = 500nA$. Nevertheless the delay of almost $5\mu s$ would generate an error of 10% for a signal sampled at 50kHz. This is indeed an intolerable error for most signal processing tasks. The error is due to two factors. The first is the delay of the comparator (about 500ns, 700nS for the rising and falling edges respectively, when the comparator is correctly biased). The second is the time needed for the ramp-generating current to charge C_r to a voltage of a few millivolts needed for the comparator to change state. After the MIAC improvement of the bias arrangement, the effect of the comparator's inherent delays was reduced, making the overall delay more commensurate with the size of the ramp generating currents.



Figure 4-7: Minimum Pulse Generator.

The solution to this problem was found to be the generation of a minimum pulse which was used by the digital logic to cancel the effect of this offset at the output. A unity gain amplifier (figure 4-7) was used to buffer the voltage V_{ref} to $\pm 50mV$, depending on the sign clock. The output of the PWM generating cell, having this voltage as an input, is the minimum pulse which is used by the digital logic to cancel the offset. The 50mV is the minimum voltage difference needed for the differential stage of the comparator to change state.

Results showing the *Palmo* signed PWM conversion of an input voltage range are shown at figure 4–6. These results use the *minimum pulse* generation for the cancellation of the offsets at the output. The voltage to PWM linearity is significantly improved because of the faster comparator response and the *minimum pulse* generation. The size of the ramp-generating currents alter the linearity of the voltage to PWM circuit; however the effect is negligible in most cases apart from the very small ramp generating current of 100nA.

4.6 Further Testing

After addressing the initial offset problems, further testing of the PALMO-I chip was performed, in order to demonstrate the use of pulse-based circuits in signal processing [30,11]. This involved testing the integrator and some elementary lowpass filter structures.

4.6.1 Integrator

The Palmo cell is an integrator implementation which can be used in the construction of active ladder filters. Therefore the operation of the integrator must be characterised, before actual filter implementations are possible.



Figure 4-8: Integrating a signed input: the pulsed based approach.

The operation of the *Palmo* integrator is illustrated by the oscilloscope traces of Figure 4-8. The top trace is the input sine wave, reconstructed from the signed-PWM signal shown in the second trace and generated using the analogue-to-signed PWM circuit described in section 4.5.2. It is possible to identify the zero crossing at this trace, from the *phase shift* which occurs at the PWM in pulse stream. This is due to the sign clock which is used in our sign-magnitude representation.

Integrator Linearity

In order to demonstrate the operation of the integrator, the PWM representation of the sine wave was applied to the *plus* signal of the cell shown in Figure 3–7 and was integrated in time, while the *minus* input was zero. External digital logic generated the ξ_+ and ξ_- inputs to the *Palmo-I* chip of figure 4–1B. The third trace in Figure 4–8 is the ξ_+ signal, containing every second positive pulse from the integrator input (second trace). For simplicity, the ξ_- has not been shown. The ξ_+ and ξ_- signals were integrated in time, resulting in the PWM-coded output signal shown in the fourth trace of Figure 4–8. The final trace is the output sine wave, reconstructed from the signed PWM output of the integrator.

These results indicate that all the individual components of the first *Palmo* filter chip are functional. However, some further testing of the integrator was performed. The graph of figure 4–9 shows the linearity of the *Palmo* integrator for various K factors. In our test chip, the capacitors are fixed, while the current sources are driven externally; therefore the output of the integrator is dependent upon I_{int} , since I_r is constant. The results displayed in figure 4–9 were taken by applying a number of constant pulses to the *Palmo* integrator and measuring the output. The output pulses (*out* in Figure 3–7) were sampled using a digital storage oscilloscope, the pulse width measurement giving the magnitude of the pulse, while the sign of the measurement was defined by the state of the *sign* clock.

These results verified the operation of the *Palmo* integrator, as well as the



Figure 4-9: Integrator programmability and linearity.



Figure 4–10: VLSI results from First Palmo Chip : 1st, 2nd and 3rd order filters at cut-off frequencies of 1kHz and 2kHz.

gain-control functionality. The final step in our testing was to demonstrate the use of the *Palmo* cells in real filter implementations.

4.6.2 Low-pass Filter Implementations

A photograph of the first Palmo device is shown in Figure 4–4. This device has been used to implement the analogue functions of a first, second and third order Butterworth filter (figure 4–10). The signal interconnection between basic Palmo cells as well as other digital functions are performed by a digital FPGA. The results from the VLSI device for cut-off frequencies of 1kHz and 2kHz are compared with the theoretical ideal. The attenuation in the stop band is 40-50dB in these examples. This response is limited by the comparator design, the external current sources which offer bad current matching and the lack of programmable capacitor ratios [31].

The results demonstrate the programmability of the *Palmo* circuits, since it is possible to alter both the shape and the cut-off frequency of the response [31, 30], by simply altering the FPGA configuration. In contrast to conventional techniques, where changing the order of a filter is a complicated task. The *Palmo* filter implementations match the theoretical ideal very well up to the 50dB, which is the *signal to noise ratio* limit for the *Palmo-I* chip.

4.7 Mixed-Signal Systems

While measuring the first chip, it was realised that pulsed-based systems offer an alternative way of implementing signal processing algorithms. This technique is uniquely suited to *Palmo* implementations, offering an alternative to most DSP based solutions, which cannot be matched by any conventional sampled-data analogue system. In this section we will present the technique, and demonstrate the validity of the approach by the use of a FIR example [11,32].

4.7.1 The technique

Digital signal-processing algorithms usually contain an array of delays (z^{-1}) ; the outputs of those memory cells are multiplied by different scaling factors and the products are added together, to form an intermediate output. In other words:

$$y = \sum_{i=0}^{n} a_i \cdot x_i$$

where y is the output, x_i , a_i i = 0, 1, ..., n are the inputs and the scaling factors respectively.



Figure 4–11: Mixed-signal technique.

DSP algorithms suffer in implementing such algorithms, because many cycles of calculations are needed in order to generate this intermediate output. However, since the Palmo technique encodes analogue quantities in time, performing arithmetic functions on pulsed signals in the digital domain is straightforward. In our mixed signal multiplexer we use digital functions to perform binary operations on the *Palmo* signals which simplifies the complexity of a DSP implementation.

PALMO-I test-chip

In our *Palmo mixed-signal* implementation, the analogue cells perform the function of a short-term analogue memory. The *Palmo* outputs Out_i gate the coefficients a_i , and the sum of those outputs is integrated in time. The waveform diagram in figure 4–11 demonstrates the operation of this circuit for two pulsed inputs Out_1 and Out_2 . The coefficients associated with these inputs are $a_0 = 3$ and $a_1 = 5$. At each integrating clock epoch, the sum of the coefficients of active inputs is added to the previous accumulated value. Thus the epochs 3+5=8 is added for the first two epochs, while for the third epoch 5 is added. The overall output appears at the end of the sample period (in this example the output would have the value 21).

In many algorithms there is a need for some feedback between this digital output and the analogue memory cells. This can be easily achieved by the use of a counter which will regenerate the pulsed signal.



4.7.2 Mixed-signal FIR implementation

Figure 4–12: Palmo mixed-signal 24 tap FIR filter implementation results.

A 24 tap FIR filter was implemented to demonstrate the use of such a mixedsignal approach (figure 4–12). A digital FPGA was used for the mixed-signal algorithm while some analogue memory cells performed the function of the delays. The FIR filter implementation is straightforward since it can be done by the circuit mentioned previously (figure 4–11). The output of this circuit is an FIR filter. It is noted that the output is in a digital format therefore filtering and A/D was done by the same *Palmo* technique. The results show excellent match to the theoretical characteristic.

4.7.3 PALMO-I Conclusions

In this section our first *Palmo* chip was presented. Results from silicon demonstrated the use of pulse-based systems in signal processing and specifically in filtering. Finally a technique for a *mixed-signal* approach to DSP specific algorithms was highlighted.

Chip Name	Palmo-I
Number of cells	4
Implementation	Voltage-domain Palmo circuit
Sampling Frequency	1kHz to 50kHz
Power consumption	$0.7 \mathrm{mW}$ percell, at $50 \mathrm{kHz}$
Programmability	By the use of an external FPGA and
	potentiometers
Dynamic range	$50\mathrm{dB}$
THD	0.8% at 10kHz sampling frequency
Q factor	UNDEFINED (depends on the poten-
	tiometer resolution)
Comparator delays	$3\mu s$ rise $4\mu s$ fall
Delay cancellation	minimum pulse

In general the first chip had many constraints posed mainly by the badly designed comparator used in the implementation, in conjunction with the use of a single-sided ramp. Nevertheless these constraints were addressed and the overall performance of the circuits was significantly improved by the use of the *minimum*

PALMO-I test-chip

pulse cancellation. However the maximum sampling frequency for the first chip can not exceed 50kHz because at this frequency the *minimum pulse* uses a very large proportion of the time slot used to represent the magnitude of the signal; thus the accuracy of the pulsed coded signal is reduced. Therefore it was realised that improving the comparator response is essential for future implementations.

Chapter 5 Palmo FPAA and prototyping board

5.1 Introduction

A second chip was designed to facilitate the generation of practical *Palmo* implementations [33,32]. This chip has got an array of 8 analogue cells and its internal SRAM can be reconfigured in order to implement different signal-processing tasks. In this chapter the chip architecture, a prototyping board and some results from practical implementations are presented.

5.2 Chip Architecture

Our second chip (figure 5–1) includes a bigger array of analogue *Palmo* cells. In that way it is possible to implement high order filters and other complex signal processing tasks, more efficiently than with our first chip. The second chip has the following features:

- 1. Eight elementary reconfigurable Palmo cells.
- 2. Internal analogue interconnect.
- 3. Digital logic: SRAM for storing the configuration and cells for accessing the internal registers.




- 4. Improved Comparator architecture with positive feedback.
- 5. An operational amplifier (OPAMP) to buffer the integrated voltage or the ramp to the output.
- 6. A supply voltage of 12V.

5.2.1 Analogue Palmo cells

Eight identical analogue cells are included in our FPAA chip. Every cell has got a reconfigurable array of 9 elementary capacitors of about 500nF each, which can be used to form the *integrating* and *ramp* capacitors C_{int} and C_r respectively. Each cell includes a six bit current DAC to charge either C_{int} , or C_r ; for slow operation the current DAC can be multiplexed in time, for fast operation two cells must be used one to integrate and one to generate the ramp. Dedicated SRAM stores the capacitor and DAC configuration for every cell [33,11,32]. Every cell has got three digital control lines and an output directed to different pads (some inputs are multiplexed with the digital input bus). Those signals are driven by an external FPGA forming a mixed-signal entity.

5.2.2 Internal analogue interconnect

There are transmission gates connecting the integrating and ramp nodes between different cells, in order to cascade them and enable the implementation of more complicated functions. The configuration SRAM of the switches shown in figure 5– 1 is controlled by two global interconnect bytes. There are also some extra CMOS switches connecting the OPAMP and the analogue input pad to different cells.

5.2.3 Digital logic

Digital logic cells are used for implementing the internal configuration SRAM and addressing the cells.

5.2.4 Comparator Architecture

In this chip an advanced clamped comparator is used. This device is designed with respect to minimising the comparator delays. Furthermore this comparator has got two extra control lines for adding some positive feedback which helps the circuit to respond faster.

5.2.5 Output Buffer

A standard library OPAMP cell is used to buffer the integrated voltage or the ramp of cell number 3 to the output. This enables the generation of an analogue signal from the signed PWM representation (DAC).

5.2.6 Supply Voltage

A 12V supply voltage is used in this chip. Therefore the dynamic range of the circuit is improved. However all the digital I/O is done on a 5V level to facilitate the interface of the FPAA to standard digital circuits, like the FPGA we use to pulse the FPAA.

5.3 Palmo FPAA Circuit Details

The following cell implementations, which were used in the design of the *Palmo* FPAA and are critical to the overall device operation, are presented in this section:

- 1. Typical Cell
- 2. Comparator
- 3. Internal addressing

- 4. SRAM implementations
- 5. Level shifters

5.3.1 Typical Cell



Figure 5-2: Typical cell circuit diagram.

The circuit diagram of a typical cell is shown in figure 5–2 [32,11]. Every cell has got three external inputs: $\overline{Pulse_j}$, $Up/\overline{Down_j}$, and Int/\overline{Ramp} , where $j = 0, 1, \ldots 7$ represents the cell number of figure 5–1.

The $Pulse_j$ signal controls the DAC current. It is inverted in order to minimise the propagation delays of the input pad structure. In practice the DAC current is not switched on and off, as it is shown in figure 5–2; an arrangement similar to the one mentioned in figure 3–9 is used to minimise the charge-injection noise. All the other switches of the diagram in figure 5–2 are CMOS transmission gates.

The signal Int/\overline{Ramp} selects between the integrating and ramp SRAMs to control the current DAC output, as it is shown in the *current DAC switch ar*rangement (B_{i/rn_j}) of figure 5-2. Furthermore it controls the switch SW_3 which directs the DAC current to either the integrating or the ramp capacitors.

The signal Up/\overline{Down}_j is responsible for *changing the direction* of the current DAC. This is done by adding the current mirror to the *Palmo* circuit (when

 $Up/\overline{Down_j}$ is high). This current mirror was designed with regard to better matching, therefore big transistors were used with a centroid layout arrangement.

The capacitor array is connected through the switches B_{Crn_j} which are controlled by the capacitor SRAM. Care should be taken by the user not to shortcircuit the integrating to the ramp capacitors.



5.3.2 Comparator Design

Figure 5-3: Palmo FPAA comparator schematic.

The comparator used in out implementation is shown in figure 5–3. It is a clamped comparator [46], the differential input transistors M_1 and M_2 are PMOS transistors in order to use the N-well provided in our process and therefore increase the CMRR. The transistors M_{12} and M_{13} are used to add some positive feedback to our system. The biasing voltages V_{fix_n} and V_{fix_p} force some current to be sinked or sourced by M_{12} and M_{13} respectively, this cancels any comparator delays due to internal mismatches of the comparator design. The generation of these voltages $(V_{fix_n} \text{ and } V_{fix_p})$ is done externally while a negative-feedback loop can be used to ensure proper offset cancellation.

The design of the comparator is the task of specifying the width (W) and length (L) of all the transistors shown in figure 5–3. The specifications we would like our comparator to meet are:

- $A_v = 2000$
- CMR = 6 8V
- $V_{DD} = 12V, V_{SS} = 0V$
- Propagation delay less than 30ns.
- Output voltage swing within 2V

Comparator Design Procedure

The calculations we performed to define the transistor sizes for our comparator are shown in the following paragraphs Setting the output current to meet slew rate requirements, by calculating the current based on the following equation

$$I = 10 \cdot C \cdot \frac{dV}{dt} = 10 \cdot 0.8p \cdot \frac{12}{30n} = 333\mu A$$

Adjusting M_5 and M_6 so that $V_{DS_i(sat)} < 2V$

$$2V > V_{DS_5} = \sqrt{\frac{2I_5}{\beta_5}} \Longrightarrow \left(\frac{W}{L}\right)_5 > 2.8947$$

Similarly for M_6

$$2V > V_{DS_6} = \sqrt{\frac{2I_6}{\beta_6}} \Longrightarrow \left(\frac{W}{L}\right)_6 > 9.7059$$

 $A_{v_{out}}$ may now be calculated by using equation (3.14)

$$A_{v_{out}} = -\frac{1}{2} \frac{g_{m5} + g_{m6}}{g_{ds_5} + g_{ds_6}} = 11.06$$

The gain of the differential stage must be about 180 in order to meet the overall 66dB gain requirement. This gain is given by equation (3.13); the individual gains of transistors M_1 , M_3 and M_{10} depend on W/L of each transistor. There are many possible solutions, but only a few which are practical. There is no formal way to find a practical solution. Therefore heuristic methods, based on an understanding of the constrains, were applied to give the appropriate solutions. Care was taken to maintain some overall negative feedback, by ensuring a difference of $8\mu m$ between $M_{3,4}$ and $M_{10,11}$.

With the device size of M_1 and M_2 calculated $((W/L)_{1,2} = 15)$, the minimum size of M_7 can be adjusted to meet the desired CMRR requirements.

$$V_{G1(min)} = V_{DD} - \sqrt{\frac{I_7}{\beta_7}} - |V_{T1(max)}| \Longrightarrow V_{DS7} = 2.48V$$

Therefore

$$V_{DS7} = \sqrt{\frac{2 \cdot I_7}{\beta_7}} \Longrightarrow \frac{W_7}{L_7} = 1.56$$

The transistor width and length ratios of our comparator are given in table 5-1.

Transistor width over length	Size (W/L)
$4(W/L)_8 = (W/L)_5 = 2.89$	$58\mu/6\mu$
$4(W/L)_9 = (W/L)_6 = 9.7$	$17\mu/6\mu$
$(W/L)_3 = (W/L)_4 = 0.28$	$6\mu/22\mu$
$(W/L)_1 = (W/L)_2 = 15$	$90\mu/6\mu$
$(W/L)_{10} = (W/L)_{11} = 0.20$	$6\mu/30\mu$
$(W/L)_7 = 1.56$	$9\mu/6\mu$

Table 5–1. Clamped Comparator Implementation Parameters

5.3.3 Precharge and Evaluate Address Decoder.

The address decoder is a Precharge and evaluate circuit [46, pp.815-821]. Such circuits make use of the transistor parasitic capacitances to reduce the amount of transistors required by CMOS equivalent implementations. In our decoder when the Enable signal is low, the transistor M_0 is off and all the PMOS transistors are on, therefore the outputs $(Out_0 \dots Out_7)$ will all be high. When enable turns high the transistor parasitic capacitance will maintain all



Figure 5–4: Address Decoder

the output high for about 2ms, apart from one line which will have all four NMOS transistors switched on, driving the output low. This selectivity is obtained by

carefully connecting the gates of the NMOS transistor array to the address lines $(a_0, a_1 \text{ and } a_2)$ or their complements.



5.3.4 SRAM implementations

Figure 5–5: SRAM implementations: A) Standard quasi-static cell, B) Dynamic SRAM cell.

Two different categories of SRAM cells were used to store the configuration data of the PALMO FPAA. The first implementation 5–5A is a standard double inverter quasi-SRAM cell. The two inverters $(M_{1,2} \text{ and } M_{3,4})$ store the input value from the *BUS* when *Select* is high. The transistor M_6 is used to cut the feedback from the second inverter -since there was space available for an extra PMOS transistor in the cell layout-, otherwise the drive circuit for the *BUS* would have to sink (source) the extra current sourced (sinked) by the transistor M_4 (M_3) during the transient period, when *select* is high.

The other SRAM cell (figure 5-5B) is a pseudostatic-dynamic cell. If the *BUS* line is high and the input transistor M_3 is on (*select* high), then the voltage on the parasitic capacitor will switch the output of the inverter (M_1, M_2) to low, even though the voltage on the parasitic capacitor is less than V_{dd} . This will result in turning M_4 fully on and thus the voltage on the capacitor will eventually become V_{dd} . Then switching M_3 off will not affect the output. When the *BUS* line is low, switching M_3 on will result in driving the output to V_{dd} , provided that the *BUS* is able to drive the extra current sourced by M_4 . If the voltage on the *BUS* line is line is maintained low for most of the time and we ensure that the leakage current of

transistor M_3 is bigger than the leakage current of transistor M_4 (by making M_3 wider than M_4), the voltage on the parasitic capacitor will remain low, driving the output high, even though M_3 will be switched off.

This cell actually operates as a SRAM cell even though it dynamically stores charge on the parasitic capacitor. It uses four transistors and it has almost half the size of the six transistor quasi-cell (figure 5–5B), because of routing constrains. This dynamic-SRAM cell is used for a software reset at the *Palmo* FPAA. This was done to test the cell, since it can provide constantly 0V (no reset) while there might be a problem with the duration of the 5V output. This will not affect functionality of the circuit since it will be long enough for resetting the analogue cells.

5.3.5 Level shifter

As it was mentioned earlier the *Palmo* FPAA is designed to operate at 12V, with all the digital I/O done at a standard 5V level. To scale down the output from 12V to 5V is an easy task performed by a standard inverter; the only disadvantage is that the raise and fall times are different, since the threshold



voltage level is 2.5V thus closer to 0V Figure 5-6: Input voltage level shifter. than to 12V. The tricky part is to shift the input level from 5 to 12V. This is done by the circuit shown in figure 5-6. In this circuit the input is within the 5V range. When the input is high (about 5V) the transistor M_1 will turn on thus the voltage at the node (1) will be close to 0V. Therefore the output will be about 12V. When the input is low the inverted input signal at the gate of M_3 will turn high (5V), switching M_3 on and the output to 0V.

5.4 The prototyping Board



Figure 5-7: Prototyping board block (and floor) diagram.

As was previously mentioned the *Palmo* FPAA chip is more complicated than the PALMO-I chip. The user must supply two biasing voltages, and program the internal configuration SRAM before the chip can do anything. The FPAA chip has got 60 pins and the digital FPGA which is used to drive our FPAA chip has got 156 pins. It is apparent that it is very difficult to design and make a wire-wrapped board for testing such an arrangement.

We realised that a prototyping board would be more appropriate for testing this *Palmo* FPAA chip, a block diagram of this board is shown in figure 5–7. Furthermore it would be feasible to link the board to a laptop computer and demonstrate the operation of the *Palmo* circuit. The author did most of the design of this board. However it was actually implemented by Olivier Chapuis -using ORCAD- who worked on this project.

5.4.1 System Level considerations

The prototyping board can operate as a stand-alone system. During the startup procedure though, a host PC is needed to download the configuration bits to the FPGA. The PC also sends the *Palmo* FPAA configuration parameters and controls the operation of a voltage DAC, which is used to generate the biasing voltages. The communication between the PC and the on-board microcontroller is established through a serial link.

5.4.2 Microcontroller and peripheral chips

The Atmel 80C2051 microcontroller (MC) is used on the board. It is a 20 pin, scaled-down version of the industry standard INTEL 8051 MC [117]. It offers two I/O ports, a serial link interface, 128 bytes of SRAM, power-down mode and 2K of flash EEPROM.

The MC clock is generated by the use of a 14.9756MHz crystal oscillator. This strange frequency ensures fast operation and can be used for accurate baud generation. A MAX-232 chip is used to interface the $0 \leftrightarrow 5V$ MC voltage level to the standard RS-232 I/O level.

The MC can also shut-down the board, through a *darlingtron* transistor, for saving energy in battery operated modes.

5.4.3 Analogue biasing

The biasing voltages needed for the operation of the *Palmo* FPAA are supplied by a serial voltage DAC through an OPAMP buffer chip.

5.4.4 FPGA

The XILINX 4005 FPGA [81] is the heart of the system. After it is configured by the MC, it takes control of almost all the signals on the board. It is used to address

the DAC, to generate the *Palmo* signals from the output of the *Palmo* comparators -by using the minimum pulse and the sign clock signals- and to interface the board to another board though a 60 pin connector.

5.4.5 Prototyping area

There is a prototyping area and headphone jacks available to the user, to interface to other circuits on the board.

