Name: A. Shiva Surya Saran
Roll No.: BE-B 20

High Performance Computing

Assignment 1

Unit 1: Parallel Processing Concepts

1. Explain VLIW Processors along with its advantages and disadvantages.

Very long instruction word or VLIW refers to a processor architecture designed to take advantage of instruction level parallelism.

- Instruction of a VLIW processor consists of multiple independent operations grouped together.
- There are Multiple Independent Functional Units in VLIW processor architecture.
- Each operation in the instruction is aligned to a functional unit.
- All functional units share the use of a common large register file.

This type of processor architecture is intended to allow higher performance without the inherent complexity of some other approaches.

VLIW Compiler

- Compiler is responsible for static scheduling of instructions in VLIW processor.
- Compiler finds out which operations can be executed in parallel in the program.
- It groups together these operations in single instruction which is the very large instruction word.
- Compiler ensures that an operation is not issued before its operands are ready.

VLIW Instruction

- One VLIW instruction word encodes multiple operations which allows them to be initiated in a single clock cycle.
- The operands and the operation to be performed by the various functional units are specified in the instruction itself.
- One instruction encodes at least one operation for each execution unit of the device.
- So length of the instruction increases with the number of execution units
- To accommodate these operation fields, VLIW instructions are usually at least 64 bits wide, and on some architectures are much wider up to 1024 bits.

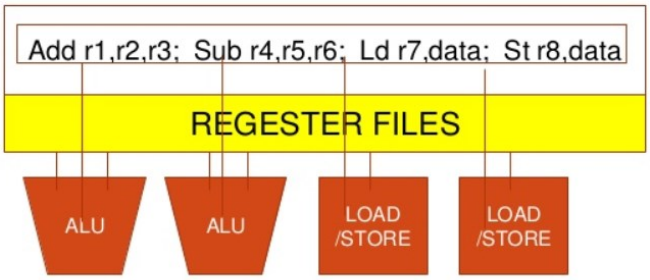


Fig. VLIW Instruction

Advantages of VLIW

- Dependencies are determined by compiler and used to schedule according to function unit latencies.
- Function units are assigned by compiler and correspond to the position within the instruction packet.
- Reduces hardware complexity.
- Tasks such as decoding, data dependency detection, instruction issues etc. becoming simple.
- Ensures potentially higher Clock Rate.
- Ensures Low power consumption

Disadvantages of VLIW

- Higher complexity of the compiler.
- Compatibility across implementations:
  Compiler optimization needs to consider technology dependent parameters such as latencies and load-use time of cache.
- Unscheduled events (e.g. cache miss) stall entire processor.
- Code density:
  In case of un-filled opcodes in a VLIW, memory space and instruction bandwidth are wasted i.e. low slot utilization.
- Code expansion:
  Causes high power consumption

Applications

- VLIW architecture is suitable for Digital Signal Processing applications.
- Processing of media data like compression/decompression of Image and speech data.

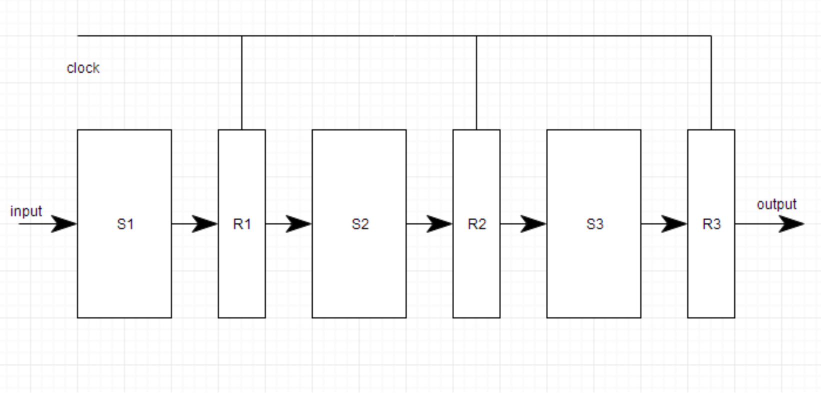
2. What are Pipelining and Superscalar Execution?

