ELEC-E8001 Embedded Real-Time Systems

EXAM 12.3.2018 MODELS

---

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) Real-time kernels provide three essential functions with respect to software tasks: scheduling, dispatching, as well as intertask communication and synchronization. **T**, *textbook p. 81*

b) In interrupt-only systems, the various tasks are scheduled via interrupts, whereas task dispatching is performed by interrupt-handling routines. **T**, *textbook p. 87*

c) In rate-monotonic systems, the priority of a task with respect to that of other tasks is adjusted dynamically as tasks are released and executed. **F**, *textbook p. 90*

d) An atomic operation refers to a group of suboperations that can be combined to appear as a single (noninterruptible) operation. **T**, *textbook p. 88*

e) The detection of a deadlock with resource sharing can be handled with watchdog timers that also solve the root problem behind any deadlock. **F**, *textbook p. 118*

f) When applying the priority ceiling protocol, a task can be blocked by a lower-priority task only once, and at most the duration of one critical section only. **T**, *textbook p. 122*

ELEC-E8001 Embedded Real-Time Systems

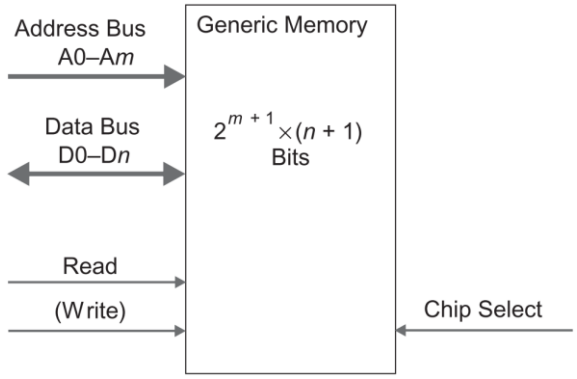
EXAM 12.3.2018 MODELS

---

2. The figure below illustrates the interface lines of a generic memory component. Assume $m = 15$ and $n = 7$. The address bus of your CPU is 17 bits wide.

a) How, in principle, could you locate this particular memory block to begin from the address 10000 (hexadecimal "Hex")? (4 p)

b) What is the corresponding end address? (2 p)



Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Binary–hexadecimal–decimal conversion

Homework assignment #4B

The size of the memory component is 2^{15+1} = 65536 memory locations, and 1 kilo is 1024 → 64 K. And the width of each memory location is 7 + 1 bits or 1 byte. With 17 address lines, the CPU can address 2^{17} = 131072 memory locations or 128 K. Hence, our 64 K block will take a half of the available memory space.

b) The *beginning* address of our 64 K block is 10000 (Hex), i.e., 1 0000 0000 0000 0000 (Binary). Since the memory component has 16 address lines (A0–A15), the highest possible address corresponds to the case when all these lines are high or "1", i.e., 1111 1111 1111 1111 (Binary). This added with the beginning block address gives the *end* address: 1 1111 1111 1111 1111 (Binary) or 1FFFF (Hex). (2 p)

a) We could locate our memory block to the address range 10000–1FFFF (Hex) by activating the Chip Select line when the CPU's highest address line A16 is "1". When A16 is "0", the Chip Select line of this memory component must be kept passive. (4 p)

ELEC-E8001 Embedded Real-Time Systems

EXAM 12.3.2018 MODELS

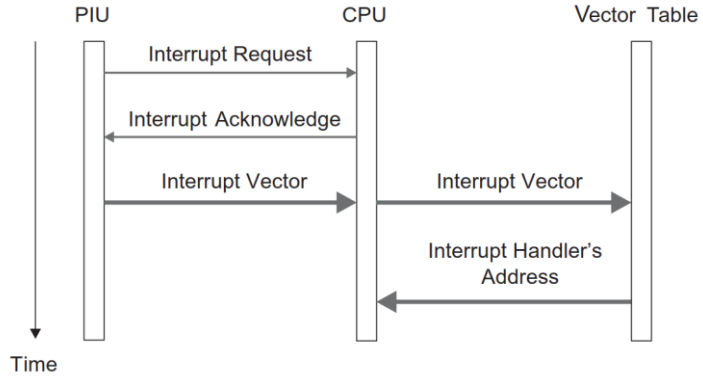
---

3. Compare the *polling* and *vectored interrupt* principles in providing service for I/O peripherals.

a) Explain the operation of both principles. (4 p)
b) Give their advantages and disadvantages. (2 p)

a) In a **polled** I/O system, the status of the I/O device is checked periodically, or, at least, regularly. Therefore, such I/O activity is software controlled; only accessible status and data registers are needed in the hardware side. *Textbook p. 52* (2 p)

Vectored interrupt



Textbook p. 54 (2 p)

b) An obvious advantage of **polling** is its simplicity, but, on the other hand, it loads the CPU due to possibly unnecessary status requests. Typically, only a minority of status requests leads to either input or output transactions with the data register. This unnecessary loading could be reduced by less frequent polling of the I/O status. However, that would increase the worst-case I/O latency. *Textbook p. 52* (1 p)

