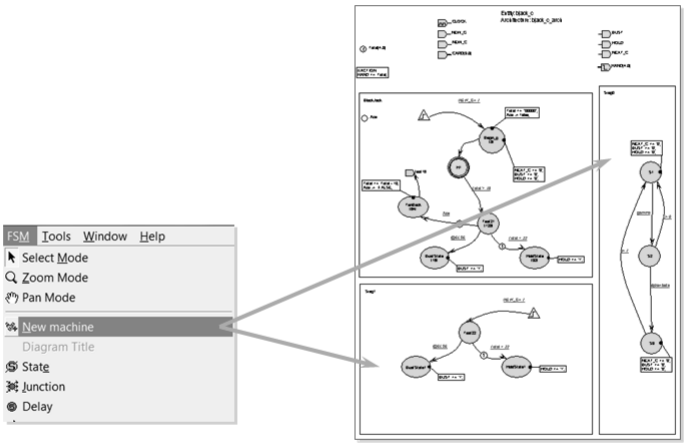
Multiple State Machines



ALDEC
THE DESIGN VERIFICATION COMPANY

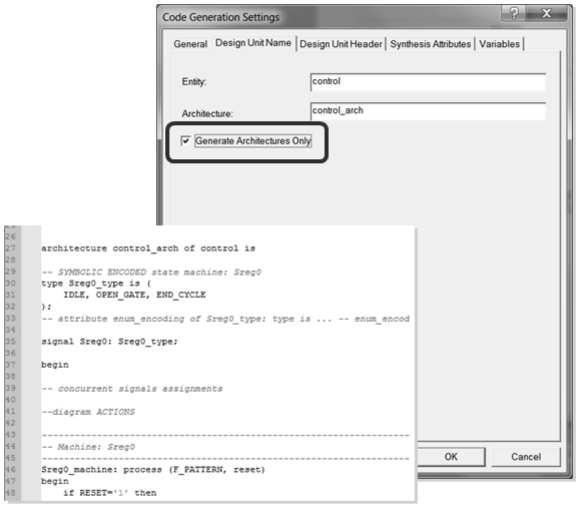
Note:

All machines defined within a single state diagram share the interface (ports).

The Active-HDL State Diagram Editor allows the user to describe the behavior of a design unit using multiple concurrent state diagrams in one document. Each state diagram can include more than one synchronous or asynchronous machine. If the state diagram consists of multiple machines, each machine is implemented with a separate process/always statement (or set of statements, depending on the settings in the Code Generation Settings dialogue box).

The space on the diagram has to be partitioned and the New Machine menu option will create a frame for the new state machine, a machine is shown on a state diagram as a rectangular frame (or rectangular frame with rounded corners in case of the hierarchical state diagram).

Multiple Architectures Support



The State Diagram Editor allows the user to generate the VHDL code that contains an architecture body only. This way you can create and use different implementations (several architectures) for the same entity.

Access the required code generations settings from the menu using **FSM→Code Generation Settings Design Unit Name Tab** and check the **Generate Architecture Only** check box. (see Figure 17)

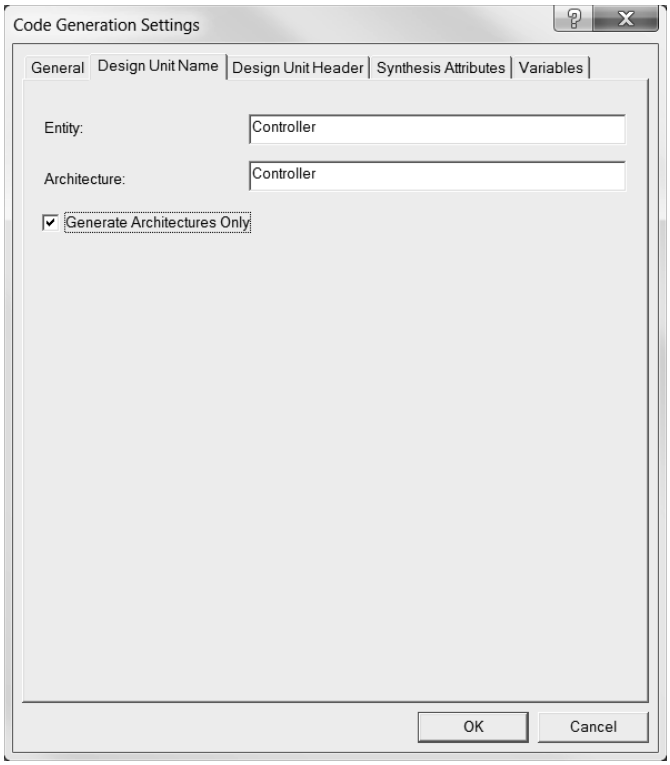
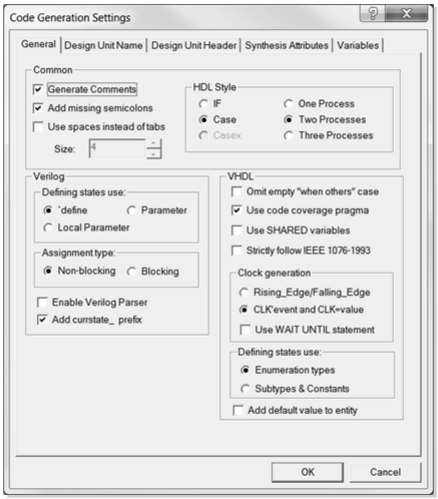


Figure 17: FSM Code Generation Settings - Design Unit Name

Code Generation Settings



To generate a code from a state diagram, you can set several HDL styles.

You can decide whether to use the if or case statements in the state register description, additionally, you can choose the final form of your state machine logic, in whether it will be described by using one, two, or three processes, the Two Process for is the default.

Users can control the header and comments insertion in the generated code.

The State Diagram Editor allows you to choose the clock specification in the generated code.

The State Diagram Editor also allows designers to use blocking or non-blocking assignments in the generated code.

The General tab gives the user control of the following:

- **G**enerate Comments

  If this option is enabled then the FSM object comments are saved in the generated code. This option is set by default.

- **A**dd missing semicolons

  Adds missing semicolons to generated HDL code. The following rules apply:

  ☐ **V**HDL

  When the Add missing semicolons check box is cleared, instructions provided by the user are transferred without any modifications to the output code. When the check box is selected, missing semicolons are added to the