EISC Architecture:
Computer Architecture for Era of Post PC
Agenda

- EISC Processor Overview
  - Trends
  - EISC: Extendable Instruction Set Computer
  - EISC processors
- AE32000; 32-bit EISC processor
  - AE32000 Features
  - AE32000 Microarchitecture
Development Cycle in Microprocessor

- **CISC**: Complex Instruction Set for Mini & Desktop PC
- **RISC**: Reduce Instruction Set for Desktop & Server System
- **EISC**: Extended Instruction Set for Post-PC Devices

http://www.adc.co.kr
Complex Instructions
- Easy programming with Assembly Code
- Many instructions for many situations

Variable Length Instruction
- Complex for hardware Implementation
- Various execution time for each instruction
- Reduced Instruction Set
  - Weak point of CISC
    - Part of commands are used even though existing of many commands
    - Hardware consumption for rarely executing commands
- Make Common Case Faster!
  - Reduce # of Instructions
  - Take advantage of Compiler Technology and Pipeline architecture
  - Simple & Efficient Hardware
Post-PC Era.

- **Devices**
  - Emergence of various digital equipments
  - Embedded System!
    - “An embedded system is a special-purpose system in which the computer is completely encapsulated by the device it controls” [wikipedia].
  - Embedded microprocessor.

- **Requirements**
  - Low cost
  - Appropriate performance for required operation
  - High power efficiency
Embedded Microprocessor

- General Microprocessor
  - 32-bit/64-bit Microprocessor
  - Accelerating for various jobs
  - Performance centered

- Embedded Microprocessor
  - System control by embedded in system
  - Executing of Special jobs

Desktop/Server Microprocessor

Cost

Performance
Increasing of Required performance
- Interconnection of complex peripherals
- Increasing of number of control units
- Requirement of network connection

Requirement for DSP operation
- Multimedia application
- Graphic user interface

General Microprocessor

Modern Embedded Microprocessor

Classical Embedded Microprocessor

Desktop/Server Microprocessor

EISC
Advanced Digital Chips Inc.

http://www.adc.co.kr
5 Criteria for Modern Embedded Microprocessors

- Low Power
- Code Density
- Peripheral Integration
- Price/Performance ratio
- Multimedia Acceleration

M.Schlett

http://www.adc.co.kr
Embedded Microprocessor

- **Code Density**
  - Inverse of static code size
  - Static code size
    - Memory size for storing program
    - Code size + Initialized static data

- **High code density**
  - Small memory size requirement for same program storing
    - Reducing memory size
  - Reducing power consumption for command call
  - Reducing performance degradation for memory access
    - Accessing time difference between memory and processor
  - High efficiency of instruction cache

http://www.adc.co.kr
Embedded Microprocessor

- Code Density
  - Low Power
  - Price-Performance Ratio

Low Power

Price / Performance ratio

http://www.adc.co.kr
Embedded Microprocessor

- Multimedia Acceleration;
- Restriction
  - Price-Performance Ratio
  - Low Power

Low Power

Price-Performance ratio

http://www.adc.co.kr
Code Density

- One of the most important metric

CISC vs RISC, and...

- CISC; Variable Length Code
  - Various instruction for each case
  - Various instruction length
    - High code density
- RISC; 32-bit Fixed field instruction
  - Using of same code
    - Low code density
Improving Code Density

- Instruction Compressing
  - Using lossless compress method
  - Increasing hardware cost
  - High interoperability

- ISA using short instruction
  - Short-length Instruction
  - Problem of interoperability

- Simple hardware
- Low code density

RISC

Compressed Code RISC

Code Compression

http://www.adc.co.kr
Improving Code Density

- Code Compression
- RISC
- Compressed Code RISC
- Hybrid
- 16-bit/32-bit Code
  - Defining new command
  - Swap of common mode
- 16-bit Compressed Code
  - Defining new instruction
- Compressed Code RISC
  - Long word control by short instruction
  - 16-bit/24-bit instruction set, 32-bit data
- Short Instruction
  - Restriction of size of bit for command encoding
- Command Encoding
  - OP-code + Operand
Operands
- Operation between Registers requires small bits
  - Same bit requirement for number of register
- Immediate/Offset
  - Constant value for operation
  - Offset of indexed addressing