5.5 Using the board

The board was fabricated by a commercial PCB manufacturer. The populated, operational board, together with the accompanying laptop PC, can be seen in figure 5–8. The debugging of the MC program was done through a development board we fabricated, using a standard 8031 chip [117].

Minor problems which were encountered, while testing the board, were fixed either by changing the MC software (or the FPGA bitstream) or by hardwiring some pins under the board. These minor problems were caused by the serial I/O LEDs, which were not flashing correctly, the wrong pin-out of the OPAMP used, and latching of the DAC. The DAC has some long-term memory (an internal capacitor to store some power for the SRAM cells). When our software forced the DAC into an undocumented state, it would take the device several days to exit this state (by discharging this capacitor). By modifying the MC software, the DAC now operates as expected.

The biggest problem encountered was the lack of an appropriate communication program for the host PC. The XILINX bitstream file is a binary file. This file must be sent trough the serial link to the board. Most communication programs use error checking methods for transmitting binary files, thus corrupting the bitstream. The solution to this problem is to transmit the text-only, hexadecimal version through the serial link, which doubles the size of the bitstream file.



Figure 5-8: Prototyping board and host laptop.

5.6 T.stin Th. FPAA Chip on th. boar.

After verifying the operation of the board, we tested the FPAA chip.

5.6.1 Initial Testing

The *PalmoFPAA* chip was fabricated using a EUROCHIP $2.4\mu m$ double-poly-silicon process. A photomicrograph of the chip is shown in figure 5–9. After fabrication a series of tests was performed, to verify the operation of the chip.



Figure 5-9: Palmo Chip Photograph.

- Power-up To test for hard shortcircuits.
- Comparator test A constant voltage was compared to an externally generated ramp, by the stand alone comparator, which exists on our chip. The comparator changed state according to the ramp, thus verifying the operation of the comparator, the level shifter, and the output-drive cells. The output of the cell was used to measure the comparator delays.

5.6.2 Digital Functionality

The most important part of the FPAA testing was to verify the operation of the digital cells. Because these cells were laid out, instead of using standard cells, which are already verified, they had themselves to be verified. In the case of malfunction the overall functionality of the FPAA would be in questioned, because without the proper operation of the digital cells, the analogue ones might not be able to function.

In order for the analogue circuits to function properly, it is necessary to configure the FPAA interconnection, DACs and capacitor arrays. To do so the FPGA schematic shown in figure 5–10 was used. The data is sent serially to the FPGA



Figure 5–10: Schematic diagram of the FPGA cell used to configure the analogue FPAA chips.

from the microcontroller, clocked by *fclk*. The address bits are shifted into the 8 bit shift register (RS8), and then the \overline{Latch} signal goes low (figure 5–11). This loads the FPGA address register (RD8), which can be used for addressing the FPGA internally. The MC then drives the \overline{ALE} signal low (figure 5–11), which is driven to the appropriate FPAA chip, according to the *address byte* already sent. Consequently, the *data* bits are shifted to the FPGA, the MC \overline{LOAD} signal (figure 5–12) is redirected to the appropriate FPAA in order to load the *data* to the FPAA. The end of the operation is signified from the MC, by taking the \overline{Latch} signal back to high.

There is no indication if the internal FPAA SRAM cells have been correctly configured, since there is no readback facility on the FPAA chip. However, it is possible to identify the correct operation of the digital cells, by looking at the output of the FPAA cells. To do so we connected the *Vref* signal of the FPAA to a ramp generated externally, while the *Input* node was connected to a constant voltage. By changing the global interconnect data, we switched on the transmission gate, which connects the *Input* to cell 5. The output of cell 5 consequently followed





Figure 5-11: Addressing in the FPGA (address = 80h).

Figure 5–12: Loading "AAh" to the FPAA.



Figure 5–13: PWM output by the use of an externally generated ramp.

the ramp (figure 5–13), thus indicating that the data byte has been successfully downloaded to the FPAA. By using the same principal, we verified that all the global interconnect was working properly

A final noteworthy point is the functionality of the reset SRAM. As was mentioned in section 5.3.4, the reset is done by programming a 4 transistor dynamic SRAM cell. The long-term operation of the cell is based on the assumption that the leakage current of the PMOS transistor M_4 (figure 5–5B) is smaller than the leakage of M_3 , assuming that the *BUS* line is low for most of the time. We programmed the reset switch to a logical zero (output high) and we kept the *BUS* line low. The comparator of cell 7 stopped changing state, an effect that was observed even after several days of continuous operation, indicating that the 4 transistor dynamic-SRAM operates equally well to a standard SRAM cell.

Discussion The testing of the digital cells was a difficult task, given that there was no direct way of observing the output of the internal SRAM cells. The strategy we followed (to observe the output of the analogue cells) is not trivial, requiring much effort to devise a methodology, in order to verify the operation of the circuits. The only significant difference between our original design specification and the performance of the circuit was that the two OPAMP controlling SRAM registers were inversely connected.

5.6.3 Testing the Analogue Cells

While writing this thesis tests were performed to the analogue cells. The comparators were characterised and their delays were measured. A new project is currently undertaken, in order to generate a library of FPGA schematics, which will be used to interconnect the analogue cells and perform signal processing functions such as filtering, signal generation, ADC and DAC. It is expected that more results from the analogue cells will be published in the near future when they will become available. However, we have used our analogue cells to prove the suitability of the palmo approach to the implementation of DACs, in particular $\Sigma - \Delta$ modulators.

5.6.4 $\Sigma - \Delta$ Modulator

Oversampled analogue to digital converters, in particular $\Sigma - \Delta$ modulators, are presented in section 2.5.3 [46]. $\Sigma - \Delta$ converters (figure 5–14A) are in fact a pulse-based system; however at a first glance there is no *Palmo* equivalent to such a modulator. This is because a typical $\Sigma - \Delta$ converter integrates in time the analogue input and compares it to zero, in order to generate a coarse estimate that oscillates about the true value of the input. Using a *Palmo* system to achieve that, requires converting the input to a pulse series, which can consequently form



Figure 5-14: Palmo Sigma-Delta modulator.

the input to a typical *Palmo* integrator. In that way the ADC is a function of the accuracy of the initial PWM conversion. However, the circuit of figure 5–14B can perform a $\Sigma - \Delta$ conversion without being limited by any PWM conversion.

The principal of operation of that circuit (figure 5–14B), is that a clock signal is directed to the *positive* input of a typical Palmo cell. This is integrated in time, until the level of the integrated value reaches the value of the *analogue input*, which forces the comparator to change state. The *digital logic* block consequently redirects the *clock signal* to the negative input of the *Palmo* cell, for a given period. This has the effect of lowering the integrated voltage. In that way a coarse estimate, that oscillates about the true value of the input is achieved. A digital filter similar to the one used by a typical $\Sigma - \Delta$ modulator, can be used to average the output of the comparator.

Such a $\Sigma - \Delta$ modulator (figure 5-14B) was implemented in one of our chips; this $\Sigma - \Delta$ converter uses a single *Palmo* cell and two NAND gates to perform the *digital logic* block. A 1MHz clock was used to integrate in time a small current. In our experiment the input voltage varied from 5-11V. The negative input was pulsed 200 times, resulting in a maximum sampling frequency of 0.4kHz. The event was repeated 10 times and the output was averaged. This experiment was



Figure 5-15: Sigma-Delta modulator results.

done 5 times and the five traces can be seen in figure 5–15. The results verify the functionality of the circuit: we observed almost 9 bits of accuracy (52.5dB), and we concluded that the conversion is sufficiently accurate.

Discussion In our attempt to implement a $Palmo \Sigma - \Delta$ modulator, we solved the initial problem of the required PWM conversion. Such a $Palmo \Sigma - \Delta$ modulator is very simple, and can be easily implemented by the use of *only one* Palmo cell and the accompanying FPGA. An added cell can be cascaded, to implement a second order $\Sigma - \Delta$ (figure 2-11). The delay caused by the negative-going integration can be reduced, if a bigger discharging current is used. However, by the use of the same charging and discharging currents, it is possible to cancel any comparator delays. Process variations within, or across, chips will alter the charging current; therefore the output of the $\Sigma - \Delta$ will vary. To minimise this effect, another cell must be used to create a digital reference value, to scale the output of the $\Sigma - \Delta$ modulator.

Chip Name	Palmo-FPAA
Number of cells	8
Implementation	Voltage-domain Palmo circuit
Sampling Frequency	1kHz to 500kHz
Power consumption	$0.95 \mathrm{mW}$ per cell, at $50 \mathrm{kHz}$
Programmability	By the use of an external FPGA and
	reconfigurable DACs-capacitor array.
Dynamic range	54dB
THD	Not Available
Q factor	70dB
Comparator delays	$0.3\mu s$ rise $1\mu s$ fall
Delay cancellation	minimum pulse and positive feedback
	currents at the comparator

Table 5–2. Summary of the Palmo FPAA characteristics

5.7 Conclusions

In this section our second test chip was presented. This chip is an integrated FPAA with digital SRAM cells and an internal Address/Data bus. A board which was built to test the chip, some initial results from testing the device, and a proposed $\Sigma - \Delta$ modulator, were also presented. Table 5–2 summarises the characteristics of the *Palmo FPAA*.

Chapter 6

Advanced implementations: Log-domain BiCMOS Palmo cells

It is apparent, from our previous discussion, that our *voltage domain Palmo* implementations suffer from constrains due to the comparator used, as well as from a small dynamic range limited by the supply voltage level [33,30]. Voltage domain comparators are relatively slow circuits (even the clamped comparator needs several hundreds of nanoseconds to change state), resulting in low sampling frequencies. Furthermore a 12V supply voltage in addition to the minimum voltage difference needed by the comparator to change stage (about 25mV), limits the theoretical maximum dynamic range to 54dB.

The obvious way to bypass the power supply voltage limitation is to operate in the *current mode* [58,55]. This has the added advantage of using current comparators which are faster than their voltage counterparts, and are therefore ideal for *Palmo* implementations [118,119,120,121]. Current comparators have been reported to switch state at high frequencies, even with very small inputs [118,121].

In order to achieve sampling frequencies higher than 1MHz a BiCMOS approach was used to implement novel advanced *Palmo* circuits, because the use of bipolar transistors results in higher operating frequencies and improved accuracy over CMOS circuits.

The transfer characteristic of the bipolar transistor is given by the equation:

$$I_C = I_S \cdot e^{V_{BE}/V_T} - 1$$

where I_C is the collector current, I_S is the saturation current, V_{BE} is the Base Emitter voltage and V_T is the *thermal voltage* (26mV at 300°K). The "-1" can be omitted for most practical implementations.

The exploitation of this $I_C \leftrightarrow V_{BE}$ exponential relation led to the implementation of *log-domain* integrators [122,123,124]. Log-domain integrators [125,126, 127,128,129] integrate logarithmically compressed input currents by making use of this transistor characteristic. The integrated voltage is subsequently expanded exponentially to generate the output current. This companding technique [130,125, 128] offers a large output dynamic range, for a small voltage swing [131]. However the leakage due to the base current and temperature dependency of I_C have to be considered during the design of log-domain circuits.

In this chapter we will present the design ideas and simulation results of a differential log-domain Palmo cell. This is based on the recent work and simulations done by Thomas Brandtner under the guidance of the author. While writing this thesis, Thomas Brandtner is doing the layout of a small BiCMOS chip to test the ideas presented here [34].

6.1 Bipolar Background

In this section the importance of temperature for the bipolar transistor and the translinear principal -which minimises temperature dependency- will be presented [132].

It is in the general belief that reliance on the $I_C \leftrightarrow V_{BE}$ relationship is to be avoided, because of the temperature sensitivity of this equation. Indeed I_C increases almost 9.5% per degree resulting in a variation of about 10⁶ over a typical $-55^{\circ}C$ to $+125^{\circ}C$ temperature range (figure 6–1)! Therefore careful temperature dependency cancellation is required, in any circuit exploiting the use of the $I_C \leftrightarrow$ V_{BE} relationship.



Figure 6–1: Temperature dependency of Is for Vb=700mV of the BiCMOS process used.

The temperature dependency of I_C led to the development of the translinear principle which eliminates temperature dependence for carefully laid-out circuits [132].

The translinear principal In a closed loop containing an even number of forward biased base emitter junctions, arranged so that there are an equal number of clockwise-facing and counterclockwise-facing polarities, the product of the collector current densities in the clockwise direction is equal to the collector current densities in the counterclockwise direction.

6.2 The Log-domain Palmo Cell

The schematic diagram of the *Palmo* cell is shown in Figure 6-2. It consists of three parts: a digital logic block, an integrator and a current comparator. The digital logic block converts the input pulses and the sign signal into two differential input currents that form the input to the integrator. The integrator is fully differential and works in the log-domain [34]. The pulsed output is generated by a current controlled comparator (CCC) which compares the integrator output



Figure 6-2: Palmo cell and typical waveform diagram.

current to a current ramp produced by an identical integrator. This comparator is almost two orders of magnitude faster than the clamped comparator used in the voltage domain circuits, enabling high sampling frequencies [30,11]. The ramp can be generated globally. For high speed operation a dedicated ramp generator may be used for each Palmo cell, in order to reduce the capacitance needed for copying the ramp current to more than one cell.

Discussion Because the circuit has got a big dynamic range and is not limited by the power supply voltage, it is possible to use a dual-slope ramp. The symmetry of the dual slope ramp eliminates inaccuracies due to comparator delays, provided that the rising and falling times of the comparator are well matched. Therefore there is no need for *minimum pulse* generation in order to cancel the comparator delays. The use of a dual-slope ramp results in each sample being represented by two pulses (Figure 6-2). The overall gain of the Palmo cell is controlled by the ratio of the integrating constants of the two integrators generating I_{ramp} and I_{int} . Logdomain integrators are not suited to conventional sampled data implementations, because the input currents should be greater than zero (the logarithm of zero or a negative number is not defined). The complex overhead needed to transform the varying sampled data input currents to positive equivalent current inputs per integrating cell, is the reason why no log-domain sampled data implementations have been reported; even though BiCMOS circuits have been used in sampled data systems [64]. However in the *Palmo* case the input has only two possible values (one for representing "0" and the other for representing "1") for the whole chip,

which makes the technique ideally suited for the implementation of log-domain sampled-data circuits.

6.3 Log-domain cell

At the time of writing this thesis a small chip is being laid-out to test the accuracy of the new current mode *Palmo* cells. This chip will have four cells. One will generate the ramp while the others will be used as standard *Palmo* cells. Three different analogue circuits are needed for the design of the new BiCMOS test chip:

1. Log-domain integrators

2. Current Controlled Comparators (CCC)

3. Voltage to current converters

In this section we will present the architecture of these circuits.

6.3.1 The Log-domain Integrator

The principal circuit of the log-domain integrator used in our *Palmo* cell is shown in Figure 6–3. It is based on [133,126,134,135], but has improved linearity over a bigger current range, due to the use of cascode current mirrors and the stabilising transistors, M13 and M14. The integrator is fully differential, hence the input value is the difference of the two currents I_p and I_n . First the input currents are compressed into log-domain by Q1 (Q8); Q2 and C1 (Q7, C2) perform integration in the log-domain. The integrated value is scaled by Q3 (Q6), the scaling factor depends on the current I_t . Finally, Q4 (Q5) expands the compressed signal. The output is represented by the difference of the currents I_{O1} and I_{O2} . Some additional current mirrors are necessary for producing a real output current which is not shown in this figure.



Figure 6-3: Log-domain integrator.

The following translinear equations [58] for the integrator circuit of Figure 6–3 may be derived from the $I_C \leftrightarrow V_{BE}$ relationship of the bipolar transistor.

$$I_{C1} \cdot I_{C3} = I_{C2} \cdot I_{C4}$$
$$I_{C5} \cdot I_{C7} = I_{C6} \cdot I_{C8}$$

where I_{Ci} is the collector current of transistor Q_i . If the early effect is taken into account the above equation yields:

$$\frac{I_{C1}}{1 + \frac{V_{BC_1}}{V_{AF}}} \cdot \frac{I_{C3}}{1 + \frac{V_{BC_3}}{V_{AF}}} = \frac{I_{C2}}{1 + \frac{V_{BC_2}}{V_{AF}}} \cdot \frac{I_{C4}}{1 + \frac{V_{BC_4}}{V_{AF}}}$$
$$\frac{I_{C5}}{1 + \frac{V_{BC_5}}{V_{AF}}} \cdot \frac{I_{C6}}{1 + \frac{V_{BC_6}}{V_{AF}}} = \frac{I_{C7}}{1 + \frac{V_{BC_7}}{V_{AF}}} \cdot \frac{I_{C8}}{1 + \frac{V_{BC_8}}{V_{AF}}}$$
(6.1)

where V_{BC_i} is the collector-base voltage of transistor Q_i and V_{AF} is the early voltage. The currents $I_{C1} \ldots I_{C4}$ (and similarly $I_{C5} \ldots I_{C8}$) are given by the following equations:

$$I_{C1} = I_p$$

$$I_{C2} = \left(aI_{O2} - I_{cap1} - \frac{I_t}{\beta} + I_c\right) \left(1 - \frac{1}{\beta}\right)$$

$$I_{C3} = I_t$$

$$I_{C4} = I_{O1}$$

where a is the mirroring ratio of the current mirror M5-8, I_{cap1} is the charging current of C_1 and β is the current gain of the bipolar transistors. Substituting these currents to the equations (6.1) yields:

$$\frac{I_p}{1 + \frac{V_{BC_1}}{V_{AF}}} \cdot \frac{I_t}{1 + \frac{V_{BC_3}}{V_{AF}}} = \frac{\left(aI_{O2} - I_{cap1} - \frac{I_t}{\beta} + I_c\right)\left(1 - \frac{1}{\beta}\right)}{1 + \frac{V_{BC_2}}{V_{AF}}} \cdot \frac{I_{O1}}{1 + \frac{V_{BC_4}}{V_{AF}}}$$
(6.2)

$$\frac{I_n}{1 + \frac{V_{BC_8}}{V_{AF}}} \cdot \frac{I_t}{1 + \frac{V_{BC_6}}{V_{AF}}} = \frac{\left(bI_{O1} - I_{cap2} - \frac{I_t}{\beta} + I_c\right)\left(1 - \frac{1}{\beta}\right)}{1 + \frac{V_{BC_7}}{V_{AF}}} \cdot \frac{I_{O2}}{1 + \frac{V_{BC_5}}{V_{AF}}}$$
(6.3)

where b is the mirroring ratio of the current mirror M9–12 and I_{cap2} is the charging current of C_2 . The output currents I_{O1} and I_{O2} respectively is given by the equation:

$$I_{Oi} = I_t e^{\frac{V_b - V_{capi}}{V_{th}}} \Longrightarrow$$

$$\frac{d}{dt} I_{Oi} = -\left(I_t e^{\frac{V_b - V_{capi}}{V_{th}}}\right) \frac{1}{V_{th}} \cdot \frac{d}{dt} V_{capi} = -I_{Oi} \frac{I_{capi}}{C_i \cdot V_{th}} \Longrightarrow$$

$$I_{capi} = -C_i \cdot V_{th} \cdot \frac{1}{I_{Oi}} \cdot \frac{d}{dt} I_{Oi} \qquad (6.4)$$

where V_{th} is the thermal voltage (26mV at 300°K), V_b is the biasing voltage (Figure 6-3). Subtracting the two equations (6.3), substituting I_{capi} from (6.4) and $k_i = 1 + V_{BC_i}/V_{AF}$, while assuming $C_1 = C_2 = C$ and $1/(1 - 1/\beta) \approx 1 + 1/\beta$ results to the final equation:

$$\frac{d}{dt} \left(I_{O1} - I_{O2} \right) = \frac{I_t}{C \cdot V_{th}} \left(1 + \frac{1}{\beta} \right) \left(I_p - I_n \right) + \frac{1}{C \cdot V_{th}} \left(\frac{I_t}{\beta} - I_c \right) \left(I_{O1} - I_{O2} \right) - \frac{1}{C \cdot V_{th}} \left(a - b \right) I_{O1} I_{O2} - \frac{I_t}{C \cdot V_{th}} \left(1 + \frac{1}{\beta} \right) \left[I_p \left(\frac{k_4}{k_1 k_3} - 1 \right) - I_n \left(\frac{k_5}{k_6 k_8} - 1 \right) \right] (6.5)$$

The first term on the right side of (6.5) shows that the difference of the output currents $(I_{O1} - I_{O2})$ is proportional to the integral of $I_p - I_n$. The integration constant is $\frac{I_t}{C \cdot V_{th}}$. It can be controlled by changing the current I_t or the capacitor C. However it is dependent to temperature variations. The overall gain of the Palmo cell is [29,31]

$$K = \frac{I_{t_{int}}}{I_{t_{ramp}}} \cdot \frac{C_{ramp}}{C_{int}}$$
(6.6)

and therefore temperature dependency has been removed from the gain of the cell.

In practice the scaling current I_t should be proportional to temperature. This is not critical to the gain of the integrator but it is needed to achieve the maximum

115

dynamic range, because temperature variations will alter the currents $I_{t_{ramp}}$ and $I_{t_{int}}$ thus changing the dynamic range, while the current ratio will remain the same.

The second term occurs because of the base currents of Q3 and Q5 which introduce a leakage current on the integrating nodes. This is a particular problem for sampled data systems because the integrated value should remain constant between samples. Since the collector current of these transistors is always equal to I_t this term can be easily cancelled by introducing two additional current sources of $I_c = I_t/\beta$. These current sources are derived from the base current of a single bipolar transistor with a collector current of I_t .

The reason for the third term is the difference of the mirroring ratios of the two current mirrors M5-8 and M9-12. The two currents I_{O1} and I_{O2} which are mirrored are not the same, therefore the mirroring ratio should be constant with respect to the mirrored current. To maximise the dynamic range of the circuit at the expense of operating frequency, operating voltage and power, a cascode current mirror is used.

The last term on the right side shows the influence of the early effect. The collector voltages of Q4 and Q5 vary due to output current changes. The transistors M13 and M14 are used to stabilise the collector voltages of Q4 and Q5 to minimise this term.

The integrator may be reset by connecting nodes 1 and 2 together. This approach suffers from charge injection into these two nodes. Although only the difference of the charge injection in the two nodes is important, it is noted that the overall error will be expanded exponentially. Another way of resetting is to increase the base compensating current I_c . If $I_c \gg I_t/\beta$, a = b, $I_p = I_n$ and if the influence of the early effect is neglected, equation (6.5) forces $I_{O1} - I_{O2}$ to exponentially decrease to zero.

6.3.2 Current Controlled Comparator



Figure 6-4: Current comparator

This circuit is mainly responsible for the high sampling frequency capability [34] of the *Palmo* cell because it is faster than the voltage mode comparators used in former *Palmo* implementations [30].

