Exam Assignments 8

Naming Conventions for intrinsic Functions

`_<vector__size>_<operation>_<suffix>`

- `<vector_size>` specifies the size of the vector that the function returns:
  - `mm` means 128-bit vectors (SSE)
  - `mm256` means 256-bit vectors (AVX, AVX2)
  - `mm512` means 512-bit vectors (AVX-512)
- `<operation>` specifies the operation that the function will perform on the inputs; some examples:
  - `add`
  - `sub`
  - `mul`
- `<suffix>` specifies the data type of the input vectors
  - `ps` means `float`
  - `pd` means `double`
  - `ep<int_type>` stands for integers like `epi32` for 32-bit signed integers or `epu16` for 16-bit unsigned integers

Latency and Throughput

The **latency** of an intrinsic functions tells us how many cycles the CPU will take to execute the function.

The **troughput** of an intrinsic function tells us how many cycles it takes for the cpu to start another call of this function.

Latency is especially important for the performance of a single call to an intrinsic function, while **throughput** lets us know how efficient many subsequent calls to the same intrinsic function will be.

Instruction-level Parallelism

Modern processors contain multiple ports that work independently of eachother and a scheduler that supplies each port with work.

Each port can perform some operations, and often two or more ports share some of the operations they can execute. Since each port works independently, multiple instructions can be performed at the same time, so long as there are open ports for the specified instruction types.

Loop Unrolling

Loop unrolling is a performance-enhancing technique to exploit instruction-level parallelism in which a loop that would normally perform one instruction per loop cycle gets transformed so that multiple instructions get performed instead.

This may allow the CPU to schedule multiple instructions on multiple ports, allowing them to execute in parallel.

IPC (Instructions per Cycle)

A high IPC-value indicates that an algorithm is utilizing the CPU well and making proper use of instruction-level parallelism, since (with a value greater than 1) multiple instructions are being executed in a single cycle.