Trends in Compilable DSP Architecture

John Glossner, Jaime Moreno, Mayan Moudgill, Jeff Derby, Erdem Hokenek, David Meltzer, Uzi Shvadron, and Malcolm Ware

IBM Communications R&D Center
Yorktown Heights, NY

glossner@us.ibm.com
Introduction

- **Broadband Applications**
  - Network & Functions
  - Market

- **DSP Algorithms**

- **DSP Architectures**
  - Classification
  - Comparison to General Purpose Architectures
  - Classical, Transitional, Modern DSP Examples

- **Compilation Issues**
  - The “C” Problem
  - Previous solutions
  - IBM solution

- **IBM e-lite DSP**
  - Compilable ultra-low power DSP

- **Conclusions / Future**
Network Functions

- Base Station
- Central Office Sw
- Central Office
- Gateway
- ADSL, Cable
- SOHO/Consumer
- Enterprise/Campus
- Ethernet
- LAN
- WAN
- High Speed Backbone
- High End Routers / Sw
- Work group
- Data center
- Wire/closet
- ACD/Ivr
- PBX
- Pbx Access
- Access Sw/Rte
Broadband Communications

- **Aggregation of multiple streams at a network access boundary**
  - streams from different ports
  - multiple streams from a single port
  - streams have different QoS requirements
  - voice, data, ...

- **Signal processing functions**
  - xDSL
  - VoIP
  - VoDSL
  - V.90

- **Network processing functions**
  - ATM with SAR
  - forwarding
  - QoS / bandwidth management
  - policing / scheduling
  - filtering
  - service enablement
Programmable DSP Market

- CAGR 34.4%
- Growing faster than the general semiconductor market

Communications 64%

Instrumentation 4%
Military 2%
Office Automation 1%
Industrial 6%
Consumer 10%
Computer 13%

General Purpose DSP Market

($B)

1999 2000 2001 2002 2003 2004 2005

4.4 6.1 8.2 10.9 14.5 19.2 25.4

Communications R&D Center

J. Glossner

SIPS-2000
Programmable DSP Mkt

DSP Market by Word Size

- 16-bit Fx: 87%
- 24-bit Fx: 7%
- Floating Point: 6%

DSP Market Share

- TI: 48%
- Lucent: 25%
- ADI: 12%
- Motorola: 10%
- Other: 5%
DSP Algorithms
# DSP Applications

<table>
<thead>
<tr>
<th>DSP Algorithm</th>
<th>System Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Coding</td>
<td>Digital cellular telephones, personal communications systems, digital cordless telephones, multimedia computers, secure communications.</td>
</tr>
<tr>
<td>Speech Encryption</td>
<td>Digital cellular telephones, personal communications systems, digital cordless telephones, secure communications.</td>
</tr>
<tr>
<td>Speech Recognition</td>
<td>Advanced user interfaces, multimedia workstations, robotics, automotive applications, cellular telephones, personal communications systems.</td>
</tr>
<tr>
<td>Speech Synthesis</td>
<td>Advanced user interfaces, robotics</td>
</tr>
<tr>
<td>Speaker Identification</td>
<td>Security, multimedia workstations, advanced user interfaces</td>
</tr>
<tr>
<td>High-fidelity Audio</td>
<td>Consumer audio, consumer video, digital audio broadcast, professional audio, multimedia computers</td>
</tr>
<tr>
<td>Modems</td>
<td>Digital cellular telephones, personal communications systems, digital cordless telephones, digital audio broadcast, digital signaling on cable TV, multimedia computers, wireless computing, navigation, data/fax</td>
</tr>
<tr>
<td>Noise cancellation</td>
<td>Professional audio, advanced vehicular audio, industrial applications</td>
</tr>
<tr>
<td>Audio Equalization</td>
<td>Consumer audio, professional audio, advanced vehicular audio, music</td>
</tr>
<tr>
<td>Ambient Acoustics Emulation</td>
<td>Consumer audio, professional audio, advanced vehicular audio, music</td>
</tr>
<tr>
<td>Audio Mixing/Editing</td>
<td>Professional audio, music, multimedia computers</td>
</tr>
<tr>
<td>Sound Synthesis</td>
<td>Professional audio, music, multimedia computers, advanced user interfaces</td>
</tr>
<tr>
<td>Vision</td>
<td>Security, multimedia computers, advanced user interfaces, instrumentation, robotics, navigation</td>
</tr>
<tr>
<td>Image Compression</td>
<td>Digital photography, digital video, multimedia computers, videoconferencing</td>
</tr>
<tr>
<td>Image Compositing</td>
<td>Multimedia computers, consumer video, advanced user interfaces, navigation</td>
</tr>
<tr>
<td>Beamforming</td>
<td>Navigation, medical imaging, radar/sonar, signals intelligence</td>
</tr>
<tr>
<td>Echo cancellation</td>
<td>Speakerphones, hands-free cellular telephones</td>
</tr>
<tr>
<td>Spectral Estimation</td>
<td>Signals intelligence, radar/sonar, professional audio, music</td>
</tr>
</tbody>
</table>

