Student Name: Temirlan Mussagaliyev

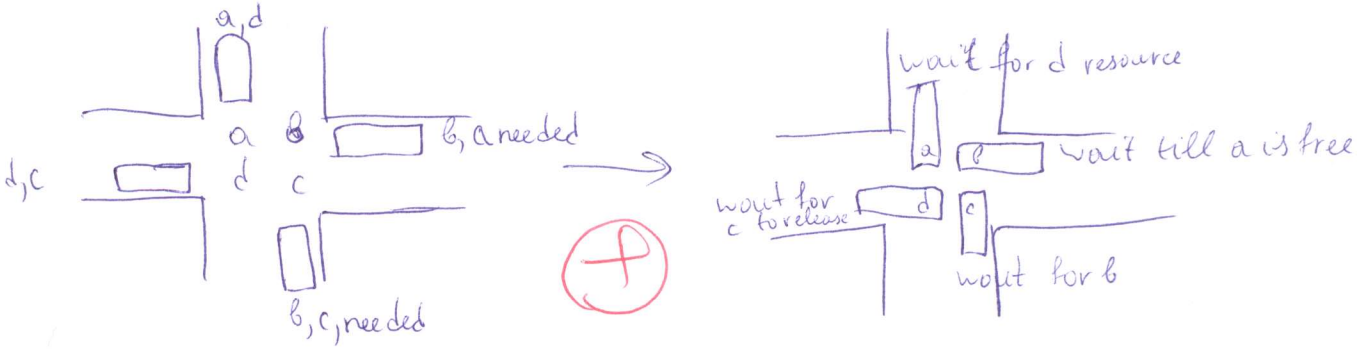
6,5

ROBT 305 – Embedded Systems Quiz #4 Section 2

Collect 6 out of 7 points. Please provide precise answers.

1. Explain and give example of a deadlock? **(1 point)**

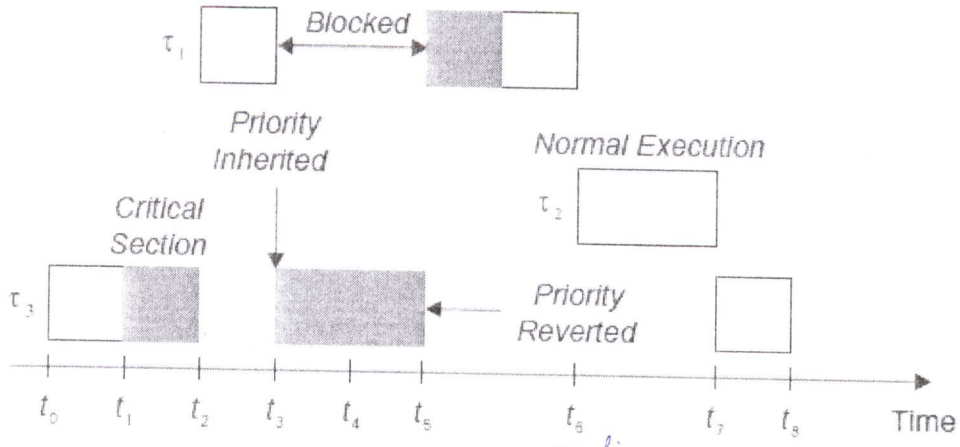
Deadlock - is a permanent blocking of a set of processes when tasks compete for resource or communicate w/ each other.



