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

Konstandinos Papathanasiou

Κωνσταντίνος Παπαθανασίου

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.
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 award-winning 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.

\(^1\)The name Palmo is derived from the Hellenic word ΠΑΛΜΟΣ which means pulse-beat, pulse palpitation or series of pulses.
# Table of Contents

List of Abbreviations and Acronyms vii

List of Symbols ix

List of Figures x

List of Tables xiv

Writing style xiv

1. Introduction 1
   1.1 The History 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 Summary 5

I Background 6

2. Signal Processing and Field Programmable Analogue Arrays 7
   2.1 Signal Processing Fundamentals 7
   2.2 Filtering 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
         Wideband 16
# Table of Contents

Narrowband ................................................. 16

2.2.7 Implementing Filters With Integrators ............ 16
2.2.8 Sampled Data filters ............................... 17

2.3 Sampled Data Signal Processing ...................... 18

2.4 Digital Signal Representation ........................ 19

2.5 Analogue 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 .... 23
2.5.4 Other ADC Implementations .......................... 24
    Time-interleaving ADC ............................... 24
    Pipeline ADC ....................................... 24

2.6 Elementary Signal Processing Functions .......... 24
    Scaling ............................................. 24
    Integration ....................................... 25
    Multiplication of two Signals ....................... 25
    Comparison ........................................ 25
    Discussion ........................................ 25

2.7 Integrator 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 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

3.1 The use of pulses in analogue systems ............... 41
    3.1.1 Advantages of pulsed systems for the implementation of FP-MAs .... 44
3.2 Typical Palmo Cell .................................. 45
3.3 Signal Representation ............................... 46
3.4 Palmo Voltage Domain Implementation ................. 50
    3.4.1 *Palmo* Miller Integrator ....................... 51
II Implementations

4. PALMO-I test-chip

4.1 Introduction ........................................ 69
4.2 PALMO-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 Circuits used in the PALMO-I Chip .............. 72
  4.3.1 Charging and discharging circuits .......... 72
4.4 Comparator .......................................... 74
  4.4.1 Capacitor array ................................ 74
  4.4.2 Input Current Mirrors ......................... 75
4.5 Testing The Chip .................................... 75
  4.5.1 Initial Testing ................................ 75
  4.5.2 Signed PWM Conversion ....................... 76
4.6 Further Testing ..................................... 80
  4.6.1 Integrator ...................................... 80
  Integrator Linearity ................................ 81
  4.6.2 Low-pass Filter Implementations ............ 83
4.7 Mixed-Signal Systems ............................... 83
  4.7.1 The technique .................................. 84
  4.7.2 Mixed-signal FIR implementation ........... 85
  4.7.3 PALMO-I Conclusions .......................... 86
### Table of Contents

5. Palmo FPAA and prototyping board  
5.1 Introduction ................................. 88  
5.2 Chip 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 prototyping 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 Testing 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 Conclusions .................................. 108  
6. Advanced implementations: Log-domain BiCMOS *Palmo* cells 109  
  6.1 Bipolar Background .......................... 110  
    The translinear principal ................... 111  
  6.2 The *Log-domain Palmo Cell* ................ 111  
    Discussion ................................ 112
# Table of Contents

6.3 Log-domain cell ........................................ 113
  6.3.1 The Log-domain Integrator ......................... 113
  6.3.2 Current Controlled Comparator .................... 116
6.4 Voltage to Current Converter .......................... 118
6.5 Simulated Results ...................................... 120
  6.5.1 Integrator Linearity ............................... 120
  6.5.2 A first-order filter implementation ............... 120
6.6 Log-domain Multiplier ................................ 122
  Log-domain multiplier discussion ...................... 123
6.7 Conclusions ............................................. 123

7. Developments and Conclusions ......................... 125
  7.1 Introduction .......................................... 125
  7.2 Our Approach .......................................... 125
  7.3 Current Developments ................................ 127
    7.3.1 Using Gate Arrays for Application Specific DSP .... 127
    7.3.2 Texas Instruments TMS-320C6x .................... 128
    7.3.3 Motorola announces the first commercial FPMA ........ 128
    7.3.4 Institute for System-Level-Integration ............ 129
  7.4 Future Developments .................................. 130
    7.4.1 Future Work ....................................... 131
      Voltage domain CMOS circuits ......................... 131
      Log-Domain BiCMOS circuits ......................... 132
      Applications .......................................... 133
    7.4.2 Overall Conclusions ............................... 134

Bibliography Categories .................................... 136

Bibliography ............................................... 137

A. Palmo FPAA Addressing Registers .................... 147

B. Palmo FPAA Pin Out .................................... 149

C. Microcontroller Code .................................. 151
  C.1 Commands ............................................. 151
  C.2 Microcontroller Code ............................... 151

D. Board Documentation .................................. 157
  D.1 Initialisation ....................................... 157
  D.2 Board Schematic Diagrams ........................... 157
E. FPGA Filter Schematics

F. Publications

ISCAS 1996 .................................................. 163
ICECS 1996 .................................................. 167
Colloquium on Analogue Signal Processing 1996 .................. 171
IEEE-CAS 1997 ............................................. 177
Submitted to IEE Electronics Letters ............................. 182
## List of Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Alternating Currents.</td>
</tr>
<tr>
<td>ADC</td>
<td>Analogue to Digital Conversion.</td>
</tr>
<tr>
<td>AIC</td>
<td>Analogue Interfacing Chip.</td>
</tr>
<tr>
<td>ALE</td>
<td>Address Latch Enable.</td>
</tr>
<tr>
<td>AMD</td>
<td>Advanced Micro Devices (Corporate name).</td>
</tr>
<tr>
<td>AN</td>
<td>Manufacturer's Application Note.</td>
</tr>
<tr>
<td>AND</td>
<td>Boolean AND operation (+).</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Networks.</td>
</tr>
<tr>
<td>BAUD</td>
<td>From the name of Emile Baudot, inventor of the Telex code: now a measure of communications capacity.</td>
</tr>
<tr>
<td>BCD</td>
<td>Binary Coded Decimal number.</td>
</tr>
<tr>
<td>BUS</td>
<td>Basic Utility System: used for digital system interconnect.</td>
</tr>
<tr>
<td>BiCMOS</td>
<td>Bipolar Complementary Metal Oxide Semiconductor.</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design (tool).</td>
</tr>
<tr>
<td>CCC</td>
<td>Current Controlled Comparator.</td>
</tr>
<tr>
<td>CM</td>
<td>Current Mode.</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor.</td>
</tr>
<tr>
<td>CMRR</td>
<td>Common Mode Rejection Ratio.</td>
</tr>
<tr>
<td>CNN</td>
<td>A programmable Cellular Neural Network universal machine.</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to Analogue Converter.</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current.</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing (or Digital Signal Processor if referring to a semiconductor device).</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable-Programmable Read-Only Memory.</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response.</td>
</tr>
<tr>
<td>FPAA</td>
<td>Field Programmable Analogue Array.</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array.</td>
</tr>
<tr>
<td>FPMA</td>
<td>Field Programmable Mixed Signal Array.</td>
</tr>
<tr>
<td>FSF</td>
<td>Filter Scaling Factor.</td>
</tr>
<tr>
<td>HD</td>
<td>Harmonic Distortion.</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines (Corporate name).</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit.</td>
</tr>
<tr>
<td>IEE</td>
<td>Institute of Electrical Engineers (UK).</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers (US).</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response.</td>
</tr>
<tr>
<td>KCL</td>
<td>Kirchoff’s Current Law.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>LC</td>
<td>Inductor-Capacitor circuit, as in filters.</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode.</td>
</tr>
<tr>
<td>MAX</td>
<td>Maxim Semiconductors (Corporate name).</td>
</tr>
<tr>
<td>MC</td>
<td>Micro-Controller.</td>
</tr>
<tr>
<td>Hz</td>
<td>Hertz: Unit of frequency.</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology.</td>
</tr>
<tr>
<td>MOS</td>
<td>Metal Oxide Semiconductor.</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit.</td>
</tr>
<tr>
<td>NAND</td>
<td>Boolean function ( y = a + b ).</td>
</tr>
<tr>
<td>NMOS</td>
<td>N (Negative) channel Metal Oxide Semiconductor.</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Networks.</td>
</tr>
<tr>
<td>NPN</td>
<td>Negative-Positive-Negative (transistor).</td>
</tr>
<tr>
<td>OPAMP</td>
<td>Operational AMPlifier.</td>
</tr>
<tr>
<td>OR</td>
<td>Boolean function ( . ).</td>
</tr>
<tr>
<td>OTA</td>
<td>Operational Transconductance Amplifier.</td>
</tr>
<tr>
<td>OTA-C</td>
<td>Operational Transconductance Amplifier - Capacitor, used for filter design.</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer.</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board.</td>
</tr>
<tr>
<td>PFM</td>
<td>Pulse Frequency Modulation.</td>
</tr>
<tr>
<td>PGA</td>
<td>Pin Grid Array (semiconductor package).</td>
</tr>
<tr>
<td>PMOS</td>
<td>P (Positive) Channel Metal Oxide Semiconductor.</td>
</tr>
<tr>
<td>PNP</td>
<td>Positive-Negative-Positive (transistor).</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation.</td>
</tr>
<tr>
<td>RC</td>
<td>Radio Communication.</td>
</tr>
<tr>
<td>RLC</td>
<td>Resistor Inductor Capacitor.</td>
</tr>
<tr>
<td>RS-232</td>
<td>Recommended Standard 232: for serial communications (by the Electronic Industries Association).</td>
</tr>
<tr>
<td>SC</td>
<td>Switched Capacitor.</td>
</tr>
<tr>
<td>SI</td>
<td>Switched Current.</td>
</tr>
<tr>
<td>SLI</td>
<td>System Level Integration (systems on a chip).</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Read Only Memory.</td>
</tr>
<tr>
<td>TH</td>
<td>Track &amp; Hold switched current circuit.</td>
</tr>
<tr>
<td>THD</td>
<td>Total Harmonic Distortion.</td>
</tr>
<tr>
<td>TMS</td>
<td>Texas Micro Systems (Corporate name).</td>
</tr>
<tr>
<td>UPS</td>
<td>Uninterpretable Power Supply/System.</td>
</tr>
<tr>
<td>VCCS</td>
<td>Voltage Controlled Current Sources.</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very Large System Integration.</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive OR Boolean function.</td>
</tr>
<tr>
<td>s-PWM</td>
<td>Signed-Pulse Width Modulated signal.</td>
</tr>
</tbody>
</table>
List of Symbols

- **C**: Capacitor (F).
- **$C_{ox}$**: MOS gate-oxide capacitance per unit area.
- **$f_c$**: Is the cut-off frequency.
- **$f_n$**: Is the normalised at $1\text{rad/sec}$ cut-off frequency.
- **$I_C$**: Is the collector current if a bipolar transistor.
- **$I_S$**: Is the saturation current of a bipolar transistor.
- **L**: MOS transistor channel length (m) or Inductor (H).

**Minus** Palmo integrator inverting input.

- **$n$**: $n = 1, 2, 3, ...$
- **Plus** Palmo integrator non inverting input.
- **R**: Resistance ($\Omega$).
- **t**: Time.
- **$V_{AF}$**: The early voltage of a bipolar transistor.
- **$V_{BE}$**: The Base Emitter voltage.
- **$V_{th}$**: Thermal voltage: $V_t = kT/q \approx 26mV$ 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 $i1$.
- **Z**: Complex resistance ($\Omega$).
- **$z^{-1}$**: Digital Delay.
- **$\beta$**: For bipolar transistors the ??gain??
- **$\Delta T$**: Time difference, often expressing the width of a pulse.
- **$\Phi$**: The hellenic letter "Phi".
- **$\Phi$ & $\Phi$**: Sampling clock and complement.
- **$\mu$**: Carrier surface mobility.
- **$\sigma$**: Statistical variance.
- **$\xi$**: The hellenic letter "xi".

**Palmo charging current control inputs** (those derive from the Plus & Minus cell inputs by the use of digital logic blocks).