Source: BDTI
Sample Rates

Sample Rate (Hz)

Radio Signaling and Radar
High Definition Television
Video
xDSL Modems
Radio Modems
Audio
Speech
Voiceband Modems
Control
Seismic Modeling
Instrumentation

ALGORITHM COMPLEXITY

Financial Modeling
Weather Modeling

Signal Processing: 12+ Orders Of Magnitude!!!

Broadband Communications

Source: BDTI
DSP Operations

■ FIR:

\[ y_k = \sum_{n=0}^{N} b_n x_{k-n} \]

■ FFT:

\[ y_k = \sum_{j=0}^{N-1} \omega^{jk} x_j \quad \omega = e^{-\frac{2\pi}{N}} \quad i = \sqrt{-1} \]

■ 2D-DCT:

\[ F(u,v) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos \left( \frac{(2m+1)u\pi}{2N} \right) \cos \left( \frac{(2n+1)v\pi}{2N} \right) \]

■ Neural Nets:

\[ y = f \left( \sum_{k=0}^{N} w_k x_k - \phi \right) \]

◆ Inner Products Easily Described By Vectors
### Code Characteristics

**General Purpose**
- Limited Parallelism
- Control Dominated
- Inherently Serial
- Branch Intensive (20%)
- Limited By Amdahl’s Law

**DSP**
- Parallel Inner Loops
- Loop Setup, then Compute
- Overlapped Parallel Processing
- Multiple Independent Streams

**30% of Dynamic Execution**

**Amdahl’s Law:**
- 30% limits speedup to about 3x

\[
\text{speedup} = \frac{1}{t_{\text{serial}} + \frac{t_{\text{parallel}}}{N_{\text{processors}}}}
\]

(for \( t_{\text{serial}} + t_{\text{parallel}} = 1 \))

\( \text{as } N \to \infty, \text{ speedup } \to \frac{1}{t_{\text{serial}}} \)

**70% of Dynamic Execution**

**Gustafson’s Law:**
- \( t_{\text{parallel}} \) is independent of \( N \)
- Parallel Portion Scales With \( N \)
- Linear slope!

\[
\text{ScaledSpeedup} = \frac{t_{\text{serial}} + t_{\text{parallel}}}{N_{\text{processors}}} \frac{N_{\text{processors}}}{t_{\text{parallel}} + t_{\text{serial}}}
\]

\[
= N + (1-N) \frac{1}{t_{\text{serial}}}
\]
Workload Comparisons

Amdahl’s Law

- Speedup vs. % Parallel Code
- General Purpose
- DSP
- Video

J. Glossner
SIPS-2000
Computational Requirements

- **MPEG II Encode, 30f/s, Full Search, P=16, (35)**
- **MPEG II Encode, MP@ML, 30f/s, ALG Search, P=16, (1.68)**
  - **P X 64 CIF, 15 f/s, 100kb/s (1.2)**
- **DFSE EQ - 2Mb/s (650)**
- **Full-rate DAB Viterbi Decoder, MPEG II MP@ML, 30fps Decode (600)**
- **16 X GSM_EFR (380)**
- **ADSL XCVR - 6.1Mb/s (360)**
- **ADSL XCVR - 1.5Mb/s (100)**
- **GSM Terminal (Baseband, HR) (52)**
- **GSM_HR, AC-3 decode, V.34 (20)**
- **GSM_EFR (16)**
- **GSM_FR (2.5)**

