DSP Platform Concept

Frank Engel

NICTA (ERTOS)

<frank.engel@nicta.com.au>

SHORT CV

- Study Electrical Engineering, Dresden Univ. of Tech. (1991 – 1997)
- PhD at Vodafone Chair Mobile Comm. Systems, Dresden (1997 – 2003)
- Researcher with NICTA since 10/2003
- Interests:
  - Digital VLSI Design
  - HW/SW Codesing for (reconfigurable) SoC
  - OS support for SoC (Design & Applications)

VODAFONE CHAIR MOBILE COMM. SYSTEMS

- Founded in 1994 by Mannesmann Mobilfunk
- Focus on Wireless Communications Technology
- ~20 PhD Students, ~10 Masters Students
- Three Main Topics:
  - Software Radio

    Platform for SW Based Wireless Modems
  - Wireless Network Simulation

    Capacity/Topology Analysis of Mobile Networks
  - Digital Signal Processors

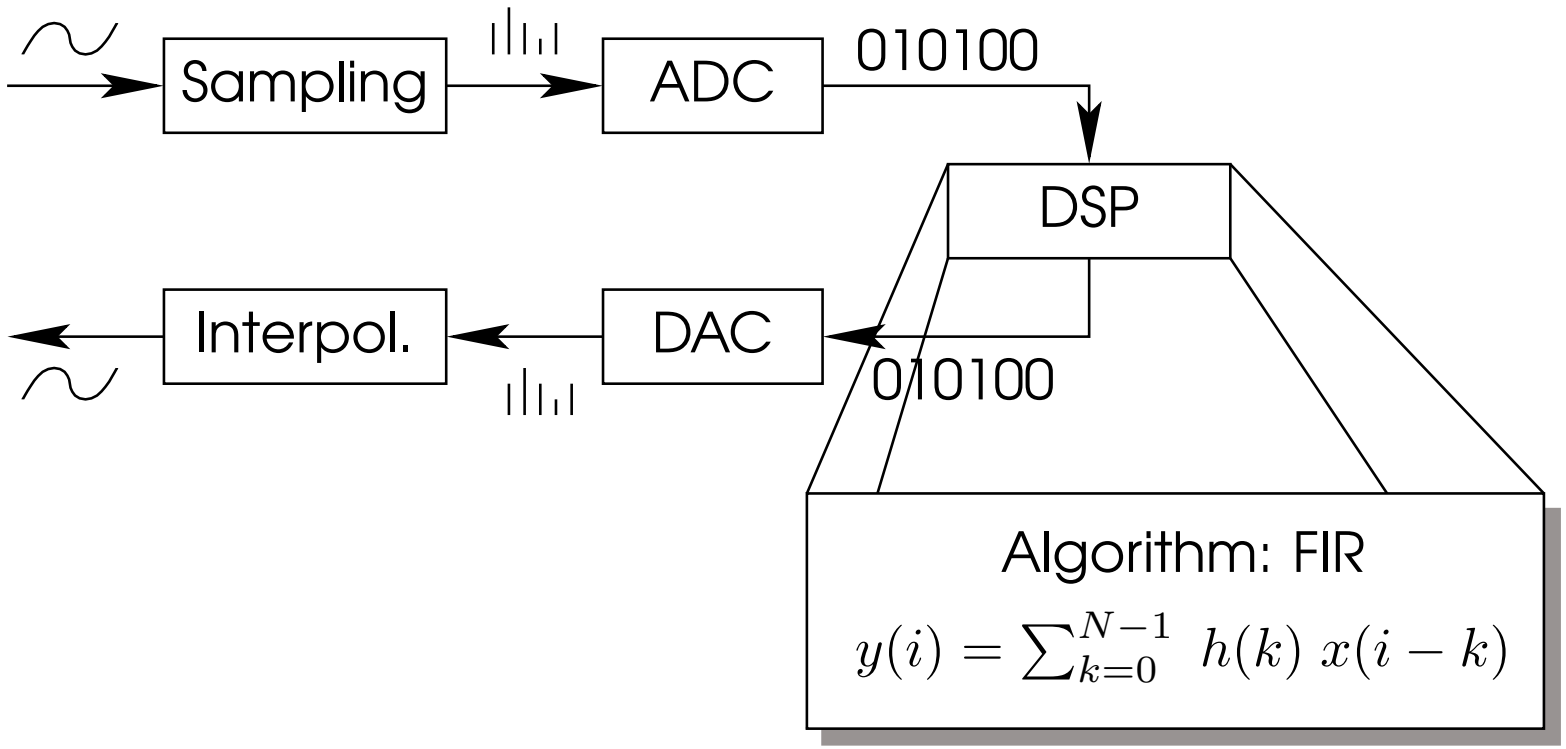
    Tailorable DSP Platform

OUTLINE

- Overview on DSP Processors (Motivation)
- Platform Approach
- Examples And Outlook

