Energy Efficient Code Generation Using rISA*

Aviral Shrivastava, Nikil Dutt
aviral@ics.uci.edu, dutt@ics.uci.edu

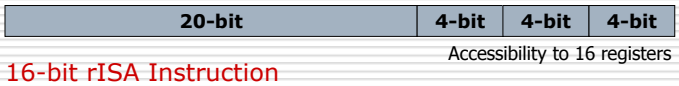
Center for Embedded Computer Systems,
University of California, Irvine, CA, USA.
http://www.ics.uci.edu/~aces

*This work was partially supported by SRC contract 2003-HJ-1111
and NSF Grants CCR 0203813 and CCR 0205712

reduced bit-width Instruction Set Architecture (rISA)

- Dual Instruction Set
  1. Normal 32 bit wide instructions
  2. 16 bit wide instructions (rIS)

Normal 32-bit Instruction



Accessibility to 16 registers

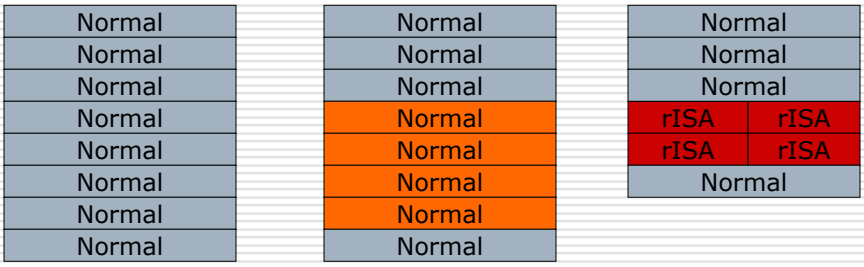
16-bit rISA Instruction

Fewer opcodes Accessibility to only 8 registers

- Popular feature to reduce code size
  - ARM7TDMI, MIPS, ARC Tangent A5

rISA Code

rISA: reduced bit width Instruction Set Architecture

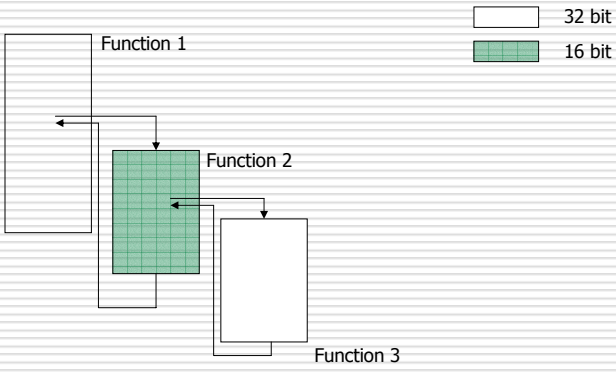


- rISA-ization (sounds like resize-ation):
  normal instructions → rISA instructions

Reducing Energy Consumption using rISA

- Existing rISA compilers aim to achieve maximum code compression
  - Energy reduction is a byproduct
- Further energy savings can be achieved by compiling for minimum energy
- We propose a code generation approach targeted to reduce energy consumption using rISA

rISAization: At what granularity?