- **500 MOPs**
- **100 GOPs**
- **10 GOPs**
- **1 GOP**
- **500 MOPs**
- **300 MOPs**
- **200 MOPs**
- **100 MOPs**

---

J. Glossner

Communications R&D Center

SIPS-2000
Architecture Domain

- GSM Terminal (HR/EF) (52M)
- VFLEX2 (30M)
- GSM_HR, V.34bis (20M)
- GSM_EFR (16M)
- GSM_FR (2.5M)

1 MAC DSP

IBM Communications R&D Center

J. Glossner

SIPS-2000
New Architectures Required

- **2k Performance**
  - Single Chip STB (1.5G)
  - H.263L + GSM Terminal (EHR/HSCSD/GPRS) (1.0G)
  - Single Chip DAB XCVR (800M)
  - DFSE EQ (UMTS) - 2Mb/s (650M)
  - MPEG II MP@ML, 30fps Decode (600M)
  - ADSL XCVR - 6.1Mb/s (500M)

- **MULTI-CARRIER GSM BTS (800M)**
  - DAB XCVR (800M)
  - 16 X GSM_HR/EFR (400M)

- **500 Performance**
  - Single Chip DAB XCVR (800M)
  - Single Carrier GSM BTS (180M)
  - 4 X GSM_HR/EFR (110M)
  - ADSL XCVR - 1.5Mb/s (100M)
  - GSM Terminal (EHR/HSCSD/GPRS) (80M)
  - AC-3/MUSICAM Decode (20M)

- **100 Performance**
  - Single Carrier GSM BTS (52M)
  - VFLEX2 (30M)
  - GSM_HR, V.34bis (20M)
  - GSM_EFR (16M)
  - GSM_FR (2.5M)

- **1 MAC DSP**
  - GSM Terminal (HR/EFR) (52M)
  - VFLEX2 (30M)
  - GSM_HR, V.34bis (20M)
  - GSM_EFR (16M)
  - GSM_FR (2.5M)

- **2 MAC DSP**
  - Single Carrier GSM BTS (180M)
  - 4 X GSM_HR/EFR (110M)
  - ADSL XCVR - 1.5Mb/s (100M)
  - GSM Terminal (EHR/HSCSD/GPRS) (80M)
  - AC-3/MUSICAM Decode (20M)

- **MP**
  - MPEG II Encode, MP@ML, 30fps, ALG Search, P=16, (1.68G)
  - Single Chip STB (1.5G)
  - H.263L + GSM Terminal (EHR/HSCSD/GPRS) (1.0G)
  - Single Chip DAB XCVR (800M)
  - DFSE EQ (UMTS) - 2Mb/s (650M)
  - MPEG II MP@ML, 30fps Decode (600M)
  - ADSL XCVR - 6.1Mb/s (500M)

( ) = OPS

J. Glossner  SIPS-2000

Communications R&D Center
Architecture Domain

New Architectures Required

MACs

2k

500

100

Performance

New Architectures Required

MP

MP

Multi-Carrier GSM BTS (800M)
DAB XCVR (800M)
16 X GSM_HR/EFR (400M)

2 MAC DSP

Single Carrier GSM BTS (180M)
4 X GSM_HR/EFR (110M)
ADSL XCVR - 1.5Mb/s (100M)
GSM Terminal (EHR/HSCSD/GPRS) (80M)
AC-3/MUSICAM Decode (20M)

