ECE 420 Assignment 1

Arun Woosaree

February 7, 2021

1   Flynn's Taxonomy

1.1   SISD

SISD stands for Single Instruction Stream, Single Data Stream. A SISD system has a single uniprocessor, which executes a single instruction stream to operate on data stored in a single.

An example that would fit in this category is a single core superscalar processor like the AMD 29050 Reference: `https://en.wikipedia.org/wiki/Superscalar_processor`

1.2   SIMD

SIMD stands for Single Instruction Stream, Multiple Data Stream. A SIMD system has multiple processing units which all simultaneously do the same operation on multiple points in a data pool in a synchronized fashion.

An example that would fit in this category is a modern graphics processing unit, like the AMD Radeon 6900 XT

1.3   MISD

MISD stands for Multiple Instruction Stream, Single Data Stream. A MISD system has multiple processing units performing different operations on the same data simultaneously. It can be used for reliability or fault tolerance, if the same operation is being done multiple times on the same data.

An example of an MISD system is the computer responsible for the Space Shuttle flight control. The Space Shuttle is also known as the low orbit spacecraft that helped launch the Hubble telescope, among other space missions. References: `https://en.wikipedia.org/wiki/MISD#cite_note-1` `https://en.wikipedia.org/wiki/Space_Shuttle`

1.4   MIMD

MIMD stands for Multiple Instruction Stream, Multiple Data Stream. A MIMD system has many processing units which each perform operations independently

from each other at the same time. That is, each unit can be doing different operations on different pieces of data at any given time.

An example of a MIMD system is almost any modern multicore processor, like the AMD Ryzen 9 5950X. Another example could be a computer cluster, or a network of workstations.

2   Shared Memory vs Distributed Memory

2.1   Shared Memory

2.1.1   Pros

- user friendly programming perspective to memory

- data sharing between tasks is fast and uniform

2.1.2   Cons

- no scalability between memory and CPUs

- if you want more power,

2.2   Distributed Memory

2.2.1   Pros

- memory is scalable with the number of processors

- this is cost effective, because lots of cheap, commodity hardware can be used as opposed to upgrading a monolithic, more expensive structure

2.2.2   Cons

- data communication between processors is more difficult

- it is difficult to map existing data structures to the memory organization

- memory access times can be very different. For example, local access times vs over a network access times can be much slower compared to a MIMD system

- more overhead for the programmer. The programmer has to think about message passing