EE 457 Unit 9b

In-Order Completion

Speculation

Credits

- Some of the material in this presentation is taken from:
  - Computer Architecture: A Quantitative Approach
    - John Hennessy & David Patterson
- Some of the material in this presentation is derived from course notes and slides from
  - Prof. Michel Dubois (USC)
  - Prof. Murali Annavaram (USC)
  - Prof. David Patterson (UC Berkeley)



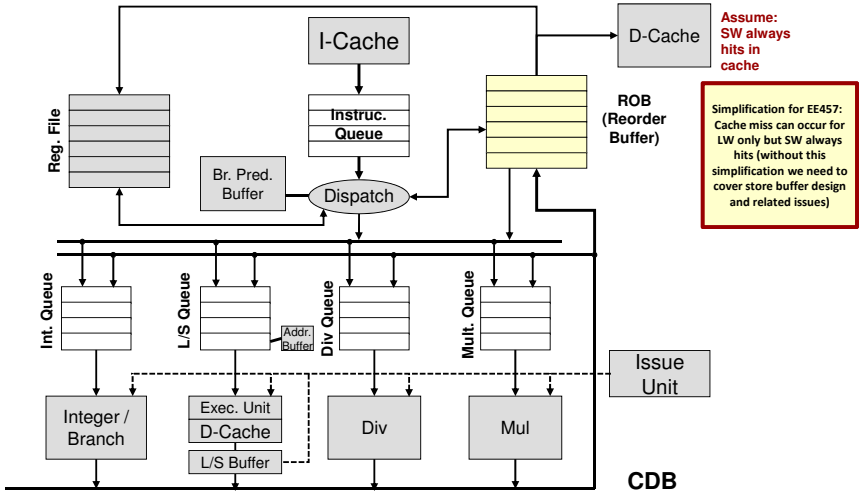
Tomasulo w/ Speculative Execution

- In-order Issue
- Out-of-Order Execution
- _____ Completion
  - Completion = Commit = Graduation

This is the Summer 2009 EE 560 project.
The Summer 2010 and later project is more advanced and uses physical register file, free register list, copy-free check-pointing, etc.

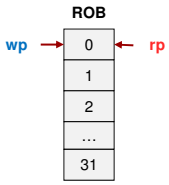
OoO Execution w/ ROB

- ROB allows for OoO execution but in-order completion



Re-Order Buffer (ROB)

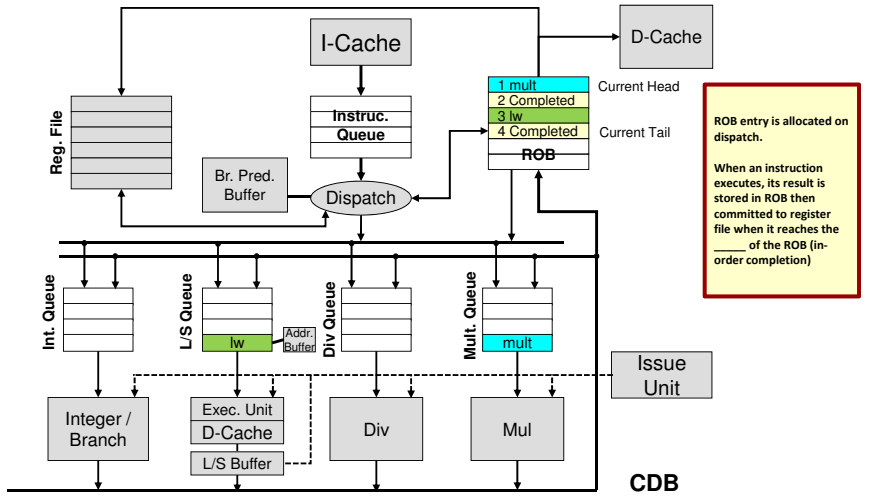
- ROB is a _____ (let's say 32 locations)
  - WP = Write pointer = Used by _____ Unit
    - Each instruction issues in order and "takes a number"
  - RP = Read pointer = Used for _____ the most senior / oldest instruction when it has completed without generating an exception



1. _____ = number of items in the FIFO (depth)
2. It is a circular FIFO/buffer

OoO Execution w/ ROB

- ROB allows for OoO execution but in-order completion



ROB entry is allocated on dispatch.