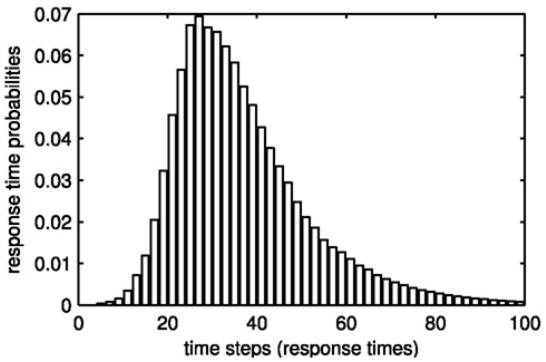
Interrupt-driven I/O processing has remarkable advantages over the straight-forward polled I/O: the service latency can, in general, be reduced and made less uncertain without increasing the loading of the CPU. *Textbook p. 53*
Vectored interrupt handling is a convenient technique for larger real-time systems, because it moves the interrupt identification burden from system software to real-time hardware. More complex than polling from the hardware viewpoint. *Textbook p. 53–54* (1 p)

ELEC-E8001 Embedded Real-Time Systems

EXAM 12.3.2018 MODELS

---

4. Below is a distribution of measured response times for a specific input–output pair in a crane control application. Give possible reasons behind such a long-tailed distribution from the *hardware* (2 p), *application software* (2 p), and *system software* (2 p) viewpoints. This multi-tasking application is running under an RTOS with preemptive-priority scheduling.



Multi-step and time-variant delay paths from inputs (excitations) to outputs (responses) create considerable timing and latency challenges. These latency and uncertainty issues are discussed throughout *textbook's Chapters 2 and 3*.

Two "possible reasons" of each type are needed for full 2+2+2 points.

Hardware (2 p)

- Pipeline flushes
- Cache misses
- Interrupt or DMA latency
- Fieldbus network's variable loading

Application software (2 p)

- I/O polling
- Alternative variable-length execution paths
- Aperiodic interrupts

System software (2 p)

- Task priorities
- Scheduling needs
- Resource sharing

ELEC-E8001 Embedded Real-Time Systems

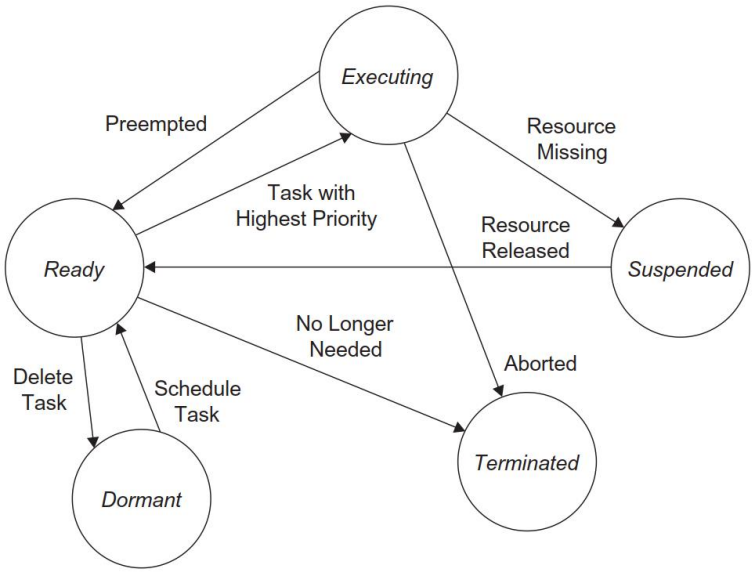
EXAM 12.3.2018 MODELS

---

5. Consider a Real-Time Executive with *preemptive-priority* scheduling.

a) Draw a representative state diagram that shows all the possible task states and allowed transitions between them. (3 p)
b) Define the states and transitions unambiguously with a few sentences (not just the short labels on the state diagram). (3 p)

a) *Textbook p. 97* (3 p)



–1 p for missing/incorrect "Ready", "Executing" or "Suspended" state.
–0.5 p for missing/incorrect "Dormant" or "Terminated" state.

–1 p for missing/incorrect "Preempted", "Task with…", "Resource Missing" or "Resource Released" transition.
–0.5 p for missing/incorrect "Delete Task", "Schedule Task", "Aborted" or "No Longer…" transition.
(Minimum 0 p)

b) *Textbook pp. 95 – 97* (3 p)

Ready: A task is ready for execution, but it is not executed because a higher priority task is in execution.
Executing: The highest priority ready task is executed.
Suspended: A task is missing some necessary resource that prevents its further execution.
Dormant: A task exists but is currently unavailable for scheduling.
Terminated: A task has finished its running permanently.

Preempted: A higher priority task is ready for execution and the currently executing task is therefore preempted.
Task with Highest Priority: A task with a higher priority than the currently executing task and the highest priority of the ready tasks moves to execution.
Resource Missing: The currently executing task needs some resource that is not available and therefore it needs to be suspended.
Resource Released: A suspended task receives its missing resource and becomes ready again.
Delete Task: A task is moved away from active scheduling while it still exists
Schedule Task: An existing task is moved to active scheduling.
Aborted: An executing task is aborted due to some problem (typically) and it becomes terminated permanently.
No Longer Needed: A ready task is aborted because it is no more needed (typically) and it becomes terminated permanently.

–1 p for missing/incorrect state definition.
–0.5 p for missing/incorrect transition definition.
(Minimum 0 p)