ELEC-E8001 Embedded Real-Time Systems

EXAM 15.12.2017

---

1. Are the following claims true (**T**) or false (**F**)? Every correct answer gives you +1 p, every incorrect –1 p, and an empty answer is worth 0 p. The minimum amount of total points is 0 p and maximum 6 p.

a) The best possible instruction completion time of an *N*-stage pipeline is $1/N^2$ times the completion time of the nonpipelined case.
b) Low-end microcontrollers with CPU clock frequencies below 8 MHz do not suffer the CPU–Memory bottleneck.
c) The locality of reference principle forms a basis for hierarchical memory organizations.
d) A benefit of using an instruction cache in hard real-time systems is that the effective access time for instructions is guaranteed to be deterministic.
e) Dynamic RAMs and static RAMs belong to the group of volatile memories.
f) The accuracy of a practical A/D-conversion channel is always the same as its resolution.

2. An embedded real-time system with three periodic tasks has the execution periods of p_i and execution times of e_i as follows (here, the possible effects of instruction pipeline and cache are considered negligible):

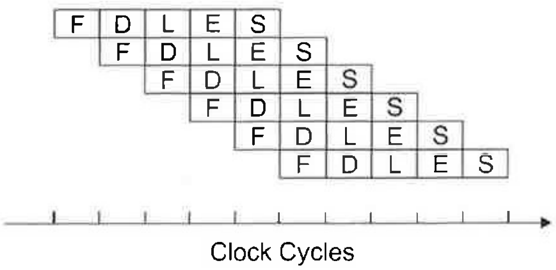
- p_1 = 250 ms and e_1 = 1,000,000/f_c
- p_2 = 50 ms and e_2 = 200,000/f_c
- p_3 = 10 ms and e_3 = 40,000/f_c

where f_c is the CPU clock frequency.

You are designing a *low-cost system* with *short lifetime*, and intend to use a 32-bit microcontroller. This microcontroller is available in three versions, which have different clock frequencies, f_c = 10 MHz, f_c = 20 MHz, and f_c = 50 MHz. And the unit prices in desired quantities are 1.25 €, 2.40 €, and 4.99 €, respectively.

Which one of those three versions would you choose for this application and why? (6 p)

3. Explain with an example of a sequence of assembly-language instructions how the instruction pipeline shown below could benefit from the availability of Harvard architecture (compared to the more traditional von Neumann architecture)? (6 p)



Fetch instruction (F), **D**ecode instruction (D), **L**oad operand from memory (L), **E**xecute ALU function (E), and **S**tore result to memory (S).

4. Priority inversion may occur in a real-time multitasking system under certain conditions.

a) Why is it harmful? (1 p)
b) Illustrate with an appropriate execution scenario how the priority inversion occurs in a three-task system under the control of a Real-Time Operating System (RTOS) with preemptive priority scheduling. (3 p)
c) What is the common solution to such a priority inversion problem, and how would it work in your scenario? (2 p)

5. Consider an RTOS with preemptive-priority scheduling.

a) Draw a representative state diagram that shows all the possible task states and allowed transitions between them. (4 p)
b) Define the states and transitions unambiguously with a few words. (2 p)