Overview on DSP Processors

DIGITAL SIGNAL PROCESSING



➜ *Constant Processing of Digital Data Sets.*

DSP APPLICATIONS

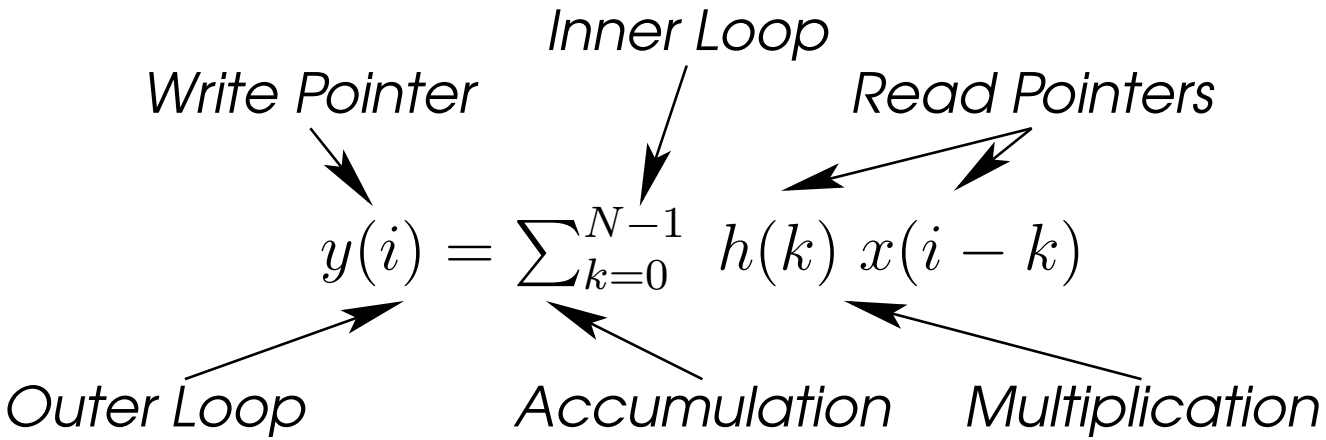
Examples:

- Digital Cameras, Cellular Phones, etc.
- Seismic Analysis
- Music Synthesis, Effects
- Disk Drives
- Modems (POTS, ISDN, Cable,...)
- Motor Control

➜ *No User Programming*

➜ *Live Lasting Applications*

FIR EXAMPLE I



- Repetitive Numeric Computations (ADD, MUL)
- Attention to Numeric Fidelity
- High Memory Bandwidth (Fetch, 2 Read, Write)
- Loop and Address Calculations
- Real Time Processing (i == Time)

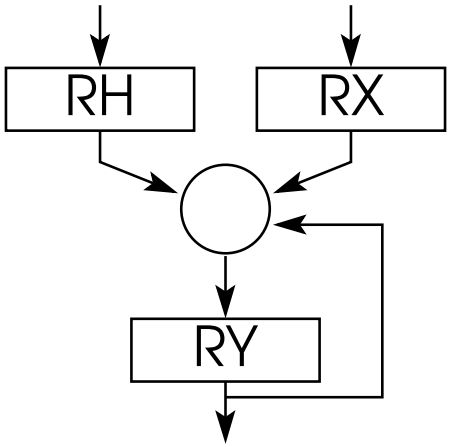
FIR Example II

```
      mov r7, #N-1     ;; init loop ctrl
loop: ld  *r0, r3      ;; h(k)
      ld  *r1, r4      ;; x(i-k)
      mul r3, r4, r5   ;; h(k)* x(i-k)
      add r6, r5, r6   ;; accumulate
      add r0, #1       ;; k+1
      sub r1, #1       ;; i-(k+1)
      sub r7, #1       ;; loop counter
      tst r7           ;; loop ctrl
      jnz loop         ;; loop ctrl
      st  *r2, r6      ;; y(i)
      add r2, #1       ;; i+1
```

FIR EXAMPLE III

- Possible HW Accelerators:
  - Special Computation Unit
    ➜ *MUL, ADD in One Cycle*
  - Harvard Architecture, 3 Port D-Mem
    ➜ *Increase Memory Bandwidth*
  - Address Pointer Generators
    ➜ *Complex Memory Access*
  - Loop Control Unit
    ➜ *No Explicit Counter Test*

FIR Example IV




```
mov rl, #2, #N-1   ;; config loop ctrl*
ldi rh, rx         ;; h(k), x(i-k), inc, inc
mac                ;; mult + acc
sti ry             ;; y(i), inc
```

*(`mov rl, #2, #N-1`: Execute Next 2 Instructions N-1 Times)

➜ *DSP Processor Features Are Algorithm Driven.*

DPS GENERATIONS I

Generation	Features	Examples
First: 1979–1985	Harvard Architecture, HW Multiplier,	Bell Labs DSP1, TI TMS320C10
Second: 1985–1990	X/Y Mem, MAC-Unit HW Instr. Loop, Modulo Addressing	AT&T DSP16 TMS320C25, ADSP-2100

➜ *DSPs as a New Class of Processors Established.*

DPS GENERATIONS II

Generation	Features	Examples
Third: 1990–1995	Multiple Data Paths, Appl.-Specific Units, Low Power, Tool Sets, *DSP-Cores (Licensing)*	DSP Group's Pine & Oak, TCSI Lode, TMS320C50
Fourth: 1995–	VLIW, Caches, Super-scalar, SIMD/MMX Compiler, *Mixing of CPU- & DSP World*	Phillips Trimedia, Infineon Carmel, TigerShark, TMS320C60,

➜ *Now DSPs Are As Complex As CPUs.*

DSP: Trends And Resume

- DSP Architectures Are Strongly Application Driven
- Applications Are Becoming More Complex and Control Oriented
  - DSP with CPU Features and Vice Versa
  - Assembler Programming Becomes Very Difficult

    ➜ Compiler etc.
- Technology Progress Allows Higher Integration
  - Faster and Faster and ..
  - More Memory, Cache, Features, ..
  - System On Chip

➜ *How to Efficiently Design a New DSP?*

PLATFORM APPROACH

DSP DEVELOPMENT GROUP

- Work Presented Here Done Mainly 1995 – 2002
- About Seven PhD Thesis
  - ➜ Talk Presents Some of This Work
  - ➜ See References
- One Prototype DSP Developed (M3-DSP)
- Group Involved in Several Industrial Projects

➜ *Talk Gives an Overview on the Groups Work*

How to Make an Impact?

