# Chapter 4: Data-Level Parallelism in Vector, SIMD, and GPU Architectures

# Introduction

- SIMD architectures can exploit significant data-level parallelism for:
  - matrix-oriented scientific computing
  - media-oriented image and sound processors
- SIMD is more energy efficient than MIMD
  - Only needs to fetch one instruction per data operation
  - Makes SIMD attractive for personal mobile devices
- SIMD allows programmer to continue to think sequentially

# SIMD Parallelism

- Vector architectures
  - SIMD first used in these architectures
  - Very expensive machines for super computing
- SIMD extensions
  - Extensions made to mainstream computers
  - For x86 processors Multimedia Extensions (MMX), Streaming SIMD Extensions (SSE) and Advanced Vector Extensions (AVX)
- Graphics Processor Units (GPUs)
  - Used for processing graphics.
  - GPUs have their own memory in addition to the general purpose CPU and its memory

# Vector Processing

- A **vector processor** is a CPU that implements an instruction set containing instructions that operate on one-dimensional arrays of data called *vectors*.
- This is in contrast to a scalar processor, whose instructions operate on single data items.
- Vector machines appeared in the early 1970s and dominated supercomputer design through the 1970s into the 90s, notably the various Cray platforms.

# Vector Architectures

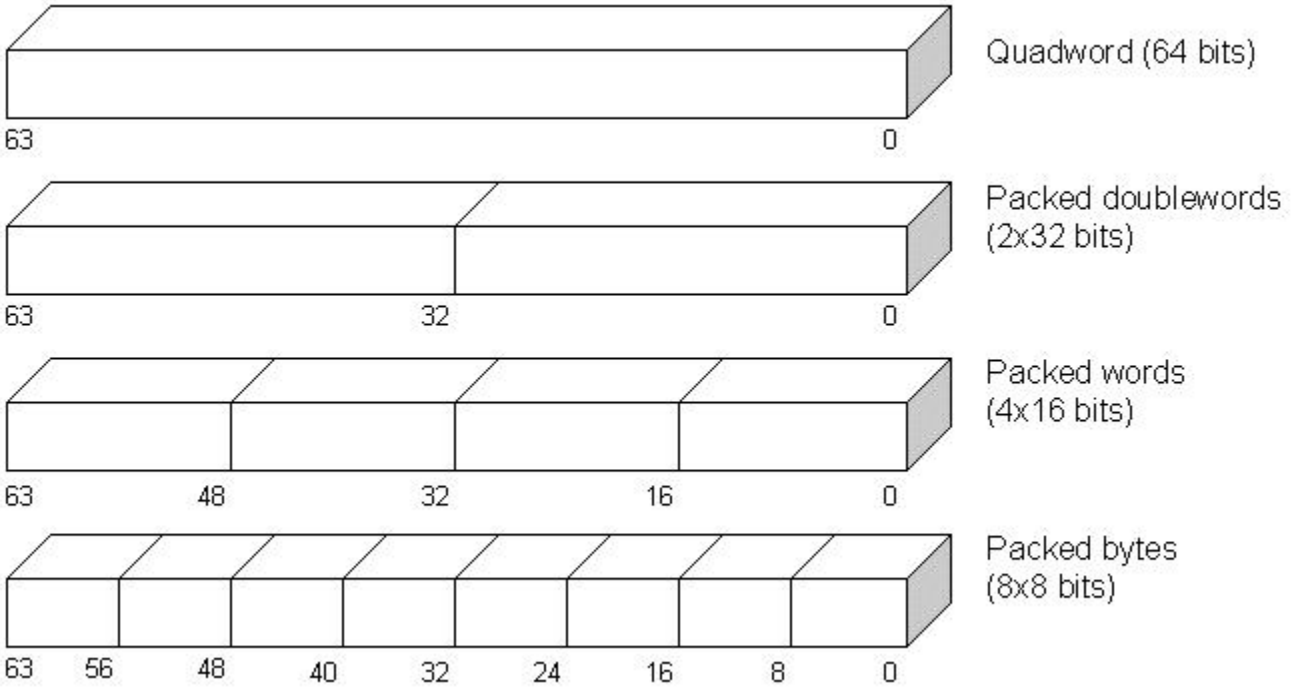
- Basic idea:
  - Read sets of data elements into “vector registers”
  - Operate on those registers
  - Disperse the results back into memory

# SIMD Extensions

- Media applications operate on data types narrower than the native word size

- Limitations, compared to vector instructions:
  - Number of data operands encoded into op code
  - No sophisticated addressing modes that vector processors were using.

# Intel MMX Technology



# SIMD Implementations

- Implementations:
  - Intel MMX (1996) (on 64 bit registers)
    - Eight 8-bit integer ops or four 16-bit integer ops
    - Same registers used for floating point
  - Streaming SIMD Extensions (SSE) (1999) (128 bits)
    - Eight 16-bit integer ops
    - Four 32-bit integer/fp ops or two 64-bit integer/fp ops
    - Separate registers for floating points and SIMD operations
  - Advanced Vector Extensions (2010) (256 bits)
    - Four 64-bit integer/fp ops
  - Operands must be consecutive and aligned memory locations

# Graphical Processing Units

- Provided to improve the performance of graphics in a system
- Computationally very capable
- Efforts are being made to use them for general purpose computing.
- Basic idea adopted by NVIDIA:
  - Heterogeneous execution model
    - CPU is the *host*, GPU is the *device*
  - Develop a C-like programming language for GPU i.e. Compute Unified Device Architecture or simply CUDA
  - Unify all forms of GPU parallelism as *CUDA thread*
  - Programming model is “Single Instruction Multiple Thread”