ELEC-E8001 Embedded Real-Time Systems

EXAM 16.12.2016

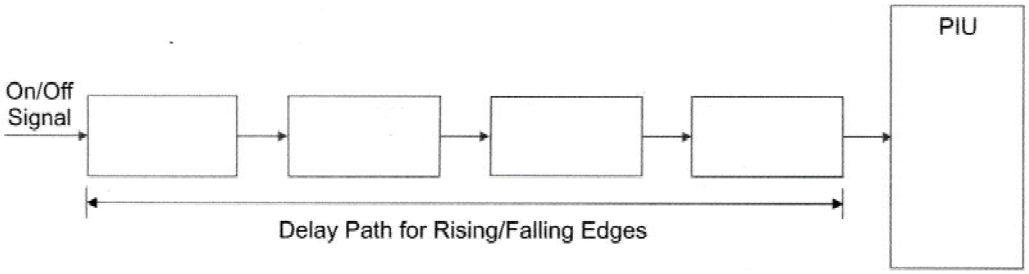
---

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) In real-time control systems, deadlines are based on the underlying physical phenomena of the system under control.
   b) In cost-effective and robust real-time systems, a pragmatic rule of thumb could be: process everything as slowly as possible and repeat tasks as often as possible.
   c) Speculative execution works well with instruction pipelines if the locality of reference remains low.
   d) The use of OSI model makes it possible to change the data transfer medium and other properties of the protocol stack independently.
   e) A task is an abstraction of a running program and is the logical unit of work schedulable by the real-time operating system.
   f) Many embedded systems, which transmit blocks of data from one unit to another, use double-buffering schemes with a software or hardware switch to alternate the buffers.

2. You have a 10-bit A/D converter and its input range is –5V to +5V. If the binary output of the A/D is `0100000100` what analog voltage is at the input? (6 p)

3. In the textbook and in one of your homework assignments, there is a discussion and problem related to a *digital input channel* intended for operating environments with high EMI levels. Below is the block diagram of that input channel without descriptions of the four blocks between the "On/Off Signal" and "PIU". Explain the purpose and function of those blocks. (6 p)



4. Consider a preemptive-priority RTOS and an embedded system with separate measurement channels for pressure and temperature, as well as a single A/D converter to be used by `Task_1` and `Task_2` for periodically measuring those two quantities. Before starting an A/D conversion, the desired measurement channel must be selected. How would you share the serially reusable resource *safely* with `Task_1` (high priority) and `Task_2` (low priority)?

   Give your answer in a few lines of program code for both of the tasks using, for instance, C-like syntax. Define all the operating system services and other functions that you may use. (6 p)

5. Consider a real-time operating system with preemptive-priority scheduling. Draw a representative state diagram that shows the possible task states and allowed transitions between them. (4 p) In addition, define all the states and transitions. (2 p)