<table>
<thead>
<tr>
<th>Signal Definition</th>
<th>Quantity</th>
<th>Subscript</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total instantaneous value</td>
<td>Lowercase</td>
<td>Uppercase</td>
<td>$q_A$</td>
</tr>
<tr>
<td>DC value</td>
<td>Uppercase</td>
<td>Uppercase</td>
<td>$Q_A$</td>
</tr>
<tr>
<td>AC value</td>
<td>Lowercase</td>
<td>Lowercase</td>
<td>$q_a$</td>
</tr>
</tbody>
</table>
## 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 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
List of Figures

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 inverting 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 Total Harmonic Distortion ........................................ 59
3–13 A) Clamped Comparator, B) Clamped Comparator with positive feedback to increase the gain. ............................... 60
3–14 Clamped Comparator small signal analysis. .................. 61
3–15 Output stage gain calculation. .................................... 63
3–16 Filter implementation using differential integrators (a) of a RLC low-pass filter (b). .................................................. 65
3–17 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 minimum-pulse cancellation. .................................................. 78
4–7 Minimum Pulse Generator. ........................................ 79
List of Figures

4-8 Integrating a signed input: the pulsed based approach. .......... 80
4-9 Integrator programmability and linearity. .......................... 82
4-10 VLSI results from First Palmo Chip: 1st, 2nd and 3rd order filters
   at cut-off frequencies of 1kHz and 2kHz. ............................ 82
4-11 Mixed-signal technique............................................... 84
4-12 Palmo mixed-signal 24 tap FIR filter implementation results. .. 85

5-1 Block diagram of the PALMO-FPAA chip. .......................... 89
5-2 Typical cell circuit diagram. ........................................ 92
5-3 Palmo FPAA comparator schematic. ................................ 93
5-4 Address Decoder .......................................................... 95
5-5 SRAM implementations: A) Standard quasi-static cell, B) Dynamic
   SRAM cell. .................................................................... 96
5-6 Input voltage level shifter. ............................................... 97
5-7 Prototyping board block (and floor) diagram. ........................ 98
5-8 Prototyping board and host laptop. ..................................... 101
5-9 Palmo Chip Photograph. ................................................... 102
5-10 Schematic diagram of the FPGA cell used to configure the analogue
   FPAA chips. .................................................................... 103
5-11 Addressing in the FPGA (address = 80h). ........................... 104
5-12 Loading “AAh” to the FPAA. ............................................ 104
5-13 PWM output by the use of an externally generated ramp. ........ 104
5-14 Palmo Sigma-Delta modulator. ......................................... 106
5-15 Sigma-Delta modulator results. ........................................ 107

6-1 Temperature dependency of Is for Vb=700mV of the BiCMOS pro-
   cess used. ........................................................................ 111
6-2 Palmo cell and typical waveform diagram. ............................ 112
6-3 Log-domain integrator. ..................................................... 114
6-4 Current comparator .......................................................... 117
List of Figures

6-5 Voltage-Current-Converter ........................................... 118
6-6 Linearity of the Palmo cell. ........................................... 121
6-7 Frequency response of a simple Palmo filter, fs=1MHz. .......... 121
6-8 Log-domain multiplier .................................................. 122

7-1 Typical Palmo FPMA implementation. ............................... 126
7-3 Dedicated minimum pulse cancellation. ............................... 131

A-1 FPAA address register ............................................... 147
A-2 FPAA interconnect registers .......................................... 148
A-3 Typical cell Capacitor and DAC registers ......................... 148

E-1 Second order Palmo filter implementation—FPGA schematic. It includes digital logic to drive the Palmo inputs, signed PWM and Ramp generation ................................................................. 160
E-2 Tap input configuration ................................................. 161
E-3 signed PWM output generation ....................................... 161
E-4 Ramp generating cell ................................................... 161
List of Tables

2–1 Modern filter approximations. ........................................... 11
2–2 Filter transformations. (Low-Pass parameters derive from tables) . 13
2–3 SC and SI characteristics. .................................................. 37

4–1 Summary of the Palmo-I characteristics ............................... 86

5–1 Clamped Comparator Implementation Parameters .................. 95
5–2 Summary of the Palmo FPAA characteristics .......................... 108

6–1 Summary of the BiCMOS chip ............................................ 124

7–1 Market-growth tendencies in the foreseeable future .................. 130
7–2 Characteristics of Analogue, Palmo and Digital signal processing implementations. .................................................. 135

B–1 Palmo FPAA Pin out part I .............................................. 149
B–2 Palmo FPAA Pin out part II ............................................ 150

Writing style

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

\(^1\)The 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 cir-
cuits 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 consid-
erations 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 mixed-
signal 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.

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

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].

\[ h(t) = \omega_0 \cdot \frac{\sin \omega_0 t}{t} \quad (2.2) \]
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.](image)

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 \( \omega_p \). 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 \(-20\log_{10} H(\omega)\). 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.
Figure 2–3: Typical L–C low-pass ladder filter.

Table 2–1. Modern filter approximations.

<table>
<thead>
<tr>
<th>Category</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Butterworth</strong></td>
<td>Flattest passband</td>
<td>Medium transition rate, medium distortion</td>
</tr>
<tr>
<td><strong>Chebyshev</strong></td>
<td>Fast transition ratio</td>
<td>High distortion, high passband ripple</td>
</tr>
<tr>
<td><strong>Elliptic</strong></td>
<td>Fastest transition rate</td>
<td>High distortion, high passband ripple</td>
</tr>
<tr>
<td><strong>Bessel</strong></td>
<td>Least distortion</td>
<td>Slow transition rate</td>
</tr>
<tr>
<td><strong>Gaussian</strong></td>
<td>Low distortion</td>
<td>Medium transition rate</td>
</tr>
</tbody>
</table>

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
(\delta_p, \delta_s), the transition region \((\omega_p, \omega_s)\) 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>
<thead>
<tr>
<th>Low-Pass</th>
<th>High-Pass</th>
<th>Band-Pass</th>
<th>Band-Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_{lp} = s/\omega_c )</td>
<td>( S_{hp} = 1/\omega_c )</td>
<td>( S_{bp} = s/\omega_c + 1/\omega_c )</td>
<td>( S_{bs} = \omega_2 - 1/\omega_c )</td>
</tr>
<tr>
<td>( L_c )</td>
<td>( L_c )</td>
<td>( L_c )</td>
<td>( L_c )</td>
</tr>
<tr>
<td>( C_{hp} = 1/L_{lp} )</td>
<td>( C_{bp} = 1/\omega_0^2 L_{lp} )</td>
<td>( C_{bs} = 1/\omega_0^2 L_{lp} )</td>
<td></td>
</tr>
</tbody>
</table>

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:

\[ FSF = \frac{f_n}{f_c} \]  

(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,

\[ FSF = \frac{1\text{rad/sec}}{2\pi f_c \text{rad/sec}} \]

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 2–4: Wideband band-pass filter realisation by cascading a low-pass and a high-pass filter.

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:

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 $L_k \ (1 < k \leq n)$ are given by the equations:

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

\[
v_{Lk}(s) = L_k \cdot s i_{L_k}
\]

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

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

### 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 \( \omega \) 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 \( \sin(\omega T/2) \approx \omega T/2 \) therefore (2.5) can be written as:

\[
H(z) \approx \frac{K}{T} \cdot \frac{e^{jT/2}}{j\omega} \tag{2.6}
\]
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 \( \omega T << 1 \) 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

\[
\begin{align*}
\sin(t) & \rightarrow s^n & \rightarrow \sin(nT) \\
0 & \rightarrow \n & \rightarrow 0 \\
-\pi & \rightarrow \pi & \rightarrow -\pi
\end{align*}
\]

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 $n = 1, 2, 3, \ldots$ 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

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

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 $F_{\text{max}}$ does not exceed some predetermined value, then we can select the sampling rate so that $F_s > 2 \cdot F_{\text{max}}$ (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.](image)

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 it 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.

![Block diagram of a double-slope ADC](image)

**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

![Diagram of E-A modulator](image)

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-subtractor* 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 continuous-time 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

![OTA-C integrators: block diagram and circuit details.](image)

**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 $I_{d1}$ and $I_{d2}$ are given by the equations:

$$i_{D1} = \frac{\beta}{2} \cdot (\nu_{GS1} - V_T)^2, \quad i_{D2} = \frac{\beta}{2} \cdot (\nu_{GS2} - V_T)^2$$

where $\nu_{GS} = \nu_{IN} - V_s$. Note that $I_s = i_{D1} + i_{D2}$. The two currents can be also expressed as a function of:

$$i_{D1} = I_D + i_d = \frac{I_s}{2} + \frac{\beta I_s}{4} \Delta \nu_p \sqrt{1 - \frac{\beta}{4I_s}(\Delta \nu_m)^2}$$

$$i_{D2} = I_D - i_d = \frac{I_s}{2} - \frac{\beta I_s}{4} \Delta \nu_m \sqrt{1 - \frac{\beta}{4I_s}(\Delta \nu_m)^2}$$

Therefore

$$i_{OUT} = i_{D1} - i_{D2} = 2i_d \simeq \sqrt{\beta I_s \Delta \nu_{IN}} = G_m \Delta \nu_{IN}$$

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].

![SC resistance simulation diagram]

**Figure 2–13**: SC resistance simulation.

The principle of SC circuits is shown in figure 2–13. During the phase $\Phi_1$ of the non overlapping clocks ($\Phi_1, \Phi_2$) the capacitor $C$ is charged to $Q_{in} = CV_1$. During $\Phi_2$ 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

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

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].
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 $\Phi_1$ is high, at the same time the capacitor $C_1$ is charged to the input $V_{in_{n-1}}$. During the next $\Phi_2$ 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 $\Phi_1$ and $\Phi_2$ 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 $\Phi_1$ 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 $\Phi_1$ opens 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 $\Phi_1$ 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 $\Phi_1$, the transistor $M_2$ "stores" the input current plus $I_f$ which is fed back from $M_4$. At the next clock $\Phi_2$ 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 feedthrough 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.

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

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 FPAAAs 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 FPAAAs; 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 FPAAAs 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 FPAAAs [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 log-domain 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 FPAAAs 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].
<table>
<thead>
<tr>
<th></th>
<th>Switched Capacitor</th>
<th>Switched Current</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fan-Out</strong></td>
<td>Big fan-out depending on the design of the OPAMP and the capacitance of the load.</td>
<td>Fan-Out is one per output stage therefore $n$ output stages are needed for $n$ outputs.</td>
</tr>
<tr>
<td><strong>Fan-In</strong></td>
<td>A switched capacitor circuit is needed per input therefore the fan-in is one per input stage.</td>
<td>Big fan-in since it is very easy to add currents at the input node (subject to reaching the limits of operation).</td>
</tr>
<tr>
<td><strong>Interconnection</strong></td>
<td>Limited because of coupling between analogue and digital lines, because of the line capacitance, and because of the use of analogue switches.</td>
<td>Better because currents are not so sensitive to noise and because it is easier to switch currents.</td>
</tr>
<tr>
<td><strong>Gain Control</strong></td>
<td>Can be easily controlled by changing the sampling frequency</td>
<td>Because of the need to control the ratio of transistors, the circuit needed for programmable gain is big and clumsy.</td>
</tr>
<tr>
<td><strong>Clock-Feedthrough noise</strong></td>
<td>Bad, because clock-feedthrough effects corrupt the input voltage samples resulting in voltage offsets</td>
<td>Very bad, clock feedthrough effects give high order effects resulting in harmonic distortion (as well as offsets).</td>
</tr>
<tr>
<td><strong>Features</strong></td>
<td>Sampled analogue cells, Analogue I/O (voltage), digitally controlled gain</td>
<td>Sampled analogue cells, Analogue I/O (current), transistor $W/L$ controlled gain</td>
</tr>
</tbody>
</table>

**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

![Diagram of Pulse-Based FPMA implementation](image)

**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 \( (f_N = f_s = 2 \cdot f_{max}) \).
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.](image)

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 \((A\text{ and }B)\) 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 \(T_8\). 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 pulse-based 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 \(x_{\text{in}}(t)\) 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", "1", "0", "0", "1", "0", "1", "0", "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 pulse-based 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

\[
y = f\left(\sum_{k=1}^{N} w_k \cdot x_k\right)
\]

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 \(\Delta T_{x_k}\) (figure 3-2). The charge which is accumulated on the capacitor over a cycle 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.](image)

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 “Παλμος” 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 ($K_{\text{int}}$) and the ramp ($K_{\text{ramp}}$)

$$K = \frac{K_{\text{int}}}{K_{\text{ramp}}} \quad (3.1)$$

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 sampled-data 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.

![PWM Generation Schemes](image)

**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 $Out$-
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 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

**Figure 3–5:** A) Scaling by the use of the Ramp, B) Saturated Ramp (double and single sided).
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 represented by a pulse which is ‘high’ for $\Delta T_{\text{pulse}}$ during the positive cycle of the sign clock, and a negative input value of ‘–A’ by a pulse which is ‘high’ for $\Delta T_{\text{pulse}}$ 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.
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 $\xi_+$ or the $\xi_-$ switch depending on the sign bit and other control signals. If $\xi_+$ is closed then charge is dumped onto the capacitor $C_{int}$ for the duration of the pulse $\Delta T$. If the $\xi_-$ 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 ($\xi_+$ and $\xi_-$) of the ramp generating circuit are driven directly by the signals $R_{up}$ and $R_{down}$, 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, \textit{out}, is a scaled, pulse width modulated representation of the integrated signal.

