Definition

Manycore Chip:
A chip having many small CPUs, typically statically scheduled and 2-way superscalar or scalar.

Manycore Accelerator:
[Definition only for this class.] A manycore chip in which each core has a vector FP unit.

These notes will describe Intel's manycore accelerator, Phi.

phi-2

Intel's Manycore GPU.

Larrabee

Initially a project to develop a GPU.

Plans to develop a GPU product dropped but design targeted at accelerator use.

Knights Corner

Code name for Intel's planned manycore accelerator product.

Like Larrabee but not intended for graphics use.

Knights Ferry

Name of manycore accelerator Intel development package, including chip.

Specs match Larrabee closely.

phi-3

Xeon Phi

Name of Intel's manycore product.

Knights Landing

Version of Phi meant to operate without a companion CPU.

In notes will sometimes use term *Larrabee* to refer to all these designs.

phi-4

Overview

Chip consists of many cores.

Xeon Phi 7120A: 61 cores, 1.1 GHz. MSRP $4235.

Xeon Phi 7210: 64 cores, 1.3 GHz. (Released 2017). $\approx$ $5000

Xeon Phi 7250: 68 cores, 1.4 GHz. (Released 2017).

A Phi core is comparable with a CPU core or CUDA multiprocessor, **not** a CUDA core.

Each Core:

A 2-way superscalar, statically scheduled (in order) Intel 64 processor.

Four-way simultaneous multi-threaded (hyperthreaded).

The processor can issue instructions to a 512-bit vector unit.

Cores are complete enough to host an operating system.

phi-5

Storage Per Core

Context for four threads.

32 kiB of L1 data cache.

32 kiB of L1 instruction cache.

Share of coherent L2 cache: 512 kiB (Through model 72xx, 2017).

L2 cache is *coherent*: stores to cached data managed reasonably (short def).

Coherence is more precisely defined as meaning that the observed order of stores to a given location is the same for all processors.

A core can access any core's L2 cache, but access is fastest to its own core.

phi-6

Interconnection

Cores interconnected by a dual ring network.

Two rings, one in each direction.

Ring network used for messages to maintain coherence, among other purposes.

Width is 512 bits.

phi-7

Instruction Issue

Two instructions per cycle.

At most one of these can be a vector instruction.

Instruction Features

Each vector instruction can access up to two registers and one memory operand.

Similar to NVIDIA GPUs.

CPU arithmetic instructions cannot directly access memory. (Including both RISC instructions and IA-32 μops.).

One vector operand can be *swizzled* (lanes rearranged).

Vector memory operand can have type conversion applied.

phi-8

Important Instruction Set Features.

Memory Instructions

Prefetch

Cannot rely on massive multithreading to hide latency.

Cache Placement Hints

Addresses for vector load can come from a vector register.

Would still need one cycle per cache line.

phi-9

Instruction Operand Features

Swizzling

Source operand conversion.

Can convert memory source of an arithmetic insn from int types to float.

Supports efficient use of memory bandwidth.

Phi Overview

phi-10

Phi Chip

Multiple cores.

Each Phi Core

Has four thread contexts.

Is two-way superscalar.

32 kiB of L1 data cache.

32 kiB of L1 instruction cache.

phi-11

Phi Core ISA

Instruction Set

A subset of Intel 64 (x86_64).

Phi-specific vector instructions.

phi-11

phi-12

Phi Register Sets

Most Intel 64 registers.

There are 16 general-purpose (integer) registers.

Reg Names: `RAX`-`RDX`, `RDI`, `RSI`, `RBP`, `RSP`, `R8`-`R15`.

Phi-specific vector registers.

There are 32 vector registers, each 512 bits (64 bytes).

Assembler names: `zmm0` to `zmm31`.

Phi-specific mask registers.

There are 8 mask registers, each 16 bits.

Assembler names: `k0` to `k15`.

phi-13

Phi Vector Arithmetic Instructions

Common Features

Two or three source operands.

One source operand can come from memory.

Mask used to control which lanes of dest are written.

phi-14

Two-Register Format:

```
vaddps zmm2, S_{f32}(zmm3/m_t), zmm1{k1}
       ----  ---------------- --------
       Src1  Src2             Dest
```

Warning: Assembler conventions vary: Sometimes destination appears first.

Argument Types

`Dest`: Vector register `zmm1`, mask register `k1`.

Only lanes of `zmm1` corresponding to `k1` are written.

`Src1`: Vector register `zmm2`.

`Src2`: Vector or converted vector /memory.

phi-15

Instruction Naming Convention

```
vaddps zmm2, S_{f32}(zmm3/m_t), zmm1{k1},
122234 Dest  Src1               Src2
```

1 = v: Vector, uses vector (z) registers.

2 = add: Operation.

3 = p: Packed: Operate on vector elements individually.

4 = s: Single-precision.

phi-16

Examples:

```
  vaddpd    %zmm7, %zmm8, %zmm1          # zmm1 = zmm7 + zmm8

  vaddpd    %zmm2{badc}, %zmm2, %zmm3



  vaddpd    112(%rsp){1to8}, %zmm0, %zmm1{%k1}            #434.4 c35

  vfmadd231pd %zmm7, %zmm6, %zmm3                         #44.4 c77
  vfmadd231pd 2368(%rsi,%rcx,8), %zmm2, %zmm8             #52.4 c33
```

phi-17

Phi Pipeline

Pipeline Features

Two-way superscalar.

Most vector instructions have a four-cycle latency and one-cycle initiation interval.

Vector insn don't raise exceptions.

Four-cycle latency assumes dependence not into shuffle/conv stage.

```
Phi: 6-stage
 Scalar: Six stages
  PPF PF D0 D1 D2 E WB
 Vector 11-stage
```

phi-18

```
PPF PF D0 D1 D2 E VC1 VC2 V1 V2 V3 V4 WV
```

D0: Decode Intel 64 insn prefixes.

D1: Decode remainder of instruction.

D2: More decode, register read. (Microcode control.)

E: Execute.

VC1, VC2: Shuffle, load conversion.

V1-V4 FP execute.

phi-19

Prefetch Buffer

Sort of an L0 insn cache.

Fed in PPF stage.

Entry in buffer is 32 B (256 b).

Each thread has four entries.

phi-20

Phi Programming

Execution Modes

Native Mode:
Program starts and runs on Phi.

Offload Mode:
Program starts on CPU and executes code on Phi as needed.