1. **(15 points)** Build a CFSM $M$ that models the following Message Selective Discarding (MSD) algorithm used for congestion control in ATM networks. At every network node MSD decides whether to accept or reject input packets (in ATM jargon called cells) based on the number of packets currently stored in the node internal buffer. Input cells are grouped into messages of variable size (i.e. each message may contain a variable number of cells). The header of each cell includes the 3-bit PTI field. PTI has value 1 or 3, if the cell is the last cell of a message, has value 4 or 5 if the cell is of type OAM (Operations And Management are special cells not belonging to any message), has value 0 or 2 or 6 or 7 if the cell is neither an OAM cell nor is the last cell of a message. The MSD algorithm ensures integrity at the message level, i.e. either all the cells in a message are accepted or all are rejected.

![Block diagram](image)

Figure 1: Block diagram

The MSD algorithm performs the following sequence of actions:

- reads an input cell at the time
- if the cell is the first one of a message
  - checks the number of cells currently stored in the internal buffer
  - compares the number of buffered cells ($n$) with a threshold value ($Th$): if $n \leq Th$ the cell is accepted, otherwise rejected.
- if the cell is not the first one of a message, it is treated (i.e. accepted or rejected) as the first cell of the same message.
- if the cell is an OAM cell ($PTI = \{4, 5\}$), it is always accepted.
(Hints. 1) CFSMs are Extended FSMs, i.e. the guard of a transition may include expressions on variables. 2) Assume that any input event a carries an entire ATM cell.)

2. (15 points) Give a simple example of two Finite State Machines $M_1$ and $M_2$ whose composition is empty. Show the transition and output relations of $M_1$ and $M_2$ and the construction used to obtain the output relation for the composition.

3. (20 points)

![State Diagram for Exercise 3](image)

- Given the FSM $M$, whose State Diagram is shown in Figure 2, find a reduced FSM $M'$ in which no distinct states are equivalent (Two states are equivalent if and only if, no matter which input sequence is applied to them, the two resulting output sequences are identical).
- From the Mealy machine $M'$ derive an equivalent Moore machine.

4. (50 points)

Model a control component of a radio transmitter using a Finite State Machine.

The component of the radio transmitter you are required to model inserts a synchronization sequence (pilot) at the beginning of each new transmission and, then, a re-synchronization sequence after every sequence of 128 symbols of payload data. The purpose of these synchronization sequences is to allow the receiver to extract the clock from a predefined sequence of symbols. Each pilot/data symbol is triggered by a rising edge of the sCLK input (symbol clock).

More detailed specs follow:

- the block has 3 inputs: ON, TX, sCLK (binary, take value 0 and 1) and 3 outputs: PILOT, DATA, PD (also binary, take value 0 and 1)
- when the radio is off (ON==0 and at reset), PILOT=DATA=0.
- when the radio goes on (ON==1), the radio enters standby mode, where PD=PILOT=DATA=0.
- from standby mode, when TX==1 the radio transmitter sets PILOT=1. Then, triggered by rising edges of sCLK, it emits the synchronization sequence consisting of 8 (PILOT=1 and PD=0), 4 (PILOT=0 and PD=1), 8 (PILOT=1 and PD=0). After that, still triggered by rising edges of sCLK, it loops emitting 128 (PILOT=0, PD=0, DATA=1) followed by a shorter synchronization sequence consisting of 8 (PILOT=1, PD=0, DATA=0).
• whenever the radio is ON and TX goes to 0, the radio enters stanby mode.
• whenever ON==0, the radio goes off

Assume sCLK is a sequence of 0s and 1s, e.g. 000110000100.

(a) Provide the state diagrams of:
   • a minimal FSM representation using the notions of concurrency and hierarchy
     (you can use StateCharts)
   • a “flat” FSM representation (with no concurrency and hierarchy).

(b) Specify the behavior of the block and run functional simulations in the VCC environment. Turn in the source code and the traces obtained from the simulation outputs using the testbench available on the class website. You are free to use either Whitebox C or Blackbox C++ or STD to describe the behavior.
(Hints: the “flat” FSM model may be a bit longer to describe but presents no difficulties using STDs, the minimal FSM description requires to use some tricks because VCC STDs do not support hierarchy and concurrency).