- ✘ Waiting for Next Technology Generation
- ✘ Targeting General Purpose Solutions
- ✔ Targeting Application Specific Solutions
- ✔ Addressing Low Power, Die Size, IP-Core (Licensing)
- ✔ Reduce Design Costs/Time by Modularity

➜ *Platform Approach:*
Basic Design Plus Specific Extensions

BACK TO BASICS: ALGORITHM ANALYSIS

$$y(i) = \sum_{k=0}^{N-1} h(k)\, x(i-k)$$

$$y(i) = \sum_{k=1}^{M-1} a(k)\, y(i-k) + \sum_{k=0}^{N-1} b(k)\, y(i-k)$$

$$y(i) = \max_{k}(a(k) + k(i-k))$$

- Operations on Data Vectors
- Access and Order of Use of Data is Identical
- Data Manipulation/Calculation Differs

➜ *Orthogonalisation Into Data Transfer and Data Manipulation*

CATEGORISING ALGORITHMS

- Split Into Tree Categories
  - **Data Transfer** – Data Flow Before & After Calculation
  - **Data Manipulation** – How To Calculate
  - **Data Control** – Algorithm Sequencing
- For Each Category
  - Detect Similarities
  - Define Algorithm Classes
- Define Overall Minimal Requirements
  - Minimal DSP Configuration

EXAMPLES OF CLASSES

- **Data Transfer** (Consider Parallel Execution)
  - Sliding Window: Propagate Data ($i \rightarrow i+1$)
  - Shuffle Data: Exchange Data ($i \leftrightarrow i+1$)
  - Vector Data: Parallel Data ($i||i+1$)

- **Data Manipulation**
  - Multiply Accumulate: MAC
  - Add Compare Select: ACS
  - Split Word Arithmetic: MMX

ORTHOGONALISATION

Split DSP Design into Several Independent Sections:

- **Data Transfer: Memory Design**
  Exploit Parallelism, Specific Data Arrangement
- **Data Manipulation: Data Path Design**
  Algorithm Specific Arithmetic
- **Data Control: Program Control Design**
  Basic Instructions plus Specific Instructions

➜ *Basic DSP Architecture with Variable Number of (Specialised) Data Paths and Scalable Memory*

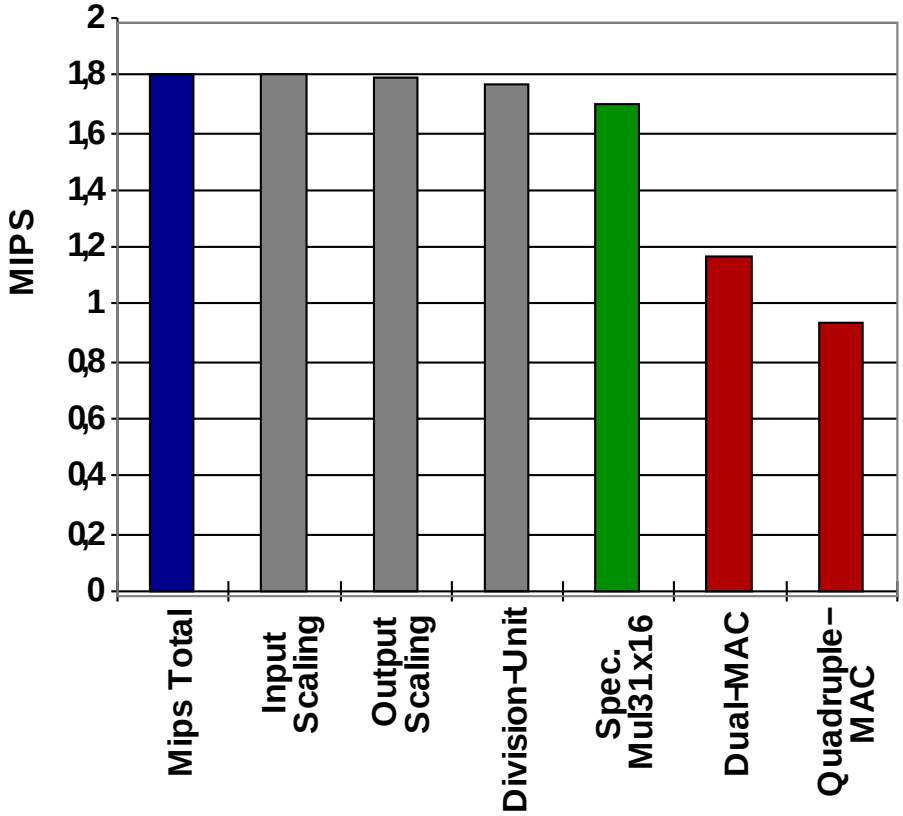
CASE STUDY: GSM FULL RATE VOCODER I

Task: Implement GSM Vocoder (Mobile Phone)

