CSD – P6: FSM design procedure

1. Specifications

a) Draw the project symbol and explain how the circuit works using function tables or similar descriptions.

b) Draw a sketch of timing diagram because in this Chapter 2 representing the system evolving in time is a fundamental idea. Besides, it will be translated as an example stimulus process for test benching.

2. Planning

c) Infer a state diagram (sometimes it is discussed in specifications because it is a tool of great help to comprehend the project). Do it always in the same way: cycles, states; arrows, state transitions and loops; parenthesis in a different colour indicating outputs for each state.

d) Draw the FSM general architecture.

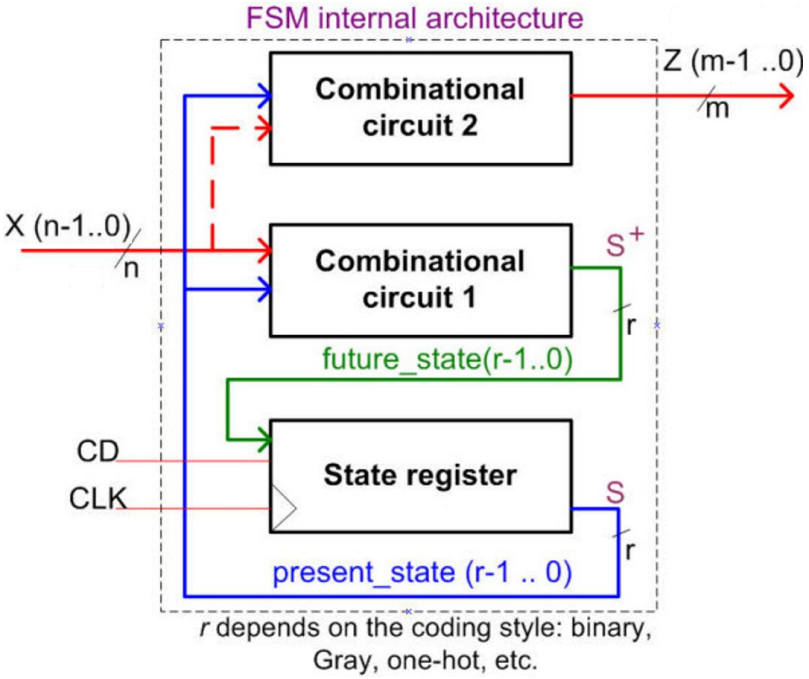

Fig. 1. FSM architecture in a single file (plan C1) using up to three processes. Signals future_state and present _state are also named next_state and current_state respectively.

D_FF is the building block acting as state register (r-bit memory cell).

The *present_state* is updated after the CLK's rising edge.

CD = '1' resets the FSM.

e) Adapt the general FSM architecture to your problem and draw the **state register** based on *D_FF*. Deduce how many *D_FF* are required if you are coding states in binary sequential, one-hot or Gray.

f) Write the truth table of **CC2** and its equivalent behavioural (plan B) interpretation as a flowchart. CC2 is for calculating outputs, thus, this combinational circuit implements all the signals drawn in parenthesis in the state diagram.

g) Write the truth table of **CC1** and its equivalent behavioural (plan B) interpretation as a flowchart. CC1 is for calculating the next state to go, thus, this combinational circuit is in charge of all the circuit state transitions (arrows).

3. Development

h) Write the VHDL file (plan C1, a single file containing three processes representing the translation of the FSM. State-type signals: state enumeration.

i) Run a project using an EDA synthesis tool for a CPLD or FPGA target chip.

j) Discuss RTL and technology schematics. Check the number of registers *D_FF* used.

4. Test (functional)

k) Generate the skeleton of the test bench fixture. Translate your initial timing diagram sketch into VHDL stimulus processes. Consider the *CLK_period* constant and how long the simulation has to run.

l) Simulate the circuit using your VHDL test bench and discuss the results. Represent the waveforms of *current_state* and *next_state* signal as well in wave diagrams.

5. Test (gate-level)

m) Simulate the technology circuit (flat *vho*, *sdo* or *sdf*) using your VHDL test bench and discuss results.

n) Using the timing analyser tool, measure t_{CO} and deduce the maximum CLK frequency that can be applied to your design considering a target chip.

6. Prototyping

It is always possible to use any of our lab training boards for CPLD or FPGA to synthesise the circuit and experimenting in the lab using real instrumentations for measurements and characterisation.

7. Report (handwritten) and presentation

Follow usual indications on reporting. It is easy for most circuits to generate more that twelve pictures, sketches, diagrams, computer results, etc. to report perfectly well how the circuit that you have invented is conceived and how does it perform. Such figures may be used later as slides for oral presentations and recordings.