When an instruction executes, its result is stored in ROB then committed to register file when it reaches the _____ of the ROB (in-order completion)

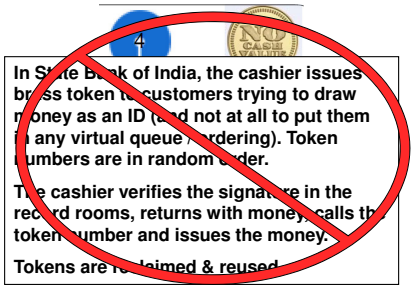
Dispatch and the ROB

- No more token FIFO (for tagging instructions) as in OoO execution and completion
  - ROB entry is _____ for an instruction on _____
  - When instruction finishes executing its result is buffered in the ROB entry until it can be _____
- It does not (and cannot) use the RST (Register Status Table) as before
  - When an instruction is dispatched, the ROB is searched for its source register (Rs and/or Rt) producers
    - If an entry in the ROB is producing Rs/Rt but has not yet executed the _____ of the producer is taken with the dependent instruction
    - If an entry in the ROB is producing Rs/Rt and the result is there waiting to be committed, _____ is taken with the dependent instruction
    - If no entry in the ROB is producing Rs/Rt, data in the _____ is taken with the dependent instruction
    - Since _____ in the ROB may match Rs/Rt a priority resolver is necessary

Take a Number vs. Take a Token

- ROB forms a virtual queue!
- ROB Tag = Paper token taken by the customer
  - Recall that we wrap back to 0 after the maximum tag number

Helps to create a virtual queue.



In State Bank of India, the cashier issues brass token to customers trying to draw money as an ID (and not at all to put them in any virtual queue / ordering). Token numbers are in random order.

The cashier verifies the signature in the record rooms, returns with money, calls the token number and issues the money.

Tokens are reclaimed & reused

Example 1

- Case 1
  - Your number is 55 and mine is 65
  - I am _____ numbers (**after / before**) you.
- Case 2
  - Your number is 55 and mine is 45
  - I am _____ numbers (**after / before**) you.

Assume now serving customer 52

Computing Distance

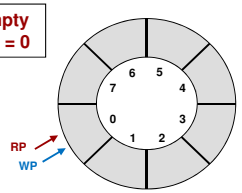
- To find how many people are waiting subtract the "Now Serving" number from the last number pulled
- Example
  - Last number pulled = 92
  - "Now Serving" = 52
  - # Waiting = _____
- But suppose the last number pulled is 32
  - Last number pulled = 32
  - "Now Serving" = 52
  - # Waiting = _____

Assume now serving customer 52

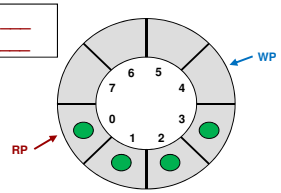
Computing Distance

- Depth = (WP – RP) mod 8

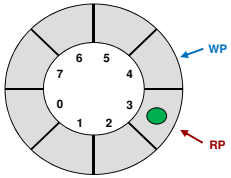
FIFO Initially Empty
D = WP-RP = 0-0 = 0



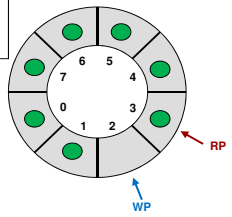
FIFO Depth = ___
= WP-RP = _____



FIFO Depth = __
D = WP-RP = _____



FIFO Depth = __
= WP-RP
= _____



ROB Dispatch for Rs

- $2 is needed by dispatch
- Which entry should be selected by you (the ROB)?