The CCC used was based upon [118] (figure 6-4A). This circuit has got three modes of operation. When I_{in} is positive, the input node is pulled high. This is then amplified by the transistors M_3 and M_4 , switching transistors M_1 fully on and M_2 fully off. In this case the input node is a low impedance node, since M_1 sinks all the input current. However during the transient period when the current changes sign (deadband) the transistor M_1 is still off and the buffer cannot supply the input current, resulting in a high impedance node. When the input is negative the input node is pulled low, turning M_1 off and M_2 on. The size of the deadband in this circuit is determined by the size of V_T . In order to obtain a faster response a technology with smaller V_T is needed.

The comparator used is an improvement of the above circuit proposed in [121] (figure 6–4B). However, it was modified in order to accommodate the lack of p-wells in the fabrication process used. As the voltages at nodes (1) and (2) (figure 6–4B) are increased towards the magnitude V_{T1} and V_{T2} respectively, the deadband in the transfer characteristic is reduced. This results in a smaller voltage swing and thus a faster response. This small output voltage is then amplified by an array of three inverters giving the output. In the circuit of figure 6–4B the deadband

voltage difference is generated by two additional resistors, R1 and R2. The value of the resistor voltage drop can be controlled by the current sources I_{1a} and I_{2b} . The smaller these currents are, the smaller the deadband of the source follower gets, hence the current comparator gets faster. On the other hand the power consumption increases due to the bigger currents in the CMOS inverter stages.

This circuit allows the detection of small input currents (-50nA to 50nA) with a short propagation delay (17ns). Normal current comparators require several hundred nanoseconds to detect the same current change. Thus the operating frequency of the *Palmo* circuit is significantly improved, with the tradeoff of power consumption which is 2mW for the comparator circuit only.

6.4 Voltage to Current Converter



Figure 6–5: Voltage-Current-Converter

This circuit (figure 6-5) will be used to generate input currents as well as the biasing currents (I_t) for the log-domain integrator. It is usually preferable to code analogue signals by the use of a voltage, initiating, this way, the need for an accurate voltage to current converter.

The most appropriate implementation of such a converter [136,137,138], that uses only a 5V power supply (a constraint posed by the process used for the logdomain chip) is shown in figure 6–5. It is based on [137], but needs only one supply voltage rather than two. It consists of a MOS differential stage which outputs a current proportional to the square of the input voltage and two translinear geometric mean circuits, to obtain the square-root of this current.

In figure 6-5, the differential stage that is formed by M_1 and M_2 , suppresses the common mode input signals. Assuming a differential input signal of $\pm V_{in}/2$ the drain currents of M_1 and M_2 is given by the following equation:

$$I_{D_{Mi}} = \frac{\beta}{2} \left(V_{dd} - V_{D_7} - \frac{V_{in_i}}{2} - V_T \right)^2$$

where $\beta = \mu_p C_{ox}(W/L)$, V_{D_7} is the common mode input voltage and V_T is the threshold voltage of the PMOS transistors M_1 and M_2 .

The subcircuit formed by Q_1 to Q_4 (and Q_5 to Q_8 respectively) is a geometric mean circuit. The translinear equation for this circuit is

$$I_{C,Q1} \cdot I_{C,Q2} = I_{C,Q3} \cdot I_{C,Q4}$$

 $I_{C,Q2}$ is equal to $I_{D,M2}$. Therefore the above equation yields

$$I_{C,Q4} = \sqrt{I_0 I_{D,M2}}$$

similarly from the translinear equation of Q_5 to Q_8 derives

$$I_{C,Q8} = \sqrt{I_0 I_{D,M1}}$$

Both currents are subtracted in the node I_{out} resulting in an output current of

$$I_{out} = I_{C,Q4} - I_{C,Q8} = \sqrt{\frac{\beta \cdot I_0}{2}} \cdot V_{in}$$

This circuit has a big input voltage range and is very accurate. Any nonlinearities are due to the non-ideal effects of the translinear circuits and the common mode rejection of the differential stage.

6.5 Simulated Results

The circuits mentioned above were simulated by the use of HSPICE, in order to characterise their operation. In addition some elementary simulations were done to demonstrate the operation of the log-domain integrator and the *Palmo* cell in filtering applications [34].

6.5.1 Integrator Linearity

The linearity of the integrator is shown in Figure 6-6. It demonstrates good linearity with an output range of $\pm 15\mu A$. In the same figure, it is shown that the integrator gain may be set by controlling I_t . It is possible to change I_t between 50nA and $20\mu A$. In addition the integrator gain can be altered by changing the capacitor C. It is planned to use 3 bits to determine the value of the capacitor. The smallest capacitor will be 0.75pF.

If C is greater than approximately 6pF the gain is proportional to 1/C. On the other hand if C is smaller, the constant is only proportional to $1/\sqrt{C}$ due to the propagation delay of the current mirrors in the circuit. When integrating a sine wave, the maximum total harmonic distortion (THD) at the output is less than 0.94%. For sampling frequencies in the kilohertz range the THD is less than 0.5%.

6.5.2 A first-order filter implementation

A first order sampled data low pass filter was implemented by adding the feedback signal shown in Figure 6–2. The frequency response of this simple filter can be seen in Figure 6–7. Here the cutoff frequency is set by controlling the current I_t . As in all sampled data systems, the sampling frequency may also be used to modify the response of the filter.



Figure 6-6: Linearity of the Palmo cell.



Figure 6-7: Frequency response of a simple Palmo filter, fs=1MHz.

6.6 Log-domain Multiplier



Figure 6–8: Log-domain multiplier

Log-domain multipliers exploit the use of logarithms to perform the multiplication. As shown in figure 6–8A, it is possible to multiply two inputs A, B by adding the logarithms of A and B and consequently obtaining the exponent of the result [88]. Bipolar transistors inherently perform the logarithm and antilogarithm functions because of the $V_B \leftrightarrow I_C$ relationship. It is therefore possible to use three of the log-domain Palmo cells to perform a multiplication, as it is shown in figure 6–8B.

By integrating a single pulse the voltage on the integrating capacitor would be a function of the logarithm of the duration of the input pulse. The clock Φ is used to add the integrating capacitor $(C_{1i} \text{ and } C_{1iii})$ in series. Therefore when Φ is high the node (1) at figure 6–8 will be equal to the sum of the two integrated values, provided that the capacitors are equal. The resultant output current will be the exponent of that value, and thus multiplication is performed. A third integrating cell is used to cancel the V_b offset. In order for the circuit to operate, V_b is used for the bias of the bipolar transistors, the voltage on the log-domain integrating capacitor is

$$V_{1_k} = V_b + Klog(I_{in}\Delta_{in})$$

By adding the two voltages we get

$$V_{1_{i+iii}} = 2 * V_b + Klog \left[I_{in} (\Delta T_A + \Delta T_B) \right]$$

The integrated values would be a few millivolts, while V_b could be larger than 1V. Therefore due to the extra V_b offset the circuit will try to force several amperes at the output of the log-domain integrator. For that reason the integrating capacitor (C_{1ii}) of the second cell is used at figure 6-8B, to subtract V_b from the integrating node. The input to this cell is "0" and it is used only for the V_b cancellation. However if the roles of cell (ii) and (iii) are interchanged, it is possible to divide A/B by the use of the same circuit.

Log-domain multiplier discussion The user must take care not to integrate a negative input to the cell, as this will have the effect of dividing the two inputs. The inputs must represent the magnitude of |A| and |B| while the sign will be derived from a digital logic block giving the exclusive OR of the two signs.

Clock feedthrough is a very important factor in this approach. Even the smallest charge added to the integrating capacitors will be expanded exponentially at the output. Therefore transmission gates and a slow clock (Φ) should be used to reduce clock feedthrough. Another important detail is that most of the charge injection occurs at the "ground" side of the capacitors C_{ii} and C_{iii} ($\bar{\Phi}$). However the injected charge will be added to the opposite direction thus cancelling out any possible clock-feedthrough. Finally the reader should note that if a differential log-domain integrator is used, similar to the one presented earlier (figure 6-3), the charge injection would be the same in both integrating capacitors, therefore the differential output should not be affected.

6.7 Conclusions

In this chapter a new log-domain approach to *Palmo* implementations was presented. The integrator circuit presented in this chapter is probably the first sampleddata log-domain integrator ever to be reported, because of the complex overhead needed for the generation of positive, non-zero inputs. Log-domain circuits are ideal for the implementation of *Palmo* circuits because:

- They operate in the current mode, thus current comparators can be used. These circuits are almost two orders of magnitude faster than their voltage counterparts.
- They offer a big dynamic range, therefore a dual-slope ramp can be used since we are not limited by the voltage supply. This virtually eliminates the effect due to comparator delays, thus elaborate *minimum pulse* cancellation is not needed.
- They are based on BiCMOS circuits, which offer a much faster sampling frequency than conventional CMOS circuits.

The cells presented in this chapter will be used in a new *Palmo* BiCMOS log-domain design with the following characteristics:

Chip Name	BiCMOS-I
Number of cells	4
Implementation	Log-domain Palmo circuit
Sampling Frequency	10kHz to 5MHz
Power consumption	$2.2 \mathrm{mW}$ (per cell, mostly due to the cur-
	rent comparator)
$\operatorname{Programmability}$	By the use of V-I converters and 3bit
	programmable capacitors
Dynamic range	70dB
THD	Maximum 0.94%, typical $< 0.5\%$
Q factor	80dB
Comparator delays	17ns for minimum input
Delay cancellation	dual-slope ramp

Chapter 7

Developments and Conclusions

7.1 Introduction

The work presented in this thesis, demonstrated the novel use of pulse-based systems in building analogue programmable cells, ideally suited for the implementation of Field Programmable Mixed Signal Arrays (figure 7–1). The development of FPMAs is an active research area, applying the novel *Palmo* signalling mechanism to this area, and enhances the capabilities of the resultant circuits.

In this chapter we will summarise our work, present the current developments in signal processing, compare our circuits with commercially available alternatives and present ideas for future development of the *Palmo* cells.

7.2 Our Approach

In chapter 3 a novel signal processing concept was introduced, incorporating the use of pulses for representing the inputs. Under this light in the following chapters (4, 5 and 6) some chip implementations were presented. Those circuits were used in simple filtering and analogue-to-digital tasks, to demonstrate the validity of this approach.

Two main circuit categories can be identified in our work, these are linear voltage domain and BiCMOS log-domain current-mode implementations. The voltage domain circuits are not as advanced as the current-mode ones. Sampling

125


Figure 7–1: Typical Palmo FPMA implementation.

frequencies reached 500kHz (table 5–2) and the harmonic distortion was 0.8% (table 4.7.3). This is due to the fact that our signal representation uses time to encode the signal information. Therefore faster operating frequency results in smaller dynamic range. The fact that voltage comparators are relatively slow circuits, poses some limitations on the use of the voltage domain circuits.

Two Palmo CMOS devices were combined with a standard FPGA on a demonstration board. The combination of the two chips provided the impetus for the implementation of analogue systems, such as filters (figure 4–10). By exploiting the mixed signal virtues of both devices more complicated algorithms were implemented, suitable mostly for DSP based solutions (figure 4–12).

However the log-domain approach offers significant advantages to the implementation of *Palmo* circuits (chapter 6). Our implementation is the first logdomain sampled-data system, due to the simplicity of the current input structure. Though our design is a conservative approach, sampling frequencies can reach 5MHz, while some minor improvements of the circuit can increase the sampling frequency to 20MHz. Harmonic distortion is reduced to typically less than 0.5%, the supply voltage is reduced to 5V, however power consumption at the maximum frequency of operation is increased (table 6.7). The reason for all this can be identified in the nature of the current mode comparator, which is almost two orders of magnitude faster, more sensitive to small inputs and energy hungry than the voltage domain counterparts.

7.3 Current Developments

While writing this thesis, we have seen new developments in the semiconductor market. These developments signify the timeliness for the introduction of the *Palmo* techniques. This section presents these advances and clarifies the suitability of the *Palmo* approach which line up to the contemporary expectations of the semiconductor industry. We therefore believe, that exploitation of the techniques introduced in this thesis, can benefit the design of programmable systems, for signal processing.

7.3.1 Using Gate Arrays for Application Specific DSP

In the summer of 1997, FPGA manufactures started promoting research on the use of their products for DSP applications. This was probably done in order to extend the application area of their devices and to respond to the development of Field Programmable Analogue Arrays.

The Palmo mixed-signal algorithm (section 4.7.2) is superior to the new FPGA approach to the implementation of DSP solutions. FPGAs are designed to implement digital circuits such as bus interfaces, counters, microprocessor support devices. The resources available on the FPGA chip are limited. Flip-Flops are needed to implement the delay (z^{-1}) in a DSP algorithm; the number of Flip-Flops in a typical FPGA is limited, thus the DSP algorithm will suffer either in accuracy or in flexibility. The *Palmo* mixed signal approach eliminates the use of Flip-Flops, because the delays are implemented by the analogue cells, while the digital part of the algorithm can be easily performed by a relatively small number of digital cells. Therefore, more complicated algorithms can be implemented, while the unused digital resources can implement memory or I/O tasks. Finally there is no need for extra chips implementing the ADC and DAC, since the *Palmo* FPMA inherently performs those functions.

7.3.2 Texas Instruments TMS-320C6x

In September 1997 Texas Instruments (a company which holds the 50% of the digital signal processing market) launched todays fastest DSP micro-controller—the TMS-320C6x [45,139]. This device incorporates the use of Harvard architecture, with separated command and data, buses and two powerful multipliers performing two 32x32 bit multiplications in 200ns. The chip delivers 1600 Million Instructions Per Second (MIPS), and is able to perform an 8 tap IIR in $0.24\mu s$ (4MHz) and an 24 tap FIR in $67\mu s$ (17MHz).

The Texas Instruments DSP is indeed very fast. However, the BiCMOS cell presented in chapter 6 can operate up to 20MHz (by using Wilson current mirrors), which means that such a circuit can compete favourably with the most powerful DSP. In addition to that the Palmo approach is more energy efficient; the DSP chip will drain a normal battery in a few minutes in mobile, battery operated, stand-alone applications. The log-domain circuit consumes a small fraction of the 6W the Texas chip uses. Furthermore the cost of the very powerful C compiler and emulator for the DSP is \$4000 and \$1000 respectively. In the case of field programmable devices, the cost of the the accompanying software is modest and in some cases is free. Therefore small-mobile applications, using the Palmo circuit, will be cheaper to design, prototype and operate.

7.3.3 Motorola announces the first commercial FPMA

In October 1997 *Motorola* announced the development of a mixed-signal FPMA [99]. This device is based on the Motorola FPAA including some FPGA cells, an analogue-to-digital and a digital-to-analogue converter [99,80].

The Motorola FPMA device is not as advanced as its Palmo counterpart [85]. It is a combination of digital FPGA, analogue FPAA and ADC-DAC cells with minimum interaction in between them. It has limited programmability and can not implement mixed-signal algorithms, such as the one presented in section 4.7.2. Finally it is more than an order of magnitude slower and bigger in size than our log-domain devices.

7.3.4 Institute for System-Level-Integration

In December 1997, Cadence Design Systems announced the creation of the world's largest design facility supported by the world's first System Level Integration (SLI) institute. The development of SLI technology allows companies to create new products quickly, by trading and integrating building blocks from various sources. Because modern designs are becoming more powerful, the complexity of the circuits increases. This leads to longer design and test cycles, which has a negative impact on the price and the reliability of the final product. It is believed that by following the Cadence initiative semiconductor manufacturers will reduce the design time, increase the quality of their chips, and expand their income base, through trading designs, as well as products.

The biggest problem of using analogue SLI is portability. Analogue cells need to be specially designed and simulated for a given process. The design itself poses numerous constraints, and interconnect variations alter the behaviour of the circuit, while crosstalk from the digital lines is responsible for noise in the analogue cells.

Reconfigurable and programmable analogue cells are in general very attractive for SLI. The *Palmo* technique is uniquely suited to the *Cadence* initiative. The voltage domain cell is common place, and porting it to a different technology can be easily done, by altering two transistors (or transistor arrays). Furthermore the signalling mechanism is robust, thus not susceptible to noise from other digital cells, since the *Palmo* technique essentially uses digital signals. The design of an analogue *Palmo* cell is very compact. Therefore it can be easily used in SLI, without effecting significantly the floorplan of the overall chip. We believe that voltage-domain *Palmo* cells are ideally suited for the implementation of SLI.

7.4 Future Developments

The electronics market-place structure is shown in figure 7-2. It is evident that the semiconductor sales is increasing by 10% per annum, while the electronic systems sales is increasing by only 8% per annum. This yields a x2 and x1.4 increase consequently during the following 5 years, when the market trends are expected to continue unaltered. In other words, in every final system sold in the market there is an increasing proportion of semiconductor device cost in it.

Site-Based	Office-Based	Home-Based	Mobile
Telecom,	Desktop PC,	Multimedia,	Wireless Com.
Industrial,	Terminal,	Security,	Portable PC,
Main-Frame	FAX	Energy	Automotive
3%–5%	8%-10%	15% - 20%	20%-30%

Table 7–1. Market-growth tendencies in the foreseeable future



Figure 7–2: 1996–1997 World-wide Electronics Market-place (source: *ICE status* 1997). In table 7–1 the current market growth tendencies are shown for the four main categories of electronic devices. It is believed that in the foreseeable future mobile applications are going to dominate the market. In mobile applications the cost of the semiconductor devices is a big proportion of the overall cost (about 50% in mobile telephones), therefore it is evident that semiconductor sales are expected

to go up in the near future.

As a result every modern integrated circuit should comply to the main factors which are essential to mobile applications. Those factors are: Frequency of operation, increased integration, low-power consumption, device quality and cost issues.

7.4.1 Future Work

The *Palmo* approach, developed during the this PhD course, is novel. Our research of the field signified the applicability of the approach, however the field, is still unexplored territory. Future research should target portable applications.

Voltage domain CMOS circuits

The research field of the voltage domain CMOS circuits is relatively small. The advantages of the CMOS *Palmo* cells are that such circuits are simple, therefore cost effective, since they can be easily designed, tested and manufactured. Furthermore, the circuits used are compact and the lack of routing constrains, enables increased integration. Finally the device quality is good. However, the frequency of operation cannot exceed one or two MHz and the power supply must be over 5V, thus power consumption can not be significantly reduced.



Figure 7-3: Dedicated minimum pulse cancellation.

Nevertheless the CMOS cells can be improved. It was proven that the *minimum* pulse cancellation (section 4.5.2) is essential to the operation of the *Palmo* circuits, which use single slope ramp. The cancellation can be achieved by the use of the

circuit presented in figure 7–3. A second comparator local to every cell, generates a dedicated *minimum-pulse* signal, which is dependent on the slope of the ramp used by the cell. This minimises any *minimum-pulse* inaccuracies due to component or ramp mismatches, with a trade-off in area and power consumption. It is believed that such a system will be able to operate at higher sampling frequencies with improved accuracy.

The comparator design is essential to the *Palmo* implementation, and thereby the use of a better design can result in better performance. Finally it is worth investigating the use of companding techniques in CMOS circuits [140,141,142]. Though the MOS transistor $V \leftrightarrow I$ characteristic is not exponential, companding techniques can reduce the operating voltage and therefore power consumption.

Log-Domain BiCMOS circuits

The log-domain circuit which was presented in chapter 6 is much better suited to *Palmo* implementations, due to better accuracy and faster response of the CCC. The simulated results from that cell look promising and it is believed that results from our test chip will verify our simulations. Nevertheless the circuits presented in chapter 6, indicate a conservative approach to log-domain *Palmo* cells, because our main concern during the design procedure was to make the circuits work, rather than to improve their performance.

Delays in the circuit are derived from the current mirrors $(M_{5,6} - M_{7,8})$ and $M_{9,10} - M_{11,12}$ of figure 6-3. In order to increase the frequency of operation a different design for those current mirrors must be used. Changing them to standard two transistor current mirrors will increase the frequency of operation to 10MHz and decrease the power supply requirements. However, using *Wilson* current mirror structures enables sampling at 20MHz, which is at least an order of magnitude grater than other sampled-data field programmable analogue approaches. Finally the use of bipolar PNP structures, for implementing these current mirrors, will enable the use of sampling frequencies as high as 100MHz.

Developments and Conclusions

At such high frequencies on chip FPGA cells should be used, as they are necessary for the proper operation of the Palmo cells. The use of different chips for the analogue and the digital cells will introduce big delays, therefore developing the first *Palmo* FPMA is critical for high frequency systems.

The power consumption of the log-domain cell is mostly due to the current comparator. This circuit is energy hungry (2mW out of the total 2.1mW consumed by the cell), however it is believed that it would be possible to reduce energy consumption, without significant loss in the frequency of operation or accuracy of the comparator. This can be done by reducing the comparator voltage, and/or switching it off for most of the time (when it is not used). A Flip-Flop can be used to store the comparator output and the CCC circuit can be switched on, only when the ramp is almost equal to the output current of the log-domain integrator. Finally other CCC designs can be investigated.

A bigger test chip will be designed in the future which will probably contain 16 log-domain cells combined in an FPAA. A fast external FPGA will be used to perform the routing.

An improvement of the log-domain input structure will increase its application base. The use of more than one input will enable the synchronous addition of different inputs. Furthermore if scaling is performed before integration, the number of cells needed to perform certain complicated functions (such as band-stop filters) will be reduced, while the maximum frequency of operation might be increased. The signalling mechanism should be investigated as well; it is believed that the sign-magnitude approach has reached its limits at high frequencies and another more appropriate coding schemes might perform better.

Applications

As it was shown above, our circuits can be significantly improved, they can be designed to work faster, use low-voltage, be more programmable or smaller in size; nevertheless there is no evident course we should follow. The analogue field programmable market is new and there is not significant feedback from users to guide the development of mixed-signal systems.

Therefore we must identify possible applications which can benefit from the use of *Palmo* techniques. This is not an easy task for any individual, however it is critical to our research.

Some application areas which should be targeted are: automotive, diagnostic equipment, medical-instrumentation, difficult conditions of operation (for example space, high-pressure, high-temperature, radioactive areas), UPS and battery control, power systems, data acquisition, distance measurement, sensor implementations, reconfigurable test equipment or process control. It might even be useful to implement application oriented *Palmo* cells for tackling a given problem.

7.4.2 Overall Conclusions

In this thesis a new technique for performing sampled data, signal processing was presented. This technique incorporates the use of pulses as the signalling mechanism, in order to facilitate the design of programmable mixed-signal hardware. This technique was pioneered by the author and was named "Palmo" after the hellenic word " $\Pi \alpha \lambda \mu os$ ". Three chips (two CMOS and a BiCMOS one) were designed to implement the Palmo concept. These chips were used in practical signal processing sampled-data applications. Furthermore a mixed-signal approach to DSP algorithms was presented and was used to implement an FIR filter.

As shown in table 7.4.2 the *Palmo* technique lies in between the analogue and the digital world. The biggest advantage of the *Palmo* approach is the extended programmability, in conjunction with the enhanced frequency of operation that it offers. In some cases it is better than analogue systems, in other cases it is more appropriate than digital ones. However, it depends on the application and the user to define the suitability of the *Palmo* approach for a given problem.