Length of Immediate/Offset
- Varying for instructions (up to 32bit)
- High utilization of short immediate value, low utilization of long immediate value
Improving Code Density

- Compressed code architecture
  - Lack of command filed for immediate value

- RISC based architecture
  - Case of long immediate value is required
    - Execute after moving immediate value to general register
    - Combination of multiple “LDI–SHIFT–ORI” commands
      - Complicate procedure
    - Load after locating immediate value to static data area
      - Require of data access
  - Reserve as many bits for immediate value
EISC: Extendable Instruction Set Computer

- Extend immediate value at Special purpose register or Extension Register (ER)
- LERI Instruction; Command for load/store from/to ER
  - Elimination of general register using and combination of complex Instructions

Immediate value of Instruction

- Assigning suitable size by workload analysis
- <20% instrs.

80% Instr. w/o LERI
20% Instr. w/ LERI
LERI makes long immediate

<table>
<thead>
<tr>
<th>LERI</th>
<th>IMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>E-Flag</td>
<td>Extension Reg.</td>
</tr>
</tbody>
</table>
LERI makes long immediate

LD DST IDX IMM

SET E-Flag

IMM Extension Reg.

http://www.adc.co.kr
LERI makes long immediate
Assigning immediate by LERI
- LERI itself is fixed length instruction

Elimination of complex procedure for immediate value access
- Low burden for long immediate
  - Assign relatively small size of bit for immediate
  - Increasing of bit size for opcode → Increasing number of instruction
  - Increasing of bit size for register indexing
- Increasing available general register
  - 16 GPRs
  - ARM–THUMB; 8GPRs
- No GPR for immediate value
Selection freedom for instruction type

- Selection according to use LERI or not
  - Instruction such as `addq` is assigned if the short immediate values are frequently used
- Good for long immediate operation
  - DSP

General `u-proc.`

- `sub`:
  - `$Rd = $Ra - $Rd$
- `subi`:
  - `$Rd = $Ra - $Rd$`

EISC

- `sub`:
  - `$Rd = $Ra - $Rd$
- `leri/sub`:
  - `$Rd = $Rd - $Ra - $immediate$`
Impact of LERI

- Media benchmarks

Instruction Distribution

LERI: about 15%

http://www.adc.co.kr
Impact of LERI

- EEMBC: Automotive, Consumer, Telecomm, Network, Office

LERI: about 12%
EISC Architecture

11.5% higher code density than ARM9TDMI

6.5% higher code density than MIPS16

Benchmark: libc, libm, libstdc++
Average 18.9% higher code density

Mediabench, gcc3.2
Memory Access

- 16GPRs
  - Most of Compressed Code RISC; 8 GPRs

- EISC can make long immediate easily
  - Most normal RISCs have multiple commands for 32bit immediate value
    - LUI, ORI sequence (MIPS)
    - memory LOAD

<table>
<thead>
<tr>
<th></th>
<th>AE32K</th>
<th>TR4101</th>
<th>ARM9TDMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCD</td>
<td>1.00</td>
<td>1.07</td>
<td>1.13</td>
</tr>
<tr>
<td>Memory Access Ratio</td>
<td>30.2%</td>
<td>48.4%</td>
<td>46.5%</td>
</tr>
</tbody>
</table>

35% less data-memory access than ARM9TDMI & MIPS16
Processor Family

High End
Microprocessor
- 50+ MIPS
- Robot Control
- Media Processing

Main Stream
Microprocessor
- 50+ MIPS
- Control and Data processing
- General OS Support

Low End
Microcontroller
Under 50 MIPS, for Control purpose

http://www.adc.co.kr
EISC architecture
- We believe it is right ISA for post-PC market
- LERI instruction
  - Flexibility
  - Makes instruction more compactor
- Better code density
Thank you...