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing.

Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.

Pipelining increases the overall instruction throughput.

In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.



It is divided into 2 categories:

1. Arithmetic Pipeline
2. Instruction Pipeline

Superscalar Execution

Superscalar execution is attempting to have more instructions processed concurrently. Techniques that allow the pipelined processor work well also apply to superscalar processor.

Superscalar microprocessors can execute two or more instructions at the same time. E.g. typically, they have at least 2 ALUs (although a superscalar processor might have 1 ALU and some other execution unit, like a shifter or jump unit.)

They can start executing two or more instructions in the same cycle. Pipelined processors can execute more than one instruction at a time, but a non-superscalar pipelined processor will only start a single instruction in any given cycle. Pipelined execution units take multiple cycles to execute end to end. Put another way: superscalar processors are usually capable of executing two non-pipelined instructions with single cycle latency per cycle, whereas non-superscalar pipelined processors cannot have two single cycle instructions in execution in the ALUs at the same time.

3. *Explain Memory Latency Hiding Techniques.*

The increase in memory latency typically occurs when need arises to access Remote memory. E.g. Distributed Shared Memory based system. The important latency hiding techniques are as follows:

i. Using Prefetching techniques
   - The prefetching is either software/ hardware controlled.
   - In software-controlled prefetching, explicit “prefetch” instruction are issued for that data is known to be remote.
   - In hardware-controlled prefetching, it is done through use of long cache line to capitalize on spatial locality or through the instruction look ahead.
   - Prefetching technique is used for latency hiding because it brings instruction or data close to the processor before their actual requirement.
   - The direct effect of instruction of this scheme is that the time duration between the issues of the instruction and it’s actual references is increased. This has significant impact when latencies are large.

ii. Use of Coherent Caching technologies

iii. Relaxing the memory consistency requirements.

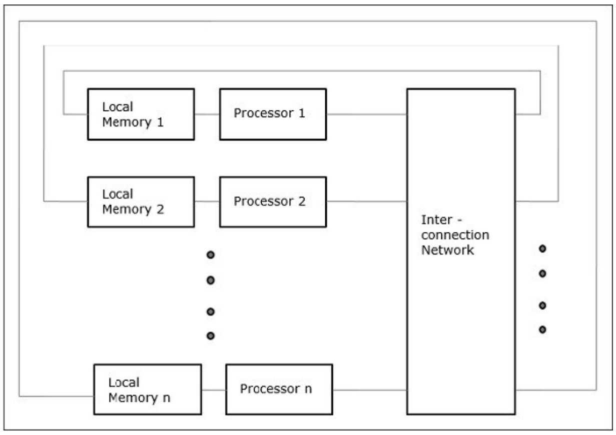
iv. Using multiple contexts to hide latency.

4. *Write a short note on:*

a) NUMA Multicomputer

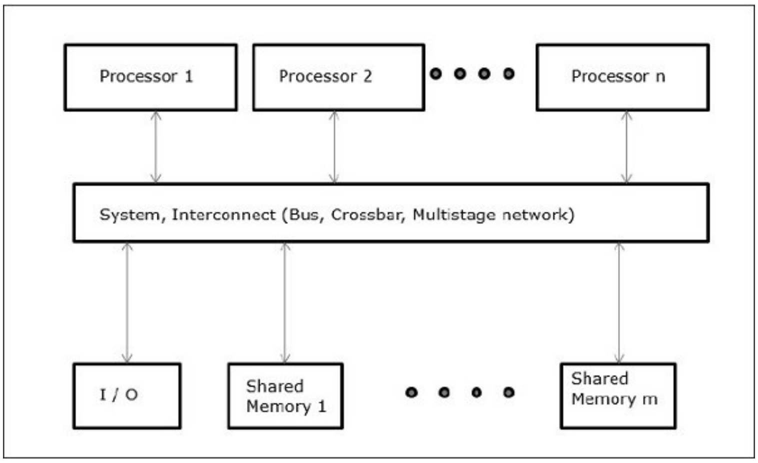
NUMA (non-uniform memory access) is a method of configuring a cluster of microprocessor in a multiprocessing system so that they can share memory locally, improving performance and the ability of the system to be expanded.

