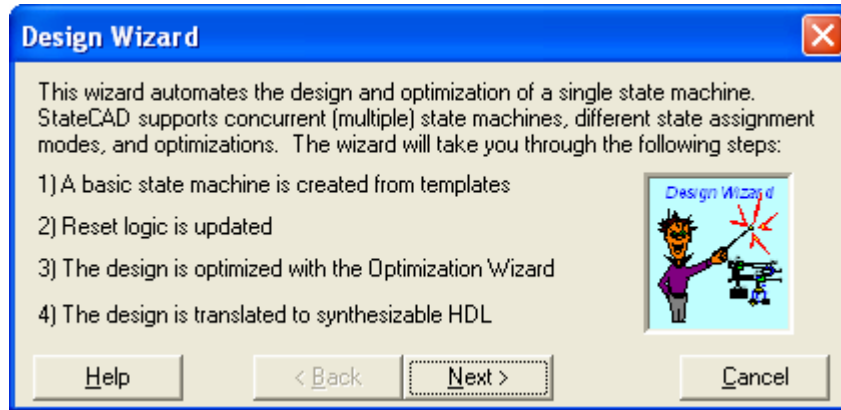


Using StateCAD to generate VHDL from State Diagrams

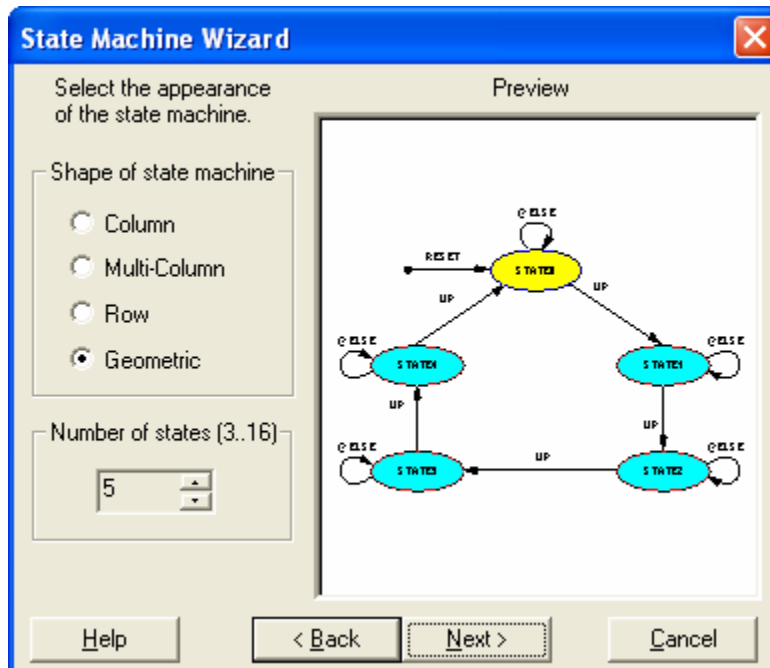
Select **Project => New Source** and select **State Diagram**.

