ECE511 Computer Architecture **29 Sept 2004**

Lecture 10 "ILP IV"

Lecturer: MattFrank

Hw#1 Discussion

- **If the L2 Cache gave a 9x performance increase that means:**
  - **8 out of 9 cycles were stalls waiting for DRAM**
  - **(i.e. Oracle is an application that has really bad spatially locality)**

- **When simulating small working sets make sure to be careful on assumptions.**

<u>Spatial Locality</u>: this concept was very important with regards to increasing the line size of the cache. Increasing the line size allowed lower cycle times and greater cache hits.

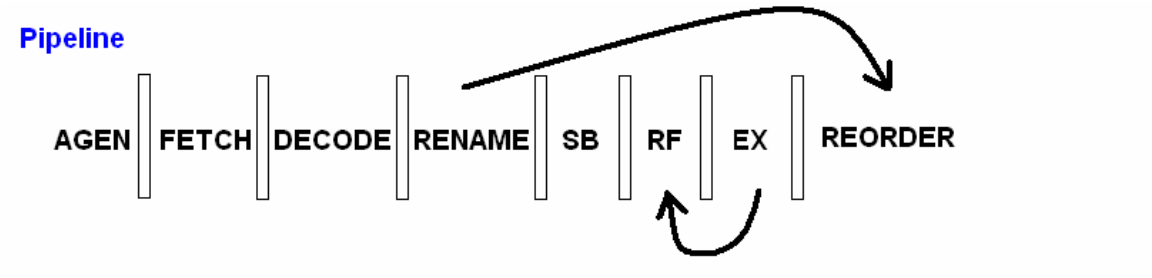
- **'GCC' compiler puts data into lists and trees, data structures that favor spatial locality.**

<u>Inclusive & Exclusive Caching Hierarchies</u>

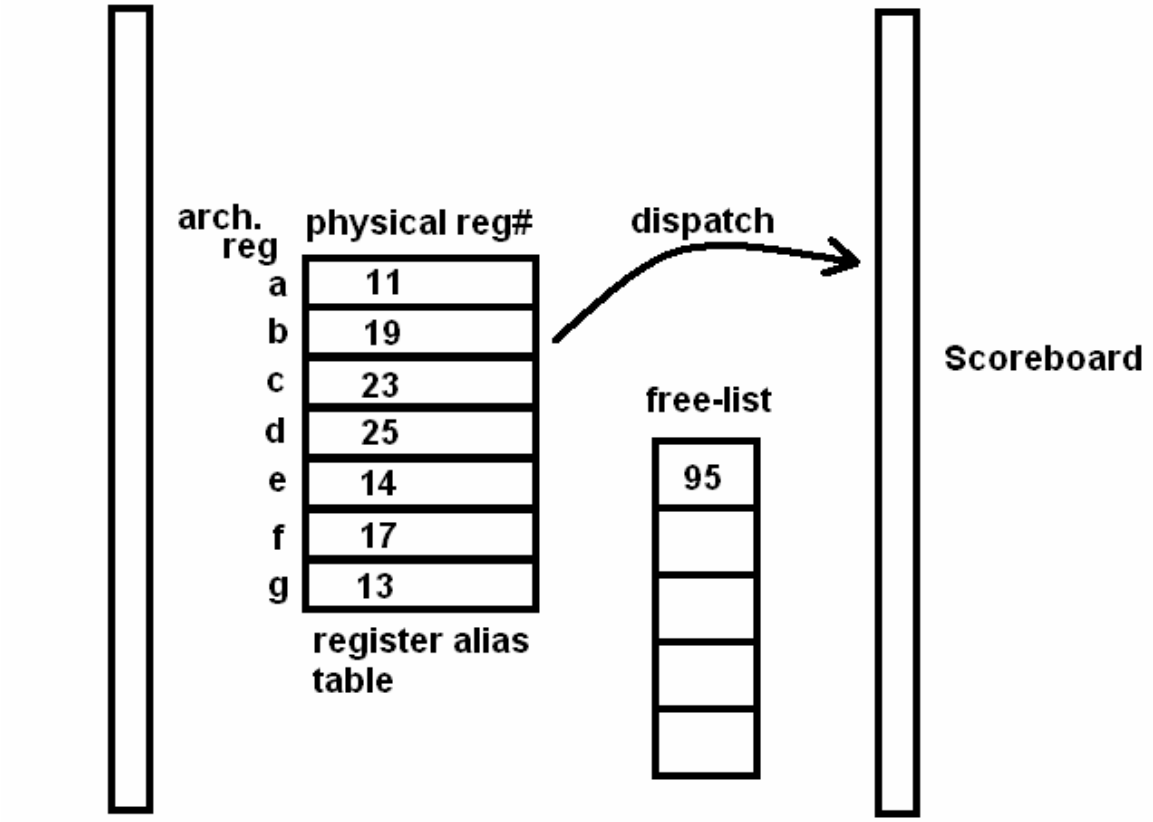
(i.e.) L1 Cache = 4K L2 Cache = 16K

- **-Exclusive Caching: The L1 Cache and L2 Cache do not contain the same data. Having the two caches contain different data allows for a total of 20K of data to be held within the caches.**

