

Overcoming data hazards

- Instructions are issued in program order, if an instruction is stalled in the pipeline, no later instructions can proceed.
- Functional unit could lie idle.
- In classic 5 stages, structural hazards and data hazards are checked in ID stage.
- Split into two parts, 1) checking for any structural hazards and waiting for the absence of a data hazard.
- In order issue, but execute as soon as data are available (out-of-order execution) and leads to out-of-order completion.

Problems

- But WAW and WAR could be introduced.
- Spilt ID into two
- Issue - Decode instructions, check for structural hazards.
- Read operands – wait until no data hazards, then read operands.

Tomasulo's Algorithm

- Invented by Robert Tomasulo (1967)
- used in IBM 360/91 fp unit
- avoid RAW hazards, by executing instructions only when its operands are available.
- WAR and WAW hazards are eliminated by register renaming.

Reservation Station

- Fetches and Buffers an operand as soon as available – no need to get it from register
- pending instructions designate the reservation station that will provide their input.
- Successive writes to a register overlap in execution, only the last one is actually used to update the register.
- As instructions are issued, the register specifies for pending operands are renamed to the names of the reservation station, which provides register renaming.
- There can be more reservation stations than register – reduce hazards from name dependencies.

Three Stages – Issue

- Issue
- get instructions from head of the instruction queue FIFO
- if matching empty reservation station, issue the instruction to the empty reservation station with the operand values, if available If no empty reservation station, structural hazard and the instruction stalls until a station or buffer is freed. if operand are not in registers, keep track of the functional units that will produce the operands. Register renaming is done in this step to eliminate WAR and WAW hazard.

Execute

- Execute
- if one (or both) operands not available, monitor the CDB (Common Data Bus) when an operand is avail, it is placed in the corresponding reservation when all operands avail, can execute
- Eliminate RAW
- Several instructions could become ready in the same cycle for fp ops choose arbitrarily

Load/Store

- Load/Store needs a 2-step
- 1. compute effective address when the base register available place it in load or store buffer. Load in the load buffer execute as soon as the memory unit is available. Stores in the store buffer wait for the value to be stored before being sent to the memory unit.

Load/Stores are maintained in program order through the effective address calculation, which help to prevent hazards through memory.

Write

- Write Result
- when result is avail write on CDB and from there to the registers and into any reservation stations (including store buffers) waiting for this result. Stores write to memory in this step.