Assignment No. 9

http://www.eng.tau.ac.il/~guy/Computer_Architecture/arch_home.html

Pipelining:

1. Discuss the state of the pipeline after power-up. (Changes to data-path to insure a correct state, bus contention issues, and even caches).

2. Assume separate caches for instructions and data.

   (a) Is it interesting to write (by the store instruction) values to memory addresses that belong to the instruction space?

   (b) Assume that there is no "coordination" or "snooping" between the instruction cache and the data cache. What is the meaning of performing a store to an address that belongs to the instruction space.

   (c) Suggest mechanisms for synchronizing the contents of the instruction cache and the data cache.

3. Read the following paper: Silvia Mueller and Robert Knuth, "Correctness of a Mechanism for Precise Nested Processing of Interrupts in Pipelined Designs", http://www-wjp.cs.uni-sb.de/~smueller/pint.ps.gz.

   (If you can't download it, please email me).

   (a) When are interrupts caught? When does the pipeline check if interrupts were caught?

   (b) Describe the new requirement from the hardware so that the interrupt mechanism functions precisely.

   (c) Describe the problem of inconsistent data in the pipeline when the complier uses optimizations that permute the order of instructions (for example, delayed branch). How does the interrupt mechanism deal with this problem (hardware and/or software solutions) ?

   (d) Why does the usage of delayed branch complicate the reconstruction of the machine state after servicing an interrupt? How is this problem dealt with?

   (e) Which program counters (PC's) need to be saved when a jump to ISR occurs? What does this choice depend on and why?

   (f) If the occurrence of an interrupt is identified, how is the machine state protected from changes caused by instructions that are already being processed in the pipeline (consider registers, memory, etc.)?