\textbf{3.4.1 Palmo Miller Integrator}

In a Miller Integrator the input at the \textit{plus} node is delayed by one clock period to the output. The input at the \textit{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 \textit{Palmo} analogue cell (figure 3-7) is defined by the digital logic block which drives the \( \xi_+ \) and \( \xi_- \) switches. The appropriate digital logic to generate the signals \( \xi_+ \) and \( \xi_- \) 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 \textit{sign} clock, \( P \) is the \textit{plus} input, \( M \) is the \textit{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 (\( C_{\text{int}} \)) is given by equation 3.2.

\[
\Delta Q_{C_{\text{int}}}(z) = I_{\text{int}} \left( \Delta T_{\text{plus}}z^{-1} - \Delta T_{\text{minus}}z^{-1/2} \right)
\]  

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

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

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

\[
V_{C_{r}}(t) = \frac{I_{r}}{C_{r}} \Delta T_{\text{out}}
\]
Palmo Techniques: The pulse based approach

and $V_{C\text{int}} = V_{C_r}$, thus:

$$\Delta T_{\text{out}} = \frac{C_r I_{\text{int}}}{C_{\text{int}} I_r} \cdot \frac{\Delta T_{\text{plus}} z^{-1} - \Delta T_{\text{minus}} z^{-1/2}}{1 - z^{-1}}$$  \hspace{1cm} (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 $\Phi$ is:

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

$$K = \frac{C_u}{C_I} = \frac{C_r}{C_{\text{int}}} \cdot \frac{I_{\text{int}}}{I_r}$$  \hspace{1cm} (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 $(\Phi)$, 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 $C_{\text{Switch}}/C$, 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
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

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 $\mu A$) 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 $\xi$ is high, transistor M1 is on, while M2 is off. This enables the voltage established on the gate of M3 by current $I_r/\text{int}$ to be transferred to the gate of transistor M4, thus discharging the capacitor with a constant current $I_r/\text{int}$. When $\xi$ is low, transistor M1 is off, while M2 is on. The voltage on the gate of M4 is now $V_{ss}$, 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 ($V_{\text{gs}}$) 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.
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 \( C_G \) 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 \( \Delta T_v \) at the PWM output.

If a double sided ramp was used, then this delay (\( \Delta T_v \)) will be the difference between raise and fall times of the comparator. Therefore the ideal cosine output (figure 3–10B) with a magnitude \( \Delta T_\mu \) 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:

\[
y(t) = \Delta T_\mu \cos(\omega t) + f(t)
\]

(3.6)

Where \( \Delta T_\mu \) is the magnitude of the desired cosine output\(^1\), and \( f(x) \) a step function given by the equation

\[
f(\omega t) = \begin{cases} 
\Delta T_v, & \frac{-\pi}{2} \leq \omega t + 2\kappa \pi \leq \frac{\pi}{2}, \text{ where } \kappa \in \mathbb{Z} \\
-\Delta T_v, & \text{otherwise}
\end{cases}
\]

\(^1\)Our 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 $\omega t$ the resultant Fourier series is a Fourier cosine series thus

$$F\{y(t)\} = F\{\Delta T_\mu \cos(\omega t)\} + F\{\Delta T_\nu f(t)\} \iff$$

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

Where $x = \omega t$,

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

And

$$\alpha_n = \frac{2}{\pi} \int_0^{\pi} f(x) \cos(nx) \, dx = \frac{2 \Delta T_\nu}{\pi n} \left[ \sin(nx) \bigg|_0^{\pi/2} - \sin(nx) \bigg|_{\pi/2}^{\pi} \right]$$

$$= \left\{ \begin{array}{ll}
0, & n = 2k \ \forall k \in \mathbb{N} \\
(-1)^{(n-1)/2} \cdot \frac{4 \Delta T_\nu}{\pi n}, & n = 2k + 1 \ \forall k \in \mathbb{N}
\end{array} \right.$$ 

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:

$$HD_i \simeq \frac{\sqrt{\alpha_2^2 + \alpha_5^2 + \alpha_7^2 + \ldots}}{\Delta T_\mu + \alpha_1} = \frac{\sqrt{(1/3)^2 + (1/5)^2 + (1/7)^2 + \ldots}}{\frac{\pi \Delta T_\nu}{4 \Delta T_\mu + 1}}$$

(3.7)

Where $\Delta T_\mu$ is the magnitude of the signed PWM cosine input and $\Delta T_\nu$ 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 $I_\nu$ and $I_\nu$ in figure 3-11, will generate harmonic distortion; because of the different magnitudes.
**Figure 3–11:** Harmonic distortion due to current differences.

$\Delta T_p$ and $\Delta T_n$ at the output $g(x)$ which is given by the equation:

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

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

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

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

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

Also

$$\alpha_1 = \frac{2}{\pi} \int_0^{\pi} g(x) \cos(x) \, dx$$
$$= \frac{2}{\pi} \left( \Delta T_p \int_0^{\pi/2} \cos^2(x) \, dx + \Delta T_n \int_{\pi/2}^{\pi} \cos^2(x) \, dx \right)$$
$$= \Delta T_p/2 + \Delta T_n/2$$

And

$$\alpha_n = \frac{2}{\pi} \int_0^{\pi} g(x) \cos(nx) \, dx$$
$$= \frac{2}{\pi} \left( \Delta T_p \int_0^{\pi/2} \cos(x) \cos(nx) \, dx + \Delta T_n \int_{\pi/2}^{\pi} \cos(x) \cos(nx) \, dx \right)$$
Because
\[
\int \cos(x) \cos(nx) \, dx = \frac{\sin((n+1)x)}{2(n+1)} + \frac{\sin((n-1)x)}{2(n-1)} \quad n > 1
\]

\(\alpha_n\) is derived:
\[
\alpha_n = \frac{2}{\pi} \left[ \Delta T_p \cdot \frac{\sin((n+1)x)}{2(n+1)} \right]_0^{\pi/2} + \Delta T_p \cdot \frac{\sin((n-1)x)}{2(n-1)} \left. \right|_0^{\pi/2} + \Delta T_n \cdot \frac{\sin((n+1)x)}{2(n+1)} \left. \right|_0^{\pi/2} + \Delta T_n \cdot \frac{\sin((n+1)x)}{2(n+1)} \left. \right|_0^{\pi/2} \]
\[
\alpha_n = \begin{cases} 
0, & n = 2k + 1 \quad \forall k \in \mathcal{N} \\
\frac{2}{(2k+1)(2k-1)\pi} (\Delta T_p - \Delta T_n)(-1)^{k+1}, & n = 2k \quad \forall k \in \mathcal{N}
\end{cases}
\]

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

\[
HD_{ii} \approx 4 \cdot \frac{\sqrt{(\frac{1}{3})^2 + (\frac{1}{5})^2 + (\frac{1}{35})^2 + \ldots}}{\frac{\Delta T_p + \Delta T_n}{|\Delta T_p - \Delta T_n|}}
\] (3.9)

Where \(\Delta T_p, \Delta T_n\) 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:

\[
THD = \frac{\sqrt{HD_{i2}^2 + HD_{i3}^2 + HD_{i4}^2 + HD_{i5}^2 + \ldots}}{\sqrt{P_{AC}}}
\]

for \(\Delta T_p/\Delta T_{\mu} \ll 1\) 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:

\[
THD \approx \sqrt{\left(\frac{1}{3}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{4}{\pi} \frac{\Delta T_{\mu}}{\Delta T_p}\right)^2 + \left(\frac{1}{3}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{4}{\pi} \frac{\Delta T_{\mu}}{\Delta T_p}\right)^2}
\] (3.10)

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

\[
THD = HD_i + HD_{ii}
\]

The comparator delays \(\Delta T_{\nu}\) are independent (to a first approximation) to the frequency of operation, however its size increases proportionally to the output \((\Delta T_{\mu})\) at high sampling frequencies. Current matching on the other hand depends heavily to the size of the currents. Therefore at different operating frequencies,
different integrating currents are used and $HD_{ii}$ will vary. Figure 3–12 shows plots of $HD_1$, $HD_{ii}$ and Total Harmonic Distortion ($THD = HD_1 + HD_{ii}$) 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 ($\mu_n$ and $\mu_p$) 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 ($A_o$) is needed for
Figure 3-14: Clamped Comparator small signal analysis.
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 $\nu_{gs1} = \nu_{gs3} = \nu_{ds3}$ 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 Kirchhoff’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

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

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

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

Subtracting (3.12) from (3.11) we obtain

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

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

\[
A_{V_d} \simeq -\frac{g_{m1}}{g_{m3} - g_{m10}} \quad (3.13)
\]
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 \(\beta_8/\beta_9\) is \(\alpha\), 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:

\[
A_{V_{out}} = -\frac{1}{2} \frac{g_{m5} + \alpha \cdot g_{m6}}{g_{ds5} + g_{ds6}}
\]

(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

\[
A_V = A_{V_d} \cdot A_{V_{out}} \approx \frac{1}{2} \cdot \frac{g_{m1}}{g_{m3} - g_{m10}} \cdot \frac{g_{m5} + \alpha \cdot g_{m6}}{g_{ds5} + g_{ds6}}
\]

### 3.7 Current Matching

The inaccuracy between the charging and discharging currents \((I_p\) 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].

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

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-

\footnote{The 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 ($I_p$ 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).](image1)

![Figure 3-17: Frequency response of the z-domain transfer function and Palmo Filter.](image2)
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

\[
FSF = \frac{1 \text{rad/sec}}{2\pi f_c \text{rad/sec}} = \frac{1}{62.8k}
\]

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 \( K'_i = \frac{1}{X T_s} \) 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 \( \xi_+ \) and \( \xi_- \) for the Miller implementation are given by the following equations

\[
\xi_+ = S \cdot P \cdot M + \bar{S} \cdot \bar{P} \cdot M \\
\xi_- = S \cdot \bar{P} \cdot M + \bar{S} \cdot P \cdot \bar{M}
\]

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 pulse-based 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
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 $\xi_+ - \xi_- \text{ and } \xi_{+2} - \xi_{-2}$, 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.
Figure 4–1: Block diagram of PALMO-I cells (signals noted with a ‘*’ are global).
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
these sources uses the switching arrangement mentioned in the previous chapter (figure 4–2). The signals $\xi_+$ and $\xi_-$ 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 ($\xi_+$ and $\xi_-$). 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].
4.4 Comparator

Because of time constraints, 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 0.75\( pF \) but process variations mean this may vary by 10%.

Figure 4–3: Comparator schematic diagram, all the NMOS bulks are connected to ground.
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 \( \frac{C_{int}}{C_{ramp}} = 4 \). 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 (\( \mu \)A). 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\( \mu \)m 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-to-signed 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 10μs for the rising edge and 11μs 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 8μA 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 50μA 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 2μA. 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 8μA 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
showed a significant improvement. The delays were 3μs for the rising edge and 4μs for the falling edge with a current of $I_r = 500nA$. Nevertheless the delay of almost 5μs 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 ±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 low-pass 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.

![Diagram](image)

**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 PWMin 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 $\xi_+$ and $\xi_-$ inputs to the *Palmo-I* chip of figure 4-1B. The third trace in Figure 4-8 is the $\xi_+$ signal, containing every second positive pulse from the integrator input (second trace). For simplicity, the $\xi_-$ has not been shown. The $\xi_+$ and $\xi_-$ 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, $x_i$, $a_i$ \(i = 0, 1, \ldots n\) are the inputs and the scaling factors respectively.

![Diagram of mixed-signal technique](image)

**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.
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.](chart)

A 24 tap FIR filter was implemented to demonstrate the use of such a mixed-signal 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.

<table>
<thead>
<tr>
<th>Table 4-1. Summary of the Palmo-I characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chip Name</strong></td>
</tr>
<tr>
<td>Number of cells</td>
</tr>
<tr>
<td>Implementation</td>
</tr>
<tr>
<td>Sampling Frequency</td>
</tr>
<tr>
<td>Power consumption</td>
</tr>
<tr>
<td>Programmability</td>
</tr>
<tr>
<td>Dynamic range</td>
</tr>
<tr>
<td>THD</td>
</tr>
<tr>
<td>Q factor</td>
</tr>
<tr>
<td>Comparator delays</td>
</tr>
<tr>
<td>Delay cancellation</td>
</tr>
</tbody>
</table>

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 pulse.
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.
Figure 5-1: Block diagram of the PALmo-FPAA chip.
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 $C_{\text{int}}$ and $C_r$ respectively. Each cell includes a six bit current DAC to charge either $C_{\text{int}}$, 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
The circuit diagram of a typical cell is shown in figure 5–2 [32,11]. Every cell has got three external inputs: \( \text{Pulse}_j \), \( \text{Up/Down}_j \), and \( \text{Int/Ramp} \), where \( j = 0, 1, \ldots 7 \) represents the cell number of figure 5–1.

