EMBSYS CP105 - Final

1. Define Real-Time.
	1. What system characteristics, e.g., hardware, OS, design, are common in a real-time systems?
	2. How are they different from regular non-realtime systems?
	3. Talk about the constraints, requirements or other issues involved in deciding to use a realtime system.
	4. What is Jitter?
2. What are interrupts?
	1. Where do they come from?
	2. Why are they used?
	3. What problems do they introduce to a system's designs?
3. ARM CPU
	1. What are the different modes? Describe each one. There are seven of them. How are they different in terms of the registers, permissions, usage, or purpose?
	2. What are CPSR and SPSR? How do they relate to the modes?
	3. What is the Vector table?
	4. What are Exceptions? How do they relate to the modes, CPSR/SPSR, registers, and Vector table? How do they work together to handle the different Exception types.
4. Exceptions
Filling the following table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exception | Cause | Where does it go? | Mode | Return instructions |
| Reset |  |  |  |  |
| Data Abort |  |  |  |  |
| Prefetch Abort |  |  |  |  |
| Interrupt |  |  |  |  |
| FIQ |  |  |  |  |
| Software Interrupt |  |  |  |  |
| Undefined |  |  |  |  |

1. Interrupt Handling
	1. What state needs to be saved?
	2. How do you save it?
	3. How do you let uCOS know you are in an interrupt handler?
	4. What state are interrupts in?
	5. What uCOS methods you should not call?
2. Critical Sections
	1. What are Critical Sections?
	2. How do you use them?
	3. Why are they important?
3. Tasks
	1. What do you need to allocate for a task before stating it?
	2. What is the Task Control Block (TCB)? When is it used?
	3. What is a stack? How is it used?
	4. Why is a task’s priority important?
4. Task Behavior
	1. How is a task started?
	2. Does a task return?
	3. Which task will run?
	4. What causes a task switch?
	5. How do you make a task wait or yield to let another task run?
5. Inter-Task Communication
	1. What are some of the issues or problems that can occur when tasks try to share data?
	2. Semaphores
		1. What are they?
		2. Describe some of the ways they are used.
		3. What does it mean to acquire, release, and be blocked?
		4. How are semaphores used from interrupts?
	3. Mutexes
		1. What are they? What problem do they solve?
		2. How are they different from semaphores?
6. Processes and Virtual Memory
	1. What is a process and how is it different from a task?
	2. What is virtual memory?
		1. Roughly, how does it work?
		2. What is a Translation Lookaside Buffer?
		3. What are Page Tables?