Code Portability and Related Issues for EPIC

Monday, September 26, 2005

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Introduction

- IA-64 Similarities/Differences
- Portability Issues
- Other Issues
- Summary

IA-64 Similarities

- 64-bit, two's complement
- ASCII/UNICODE
- IEEE Floating Point
- Sequential Execution with branching

IA-64 Differences

- Very Wide Instruction Words
- Multiple Operations per bundle
- Predication
- Register Stack
- WYSIWYG – No Instruction Reordering
- Not intended for hand-coding

Is this new?

- Overall – no; FPS did it in the 1980's
- For a widely used machine – yes
- Removal of reordering and renaming, scoreboards
- Compilers have been playing "hide and seek" with the hardware for years
- "Hide and Seek" is counterproductive

Portability Issues

- Is IA-64 different?
- Do these differences matter to developers?
- Are the details of the instruction set relevant?

Is IA-64 different? –

- Technical review – Data Formats
- Issues in porting
- History and context of EPIC (CISC, RISC, EPIC)
- Technical emphasis
- Sizing and migration

Don’t know means DON’T KNOW –

- Comparison against ALPHA, a 64-bit RISC architecture
  - Published Alpha specifications
  - Published Itanium(tm) specifications
  - Digital Technical Journal
  - Validated against field test experience

Don’t know means DON’T KNOW (cont’d) –

- and upon applicable experience
  - PDP-11 to VAX (1978 – present)
  - VAX to Alpha (1992 – present)
  - General experience

My personal background –

- 25 years of experience on multiple platforms
- Platforms (integer size/address size/ integer format)
  - IBM System/360/370 (32/24/2)
  - Digital PDP-11 (16/16/2)
  - Digital VAX (32/32/2)
  - CDC 6600 (60/18?/1)
  - Digital PDP-10 (36/9(and/or)7/2)
  - Digital/Compaq/HP Alpha (64/64/2)

My personal background (cont'd) –

- Compiler code generator developer
- FPS-164 array processor experience
- Portable software developer

Architectural Attributes

	PDP-11	VAX	Alpha	Itanium
Architecture Type	1/2 Address	CISC	RISC	EPIC
Address Size	16	32	64	64
Integer Size	16	32	64	64
Byte Order	little	little	little	little
Alignment	word	none	quad	quad

Architectural Attributes

	PDP-11	VAX	Alpha	Itanium
Architecture Type	1/2 Address	CISC	RISC	EPIC
Address Size	16	32	64	64
Integer Size	16	32	64	64
Byte Order	little	little	little	little
Alignment	word	none	quad	quad

Architectural Attributes

	PDP-11	VAX	Alpha	Itanium
Architecture Type	1/2 Address	CISC	RISC	EPIC
Address Size	16	32	64	64
Integer Size	16	32	64	64
Byte Order	little	little	little	little
Alignment	word	none	quad	quad

Porting –

- Cross Platform/OS
  (Solaris C/C++ to OpenVMS Alpha)
- Cross O/S
  (OpenVMS C/C++ to Tru64 C/C++)
- Cross Platform/Same OS
  OpenVMS VAX to/from Alpha

Porting Difficulty –

	Operating System	
	Same	Different
Same Platform	0	10*
Different Platform	1	15*

* Highly Application Sensitive

Itanium Issues–

- Atomicity
- Precision
- Address Size
- Granularity
- Alignment
- Byte Ordering

Atomicity –

- on VAX, INCx was accidentally thread atomic
- on Alpha, translated as load/add/store
- Alpha translation was not safe
- accidental atomicity was not part of the specification
- solution – use ADAWI

Precision –

- VAX floating point/integer sizes/formats different from Alpha
- Alpha and Itanium – same precision/formats

Address Size

- VAX – 32 bits
- Alpha/Itanium – 64 bits
- VAX to Alpha required data structure changes

Granularity –

- VAX was byte aligned for all operands
- Alpha/Itanium require natural alignment
- VAX was prone to fractured loads/stores

Data Alignment –

- VAX was byte aligned – all operands
- Alpha/Itanium require natural alignment
- No difference between Alpha/Itanium

Instructions, Bundles, etc. –

- IA-64 is designed to be compiled
- CISC/RISC were intended for human coding
  - scoreboard for safety
  - easily decodeable
  - simple syntax
  - IA-64 bundles are designed to be generated

Byte Ordering –

- VAX is little endian (low byte addressed)
- Alpha is little endian
- Itanium operates little/big endian

In terms of portability –

- from a programming level, Itanium and Alpha have similar restrictions
- there are few technical/programming impediments to porting applications between Alpha/Itanium

Do these differences matter to Developers?

- Do you work in:
  - MACRO-64
  - MACRO-32
  - Write Exception and Trap Handling Code
  - Write Device Drivers and Similar Hardware sensitive code

Sizing and Migration–

- sizing (speed and/or size) is quite application sensitive
- strategy – get smallest/cheapest system
- do science – DO NOT guess
- optimization may have substantial impact

The Challenges –

- Performance
- Effects on the margin
- Traps/Faults

Performance

- not linear with other architectures
- a challenge
- good code should be good elsewhere, not necessarily the converse

Effects on the margin

- local copies
- basic blocks
- code sensitivity

Traps/Faults

- PDP-11 – tens of instructions
- VAX – low hundreds of instructions
- Alpha – hundreds
- IA-64 – thousands

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Session Notes & Materials:
http://www.rlgsc.com/ieee/vermont/2005-09/codeportability.html