High Performance Computing

CPU Pipeline

Martin Raum

Instruction execution

The execution of an instruction on the CPU is split up into micro-operations:

IF: instruction fetch,

ID: instruction decode,

EX: execute,

M: memory, register access;

An instruction can split up into many micro-operations.

Instruction pipeline

Execution of instructions is interleaved in pipelines, if dependencies make this possible.

One (simple) instruction is completed per clock cycle in each pipeline, once they are filled.

There are extra pipelines for floating point operations.

IF	ID	EX	M	M				
	IF	ID	EX	M	M			
		IF	ID	EX	M	M		
			IF	ID	EX	M	M	
				IF	ID	EX	M	M

Broken instruction pipeline

If for example the instruction to be executed depends on the result of a previous instruction, the pipeline breaks.

IF	ID	EX	M	M								
				IF	ID	EX	M	M				
								IF	ID	EX	M	M

Instruction latency and throughput

Like cache and memory, instructions exhibit latency. Floating point operations (divison, square root) can take significantly longer.

Instruction	Latency	Recip. throughput
MOV (r64,r64)	0	0.25
MOV (m,r64)	0–4	1
JMP (r)		2
ADD (r,r)	1	0.3
IMUL (r,r)	3	1
IDIV (r,r or m)	13–44	13–44
FSQRT	22	8
FSIN		50–170

Out-of-order execution and branch prediction

In order to compensate, for example,

- delays from cache and memory access
- the fact fact some instructions need more than one cycle to finish

CPUs reorder instructions.

This includes the possibly of statistically predicting the outcome of an instruction if it determines which instruction will be executed next.

Acquiring information about the CPU

In the terminal, one can list information about the CPU with the commands lscpu and cat /proc/cpuinfo.

Data sources

Technical data on chips can found at en.wikichip.org.

Instructions and their characteristics can be found
at www.felixcloutier.com/x86
and www.agner.org/optimize/instruction_tables.pdf.

The micro-architecture for many CPUs can be found
at www.agner.org/optimize/microarchitecture.pdf.