1 MAC DSP

GSM Terminal (HR/EFR) (52M)
VFLEX2 (30M)
GSM_HR, V.34bis (20M)
GSM_EFR (16M)
GSM_FR (2.5M)

MP

MPEG II Encode, MP@ML, 30fps, ALG Search, P=16, (1.68G)
Single Chip STB (1.5G)
H.263L + GSM Terminal (EHR/HSCSD/GPRS) (1.0G)
Single Chip DAB XCVR (800M)
DFSE EQ (UMTS) - 2Mb/s (650M)
MPEG II MP@ML, 30fps Decode (600M)
ADSL XCVR - 6.1Mb/s (500M)

() = OPS

Symphonic Synthesis
Natural Language Processing
Real-time Speech Recognition
3G Wireless
Software Radio

Symphonic Synthesis
Natural Language Processing
Real-time Speech Recognition
3G Wireless
Software Radio

J. Glossner
SIPS-2000

Communications R&D Center
DSP Classifications
Processor Classification

Processor

- DSP
- General Purpose
Processor Classification

- Processor
  - DSP
    - Floating Point
      - 32 bit IEEE
      - Other
  - General Purpose
    - Floating Point
      - 32/64 bit IEEE
      - Other (80 bit)
Processor Classification

Processor

DSP

- Fixed Point
  - 16 bit
  - 20 bit
  - 24 bit
  - 32 bit IEEE
  - Other

Floating Point

- 32 bit + subsets
- 64 bit + subsets
- 32/64 bit IEEE
- Other (80 bit)

General Purpose

- Integer
- Floating Point
Numeric Representations

Fixed Pt. Fractional

Integer 2's Complement

Floating Point

-2^0 2^1 2^2 2^3 2^4 2^5 2^6 2^7
-2^6 2^5 2^4 2^3 2^2 2^1 2^0
-2^1 2^0 2^1 2^2 2^3 2^4 2^5
-2^3 2^2 2^1 2^0

1 0 1 0 1 1 0 0
1 0 1 0 1 1 0 0
0 1 1 0 1 0 0
0 1 0 1

-1 + .25 + .0625 + .03125 = -.65625
-128 + 32 + 8 + 4 = -84
1 + .5 + .125 = 1.625

Multiplication complicates fractional representations

1.625 x 2^5 = 52.0

Source: BDTI
DSP vs. General Purpose

- **Execution Predictability**
  - Required to guarantee real-time constraints
- **1 cycle MAC**
- **0-overhead Loop Buffer**
- **Complex Instructions**
  - Multiple Operations Issued
- **Harvard Memory Architecture**
  - Multiple memory access
- **Specialized Addressing Modes**
- **Operate on Vector Stream Data**
- **Data-independent Execution**
- **Fractional Arithmetic**
- **Pipeline Non-interlocked**
  - Shallow Pipeline (3-5 stage)
- **Delayed Branch**

- **Fast But Non-predictable**
  - Dynamic Instruction Issue
  - Non-deterministic caches
- **Multicycle MAC**
- **Branch Prediction**
- **RISC Superscalar Instructions**
  - Multiple Instructions Issued
- **Von Neumann Architecture**
  - Split Cache has similar benefit
- **Typically Linear Addressing**
- **Caches Assume Locality**
- **Data-dependent Execution**
  - Dependent upon operands
- **Integer Arithmetic**
- **Pipeline Typically Interlocked**
  - Deep Pipeline (5+ stage)
- **Multicycle Branch**
### ISA Comparison

<table>
<thead>
<tr>
<th>ISA</th>
<th>Orthogonality</th>
<th>Parallelism Within an Instruction</th>
<th>Number of Instructions / Addressing Modes</th>
<th>Width of Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>RISC</td>
<td>High</td>
<td>None</td>
<td>Small</td>
<td>Fixed</td>
</tr>
<tr>
<td>CISC</td>
<td>Low-Medium</td>
<td>Medium</td>
<td>Large</td>
<td>Variable</td>
</tr>
<tr>
<td>DSP</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
<td>Mostly Fixed</td>
</tr>
</tbody>
</table>
DSP Architectures
Trends in DSP Processors