- Typical Signal Processing Application
  - Filters (FIR, IIR)
  - Auto & Cross Correlation
- Accepted Benchmark

➜ *Apply Orthogonalisation*

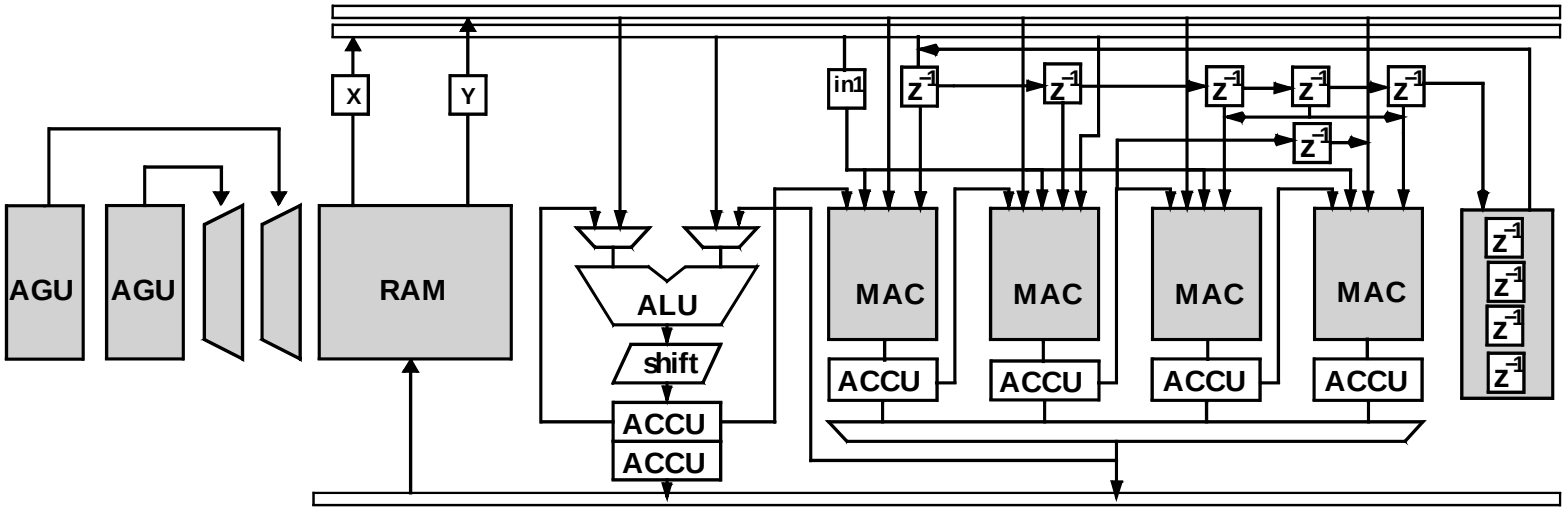
CASE STUDY: GSM FULL RATE VOCODER II



- Data Transfer
  - X/Y Memory
  - Sliding Window
  - Shuffle Data
- Data Manipulation
  - MAC