Scenario 0

Top (rp) →

Bottom (wp) →

	Valid	Rd	RegWrite
0	0	0	1
1	0	$2	1
2	0	0	0
3	1	$1	1
4	1	$2	1
5	1	$15	1
6	1	$2	1
7	1	$12	1
8	1	$2	0
9	1	$7	0
10	0	$13	1
11	0	0	1
12	0	$4	0
13	0	$2	1
14	0	0	1

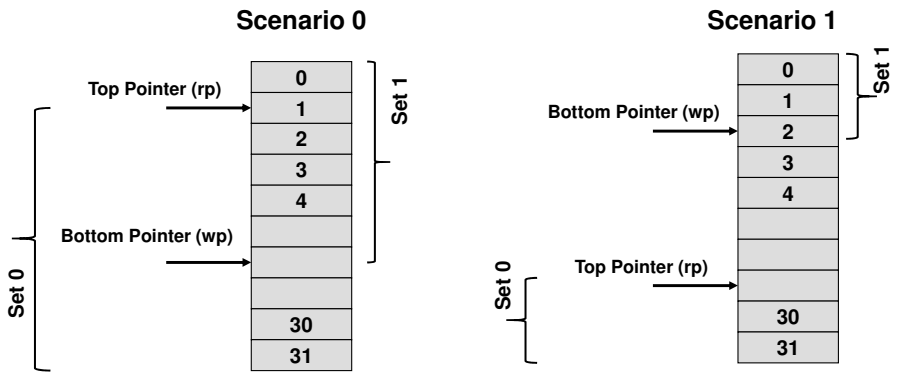
Scenario 1

Bottom (wp) →

Top (rp) →

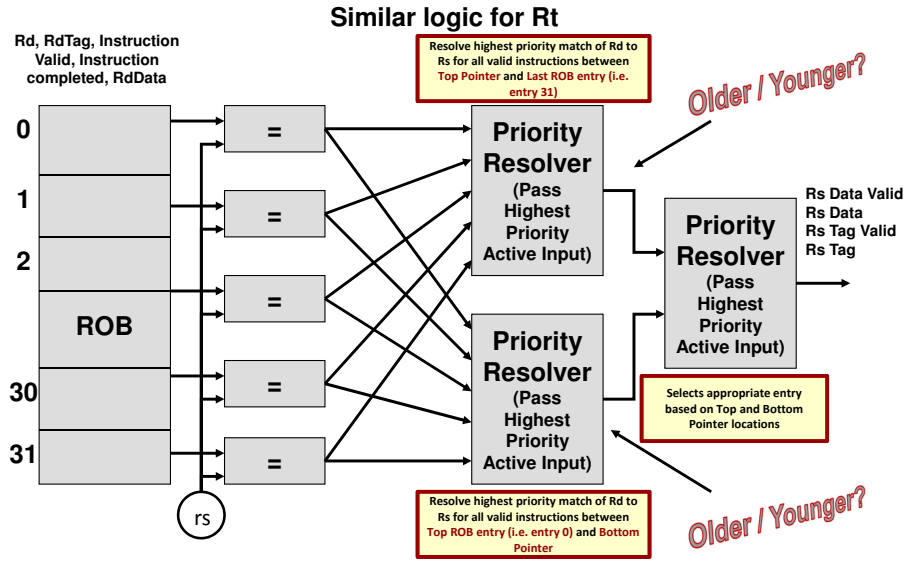
	Valid	Rd	RegWrite
0	1	0	1
1	1	$2	1
2	1	$10	1
3	0	$1	0
4	0	$21	1
5	0	$12	1
6	0	$2	0
7	0	$15	1
8	0	$22	1
9	1	$7	1
10	1	$13	0
11	1	$2	1
12	1	$1	1
13	1	$2	0
14	1	$3	1