It is negotiable if this approach will have a significant impact in the field programmable or signal processing market. This is not due to any defects or limitations of our signalling mechanism; the reason for that can be identified in the Table 7–2. Characteristics of Analogue, Palmo and Digital signal processing implementations.

	Analogue	Palmo	Digital
Programmability:	limited	fair	extended
Frequency of operation: ^a	high	adequate	slow
Power consumption:	low	average	extremely high
Signal robustness:	low	adequate	highest
Size:	\mathbf{small}	smallest	biggest
Design time: ^b	long	shortest	\mathbf{short}
Design CAD tools:	limited	available	extended
Timeliness:	old	just emerged	at their peek

^aIn general, for programmable analogue hardware the *Palmo* approach is best. ^bTime needed for solving a signal processing task.

inherent consciousness of the market. It is not easy for our approach to outperform well established conventional techniques such as SC or even SI. However, in a particular application area such as data acquisition, automotive applications, or even mobile communications and portable computers, the *Palmo* approach might provide the means to improve performance, which will give a boost to the development of our technique.

The timeliness of our research is good, as indicated by some recent developments in the semiconductor industry (section 7.3). Furthermore our work compares favourably to conventional analogue techniques. However, there is still much ground for improvements, since the field of *Palmo* signal processing is just being investigated. If the market and other researchers adopt this technique, will be clarified in the future.

Bibliography Categories:

In this section our reference list is sorted by category, in order to facilitate the reader to locate the desired reference.

Analogue Voltage Domain VLSI: [47] [52] [53] [54] [106] [132]

Artificial Neural Networks: [5] [6] [7] [9] [10] [101] [103] [104] [116]

Author's Publications: [11] [29] [30] [31] [32] [33] [34]

Current Controlled Comparators: [118] [119] [120] [121]

Current Mode: [48] [49] [55] [56] [57] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69]

 Field Programmable Analogue Arrays:
 [51]
 [76]
 [77]
 [78]
 [79]
 [80]

 [82]
 [83]
 [84]
 [85]
 [86]
 [87]
 [89]
 [90]
 [91]
 [95]
 [96]
 [97]
 [98]
 [99]
 [102]

Literature: [8] [14] [19] [22] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [46] [50] [58]

Log-Domain: [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [133] [134] [135] [140] [141] [142]

Manuals: [73] [74] [75] [81] [92] [93] [94] [117]

Matching: [69] [100] [105] [107] [108] [109] [110] [111] [112] [113]

Various: [1] [2] [3] [4] [139]

Voltage to Current Converters: [136] [137] [138]

Wavelets: [12] [13] [15] [16] [17] [18] [20] [21] [23] [24] [25] [26] [27] [28]

Bibliography

- [1] Cicero, De re Publica. by. C.W. Geyes, Loeb, 1969.
- [2] J. Field and M. Wright, Earlly Gearing: Geared Mechanisms in the Ancient and Medieval World. London: Science Museum, 1985.
- [3] D. J. D. S. Price, "Gears from the Greeks: The antikythera mechanism—a calendar computer from ca. 80 BC.," Science Histoy Publications, 1975.
- [4] D. J. D. S. Price, "An ancient Greek computer," Scientific Ammerican, June 1959.
- [5] A. Murray, "Pulse arithmetic in VLSI neural networks," *IEEE Micro*, vol. 9, no. 6, pp. 64–74, 1989.
- [6] A. Hamilton, S. Churcher, P. J. Edwards, G. B. Jackson, A. Murray, and H. M. Reekie, "Pulse stream VLSI circuits and systems : The EPSILON neural network chipset," *International Journal of Neural Systems*, vol. 4, pp. 395–406, December 1993.
- [7] A. Hamilton, A. Murray, D. Baxter, S. Churcher, H. Reekie, and L. Tarassenko, "Integrated pulse-stream neural networks - results, issues and pointers," *IEEE Trans. Neural Networks*, vol. 3, pp. 385–393, May, 1992.
- [8] T. Lehmann, Hardware Learning in Analogue VLSI Neural Networks. PhD thesis, Electronics Institute, Technical University of Denmark, Lyngby, 1994.
- .[9] J. Quero, J. Ortega, C. Janer, and L. Franquelo, "VLSI implementation of a fully parallel stochastic neural network," in *International Conference on Neural Networks*, pp. 2040–5, IEEE, 1994.
- [10] G. Cauwenberghs, "An analog VLSI recurrent neural network learning a continuous-time trajectory," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 346-61, 1996.
- [11] A. Hamilton and K. Papathanasiou, "Preprocessing for pulsed neural VLSI systems," in *Pulsed Neural Networks* (W. Maass and C. Bishop, eds.), MIT Press, in press.
- [12] R. Lyon, "Analog VLSI hearing systems," in VLSI Signal Processing, III, IEEE Press, 1988.
- [13] R. Lyon and C. Mead, "A CMOS VLSI Cochlea," in International Conference on Acoustics, Speech, and Signal Processing, IEEE, 1988.
- [14] Y. C. Pati, Wavelets and time-frequency methods in linear systems and neural networks. PhD thesis, University of Maryland, College Park, 1992.

- [15] M. Desai and D. Shazeer, "Acoustic transient analysis using wavelet decomposition," in Conference on Neural Networks for Ocean Engineering, IEEE, 1991.
- [16] J. Lin, W.-H. Ki, K. Thompson, and S. Shamma, "Realization of Cochlear filters by VLT Switched- Capacitor Biquads," in *International, conference* on Acoustics, Speech and Signal Processing, IEEE, 1992.
- [17] C. Summerfield and R. Lyon, "ASIC implementation of the Lyon Cochlea model," in *International Conference on Acoustics, Speech and Signal Pro*cessing, IEEE, 1992.
- [18] A. N. Akansu and R. A. Haddad, Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets. Academic Press, 1992.
- [19] P. T. J.M. Combes, A. Grossmann, WAVELETS time-frequency methods and phase space. Spiegel, 1988.
- [20] O. Rioul and M. Vetterli, "Wavelets and signal processing," IEEE Signal Processing Magazine, no. 4, pp. 14–38, 1991.
- [21] P. M. Bentley and J. T. E. Mcdonnell, "Wavelet transforms an Introduction," *Electronics & Communication Engineering Journal*, no. 4, 1994.
- [22] X. Huang, Y. Ariki, and M. Jack, Hidden Markov Models for Speech Recognition. Edinburgh University Press, 1990.
- [23] J. F. Lin, W. H. Ki, T. Edwards, and S. Shamma, "Analog VLSI implementations of Auditory Wavelet transforms using Switched-Capacitor circuits," *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications*, no. 9, 1994.
- [24] R. T. Edwards and G. Cauwenberghs, "Analog VLSI processor implementing the continuous wavelet transform," in *NIPSS*, Morgan Kauffman, 1995.
- [25] J. Hoyt and H. Weschsler, "The wavelet transform-A CMOS VLSI ASIC implementation," in International Conference on Pattern Recognition, conference D: Architectures for Vision and Pattern Recognition, IEEE, 1992.
- [26] M. Misra and V. Prasanna, "Parallel computation of 2-D Wavelet transforms," in International Conference on Pattern Recognition, conference D: Architectures for Vision and Pattern Recognition, IEEE, 1992.
- [27] H. Sahinoglou and S. Cabrera, "A high-speed pyramid image coding algorithm for a VLSI implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 4, pp. 369–74, 1991.
- [28] S. Ganesan, S. Mahalingam, and S. Nagabhushana, "VLSI synthesis of a programmable DWT chip for the optimal choice of a prototype Wavelet," in *Proceedings. Third Great Lakes Symposium on VLSI Design Automation of High Performance VLSI Systems*, IEEE, 1993.
- [29] K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in *International Symposium on Circuits* and Systems (ISCAS), vol. I, (Atlanta), pp. 270–273, May 1996.

- [30] K. Papathanasiou and A. Hamilton, "Advances in programmable pulse based mixed-signal processing VLSI," in *IEEE-CAS Workshop on Analog and Mixed IC Design, Baveno*, (F. Maloberti, A. Baschirotto, and V. Liberali, eds.), (Baveno-Italy, 12-13 Sep), pp. 106-110, IEEE, 1997.
- [31] K. Papathanasiou and A. Hamilton, "Palmo signal processing: VLSI results from an integrated filter," in *The Third IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, vol. 2, (Rhodes-Ellas), pp. 832– 835, October 1996.
- [32] K. Papathanasiou and A. Hamilton, "Novel palmo techniques for electronically programmable mixed signal arrays," in *International Symposium on Circuits and Systems (ISCAS)*, (California), May 1998.
- [33] K. Papathanasiou and A. Hamilton, "Novel palmo analogue signal processing IC design techniques," in *Colloquium on Analogue Signal Processing*, (Oxford,UK), pp. 5/1-5/6, November 1996.
- [34] T. Brandtner, K. Papathanasiou, and A. Hamilton, "A palmo cell using sampled data log-domain integrators.," submitted to the IEE electronics letters, 1998.
- [35] J. G. Proakis and D. G. Manolakis, *Digital signal processing : principles, algorithms, and applications.* Prentice Hall, 1996.
- [36] P. M. Grant, C. F. N. Cowan, B. Mulgrew, and J. H. Dripps, Analogue and digital signal processing and coding. Chartwell-Bratt, 1989.
- [37] J. H. Mole, Filter design data for communication engineers. E. & F. N. Spon, 1952.
- [38] J. K. Skwirzynski, Design theory and data for electrical filters. Van Nostrand, 1965.
- [39] B. Williams and F. J. Taylor, Electronic Filter Design Handbook. McGraw-Hill, 1995.
- [40] H. J. Blinchikoff and A. I. Zverev, Filtering in the time and frequency domains. Wiley, 1996.
- [41] J. L. Herrero and G. Willoner, Synthesis of filters. Prentice-Hall, 1966.
- [42] M. Ismail and T. Fiez, Analog VLSI signal and information Processing. New York: Mc Graw Hill, 1992.
- [43] Y. Tsividis and P. Antognetti, Design of MOS VLSI circuits for telecommunications. Prentice-Hall, 1985.
- [44] K. Laker and W. Sansen, Design of analog integrated circuits and systems. McGraw-Hill, 1994.
- [45] K. S. Lin, G. A. Frantz, and R. Simar, "The TMS320 family of digital signal processors," *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1143–1159, 1987.

- [46] N. S. R.L. Geiger, P.E. Allen, VLSI design techniques for analog and digital circuits. McGraw-Hill, 1990.
- [47] G. A. S. Machado, N. C. Battersby, and C. Toumazou, "On the development of analogue sampled-data signal processing," *Analog Integrated Circuits and Signal Processing*, vol. 12, no. 3, pp. 179–199, 1997.
- [48] J. Ramirez-Angulo and E. Sanchez-Sinencio, "High frequency compensated current-mode ladder filters using multiple output OTAs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 9, pp. 581-6, 1994.
- [49] T. Inoue, Q. Y. Pan, F. Ueno, and Y. Ohuchi, "Design and analysis of OTA switched-current mirrors," *IEICE Transactions on Fundamentals of Elec*tronics Communications and Computer Sciences, vol. E76A, no. 6, pp. 940– 946, 1993.
- [50] J. Maxwell, A Treastise on Electricity and Magnetism. Clarendon Press, 1891.
- [51] D. J. Allstot, R. W. Brodersen, and P. R. Gray, "An electricallyprogrammable switched capacitor filter," *IEEE Journal of Solid-State Circuits*, pp. 1034–1041, December 1979.
- [52] A. Sedra, "Switched-capacitor filter synthesis," in Design of MOS VLSI circuits for telecommunications (Y. Tsividis and P. Antognetti, eds.), pp. 272– 313, Prentice-Hall, 1985.
- [53] P. Gray and R. Castello, "Performance limitations in switched-capacitor filters," in *Design of MOS VLSI circuits for telecommunications* (Y. Tsividis and P. Antognetti, eds.), pp. 314–333, Prentice-Hall, 1985.
- [54] D. L. Fried, "Analog sample-data filters," IEEE Journal of Solid-State Circuits, August 1972.
- [55] S. Daubert, D. Vallancourt, and Y. Tsividis, "Current copier cells," *Electronics Letters*, vol. 24, no. 25, pp. 1560-2, 1988.
- [56] G. Wegmann and E. Vittoz, "Analysis and improvements of accurate dynamic current mirrors," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 699-706, 1989.
- [57] T. Fiez, G. Liang, and D. Allstot, "Switched-current circuit design issues," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 3, pp. 192–202, 1990.
- [58] F. L. C. Tomazou and D. Haigh, Analogue IC design: the current-mode approach. London: IEE Circuits and systems series, 1990.
- [59] S.-S. Lee, R. Zele, D. Allstot, and G. Liang, "A continuous-time currentmode integrator," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 10, pp. 1236-8, 1991.
- [60] C. Toumazou and N. Battersby, "Switched-transconductance techniques," *Electronics Letters*, vol. 29, no. 21, pp. 1816–18, 1993.

- [61] C. Toumazou and N. Battersby, "Switched-transconductance techniques: a new approach for tunable, precision analogue sampled-data signal processing," in *IEEE International Symposium on Circuits and Systems*, pp. 1247-50, IEEE, 1993.
- [62] T. Serrano and B. Linares-Barranco, "The active-input regulated-cascode current mirror," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 6, pp. 464-7, 1994.
- [63] R. Balmford and W. Redman-White, "New high-compliance CMOS current mirror with low harmonic distortion for high-frequency circuits," *Electronics Letters*, vol. 29, no. 20, pp. 1738–9, 1993.
- [64] T. Reimann, F. Krummenacher, and M. J. Declercq, "An 8-b, 40 msamples/s switched-current-mode track-and-hold circuit on a BiCMOS sea-of-gates array," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 304–311, 1996.
- [65] D. Nairn, "Analytic step response of MOS current mirrors," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 40, no. 2, pp. 133-5, 1993.
- [66] S. J. Daubert and D. Vallancourt, "Operation and analysis of current copier circuits," *IEE Proceedings-G Circuits Devices and Systems*, vol. 137, no. 2, pp. 109–115, 1990.
- [67] T. Fiez and D. Allstot, "CMOS switched-current ladder filters," IEEE Journal of Solid-State Circuits, vol. 25, no. 6, pp. 1360-7, 1990.
- [68] A. Yufera, A. Rueda, and J. Huertas, "Programmable switched-current wave analog filters," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 8, pp. 927– 35, 1994.
- [69] J. B. Shyu, G. C. Temes, and F. Krummenacher, "Random error effects in matched MOS capacitors and current sources," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 6, pp. 948–955, 1984.
- [70] J. R. Ashley, Introduction to analog computation. Wiley, 1963.
- [71] A. Hausner, Analog and analog/hybrid computer programming. Prentice-Hall, 1971.
- [72] V. Paschkis and F. L. Ryder, Direct analog computers. Interscience Publishers, 1968.
- [73] 4th and 8th-Order Continuous-Time Active Filters, MAX274 / MAX275 / Software / EV Kit ed., 1992.
- [74] LMF100—High Performance Dual Switched Capacitor Filter, 1994.
- [75] UFA42—Universal Active Filter, 1993.
- [76] S. Espejo, R. Carmona, R. Dominguezcastro, and A. R. vazquez, "A CNN universal chip in CMOS technology," *International Journal of Circuit Theory* and Applications, vol. 24, no. 1, pp. 93–109, 1996.

- [77] K. R. Crounse and L. O. Chua, "The CNN universal machine is as universal as Turing machine," *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications*, vol. 43, no. 4, pp. 353-355, 1996.
- [78] T. Roska and L. O. Chua, "The CNN universal machine an analogic array computer," *IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 163–173, 1993.
- [79] The University of Toronto Field-Programmable Analog Array Web Site:, http://www.eecg.toronto.edu/eecg/research/fpga.html ed.
- [80] P. G. Gulak, "Field-programmable analog arrays: a status report," in The fourth Canadian workshop on Field-Programmable Devices, (Canada), pp. 138-141, May 1996.
- [81] XILINX—The Programmable Logic Data Book. San Jose California, 1993.
- [82] M. A. Sivilotti, "A dynamically configurable architecture for prototyping analog circuits," in 5th Mit Conf. on Advanced Research in Vlsi (J. Allen and F. T. Leighton, eds.), (Cambridge MA 02142), pp. 237-258, M I T Press, March 1988.
- [83] E. Lee and P. Gulak, "A CMOS Field-Programmable Analog Array," in International Solid-State Circuits Conference, vol. 38, (Francisco, Ca), pp. 186-187, IEEE, 1991.
- [84] H. Kutuk and S. Kang, "A Field-Programmable Analog Array (FPAA) using switched-capacitor techniques," in *International Symposium on Circuits and* Systems (ISCAS), vol. IV, (Atlanta), May 1996.
- [85] A. Bratt and I. Macbeth, "Design and implementation of a Field Programmable Analogue Array," in FPGA '96, Monterey California: ACM, February 1996.
- [86] H. Kutuk and S. M. S. Kang, "Filter design using a new Field-Programmable Analog Array (FPAA)," Analog Integrated Circuits and Signal Processing, vol. 14, no. 1-2, pp. 81-90, 1997.
- [87] E. Pierzchala, M. A. Perkowski, P. Vanhalen, and R. Schaumann, "Currentmode amplifier integrator for a Field-Programmable Analog Array," in *International Solid-State Circuits Conference* (J. H. Wuorinen, ed.), vol. 38, (Francisco, Ca), pp. 196–197, IEEE, 1995.
- [88] D. L. Grundy, "A computational approach to VLSI analog design," Journal of VLSI Signal Processing, vol. 8, no. 1, pp. 53–60, 1994.
- [89] A. Hematy and G. W. Roberts, "A fully-programmable analog log-domain filter circuit," in *IEEE-CAS Workshop on Analog and Mixed IC Design*, (F. Maloberti, A. Baschirotto, and V. Liberali, eds.), (Baveno-Italy, 12-13 Sep), pp. 91–96, IEEE Service Center, 1997.
- [90] A. Wyszynski and R. Schaumann, "Frequency and phase tuning of continuous-time integrated filters using common-mode signals," in *International Symposium on Circuits and Systems*, (London), pp. E 269–E 272, IEEE, June 1994.

- [91] E. K. F. Lee and P. G. Gulak, "A transconductor-based Field-Programmable Analog Array," in *International Solid-State Circuits Conference* (J. H. Wuorinen, ed.), vol. 38, (Francisco, Ca), pp. 198–199, IEEE, 1995.
- [92] IMP50E20 EPAC: advance product information, Jan. 1996.
- [93] I. H.-P. http://www.impweb.com/, Electrically Programmable Analog Circuit: Datasheet. San Jose California, April 1996.
- [94] Z. H.-P. http://www.zetex.com/, Totally Reconfigurable Analog Circuit (TRAC): Datasheet. Chadderton-Oldham, November 1996.
- [95] P. C. P. Chow and P. Gulak, "A Field-Programmable Mixed-Analog-Digital Array," in *The fourth Canadian workshop on Field-Programmable Devices*, (Canada), pp. 104–108, May 1996.
- [96] K. A. El-Ayat, "Mixed-mode analog/digital programmable interconnect architecture," tech. rep., US Patent No 5,107,146, Apr 21 1992.
- [97] M. W. O.P. Agrawal and S. Sidman, "Programmable logic device incorporating voltage comparator," tech. rep., US Patent No 5,153,462, Oct. 6 1992.
- [98] O. Agrawala and M. Wright, "Programmable logic device incorporating digital-to-analog converter," tech. rep., US Patent No 5,191,242, Mar. 2 1993.
- [99] http://www.electronicsweekly.co.uk, "Mixed signal makes sense," Electronics Weeklly, pp. 22-23, Octomber 8 1997.
- [100] J. H. Shieh, M. Patil, and B. J. Sheu, "Measurement and analysis of charge injection in MOS analog switches," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 2, pp. 277-281, 1987.
- [101] A. M. G. Jackson, A. Hamilton, "Pulse stream vlsi neural systems: Into robotics," in *International Symposium on Circuits*, vol. 4, pp. 375–378, IEEE, 1994.
- [102] A. Iwata and M. Nagata, "A concept of analog-digital merged circuit architecture for future VLSIs," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E79A, no. 2, pp. 145–157, 1996.
- [103] J. Alspector, J. Gannett, S. Haber, and M. Parker, "Generating multiple analog noise sources from a single linear feedback shift register with neural network applications," in 1990 IEEE International Symposium on Circuits and Systems, pp. 1058-61, IEEE, 1990.
- [104] T. Lehmann, "Implementation issues of self-learning pulsed integrated neureal systems," in 13th NORCHIP Conference, (Copenhagen, Denmark), pp. 145-152, 1995.
- [105] K. R. Lakshmikumar, R. A. Hadaway, and M. A. Copeland, "Characterization and modeling of mismatch in MOS-transistors for precision analog design," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 6, pp. 1057–1066, 1986.

- [106] P. O. Leary, "Practical aspects of mixed analogue and digital design," in Analogue-Digital ASICs: circuit techniques, design tools and applications (F. M. R. Soin and J. Franca, eds.), vol. III, pp. 213-238, IEE Circuits and systems series, 1989.
- [107] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–9, 1989.
- [108] M. McNutt, S. LeMarquis, and J. Dunkley, "Systematic capacitance matching errors and corrective layout procedures," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 5, pp. 611–16, 1994.
- [109] R. Singh and A. Bhattacharyya, "Matching properties of linear MOS capacitors," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 3, pp. 465– 7, 1989.
- [110] C. Michael and M. Ismail, "Statistical modeling of device mismatch for analog MOS integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 2, pp. 154–66, 1992.
- [111] C. Conroy, W. Lane, M. Moran, and Lakshmikumar, "Comments, with reply, on 'characterization and modeling of mismatch in MOS transistors for precision analog design'," *IEEE Journal of Solid-State Circuits*, vol. 23, no. B20, pp. 294--6, 1988.
- [112] F. Forti and M. Wright, "Measurement of MOS current mismatch in the weak inversion region," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 2, pp. 138–42, 1994.
- [113] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. Fort Worth: Holt, Rinehart and Winston Inc., 1987.
- [114] D. H. Philip Allen, CMOS Analog Circuit Design. Saunders College Publishing, 1987.
- [115] C. M. Wu and K. W. Yeh, "Safety drain voltage avoiding avalanche breakdown in MOSFETs," *Electron Device Letters*, vol. 3, no. 9, pp. 245–247, 1982.
- [116] B. Buchan, Standard CMOS Floating Gate Memories for Non-Volatile Parametersiaion of Pulse-Stream Radialk Basis Function Neural networks. PhD thesis, dpt. of Elecrical Engineering, University of Edinburgh, November 1997.
- [117] Intel-Embeded Microcontrollers and Processors, intel ed., 1992.
- [118] H. Traff, "Novel approach to high speed CMOS current comparators," *Electronics Letters*, vol. 28, no. 3, pp. 310–12, 1992.
- [119] G. Palumbo, "CMOS current comparator simplified analysis of the delaytime," International Journal of Circuit Theory and Applications, vol. 22, no. 2, pp. 157-162, 1994.

