Multiple State Machines



www.aldec.com

v2.1

ALDEC.

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 Architecture Support

98	Multip	le Architectures Su	pport
l		Code Generation Settings General Design Unit Name Design Unit Header Synthesis Attribut Entity: control Architecture: control_arch () Generate Architectures Only	e Veriables
ww.aldec.com	26 27 29 30 31 32 34 35 43 34 35 34 35 39 39 42 23 34 45 45 45 45 45 45 45 45 45 45 45 45 45	architecture control_arch of control is SINGULIC ENCODED state machine: Srep0 typ Srep0.ype is (
ž	v2.1	FPGA Design & Verification in Aldec Active-HDL	THE DESIGN VERIFICATION COMPANY

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

Common Co	Header symhesis Abloudes Vanables L Style IF C One Process Case C Two Processes Case C Three Processes Cosex C Three Processes WHDL Mont empty "when others" case IF Use code coverage pragma Use SHARED variables Sticky follow IEEE 1076-1993 Clock generation C Rising_Edge/Falling_Edge If CLK event and CLK-value Use WAIT UNTL statement Dafing states use. E Enumeation types C Subtypes & Constants C Add default value to ently
	OK Cancel



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:

Generate Comments

ww.aldec.com

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

• Add missing semicolons

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

□ VHDL

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