StateCAD starts

Select **File => Design Wizard** or click on the **Draw State Machine** button

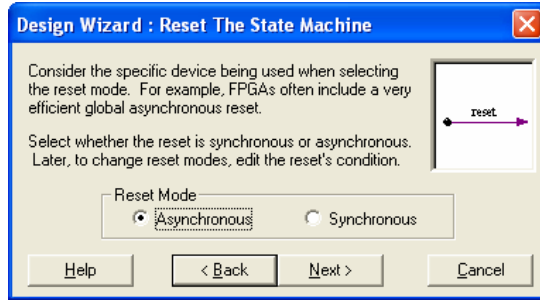


Click on **Next**

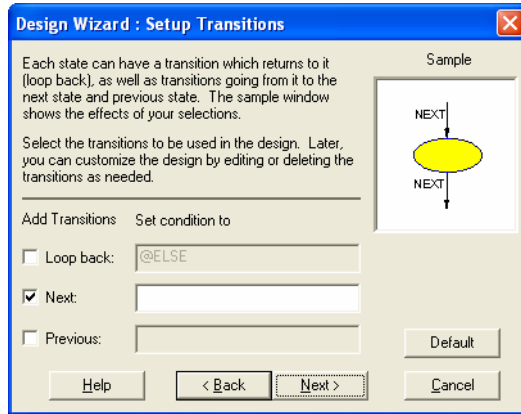


To make our SM1 state machine, select 4 states

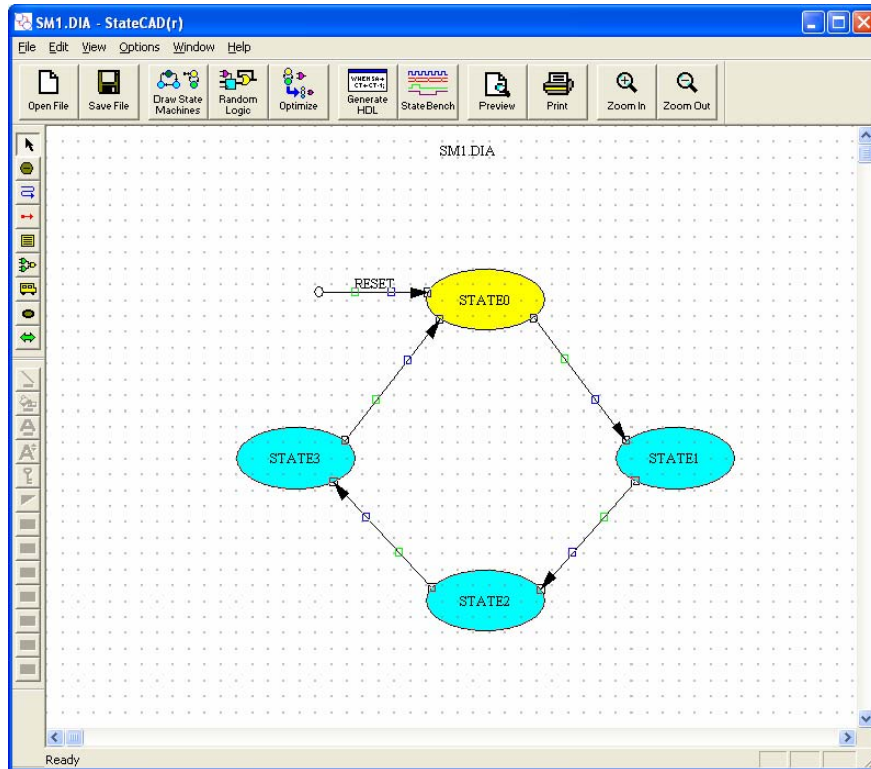
Click on **Next**



Select **Asynchronous** reset then **Next**

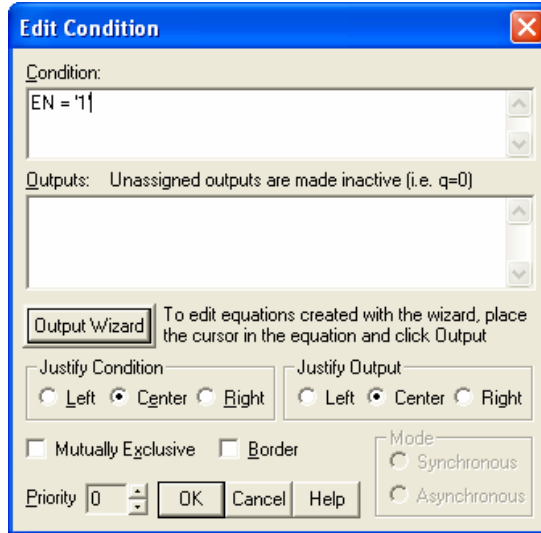


The Optimization Wizard may open at this point (click cancel – we will run this later)