+

2. Explain a typical priority inheritance protocol using the figure below? **(1 point)**

- Tasks τ_1, τ_2 and τ_3 have following priorities: $\tau_1 > \tau_2 > \tau_3$
- Tasks τ_1 and τ_3 share some data or resource with exclusive access



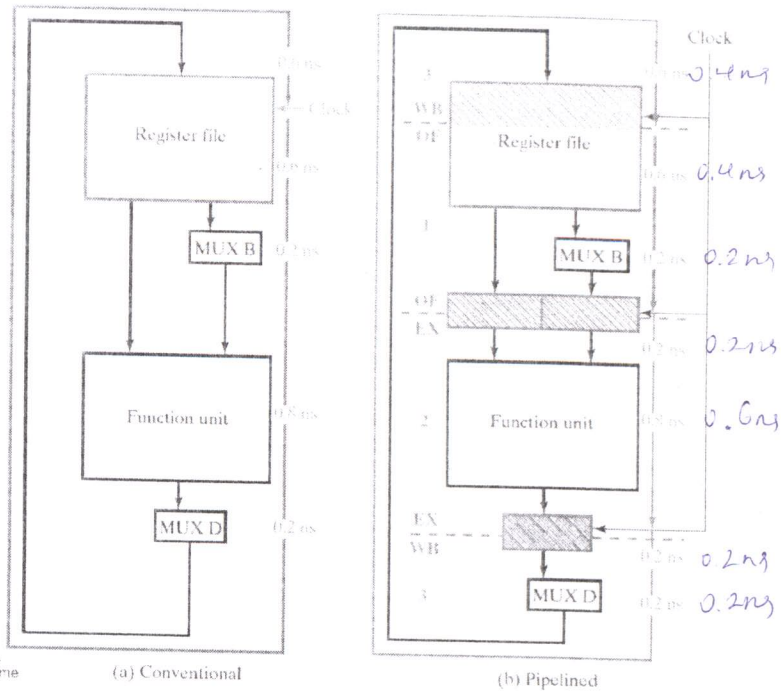
Priority inheritance protocol applies so that lower priority tasks temporarily inherits a priority of higher priority task until its end of execution.

+

Here τ_3 inherits the priority of τ_2 and, due to transitivity, it inherits τ_1's priority. So, during its execution it is not interrupted, and then τ_1 is executed because it has the highest priority now. At the end, τ_2 is executed normally.

3. The pipelined datapath is similar to the one in Fig (b) with the delays **from top to bottom** replaced by the following values: 0.4 ns, 0.4 ns, 0.2 ns, 0.2 ns, 0.6 ns, 0.2 ns and 0.2 ns.

a) Define the maximum clock frequency **(1 point)**



$0.2 + 0.6 = 0.8$ ns – the longest time for one of the stages, which is EX (execution)

max frequency $= \frac{1}{0.8\text{ ns}} = 1.25$ GHz

+

b) The latency time (time to process an instruction) **(1 point)**

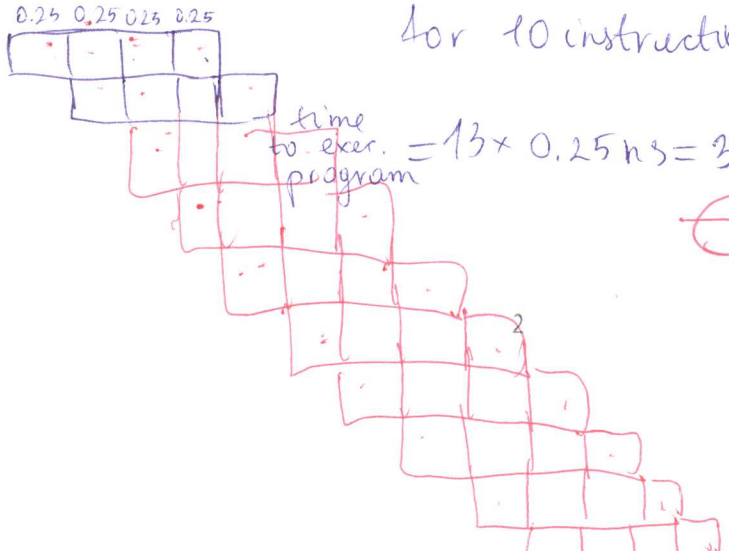
The latency time $= 0.4 + 0.4 + 0.2 + 0.2 + 0.6 + 0.2 + 0.2 =$

