1. Draw an FSM equivalent to the following StateChart description

![StateChart Diagram](image)

2. Figure below shows a simple embedded system: a process T takes readings from a set of thermocouples (via an analog to digital converter, ADC) and makes appropriate changes to a heater (via a digitally controlled switch). Process P has a similar function, but for pressure (it uses a digital to analog converter, DAC). Both T and P must communicate data to S, which presents measurements to an operator via a screen. Note that P and T are active; S is a resource (it just responds to requests from T and P). The overall objective of this embedded system is to keep the temperature and pressure of some chemical process within defined limits. (A real system of this type would be more complex – allowing, e.g., the operator to change the limits.) The system implementation can be:
   a. a single (sequential) program is used which ignores the logical consistency of T, P and S. No OS support is needed.
   b. T, P and S are written in a sequential programming language (either as separate programs or distinct procedures in the same program) and OS primitives are used for program/process creation and interaction.

Illustrate each of the three solutions with sample pseudocode and pros and cons for each of the three approaches. **Hint: Assume existence of functions to: convert temperature reading to heater setting, pressure reading to pressure setting**
3. Here is a system we are looking to design:

\[ y = f(w)|x\cos\left(\frac{2\pi n}{T}\right) - f(w)|y\sin\left(\frac{2\pi n}{T}\right) \]

a) Implement this system using SDL; assume you have two functions, \( f(x) \) and \( f(y) \) which can be called on the transition of Mapper.

b) Implement it using a data flow graph
4. Use Petri nets to describe a function of a simple protocol described below. Transmission is initiated by TXInterface with a TXRequest signal. The receiver answers with an ACK. TXInterface then sends a series of data depending on the number of words (n) input into TXInterface. To close the session EndTX signal is sent and close occurs once the signal is acknowledged. For an outline see graph below.

5. Implement the design shown in figure below using the discrete event model/language (e.g. VHDL). Note: + is an xor function, the squares represent one-bit registers, outputs are values stored at the current clock cycle in each 1-bit register, where MSB is represented by x4, and LSB is represented with 1. Assume the device can be initialized with any number (e.g. 11001).
   a. Assume there is no delay in each register – what kind of outputs do you get at each register?
   b. Assume there is a finite delay in each register – what kind of outputs do you get at each register now?

What is this design’s function?

7. Describe what the following code does. (Hint: tick is a special pure signal that represents the activation clock of the reactive program. Its status is present at each instance.)

```plaintext
module STATION :
  input Request;  % from user
  output Granted; % to user
  input PreviousPassed; % from previous station
  output Pass;  % to next station
  input Token; % from previous station
  output PassToken; % to next station
  loop
    present [ Token or PreviousPassed ] then
      present Request then
        emit Granted
      else
        emit Pass
      end present
    end present
    each tick
    ||
    loop
      present Token then
        await tick; await tick
        emit PassToken
      else
        await tick
      end present
    end loop
end module
```
8. Petri nets can be used to model SDF graphs. Given a Petri net shown below, answer the following questions:

![Petri net diagram]

a. Write its incidence matrix.
d. Derive a PASS assuming e=6.
e. Draw a coverability tree assuming that you are starting from $x_0=[0,0]$ and have enough tokens to go through schedule $S=\{t1,t1,t1,t2,t2\}$
f. What initial condition is needed in order for schedule $S$ to be executable?
g. What is the maximum buffer size needed in places $p1$ and $p2$ when using schedule $S$?