Problem 2: Multithreading (28 pts)

Suppose your friend designed the following fine-grained multithreaded machine:

- The pipeline has 22 stages and is 1 instruction wide.
- Branches are resolved at the end of the 18th stage and there is a 1 cycle delay after that to communicate the branch target to the fetch stage.
- The data cache is accessed during stage 20. On a hit, the thread does not stall. On a miss, the thread stalls for 100 cycles, fixed. The cache is non-blocking and has space to accommodate 16 outstanding requests.
- The number of hardware contexts is 200.

Assuming that there are always enough threads present, answer the following questions:

A) [7 pts] Can the pipeline **always** be kept full and non-stalling? Why or why not? *(Hint: think about the worst case execution characteristics.)*

CIRCLE ONE: **YES** **NO**

NO - will stall when more than 16 outstanding misses in pipe

B) [7 pts] Can the pipeline **always** be kept full and non-stalling if all accesses hit in the cache? Why or why not?

CIRCLE ONE: **YES** **NO**

YES - switching between 200 threads is plenty to avoid stalls due to branch prediction delay

C) [7 pts] Assume that all accesses hit in the cache and your friend wants to keep the pipeline **always** full and non-stalling. How would you adjust the hardware resources (if necessary) to satisfy this while minimizing hardware cost? You cannot change the latencies provided above. Be comprehensive and specific with numerical answers. If nothing is necessary, justify why this is the case.

Reduce hardware thread contexts to 19, the minimum to keep pipe full/non-stalling

D) [7 pts] Assume that all accesses miss in the cache and your friend wants to keep the pipeline **always** full and non-stalling. How would you adjust the hardware resources (if necessary) to satisfy this while minimizing hardware cost? You cannot change the latencies provided above. Be comprehensive and specific with numerical answers. If nothing is necessary, justify why this is the case.

Reduce hardware thread contexts to 100, the minimum to keep pipe full/non-stalling. Increase capability to support 100 outstanding misses