➜ *Multiple MAC-Operations: Half MIPS Count, Half Clock Rate, Low Power*

CASE STUDY: GSM FULL RATE VOCODER III



Base Configuration

- Dual Read Bus
- X/Y Memory
- ALU, SHIFT

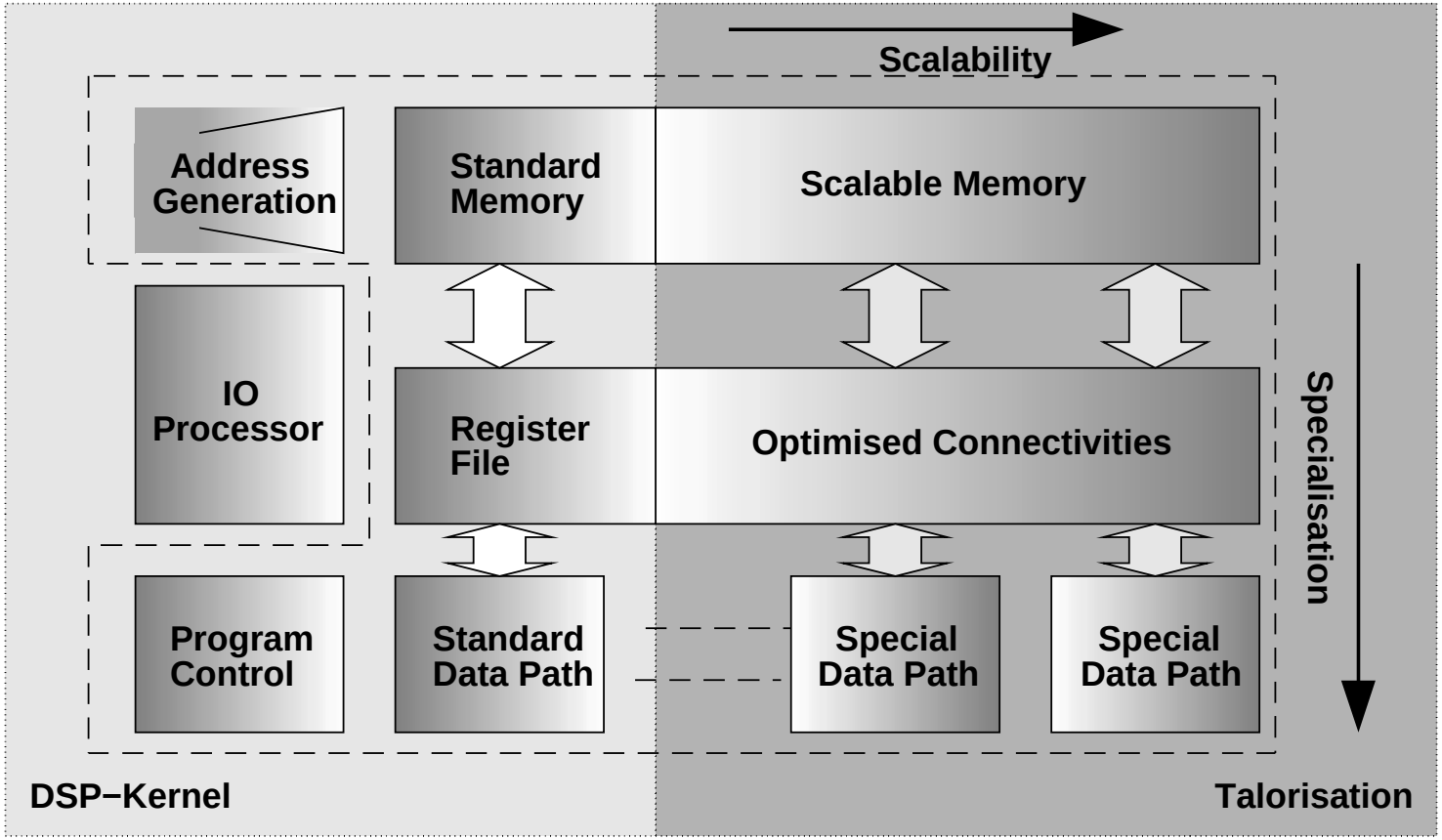
Extension

- Multiple MAC Data Paths
- Local Value Propagation Unit (z^1)

➜ *Find Trade Off Between MIPS- and Gate-Count*

CATS Platform DSP

Concept of Application Tailored Signalprocessors



But: What About Programming and Instruction Set?

VLIW: Orthogonal Programming

Each Functional Unit is Coded Separately

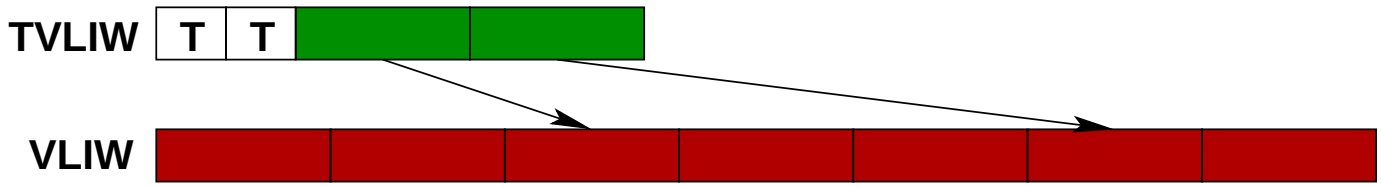
- ✔ Fits Into Orthogonalisation Scheme
- ✔ Scalable Number of Function Fields
- ✔ Individual Coding Within Function Fields
- ✘ Large Instructions
- ✘ Redundant Coding (Many NOPs)

➜ *Fits Great Into CATS, But*
Need a Way to Reduce Memory Footprint!

