A 1967 Secret Decoder Ring

- FLR = register
- CDB = result bus, completion bus
- sink = destination register
- FLB/SDB = load/store buffer
- FLOS = instruction window / dispatch window
What was wrong with score-boarding?

- Because of WAR and WAW
- (not different from scoreboard, but overcoming compiler cheezyness)
When can we execute an instruction?

- When all dependencies with all previous active instructions are known
Cardinal Precedence Principle

- No register may participate in an operation if it the destination of another, incomplete instruction
3 Things hardware must support..

- (1) Recognize a dependence
- (2) Cause the correct ordering
- (3) Recognize independence
What is a “reservation station”?

- Buffer for what?
  - For operands
  - For instruction
Mem -> Issue Window

• Wait for
  – Free entry in the window

• Do
  – Move instruction there
Issue Window -> Issue

• Waitfor
  – Free reservation station of the correct type

• Do
  – Move data or the tag to the reservation station
  – Set reservation station busy
  – Move the op code
  – Update management table of all reservation stations
  – Update destination register status and tag
Issue -> Execute

- Wait for
  - All operands are available
- Do
  - COMPUTE
Execute -> complete

- Wait for
  - Done executing
  - Need free result bus

- Do
  - Allocate bus (make sure no one else does!)
  - Write result
  - Free up reservation station
    - Turn off busy signal
Advantages

- Distribution
- Avoid WAW and WAR
- Add physical registers without changing the instruction set (by adding logical registers)
- Dynamic loop unrolling
Disadvantages

- Hardware complexities
- Not scalable, large busses
- Interrupts are a pain
O(n) structures/functions
$O(n^2)$ structures/functions
Space Complexities

• Tomasulo
  – Check of RAT $O(R)$
  – Check of Reservation-Queue is $O(F*R)$
  – Send result to completion bus $O(C*F)$
  – Completion bus(s) to reservation station $O(C*R)$
  – Complexity of Load/Store buffer $O(AS*L)$

• Multiple-In-Order-Issue Tomasulo
  – Check of RAT $O(M^2 + M*R)$
Load/Store

• Total Load-Store ordering
  – All loads are dependent on all previous stores
  – All stores occur after all previous loads
  – All stores are dependent on all previous stores

• Shorter version
  – All loads are dependent on the previous store
  – A store is dependent on the previous store
  – A store is dependent on all loads between it and the previous store