The \( \text{Pulse}_j \) 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 \( \text{Int/Ramp} \) selects between the integrating and ramp SRAMs to control the current DAC output, as it is shown in the current DAC switch arrangement \( (B_{i/r_{nj}}) \) 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 \( \text{Up/Down}_j \) is responsible for changing the direction of the current DAC. This is done by adding the current mirror to the Palmo circuit (when
Up/Down 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 $B_{C_{rnj}}$ which are controlled by the capacitor SRAM. Care should be taken by the user not to short-circuit the integrating to the ramp capacitors.

### 5.3.2 Comparator Design

![Comparator Schematic](image)

**Figure 5-3:** Palmo FPAA comparator schematic.

The comparator used in our 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 $V_{fixn}$ and $V_{fixp}$ force some current to be sunk 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 ($V_{fixn}$ and $V_{fixp}$) 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:
Palmo FPAA and prototyping board

- $A_v = 2000$
- $CMR = 6 - 8V$
- $V_{DD} = 12V, V_{SS} = 0V$
- 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 $V_{DS_{(sat)}} < 2V$

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

Similarly for $M_6$

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

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

$$A_{v_{out}} = -\frac{1}{2} \frac{g_{m5} + g_{m6}}{g_{ds5} + g_{ds6}} = 11.06$$

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.

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

Therefore

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

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

<table>
<thead>
<tr>
<th>Transistor width over length</th>
<th>Size (W/L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4($W/L)_8 = (W/L)_5$</td>
<td>2.89</td>
</tr>
<tr>
<td>4($W/L)_9 = (W/L)_6$</td>
<td>9.7</td>
</tr>
<tr>
<td>($W/L)_3 = (W/L)_4$</td>
<td>0.28</td>
</tr>
<tr>
<td>($W/L)_1 = (W/L)_2 = 15</td>
<td>90μ/6μ</td>
</tr>
<tr>
<td>($W/L)_10 = (W/L)_11$</td>
<td>0.20</td>
</tr>
<tr>
<td>($W/L)_7 = 1.56</td>
<td>9μ/6μ</td>
</tr>
</tbody>
</table>

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 ($Out_0 \ldots Out_7$) will all be high. When enable turns high the transistor parasitic capacitance will maintain all 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

![Address Decoder](image)
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

![SRAM Implementations Diagram](image)

**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 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 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.

\[\text{Figure 5–6: Input voltage level shifter.}\]
5.4 The prototyping Board

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 ↔ 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 through 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.
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 2.4μm 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 $f_{clk}$. The address bits are shifted into the 8 bit shift register (RS8), and then the $\text{Latch}$ 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 $\text{ALE}$ 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 $\text{LOAD}$ 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 $\text{Latch}$ 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 $V_{ref}$ signal of the FPAA to a ramp generated externally, while the $\text{Input}$ node was connected to a constant voltage. By changing the global interconnect data, we switched on the transmission gate, which connects the $\text{Input}$ to cell 5. The output of cell 5 consequently followed
Figure 5-11: Addressing in the FPGA \((address = 80h)\).

Figure 5-12: Loading “Ah” 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 func-
tions 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 suitab-
ility 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
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
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.
Table 5–2. Summary of the Palmo FPAA characteristics

<table>
<thead>
<tr>
<th>Chip Name</th>
<th>Palmo-FPAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cells</td>
<td>8</td>
</tr>
<tr>
<td>Implementation</td>
<td>Voltage-domain Palmo circuit</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>1kHz to 500kHz</td>
</tr>
<tr>
<td>Power consumption</td>
<td>0.95mW per cell, at 50kHz</td>
</tr>
<tr>
<td>Programmability</td>
<td>By the use of an external FPGA and reconfigurable DACs-capacitor array.</td>
</tr>
<tr>
<td>Dynamic range</td>
<td>54dB</td>
</tr>
<tr>
<td>THD</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>Q factor</td>
<td>70dB</td>
</tr>
<tr>
<td>Comparator delays</td>
<td>0.3(\mu)s rise 1(\mu)s fall</td>
</tr>
<tr>
<td>Delay cancellation</td>
<td>minimum pulse and positive feedback currents at the comparator</td>
</tr>
</tbody>
</table>

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.
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 $10^6$ 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 $I_s$ for $V_b=700$ mV 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 $I_{int}$. Log-domain 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 $Q_1$ ($Q_8$); $Q_2$ and $C_1$ ($Q_7$, $C_2$) perform integration in the log-domain. The integrated value is scaled by $Q_3$ ($Q_6$), the scaling factor depends on the current $I_t$. Finally, $Q_4$ ($Q_5$) expands the compressed signal. The output is represented by the difference of the currents $I_{O_1}$ and $I_{O_2}$. 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:

$$\frac{I_{C1}}{1 + \frac{V_{BC1}}{V_{AF}}} \cdot \frac{I_{C3}}{1 + \frac{V_{BC3}}{V_{AF}}} = \frac{I_{C2}}{1 + \frac{V_{BC2}}{V_{AF}}} \cdot \frac{I_{C4}}{1 + \frac{V_{BC4}}{V_{AF}}}$$

$$\frac{I_{C5}}{1 + \frac{V_{BC5}}{V_{AF}}} \cdot \frac{I_{C6}}{1 + \frac{V_{BC6}}{V_{AF}}} = \frac{I_{C7}}{1 + \frac{V_{BC7}}{V_{AF}}} \cdot \frac{I_{C8}}{1 + \frac{V_{BC8}}{V_{AF}}}$$

(6.1)

where $V_{BCi}$ 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 \( \beta \) is the current gain of the bipolar transistors. Substituting these currents to the equations (6.1) yields:

\[
\begin{align*}
\frac{I_p}{1 + \frac{V_{BC}}{V_{AF}}} \cdot \frac{I_t}{1 + \frac{V_{BC}}{V_{AF}}} &= \left(\frac{aI_{O2} - I_{cap1} - \frac{I_t}{\beta} + I_c}{1 + \frac{V_{BC}}{V_{AF}}} \right) \left(1 - \frac{1}{\beta}\right) \cdot \frac{I_{O1}}{1 + \frac{V_{BC}}{V_{AF}}} \\
\frac{I_n}{1 + \frac{V_{BC}}{V_{AF}}} \cdot \frac{I_i}{1 + \frac{V_{BC}}{V_{AF}}} &= \left(\frac{bI_{O1} - I_{cap2} - \frac{I_t}{\beta} + I_c}{1 + \frac{V_{BC}}{V_{AF}}} \right) \left(1 - \frac{1}{\beta}\right) \cdot \frac{I_{O2}}{1 + \frac{V_{BC}}{V_{AF}}}
\end{align*}
\]

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:

\[
\begin{align*}
\frac{I_{O1}}{I_{O2}} &= \frac{\frac{I_t e^{-\frac{V_c}{V_{th}}}}{V_{th}}}{V_{th} \cdot \frac{I_t e^{-\frac{V_c}{V_{th}}}}{V_{th}}} = \frac{I_{cap1}}{C_i \cdot V_{th}} \\
\frac{d}{dt} I_{O1} &= -\left(\frac{I_t e^{-\frac{V_c}{V_{th}}}}{V_{th}}\right) \frac{1}{V_{th}} \cdot \frac{d}{dt} V_{cap1} = -\frac{I_{cap1}}{C_i \cdot V_{th}} \\
I_{cap1} &= -C_i \cdot V_{th} \cdot \frac{1}{I_{O1}} \cdot \frac{d}{dt} I_{O1}
\end{align*}
\]

where \( V_{th} \) is the thermal voltage (26mV at 300°C), \( V_b \) is the biasing voltage (Figure 6–3). Subtracting the two equations (6.3), substituting \( I_{cap1} \) from (6.4) and \( k_i = 1 + V_{BC}/V_{AF} \), while assuming \( C_1 = C_2 = C \) and \( 1/(1 - 1/\beta) \approx 1 + 1/\beta \) results to the final equation:

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

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]

\[
K = \frac{I_{int}}{I_{tramp}} \cdot \frac{C_{tramp}}{C_{int}}
\]

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
dynamic range, because temperature variations will alter the currents $I_{\text{tramp}}$ and $I_{\text{int}}$ 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
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

**Figure 6–4:** Current comparator
Advanced implementations: Log-domain BiCMOS Palmo cells

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](image)

This circuit (figure 6-5) will be used to generate input currents as well as the biasing currents ($I_i$) 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 log-domain chip) is shown in figure 6–5. It is based on [137], but needs only one supply
Advanced implementations: Log-domain BiCMOS Palmo cells

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:

\[
I_{DM_i} = \frac{\beta}{2} \left( V_{dd} - V_{D_i} - \frac{V_{in}}{2} - V_T \right)^2
\]

where \( \beta = \mu p C_{ox} (W/L) \), \( V_{D_i} \) 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 non-linearities 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 ±15μA. 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 20μA. 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, $f_s=1$MHz.
6.6 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 $\Phi$ is used to add the integrating capacitor ($C_{ii}$ and $C_{iiii}$) in series. Therefore when $\Phi$ 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

$$V_{1i} = V_b + K \log(I_{in} \Delta_{in})$$

By adding the two voltages we get

$$V_{i+iii} = 2V_b + K \log [I_{in}(\Delta T_A + \Delta T_B)]$$
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 ($C_{ii}$) 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 ($\Phi$) 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 $C_{iii}$ ($\Phi$). 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 sampled-data 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:

<table>
<thead>
<tr>
<th><strong>Chip Name</strong></th>
<th><strong>BiCMOS-I</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cells</td>
<td>4</td>
</tr>
<tr>
<td>Implementation</td>
<td>Log-domain Palmo circuit</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>10kHz to 5MHz</td>
</tr>
<tr>
<td>Power consumption</td>
<td>2.2mW (per cell, mostly due to the current comparator)</td>
</tr>
<tr>
<td>Programmability</td>
<td>By the use of V-I converters and 3bit programmable capacitors</td>
</tr>
<tr>
<td>Dynamic range</td>
<td>70dB</td>
</tr>
<tr>
<td>THD</td>
<td>Maximum 0.94%, typical &lt; 0.5%</td>
</tr>
<tr>
<td>Q factor</td>
<td>80dB</td>
</tr>
<tr>
<td>Comparator delays</td>
<td>17ns for minimum input</td>
</tr>
<tr>
<td>Delay cancellation</td>
<td>dual-slope ramp</td>
</tr>
</tbody>
</table>
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
Developments and Conclusions

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 log-domain 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 today's 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 0.241 $\mu$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.

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 op-
eration, increased integration, low-power consumption, device quality and cost issues.

7.4.1 Future Work

The Palmo approach, developed during 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.

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

Figure 7–3: Dedicated minimum pulse cancellation.
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 greater 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.
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 “Παλμος”. 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
Developments and Conclusions

Table 7-2. Characteristics of Analogue, Palmo and Digital signal processing implementations.