- [120] G. Palmisano and G. Palumbo, "High performance CMOS current comparator design," *IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing*, vol. 43, no. 12, pp. 785–790, 1996.
- [121] A. Tang and C. Toumazou, "High performance CMOS current comparator," *Electronics Letters*, vol. 30, no. 1, pp. 5–6, 1994.
- [122] D. R. Frey, "Log-domain filtering an approach to current-mode filtering," *IEE Proceedings-G Circuits Devices and Systems*, vol. 140, no. 6, pp. 406–416, 1993.
- [123] D. R. Frey, "Log domain filtering for RF applications," IEEE Journal of Solid-State Circuits, vol. 31, no. 10, pp. 1468-1475, 1996.
- [124] C. Toumazou and A. Payne, "Analogue electronics—log domain processing and its applications," *Electronic Engineering*, vol. 68, no. 840, pp. 45–46, 1996.
- [125] E. Seevinck, "Companding current-mode integrator: a new circuit principle for continuous-time monolithic filters," *Electronics Letters*, vol. 26, no. 24, pp. 2046–7, 1990.
- [126] M. Punzenberger and C. Enz, "A new 1.2v BiCMOS log-domain integrator for companding current-mode filters," in *International Symposium on Circuits and Systems — Circuits (ISCAS)*, (Atlanta, Ga, 12 — 15 Ma), pp. 125– 128, IEEE, 1996.
- [127] J. Mahattanakul and C. Tomazou, "Instantaneous expressing and companding: A dual approach to linear integrator synthesis," in *Colloquium on Analogue Signal Processing*, (Oxford,UK), November 1996.
- [128] Y. Tsividis, "Instantaneously companding integrators," in International Symposium on Circuits and Systems (ISCAS), vol. I, (Hong Kong), pp. 447– 450, IEEE, June 1997.
- [129] C. T. J. Mahattanakul, "A non-linear design approach for high frequency linear integrators," in *International Symposium on Circuits and Systems* (ISCAS), vol. I, (Hong Kong), pp. 485–488, IEEE, June 1997.
- [130] Y. Tsividis, V. Gopinathan, and L. Toth, "Companding in signal processing," *Electronics Letters*, vol. 26, no. 17, pp. 1331–2, 1990.
- [131] F. J. Yang, C. Enz, and G. Vanruymbeke, "Design of low-power and low-voltage log-domain filters," in *International Symposium on Circuits and Systems Circuits*, (Atlanta, Ga, 12 15 Ma), pp. 117–120, IEEE, 1996.
- [132] B. Gilbert, "Current-mode circuits from a translinear viewpoint: a tutorial," in Analogue IC design: the current-mode approach (F. L. C. Tomazou and D. Haigh, eds.), vol. II, ch. 2, pp. 10–120, London: IEE Circuits and systems series, 1990.
- [133] M. Punzenberger and C. Enz, "Low-voltage companding current-mode integrators," in Symposium on Circuits and Systems, pp. 2112-15, IEEE, 1995.

- [134] M. Punzenberger and C. Enz, "A 1.2v BiCMOS class AB log-domain filter," in *International Solid-State Circuits Conference*, vol. 40, (San Francisco, Ca, Feb), pp. 56-57, IEEE, 1997.
- [135] D. P. C. Enz, M. Punzenberger, "Low-voltage log-domain signal processing in CMOS and BiCMOS," in *International Symposium on Circuits and Sys*tems (ISCAS), vol. I, (Hong Kong), pp. 489–492, IEEE, June 1997.
- [136] I. E. Opris and G. T. A. Kovacs, "Large-signal subthreshold CMOS transconductance amplifier," *Electronics Letters*, vol. 31, no. 9, pp. 718–720, 1995.
- [137] J. Mahattanakul and C. Toumazou, "Independent control of transconductance gain and input linear range in a MOS linear transconductance amplifier," *Electronics Letters*, vol. 32, no. 18, pp. 1629–1630, 1996.
- [138] R. Sarpeshkar, R. F. Lyon, and C. Mead, "A low-power wide-linear-range transconductance amplifier," Analog Integrated Circuits and Signal Processing, vol. 13, no. 1–2, pp. 123–151, 1997.
- [139] "New products showcase," Texas Instruments, September 1997.
- [140] J. Mulder, A. C. Vanderwoerd, W. A. Serdijn, and A. H. M. Vanroermund, "Current-mode compounding root-x-domain integrator," *Electronics Letters*, vol. 32, no. 3, pp. 198–199, 1996.
- [141] R. Fried, D. Python, and C. C. Enz, "Compact log-domain current-mode integrator with high transconductance-to-bias current ratio," *Electronics Let*ters, vol. 32, no. 11, pp. 952–953, 1996.
- [142] R. Fried and C. Enz, "A family of very low-power analog building blocks based on CMOS translinear loops," in *IEEE-CAS Workshop on Analog and Mixed IC Design, Baveno*, (F. Maloberti, A. Baschirotto, and V. Liberali, eds.), (Baveno-Italy, 12-13 Sep), pp. 73-78, IEEE, 1997.

Appendix A Palmo FPAA Addressing Registers

8 bit address register





Figure A–1: FPAA address register

This register is used to address the FPAA chip:

- A_0 and A_1 bits are used for internal cell addressing, to select in between the DAC and capacitor array registers.
- A_2 , A_3 and A_4 determine the cell number.
- A_5 can be used for accessing the two FPAAs on board.
- A_6 and A_7 are used to access the global interconnect.

	D7	D6	D5	D4	D3	D2	DI	D0
Upper interconnect data register	Reset 3	\times	\times	Ramp 6-7	Int 6-7	Int 4-5	Ramp 4-5	Input
Lower interconnect data register	Ramp 3-7 Int 3-7	Int 3OP	Ramp 30P	Ramp 2-3	Int 2-3	Int 0-1	Ramp 0-1	Ramp 0-4 Int 0-4

Figure A-2: FPAA interconnect registers

The FPAA interconnect registers control the global interconnect switches shown in figure 5–1.



Figure A-3: Typical cell Capacitor and DAC registers

There are three SRAM registers (addressed by the bits A_0 and A_1 of the address register) local to every cell, which store the I_{int} and I_{ramp} DAC configurations and the capacitor interconnect (figure 5–2 in section 5.3.1).

Appendix B Palmo FPAA Pin Out

Description	Pad #	PGA	Pad-Type
ALE	1	B2	CPDI
Up/Down4 / D0	2	C2	IC
Int/Ramp4 / D1	3	B1	IC
Up/Down5 / D2	4	C1	IC
Int/Ramp5 / D3	5	D2	IC
Int/Ramp6 / D4	6	D1	IC
Up/Down6 / D5	7	E3	IC
Up/Down7 / D6	8	E2	IC
Int/Ramp7 / D7	9	E1	IC
Reset2	12	G3	IC
INPUT	13	G1	PFPPD
Pulse5	14	G2	IIC
Out6	16	H1	OI2
Out5	17	H2	OI2
Pulse6	18	J1	IIC
Pulse7	19	K1	IIC
Out7	20	J2	OI2
Vfixn	21	L1	CPAI
Vssa	26	K4	CPVSSA
Vfixp	29	K5	CPAI
Vcomp	30	L5	PFPTR
Vssa	32	J6	VSSA
VDAC	33	J7	PFPPD
OPAMP	34	L7	PFPPD
Vdda	36	L6	VDDA
Outcomp	37	L8	OI2

Table B–1. Palmo FPAA Pin out part I

Description	Pad #	PGA	Pad-Type
FiComp	38	K8	
Out3	39	L9	OI2
Nin	40	L10	CPAI
Vssa	41	K9	CPVSSA
PIN	43	K10	CPAI
CompPulse	44	J10	IC
MinOut	45	K11	OI2
Pulse3	46	J11	IIC
Int/Ramp3	47	H10	IC
Up/Down3	48	H11	IC
Up/Down2	50	G10	IC
Int/Ramp2	51	G11	IC
Pulse2	52	G9	IIC
Out2	55	E11	OI2
Out1	56	E10	OI2
Pulse1	57	E9	IIC
Int/Ramp1	58	D11	IC
Up/Down1	59	D10	IC
Up/Down0	60	C11	IC
Int/Ramp0	61	B11	IC
Pulse0	62	C10	IIC
Reset1	63	A11	CPDI
Vssa	65	B9	CPVSSA
Load	67	A9	CPDI
Vdd 5V	68	B8	VDD
Vddh	70	B6	PadCon
Vss	74	C6	VSS
Out0	76	A5	OI2
Vref	77	B5	PVREF
Out4	79	A4	OI2
Pulse4	80	B4	IIC
Reset0	81	A3	CPDI
Vssa	83	B3	CPVSSA

Table B-2. Palmo FPAA Pin out part II

÷

Appendix C Microcontroller Code

An ATML 80C2051 20 pin microcontroller is used, for controlling the serial communication link between the board and the host PC. The same microcontroller downloads the configuration data to the FPGA, during the initialisation procedure.

C.1 Commands

There is an on-line help available by the microcontroller software, by pressing 'h' or '?'. The commands available to the user are:

D	xx	(xx)	: Sents bytes to the DAC
F	xd	xa :	Sents the byte 'xd' to the FPGA address 'xa'
H	or	?:	Prints this hepl
I	xa	:	Inputs P1
0	xd	:	Outputs 'xd' to P1
Q	:		Shuts down the PALMO board
R	:		Reads back DAC registers
X	:		Downloads configuration to the FPGA

(Were 'xi' is a HEXIMAL number in ASCII format)

C.2 Microcontroller Code

In the following pages, the reader can find a listing of the code which is downloaded into the microcontroller *Flash EEPROM*. The microcontroller is an *Intel* 8051 [117] compatible chip. The microcontroller code has interesting subroutines to perform automatic BAUD detection program the FPGA and configure the DAC.

Apr 16 1998 17	2:07:43	Mic	rocontroller Code	Apr 16 1998 1	7:07:43	Microcon	troller Code
; Define the I	/O pins done.	on the 8051 90h		cmd5:	cjne ajmp	a,#'F',cmd6 FPGA	; Download Configuration DATA ; To the FPGA
.equ .equ	cclk, din, prog,	91h 92h 93h		cmd6:	cjne ajmp	a,#'?',cmd7 help	; Print HELP
.equ .equ	ипіс, M1, M0,	94n 95h 96h		cmd8:	cjne ajmp	a,#'H',cmd8 help	; Print HELP
. equ . equ . equ	DAC_CS OFF, OFF_pr	, 96h 97h og, 88h		cmd9:	cjne ajmp	a,#'R',cmd9 readback a.#'O',cmd.error	; Readback from DAC
. equ . equ . equ	ldFPGA AdrPalı DatPalı	, 0xb3 no, 0xb4 no, 0xb5			mov acall	DPTR,#byemsg print	
. equ	fclk,	94h	: Begin Address after Reset	OFFloop:	clr ajmp	OFF OFFloop	; switching the board OFF
.org	ljmp 33h	main	; End of interrupt vectors	cmd_error:	push mov	acc dptr,#cmderr_msg	; If it has come here then there
; print the we main:	clr mov	ie.7 SP,#30h	; Initialize stack Pointer		pop acall sjmp	acc putc commandloop	, was an error reported
	mov mov	P1,#0ffh P3,#0ffh	;; Make sure that P1 has the state it ;; would have after reset (usefull only ; for use with PAULMON)is OK.	prompt: welcome: NLCR:	.db "\ .db "\ .db "\	n\r>",0 rPalmo Board Downloader" n\r",0	
	acall	autobaud		cmderr_msg: byemsg:	.db "\ .db "\	n\rNo such command: ",0 n\rBye\n\r",0	
	clr clr	AdrPalmo DatPalmo	; Initialize the palmo Control lines ;	helpmsg:	.db "\ .db "	n\n\r" D xx (xx) : Sents	bytes to the DAC\n\r"
wlcmLOOP:	mov mov acall djnz	RI,#25 dptr,#NLCR print R1,wlcmLOOP	; Clears the screen		.db. .db. .db. .db. .db.	r xa xa : sents "FPGA H or ? : Prints I xa : Inputs O xd : Output Q : Shuts P - Poode	the byte "xa'\n\r" address 'xa'\n\r" ; this hepl\n\r" ; Pl\n\r" ; s 'xd' to Pl\n\r" down the PALMO board\n\r" back Da registers\n\r"
	mov acall	dptr,#welcome print	; Prints the welcome message		.db " .db "\ .db "\	X : Downlo n(Were 'xi' is a HEXIMA n\r",0	ads configuration to the FPGA\n\r" L number in ASCIIformat)\n\r"
; prompt for c commandloop:	ommands mov	and command inte dptr,#prompt	rpretation ;print prompt	; XILINX confi	iguration	routines	
getcmd:	acall cjne sjmp	getc a,#'\n',cmd0 getcmd	;wait for serial input	not_present:	mov acall ret	dptr,#no_fpga_msg print	; Error message because the ; INIT line did not go low
cmd0:	acall	putc		file_err:	mov	dptr.#ferr msg	: The chars transmied were not
cmd1:	cjne sjmp	a,#'\r',cmd1 commandloop	; Check for "ENTER"		acall ret	print	; a XILINX BIT file
	cjne acall ajmp	a,#'X',cmd2 xilinx commandloop	;If command='X' download ;XILINX configuration	xilinx:	mov acall	dptr, #xilwelcome print	; Prins message
cmd2 :	cjne ajmp	a,#'D',cmd3 setDAC	; Sent DATA to the DAC				; 00,09,0F if the first three ; bytes do not indicate a : bitstream ABORT
cmd4:	cjne ajmp	a,#'0',cmd4 output	; Set a MC port		acall cjne acall	getb a,#00,file_err getb	
	cjne ajmp	a,#'I',cmd5 input	; Read a MC port			-	

152

Microcontroller
Code

Apr 16 1998 17:07:43 Microcontroller Code

Million A. Constant	Marshall manual (Car	Section 2	C 688862 - 20155 - 2017	Contraction of the second second	infilian			2000 C 1000 C 10	30003024
		cjne	a,#9,file_e	rr	;	Checking for the validity		mov	r5 b
		cine	a.#0fh.file	err	'	or the bittile		mov	r6
								mov	r7
	;assume	the bit	file is arr	iving we	bet	tter		mov	r0
	;start :data s	the prog starts to	gram sequence show up	berore the	rea	a1	sh_loop	:	14
	,							rlc	a
		setb	din		;	Initialise all inputs		mov	di
		setb	cclk			(Port 1 has floatting Gates		nop	
		setb	done		;	with Internal 10k pul-ups)		nop	
		setb	M1 M0				;	acall setb	c1 cc
		5005						djnz	r0
		clr	prog		;	Send the PROGRAM command		nop	
		mov dinz	r0, #240 r0 *		;	Which resets and clears the FPGA and delay for reset		nop nop	
			10,		'	The and deley for report	;	acall	cl
		jb	done, not_p	resent				jnb	in
		jb olr	init, not_p	resent	;	Check for Presence of FPGA		nop nop	
		CII	um					jnb	do
•		setb	prog		ï	Set PROG back to high		- -	
					;	to enable programming		, now to	nat nuls
;xilin:	x bitstre	am downl	Loader, Paul	Stoffregen,	Mas	r 1996		; in 19	93 x
;beta	version 0).2		_				setb	di
.0005	screwed	up 4000	series downly	oad in versi	ion	0.1. fixed now	startup	mov :clr	ru
, 0553,	BCLEWEG	up 4000	Serres downin	uu 1 verbi		····		nop	
;To us	e this th	ing, jus	st connect th	e 8051's uar	rt (to your		nop	-1
;seria.	i port (v necessarv	via drive v lines f	from port1 to	vour xilinx	caci x cl	n the	,	setb	CL
;Just	send the	binary .	bit file dir	ectly to the	e se	erial		djnz	r0
;port	(at the r	ight bau	id rate) and	this code wi	ill	remove		mov	dp
;cne .; xilin:	oit file' x chip.	There is	s little to n	o error chec	ckir	ng for		acall	de
;3000 :	series pa	rts, so	be careful.					acall	de
;Pau	ul Stoffr	egen (pa	aul@ece.orst.	edu)				acall	de
;to do	:							ret	
; -> ao	dd status	led out	puts -downlo	ad "started"	" aı	nd "finished"			
;-> ci ; ai	neck for nd an int	anotner o high i	mpedance unt	il it's fini	ing ishe	a download	error:	;init	low
; -> a	dd mode i	nput (sh	nown on schem	atic) and it	t ma	aster serial			
; ma	ode is se	lected g	general progra	am pulse in	res	sponse to		mov	a,
; -> be	eset pin etter mes	sages to	host compute	er in the un	ali	e work kelv event		mov	a,
; 50	omeone is	using a	terminal pro	ogram and ac	ctua	ally reading it		acall	pu
; -> ac	dd jumper	and ext	ra line to O	E/R pin on s	ser	ial prom and		acall	pu
; -> cl	neck that	init si	ignal goes is	low before	we	see it go		acall	pu
; h:	igh so th	e chip-n	not-connected	error can b	oe o	letected		mov	a,
; 11 ; Wi	nstead of as succes	sful bec	ause of pull	to nownere a -up resistor	ana rs.	thinking it		mov	pr. a,
,								acal1	pr
	setb	done dob #0	h					mov	a, pu
	1000	ари, то	•					mov	a,
wait25	5:				1	- Even the seriel link		acall	pr
	acall inc	getb dotr		; Gets	pyt	te from the serial link		acall	a, pu
	cjne	a, #255	, wait255	; the F	REAL	L data for the FPGA are		mov	a,
	mov	r2, #0		; furth	her	down the line when the		acall	pr
next.	sjmp acall	snift geth		; data	bed	Jons a series of "FF"		acall	а, рц
	inc	dptr		; MAIN	FPO	GA LOOP		mov	a,
shift:	mov	b, r5		; r5,6,	,7 8	are used to report an error		acall	pr
	MON	b, r6		; 11 CD	. e.	TWIT TIME GOES HIGH	ļ		

Apr 16	1998 17:	07:43	Microcontroller Code
	mov mov	r5, b b, r7 r6, b	; Shifting is needed to update r5-7
	mov mov inc	r7, a r0, #8 r2	; Set counter for programming 8 bits
sn_100p	rlc mov clr	a din,c cclk	; Rotate to get bit ; Output bit ; Pulse cclk
;	nop nop acall setb	clkdelay cclk	; delay for the FPGA ; Stop cclk PULSE
	djnz nop nop pop	r0, sh_loop	; repeat 8 times for 8 bits ; delay
;	acall jnb nop	clkdelay init, error	; Check the INIT line for an ; error reported by the FPGA
	nop jnb ;now th	done, next at it says it's d	; IF DONE went high the bitstream ; was sucessfully downloaded done, it needs a few more
	;cclk p ;in 199 setb	ulses to actually 3 xilinx databook din 	y start up see pg 2-29 k. ;in case a 3000 series .bit file left it low
startup	nop nop	cclk	, need more cork purses to rantan startup
;	acall setb djnz mov	clkdelay cclk r0, startup dptr, #msg_done	
	acall acall acall acall	print delay delay delay	
	ret	-	; FPGA initialised
error:	;if we ;init l	get here, it mean ow to tell us it	ns the xilinx chip pulled got a checksum error!
	mov acall mov	a, #'E' putc a, #'r'	
	acall mov acall	putc putc a, #'@' putc	
	mov acall mov acall	a, dph prhex a, dpl prhex	;offset in .bit file where error detected
	mov acall mov	a, #' ' putc a, r4 prhex	
	mov acall mov	a, #' ' putc a, r5	
	acall mov acall mov	prnex a, #' ' putc a, r6	
	acall	prhex	

153

Apr 16 1998 17	7:07:43	Mic	rocontroller Code)	Apr 16 1998 1	7:07:43	Mic Mic	rocontroller Code
mov acall mov	a, #' ' putc a, r7				sendDACbyte:	clr clr	cclk DAC_CS ;set	;Prepare cclk for transmition ; Set DAC CS low
acall mov acall acall acall	prhex a, #' ' putc putc putc				sendDACbits:	mov rrc mov setb	r0,#8 a din,c cclk	;set counter r0 for 8 bits ;rotate to get the bit ;output the bit ;Up-Going clock
acall ret	a, 12 prhex				;	nop acall clr djnz	clkdelay cclk r0,sendDACbits	wait for propagation delays ;zero cclk ;Repeat 8 times for 8 bits
<pre>msg_done: xilwelcome: no_fpga_msg: ferr_msg: ;</pre>	.db .db .db .db .db	"Done signal wen 13,10,"Xilinx B "FPGA not prese "Transmited fil 10,0	t high.",13,10,0 itstream Downloader",1 nt check connections", e is not a XILINX 4000	3,10,0 13,10,0 bit-stream",13	readback:	ret mov acall	DPTR, #NLCR print	
delay: mov delay2: nop mov djnz dinz	r3, #20 r2, #22 r2, * r3 del	0 8 av2				mov acall setb	a,#/En sendDACbyte cclk	;sent readback command ;sent a dummy pulse needed ;by the DaC
output:	acall	rdbyte	; Get the dat	a byte	;	nop acall clr	clkdelay cclk	;delay ;end of cclk pulse
	jc orl mov	outabort a,#OFF_prog P1,a	; if not hex	abort	· rdbckbytes:	setb mov	din r0,#4 ~1 #8	When a pin is set it can be ;used as an input ;Set counter for 4 bytes
outabort:	mov acall ajmp	DPTR,#outabortm print commandloop	sg ; Report erro	r	rdbckloop:	mov acall setb	a,#' ' putc cclk	; print space ; clock DAC
outabortmsg: input:	.db	"\n\rOutput abo	rted !",0		;	nop nop acall mov	clkdelay c,din	;get bit from DAC
	mov acall mov	a, # putc a, P1 prbyte	; Print a spa	ce cnar		rlc djnz acall	ccik a rl,rdbckloop prbyte	;stop the clock pulse ;rotate a to obtain the byte ;repeat 8 times for 8 bits ;Print the byte to the output
help:	ajmp mov acall	commandloop DPTR, #helpmsg print	; Print the h	elp message		djnz setb setb	r0,rdbckbytes DAC_CS cclk	Repeat 4 times for 4 bytes; Reset control lines to inputs;
setDAC:	ajmp	commandloop			FPGA:	ajmp clr	fclk	;Sent DATA to the FPGA
	acall jc mov acall jc	rdbyte setDACend b,a rdbyte setDACend	;Get the DAC command ;If not hex exit ; Store temporary the ; Get the data ; if not hex exit	command		acall jc acall clr clr nop	rdDyte ExitFPGA sendFPGA ldFPGA AdrPalmo	;Get the address byte ;if not hex exit ;sent adress to the the FPGA ;clear the FPGA LATCH for internal adress ;Sent the ALE pulse to the PALMO chip
	push mov acall	ACC a,b sendDACbyte	; save temporarry the ; get the command dat ; sent command	a		nop nop nop setb	AdrPalmo	;Wait even longer for PALMO chip delays ;Clear the pulse
setDACend:	pop acall setb	ACC sendDACbyte DAC_CS	; get temporary saved ; sent data ; restore pins	data		acall jc acall	rdbyte ExitFPGA sendFPGA	;Get the DATA byte ;Sent it to the FPGA
	setb lcall	cclk commandloop				clr	DatPalmo	;Pulse Palmo LOAD