NUMA is used in a symmetric multiprocessing (SMP) system. NUMA adds an intermediate level of memory shared among a few microprocessors so that all data accesses don't have to travel on the main bus.



b) UMA Multicomputer

In this model, all the processors share the physical memory uniformly. All the processors have equal access time to all the memory words. Each processor may have a private cache memory. Same rule is followed for peripheral devices. When all the processors have equal access to all the peripheral devices, the system is called a symmetric multiprocessor. When only one or a few processors can access the peripheral devices, the system is called an asymmetric multiprocessor.



5. *Explain Snoopy cache protocol & Directory based schemes.*

Snoopy Protocols

- Write Invalidate Protocol:
  - Multiple readers, single writer
  - Write to shared data: an invalidate is sent to all caches which snoop and invalidate any copies
  - Read Miss:
    - Write-through: memory is always up-to-date
    - Write-back: snoop in caches to find most recent copy

- Write Broadcast Protocol:
  - Write to shared data: broadcast on bus, processors snoop, and update copies
  - Read miss: memory is always up-to-date

- Write serialization: bus serializes requests
  - Bus is single point of arbitration

Implementing Snooping Caches

- Multiple processors must be on bus, access to both addresses and data.
- Add a few new commands to perform coherency, in addition to read and write.
- Processors continuously snoop on address bus
  – If address matches tag, either invalidate or update
- Bus serializes writes, getting bus ensures no one else can perform memory operation
- On a miss in a write back cache, may have the desired copy and its dirty, so must reply
- Add extra state bit to cache to determine shared or not
- Since every bus transaction checks cache tags, could interfere with CPU just to check:
  - solution 1: duplicate set of tags for L1 caches just to allow checks in parallel with CPU
  - solution 2: L2 cache that obeys inclusion with L1 cache

Directory Protocol

- Similar to Snoopy Protocol: Three states
  - Shared: 1 processors have data, memory up-to-date
  - Uncached (no processor has it; not valid in any cache)
  - Exclusive: 1 processor has data; memory out of-date

- In addition to cache state, must track which processors have data when in the shared state (usually bit vector, 1 if processor has copy)
- Keep it simple(r):
  - Writes to non-exclusive data => write miss
  - Processor blocks until access completes
  - Assume messages received and acted upon in order sent.
- No bus and don't want to broadcast:
  - – interconnect no longer single arbitration point
  - – all messages have explicit responses
- Terms:
  - – Local node is the node where a request originates
  - – Home node is the node where the memory location of an address resides
  - – Remote node is the node that has a copy of a cache block, whether exclusive or shared

6. Explain difference between SIMD and MIMD architecture.

SIMD	MIMD
SIMD stands for Single Instruction Multiple Data.	While MIMD stands for Multiple Instruction Multiple Data.
SIMD requires small or less memory.	While it requires more or large memory.
The cost of SIMD is less than MIMD.	While it is costlier than SIMD.
It has single decoder.	While it have multiple decoders.
It is latent or tacit synchronization.	While it is accurate or explicit synchronization.
SIMD is a synchronous programming.	While MIMD is a asynchronous programming.
SIMD is a simple in terms of complexity than MIMD.	While MIMD is complex in terms of complexity than SIMD.
SIMD is less efficient in terms of performance than MIMD.	While MIMD is more efficient in terms of performance than SIMD.