GENERIC ISA APPROACH: TVLIW

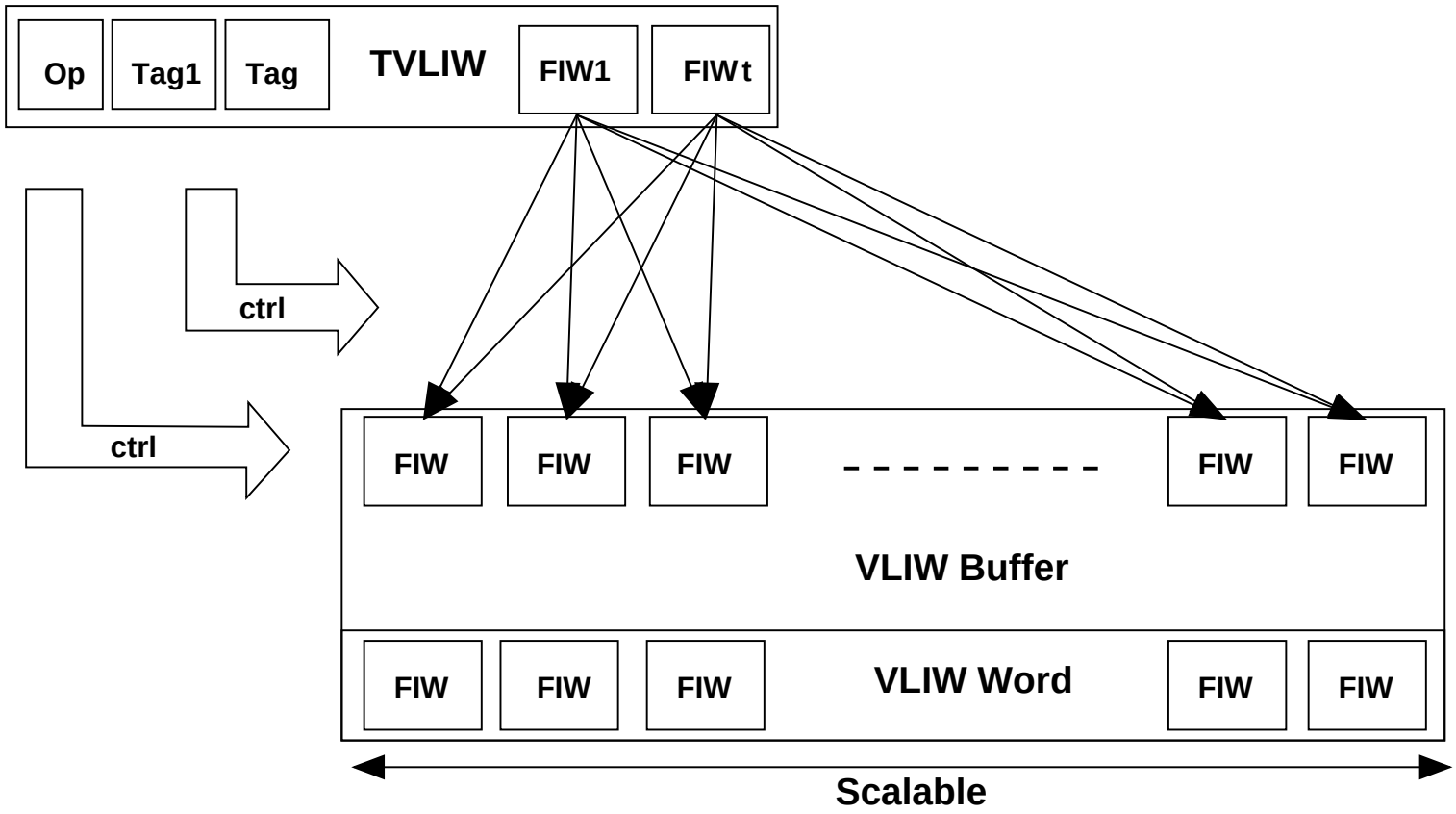
Tagged VLIW: Short VLIW With Tags

- Tag Ctrl: Manipulate VLIW Instructions
  Short VLIWs Are Building Blocks
- Differential Assembling: Avoid Redundancy
  Recycle Function Fields from Previous Instructions
- Exploit Loops: Repetition of Sequences
  Cache Assembled VLIW Instructions