•

Microcontroller Code

154

.

		nop nop nop			;long wait
		setb	DatPalmo	>	;Done senting the byte
ExitFPG	A:	setb setb ajmp	ldFPGA fclk commandI	loop	;set FPGA LATCH & fclk to high again
sendFPG	A:	clr	fclk		;Prepare fclk fro pulsing bits in
sendFPG	Abits:	mov rlc mov setb nop	r0,#8 a din,c fclk		;set r0 to 8 for 8 bits ;rotate to obtain the bit ;output the bit ;clock the FPGA
		nop	-11-9-1	_	;wait
;		acall clr djnz	fclk fclk r0,send	7 FPGAbits	;zero the fclk pulse ;Repeat 8 times
		ret			
;clkdela	ay: push	acc			
; ;clkloop	mov o: djnz	a,#10 acc,clk:	Loop		
;	pop ret	acc	-		
prbyte:	push swap anl acall pop push anl acall pop ret	acc a, #0fh prhex acc a, #0xfh prhex acc		; save A ; swap t ; Mask H ; Print ; Restor ; and sa ; Mask a ; Restor	ACC co prin MSByte first pefore printing DIGIT 1 re ACC ave it to return with it and print DIGIT 2 re ACC and return
print: print1:	push mov acall inc çjne pop	acc a,#0 a,@a+dp1 putc dptr a,#0,pr: acc	r intl	<pre>;Save th ;zero po ;get the ;print t ;increas ;if the ;'\0' pr ;restore</pre>	ne ACC value pinter to get the first char a character the char se the pointer char is not the end of string fint next char a ACC
pute:					
	jnb mov clr ret	scon.1, <u>1</u> sbuf,a scon.1	outc	;wait ur ;transmi ;clear s	ntil prev byte is send t a byte scon for next byte
getb:	jnb clr mov ret	ri,getb ri a,sbuf		;wait ur ;clear f ;get byt	ntil the byte is recieved lag for nex byte re
getc:	jnb clr mov cjne	ri,getc ri a,sbuf a,#60h,g	jetc_nEQ	;get cha ;check i	ir f low-case
	rec			, chai 18	, (ov recurii

Apr 16	1998 17	07:43 Microcontroller Code				
fixasc:	ret subb ret	;char is upper case, return a,#20h ;subb 32 to make low-case upper-case				
prhex:	mov movc acall ret	dptr,#HEXDAT a,@a+dptr ;get the char ponted by HEXDAT+a putc ;print it				
HEXDAT:	. DB	"0123456789ABCDEF"				
rdbyte:		;Reads a byte (2 characters) from the serial link ; on return if failed C is set (all the other flags ; are undefined) ; and ACC contains the ASCII character that came in ; on success C=0 and ACC contains the Byte ; (Not a HEX number)				
rdbyte_	acall acall cjne sjmp 1.	getc ; Get the char				
		a,#' ',rdbyte_1 ; Skip if it is space rdbyte				
	push acall swap mov	ACC ; Save input (incase of an error). hexdig ; THe actual transformation from ASCII to the ACC a ; Make the 4 bits MSBits (they were in first) 123,a ; Save it temporaly				
	inc iz	a ; If ACC is 0xff there was an error in the HEX no rdbyterr				
	pop	ACC ; No error occured so input is useless				
	acall acall push acall	getc putc ACC ; Save input (incase of an error). hexdig				
rdbyter	cjne	a,#0ffh,exitrdbyte ; Chech 0xff for HEX error				
	pop setb ret	ACC ; Set C to report error the error in HEX no				
exitrdby	yte: orl	a,123 : Fix the returned Byte SP ; get rid of the saved input for possible error c ; Report is OK.				
	dec clr ret					
hexdig:	clr subb jc l cjne i ret	; Returns ASCII to hex or 0xff on error				
		c a,#'0' ; Subtract 48 from the ASCII input hexdigerr ; If char < 48 report error a,#9,hexdig_1 ; Compare input to 9				
hexdig_	l: jnc ret	hexdig2 ;if input > 9 check for A-B-C-D-E-F ;Char <9 OK.				
hexdig2	: alr					
	subb jc add cjne ret	a,#'A'-'0' ; Clear the byte if >= A hexdigerr ; IF one of chars < '0' or <'A' error a,#10 ; Put back the 10 which is missing in the number a,#15,hexdig_2 ; Compare to 15				
hexdig_:	<pre>inc inc ret</pre>	hexdigerr ; If ACC >15 error				

Apr 16 1998 17:07:43 Microcontroller Code hexdigerr: : Error occured return 0xff a.#0xff mov ret ; To set the baud rate, use this formula or set to 0 for auto detection ; baud_const = 256 - (crystal / (12 * 16 * baud)) baud_const, 0 ;automatic baud rate detection . eau baud const. 255 :57600 baud w/ 11.0592 MHz ; . equ :19200 baud w/ 11.0592 MHz baud_const, 253 ;.equ baud const, 252 ;19200 baud w/ 14.7456 MHz ;.earu ;4808 baud w/ 12 MHz ;.equ baud const, 243 ; to do automatic baud rate detection, we assume the user will press the carriage return, which will cause this bit pattern ;to appear on port 3 pin 0 (CR = ascii code 13, assume 8N1 format) : 0101100001 : ; start bit----+ +--lsb msb--+ +----stop bit ;we'll start timer #1 in 16 bit mode at the transition between the start bit and the LSB and stop it between the MBS and stop bit. ;That will give approx the number of cpu cycles for 8 bits. Divide ; by 8 for one bit and by 16 since the built-in UART takes 16 timer ;overflows for each bit. We need to be careful about roundoff during :division and the result has to be inverted since timer #1 counts up. Of ; course, timer #1 gets used in 8-bit auto reload mode for generating the ; built-in UART's baud rate once we know what the reload value should be. autobaud: tmod, #0x11 ;get timer #1 ready for action (16 bit mode) mov tcon, #0x00 mov clr а th1, a mov tl1, a mov a, #baud_const ; skip if user supplied baud rate constant mov autoend jnz autob2: jb p3.0, * ;wait for start bit p3.0, autob2 jb ĵЪ p3.0, autob2 ; check it a few more times to make jb p3.0, autob2 : sure we don't trigger on some noise jb p3.0, autob2 p3.0. * :wait for bit #0 to begin jnb ; and now we're timing it setb tr1 p3.0, * ;wait for bit #1 to begin jb jnb p3.0, * wait for bit #2 to begin p3.0, * ;wait for bit #4 to begin jb p3.0, * ;wait for stop bit to begin jnb clr tr1 ;stop timing mov a, tll c, acc.6 ; save bit 6 for rounding up if necessary mov mov f0, c grab bit 7... it's the lsb we want c. acc.7 mov mov a, th1 rlc :do the div by 128 а c, f0 mov addc a, #0 ;round off if necessary ; invert since timer #1 will count up cpl а inc а :now acc has the correct reload value (I hope) autoend: a,#252 ;mov mov th1, a tl1, a mov mov tmod, #0x21 ;set timer #1 for 8 bit auto-reload pcon, #0x80 ; configure built-in uart mov scon, #0x52 mov ;start the baud rate timer setb tr1 :mov P1.a

Apr 16 1998 17:07:43 Microcontroller Code

retSB: ret

156

Appendix D

Board Documentation

D.1 Initialisation

By applying power and pressing the "On" button the board turns on. The automatic BAUD detection will calculate the asynchronous communication BAUD rate, by pressing "*Return*" on the host PC. When connection is established the welcome message will be printed on the terminal screen:

Palmo Board Downloader >_

The user has to download the XILINX bitstream to configure the FPGA, by the use of the 'X' command. Finally the FPMA chip must be set in an idle state by the use of the following commands:

F01 00	F02 00	F04 00	F05 00	F06 00
F08 00	F09 00	F0A 00	F0C 00	F0D 00
F0E 00	F10 00	F11 00	F12 00	F14 00
F15 00	F16 00	F18 00	F19 00	F1A 00
F1C 00	F1D 00	F1E 00	F80 00	FC0 80

D.2 Board Schematic Diagrams

The board schematic diagrams are presented in the following two pages.





Appendix E

FPGA Filter Schematics



Figure E-1: Second order *Palmo* filter implementation—FPGA schematic. It includes digital logic to drive the *Palmo* inputs, *signed PWM* and *Ramp* generation.







Figure E-3: signed PWM output generation

.



Figure E-4: Ramp generating cell
Appendix F

Publications

- 1. K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in *International Symposium on Circuits and Systems (ISCAS)*, vol. I, (Atlanta), pp. 270–273, May 1996.
- K. Papathanasiou and A. Hamilton, "Palmo signal processing: VLSI results from an integrated filter," in *The Third IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, vol. 2, (Rhodes-Ellas), pp. 832-835, October 1996.
- K. Papathanasiou and A. Hamilton, "Novel palmo analogue signal processing IC design techniques," in *Colloquium on Analogue Signal Processing*, (Oxford,UK), pp. 5/1-5/6, November 1996.
- K. Papathanasiou and A. Hamilton, "Advances in programmable pulse based mixed-signal processing VLSI," in *IEEE-CAS Workshop on Analog and Mixed IC Design, Baveno*, (F. Maloberti, A. Baschirotto, and V. Liberali, eds.), (Baveno-Italy, 12-13 Sep), pp. 106-110, IEEE, 1997.
- 5. K. Papathanasiou and A. Hamilton, "Novel palmo techniques for electronically programmable mixed signal arrays," in *International Symposium on Circuits and Systems (ISCAS)*, (California), May 1998.
- 6. T. Brandtner, K. Papathanasiou, and A. Hamilton, "A palmo cell using sampled data log-domain integrators.," submitted to the IEE electronics letters, 1998.
- 7. A. Hamilton and K. Papathanasiou, "Preprocessing for pulsed neural VLSI systems," in *Pulsed Neural Networks* (W. Maass and C. Bishop, eds.), MIT Press, in press.

K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in *International Symposium on Circuits and Systems (ISCAS)*, vol. I, (Atlanta), pp. 270–273, May 1996.



A filter may be constructed from a number of differential integrators (Figure 3a) each having an individual scaling factor, K.-For the purposes of analysis, we shall consider the differential integrator and scaler separately.

and $V_{Cint} = V_{Cr}$. Equating 1 and 2 yields an expression for the scaling factor, K, which is defined as the ratio of the







output pulse width to input pulse width.

$$K = \frac{\Delta T_{out}}{\Delta T_{in}} = \frac{C_r}{C_{int}} \cdot \frac{I_{int}}{I_r}$$
(3)

Thus scaling is a function of the ratio of two capacitances Thus scaling is a function of the ratio of two capacitances multiplied by the ratio of two currents. The ratio of capa-citors in equation 3 is fixed at chip design time and can be accurately controlled [13] [14] [15]. The currents in equa-tion 3 can be electrically modified and their ratio can be accurately controlled [16] [17], therefore the scale factor Kis fully programmable and insensitive to absolute values. By allowing signals arriving at the *plus* and *minus* inputs to be continuously integrated, the resultant output signal, *out*, is a scaled, pulse width modulated representation of the integrated signal.

. 4. CIRCUIT DETAILS

As seen from Figure 1, the circuit building blocks required to implement the *Palmo* filter tap are simple and common-place. The critical structure in the implementation is the charge dump/remove circuit.

place. The critical structure in the implementation is the charge dump/remove circuit. By nsing a standard switching arrangement we would require relatively large currents (10's of μA) and would require careful consideration of switching noise. In con-ventional techniques, a current from a current source is switched on and off using a transistor (or arrangement of transistors) as a switch. During the switching transition a large voltage swing results in *charge injection* into the data-holding capacitor thus corrupting the data. The circuit techniques used here [18] overcome this prob-lem. Instead of switching the current from the current source on and off. this circuit switches the actual current source on and off. This virtually eliminates charge injec-tion. The details of this are shown in Figure 2. When ξ_{-} is high, transistor M1 is on, while M2 is off. This enables the voltage established on the gate of M3 by current Ir/lint to be transistor with a constant current Ir/lint. When ξ_{-} is low, transistor M1 is off, while M2 is on. The voltage on the gate of M4 is now Vss, switching the current transistor the same principles apply to the top half of the circuit. In simulation an input current of 5nA and standard transistor simulation an input current of 5nA and standard transistor inverters were used yet no switching noise was discernible at the output.







Figure 4. Frequency responses of the, z-domain transfer function and Palmo Filter

5. SIMULATION EXAMPLE

A filter has been approximated using this technique and the simple Backward Difference transformation between the s and z domains. The results from the HSPICE simulation of the Palmo filter implementation are very close to the theoretical z domain response as shown in Figure 4 demonstrating the validity of the Palmo filter technique. Other approximations, for example the Bilinear transformation may be implemented by simply changing the contents of the logic block in Figure 1. A fourth of tRLC Butterworth low pass filter with a cut-off frequency of 1kHz (Figure 3b) was approximated

cut-off frequency of 1kHz (Figure 3b) was approximated using the Backward Difference transformation between the s and z domains, where

$$s = \frac{1 - z^{-1}}{T}$$
 (4)

and where T is the sampling interval. The resultant scaling factors for $T = 100\mu S$ are $K_1 = K_4 = -0.821$, $K_2 = K3 = 0.34$. Using these scaling factors the filter structure of Figure 3a was implemented using the Palmo Filter circuit of Figure 1. The appropriate digital logic to generate the signals ξ_+ and ξ_- for the *Backward Difference* transformation are given by the following equations

$$\xi_{+} = S \cdot P \cdot \overline{M} + \overline{S} \cdot \overline{P} \cdot M \quad \xi_{-} = S \cdot \overline{P} \cdot M + \overline{S} \cdot P \cdot \overline{M}$$



version: measurement of input voltage against signed PWM output for two l_r settings. Figure 5. Analogue to signed PWM con-

where S is the sign clock, P is the plus input and M is the minus input.

The frequency responses of the z domain transfer func-tion, H(z), and the resultant Palmo Filter Implementation were calculated. These are shown in Figure 4. The results from the HSPICE simulation of the Palmo Filter Implementation are very close to the theoretical z domain response

6. PRELIMINARY RESULTS FROM VLSI

At the time of writing, our first *Palmo* chip is undergo-ing initial testing. The results presented here are therefore preliminary and it is anticipated that more comprehensive results will be presented at conference. Nevertheless, the results herein demonstrate the functionality of all the circuit components described.

6.1. Analogue to signed PWM conversion

In order to convert an analogue signal to a signed PWM representation, V_{Cint} in Figure 1 was driven via an analogue pad from an external voltage source. The V_{ref} voltage for the ramp waveform (Figure 2) was set to a zener voltage reference of 2.7V. The resultant output pulse (out in Figreference of 2.1V. The resultant output puse (our in Fig-ure 1) was sampled using a digital storage oscilloscope, the pulse width measurement giving the magnitude of the out-put pulse, while the sign of the measurement was defined by the state of the sign clock. Measurements were taken for two *I*, settings indicating that the scaling factor, *K*, can be varied. These results are shown in Figure 5.

6.2. Palmo integrator

The operation of the *Palmo* integrator is illustrated by the oscilloscope traces of Figure 6. The top trace is the input sine wave reconstructed from the signed PWM signal shown in the second trace and generated using the analogue

to signed PWM circuit described above. In order to demonstrate the operation of the integrator, the PWM representation of the sine wave is input to the plus signal of the filter (Figure 1), while the minus input puts signal of the first (Figure 1), while the human sinple is zero. As a precursor to implementing a *Palmo* filter us-ing the *Bilinear transformation*, a delay between the plus and minus inputs of Figure 1 is introduced. This delay is introduced in the logic block of Figure 1. The appropriate digital logic to generate the signals ξ_{+} and ξ_{-} for the *Bi-linear Transformation* are given by the following equations



Palmo integrator. Top trace: reconstructed input sine wave. Second trace: ξ_+ . Fourth trace: PWM output signal. Fifth trace: reconstructed output waveform.

 $\xi_+ = P \cdot S \cdot \overline{B} + M \cdot \overline{S} \cdot B \quad \xi_- = P \cdot \overline{S} \cdot \overline{B} + M \cdot S \cdot B$

where S is the sign clock, P is the plus input, M is the minus input, and B is produced by the delay. The third trace in Figure 6 is the ξ_+ signal, containing every second positive pulse from the integrator input Ing etc.) acom positive part the ξ_{-} has not been shown. The ξ_{+} and ξ_{-} signals are integrated in time, resulting in the PWM coded output signal shown in the fourth trace of Figure 6. The final trace is the output sine wave reco structed from the signed PWM output of the integrator. is the output sine wave recon-

6.3. Evaluation of preliminary results

These results indicate that all the individual components of These results indicate that an the individual components of the first Palmo filter this are functional. Clearly we are at a very early stage in testing, but results obtained so far are extremely encouraging and confirm our simulation results. Further testing will concentrate on assessing the linearity of the integrator, the programmability of the scaling factor, K, and the functionality and performance of filter sections.

7. CONCLUSIONS

This paper presented a new method for implementing fil-ters that is ideally suited to VI.SI. The Palmo Filter build-ing blocks can be used to design any type of sampled fil-ter. Signalling between filter stages is performed using digital pulse width signals which are robust, noise-tolerant, and easily distributed within and between chips. Integand easily distributed within and between chips. Integ-rator scale factors can be set easily using a simple ratio of rator scale factors can be set easily using a simple ratio of currents. Techniques such as the dynamic current mirror may be used to give accurate ratios [19]. Using these tech-niques it is possible to design a 'programmable' filter clip, which would include an array of task. Uniquely, such a chip could be reconfigured to implement any filtering structure

required by the user. Apart from the obvious advantages of using noise tolerant pulses for signal representation, in principle, this approach offers advantages compared to conventional switched capa-citor and switched current implementations. By switching



Publications

K. Papathanasiou and A. Hamilton, "Palmo signal processing: VLSI results from an integrated filter," in The Third IEEE International Conference on Electronics, Circuits and Systems (ICECS), vol. 2, (Rhodes-Ellas), pp. 832-835, October 1996.



¹The name *Palmo* is derived from the Hellenic word $\Pi AAMO\Sigma$ which means pulsebeat, pulse palpitation or

series of pulses



.

Figure 2. Palmo filter tap : integrator, scaler and typical waveform diagram.

ensures the regeneration of the signed pulse representation described earlier.

2.3. Palmo Miller Integrator

In a Miller Integrator the input at the *plus* node is delayed by one clock period to the output. The input at the *minus* is inverted and delayed by onehalf clock period to the ontput. This initiates the need for a delay clock (*B*). The function of the proposed *Palmo* analogue cell (figure 2) is defined by the digital logic block which drives the ξ_+ and $\xi_$ switches. The appropriate digital logic to generate the signals ξ_+ and ξ_- in order to implement a Miller integrator are given by the following equations

$$\xi_+ = P \cdot S \cdot \overline{B} + M \cdot \overline{S} \cdot B$$

$$\xi_{-} = P \cdot \overline{S} \cdot \overline{B} + M \cdot S \cdot B$$

where S is the sign clock, P is the plus input, M is the minus input, and B is produced by the delay. By the use of this digital logic the charge accumulated on the integrating capacitor during one cycle (C_{int}) is given by equation 1.

$$\Delta Q_{Cint}(z) = I_{int} \left(\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2} \right)$$

(1) The voltage (V_{Cint}) on the integrating capacitor at the end of an integrating cycle is given by the following equations:

$$V_{Cint}(z) = V_{Cint} z^{-1} + \frac{\Delta Q_{Cint}(z)}{C_{int}} \iff$$

$$V_{Cint}(z) = \frac{I_{Cint}}{C_{int}} \cdot \frac{\Delta I_{plus} z^{-1} - \Delta I_{minus} z^{-1/2}}{1 - z^{-1}}$$

The voltage (V_{Cint}) accumulated on the integrating capacitor (figure 2) is compared with the ramp (V_{Cr}) in order to regenerate the pulsed output. When the voltage V_{Cr} , becomes equal to the voltage on C_{int} , the comparator output will change state, defining the end of the pulse-width output. In this time the voltage on the ramp capacitor is

$$V_{Cr}(t) = \frac{I_r}{C_r} \Delta T_{out}$$

and
$$V_{Cint} = V_{Cr}$$
. thus

$$\Delta T_{out} = \frac{C_r \ l_{int}}{C_{int} \ l_r} \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}}$$
(2)



Figure 3. Circuit detail of the charge dump/remove stage.

In the well established S–C active RC filter implementation a switched-capacitor replaces R. The transfer function of the S–C Miller integrator with its output sampled on Φ is:

$$V_{out}(z) = \frac{C_u}{C_I} \left(\frac{V_1 z^{-1} - V_2 z^{-1/2}}{1 - z^{-1}} \right)$$
(3)

It is noticeable from equations 2 and 3 that the proposed Palmo basic building block and the S-C Miller integrator have identical transfer functions with

$$K = \frac{C_u}{C_I} = \frac{C_r}{C_{int}} \cdot \frac{I_{int}}{I_r}$$
(4)

Because of this similarity existing S–C synthesis techniques and tools can be applied to the Palmo realisation. On the other hand it is very important to note that scaling (K) is a function of the ratio of two capacitances multiplied by the ratio of two carrents, resulting in greater dynamic range of filter coefficients, compared to conventional S–C (or S–1) techniques. Since the ratio of capacitors in equation (4), can be modified by switching between the elements can be electrically modified, with sufficient accuracy; it is realised that the scale factor K is fully programmable and insensitive to absolute values.

3. NOISE

The main source of noise in the *Palmo* circuit is switching noise from the current dump-remove circuit. Yet as explained in [12, 9] this noise is minimal. In conventional techniques, a current from a current source is switched on and off using a transistor (or arrangement of transistors) as a switch [6, 7, 13, 14]. During the switching transition a large voltage swing results in *charge injection* into the data-holding capacitor thus corrupting the data. In our circuit (figure 3) the sourcing transistors of the current mirrors (Mp and Mn) are turned on and off, switching the actual current source on and off. This virtually eliminates charge injection. In simulation an input current of 5nA and standard transistor inserters were used yet no switching noise was discernible at the output. Test-chip results virty these simulations.