■ Software programmability
  ● Focus on compilation

■ Ultra-low power

■ Very high performance

■ Computational performance with control processing
Anticipated/Projected Entrants:
- TigerSHARC: 1.2 GMAC/Sec @ 2-8 W
- AltiVec: 4 GMAC/Sec @ 5+W
- C62x: 400 MMAC/Sec @ 1.8W
- C64x: (2005+ / 1.1GHz) 4.4 GMAC/Sec @ ??W

Future Sweet Spot
Classification

- Classical DSPs
- Transitional DSPs
- Modern DSPs
- Future DSPs
Classical DSP Architectures

- Dot product processors
- Poor compiler targets
- Non-orthogonal
- Small Address space
- Multiple address spaces
- Compound ISA
- Highly focused on an application
 TI C54x

### Quintessential Classical Architecture
- 8, 16-bit busses
- 40-bit ALU
- 2, 40-bit accumulators
  - 8 guard bits
- 40-bit Barrel shifter
- 17x17 multiply unit with 40-bit adder & 1-cycle throughput
  - 0 detect, rounding, saturation
- Compare, select, store unit
  - Viterbi algorithm
- Exponent encoder
- 16-bit address space
  - 548/549 use segments to give bit address
  - Circular, bit-reversed addressing
- Bock repeat

Source: TI C54x CPU Reference

---

J. Glossner  SIPS-2000

Communications R&D Center
A 16-bit machine with many modes!

- Block repeat active
- Overflow mode
- Sign extension mode
- Double precision or dual 16-bit precision mode
- Fractional mode (left shift multiply $<<1$)
- Accumulator shift mode (5-bit shift field mode)
- Saturation on multiplication mode
  - before accumulation
  - ETSI GSM operation
- Saturation on Store
- Compiler mode
  - Relative addressing using Data Page Pointer or Stack Pointer
Transitional Architectures

- Characteristics of both Classical and Modern DSPs
- More programmable but not architected for compilation
- Typically small address space (64 kB)
- More computational units (dual-MAC)
- Parallel instruction issue
  - Versus compound instructions
- More registers with RISC-like ISA
- Media Processors
- General Purpose Processors with SIMD
Infineon Carmel

- **Superscalar**: two 24-bit instructions issued every cycle
- **Up to six instructions executed per cycle with CLIW™**
- **Conditional execution**
- **Memory-based architecture**
  - Memory operands used directly
- **Memory accesses**: 4x16 data read and 2x16 data write (total of 4 memory access per cycle)
- **Data buffers addressing**: linear, modulo, special and bit-reversal
- **Execution units**: 2xALU, 2xMAC, Barrel Shifter, Exp. Unit
- **Six 40-bit accumulators**
- **Four nesting levels of zero-overhead loops**
- **8 Stage pipeline**

Used Courtesy of Infineon Technologies
ZSP Block Diagram

- **4-issue Superscalar engine**
  - Simple RISC-like programming model
  - Orthogonal Instruction Set
  - Register-based Operations

- **Pipeline complexity managed by Hardware**
  - 5-stage Hardware controlled Pipeline
  - Relieves DSP programmers from having to deal with pipeline nuances
  - Eliminates programming errors due to hidden states and execution restrictions

- **Parallel execution optimized by Hardware**
  - Hardware automatically schedules instructions
  - Programmers don’t need to find parallelism. Simple straight-line coding.
  - No delay slots, No prefixes, No wasteful NOPs.