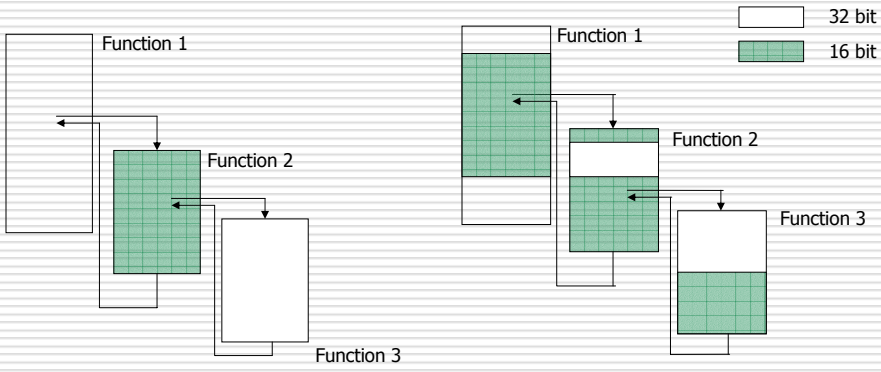
Previous approachs : Function level

rISAization : Increased Register Pressure

Accessibility to only 8 registers

Fewer opcodes

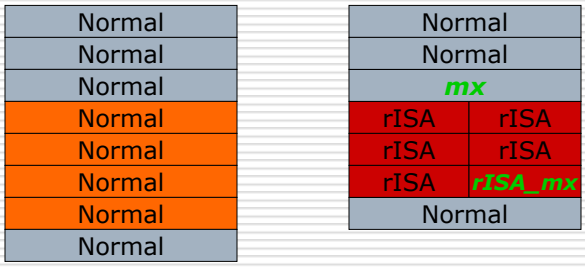
rISAization: Instruction Level



Previous approach: Function level **Our approach: Instruction level**

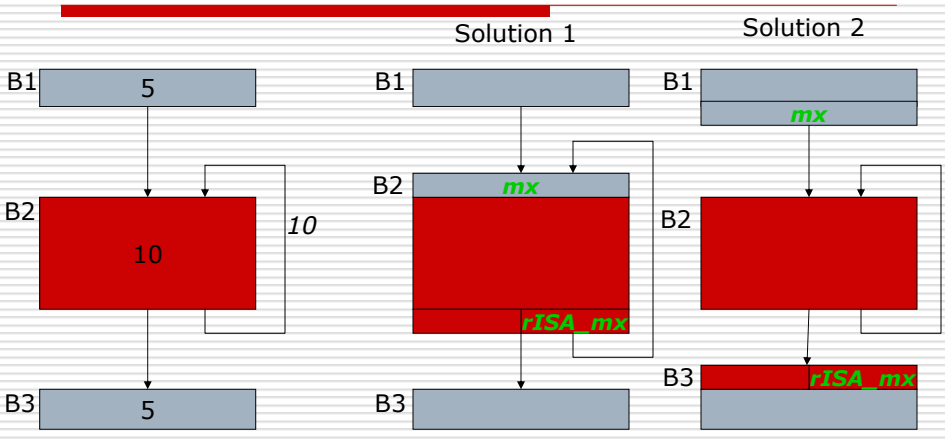
rISAization at Instruction Level granularity is better!

Instruction-Level rISAization: Overhead



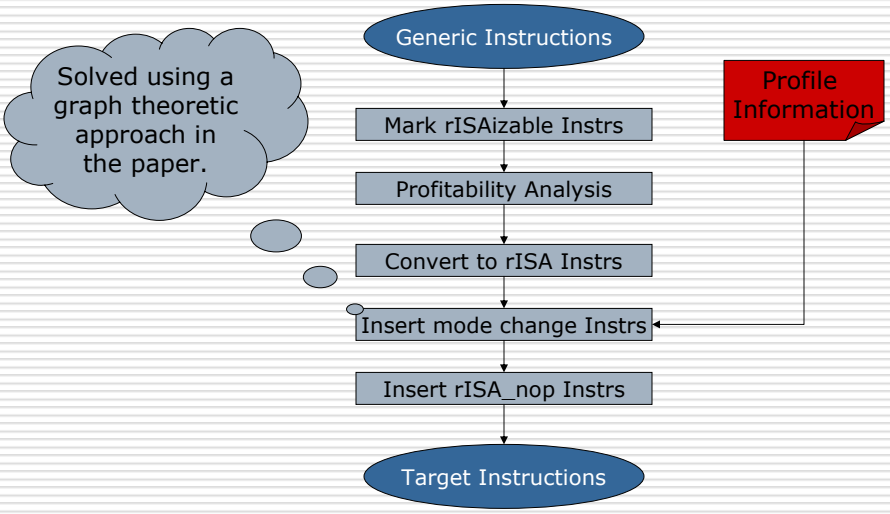
- Need Mode change instructions
  - ***mx*** – change mode from normal to rISA
  - ***rISA_mx –*** change mode from rISA to normal

Where to Insert Mode Change Operations?



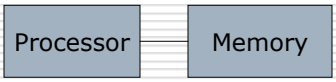
Energy (Solution1) > Energy(Solution2)

Compilation Flow for rISA



Experimental Setup

- Non-cached Architecture



- Cached Architecture

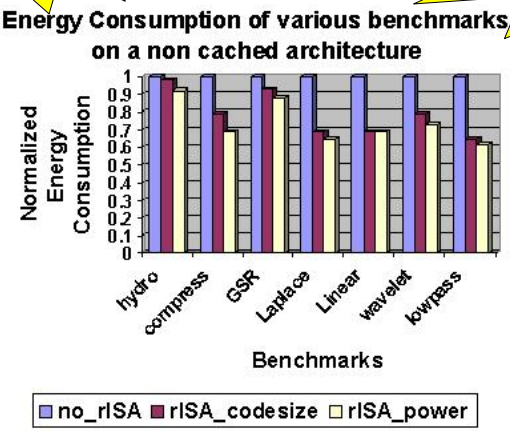
Modeling Infrastructure

- Platune on chip Power Models [VaGi00]
- Instruction Memory Subsystem Energy
  - Cache Energy
  - Memory Energy
  - Bus Energy
  - Translation Logic Energy

26% Energy Savings on Non Cached Architecture

26% overall energy savings

5% savings due to energy aware compilation

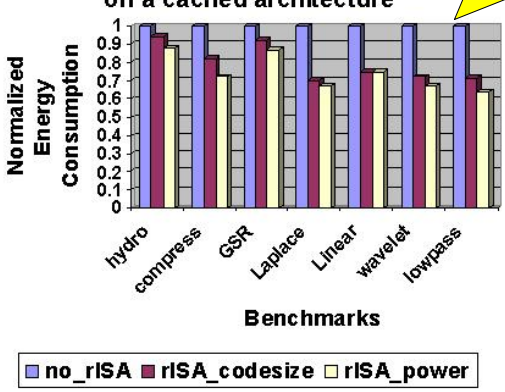


***Instruction Memory Energy Savings**

33% Energy Savings On Cached Architecture

33% overall energy savings

10% savings due to energy aware compilation



*Instruction Memory Energy Savings

Conclusion

- rISA is a popular architectural feature in processors to reduce code size and energy
- We presented a rISA compilation technique to further reduce instruction memory energy
- Our approach shows an average 30% reduction in instruction memory energy