➜ *Again: Exploit Algorithm Characteristics*

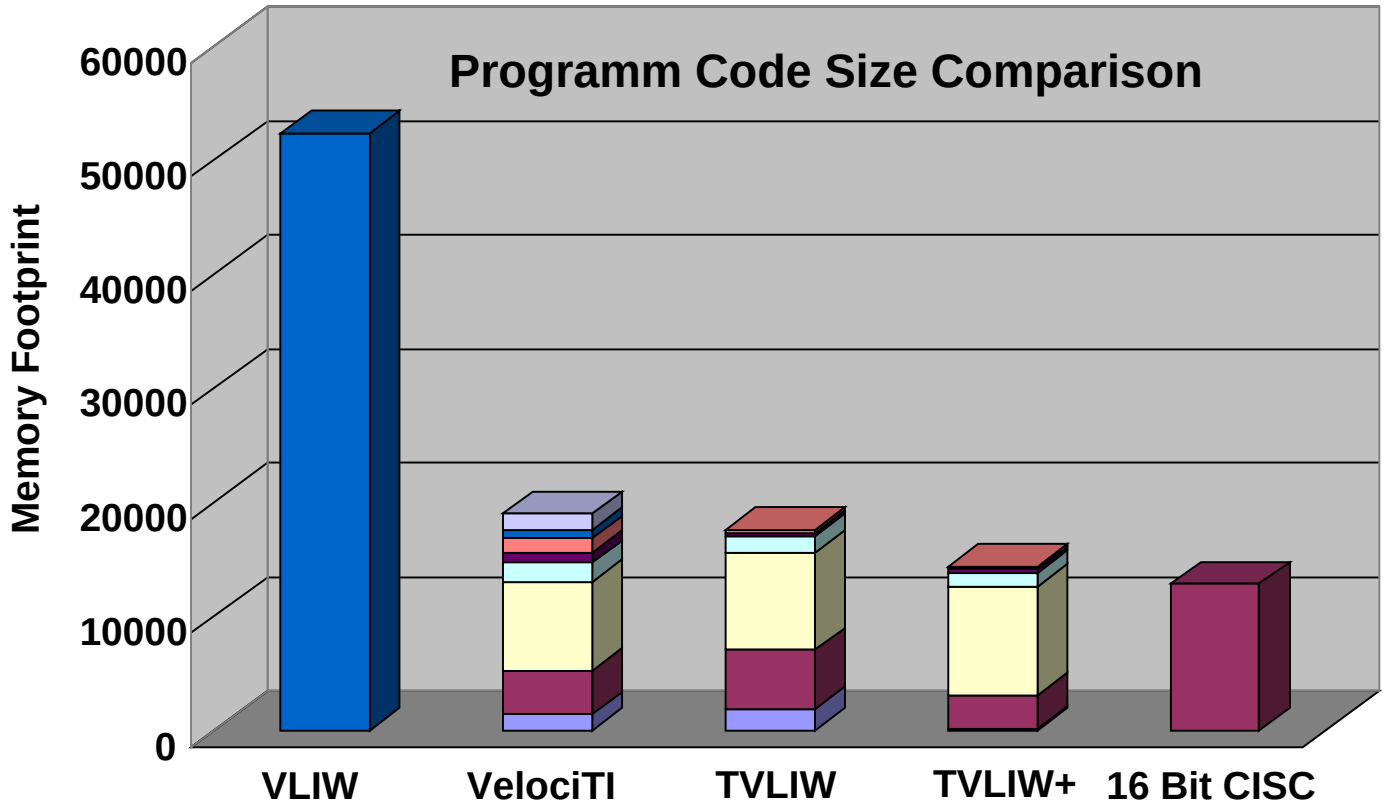
TVLIW Buffer



➜ *TVLIW: Constant User Interface*

VLIW: Scales With Number of Units

CASE STUDY: GSM FULL RATE VOCODER III



➜ *Code Compaction of 75%!*

➜ *Memory Footprint Similar to CISC!*

SUMMARY ON CATS

- Algorithm Orthogonalisation Allows Classification
- Come Up With Design for Each Class
  ➜ Modular Design
- Scalable ISA, But Fixed Programming Interface
  ➜ Unified Programming Tools
- Small Collection of Base Architectures
  ➜ Easy Generation of Deviates
- Reuse Increases Reliability

➜ *System of Modules for IP Core Design*

EXAMPLES AND OUTLOOK

CARMEL DSP (2000)

- Infineon's (Attempt of a) Next Generation DSP Core
- Target Applications: 3G Wireless, xDSL, ..
- Involved in Architecture Design (ISA)
- 16-bit Fixed-Point
- Dual Data Path (MAC, ALU Each)
- Uses CLIW: Configurable LIW
  - *Normal*: Single/Dual Instruction Issue
  - *CLIW*: Buffered Six Instruction Issue
    Allowing Complex Instruction Sequences

➜ *Beaten by Infineon's TriCore CPU :(*

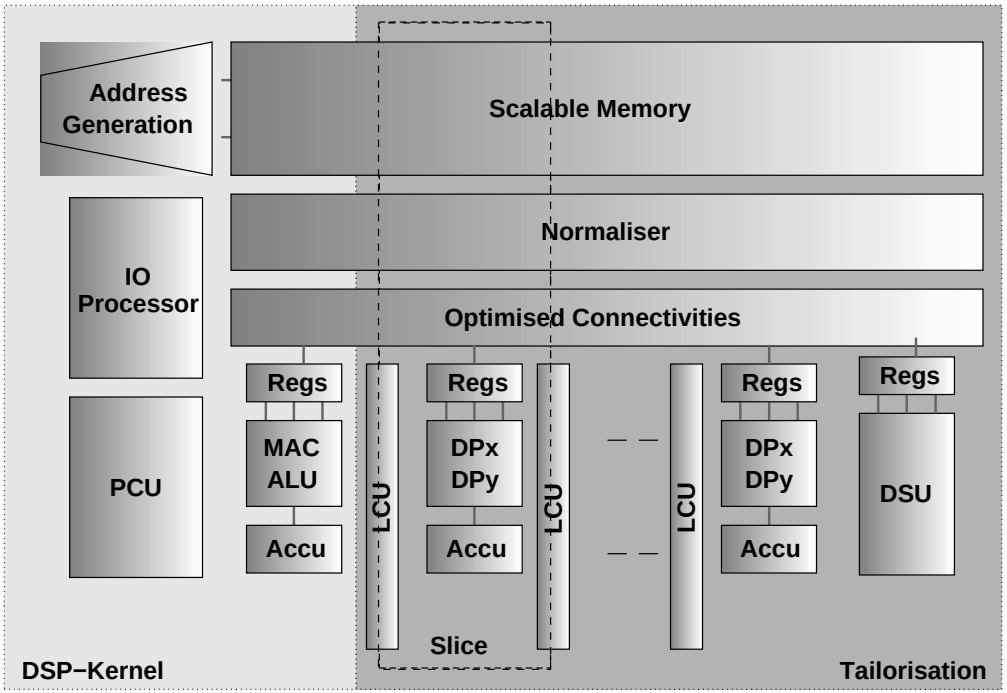
M3-DSP I

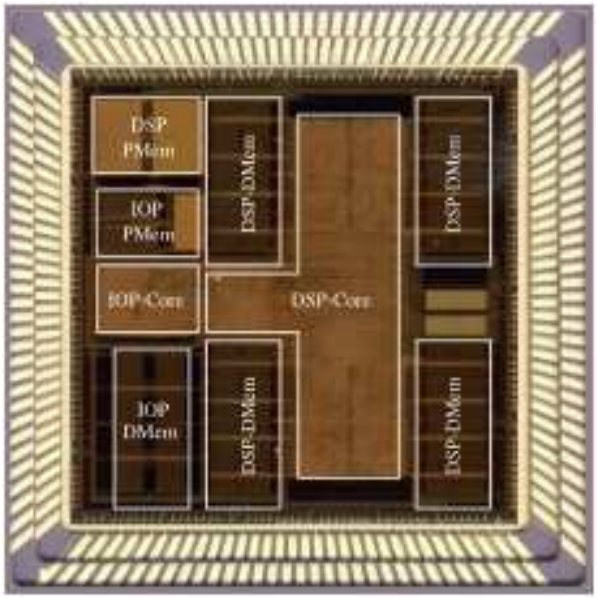
- Prototype as Proof of Concept
- Target Applications: ZATM (WIFI 11b), ADSL
- TVLIW Instruction Set
- Slice Architecture
  - 16x Data Word Wide Scalable Memory
  - 16 Parallel Special Data Paths – SIMD Approach
    - Vector Arithmetic
    - Highly Parallel (C-1024-FFT: 55µs, 52.9%)*
    - Galois Multiplier (BCH-Dec.: 6.1µs, 24.5%)*

➜ *1st Base Architecture*

*Compared to TI C62x @200MHz (FFT: 104µs, BCH: 25µs).