<table>
<thead>
<tr>
<th></th>
<th>Analogue</th>
<th>Palmo</th>
<th>Digital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmability:</td>
<td>limited</td>
<td>fair</td>
<td>extended</td>
</tr>
<tr>
<td>Frequency of operation:</td>
<td>high</td>
<td>adequate</td>
<td>slow</td>
</tr>
<tr>
<td>Power consumption:</td>
<td>low</td>
<td>average</td>
<td>extremely high</td>
</tr>
<tr>
<td>Signal robustness:</td>
<td>low</td>
<td>adequate</td>
<td>highest</td>
</tr>
<tr>
<td>Size:</td>
<td>small</td>
<td>smallest</td>
<td>biggest</td>
</tr>
<tr>
<td>Design time:</td>
<td>long</td>
<td>shortest</td>
<td>short</td>
</tr>
<tr>
<td>Design CAD tools:</td>
<td>limited</td>
<td>available</td>
<td>extended</td>
</tr>
<tr>
<td>Timeliness:</td>
<td>old</td>
<td>just emerged</td>
<td>at their peek</td>
</tr>
</tbody>
</table>

\(^a\text{In general, for programmable analogue hardware the Palmo approach is best.}\)

\(^b\text{Time needed for solving a signal processing task.}\)

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] [88] [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]


Appendix A

Palmo FPAA Addressing Registers

8 bit address register

<table>
<thead>
<tr>
<th>A7</th>
<th>A6</th>
<th>A4</th>
<th>A3</th>
<th>A2</th>
<th>A1</th>
<th>A0</th>
</tr>
</thead>
</table>

A1 A0 → Load directly-to-cell data
A4 A3 A2 → Determine the cell (cells are numbered 0 to 7)
A7 A6 →

<table>
<thead>
<tr>
<th>A7</th>
<th>A6</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>access the cells</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>access interconnect: Lower data</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>access interconnect: Upper data</td>
</tr>
</tbody>
</table>

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 FPAAAs on board.
- $A_6$ and $A_7$ are used to access the global interconnect.
### Figure A–2: FPAA interconnect registers

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

<table>
<thead>
<tr>
<th>Upper interconnect data register</th>
<th>Lower interconnect data register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramp 6-7</td>
<td>Int 6-7</td>
</tr>
<tr>
<td>Ramp 3-7</td>
<td>Int 3-7</td>
</tr>
</tbody>
</table>

![Interconnect Registers Diagram]

### 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

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

*Table B–1. Palmo FPAA Pin out part I*
<table>
<thead>
<tr>
<th>Description</th>
<th>Pad #</th>
<th>PGA</th>
<th>Pad-Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiComp</td>
<td>38</td>
<td>K8</td>
<td></td>
</tr>
<tr>
<td>Out3</td>
<td>39</td>
<td>L9</td>
<td>OI2</td>
</tr>
<tr>
<td>Nin</td>
<td>40</td>
<td>L10</td>
<td>CPAI</td>
</tr>
<tr>
<td>Vssa</td>
<td>41</td>
<td>K9</td>
<td>CPVSSA</td>
</tr>
<tr>
<td>PIN</td>
<td>43</td>
<td>K10</td>
<td>CPAI</td>
</tr>
<tr>
<td>CompPulse</td>
<td>44</td>
<td>J10</td>
<td>IC</td>
</tr>
<tr>
<td>MinOut</td>
<td>45</td>
<td>K11</td>
<td>OI2</td>
</tr>
<tr>
<td>Pulse3</td>
<td>46</td>
<td>J11</td>
<td>IIC</td>
</tr>
<tr>
<td>Int/Ramp3</td>
<td>47</td>
<td>H10</td>
<td>IC</td>
</tr>
<tr>
<td>Up/Down3</td>
<td>48</td>
<td>H11</td>
<td>IC</td>
</tr>
<tr>
<td>Up/Down2</td>
<td>50</td>
<td>G10</td>
<td>IC</td>
</tr>
<tr>
<td>Int/Ramp2</td>
<td>51</td>
<td>G11</td>
<td>IC</td>
</tr>
<tr>
<td>Pulse2</td>
<td>52</td>
<td>G9</td>
<td>IIC</td>
</tr>
<tr>
<td>Out2</td>
<td>55</td>
<td>E11</td>
<td>OI2</td>
</tr>
<tr>
<td>Out1</td>
<td>56</td>
<td>E10</td>
<td>OI2</td>
</tr>
<tr>
<td>Pulse1</td>
<td>57</td>
<td>E9</td>
<td>IIC</td>
</tr>
<tr>
<td>Int/Ramp1</td>
<td>58</td>
<td>D11</td>
<td>IC</td>
</tr>
<tr>
<td>Up/Down1</td>
<td>59</td>
<td>D10</td>
<td>IC</td>
</tr>
<tr>
<td>Up/Down0</td>
<td>60</td>
<td>C11</td>
<td>IC</td>
</tr>
<tr>
<td>Int/Ramp0</td>
<td>61</td>
<td>B11</td>
<td>IC</td>
</tr>
<tr>
<td>Pulse0</td>
<td>62</td>
<td>C10</td>
<td>IIC</td>
</tr>
<tr>
<td>Reset1</td>
<td>63</td>
<td>A11</td>
<td>CPDI</td>
</tr>
<tr>
<td>Vssa</td>
<td>65</td>
<td>B9</td>
<td>CPVSSA</td>
</tr>
<tr>
<td>Load</td>
<td>67</td>
<td>A9</td>
<td>CPDI</td>
</tr>
<tr>
<td>Vdd 5V</td>
<td>68</td>
<td>B8</td>
<td>VDD</td>
</tr>
<tr>
<td>Vddh</td>
<td>70</td>
<td>B6</td>
<td>PadCon</td>
</tr>
<tr>
<td>Vss</td>
<td>74</td>
<td>C6</td>
<td>VSS</td>
</tr>
<tr>
<td>Out0</td>
<td>76</td>
<td>A5</td>
<td>OI2</td>
</tr>
<tr>
<td>Vref</td>
<td>77</td>
<td>B5</td>
<td>PVREF</td>
</tr>
<tr>
<td>Out4</td>
<td>79</td>
<td>A4</td>
<td>OI2</td>
</tr>
<tr>
<td>Pulse4</td>
<td>80</td>
<td>B4</td>
<td>IIC</td>
</tr>
<tr>
<td>Reset0</td>
<td>81</td>
<td>A3</td>
<td>CPDI</td>
</tr>
<tr>
<td>Vssa</td>
<td>83</td>
<td>B3</td>
<td>CPVSSA</td>
</tr>
</tbody>
</table>

**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...) : Sends bytes to the DAC
- F xd xa : Sends the byte 'xd' to the FPGA address 'xa'
- H or ? : Prints this help
- I xa : Inputs P1
- O 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 compatible chip. The microcontroller code has interesting subroutines to perform automatic BAUD detection program the FPGA and configure the DAC.
Microcontroller Code

: Define the I/O pins on the 8051

.equ done, 90h
.equ cclk, 91h
.equ din, 92h
.equ prog, 93h
.equ init, 94h
.equ M1, 95h
.equ M0, 96h
.equ DAC_CS, 97h
.equ OFF, 98h
.equ OFF_DDR, 99h
.equ dFPGA, 0xb3
.equ AddrPalmo, 0xb4
.equ DatPalmo, 0xb5
.equ fclk, 9ah

.mainloop: clr in7 ; Initialize stack Pointer
mov SP,#30h ; Initialize SP
mov Pl,#0ffh ; Make sure that P1 had the state it would have after reset (useful only for use with PAULMON) is OK.
acall autobaud
clr AddrPalmo ; Initialize the palmo Control lines
clr DatPalmo
mov R1,#25
w1cmLOOP:

mov dptr,#NLCR ; Clear the screen
acall print
djnz R1,w1cmLOOP
mov dptr,#welcome ; Prints the welcome message
acall print

prompt: ; Prompt for commands and command interpretation
commandloop: mov dptr,#prompt ; Print prompt
acall print
getcmd: acall getc ; Wait for serial input
cjne a,#' ',cmd0
; If command = ' ' , do nothing

cmd0:

acall putc
; Check for "ENTRY"

cmd1:
cjne a,#'X',cmd2
; If command = 'X' download
acall xilinx
; XILINK configuration routines

cmd2:
cjne a,#'D',cmd3
; Set DAC
acall xilinx

cmd3:
cjne a,#'O',cmd4
; Set a MC port
acall xilinx

cmd4:
cjne a,#'I',cmd5
; Read a MC port
acall xilinx

xilinx:
mov dptr,#xilwelcome
acall print
;
; XILINK configuration routines

not_present: nov dptr,#no_fpga_msg ; Error message because the
acall print ; INIT line did not go low
ret

file_err: nov dptr,#err_msg ; The chars transmitted were not
acall print ; a XILINK BIT file
ret

xilinx:
mov dptr,#xilinx ; Prins message
acall print
;
; XILINK configuration routines

: Download Configuration DATA
ajmp FPGA ; To the FPGA

: Print HELP
ajmp help

: Print HELP
ajmp help

: Readback from DAC
ajmp readback

: Print HELP
ajmp help

: Readback from DAC
ajmp readback

: Print HELP
ajmp help

: Print HELP
ajmp help

: Print HELP
ajmp help

: Print HELP
ajmp help

: Print HELP
ajmp help

: Print HELP
Microcontroller Code

Apr 16 1998 17:07:43

cjne a,#9,file_err ; Checking for the validity
acall getb ; of the bitfile

if the bitfile

; assume the bit file is arriving... we better
; start the program sequence before the real
; data starts to show up

setb din ; Initialize all inputs
setb init ; And outputs to '1'
setb clk ; (Port 1 has floating Gates
setb done ; with Internal 10k pull-ups)
setb M1
setb M0
clr prog ; Send the PROGRAM command
mov r0,#240 ; Which resets and clears the
djnz r0,* ; FPGA and delay for reset
;
;jb done, not_present ; Check for Presence of FPGA
clr din

;jb done, not_present ; Set PROG back to high
; to enable programming

xilinx bitstream downloader, Paul Stoffregen, Mar 1996

beta version 0.2

To use this thing, just connect the 8051's uart to your
serial port (via driver/receiver chip) and attach the
five necessary lines from port 1 to your xilinx chip.
Just send the binary bit file directly to the serial
port (at the right baud rate) and this code will remove
the bit file's header and download the data into your
xilinx chip. There is little to no error checking for
3000 series parts, so be careful.

--Paul Stoffregen (paul9ece.orst.edu )

to do:
-
s add status led outputs -download 'started' and 'finished'
-> check for another device (xchecker) starting a download
and go into high impedance until it's finished
-> add mode input (shown on schematic) and it master serial
mode is selected general program pulse in response to
reset pin but then let the serial prom do the work
-> better messages to host computer in the unlikely event
someone is using a terminal program and actually reading it
-> add jumber and extra line to OE/R pin on serial prom and
support multiple consecutive configurations in the proms.
-> check that init signal goes low before we see it go
high so the chip-not-connected error can be detected
; instead of just downloading into nowhere and thinking it
; was successful because of pull-up resistors.

setb done
mov dph, #0

wait255:
acall getb
inc dptr
cjne a,#255,wait255 ; Gets byte from the serial link
mov r2,#0 ; the REAL data for the FPGA are
jmp shift ; further down the line when the
next: acall getb
inc dptr
; data becomes a series of 'F'
shift: mov b,r5 ; MAIN FPGA LOOP
mov r4,b ; r5,6,7 are used to report an error
mov r6,b ; if the INIT line goes high
mov r7,a
mov r8,#8 ; Set counter for programming 8 bits
inc r2

sh_loop:
inc r2
mov b,r5 ; Shifting is needed to update r5-7
mov r7,a
mov r8,#8 ; Set counter for programming 8 bits
inc r2

wait255:
acall getb ; Check the INIT line for an
inc r2
mov r7,a ; error reported by the FPGA
jmp shift ; delay

acall clkdelay ; IF DONE went high the bitstream
was sucessfully downloaded

; now that it says it's done, it needs a few more
; cclk pulses to actually start up... see pg 2-29
; in 1993 xilinx databook

; in case a 3000 series .bit file left it low
mov r0,#32 ;need more cclk pulses to finish startup

startup:
call clk
nop
nop
acall clkdelay

djnz r0, startup
mov dptr,#msg_done
acall print
acall delay
acall delay
acall delay
ret ; FPGA initialised

error: if we get here, it means the xilinx chip pulled
; init low to tell us it got a checksum error!

mov a,'E'
acall putc
mov a,'r'
acall putc
acall putc
mov a,'@'
acall putc
mov a,dph ;offset in .bit file where error detected
acall prhex
mov a,dpl
acall prhex
mov a,'r'
acall putc
mov a,'t'
acall putc
mov a,'p'
acall putc
mov a,'u'
acall putc
mov a,'a'
acall putc
mov a,'d'
acall putc
mov a,'b'
acall putc
mov a,'c'
acall putc
mov a,'r'
acall putc
mov a,'c'
acall putc
mov a,'h'
acall prhex
Microcontroller Code

```assembly
mov a, #'
acall putc
mov a, r7
acall prhex
mov a, #'
acall putc
acall putc
acall putc
mov a, r2
acall prhex
ret

msg_done: .db "Done signal went high",13,10,0
xilwelcome: .db 13,10,"Xilinxx Bitstream Downloader",13,10,0
ferr_msg: .db "Transmitted file is not a XILINX 4000 bit-stream",13,10,0

delay: mov r3, #200
delay2: nap
mov r2, #228
djnz r2, *
djnz r3, delay2
ret

output: scall rdbyte ; Get the data byte
jc outabort ; if not hex abort
cri a,#OFF_prog
mov r1,a
ajmp commandloop

outabort: scall DPTR,#outabortmsg ; Report error
ajmp commandloop

outabortmsg: .db "Output aborted !",0

input: mov a, #'
scall putc
mov a,r1
scall prbyte
ajmp commandloop

help: mov DPTR,#helpmsg
scall print
mov a,#7Eh ;DAC comand for readback
acall sendDACbyte ;sent readback command
setb cclk ;sent a dummy pulse needed
nop ;by the DAC
nop ;delay

scall clkdelay ;send of clk pulse
setb din ;When a pin is set it can be used as an input
nop ;used as an input
nop ;by the DAC
nop ;delay

rdbackbytes:
mov r0, #
mov a, #'
acall putc ;print space

rdbckloop: setb cclk ;clock DAC
nop
scall clkdelay ;get bit from DAC
cir cclk ;stop the clock pulse
mov c,din ;get the data byte
acall sendFPGA ;sent it to the FPGA
setb DAC_CS ;Reset control lines to inputs
setb cclk
ajmp commandloop

sendDACbyte:
clr clck ;Prepare clck for transmission
clr DAC_CS ;set DAC-low

sendDACbits:
mov r0, #8 ;set counter r0 for 8 bits
mov a, #0 ;rotate to get the bit
setb clck ;Up-going clock
nop ;wait for propagation delays
scall clkdelay ;send an address to the FPGA
clr clck ;zero clck
djnz r0, sendDACbyte ;Repeat 8 times for 8 bits
ret

readback:
mov DPTR,#HLDR
scall print
mov r0, #8
acall sendDACbyte ;sent readback command
setb cclk ;sent a dummy pulse needed
nop ;by the DAC
nop ;delay

scall clkdelay ;send of clck pulse
setb din ;When a pin is set it can be used as an input
nop ;used as an input
nop ;by the DAC
nop ;delay

FPGA: ;Sent DATA to the FPGA
dr fclk
acall rdbyte ;Get the data byte
je ExitFPGA ;if not hex exit
mov b,a ;Store temporary the command
scall rdbyte ;Get the data
je setDACend ;if not hex exit
push ACC ;save temporary the data
mov a,b ;get the command data
scall sendDACbyte ;sent command
pop ACC ;get temporary saved data
scall sendDACbyte ;sent data
setb DAC_CS ;restore pins
setb clck
lcall commandloop
```

