Chapter 07: Instruction–Level Parallelism—VLIW, Vector, Array and Multithreaded Processors …

Lesson 04:

Vector Processors

Objective

- To learn Vector processor
- To understand how it processes vector elements

Vector processors

Vector Processor

- A processor capable of processing vector (one-dimensional array) elements
- Uses single instructions for mathematical operations, for example, SUM *I*, *J*, *N* where *I* is at an array base address, *J* is at another array-base address, and *N* is the number of array elements

Sum operations for each vector element in one dimensional matrix

- Needs n sums of the elements
- K (r) = I (r) + J (r) for r = 0, 1, 2, …, up to (n–1)

Multiplication operation for each vector element in one dimensional matrix

- Needs n sums of the multiplicands
- $K = \sum I\,(r).\ J\,(r)$ for $r = 0, 1, 2, \ldots,$ up to $(n-1)$

Σ

Multiplication operation for each vector element in two dimensional matrix

- Needs n sums of the multiplicands
- $K\,(p, q) = \sum I\,(\mathrm{p, r}).\ J\,(\mathrm{r, q})$ for r = 0, 1, 2, …, up to (n–1)
- Where p = 0, 1, 2, …, up to (n–1)
- q = 0, 1, 2, …, up to (n–1)

Four operations Sum_int, Sum_flp, Mult_Int and Mul_flp in a vector processor

Operation1 Sum_int	Operation 2 BaseAddr1	Operation 3 BaseAddr2	Operation 4 length

Operation1 Sum_flp	Operation 2 BaseAddr1	Operation 3 BaseAddr2	Operation 4 length

Operation1 Sum_int	Operation 2 BaseAddr1	Operation 3 BaseAddr2	Operation 4 length

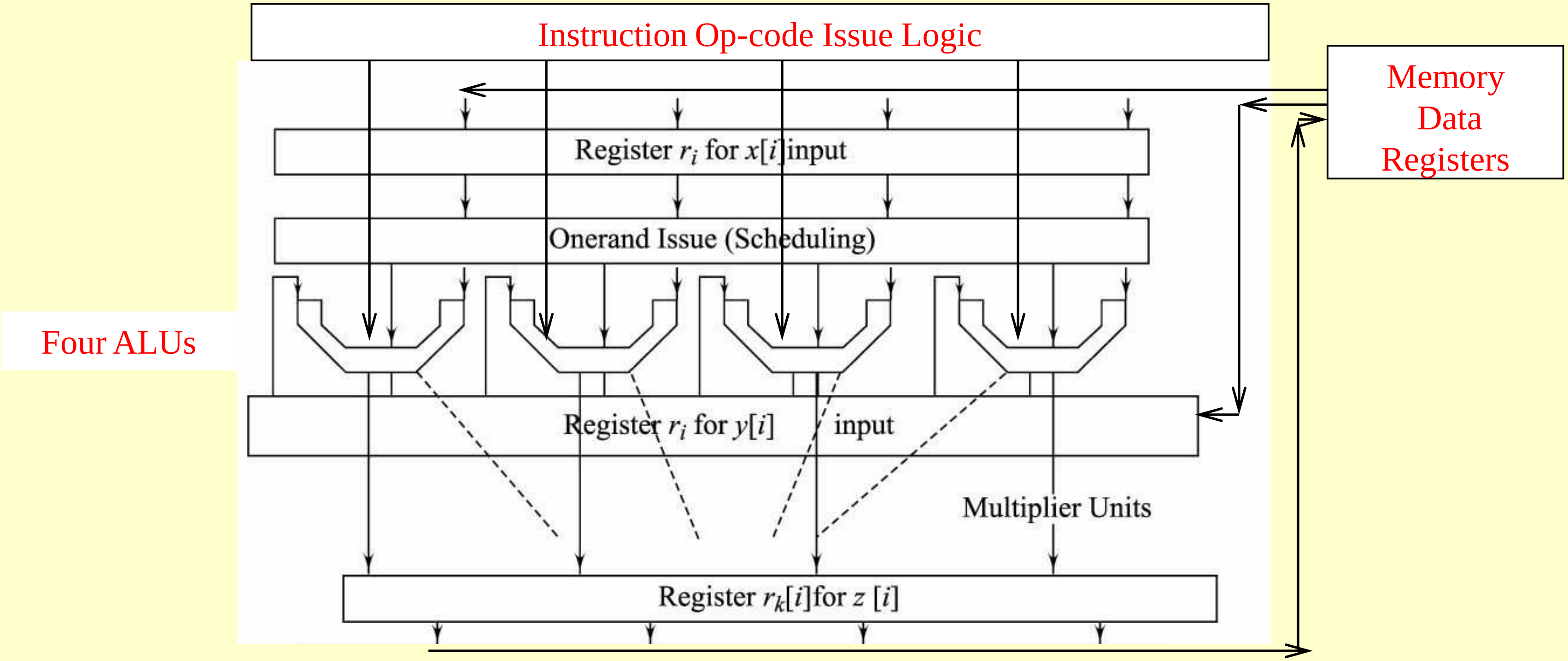
Operation1 Sum_flp	Operation 2 BaseAddr1	Operation 3 BaseAddr2	Operation 4 length

A design of a pipelined instructions parallelism in Vector Processor

- Summing Unit or multiplier unit pipelines are in Parallel Computing x[i] + y[i] or x[i] × y[i] in Number of pipeline stages
- Pipelines where x and y are fixed point integers or floating point numbers
- Length equals 4

Instructions Level Parallelism in Vector processors

Instructions Level Parallelism for Multiplier Unit x(i) × y(i)


Instruction Op-code Issue Logic
Memory Data Registers
Register r_i for $x[i]$input
Onerand Issue (Scheduling)
Four ALUs
Register r_i for $y[i]$ input
Multiplier Units
Register $r_k[i]$for z $[i]$

Vector Processor

- Floating point operations in vector processor can be implemented in the pipelines by the nature of their operations

Supercomputer (CRAY-1) vector processing features of 16-pipelines

- Refer Figure 7.8
- A set of pipelines scalar, second set for fixed point operations, third set for floating point operations, fourth for address computations

Memory Interleaving in Vector processing

Vector processing Requirement

- Vector processing requires array of memory addresses with each array storing the elements $r = 0, 1, 2, \ldots,$ up to $(n-1)$

Memory Interleaving in Vector processing

- Memory arrays are used to load input data to the scalar or vector or address register files and get output from the vector register files
- Memory arrays are used to store output data from the scalar or vector or address register files and get output from the vector register files
- Data of two inputs and output data interleaved

Example

- Vector A at addresses 0x00000,0x00004, 0x00008, 0x0000C, 0x00010, 0x00014, …
- Vector B at addresses 0x01000,0x01004, 0x01008, 0x0100C, 0x00010, 0x01014, …
- Output Vector C at addresses 0x02000,0x02004, 0x02008, 0x0200C, 0x02010, 0x02014, …

Example

- Interleaving means r(i) getting inputs from interleaved memory data registers 0x000, 0x008, 0x010 and 0x018 and r(j) inputs from 0x004, 0x0C, 0x14 and 0x1C

Summary

We Learnt

- Vector processor
- Processing of the vector elements of one dimensional array using operand issue logic
- Memory interleaving

End of Lesson 02 on

Vector Processors