Dealing with Wrapping

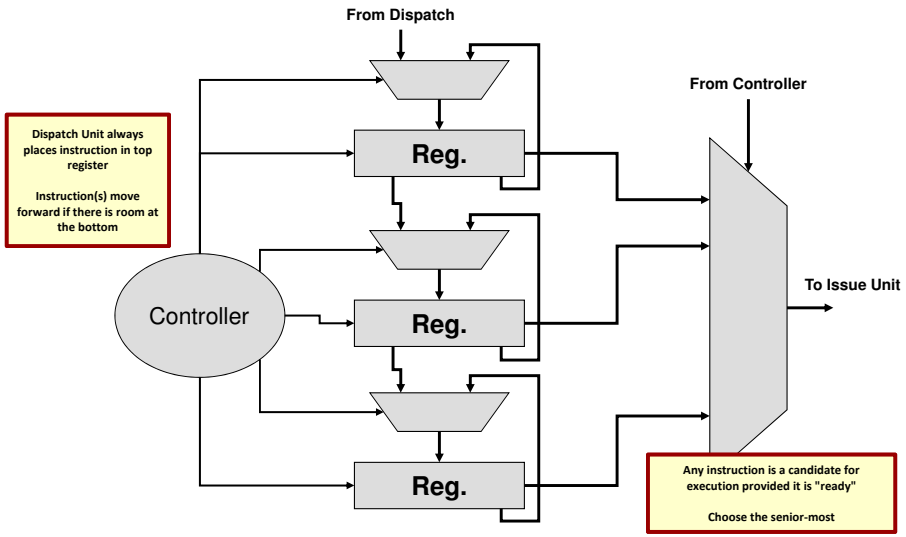


In each scenario, which set should be given higher priority of selection to forward the value of a particular register?

ROB Dispatch for Rs

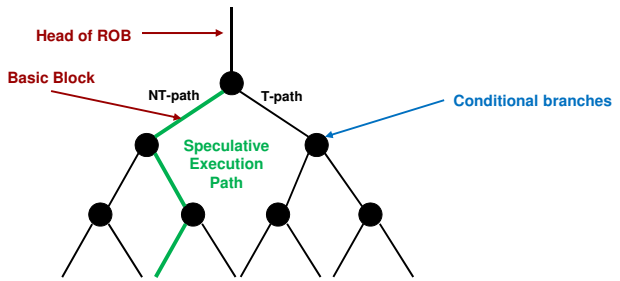


Issue Queues



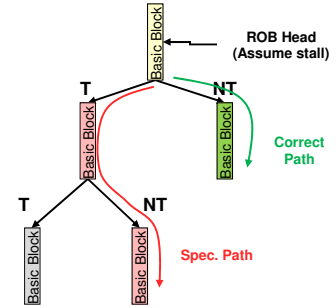
Branch Prediction + Speculation

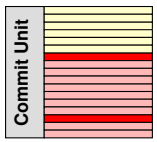
- To keep the backend fed with enough work we need to _____ a branch's outcome and perform _____ execution beyond the predicted (unresolved) branch
  - Roll back mechanism (flush) in case of misprediction



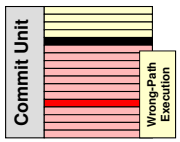
Speculation Example

- Predict branches and execute most likely path
  - Simply flush ROB entries after the mispredicted branch
  - Need good prediction capabilities to make this useful

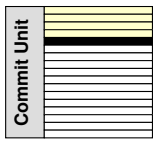




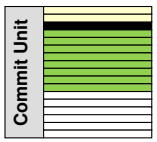
Time 1: ROB
Red Entries = Predicted Branches



Time 2a: ROB
Black Entry = Mispredicted branch



Time 2b:
Flush ROB/Pipeline of instructions behind it



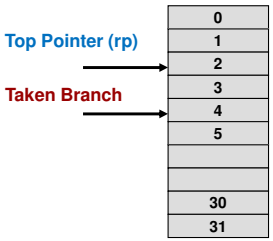
Time 3: ROB
Pipeline begins to fill w/ correct path

Handling Jumps and Branches

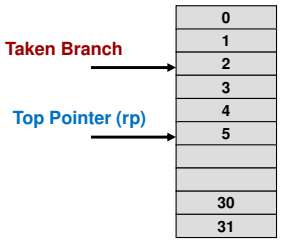
- IFQ is flushed every time a jump instruction enters the dispatch unit
- When a branch enters the dispatch unit, branch prediction is performed using the BPB (Branch Prediction Buffer)
  - Last n (e.g. 3) bits of PC are used by the branch predictor
  - Branches are handled aggressively
    - Executed as soon as they arrive on the CDB without waiting for instruction to become the _____ so as to determine if prediction was correct and take appropriate action
    - Selective flushing mechanism is used to flush instructions in backend in case of mispredicted branch

Flushing Mechanism

- In order to flush instructions in the backend a 'flush' signal along with the following are conveyed to the backend
  - Current Top of ROB
  - Depth of the Branch Instruction
- All instructions in the backend (as well as the ROB) with depth greater than the successful branch need to leave (be flushed)