Microcontroller Code

```assembly
sendDACbyte:
clr clck ;Prepare clck for transmission
clr DAC_CS ;set DAC-low

sendDACbits:
mov r0, #8 ;set counter r0 for 8 bits
mov a, #0 ;rotate to get the bit
setb clck ;Up-going clock
nop ;wait for propagation delays
scall clkdelay ;send an address to the FPGA
clr clck ;zero clck
djnz r0, sendDACbyte ;Repeat 8 times for 8 bits
ret

readback:
mov DPTR,#HLDR
scall print
mov r0, #8
acall sendDACbyte ;sent readback command
setb cclk ;sent a dummy pulse needed
nop ;by the DAC
nop ;delay

scall clkdelay ;send of clck pulse
setb din ;When a pin is set it can be used as an input
nop ;used as an input
nop ;by the DAC
nop ;delay

FPGA: ;Sent DATA to the FPGA
dr fclk
acall rdbyte ;Get the data byte
je ExitFPGA ;if not hex exit
mov b,a ;Store temporary the command
scall rdbyte ;Get the data
je setDACend ;if not hex exit
push ACC ;save temporary the data
mov a,b ;get the command data
scall sendDACbyte ;sent command
pop ACC ;get temporary saved data
scall sendDACbyte ;sent data
setb DAC_CS ;restore pins
setb clck
lcall commandloop
```
Microcontroller Code

```assembly
nop
nop
nop ;long wait

setb DelayPalmo ;Done sending the byte

ExitFPGA:
setb 1DFFGA ;set FPGA LATCH & folk to high again
setb folk
ajmp commandloop

sendFPGA:
clr folk ;Prepare folk for pulsing bits in

mov r0,#8 ;set r0 to 8 for 8 bits
mov din,c ;output the bit
setb folk ;clock the FPGA

nop
nop

sendFPGAbits:
rci a ;rotate to obtain the bit
mov din,c ;output the bit
setb folk ;clock the FPGA

nop
nop ;wait

acall clkdelay

dr folk ;zero the fclk pulse

djnz r0,sendFPGAbits ;Repeat 8 times

ret

clkdelay: push acc

mov a,#10

clkloop:

djnz a,clkloop

ret

print: push acc ;Save the ACC value

print1: mov a,#0 ;zero pointer to get the first char

acall phex ;Print DIGIT 1

pop acc ;Restore ACC

push acc ;and save it to return with it

acall phex

pop acc ;Restore ACC and return

ret

putc: jnb scan1,putc ;wait until prev byte is send

mov sbuf,a ;transmit a byte

clr scan1 ;clear scan for next byte

ret

getc: jnb ri,getc ;wait until the byte is received

clr ri ;clear flag for next byte

mov a,subf ;get byte

ret

getc: jnb ri,getc

clr ri ;get char

mov a,subf ;get low-case

cjs a,#0h,getc_NQ ;check if low-case

getc_NQ: inc fixasc ;if it is low-case goto fixasc

HEXDAT: .DB "0123456789ABCDEF"

rdbyte: ;Reads a byte (2 characters) from the serial link

acall getc ;Get the char

acall putc

cjne a,#',rdbyte_l ;Skip if it is space

sjmp rdbyte

rdbyte_l:

push ACC ;Save input (in case of an error)

acall phex ;The actual transformation from ASCII to the ACC

push a ;Make the 4 bits MSBits (they were in first)

inc a ;If ACC is 0xff there was an error in the HEX no

rdbyterr:

pop ACC ;No error occured so input is useleess

acall getc

acall putc ;Save input (in case of an error)

acall phex ;Value is ASCII

cjne a,'#ffh,exitrdbyte ;Check 0xff for HEX error

rdbyterr

POP ACC

setb c ;Set C to report error the error in HEX no

ret

exitrdbyte:

orl a,123 ;Fix the returned Byte

acall getc

acall putc

inc ACC ;Save input (in case of an error)

acall hexdig

cjne a,#15,hexdig_2 ;Compare to 15

ret

hexdig_2:

inc fixasc ;If ACC >15 error

ret

hexdig: Returns

ACSI to hex or 0xff on error

subb a,'#O' ;Subtract 48 from the ASCII input

jc hexdigerr ;If one of chars

subb a,'#A'-'O' ;Clear the byte if >= A

jc hexdigerr ;IF of chars < '0' or <'A' error

add a,#10 ;Put back the 10 which is missing in the number

cjne a,#15,hexdig_2 ;Compare to 15

ret

hexdig2:

cjne input > 9 check for A-B-C-D-F-F

ret ;Char <9 OK.

hexdigerr:

clrc ;Return ASCI to hex or 0xff on error

subb a,'#0' ;Subtract 48 from the ASCII input

jc hexdigerr ;If char < 48 report error

cjs a,#9,hexdig_l ;Compare input to 9

hexdig_l:

jnc hexdig2 ;if input < 9 check for A-B-C-D-F-F

ret ;Char <9 OK.
```
Microcontroller Code

hexdigiterr:
    mov a,#0xff
    ; Error occurred return 0xff
    ret

To set the baud rate, use this formula or set to 0 for auto detection
baud_const = 256 - (crystal / (12 * 16 * baud))

.equ baud_const, 0
.equ baud_const, 255
.equ baud_const, 252
.equ baud_const, 250

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 LBS 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:
    mov tmod, #0011
    ; get timer #1 ready for action (16 bit mode)
    mov tcon, #00
    clr a
    mov th1, a
    mov t1l, a
    mov a, #baud_const
    jnz autobaud
    setb tcon
    ; start timer #1

autob2
    jb p3.0, * ; wait for start bit
    jb 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
    jsr autob2
    setb tcon
    ; and now we're timing it
    jb p3.0, * ; wait for bit #0 to begin
    jb p3.0, * ; wait for bit #1 to begin
    jb p3.0, * ; wait for bit #2 to begin
    jb p3.0, * ; wait for bit #4 to begin
    clr tcon
    ; stop timing
    mov a, t1l
    ; save bit 6 for rounding up if necessary
    mov f0, a
    mov c, acc.7
    ; grab bit 7... it's the lab we want
    mov th1, a
    rlc a
    ; do the div by 128
    mov c, f0
    ; round off if necessary
    cli
    ; invert since timer #1 will count up
    mov a, #252
    ; now acc has the correct reload value (I hope)
    autobaud
    mov th1, a
    mov t1l, a
    mov tmod, #00x21
    ; set timer #1 for 8 bit auto-reload
    mov tcon, #0x80
    ; configure built-in uart
    mov econ, #0x52
    setb t1l
    ; start the baud rate timer
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:

<table>
<thead>
<tr>
<th>F0100</th>
<th>F0200</th>
<th>F0400</th>
<th>F0500</th>
<th>F0600</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0800</td>
<td>F0900</td>
<td>F0A00</td>
<td>F0C00</td>
<td>F0D00</td>
</tr>
<tr>
<td>F0E00</td>
<td>F1000</td>
<td>F1100</td>
<td>F1200</td>
<td>F1400</td>
</tr>
<tr>
<td>F1500</td>
<td>F1600</td>
<td>F1800</td>
<td>F1900</td>
<td>F1A00</td>
</tr>
<tr>
<td>F1C00</td>
<td>F1D00</td>
<td>F1E00</td>
<td>F8000</td>
<td>FC080</td>
</tr>
</tbody>
</table>

D.2 Board Schematic Diagrams

The board schematic diagrams are presented in the following two pages.
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-2: Tap input configuration

Figure E-3: *signed* PWM output generation

Figure E-4: Ramp generating cell
Appendix F

Publications


PULSE BASED SIGNAL PROCESSING: VLSI IMPLEMENTATION OF A PALMO FILTER

K. Papathanasiou
Alister Hamilton
Department of Electrical Engineering, University of Edinburgh, Scotland, EU.
Kos.Papathanasiou@ee.ed.ac.uk
A.Hamilton@ee.ed.ac.uk

ABSTRACT
A new VLSI signal processing implementation technique is presented that uses a Pulse Width Modulation (PWM) signal representation combined with simple analogue processing to produce an electronically-programmable, process-tolerant filter building block. The principle benefits of this technique include full programmability, reconfigurability and testability which makes the technique an attractive proposition for VLSI. A 4th order Palmo filter is simulated and compared with the ideal response to demonstrate the validity of this novel approach. Preliminary results from a working chip demonstrate the operation of an analogue to PWM signal converter and the Palma integrator.

1. INTRODUCTION
This paper presents a new approach to VLSI filter implementation combining a pulse-based signal representation with simple analogue processing. The resultant filter building block is simple, compact, robust and programmable. The basic operating principles of this novel filter structure are presented here together with simulation results from a 4th order filter section and preliminary results from a VLSI device.

Our interest in filter implementation has arisen from a requirement to preprocess data for our pulse based neural network chips [1, 2, 3], in order to extend their area of application. While assessing conventional techniques such as digital [4], analogue continuous-time [5, 6, 7, 8] switched-capacitor (S—C) [9] and switched-current (S—I) [10, 11, 12] we realised that a pulse based signal processing approach offered distinct advantages.

2. SIGNAL REPRESENTATION
In the pulse width modulated signal representation proposed here, the magnitude of a signal is represented by the duration of a pulse, while the sign is determined by whether the pulse occurred in the positive or negative cycle of a global sign clock. This representation has the advantages that a zero signal value results in the absence of any transients and that, apart from the global sign clock, there is only one data line for each signed pulse signal. This reduces noise around the zero signal level and reduces the amount of interconnect required.

3. PALMO INTEGRATOR AND SCALER
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.

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

Consider the circuit of Figure 1. The input pulses are directed from the plus and minus nodes to either the + or the - switch depending on the sign bit. If the + switch is closed then charge is dumped onto the capacitor $C_{in}$ for the duration of the pulse $\Delta T$. If the - switch is closed, charge is removed from $C_{in}$. Suppose that a pulse of width $\Delta T_i$ arrives at the input to the Palmo filter block of Figure 1, resulting in the closure of $\xi_1$ for time $\Delta T_i$. The resultant voltage on the integrating capacitor is defined by equation 1.

$$V_{out}(t) = \frac{1}{C_{int}} \int I_{in} \, dt = \frac{C_{in}}{C_{int}} \Delta T_i$$

(1)

In order to generate a scaled output pulse representation of this signal, $V_{out}$ is compared to a linear ramp voltage. The pulse output starts at the beginning of each positive or negative ramp and ends when the ramp voltage on the capacitor $C_{ramp}$ becomes equal to the voltage on the integrating capacitor $C_{int}$. The combination of the comparator, XOR gate and the global sign clock ensures the regeneration of the signed pulse representation described earlier.

When the voltage $V_{ cmp}$ becomes equal to the voltage on $V_{ ramp}$, the comparator output will change state, defining the end of the pulse-width output. At this time the voltage on the ramp capacitor is

$$V_{ramp}(t) = \frac{I}{C_{ramp}} \Delta T_{ramp}$$

(2)

and $V_{ramp} = V_{cmp}$. Equating 1 and 2 yields an expression for the scaling factor, $K$, which is defined as the ratio of the
differential integrators (b) RLC low pass filter.