Figure 4. Harmonic Distortion A) Compar-ator inverting stage B) cos(x) C) Minimum pulse effect (f(x)) D) Output distorted signal

3.1. Harmonic Distortion

The non-ideal effects of the input-offset voltage

The non-ideal effects of the input-offset voltage of a typical comparator would generate offsets at the output of the Palmo circuit. While propagation delays could result in Harmonic Distortion. Considering the differential stage of a standard comparator, we can assume that the biassing cur-rent (I_{bias}) flows in one side of the stage or the other (if the differential stage is sufficiently unbalanced). In the onerential stage is sufficiently unbalanced). Therefore it can be assumed that the parasitic load capacitor of the inverting stage (C_{Gp} in figure 4A) is charged or discharged through a constant current, giving approximately the same rise and fall times. However for the inverting stage of the comparator However for the inverting stage of the comparator (figure 4A) there is a significant difference between the positive and negative going delay times. As-suming that the biassing currents are constant the positive delay time (ΔT_{+}) is given by the following equation:

$$\Delta T_+ = C_L \cdot \frac{V_{TRP} - V_{SS}}{I_6 - I_7}$$

While the negative going delay (ΔT_{-}) time is:

$$\Delta T_{-} = C_L \cdot \frac{V_{DD} - V_{TRP}}{l_7}$$

Where V_{TRP} is the trip voltage of the inverting stage. These differences between the positive and negative delay times introduce the effect of adding a constant delay of $\Delta T_{min} = \Delta T_{+} - \Delta T_{-}$ (figure 4) to the positive signals; while the same delay ΔT_{comp} is subtracted from the negative ones, because of our signed pulsed representation. This minimum pulse (ΔT_{min}) effect results in the signal of figure 4D. The Fourier cosine series of the output signal (fig-

ure 4D) is equal to the sum of the Fourier series the two signals shown in figures 4B and 4C thus

 $\Delta T_{out} = \mu \cos(x) + \alpha_0 + \alpha_1 \cos(x) + \alpha_2 \cos(2x) + \dots$ Where

$$\alpha_0 = \int_0^\pi f(x) \, dx = \nu$$

$$\alpha_n = \frac{2}{\pi} \int_0^{\pi} f(x) \cos(nx) \, dx = (-1)^{n+1} \frac{4\nu}{\pi n}$$
(5)

The Total Harmonic Distortion (THD) due to the propagation delay of the comparator is derived from, a_n in (5), for n = 2, 3, 4, ... therefore:

$$THD_1 = \frac{\sqrt{(\frac{1}{2})^2 + (\frac{1}{3})^2 + (\frac{1}{4})^2 + \dots}}{\frac{\pi \mu}{4\nu} + 1} \tag{6}$$



Figure 5. Palmo Integrator linearity.

Where μ is the magnitude of the signed PWM co-sine input and ν is the size of the minimum pulse (ΔT_{min}) . As shown in equation (2) capacitor or cur-rent mismatches generate K mismatches or offsets. However these mismatches do not generate distor-tion. On the other hand mismatches between the transition Mr. Ma of fours 2 will distort the out transistors Mn, Mp of figure 3 will distort the output signal. The *Fourier* transformation of this signal gives a second THD component:

$$THD_2 = \frac{4}{\pi} \cdot \frac{\sqrt{(\frac{1}{3})^2 + (\frac{1}{15})^2 + (\frac{1}{35})^2 + \dots}}{\frac{I_n + I_p}{2 \cdot [I_n - I_p]}}$$
(7)

Where I_n , I_p are the currents driven by the transistors Mn, Mp. For small values of THD_1 and THD_2 the Total Harmonic Distortion of the Palmo circuit can be approximated by adding (6) and (7), thus $THD = THD_1 + THD_2$.

4. RESULTS FROM VLSI

Our first *Palmo* filter chip is currently being tested and all circuit elements are operating as described.

4.1. Integrator linearity

The graph of figure 5 shows the linearity of the Palmo filter for various K factors. In out test chip, the capacitors are fixed while the current sources the capacitors are fixed while the current sources are driven externally, therefore the output of the integrator is dependent upon l_{int} , since l_r is con-stant. The results displayed in figure 5 were taken by applying a number of constant pulses to the *Palmo* integrator and measuring the output. The the output pulses (*out* in Figure 2) were sampled using a digital storage oscilloscope, the pulse width measurement giving the magnitude of the pulse, while the sign of the measurement was defined by the state of the sign clock.

4.2. Palmo filter response

4.2. Paimo inter response In figure 6 the response of a first order Butter-worth filter with a cut-off frequency of 860Hz and a sampling frequency of 8620Hz is presented and it is compared to the ideal. Unfortunately because of the nature of the test chip there is mismatch between the external currents. Yet it is obvious that the differences between the measured and ideal characteristics are small characteristics are small.

characteristics are small. Though our test chip was designed to work at a much higher sampling frequency, our present test-ing equipment limits the measurable frequency. We plan to investigate higher frequencies and bigger fil-ter arrays.



Figure 6. Palmo Filter Response.

4.3. Measured THD

A cosine input of 860Hz was presented to the first order Butterworth filter mentioned earlier. The output of the filter was sampled by the use of a digital scope. The output data was transformed into the frequency domain by the use of a 1024 point *FFT*. From the resultant spectrum a *THD* = 0.86% was calculated .

Was calculated . frequency increases, the integrating and ramp currents will be forced to increase as well, resulting in smaller matching errors and thus decreasing THD_2 . However THD_1 will increase since the pulse ΔT_{min} will increase compared to the maximum output pulse. The reverse effect takes place when the frequency decreases. Therefore the THD is expected to remain stable over a considerable range of frequencies, because $THD = THD_1 + THD_2$; though further investigation is needed to verify the above assumption.

5. CONCLUSIONS

This paper presented a new method for implementing filters that is ideally suited to VLSI. The *Palmo* Filter building blocks can be used to design any type of sampled filter. Signalling between filter stages is performed using digital pulse width signals which are robust, noise-tolerant, and easily distributed within and between chips. Integrator scale factors can be set easily and can be modified dynamically. Techniques such as the dynamic current mirror may be used to give accurate ratios [14]. Using these techniques it is possible to design a 'programmable' filter chip, which would include an array of taps. Uniquely, such a chip could be reconfigured to implement any filtering structure required by the user. The circuit presented in this paper may be used

The circuit presented in this paper may be used to implement a range of signal processing functions By changing the digital logic block in the circuit of figure 2 it is possible to realise the same building block as a scaler, multiplier, analogue memory block or even a non-linear building block. It is obvious that because of the use of pulses as the signalling mechanism a new analogue cell was implemented with unique advantages and an extended application field. The promising results together with some improvements that we are investigating for future testing chips (incorporating reconfigurable capacitor ratios and better current matching); demonstrate that the use of pulses in VLSI circuits, could become an alternative method for programmable analogue signal processing.

REFERENCES

- A. Bratt and I. Macbeth, "Design and implementation of a field programmable analogue array," in FPGA '96, February 1996.
- [2] http://www.impweb.com/, "IMP inc Home-Page," 1996.
- [3] H. Kutuk and S. Kang, "A field-programmable analog array (FPAA) using switched-capacitor techniques," in *International Symposium on Circuits and Systems (ISCAS)*, vol. IV, (Atlanta), May 1996.
- [4] M. B. Y. Tsividis and J. Khoury, "Continuoustime MOSFET-C filters in vlsi," *IEEE trans.* in Circuits Syst, vol. 33, no. 1635, pp. 125-139, Feb 1986.
- [5] N. I. Khachab and M. Ismail, "MOS multiplier/divider cell for analog VLSI," *IEE Electron. Let.*, vol. 25, pp. 1550–1552, Nov 1989.
- [6] M. Ismail and T. Fiez, Analog VLSI signal and information Processing. New York: Mc Graw Hill, 1992.
- [7] M. Banu and Y. Tsividis, "Fully integrated active RC-filters in MOS technology," *ISSCC Digest of Technical Papers*, vol. 26, p. 244, 1983.
- [8] A. Hamilton, A. F. Murray, D. J. Baxter, S. Churcher, H. M. Reekie, and L. Tarassenko, "Integrated pulse-stream neural networks - results, issues and pointers," *IEEE Transactions* on Neural Networks, pp. 385-393, 1992.
- [9] K. Papathanasiou and A. Hamilton, "Pulse based signal processing: Vlsi implementation of a palmo filter," in *International Symposium* on Circuits and Systems (ISCAS), vol. 1, (Atlanta), pp. 270-273, May 1996.
- [10] A. F. Murray and A. Smith, "Asynchronous VLSI neural networks using pulse stream arithmetic," *IEEE Journal of Solid-State Circuits* and Systems, vol. 23, no. 3, pp. 688–697, 1988.
- [11] A. Hamilton, S. Churcher, P. J. Edwards, G. B. Jackson, A. F. Murray, and H. M. Reekie, "Pulse stream VLSI circuits and systems: The EPSILON neural network chipset," *International Journal of Neural Systems*, vol. 4, pp. 395-405, Dec 1993.
- [12] T. Lehmann, "Implementation issues of selflearning pulsed integrated neureal systems," in 13th NORCHIP Conference, (Copenhagen, Denmark), pp. 145-152, 1995.
- [13] T. Fiez, G. Liang, and D. Allstot, "Switchedcurrent circuit design issues," *IEEE Journal* of Solid-State Circuits, vol. 26, no. 3, pp. 192-202, 1990.
- [14] G. Wegmann and E. Vittoz, "Analysis and improvements of accurate dynamic current mirrors," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 699-706, 1989.

Publications

K. Papathanasiou and A. Hamilton, "Novel palmo analogue signal processing IC design techniques," in *Colloquium on Analogue Signal Processing*, (Oxford,UK), pp. 5/1-5/6, November 1996.

NOVEL PALMO¹ ANALOGUE SIGNAL PROCESSING IC DESIGN TECHNIQUES

K. Papathanasiou and A. Hamilton Department of Electrical Engineering, University of Edinburgh

1 Introduction

Following the success of Field Programmable Gate Arrays (FPGAs) in implementing custom digital designs, the use of Field Programmable Analogue Arrays (FPAAs) [1, 2, 3] is about to offer similar benefits for rapid analogue circuit prototyping. Since analogue circuits are difficult to design, layout and test, FPAAs are likely to emerge as a significant new market sector.

The circuits proposed for these new FPAA devices use conventional - predominantly switchedcapacitor (S-C) - techniques. This imposes limitations on the application area of the new concept. The authors believe that novel circuits should be introduced to programmable analogue VLSI to enhance the novel FPAA approach.

Palmo signal processing is a new alternative to traditional switched-capacitor and switchedcurrent analogue signal processing techniques. In the *Palmo* processing technique the analogue input signal is represented by a series of modulated digital pulses rather than a voltage or current. This signal representation is robust, easily distributed, regenerated and rerouted in a chip and is inherently low power. This signal representation opens up new opportunities in programmable analogue circuit implementation where all interconnect is digital and all signal processing is performed by fast, compact, parallel analogue circuits.

This paper presents new *Palmo* circuits connected as an elementary filter tap, consideration of switching noise and analysis of harmonic distortion, results from a VLSI test chip and suggested improvements for a new chip.

2 Field Programmable Mixed-Signal Arrays

The readily available Field Programmable Mixed-Signal Arrays (FPMA) are hybrid chips combining analogue and digital cells. The digital cells are standard FPGA circuits, while the analogue array is usually a standard S–C circuit. The operational amplifier, local and global interconnect have been carefully designed in the light of potential applications. The interconnect will have been carefully laid-out in order to avoid noise from the digital circuits or the switches distorting the analogue signal voltage. A digital interface is needed between the digital and the analogue cells, whose use is mostly unidirectional (in order to drive the S–C switches).

The Palmo implementation of a FPMA offers distinct system level advantages over S-C techniques. Since analogue signals are represented by digital pulse width modulated signals there are no restrictions imposed upon signal routing, as the digital signal has a natural high noise immunity. No special interconnection is needed in order to drive the analogue cells from the digital array as programmable interconnect and cell parameters may be set very simply using static RAM. Since the Palmo circuit does not need an operational amplifier it is possible to realise simple low-voltage, low-power implementations.

¹The name Palmo is derived from the Hellenic word $\Pi A \Lambda MO\Sigma$ which means pulsebeat, pulse palpitation or series of pulses.



$$\Delta Q_{Cint} = I_{int} \cdot \Delta T_{in}$$

The capacitor may therefore act as a memory, storing charge as a result of an incoming pulse or as an integrator, integrating charge resulting from the arrival of a series of pulses.





Figure 2: *Palmo* filter tap : integrator, scaler and typical waveform diagram

Figure 3: Circuit detail of the charge dump/remove stage.

4.2 Multiplier-Scaler

In order to convert the voltage on the integrating capacitor into a pulse, we compare the voltage C_{int} to a ramp voltage waveform generated with a circuit similar to that shown in Figure 1. The resultant output pulse is multiplied or scaled by a factor controlled by the product of a ratio of two currents and a ratio of two capacitors.

$$\Delta T_{out} = \frac{C_r I_{int}}{C_{int} I_r} \cdot \Delta T_{ir}$$

The complete circuit is shown in Figure 2.

4.3 Palmo Miller Integrator

In a Miller Integrator the delay between the input at the *plus* node and the output is one clock period while the input at the *minus* node is inverted and delayed by half a clock period. Therefore a delay clock (B) is required. The appropriate digital logic to generate the signals ξ_+ and ξ_- in order to implement a Miller integrator (figure 2) is defined in the following equations :-

$$\xi_{+} = P \cdot \overline{S} \cdot \overline{B} + M \cdot \overline{S} \cdot \overline{B}$$
$$\xi_{-} = P \cdot \overline{S} \cdot \overline{B} + M \cdot S \cdot B$$

where S is the sign clock, P is the plus input, M is the minus input, and B is produced by the delay. The resultant Z domain transfer characteristic is discussed in [6] and given by equation (1).

$$\Delta T_{out} = \frac{C_r I_{int}}{C_{int} I_r} \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}}$$
(1)

It is noticeable from equation 1 that the proposed *Palmo* basic building block and the S-C Miller integrator have identical transfer functions.

Because of this similarity existing S-C synthesis techniques and tools can be applied to the *Palmo* realisation. In the *Palmo* implementation scaling (K) is a function of the ratio of two capacitances multiplied by the ratio of two currents, resulting in greater dynamic range of filter coefficients, compared to conventional S-C (or S-I) techniques. Since the ratio of capacitors in equation (1), can be modified by switching between the elements of a capacitor array, and an accurate current ratio may be electrically modified, the scale factor K is fully programmable and insensitive to absolute values.





Figure 4: Total Harmonic Distortion

Figure 5: *Palmo* Integrator linearity - results from the test chip.

5 Switching Noise

The main source of noise in the *Palmo* circuit is switching noise from the current dump-remove circuit. Yet as explained in [7, 8] this noise is minimal. In conventional techniques, a current from a current source is switched on and off using a transistor (or arrangement of transistors) as a switch [9, 10]. During the switching transition a large voltage swing results in *charge injection* into the data-holding capacitor thus corrupting the data. In our circuit (figure 3) the sourcing transistors of the current mirrors (Mp and Mn) are turned on and off, switching the actual current source on and off. This virtually eliminates charge injection. In simulation an input current of 5nA and standard transistor inverters were used yet no switching noise was discernible at the output. Test-chip results verify these simulations.

5.1 Harmonic Distortion

The non-ideal effects of the input-offset voltage of a typical comparator generate offsets at the output of the *Palmo* circuit. Comparator propagation delays result in Harmonic Distortion as discussed in [6]. The Harmonic Distortion (HD_1) due to the propagation delay of the comparator is given by equation 2.

$$HD_1 = \frac{\sqrt{(\frac{1}{2})^2 + (\frac{1}{3})^2 + (\frac{1}{4})^2 + \dots}}{\frac{\pi \mu}{4\nu}}$$
(2)

Where μ is the magnitude of a signed PWM cosine input and ν is the size of the minimum pulse (ΔT_{min}) due to the non-ideal nature of the comparator. Differences in the two current sources which charge and discharge the integrating capacitor result in a second HD component, HD_2 , given by equation 3.

$$HD_2 = \frac{4}{\pi} \cdot \frac{\sqrt{\left(\frac{1}{3}\right)^2 + \left(\frac{1}{15}\right)^2 + \left(\frac{1}{35}\right)^2 + \dots}}{\frac{l_n + l_p}{2 \cdot |I_n - I_p|}}$$
(3)

Where I_p , I_n are the charging and discharging current sources respectively as shown in figure 3.

Figure 4 shows plots of HD_1 , HD_2 and Total Harmonic Distortion $(THD = HD_1 + HD_2)$ calculated using equations 2 and 3 and measured parameters from the test chip. At the time of writing, a single measurement of THD from this test chip has been made. A THD = 0.86% was measured at a sampling frequency of 8620Hz, as shown in Figure 4.

In our graph, distortion introduced by the comparator is dominant at high frequencies where the comparator delays become comparable to the sampling rate. Conversely, distortion due to current





Figure 6: Programmable Palmo Filter Response - results from the test chip with cut-off frequencies of 860Hz and 2kHz.

matching dominates at low frequencies due to the use of small and therefore less well matched integrating currents. On initial inspection, a THD of 2% for the Palmo filter compares favourably with a figure of 2.5% for early switched-current circuits [10] but not so favourably with a figure of 0.4% [2] quoted for a more mature S-C FPAA cell.

We anticipate considerable improvement in these THD figures for the next generation of Palmo circuits. These may be achieved by improving the current sources on chip by using cascode stages for example, and by investigating fast comparator circuits.

6 VLSI Results

A first test chip has helped us identify the limitations of the current circuit building blocks and allowed us to develop a new device to be fabricated in the near future. Here we present the results from that first test chip.

The graph of figure 5 shows the linearity of the *Palmo* filter for various K factors. In out test chip, the capacitors are fixed while the current sources are driven externally, therefore the output of the integrator is dependent upon I_{int} , since I_r is constant. The results displayed in figure 5 were taken by applying a number of constant pulses to the *Palmo* integrator and measuring the output. The output pulses (*out* in Figure 2) were sampled using a digital storage oscilloscope, the pulse width measurement giving the magnitude of the pulse, while the sign of the measurement was defined by the state of the sign clock.

6.1 Palmo filter response

In figure 6 the response of two first order Butterworth filters with cut-off frequencies of 860Hz and 2KHz are presented and compared to the ideal. Due to minor problems with the test chip there are mismatches between the ideal and actual filter characteristics. These results illustrate the programmability of the filter. We intend to construct an array of *Palmo* cells that may be programmed and reconfigured dynamically using an FPGA. This will allow us to implement higher order functions.

7 Conclusions and further work

This paper has introduced and analysed a novel technique for implementing programmable analogue hardware. Novel programmable *Palmo* circuits have been introduced which have low switching noise, low THD, and may be configured to perform many signal processing functions. An initial test chip has demonstrated the viability of the approach and aided in understanding the practical application of the technique. This has led to new ideas for the implementation of the *Palmo* building blocks to reduce THD and improve overall programmability and performance. Such circuits are ideal for analogue FPAA cells.

The new *Palmo* circuit blocks have a dedicated 6-bit current DAC and 3-bit programmable capacitor array giving 9-bits of programmability per cell. Cell interconnect to local neighbours or to pads further enhance programmability and high frequency performance. Dedicated static RAM is used to store the cell parameters and interconnect - which may also be dynamically reconfigured. A new improved comparator is under development to dramatically reduce harmonic distortion at high frequencies. These circuits are currently being designed and a new chip is to be fabricated in the near future.

References

- IMP Home-Page http://www.impweb.com/, Electrically Programmable Analog Circuit: Datasheet. San Jose California: IMP, April 1996.
- [2] A. Bratt and I. Macbeth, "Design and implementation of a field programmable analogue array," in FPGA '96, Monterey California: ACM, February 1996.
- [3] H. Kutuk and S. Kang, "A field-programmable analog array (FPAA) using switched-capacitor techniques," in *International Symposium on Circuits and Systems (ISCAS)*, vol. IV, (Atlanta), May 1996.
- [4] A. Hamilton, A. Murray, D. Baxter, S. Churcher, H. Reekie, and L. Tarassenko, "Integrated pulse-stream neural networks - results, issues and pointers," *IEEE Trans. Neural Networks*, vol. 3, pp. 385-393, May, 1992.
- [5] A. Hamilton, S. Churcher, P. J. Edwards, G. B. Jackson, A. Murray, and H. M. Reekie, "Pulse stream VLSI circuits and systems : The EPSILON neural network chipset," *International Journal* of Neural Systems, vol. 4, pp. 395-406, December 1993.
- [6] K. Papathanasiou and A. Hamilton, "Palmo signal processing: VLSI results from an integrated filter," in *The Third IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, vol. 2, (Rhodes-Ellas), pp. 832-835, October 1996.
- [7] T. Lehmann, "Implementation issues of self-learning pulsed integrated neureal systems," in 13th NORCHIP Conference, (Copenhagen, Denmark), pp. 145-152, 1995.
- [8] K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in *International Symposium on Circuits and Systems (ISCAS)*, vol. I, (Atlanta), pp. 270-273, May 1996.
- [9] M. Ismail and T. Fiez, Analog VLSI signal and information Processing. New York: Mc Graw Hill, 1992.
- [10] T. Fiez, G. Liang, and D. Allstot, "Switched-current circuit design issues," IEEE Journal of Solid-State Circuits, vol. 26, no. 3, pp. 192-202, 1990.

Publications

K. Papathanasiou and A. Hamilton, "Advances in programmable pulse based mixed-signal processing VLSI," in *IEEE-CAS Workshop on Analog and Mixed IC Design, Baveno*, (F. Maloberti, A. Baschirotto, and V. Liberali, eds.), (Baveno-Italy, 12-13 Sep), pp. 106-110, IEEE, 1997.



II. A NOVEL PULSE BASED SIGNAL PROCESSING TECHNIQUE:

Our novel signal processing approach encodes analogue information by modulating a digital pulse waveform. While the signals distributed between processing cells are digital pulses, the signal processing within each cell is performed using analogue circuit techniques.

The pulsed signal representation combines the virtues of both analogue and digital domains. Pulses are digital signals. They are noise tolerant and therefore robust. Pulses have a high drive capability over long distances and may be easily distributed within and between chips. Pulsed signals are inherently low power. Pulsed signals can be manipulated with digital circuits for signal routing or to perform mathematical functions using primitive logic gates. Pulsed signaling is viable in low voltage sub micron processes. Pulsed signaling results in compact analogue circuits which may be integrated in large numbers in a standard digital process. Many different pulse

K. Papathanasiou, (+)44-131-6505665, fax (+)44-131-6506554, kap@ee.ed.ac.uk, http://www.ee.ed.ac.uk/skap; A. Hamilton (+)44-131-6501000, fax (+)44-131-6506554, Alister.Hamilton@ee.ed.ac.uk, http://www.ee.ed.ac.uk/salister

plemented by simple arithmetic circuits. Some functions

require additional components external to the chip [3]. A classic parasitic-insensitive switched-capacitor integra-

tor stage has been used to implement a signal processing

cell which may be programmed using an arrangement of

switches and capacitors [4], [5]. Current-mode FPAA cir-

An FPMA combines digital FPGA and analogue FPAA

cells on a single chip with an analogue/digital interface

between the two arrays. The interface between the ana-

logue and digital sections of the FPMA requires specially

cuits have also been proposed [1].

This work is supported by the UK EPSRC grant reference GR/L36031. $^{1}Palmo$ is derived from the Hellenic word $\Pi A \Delta MO\Sigma$ which

¹*Palmo* is derived from the Hellenic word $\Pi A \Lambda M O \Sigma$ which means pulsebeat, pulse palpitation or series of pulses.

modulation techniques are possible including Pulse Frequency Modulation and Pulse Width Modulation (PWM) both of which have been used in neural network VLSI [6], [7].