Used Courtesy of LSI Logic

---

J. Glossner

SIPS-2000

Communications R&D Center
Modern DSP Architectures

- Focus on compilability
- RISC based with Control + DSP processing
- Highly Parallel
- Multiple instruction issue
- Multiple operation issue
  - MAC
  - ALU
  - Load/Store
- Predominately VLIW
  - Some use of SIMD
- 32-bit unified address space
A groundbreaking machine
- VLIW with 8 functional units
  - Up-to 8, 32-bit instructions issued per cycle
  - 2 MPY, 6 Arithmetic
    Int 32x32 -> 64-bit result
- Instruction packing
- RISC-based
- Conditional execution
- 8/16/32/40-bit types
- Saturation / Normalization
- Bit field manipulation
- Circular addressing
- Deep pipeline
StarCore SC140 Core Block Diagram

-300 MHz @ 1.5 V; Low Power, Static Design
-16 Functional Units Total
-16 Bit Data, 40 Bit Accumulators
  -Single cycle MAC, Integer and fractional data
-32 Bit Address, Byte addressable
  -One Unified data and program space
-Data Register File: 16 40-bit General Purpose Registers

Address Register File (32-bits, 27 Total, 16 General Purpose)
  -Also 4 modulo, 4 offset, 2 Stack Pointers, 1 modulo control
-Branch Registers: 8 hardware loop registers in Branch Unit
-128 bit VLES
  -Up to 6 instructions per clock, including 4 MACS
-128 Bit Data Bandwidth
  -Up to 8 data words per clock (4.8 GBytes per second)
Summary DSP Architectures

■ Classical
  ● Dot product processors
  ● Poor compiler targets
  ● Non-orthogonal
  ● Small Address space
  ● Multiple address spaces
  ● Compound ISA
  ● Highly focused app.

■ Modern
  ● Focus on compiler / architecture pair
  ● Highly parallel
  ● Multiple MACs
  ● 32-bit unified address space
  ● RISC-based
    ◆ Control + DSP

■ Transitional
  ● More programmable
  ● Some classical features
  ● Some modern features
DSP Compilation
DSP Application Complexity

10x Complexity every 10 years
Compiler Productivity

Design Algorithms → Map to Fixed Point C → Write DSP Specific C → Write DSP Assembly → Hand Schedule Operations on DSP → Final Product

6-9 Months!
Compiler Productivity

1. Design Algorithms
2. Map to Fixed Point C
3. Write DSP Specific C
4. Write DSP Assembly
5. Hand Schedule Operations on DSP
6. Compile
7. Final Product

If floating point implemented:
- 6-9 Months!

Final Product:
- 6-9 Months!
Compilable Architecture

- Optimize
  - Cost / Power
  - Performance

Compiler

Architecture

Implementations
- GSM
- DSL
- VoIP
- 3G

Algorithms
DSP Compilation Problem

- **Mismatch between C & DSP**
  - 16-bit fixed point
  - 40-bit accumulators with mixed type arithmetic
  - Saturation arithmetic vs. modulo semantics

- **Historically...**
  - DSPs have had compiler unfriendly architectures
    - very complex instructions
    - non-orthogonal, specialized resources
    - exposed pipelines
  - DSP compiled performance
    - Typical: 1/10 speed of handwritten assembly
    - Assembly code is required for performance
DSP Compilation Solutions

- **Extensive libraries**
  - Often more than 1000 functions
  - Resource consuming but high reuse

- **C language extensions (DSP-C)**
  - Type support (Q15)
  - Memory disambiguation

- **Intrinsics**

- **Handwritten assembly code**

- **Matlab compiler (BOPS)**
  - 64-bit double precision of Matlab problematic

- **Tensor compiler**
  - Algorithm specific
  - Highly skilled algorithm designers required
DSP Intrinsics

- Intrinsics allow programmers to use instructions a compiler can not generate
- Has appearance of a function call in C
  - Replaced with assembly statements by compiler
  - Highly architecture dependent
- Often condense 10 assembly instructions into 1
- Early attempts were blocking
  - Inlined asm statement
- Non-blocking pioneered by Lucent
  - Written in the compiler’s intermediate language
  - Semantics of side effects well defined
  - Allowed for further optimization
  - Architecturally neutral
IBM DSP Compilation Solution

- **Intrinsics work well but...**
  - Compiler writers become DSP assembly language programmers
  - Only work for a specific application