Figure 2. Circuit detail of the charge dump/remove stage.

output pulse width to input pulse width.

Thus scaling is a function of the ratio of two capacitances multiplied by the ratio of two currents. The ratios of capacitance in equation 3 is fixed at chip design time and can be accurately controlled [13] [14] [15]. The currents in equation 3 can be electrically modified and their ratio can be accurately controlled [16] [17], therefore the scale factor \( K \) is 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, \( \text{out} \), 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 Palms filter tap are simple and commonplace. 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 \( \mu A \)) and would require careful consideration of switching noise. In conventional 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 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 2. When \( E_+ \) is high, transistor M1 is on, while M2 is off. This enables the voltage established on the gate of M3 by current \( I_0 \) (site to be transferred to the gate of transistor M4), thus discharging the capacitor with a constant current \( I_0 \). When \( E_- \) is low, transistor M1 is off, while M2 is on. The voltage on the gate of M4 is now \( V_{an} \), switching the current source off. The same principles apply to the top half of the circuit. In simulation an input current of 5nA and standard transistor inverters were used yet no switching noise was discernible at the output.

Figure 3. (a) Filter implemented using differential integrators (b) RLC low pass 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 Palms filter implementation are very close to the theoretical \( z \) domain response as shown in Figure 4 demonstrating the validity of the Palms 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 order RLC Butterworth low pass filter with a cut-off frequency of 1kHz (Figure 3a) was approximated using the Backward Difference transformation between the \( s \) and \( z \) domains, where

\[
\begin{align*}
\tau &= \frac{1}{2}\pi f_0 \\
\tau &= \frac{T}{2}
\end{align*}
\]

and where \( T \) is the sampling interval. The resultant scaling factors for \( T = 100\text{ns} \) are \( K_1 = K_2 = 0.021 \), \( K_3 = K_4 = 0.36 \). Using these scaling factors the filter structure of Figure 3a was implemented using the Palms Filter circuit of Figure 1. The appropriate digital logic to generate the signals \( E_+ \) and \( E_- \) for the Backward Difference transformation are given by the following equations

\[
\begin{align*}
E_+ &= S \cdot P \cdot \bar{M} + \bar{S} \cdot \bar{P} \cdot M \\
E_- &= S \cdot \bar{P} \cdot M + \bar{S} \cdot P \cdot \bar{M}
\end{align*}
\]
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 Palio Filter Implementation were calculated. These are shown in Figure 4. The results from the HSPICE simulation of the Palio Filter Implementation are very close to the theoretical $z$ domain response.

6. PRELIMINARY RESULTS FROM VLSI

At the time of writing, our first Palio chip is undergoing 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, $V_{CH}$ in Figure 1 was driven via an analogue pad from an external voltage source. The $V_{DD}$ voltage for the ramp waveform (Figure 2) was set to a zero voltage reference of 2.7V. The resultant output pulse (out in Figure 1) was sampled using a digital storage oscilloscope, the pulse width measurement giving the magnitude of the output pulse, while the sign of the measurement was defined by the state of the sign clock. Measurements were taken for two $L_i$ settings indicating that the scaling factor, $K$, can be varied. These results are shown in Figure 5.

6.2. Palio integrator

The operation of the Palio integrator is illustrated by the oscilloscope traces of Figure 6. The top trace is the output 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 is zero. As a precursor to implementing a Palio Filter using 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 $\xi_1$ and $\xi_2$ for the Bilinear Transformation are given by the following equations:

$$\xi_1 = P \cdot S \cdot B + M \cdot S \cdot B \quad \xi_2 = P \cdot S \cdot B + M \cdot S \cdot B$$

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 $\xi_1$ signal, containing every second positive pulse from the integrator input (second trace). For brevity, the $\xi_2$ has not been shown. The $\xi_1$ and $\xi_2$ 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 reconstructed from the signed PWM output of the integrator.

6.3. Evaluation of preliminary results

These results indicate that all the individual components of the first Palio filter chip 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 filters that is ideally suited to VLSI. The Palio 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 using a simple ratio of currents. Techniques such as the dynamic current mirror may be used to give accurate ratios [19]. 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.

Apart from the obvious advantages of using noise tolerant pulses for signal representation, in principle, this approach offers advantages compared to conventional switched capacitor and switched current implementations. By switching...
current sources as described above, switching noise is minimised resulting in smaller operating currents and thus in reduced power consumption. This approach is potentially faster than switched capacitor techniques for a given technology - since the sampling frequency is equal to the switching frequency, while in switched capacitor implementations a higher switching frequency is required in order to implement the resides.

Finally, the concept may be implemented in analogue or digital VLSI using very simple circuit elements that are easy to design, and particularly easy to test since input and output signals are digital.

* The name Pulse is derived from the hellenic word ΠΑΛΜΟΣ which means pulse, pulse palpitation or series of pulses.

REFERENCES


PALMO SIGNAL PROCESSING: VLSI RESULTS FROM AN INTEGRATED FILTER
K. Papathanasiou and A. Hamilton
Department of Electrical Engineering, University of Edinburgh, Scotland, E.U.
Kostas.Papathanasiou@ee.ed.ac.uk
A.Hamilton@ee.ed.ac.uk

ABSTRACT
In this paper a new signal processing technique is presented. This technique exploits the use of pulses as the signalling mechanism. This Palmo signalling method applied to signal processing is novel, combining the advantages of both digital and analogue techniques. To demonstrate the inherent suitability of the technique to programmable analogue implementations, a Palmo Miller integrator was implemented. The circuits, distortion and noise analysis, as well as results from a VLSI device are presented in this paper.

1. INTRODUCTION
A new electronics sector is about to emerge in the area of programmable analogue VLSI for rapid prototyping and manufacturing of systems [1, 2]. With such a reconfigurable chip several markets/customers can be targeted simultaneously with obvious benefits in terms of both volume of sales and reduced time to market. These implementations use standard Switched-Capacitor (S-C) [3] or Switched-Current (S-I) [4, 5, 6, 7] techniques for their analogue cells. Yet we believe that for such implementations new techniques should be applied in order to overcome some of the limitations posed by S-C or S-I.

This paper presents a new approach to VLSI system implementation combining a pulse-based signal representation with simple analogue processing. Palmo signal processing exploits digital pulses as signal using, for example, modulation of the width of the signal, to represent analogue quantities. Communication of signals between processing blocks is therefore by robust digital pulse width modulated signals, while processing within blocks is performed using compact analogue circuit techniques. As it will be demonstrated in the rest of this paper the combination of pulse based signalling and analogue processing can result in simple, programmable circuits for signal processing that are ideally suited to Field Programmable Analogue Arrays (FPAA's).

Our interest in Palmo signal processing has arisen from a requirement to pre-process data for our pulse based neural network chips [8], in order to extend their area of application. A Palmo filter test chip [9] has been fabricated, using CMOS technology, results and analysis from this device are presented here.

1The name Palmo is derived from the Hellenic word FLAMO which means pulseless, pulse substitution or series of pulses.

![Figure 1. Palmo filter integrator and typical waveform diagram](image-url)

Figure 1. Palmo filter integrator and typical waveform diagram

2. FILTER BLOCKS
Fundamental to the implementation of active RC filter structures is the Miller integrator. This elementary filter tap has two inputs, plus and minus. The signal at the minus node is subtracted from the input at the plus node and the result is integrated in time. The output from the integrator is scaled by a factor K. It is this functional building block that we have implemented using pulse based signal processing techniques.

2.1. Signal representation
Our novel approach uses pulses to represent the input signals [10, 11]. In particular we are investigating the use of Pulse Width Modulation. 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. Therefore a positive signal of a value '+A' is represented by a pulse which is 'high' for ΔT₁₄ during the positive cycle of the sign clock, and a negative input value of '-A' by a pulse which is 'high' for ΔT₂₄ during the negative cycle of the sign clock. This representation has the advantages that without including 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.

2.2. Palmo Integrator
Consider the circuit of Figure 1. The input pulses are directed from the plus and minus nodes to either the 'L' or the 'C' switch depending on the sign bit. If 'L' is closed then charge is dumped onto the capacitor C₄ for the duration of the pulse ΔT. If the 'C' switch is closed, charge is removed from C₄.

The charge accumulated on the integrating capacitor V₁₄ is then compared with a ramp (with trace in figure 2 = V₁₄). The combination of the comparator, XOR gate and the global sign clock
ensures the regeneration of the signed pulse representation described earlier.

2.3. Palms 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 Palms analogue cell (figure 2) is defined by the digital logic block which drives the $E_1$ and $E_0$ switches. The appropriate digital logic to generate the signals $E_1$ and $E_0$ in order to implement a Miller integrator are given by the following equations:

$$E_1 = P \cdot S \cdot \overline{B} + M \cdot \overline{S} \cdot B$$

$$E_0 = P \cdot S \cdot\overline{B} + M \cdot S \cdot B$$

where $P$ is the sign clock, $S$ 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.

$$\Delta Q_{con}(2) = \int_{t_1}^{t_2} (\Delta T_{rch} + \Delta T_{min}) \, dt$$

(1)

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

$$V_{con}(t) = V_{con}(t) + \frac{\Delta Q_{con}(t)}{C_{int}}$$

$$V_{con}(t) = \frac{C_{int}}{C_{int}} \Delta T_{rch} + \Delta T_{min}$$

(2)

The voltage ($V_{con}$) accumulated on the integrating capacitor (figure 2) is compared with the ramp ($V_{ramp}$) in order to regenerate the pulse output. When the voltage $V_{con}$ 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

$$V_{ramp}(t) = \frac{I}{C_{int}}$$

and $V_{con} = \frac{C_{int}}{C_{int}} I R$, thus:

$$\Delta T_{rch} = \frac{C_{int}}{I R} \Delta T_{rch}$$

$$\Delta T_{min} = \frac{C_{int}}{I R} \Delta T_{min}$$

(3)

### 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 $\Phi$ is:

$$V_{out}(s) = \frac{C_s}{C_{int}} \left( \frac{V_{in}(s) - V_{in}(s)}{1 - e^{-s}} \right)$$

(4)

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

$$K = \frac{C_s}{C_{int}} \frac{C_{int}}{C_{int}} \frac{I_{ramp}}{I}$$

Because of this similarity existing S-C synthesis techniques and tools can be applied to the Palms realization. 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. Since the ratio of capacitances in equation (4), 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, it is realized that the scale factor $K$ is fully programmable and insensitive to absolute values.

### 3. NOISE

The main source of noise in the Palms circuit is switching noise from the current dump-remove circuit. Yet as explained in [12, 0] 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 ($M_p$ and $M_n$) are turned on and off, switching the actual current source on and off. This virtually eliminates charge injection. In simulation all input current of $5nA$ and standard transistor parameters were used yet no switching noise was discernable at the output. Test-chip results verify these simulations.
3.1. Harmonic Distortion

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 biasing current \( i_{bi} \) flows in one side of the stage or the other (if the differential stage is sufficiently unbalanced).

Therefore it can be assumed that the parasitic load capacitor of the inverting stage (\( C_{0} \) 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 (figure 4A) there is a significant difference between the positive and negative going delay times. Assuming that the biasing currents are constant the positive delay time \( T_{+} \) is given by the following equation:

\[
T_{+} = C_{L} \frac{V_{TRP} - V_{SS}}{I_{+}}
\]

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

\[
T_{-} = C_{L} \frac{V_{00} - V_{np}}{I_{-}}
\]

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 \( AT_{+} \), while the same delay is subtracted from the negative ones, because of our signed pulse representation. This results in the signal of figure 4D.

The Fourier cosine series of the output signal (figure 4D) is equal to the sum of the Fourier series of the two signals shown in figures 4B and 4C thus:

\[
\sum_{n=0}^{\infty} a_{n} \cos(\omega_{n} t) + a_{0} \cos(2\omega_{n} t) + a_{2} \cos(2\omega_{n} t) + \ldots
\]

Where

\[
a_{n} = \frac{1}{T} \int_{0}^{T} f(t) \cos(\omega_{n} t) \, dt
\]

\[
a_{\omega} = \frac{1}{2} \int_{0}^{T} f(t) \cos(\omega_{n} t) \, dt = (-1)^{n-1} \frac{a_{n}}{\omega_{n}}
\]

The Total Harmonic Distortion (THD) due to the propagation delay of the comparator is derived from, \( a_{n} \) in (5), for \( n = 3, 4, \ldots \) therefore:

\[
THD_{D} = \sqrt{\left(\frac{T_{+}}{T_{-}}\right)^{2} + \left(\frac{T_{-}}{T_{+}}\right)^{2} + \left(\frac{T_{+}}{T_{-}}\right)^{2} + \left(\frac{T_{-}}{T_{+}}\right)^{2}}
\]

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 our test chip, the capacitors are fixed while the current sources are driven externally, therefore the output of the integrator is dependent upon \( I_{m} \), since \( i_{bi} \) is constant. The results displayed in figure 5 were taken by applying a number of constant pulses to the integrator and measuring the output. The output pulses (out in figure 3) 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

In figure 6 the response of a first order Butterworth filter with a cut-off frequency of 800Hz and a sampling frequency of 8020Hz 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.

