Computer Architecture

Quiz 4 → Guide to Reading Chapter 4

April 18, 2021

Abstract

The underlying idea of Chapter 4 is parallelism at the level of data.

1 Vocabulary

We gain power over ideas which we can associate with a word. Instead of being fleeting collections of activations of neurons, alone, named ideas can be recalled, communicated, built upon.

Thus, we need vocabulary. Provide a definition for:

- dynamic register typing – a property of a language that occurs when values are not associated with a certain type, allowing it to decide what type it should be at runtime based on how it is attempting to be used.

- loop carried dependence – occurs when a statement in an iteration of a loop relies on a statement in a different iteration of the same loop

2 Relationships

Describe some relationship between the two terms, or explain why they are not related:

- chaining in the context of vector processor pipeline vs. flexible chaining – these two are not related, chaining in vector processing refers to when **scalar** and **vector** registers yield results that can be used immediately, flexible chaining allows a **vector** instruction to chain to any other active **vector** operation as long as we don't generate a structural hazard

- convoy and chime – They are related, a convoy is the set of vector instructions that could potentially execute together and does not contain any structural hazards. Chime is simply the unit of time taken to execute a convoy.

- if (statement) conversion, vector mask register, GPU internal mask register – These terms are not directly related. If statement conversion is a term used by compiler writers to transform an if statement into a straight-line code sequence utilizing conditional execution, whereas vector mask registers are explicitly part of the architectural state and GPU internal mask register are internal to the hardware.

- thread block scheduler, SIMD thread scheduler – These terms are not directly related as they are two different types of schedulers, the thread block scheduler assigns multiple thread blocks to multithreaded SIMD processors, whereas SIMD thread scheduler is a hardware unit that schedules and issues SIMD instructions when they are ready to be executed.