$= 2.2$ ns

$0.8 \cdot 3$ stages $= 2.4$ ns

+

4. A program consisting of a sequence of ten instructions without branch or jump instructions is to be executed in a 4 stage pipelined RISC computer with a clock period of 0.25 ns. Compute the time required to execute the program **(1 point)**



for 10 instructions – 13 clock cycles

time to exec. program $= 13 \times 0.25\text{ ns} = 3.25$ ns

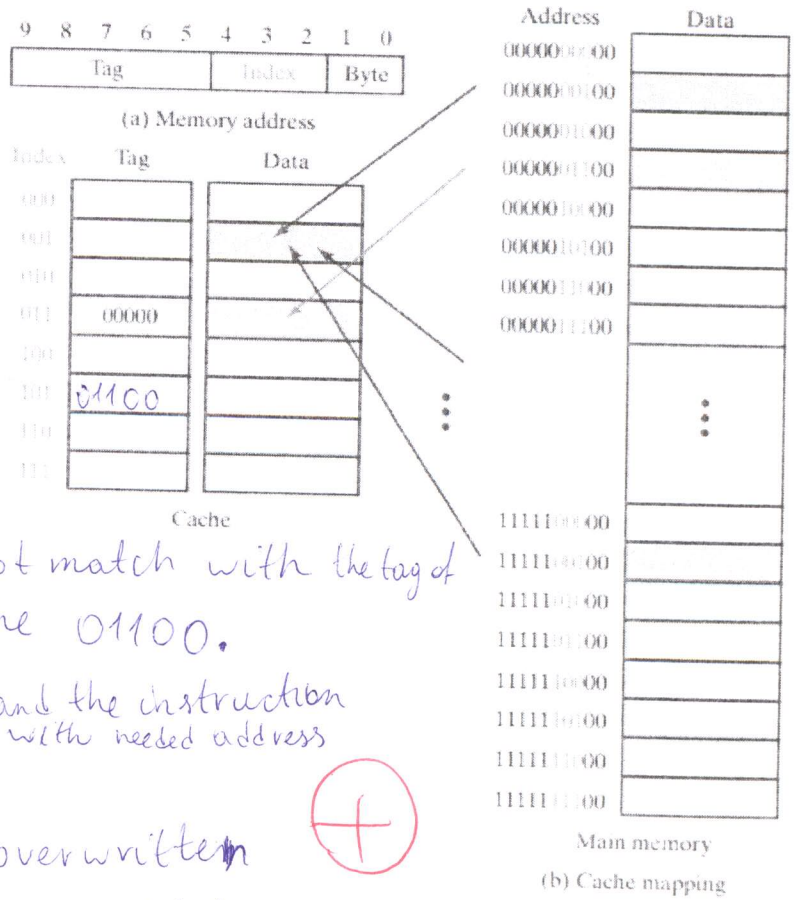
+

5. Describe by your own worlds, how a processor handles an interrupt request? **(1 point)**

1) Interrupt program activated on hardware line
2) Interrupt program latched by CPU
3) Execution of current program is stopped
4) program instructions in counter (PC of current program) are pushed to stack
5) status register's SR of current program are pushed to stack
6) Interrupt program is executed
7) The content of PC (stack for) is popped back from the stack
8) The content of SR (stack for) is popped back from the stack
9) The execution of the original program, which was interrupted, continues.

+

6. In the cashe memory example, Assume that the cashe cell with index 101 contains tag 01100. Describe by your own worlds how CPU will fetch an instruction with memory address 0111010100 **(1 points)**



CPU check the tag of the address index 101.

The tag of the desired instruction 01110 does not match with the tag of the instruction in the cache 01100.

CPU retrieves the data and the instruction with needed address from the main memory.

The cache cell 101 is overwritten with new tag, initially requested by CPU 01110, and the new data from the main memory for future uses.

+