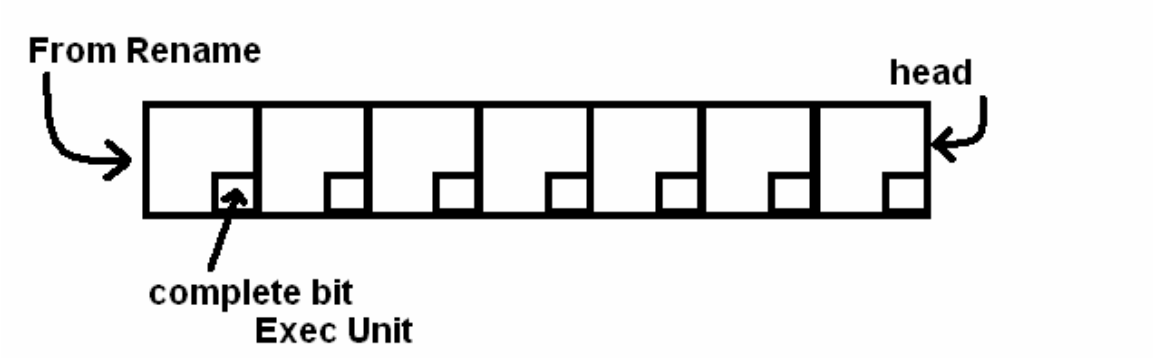
- **Inclusive Caching: The L2 Cache has the same data that is in the L1 Cache. This means that at most 16K of data is being held within the caches.**



<u>Rename</u>: Register Alias Table



Reorder Buffer: (FIFO Structure)



- **-Instructions are stored in the re-order buffer and when they reach the head are retired. Retiring an instruction involves adding a physical register back to the free-list.**

Branch Misprediction & Register Renaming

Code

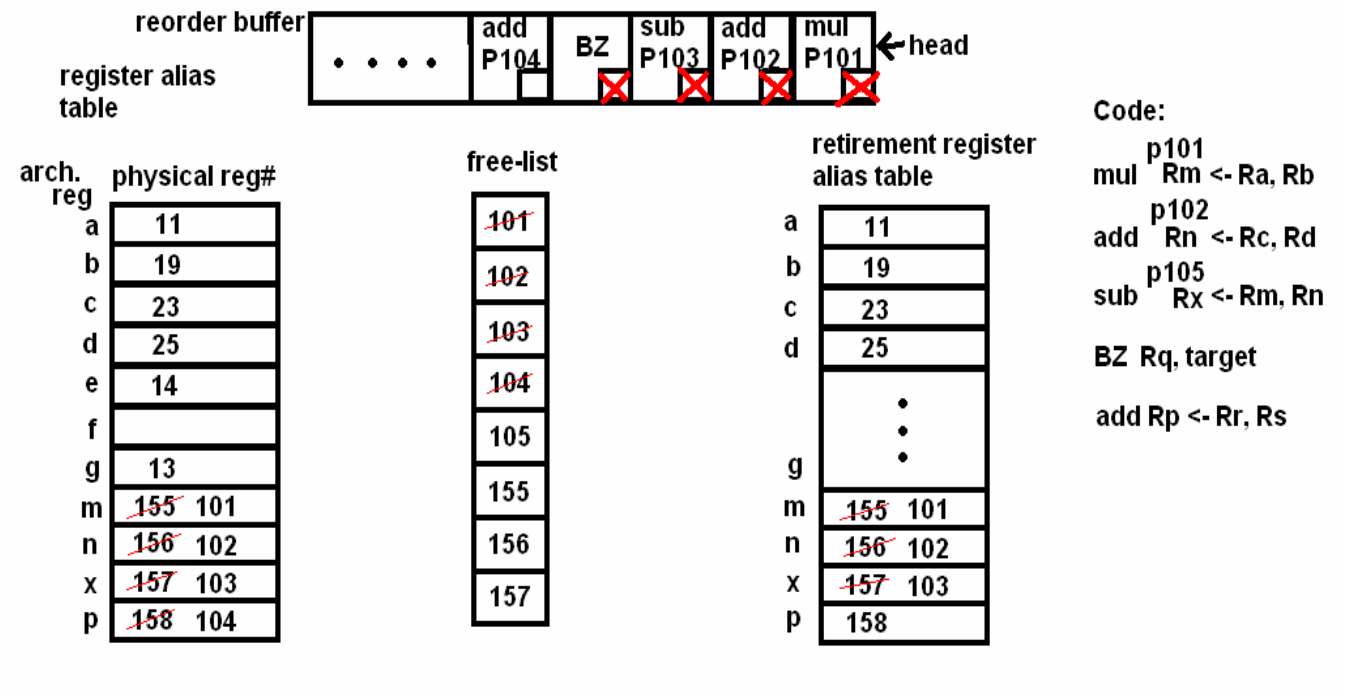
- **mul Rm <= Ra, Rb**
- **add Rn <= Rc, Rd**
- **sub Rx <= Rm, Rn**
- **Bz Rq, target**
- **add Rm <= Rr, Rs**

- **Due to the code arrangement and the branch there is a possibility that register M could be in an incorrect state if the branch is mispredicted.**

Assume that there are an infinite amount of registers!

- **This will change the instruction (add Rm <= Rr, Rs) to (add Rp <= Rr, Rs)**

(i.e.) Branches and Register Renaming



- **Destination registers are assigned physical addresses from the free list.**
- **The new physical register number is written into the register alias table and the register number is removed from the free-list.**
- **Instructions (along with register numbers) are placed in the re-order buffer.**
- **Instructions are retired from the head of the re-order buffer and the value in the retirement table is updated.**

Handling the Branch Mispredict:

- **When the "BZ" instruction is found to be mispredicted, we must clean up the re-order buffer and place the alias table back to the state before the prediction.**
- **We can copy the retirement register alias table to the register alias table to fix the misprediction effects.**
- **All the instructions after the branch in the re-order buffer are marked as 'BAD'**
- **Step through each of the 'BAD' instructions in the re-order buffer and place the destination registers back into the free-list.**