FPMA circuits are easier to design, layout and test using pulsed analogue signal processing than those designed using conventional signal processing techniques. For example, PWM signals allow a simple pulsed analogue/digital interface in an FPMA architecture. PWMto-digital conversion may be performed very simply using an N-bit digital counter enabled by the pulse signal and clocked at a suitably high frequency. The resultant N-bit value in the counter represents the analogue information. Digital-to-PWM conversion may be performed by parallel loading an N-bit digital number into a counter and downcounting to zero. The length of the down-count period represents the output pulse width.

III. PULSED SIGNALING MECHANISMS

Most signal processing applications are performed by the use of an integrator, a differentiator and a scaler. Filtering, FFT, even adaptive filtering algorithms scale the output by a constant factor K, which might very in time. Integration and differentiator is performed by a capacitor in all analogue techniques while there are many different analogue scaler implementations available.

In our case it is possible to realize alternative scaler implementations depending upon the pulsed signaling mechanism which is used. Pulsed Frequency Modulated (PFM) signals can be multiplied by a single AND gate, provided that the input signals are statistically uncorrelated. Nevertheless the maximum frequency of operation of a PFM circuit is very slow for most applications. Pulse Width Modulated (PWM) signals on the other hand are much faster and suitable for real-time implementations. In our case an Sign-Magnitude PWM signal is used were the Sign is defined by a global clock and the magnitude by the width of the pulse.

IV. PROGRAMMABLE PALMO VLSI DEVICES

Programmable digital logic may be used to manipulate the pulse width modulated signals arriving at the basic Palmo cell before they reach the ξ_+ and ξ_- switches of the integrator (Figure 1B) [8]. Signal routing between basic Palmo cells may also be controlled by programmable digital logic. In the two devices reported here, an array of basic Palmo cells has been implemented on chip and all the programmable digital logic has been implemented on a separate FPGA. The scaling factor, K, may be individually set for each basic Palmo cell and is defined as a ratio of capacitors multiplied by a ratio of currents. The use of a product of capacitor and current ratios makes K insensitive to process variations.



Fig. 2. Implementation of a Palmo FPAA cell.

V. THEORETICAL CONSIDERATIONS

Appropriate selection of digital logic to drive the ξ_+ and the ξ_- switches in Figure 1B allows the implementation of a Miller integrator which is directly equivalent to the switched-capacitor implementation. The transfer characteristic of the Palmo implementation is given by :-

$$\Delta T_{out} = K \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}}$$
(1)

where ΔT_{plus} and ΔT_{minus} are the input pulse widths at the differential inputs to the integrator and ΔT_{out} is the integrator output pulse width [9].

Harmonic distortion in the basic Palmo cell comes from the non-ideal nature of the comparator and of the integrator [10]. It is the propagation delay of the comparator which introduces harmonic distortion to the signal. Nonequal comparator rise and fall times make the width of the pulse at the output of the comparator longer or shorter than it should ideally be. This therefore represents a distortion to the magnitude of the signal.

This Harmonic Distortion (HD_1) component is given by equation 2.

$$HD_{1} = \frac{\sqrt{(\frac{1}{2})^{2} + (\frac{1}{3})^{2} + (\frac{1}{4})^{2} + \dots}}{\frac{\pi\mu}{4\nu}}$$
(2)

Where μ is the magnitude of a signed PWM cosine input and ν is the size of the minimum pulse (ΔT_{min}) due to the non-ideal nature of the comparator.

Differences in the current sources which charge and discharge the integrating capacitor result in a second HD component, HD_2 , given by equation 3.

$$HD_2 = \frac{4}{\pi} \cdot \frac{\sqrt{(\frac{1}{3})^2 + (\frac{1}{15})^2 + (\frac{1}{35})^2 + \dots}}{\frac{I_n + I_p}{2 \cdot |I_n - I_p|}}$$
(3)

Where I_p , I_n are the charging and discharging current sources respectively as shown in Figure 1B.



Fig. 3. Clamped comparator with positive feedback.

Distortion introduced by the comparator is dominant at high frequencies where the comparator delays become comparable to the sampling rate. Conversely, distortion due to current matching dominates at low frequencies due to the use of small and therefore less well matched integrating currents.

VI. PALMO FPAA IMPLEMENTATION

After demonstrating the practicality of using pulsed signals to implement analogue circuits, with first test device [8]–[10], a second programmable chip with regard to programmability and applicability was designed.

This device has an analogue to PWM converter, a PWM to analogue converter and 8 basic Palmo cells. Each of these Palmo cells has a programmable capacitor ratio (3 bit) and two on-chip current DAC's (6 bit) to control the current ratio. Each Palmo cell therefore has an electronically programmable 9 bit scaling factor and electronically programmable interconnect via the separate FPGA. Finally there is programmable analogue interconnect between the Palmo cells and two analogue I/O pins.

A typical Palmo cell of the new device is shown in figure 2. The current from the current DAC is driven through the Ramp/Int CMOS switch to either the integrating or the ramp capacitor [8]. The capacitors can be selected from a fully interconnected array of 9 elementary capacitors per cell. The input pulse controllers the Pulse switch. This is not implemented as a standard CMOS switch. The Pulse digital input controls the DAC itself by switching on and off the current source itself [8], [11]. If the configuration CMOS switchs switch during the reset period of the ramp, this approach limits the clock feedthrough noise to virtually neglectable levels.

A. Harmonic Distortion

Extra care was given in order to minimize the THD of this device. The desired linearity defines the minimum

acceptable distortion (delay) of the circuit and therefore the maximum frequency of operation.

A set of CMOS switches configures the function of an individual basic cell. The Up/Down switch drives the output current of the current DAC through an accurate PMOS current mirror. Centroid lay-out and big size matched transistors were used for this PMOS current mirror [12]–[15]. In that way the HD_2 component, due to current inaccuracies, of the THD is minimized.

Extra care was taken in order to maximize comparator performance by minimizing the comparator delays. These delays cause significant linearity inaccuracies which limit the overall performance of the circuit. A clamped comparator was used (figure 3). Clamped comparators are mostly suitable for our Palmo circuits because of the continuity of their response. Faster frequencies of operation could be achieved by the use of a reset switch within the comparator. Nevertheless such a switch will effectively quantify the output. A clamped comparator switches faster than a standard differential one. Since the circuit gradually changes state, the effects of parasitic capacitances are limited. In order to maximize the frequency of operation positive feedback is used in our comparators. The transistors M10 and M11 add or subtract some current to M7 (figure 3) depending to the slope of the ramp. Therefore the comparator changes state faster than it would have without this positive feedback. The bias voltages V_{fixp} and V_{fixn} which define the size of these currents are controlled externally.

By the use of the above techniques the THD of the new circuit will be limited, therefore it would operate significantly faster than the first device reaching the sampling frequency of 1MHz. Further improvement can be achieved by the use of a smaller geometry process than the 2.4micron used for this chip. Finally the symmetry of the comparator, in addition to the use of the same DAC for the integration and the ramp, limits the DC offsets.

VII. VLSI RESULTS

At the time witting the new device has being fabricated. Extensive checks are performed in order to verify its operation. The clamped comparator and the internal addressing circuit which controls the configuration SRAM cells have being characterized. On the other hand the operation of the analogue circuits have not yet being being fully checked. Furthermore a development board integrating an FPGA and two of our *Palmo* chips is under construction. This board will enable us to demonstrate the use of our chips in real-time applications. The results presented here are from a similar board testing our previous chip.

A. Analogue IIR filters

Our first Palmo device has an analogue to PWM converter and 3 basic Palmo cells. Each Palmo cell has a



Fig. 4. First Palmo Chip Photograph

fixed capacitor ratio, and a current ratio that may be set by off-chip potentiometers.

A photograph of the first Palmo device is shown in Figure 4. This device has been used to implement the analogue functions of a first, second and third order Butterworth filter (figure A.). The signal interconnection between basic Palmo cells and other digital functions are performed by a digital FPGA. The results from the VLSI device for cut-off frequencies of 1kHz and 2kHz are compared with the theoretical ideal. The attenuation in the stop band is 40-50dB in these examples. We expect improvements in our second chip due to a much improved comparator design, improved current sourcing, current matching, and programmable capacitor ratios.

B. Mixed-signal FIR filters

Figure 6 shows the results form a 24 tap FIR filter. This filter was implemented by the use of a mixed-signal circuit. Part of the digital FPGA circuit is used for configuring the interconnection On the other hand some of the functionality of the FIR algorithm is performed by other FPGA configurable blocks. It is clear that this FIR filter shares the advantages of an analogue stand-alone implementation and the accuracy of the digital DSP based solution. This mixed-signal implementation can be used in more complex digital algorithms and can generate an alternative to digital signal processing.

VIII. CONCLUSIONS-FUTURE WORK

A new technique for programmable analogue signal processing has been presented that uses a pulse based signal representation. An advanced implementation of an Palmo FPAA was demonstrated clarifying the critical points of







Fig. 6. Mixed-signal 24 tap FIR filter results.

the circuit. That circuit uses digital I/O and is ideally suited to mixed-signal applications.

Results from VLSI demonstrate first, second and third order IIR filter operating at different frequencies. These results have been obtained by connecting basic Palmo cells and selecting clock frequencies via an FPGA while current ratios have been set using off-chip potentiometers. Furthermore results from a 24th order mixed-signal FIR filter, demonstrate the possibility of implementing digital algorithms with pulsed based devices.

A new chip with a 9-bit fully programmable scaling factor and programmable interconnect is currently being tested. Two of these chips will be integrated on a board using an FPGA to manipulate the digital I/O signals. This board will demonstrate practical examples of FPAA applications. Furthermore the software which is needed for programming the FPAA chips can be incorporated into the already available the FPGA software, by a set of macros. It is therefore clear that pulsed based analogue

circuits can be easily integrated into programmable analogue arrays, opening a new application area for analogue VLSI.

References

- P. G. Gulak, "Field-programmable analog arrays: a status report," in *The fourth Canadian workshop on Field-Programmable Devices*, Canada, May 1996, pp. 138-141.
 D. L. Grundy, "A computational approach to VLSI analog design," *Journal of VLSI Signal Processing*, vol. 8, no. 1, pp. 53-60, 1994.
 G. D. Harrow, D. Lin, Charles, J. M. Status, J. Sta
- 53-60, 1994. ZETEX Home-Page http://www.zetex.com/, Totally Re-configurable Analog Circuit (TRAC): Datasheet, ZETEX, Chadderton-Oldham, November 1996. IMP Home-Page http://www.impweb.com/, Electrically Pro-grammable Analog Circuit: Datasheet, IMP, San Jose Califor-ica Acadi 1006. [3]
- [4] nia, April 1996.
- [5]
- nia, April 1996. Adrian Bratt and Ian Macbeth, "Design and implementation of a field programmable analogue array," in FPGA '96. ACM, Monterey California, February 1996. A. Hamilton, A.F. Murray, D.J. Baxter, S. Churcher, H.M. Reekie, and L. Tarassenko, "Integrated pulse-stream neural networks results, issues and pointers," *IEEE Trans. Neural Networks*, vol. 3, no. 3, pp. 385-393, May, 1992. A. Hamilton, S. Churcher, P. J. Edwards, G. B. Jackson, A.F. Murray, and H. M. Roekie, "Pulse stream VLSI circuits and systems : The EPSILON neural network chipset," *Interna-tional Journal of Neural Systems*, vol. 4, no. 4, pp. 395-406, [6]
- [7] tional Journal of Neural Systems, vol. 4, no. 4, pp. 395-406, December 1993.

- K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in International Symposium on Circuits and Systems (ISCAS), Atlanta, May 1996, vol. 1, pp. 270-273.
 K. Papathanasiou and A. Hamilton, "Palmo signal processing: VLSI results from an integrated filter," in The Third IEEE International Conference on Electronics, Circuits and Systems (ICECS), Rhodes-Ellas, October 1996, vol. 2, pp. 832-835.
 K. Papathanasiou and A. Hamilton, "Newle palmo analogue signal processing ic design techniques," in Colloquium on Analogue Signal Processing, Oxford, UK, Novenber 1996, pp. 5/1-5/6. [10]
- logue Signai Freecosing, Santa Sa
- [13] J. B. Shyu, G. C. Temes, and F. Krummenacher, "Random error effects in matched MOS capacitors and current sources," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 6, pp. 948– 027 (2017) 100 (2017) 100 (2017) 100 (2017) 955, 1984.
- [14]
- 955, 1984. C. Michael and M. Ismail, "Statistical modeling of device mismatch for analog MOS integrated circuits," *IEEE Journal* of Solid-State Circuits, vol. 27, no. 2, pp. 154-66, 1992. M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of* Solid-State Circuits, vol. 24, no. 5, pp. 1433-9, 1989. [15]

Publications

T. Brandtner, K. Papathanasiou, and A. Hamilton, "A palmo cell using sampled data log-domain integrators.," submited to the IEE electronics letters, 1998.

TO APPEAR IN IEE ELECTRONICS LETTERS, p. 1

A PALMO¹ CELL USING SAMPLED DATA LOG-DOMAIN INTEGRATORS.

T. Brandtner, K. Papathanasiou, A. Hamilton

Indexing terms: Analogue Signal Processing, FPAA, FPMA, Palmo, Pulses, Log-domain, current mode, VLSI

> This paper presents the first log do-main integrator for programmable analogue sampled data signal processing. This cir-cuit is specific to the implementation of emergent programmable pulse based sig-nal processing systems yielding greater dy-namic range, reduced power supply voltage and increased operating frequency. Simu-lation results demonstrate the validity of the approach. the approach.

Introduction :Programmable pulse based signal pro-cessing systems [1] use digital pulses to represent all signals. The analogue information is encoded in time by modulating the width of a digital pulse(s) rather than the magnitude of a current or a voltage. Pulsed signals are robust, inherently low-power, easily regen-erated, and easily distributed across and between chips. The Palmo cells used to perform analogue operations on the pulsed signals are compact, fast, simple and programmable. programmable.

Log-domain integrators [2] integrate logarithmic-ally compressed input currents by making use of the

ally compressed input currents by making use of the transistor characteristics of bipolar (or subtreshold MOS) transistors. The integrated voltage is subsequently expanded exponentially to generate the output cur-rent. This companding technique offers a large output dynamic range for a small voltage swing. The use of bipolar transistors results in higher operating frequen-cies and improved accuracy over CMOS integrators. The Palmo signal representation results in an effi-cient implementation of a log-domain sampled data integrator. Since the input to the integrator is di-gital it has only two values which may be represented by two non-zero currents generated globally for the whole chip. This is significantly simpler than gener-ating multi-valued inputs for each sampled data cell. The advantages of using log-domain Palmo cells are a high degree of programmability, a large dynamic range of 80dB and sampling rates in excess of 5MHz com-pared to 54dB and 1MHz of a commercially available device [3]. device [3].



Figure 1: Palmo cell and typical waveform dia-gram

The Palmo Cell: Pulse Width Modulation is used to represent the analogue input and output signals [1].

THE NAME PALMO IS DERIVED FROM THE HELLENIC WORD IIAAMOE WHICH STANDS FOR PULSEBEAT, PULSE PALPITATION OR SERIES OF PULSES.

The magnitude of the signal is represented by the duration of the pulses, while the sign is determined by whether the pulses occurred in the positive or negative cycle of the sign signal (sign-magnitude coding). Analogue signals are converted into pulses by comparing them to a ramp (Figure 1).

them to a ramp (rigure 1). A block diagram of the *Palmo* cell is shown in Fig-ure 1. It consists of three parts: digital logic, an in-tegrator and a current comparator. The digital logic block converts the input pulses and the sign signal into two differential input currents that form the input to the integrator. The integrator is fully differential and works in the log-domain. The pulsed output is gener-ated by a current comparator (CCC) which ated by a current controlled comparator (CCC) which ated by a current controlled comparator (CCC) which compares the integrator output current to a current ramp produced by an identical integrator. The ramp can be generated globally. For high speed operation a dedicated ramp generator may be used for each Palmo cell. The symmetry of the dual slope ramp eliminates inaccuracies due to comparator delays provided that the rising and falling times of the comparator are well. the rising and falling times of the comparator are well matched. This results in each sample being represen-

ted by two pulses (Figure 1). The gain of the Palmo cell is controlled by the ra-tio of the integrating constants of the two integrators generating I_{ramp} and I_{int} .



Figure 2: Log-domain integrator

The Log Domain Integrator : The circuit of the log-domain integrator used in our *Palmo* cell is shown in Figure 2. It is based on [2], but has improved linearity over a bigger current range due to the use of cascode current mirrors and the stabilising transistors, M13 and M14. The integrator is fully differential, hence the input value is the difference of the two currents I_P birst the input currents are compared into and $I_{n.}$ First the input currents are compressed into log-domain by Q1 (Q8); Q2 and C1 (Q7, C2) perform integration in the log-domain. The integrated value is scaled by Q3 (Q6), the scaling factor depends on the current $I_{t.}$ Finally, Q4 (Q5) expands the compressed signal. The output is represented by the difference of the current $I_{t.}$ and $I_{t.}$

the currents I_{O1} and I_{O2} . From the translinear equations for the integrator circuit of Figure 2 the following formula may be derived:

$$\frac{d}{dt} (I_{O1} - I_{O2}) = \frac{I_t}{C \cdot V_{th}} \left(1 + \frac{1}{\beta}\right) (I_p - I_n) + \frac{1}{C \cdot V_{th}} \left(\frac{I_t}{\beta} - I_c\right) (I_{O1} - I_{O2}) - \frac{1}{C \cdot V_{th}} (a - ab) I_{O1} I_{O2} - \frac{I_t}{C \cdot V_{th}} \left(1 + \frac{1}{\beta}\right) \left[I_p \left(\frac{k_4}{k_1 k_3} - 1\right) \left(\frac{k_5}{k_6 k_8} - 1\right)\right]$$
(1)

where a and b are the mirroring ratios of the current mirrors M5-8 and M9-12 respectively, β is the current gain of the bipolar transistor, $V_{th} = 26mV$ at $300^{\circ}K$, $k_i = 1 + V_{BC_i}/V_{AF}$, V_{BC_i} is the collector-base voltage of transistor Q_i and V_{AF} is the early voltage. The first term on the right side of the equation shows that the difference of the output average f.

shows that the difference of the output currents $(I_{O1}$.

TO APPEAR IN IEE ELECTRONICS LETTERS, p. 2

 I_{O2}) is proportional to the integral of $I_p - I_n$. The integration constant can be controlled by changing the current I_1 or the capacitor C. The overall gain of the Palmo cell is [1]

$$K = \frac{I_{t_{int}}}{I_{t_{ramp}}} \cdot \frac{C_{ramp}}{C_{int}} \tag{2}$$

and therefore temperature dependency has been re-

and therefore temperature dependency has been re-moved from the gain of the cell. The second term occurs because of the base cur-rents of Q3 and Q5 which introduces a leakage current on the integrating nodes. This effect is cancelled by introducing a current $I_e = I_t/\beta$, which is derived from the base current of a single bipolar transistor with a collector current of I_t . The reason for the third term is the difference of the mirroring ratios of the two current mirrors M5-8 and M9-12. To maximise the dynamic range of the circuit at the expense of operating frequency, operating voltage and power, a cascode current mirror is used.

voltage and power, a cascode current mirror is used. The last term shows the influence of the early ef-fect. The collector voltages of Q4 and Q5 vary due to output current changes. The transistors M13 and M14 are used to stabilise the collector voltages of Q4 and O5 to environment the term.

Q5 to minimise this term. The integrator may be reset by either connecting integrating nodes 1 and 2 together temporarily, or by increasing the base compensating current.

Current Controlled Comparator (CCC): This circuit is mainly responsible for the high sampling fre-quency capability of our *Palmo* cell because it is faster than the voltage mode comparators used in former Palmo implementations.

Paimo implementations. The CCC used was based upon [4] with a slight modification to accommodate the lack of p-wells in the fabrication process used. This circuit allows the detec-tion of small changes in input currents (-50nA to 50nA) with a short propagation delay (17ns). Normal cur-rent comparators require several hundred nanoseconds to detect the same current change.



Figure 3: Linearity of the Palmo cell, for varying Īt

Simulation Results: The linearity of the integrator is shown in Figure 3. It demonstrates good linear-ity with an output range of $\pm 15\mu A$. In the same fig-ure it is shown that the integrator gain may be set by controlling I_i . It is possible to change I_i between $50\pi A$ and $20\mu A$. In addition the integrator gain can be altered by changing the capacitor C. When integrating a sine wave the maximum total harmonic distortion (THD) at the output is less than 0.94%. For sampling frequencies in the kilohertz range the THD is less than

(11D) at the output is less than 0.94%. For sampling frequencies in the kilohertz range the THD is less than 0.5%. A first order sampled data low pass filter was im-plemented by adding the feedback signal shown in Fig-ure 1. The frequency response of this simple filter can be seen in Figure 4. Here the cutoff frequency is set by controlling the current I_t . As in all sampled data



Figure 4: Frequency response of a simple Palmo filter, $f_s = 1 M H z$

systems the sampling frequency may also be used to modify the response of the filter.

modify the response of the filter. **Conclusions:** The log-domain sampled data integ-rator presented here is ideal for use in emergent pulse based signal processing systems. Pulse based signal processing is a technique that is ideally suited to the implementation of programmable mixed-signal electron-ics, especially Field Programmable Analogue and Mixed-Signal Arrays. The advantages of the log domain cir-cuit to pulse based signal processing are lower power supply voltages and greater dynamic range. The use of a current mode technique gives higher sampling fre-quencies due to the speed of the current comparator used in the Palmo cell. The circuits reported here are optimised for dy-namic range and designed to operate at 5V and sampling frequencies up to 5MHz. Minor modifications to the current mirrors in the integrator can reduce the oper-ating voltage as low as 1V [2] or increase the sampling frequency by a factor of 4 to 20MHz. **Acknowledgements:**This work is supported by the

Acknowledgements: This work is supported by the UK EPSRC grant reference GR/L36031.

References

- K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a palmo filter," in *International Symposium on Circuits and* Systems (ISCAS), vol. 1, (Atlanta), pp. 270–273, May 1996.
- Punzenberger and C. Enz, "A new 1.2v [2] M. M. Punzenberger and C. Enz, "A new 1.2v BiCMOS log-domain integrator for companding current-mode filters," in 1996 IEEE International Symposium on Circuits and Systems - Circuits (ISCAS) 1996, (Atlanta, Ga, 12 - 15 Ma), pp. 125-128, IEEE Service Center, 1996.
- [3] A. Bratt and I. Macbeth, "Design and implement-ation of a field programmable analogue array," in *FPGA '96*, Monterey California: ACM, February 1000 1996.
- A. Tang and C. Toumazou, "High performance CMOS current comparator," *Electronics Letters*, vol. 30, no. 1, pp. 5-6, 1994. A