Though our test chip was designed to work at a much higher sampling frequency, our present testing equipment limits the measurable frequency. We plan to investigate higher frequencies and bigger filter arrays.
4.3. Measured THD

A cosine input of 86011z 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 = 086% was calculated.

If the sampling frequency increases, the integrating and ramp currents will be forced to increase as well, resulting in smaller matching errors and thus decreasing THD. However THD  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 = THD1 + THD5; 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. Signaling between filter stages is performed using digital pulse width signals which are robust, noise-tolerant, and easily distributed between 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 to implement a range of signal processing functions by changing the digital logic block in the circuit of figure 2. It is possible to realize the same building block as a scalar, multiplier, adaptive memory block or as a non-linear building block. It is obvious that because of the use of pulses as the signaling 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


NOVEL PALMO\(^1\) 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 (FPAA)\(^1\) is about to offer similar benefits for rapid analogue circuit prototyping. Since analogue circuits are difficult to design, layout and test, FPAA are likely to emerge as a significant new market sector.

The circuits proposed for these new FPAA devices use conventional - predominantly switched-capacitor (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 switched-current 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.

\(^1\)The name Palmo is derived from the Hellenic word ΠΑΛΜΟΣ which means pulsebeat, pulse palpitation or series of pulses.
3 Signal Representation

The focal point of the proposed Palmo Signal Processing is the pulsed representation of the analogue input signals. Pulses are easily regenerated, stored in short term analogue memory, and modified. Since pulses are digital signals they are robust, noise-free and easily distributed among and within chips.

Our novel approach uses pulses to represent the input signals [4, 5]. In particular we are investigating the use of Pulse Width Modulation, since it gives unique advantages in both noise and frequency of operation. 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. Therefore a positive signal of a value \( A \) is represented by a pulse which is 'high' for during the positive cycle of the sign clock, and a negative input value of \( -A \) by a pulse which is 'high' during the negative cycle of the sign clock. 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.

4 Palmo Building Blocks

Fundamental to the implementation of signal processing circuits are the adder-integrator short-term memory and the multiplier-scaler. It is these building blocks that we have implemented using analogue VLSI techniques.

4.1 Adder-Integrator-Memory

A capacitor is an elementary memory cell. The charge accumulated on the capacitor (figure 1) can be controlled by the switches \( \xi_+ \) and \( \xi_- \). A positive signed input pulse is diverted by the digital logic and closes the switch \( \xi_+ \), for the duration of the pulse \( \Delta T_{\text{in}} \), this dumps some charge on the capacitor given by the following equation:

\[
\Delta Q_{\text{Cint}} = I_{\text{int}} \cdot \Delta T_{\text{in}}
\]

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.
4.2 Multiplier-Scaler

In order to convert the voltage on the integrating capacitor into a pulse, we compare the voltage $V_{int}$ 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.

$$\Delta T_{out} = \frac{C_{int}}{C_{int} I_r} \cdot \Delta T_{in}$$

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 $\xi_+$ and $\xi_-$ in order to implement a Miller integrator (Figure 2) is defined in the following equations:

$$\xi_+ = P \cdot S \cdot B + M \cdot S \cdot B$$
$$\xi_- = P \cdot S \cdot B + M \cdot S \cdot B$$

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).

$$\Delta V_{out} = \frac{C_{int} I_r}{C_{int} I_r} \cdot \frac{\Delta T_{plus} z^{-1} - \Delta T_{minus} z^{-1/2}}{1 - z^{-1}} \tag{1}$$

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.
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 (HD1) due to the propagation delay of the comparator is given by equation 2.

$$H D_1 = \sqrt{\left(\frac{P}{2}\right)^2 + \left(\frac{I}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + ...}$$

Where $P$, $I$, and $\pi$ is the magnitude of a signed PWM cosine input and $v$ is the size of the minimum pulse ($\Delta T_{min}$) 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, $H D_2$, given by equation 3.

$$H D_2 = \frac{4}{\pi} \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + ...}$$

Where $P_c$, $I_c$ are the charging and discharging current sources respectively as shown in figure 3.

Figure 4 shows plots of $H D_1$, $H D_2$ and Total Harmonic Distortion (THD = $H D_1 + H D_2$) 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 8020Hz, 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
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 our test chip, the capacitors are fixed while the current sources are driven externally, therefore the output of the integrator is dependent upon $I_{\text{ref}}$, since $I_c$ 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 Polno 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 Polno building blocks to reduce THD and improve overall programmability and performance. Such circuits are ideal for analogue FPAA cells.

The new Polno 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


Advances in Programmable Pulse Based Mixed-Signal Processing VLSI

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

Abstract—This paper describes a pulse based signal processing technique for VLSI implementation of fully programmable analogue arrays. A pulse width modulation signal representation and a basic Palmo\(^1\) analogue cell are introduced. The equivalence of the Palmo cell to a switched capacitor building block is demonstrated and the sources of harmonic distortion are discussed. VLSI results showing 1st, 2nd, 3rd order IIR filters and a 24th order FIR filter are presented. Improvements for a fully programmable analogue array currently under testing are discussed.

I. INTRODUCTION

Recent advances in the implementation of Field Programmable Analogue Arrays (FPAA’s) have generated considerable interest in circuits for programmable analogue systems [1]. This paper presents a novel strategy for the implementation of FPAA’s using a pulse based signal processing technique, ideally suited to VLSI, that overcomes some of the restrictions of conventional strategies.

Continuous time circuits have been proposed that allow an analogue design to be broken down into a subset of smaller instructions from an analogue instruction set [2]. Log and anti-log circuits are used so that multiplication, division and raising to a power may be implemented by simple arithmetic circuits. Some functions require additional components external to the chip [3].

A classic parasitic-insensitive switched-capacitor integrator 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 circuits have also been proposed [1].

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 analogue and digital sections of the FPMA requires special

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 pulse 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

---

\(^{1}\)Palmo is derived from the Hellenic word ΠΛΑΜΟ 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. PWM-to-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 down-counting 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 vary 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 that 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 $\xi$ and $\zeta$ switches of the integrator (Figure 1B) [8]. Figure 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.

\[ H_D_1 = \sqrt{\left(\frac{1}{\pi}\right)^2 + \left(\frac{1}{\pi}\right)^2 + \left(\frac{1}{\pi}\right)^2 + \ldots} \]

where $I_{mp}$ is the charging current into the integrator from the input and $I_n$ is the charging current into the integrator from 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, $H_D_2$, given by equation 3.

\[ H_D_2 = \frac{4}{\pi} \sqrt{\left(\frac{1}{\pi}\right)^2 + \left(\frac{1}{\pi}\right)^2 + \left(\frac{1}{\pi}\right)^2 + \ldots} \]

where $I_{mp}$, $I_n$ are the charging and discharging current sources respectively as shown in Figure 1B.
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 [10]-[15]. In that way the $H_2$ 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 $V_{bias}$ 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 writing the new device has been 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 been characterised. On the other hand the operation of the analogue circuits have not yet being been 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

![Diagram of clamped comparator with positive feedback]
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 from 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 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 FIR 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


TO APPEAR IN IEE ELECTRONICS LETTERS, p. 1

A PALMO\(^1\) CELL USING SAMPLED DATA LOG-DOMAIN INTEGRATORS.

T. Brandtner, K. Papathanasiou, A. Hamilton

Indexing terms: Analogue Signal Processing, FPAA, FPA, Palmo, Pulses, Log-domain, current mode, VLSI.

This paper presents the first log domain integrator for programmable analogue sampled data signal processing. This circuit is specific to the implementation of emergent programmable pulse based signal processing systems yielding greater dynamic range, reduced power supply voltage and increased operating frequency. Simulation results demonstrate the validity of the approach.

Introduction: Programmable pulse based signal processing 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 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.

Log-domain integrators [2] integrate logarithmically compressed input currents by making use of the transistor characteristics of bipolar (or subthreshold MOS) transistors. The integrated voltage is subsequently expanded exponentially to generate the output current. This companding technique offers a large output dynamic range for a small voltage swing. The use of bipolar transistors results in higher operating frequencies and improved accuracy over CMOS integrators.

The Palmo signal representation results in an efficient implementation of a log-domain sampled data integrator. Since the input to the integrator is digital 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 generating 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 50dB and sampling rates in excess of 5MHz compared to 54dB and 1MHz of a commercially available device [3].

The Name Palmo is derived from the Hellenic word ΠΑΛΜΟΣ which stands for PulseBeat, Pulse Pulsitation 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).

A block diagram of the Palmo cell is shown in Figure 1. It consists of three parts: digital logic, 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. The output is generated 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 comparators are well matched. This results in each sample being represented by two pulses (Figure 1).

The gain of the Palmo cell is controlled by the ratio of the integrating constants of the two integrators generating $I_{\text{ramp}}$ and $I_{\text{int}}$.

The Log Domain Integrator: The circuit of the log-domain integrator used in our Palmo cell is shown in Figure 2. It is basically a circuit that integrates 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 $I_a$ and $I_b$. First the input currents are compressed into log-domain by Q1 (Q8). Q2 and Q1 (Q7, Q2) perform integration in the log-domain. The integrated value is scaled by Q3 (Q6), the scaling factor depends on the current $I_i$. Finally Q4 (Q5) expands the compressed signal. The output is represented by the difference of the currents $I_{o1}$ and $I_{o2}$.

From the translinear equations for the integrator circuit of Figure 2 the following formula may be derived:

$$
\frac{d}{dt}(I_{o1} - I_{o2}) = \frac{I_i}{C \cdot V_{th}} \left(1 + \frac{1}{\beta}\right) (I_b - I_a) + \frac{1}{C \cdot V_{th}} \left(\frac{I_b}{\beta} - I_a\right) (I_{o1} - I_{o2}) - \frac{1}{C \cdot V_{th}} \left(\frac{I_i}{\beta} + \frac{I_i}{\beta} \right) \left[\left(\frac{k_4}{k_3} - 1\right) \left(\frac{k_4}{k_3} - 1\right)\right]
$$

(1)

where $a$ and $b$ are the mirroring ratios of the current mirrors M5-9 and M10-12 respectively, $\beta$ is the current gain of the bipolar transistor, $V_{th} \approx 25mV$ at 300°K, $k_4 = 1 + V_{BE}/V_{AR}$, $V_{BE}$ is the collector–base voltage of transistor Q2 and $V_{AR}$ is the early voltage.

The first term on the right side of the equation shows that the difference of the output currents $I_{o1}$ and $I_{o2}$...
Publications

output current changes. The transistors M13 and M14 on the integrating nodes. This effect is cancelled by controlling the current $I_2$ seen in Figure 4. The cutoff frequency is set here.

Figure 1. The frequency response of this simple filter can be implemented by adding the feedback signal shown in Figure 1. The circuit is mainly responsible for the high sampling frequency of our Palmo cell because it is faster than the voltage mode comparators used in former Palmo implementations.

Current Controlled Comparator (CCC): This circuit is mainly responsible for the high sampling frequency capability of our Palmo cell because it is faster than the voltage mode comparators used in former Palmo implementations.

The CCC used was based upon [6] with a slight modification to accommodate the lack of p-wells in the fabrication process used. This circuit allows the detection of small changes in input currents ($50\mu A$ to $500\mu A$) with a short propagation delay (17ns). Normal current comparators require several hundred nanoseconds to detect the same current change.

Simulation Results: The linearity of the integrator is shown in Figure 3. It demonstrates good linearity with an output range of $\pm 15\mu A$. In the same figure is shown that the integrator gain may be set by controlling $I_1$. It is possible to change $I_1$ between $50\mu A$ and $250\mu A$. 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 0.5%.

A first order sampled data low pass filter was implemented by adding the feedback signal shown in Figure 1. The frequency response of this simple filter can be seen in Figure 4. Here the cutoff frequency is set by controlling the current $I_1$. As in all sampled data systems the sampling frequency may also be used to modify the response of the filter.

Conclusions: The log-domain sampled data integrator 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 electronics, especially Field Programmable Analogue and Mixed-Signal Arrays. The advantages of the log domain circuit to pulse based signal processing are lower power supply voltages and greater dynamic range. The use of a current mode technique gives higher sampling frequencies due to the speed of the current comparator used in the Palmo cell.

The circuits reported here are optimised for dynamic 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 operating 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 UK EPSRC grant reference GR/L56031.

References