- **IBM Solution: Semantic Analysis**
  - Type inference
  - no intrinsics: out-of-the-box C compiler
  - near-parity with assembly code
  - novel DSP optimizations
  - existing optimizations adapted for DSPs
  - power-driven optimizations
Compiled Simulator

- **Aids compiler debug**
  - Fast compile/execute/check for correctness turn-around time

- **Provides profile information**
  - Add extra instructions to gather statistics

- **Debug using host debugger**
  - One-to-one mapping between instruction breakpoints and compiled simulator code sequences
  - One-to-one mapping between architected state and compiled simulator state

- **Mix and match with native code**
  - Can use native libraries
  - Debug by compiling some files using native compiler
IBM e-lite Architecture
Architecture Domain

New Architectures Required

- MPEG II Encode, MP@ML, 30fps, ALG Search, P=16, (1.68G)
- Single Chip STB (1.5G)
- H.263L + GSM Terminal (EHR/MSCSD/GPRS) (1.0G)
- Single Chip DAB XCVR (800M)
- DFSE EQ (UMTS) - 2Mb/s (650M)
- MPEG II MP@ML, 30fps Decode (600M)
- ADSL XCVR - 6.1Mb/s (500M)

Symphonic Synthesis
Natural Language Processing
Real-time Speech Recognition
3G Wireless
Software Radio

MACs

2k

Performance

500

100

( ) = OPS

J. Glossner

SIPS-2000

Communications R&D Center
e-lite Objectives

■ Fully compilable DSP
  ● “out-of-the-box” C compilation

■ Low-power focus
  ● Algorithm techniques
  ● Software techniques
  ● Architectural techniques
  ● Microarchitectural techniques
  ● Circuit techniques
  ● Process techniques

■ Applications Area: Broadband Communications
  ● 3G wireless, VoIP, xDSL
Prior Research Contributions

- Low Power
  - Methodology
  - Circuits
  - Architecture
  - Process

- DSP Compilation
  - SIMD Vectors
  - Scheduling
  - Type Recognition

- Chameleon
- Vector Research

- SiGe
- SOI

- 3G Wireless
  - Wide Band CDMA
  - 2.5G GSM
  - S/W Radio

- Processor Design
  - PPC
  - IBM DSP

- WLAN
- Bluetooth
e-lite Architecture

- Developed hand-in-hand with the compiler
- 64-bit multiple instruction bundles
  - 3-instruction issue per cycle peak
  - Each instruction may specify multiple operations
- Pre-decoded Instruction cache
  - 5-issue per cycle peak
- Non-interlocked pipeline
  - Except long loads
  - Minimal Control paths
- SIMD execution
- Streaming Register File
- Fully visible hardware resources

<table>
<thead>
<tr>
<th></th>
<th>20-bit</th>
<th>20-bit</th>
<th>20-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>30-bit</td>
<td></td>
<td>30-bit</td>
</tr>
</tbody>
</table>

J. Glossner  SIPS-2000  Communications R&D Center
e-lite Execution Units

- **Integer Unit**
  - 32-bit modulo arithmetic

- **Storage Access Unit**
  - Byte (8)/half-word (16)/word (32) transfers
  - 64-bit Vector transfers

- **Vector (SIMD) Unit**
  - 16-bit Q15 format

- **Vector Reduction Unit**
  - Parallel accumulation

- **Branch Unit**
e-lite Compiler Results

GSM EFR/AMR Speech Coder
Typical w/ and w/o intrinsics
e-lite untouched C Code
Conclusions

■ **DSP design has undergone major paradigm shift**
  - Soaring costs of assembly programming have altered DSP architectures
  - Compilable DSPs are required
  - Ultra-low power implementations are desirable
  - Deterministic execution still required

■ **Multiple-issue highly parallel architectures will become more prevalent**
  - Mix of control and compute codes
  - Vectorization technology makes SIMD implementations possible

■ **Compilers will play an even more important role in the future**
  - Take into account memory hierarchy
  - Parallelism extraction
  - Type analysis
  - Precision analysis