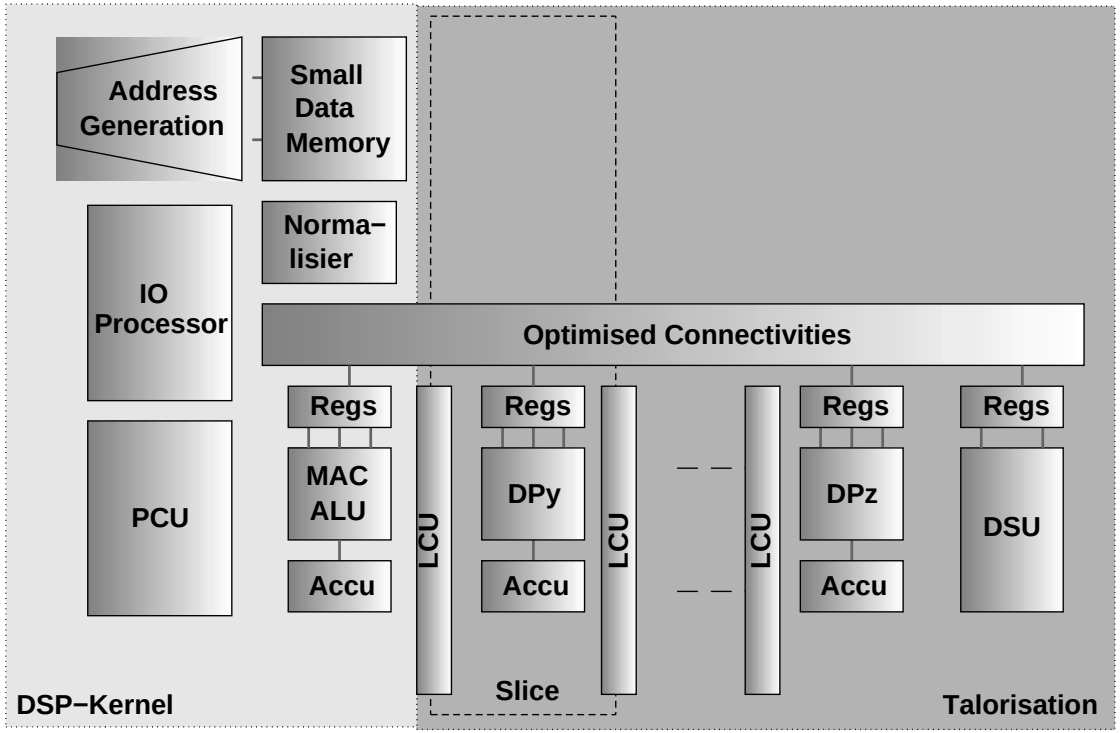
M3-DSP II





- Dual Core
  - DSP Core: Parallel Signal Processing
  - I/O Core: Stream/Bit Processing

LOGO-DSP



- 2nd Base Architecture, Derivatives:
  - GSM Vocoder
  - UMTS Baseband Processor

System View

Apply Orthogonalisation on System Level

- Manipulation: DSP Algorithms
  - Data Word Processing
  - Loops: Parallel Processing
  - Computational Intensive Tasks

➜ DSP Core

- Transfer: I/O Processing
  - Bit Manipulations
  - Bus Operations
  - Control Dominated Tasks

➜ MCU Core

OTHER DSP PROJECTS

- Industry Cooperations
  - PUMA: Processor for UMTS and IEEE 802.11b (Siemens)
  - MMFilt: Multimode Baseband Filter (Siemens)
  - Protocol Processor (Infineon)

- Research
  - Applying Orthogonalisation to RISC: TRISC
  - Compiler Design
  - Automated Interconnectivity Design

REFERENCES, FURTHER READING

LINKS

- My Homepage
  www.cse.unsw.edu.au/~frankie
- DSP Overview: Berkeley Design Technology
  http://www.bdti.com
- Vodafone Chair Homepage
  http://www.vodafonechair.de

PAPERS

- G. P. Fettweis
  Design Methodology for Digital Signal Processing,
  ASAP'97

- G. P. Fettweis
  DSP Cores for Mobile Communications: Where are we going?,
  ICASSP'97

- M. Weiss, U. Walther, and G. P. Fettweis
  A Structural Approach for Designing Performance Enhanced DSPs: 1-MIPS GSM Fullrate Vocoder Case-Study,
  ICASSP'97

- M. Weiss, F. Engel, and G. P. Fettweis
  A New Scalable DSP Architecture for System on Chip (soc) Domains,
  ICASSP'99

- T. Richter, et al.
  A Platform-Based Highly Parallel Digital Signal Processor,
  CICC'01

- T. Dräger and G. Fettweis
  Energy Savings with Appropriate Interconnection Networks in Parallel DSP,
  DFG'02

- M. Hosemann, R. Habendorf, and G. Fettweis
  Hardware-software codesign of a 14.4 Mbit - 64 - state Viterbi decoder for an application-specific DSP,
  SIPS'03