PART #1 (DSP programming)

Shown below is the design of an “IIR direct form type II filter”, described in Chapter 5 of the text (see examples 5.3 and 5.4). Design a program for the TMS 32C5x DSP, discussed in class, to implement this filter, using the format of the class examples. Data sample x(n) is to be read from an input port, and filter output y(n) is to be sent to an output port, as in the “Low Pass Filter” example in the class slides. The values a1, a2, a3, b0, b1, b2, b3 are constants, stored in program memory. Variables v0, v1, v2, v3 are to be stored in data memory. You may assume that all values are in the “proper format” to use for this calculation, as shown in the examples on the class slides.

1. Submit the source program (typed or hand written)
2. On the submitted program, show how you have the variables arranged and stored in memory.

PART #2 (UML – related to Chapter 3, Section 3.2)

At the end of Chapter 3, answer/work the following, which deal with input/output operations:

- Q3-6 UML sequence diagram for a busy-wait device read
- Q3-8 UML sequence diagram for copying characters between devices
- Q3-13 UML sequence diagram for interrupt-driven device read
- Q3-17 UML sequence diagram for prioritized nested interrupts
- Q3-18 UML sequence diagram for prioritized nested interrupts