Unit 4: Pipeline Scheduling and Speculative Execution

A. Compiler Techniques to Explore Instruction Level Parallelism (ILP)

1. Types of approaches to achieve ILP
    a. Compiler-based Static (brief)
    b. H/W based Dynamic (brief)
2. Formula to calculate Pipeline CPI
3. Parallelism limitations within basic block
    a. Definition of basic block
4. Data dependence
    a. Name dependence
    b. Antidependence
    c. Output dependence
    d. Control dependence
5. Compiler techniques for exposing ILP
    a. Pipeline Stalls
    b. Pipeline Scheduling
    c. Loop Unrolling (steps, limitations)
    d. Strip Mining

B. Dynamic Scheduling to explore ILP

1. Dynamic scheduling: Definition, advantages, disadvantages
2. Working process of dynamic scheduling
3. Register renaming to solve WAR and WAW hazards
4. Tomasulo's Algorithm
    a. Steps
    b. Framework:
        i. Components
        ii. Reservation stations
        iii. Register Status Indicator (RSI)
        iv. Common Data Bus
    c. Examples
    d. Illustration
    e. Tomasulo's algorithm with loop-example
5. Load-store order conflict

C. Speculative execution

1. Techniques to overcome control hazards in Tomasulo's approach
    a. Speculate
    b. Instruction commit
    c. Reorder Buffer (ROB)
    d. Modify reservation stations
2. Hardware based speculation
    a. Dynamic branch prediction
    b. Speculation
    c. Dynamic Scheduling
3. Reorder Buffer (ROB)

a. Structure of ROB (Fields)
b. Comparison of Tomasulo's algo and Tomasulo's algo with ROB (comparison through figure)
c. Operations with ROB
    i. Issue
    ii. Execute
    iii. Write result (complete)
    iv. Commit

Unit 5: Superscalar Processor and GPU Architectures

A. Advanced Pipelining and Superscalar Processor

1. Advanced Pipelining
    a. Superpipelining
    b. Multiple Issue (Super scalar)
        i. Statically scheduled scalar processor
        ii. VLIW (Very Large Instruction Word) processor
        iii. Dynamically scheduled superscalar processor
    c. VLIW
        i. Detail
        ii. Disadvantages
        iii. Example
    d. Extreme optimization- through combination of dynamic scheduling + multiple issues + speculation
    e. Handling multiple issues
        i. Example without speculation
        ii. Example with speculation
2. Superscalar processor
3. Multithreading
    a. Types of multithreading
    b. Hyperthreading
    c. Comparison of resource utilization

B. GPU Architectures
    1. What is a GPU?
    2. Requirement of GPU
    3. Shader programs
    4. CPU-GPU Interaction
    5. CPU vs GPU
    6. Flyn's Classification
        a. SISD/ Uniprocessor (Single Instruction, Single Data stream)
        b. SIMD/ Vector processing, Parallel Processing (Single instruction, multiple data)
        c. MISD/ Pipelined computers
        d. MIMD/ Multi-computers, Multi-processor
    7. Exploiting parallelism/ Vectorization

a. Vectorization
    i. Vector machine/ SIMD
b. Programming and execution model
    i. Model 1: Sequential (SISD)
    ii. Model 2: Data Parallel (SIMD)
    iii. Model 3: Multithreaded
c. GPU as SIMT machine
    i. Definition
    ii. SIMT illustration
        1. Warps
        2. Multithreaded warps
        3. Warp-level FGMT
        4. SIMD execution unit
        5. Warp instruction level parallelism
        6. SIMT memory access

C. Case study of GPU architectures
1. GPU concepts
    a. CPU-GPU interactions
    b. GPU Kernels
        i. Grid
        ii. Block
        iii. Thread
    c. Thread scheduling
    d. Memory hierarchy in GPU
2. CUDA
    a. Defining CUDA
    b. Processing flow in CUDA with flow architecture
    c. Indexing and memory access
        i. Indexing and memory access: 1D grid
        ii. Indexing and memory access: 2D grid
3. Case study: NVIDIA Tesla- Architecture
4. Case study: NVIDIA Fermi- Architecture
5. Case study: NVIDIA GPU Series
6. Performance parameters
    a. Memory access
        i. Latency hiding
            1. Occupancy
        ii. Memory coalescing
            1. Uncoalesced memory access
            2. Coalesced memory access
            3. Choice of data structures used
        iii. Data reuse- Tiling
        iv. Shared memory back conflicts
            1. Resolving bank conflicts
    b. SIMD warp utilization: Divergence
        i. Vector reduction: Naïve mapping
        ii. Divergence free mapping

c. Data transfer between CPU and GPU
    i. Asynchronous transfers
        1. Define
        2. Overlap communication and computation

Unit 6: Cache Memory Principles

1. Introduction to Cache memory
    a. Basics of memory
        i. Pipelined RISC data path
        ii. Processor memory performance gap
        iii. Relationship of Caches and Pipeline
        iv. Role of memory
        v. Memory hierarchy
    b. Cache Memory
        i. Introduction, Principal of locality, Access patterns
        ii. Cache fundamentals
            1. Block/line
            2. Hit
            3. Miss
            4. Hit time
            5. Hit rate/miss rate
            6. Miss penalty
        iii. CPU-cache interaction
        iv. General organization of a Cache
            1. Organization diagram
            2. Addressing Caches
            3. Index and offset calculation
        v. Four Cache memory design choices
            1. Block Placement/ Cache Mapping
                a. Direct mapped
                    i. Accessing direct-mapped Caches
                b. Set associative
                    i. Accessing set-associative Caches
                c. Fully associative- brief
            2. Block Identification
                a. Block Identification- Direct mapped
                b. Block Identification- Set associative
                c. Block Identification- Fully associative
                d. Cache Indexing
            3. Block Replacement
            4. Write Strategy

2. Block replacement and write strategy
    a. Block replacement
        i. Block replacement algorithms

1. Random policy
2. First In First Out (FIFO)
3. Last In First Out (LIFO)
4. Least Recently Used (LRU)
5. Pseudo-LRU (PLRU)
6. Not Recently Used (NRU)
7. Least Frequently Used (LFU)
8. Re-Reference Interval Prediction (RRIP)
9. Optimal

ii. Look-aside vs look-through caches

b. Write strategy
   i. Write hits
      1. Write through
      2. Write back
   ii. Write miss
      1. Write allocate
      2. No-Write allocate
   iii. Write through Cache with No-Write allocation
   iv. Write back Cache with Write allocation
c. Types of Cache misses
   i. Compulsory
   ii. Capacity
   iii. Conflict

3. Numerical on Design Concepts in Cache Memory
   a. Cache block concepts
   b. Index and offset calculations
   c. Tag and data array Access
   d. MPKI- miss rate relation
   e. Block replacement algorithm example
      i. Through Pseudo LRU block replacement policy
      ii. LIFO block replacement policy
   f. Cache mapping