Depth = 2 = (4-2)



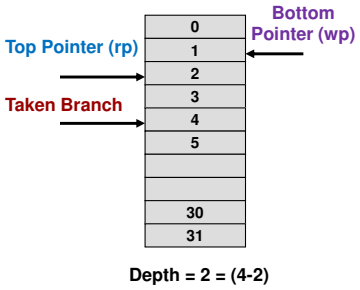
Depth = 29 = (2-5) mod 32

Selective Flushing for Branch Misprediction

- Paper token analogy
  - Say the store is going to close in 20 min. and they noticed too many people are waiting
  - They may announce that they will serve up to token #72 and people having tokens after that may leave now
- If the last token pulled is 92, then people with tokens #73 to #92 will leave
- If the last token pulled is #32, then people with tokens _____ and _____ will leave
- Because of the circular nature of the tokens/ROB FIFO mechanism, one cannot simply compare his token with #72 to decide whether to stay or leave
- Leave if you are more than 20 people away from current person being served (i.e. #52)

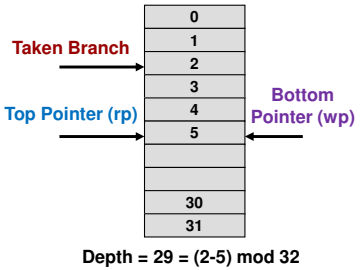
Selective Flushing for Branch Misprediction

- Anyone with greater depth (distance from top pointer to mispredicted branch) than the branch should leave
- Suppose the bottom (WP) is at 1
  - Is it (ROB) full? Yes / No
  - Total Populated Area = 1 / 31 / 32
- Who should leave (be flushed)?
  -
  - Note: #1 is empty



Selective Flushing for Branch Misprediction

- Who should leave in this scenario?
  -



RAW Hazard Refresher

- Recall, RAW hazard for registers was handled by
  - Dependent instructions are given the ROB tag of their specific senior to wait on in the backend
  - When the specific senior comes on the CDB an announces the value, then the dependent instruction grabs the value
  - Once the dependent instruction has all its sources, it raises his hand to say, "I am ready to go the execution unit" and waits for the issue unit to grant permission
- WAR and WAW are handle via ROB tags and In-Order Completion

RAW, WAR, WAW for Memory

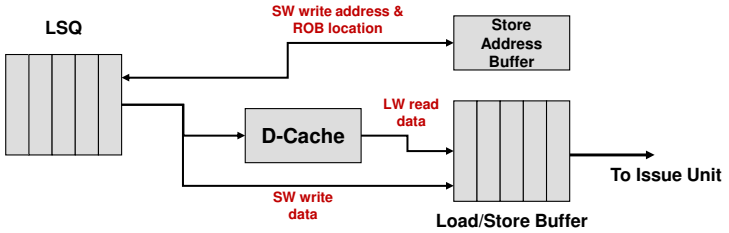
- WAR and WAW hazards are handled through In-Order Completion
  - R = Read = LW (load word)
  - W = Write = SW (store word)
- An 'LW' reads cache in the execution unit before going to ROB
- An 'SW' writes into cache (i.e. commits) when it reaches the "top" of ROB (meaning it became the senior-most)

LW Issuing

- To handle RAW properly an LW must wait in LSQ until:
  - It knows its _____
  - All _____ know their write address (SW may be waiting on some earlier instruction for its write address)
- Then either
  - Read data from cache if no _____ are in the _____ or _____…OR…
  - Get data directly from _____ (_____ of those SW's with matching addresses) out of the _____
- We use a "Store Address Buffer" to maintain a record of the write addresses which can be used for fast comparison and prioritization to find matches and the "youngest" of the "oldest"

