HP/Intel IA-64

http://developer.intel.com/design/ia64/index.htm

EPIC
- Explicitly Parallel Instruction Computer
- 64 bit architecture
- Based on HPL-PD

Advanced Features
- Full implementation of predicated execution
- Supports speculative loads
- Rotating registers
- Support for software pipelining
- Multimedia: MMX + HP MAX-2

Registers
- 128 64-bit General Purpose Registers (GR)
- 128 82-bit Floating Point Registers (FR)
- 64 1-bit Predicate Registers (PR)
- 8 64-bit Branch Registers (BR)
- 128 64-bit “Application Registers” (AR)

- GRs organised in a register stack - 32 register static subset plus 0 - 96 register stacked subset (GR0 is always 0)
- FPs organised into 32 register static subset and 96 register rotating subset (F0 = 0.0; F1 = 1.0)
- PRs organised into 16 register static subset and a 96 register rotating subset

- Stacked registers are used to support fast argument passing in procedure calls
- Rotating registers are used to support software pipelining of numerical loops
Predicated Execution

- Generally, all instructions that writes to the GR or FR have both speculative and non-speculative versions
- The PR checked in speculative instructions is known as the qualifying predicate (qp)
- PR0 is hardwired to 1

Compare instructions set the predicate registers
- A single compare instruction can set two predicates at a time, e.g. one for the true and one for the false condition
- It is possible to move to or from several the predicate registers in a single instruction

Instruction Issue

- Instructions (41-bits each) are grouped according to classes
- An instruction bundle is a collection of 3 instructions grouped according to fixed templates (5 template bits per instruction)
- An arbitrary number of bundles form an instruction group – separated by stop bit

Semantics: All instructions in an instruction group appears as if they read the register files all at the same time
- All results appears written simultaneously immediately “at” a stop bit

Instruction Issue

- In each cycle two bundles are issued to 9 ports by a dispersal network
- Issue occurs till a stop bit or resource oversubscription

Memory Hierarchy

- L1 – split data and instruction cache.
  - L1-D – 4-way associative, write-through, 2 parallel memory access per clock, hits are 2 clock cycles, 32-byte lines
  - L1-I – 4 way associative with 32 byte lines
  - Bandwidth to L2 – 32 bytes per clock
  - FP loads always bypass L1-D
Memory Hierarchy
- L2 – unified, on-chip, 6-way associative, write-back, 2 parallel memory access per clock, up to 8 outstanding misses, hits are 6 clock cycles (for integer), 9 cycles (for floating point), 64-byte lines
- L2 has its own pipeline
- Bandwidth to L3 – 16 bytes per clock

Memory Hierarchy
- L3 – unified, off-chip-on-package, implementation dependent, hits are 21 clock cycles (for integer), 24 cycles (for floating point)
- Bandwidth to memory – 2.1GB/sec

Semantic Gap with HPL-PD
- Procedure call semantics
- Register set – esp. rotating subset
- 32 bit vs 64 bits
- Differs in richness of instruction set

```plaintext
alloc r40 = ar.pfs, 2, 34, 2, 8
mov r41 = lo
mov r42 = r12

adds r12 = -32, r12

sub r35 = r33, r32
mov ar.ec = 0
mov pr.rot = 1 << 16
mov ar.fc = r35

mov r33 = r0
mov r35 = r0

.L3:
(p16) add r32 = r35, r33
(p16) adds r34 = 1, r35
br.ctop.sptk.many .L3

.L4:
c1rrb
```

```plaintext
mov r33 = r0
mov r35 = r0

.L3:
(p16) add r32 = r35, r33
(p16) adds r34 = 1, r35
br.ctop.sptk.many .L3

.L4:
c1rrb
```