Data and Resource Dependence in Instruction Level Parallelism (ILP)

Dependencies are a property of programs

- Pipeline organization determines if dependence is detected and if it causes a stall
- Parallel processing is another significant concern

Data dependence conveys:

- Possibility of a hazard
- Order in which results must be calculated
- Upper bound on exploitable instruction level parallelism
- Dependencies that flow through memory locations are difficult to detect
- may be caused by “named register/resource reuse”
  … a name (only) dependence: it is not a true data dependence, but is a problem when reordering instructions

Control Dependence

- For a computational loop in a pipelined machine, ordering of instruction i with respect to a branch instruction
  - Instruction control dependent on a branch cannot be moved before the branch so that its execution is no longer controller by the branch
  - An instruction not control dependent on a branch cannot be moved after the branch so that its execution is controlled by the branch


Data Dependence

1.) Flow Dependence: Instruction output to instruction input signal flow
2.) Anti Dependence: Instruction output replaces a previous input
3.) Output Dependence: Instruction output replaces a previous output
4.) I/O Dependence: I/O or peripheral access, open file read then write or magnetic tape read then write
5.) Unknown: Dependence can not be determined
   Indirect addressing causes access to possibly conflicting location
   Loops with variable or multiple subscript indexes
Berstein’s Conditions

There are three conditions based on instruction input sets, \( I_i, I_j, \cdots \), and instruction output sets, \( O_i, O_j, \cdots \).

Instruction may be executed in parallel if the following three conditions are met:

1). \( O_i \cap I_j = \emptyset \) “Flow Dependence”

2). \( I_i \cap O_j = \emptyset \) “Anti Dependence”

3). \( O_i \cap O_j = \emptyset \) “Output Dependence”

Base on these concepts, instruction can be identified and a flow graph developed to visually map the dependences and develop allowed parallel instruction execution.


Data Hazards

Read after write (RAW) – related to flow dependence

Write after write (WAW) – related to output dependence

Write after read (WAR) – related to anti-dependence