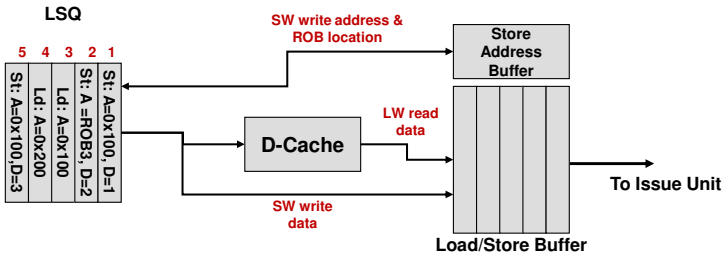
Load Store Queue

- LW accesses memory and result is written into Load/Store Buffer
- SW _____ access memory while getting issued from LSQ and goes to Load/Store buffer directly
- Whenever SW issues, its _____ and _____ are stored in the Store Address Buffer
- Once an SW is committed from the top of the ROB its entry in the Store Address Buffer is cleared



Load Store Queue

- Can Ld @ 3 issue before St @ 1 or 2?
  -
- Can Ld @ 4 issue before St @ 1 or 2?
  -

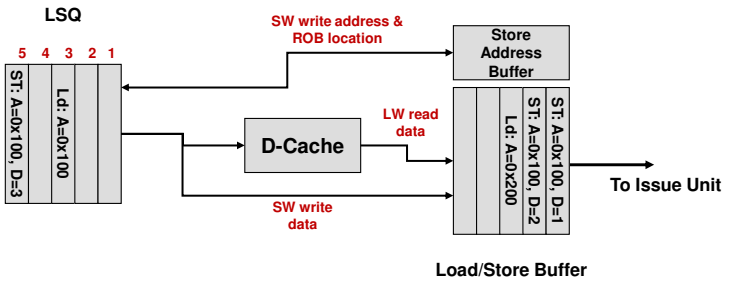


Issuing LW and SW

- In fact, even _____ SW's are allowed to bypass a waiting senior LW but the LW will _____ of how many bypassing SW's matched its address
- When LW issues it can grab the appropriate _____ of its data by counting _____ from matches in the store buffer

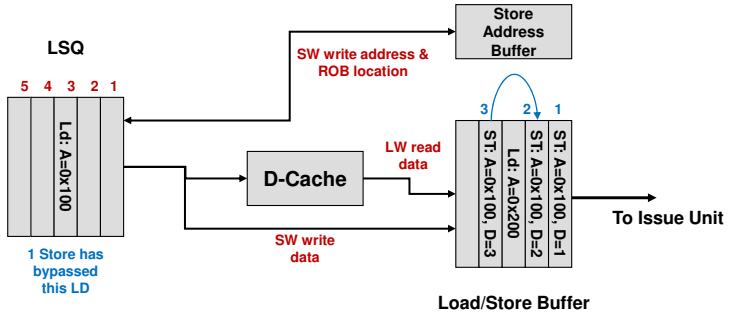
Load Store Queue

- Can St @ 5 bypass the Ld @ 3
  - Yes, using the mechanism just described. Load can count how many junior Stores bypassed it and count back that many in the Store buffer to attain the correct data value



Load Store Queue

- Can St @ 5 bypass the Ld @ 3
  - Yes, using the mechanism just described. Load can count how many junior Stores bypassed it and count back that many in the Store buffer to attain the correct data value

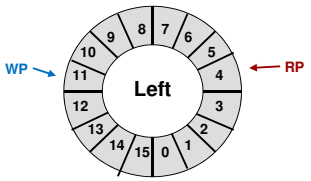


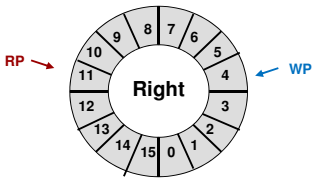
Store Word Issuing

- An SW can issue when its
  - Write _____ is known
  - Write _____ is known
  - No LW in front of it has an _____
    - Because LW won't be able to keep track of the count of how many matching SW's bypassed it

Spring 2011 Final Exam Question

- In the illustrations below, the ROB is (more / less) than half-full in the left case and is (more / less) than half-full in the right case. The left has ____ locations occupied and the right has ___ locations occupied. WP is (always / sometimes / never) ahead of RP.





Spring 2011 Final Exam Question

- If the instruction with ROB Tag 11 is found to be a mispredicted branch, which instructions with what ROB tags would you flush?

- And would you adjust RP or WP or both? And to what value(s)?