Known Failures:

1. The i-cache and d-cache memory modules (when enabled) will read and write for as many cycles as the other is stalled, even if the write signal is only asserted for one cycle. This causes memory to be read and written multiple times when it should not be. I do not know what the proper inpuits should be so that this is the case. I had assumed it was the rd and wr inputs acting as an enable signal but that does not seem to be the case.
2. If branch prediction happens in the same cycle as an i-cache stall is initiated, the incoming signal is not held until the i-cache unstalls, and the PC register is not flushed correctly. There should be a mechanism that halts the latches between stages during a stall so that signals like branch misprediction do not propagate until the stall is completed. (I believe our implementation of this is what was causing the cycle limit to be hit on our first submission)