Assignment No. 4

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

More questions about supporting interrupts.

1. When the status is saved, the ISR checks if a reset had occurred. Should this be done by inspecting $ECA[0]$ or by inspecting $CA[0]$? What is the difference?
2. Similarly, the interrupt that should be serviced is computed as follows:
$$\ell = \min\{j : ECA[j] = 1\}$$
Why shouldn't it be computed as follows:
$$\ell = \min\{j : CA[j] = 1\}$$
3. Why should the registers PC,CA,SR,MAR be copied to the exception registers EPC, ECA,ESR,EMAR? Could one copy them directly to the interrupt stack? Give a reason for each register.
4. Consider swapping the order of some actions during save status and restore status. Show how that would effect correctness.
5. Note that the cause register environment is designed only to "clear its contents" or "accumulate pending interrupts". Consider an internal interrupt event signal $env[i]$. When $evn[i]$ is active, $CA[i] = 1$. How and when is $CA[i]$ reset?
6. Define when the event signal $evn[i]$ should be active for a maskable internal interrupt. Take into account the following situation: suppose $evn[i] = 1$ when $SR[i] = 0$ (interrupt i is masked). This will cause $CA[i] = 1$, however $MCA[i] = 0$. After several clock cycles the user unmasks the interrupt, namely, $SR[i] = 1$. Since $CA[i]$ stays equal to 1, the interrupt suddenly wakes up long after it had occurred. How should $evn[i]$ be defined to avoid such a situation?
7. Why should the instructions movei2s and moves2i be privileged (namely, can be executed only by the kernel)? How can a user change the masking of the overflow interrupt?
8. Define a invariant describing the signal
$$interrupt = OR(MCA[31], MCA[30], \ldots, MCA[0])$$
For example, consider the invariant: "the signal interrupt is active if and only if there is a pending interrupt". Show that this suggestion is wrong, and try to fix it.