SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME :19IT301 COMPUTER ORGANIZATION AND ARCHITECTURE

II YEAR /III SEMESTER

Unit 5: I/O ORGANIZATION AND PARALLELISM

Topic 7: Instruction Level Parallelism : Concepts and Challenges

ILP

- The simultaneous execution of multiple instructions from a program.

While pipelining is a form of ILP, the general application of ILP goes much further into more aggressive techniques to achieve parallel execution of the instructions in the instruction stream.

ILP

Two basic approaches:

1. rely on hardware to discover and exploit parallelism dynamically, and
2. rely on software to restructure programs to statically facilitate parallelism.

These techniques are complimentary. They can be and are used to improve performance

Dependencies and Hazards

- 3 types of dependencies:
- data dependencies (or true data dependencies),
- name dependencies, and
- control dependencies.

Examples of each dependence in ILP

1. Data Dependence

- Read After Write(RAW)

Instruction j tries to read operand before I Instrn writes it

I: add r1,r2,r3

J: sub r4,r1,r3

2.Anti-dependence

Instr J writes operand *before* Instr I reads it

I: sub r4,r1,r3

J: add r1,r2,r3

K: mul r6,r1,r7

3. Output dependence

InstrJ writes operand *before* Instr I writes it.

I: sub r1,r4,r3

J: add r1,r2,r3

K: mul r6,r1,r7

Data hazards

A hazard exists whenever there is a name or data dependence between two instructions and they are close enough that their overlapped execution would violate the program's order of dependency.

Possible data hazards:

RAW (read after write)

WAW (write after write)

WAR (write after read)

RAR (read after read) is not a hazard.

Parallel processing challenges and solutions

Technique	Reduces
Forwarding and bypassing	Potential data hazard stalls
Delayed branches and simple branch scheduling	Control hazard stalls
Basic dynamic scheduling (scoreboarding)	Data hazard stalls from true dependences
Dynamic scheduling with renaming	Data hazard stalls and stalls from anti dependences and output dependences
Dynamic branch prediction	Control stalls
Issuing multiple instructions per cycle	Ideal CPI
Speculation	Data hazard and control hazard stalls
Dynamic memory disambiguation	Data hazard stalls with memory
Loop unrolling	Control hazard stalls
Basic compiler pipeline scheduling	Data hazard stalls
Compiler dependence analysis	Ideal CPI, data hazard stalls
Compiler speculation	Ideal CPI, data, control stalls

Assessment

What is ILP?
What are the challenges?
What is data dependency?
What is output dependency?

Reference

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6th Edition 2012.