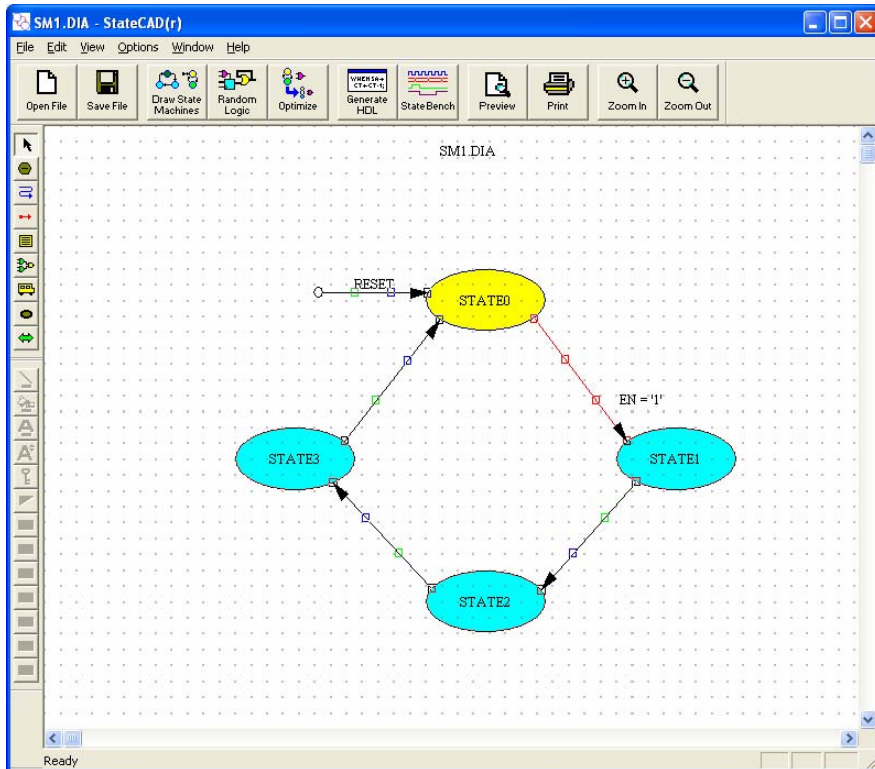


Now we need to add the transitions from state to state.

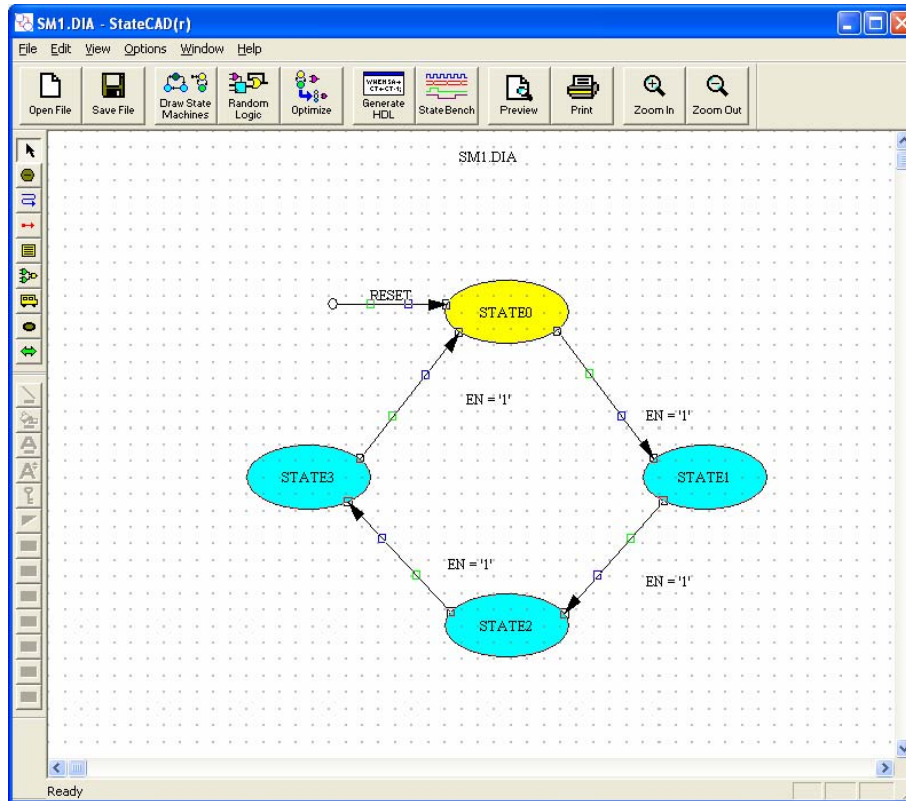
Click on the transition from State0 to State1 and add $EN = '1'$ as shown:



Click on **OK**

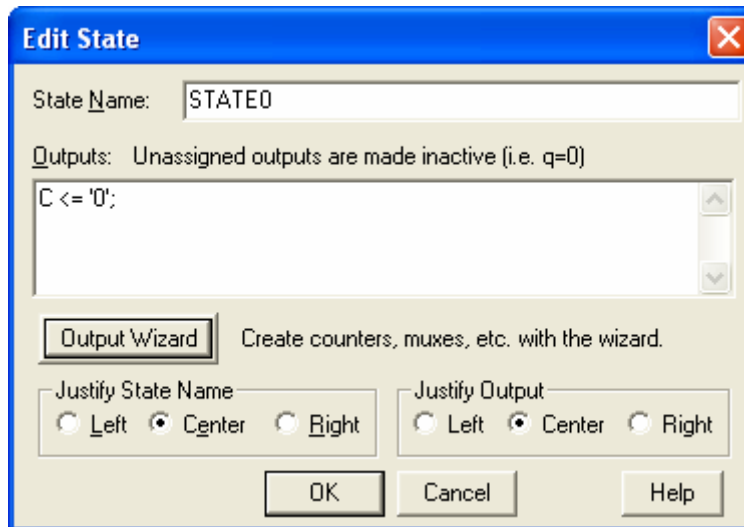


Repeat for the other transitions:

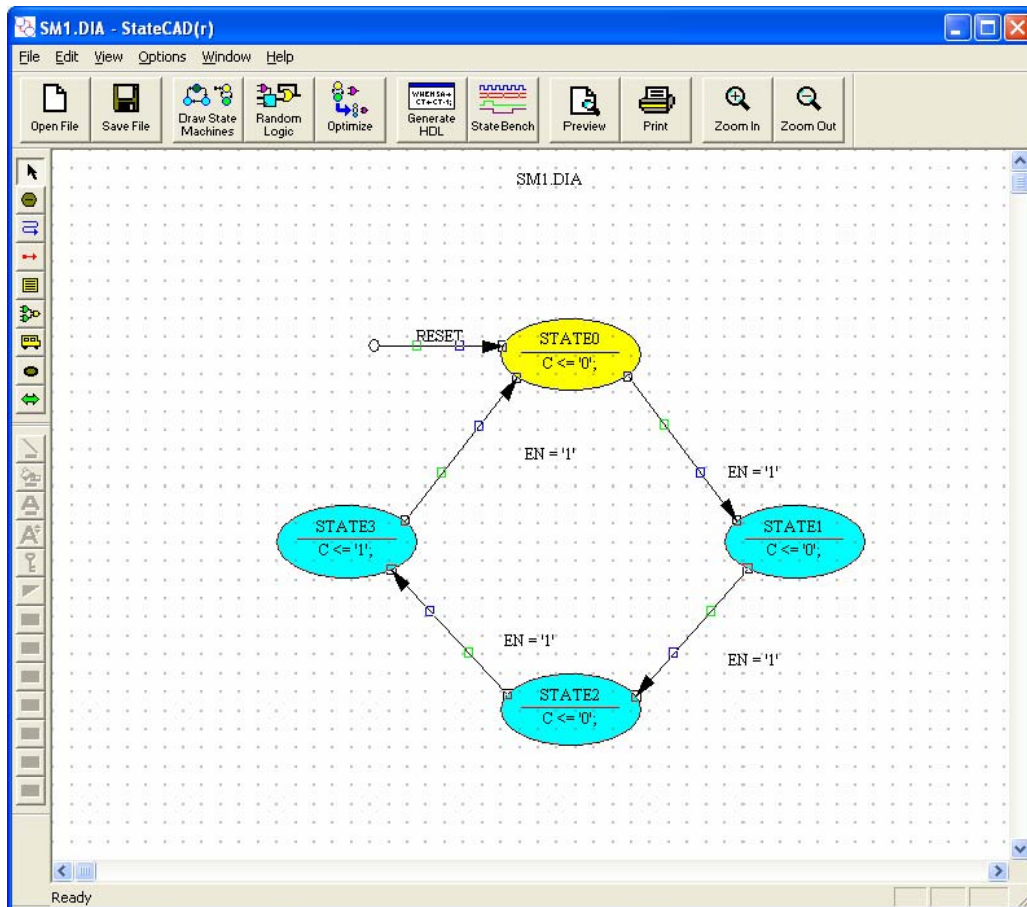


Now we need to set the output conditions:

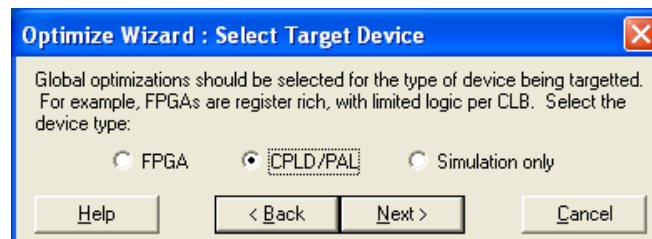
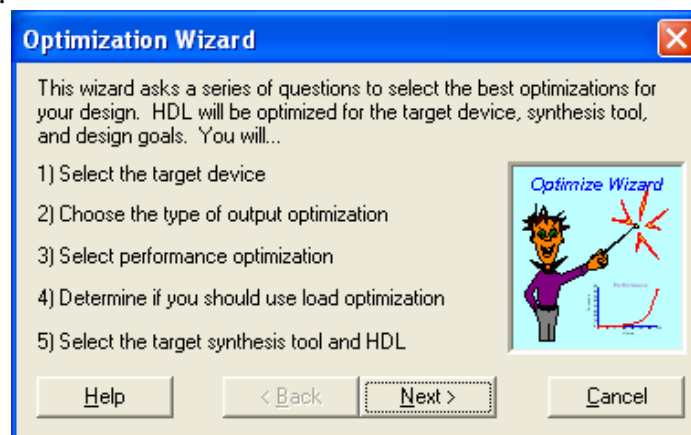
Double-click on State0 and add $C \leq '0'$; as shown



and repeat for the other states



You can go straight to Generate HDL (Options => Compile), but first click on the **Optimize** button.



Optimize Wizard : Optimize Outputs

Speed optimized outputs register the next value. Area optimized outputs are decoded after the clock. Individual outputs may be optimized in Manual mode.

State Machine

Area Optimized (Combinatorial)

Speed Optimized (Registered)

This example above is optimized for area and speed. The state machine yields the same output sequence, but the timing and area vary dramatically.

Manual
 Speed (register outputs)
 Area (decoded outputs)

Optimize Wizard : Improve Performance

StateCAD can guarantee the state machine never gets lost, and all conditions are covered. The price of guaranteed coverage is increased logic utilization, and slower state machines. It is recommended to use implied else to minimize design errors.

The example shows a state with a 'loop back' condition of !UP. If this transition was not present, then when UP=0, what is the next state? Implied else causes the design to remain in the current state.

Guarantee coverage (implied else)
 Maximize speed, reduce area

Optimize Wizard : Optimize Loading

Feedback signals can be automatically buffered to ensure loading changes outside a state machine do not affect its clock speed.

Enable buffering
 Disable buffering

Loading can be reduced by making feedback signals internal nodes. StateCAD flags feedbacks and will make them into internal nodes automatically.

Optimize Port I/O: Make feedback signals internal nodes

Unassigned outputs can be retained between clocks or made inactive. Retaining simplifies registered outputs, but can increase loading.

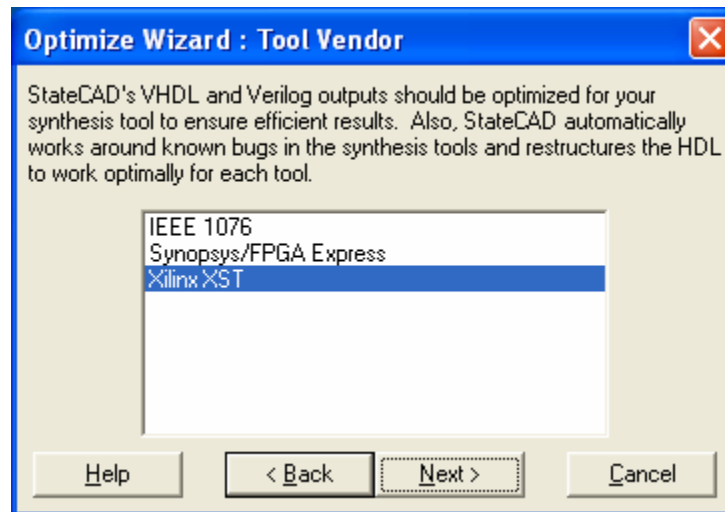
Retain unassigned outputs between clocks

Optimize Wizard : Select Target HDL

StateCAD automatically translates state diagrams to synthesizable and simulatable HDL. Of course, when you need to port a design, just change the target HDL and recompile - StateCAD automatically generates HDL in the new language.

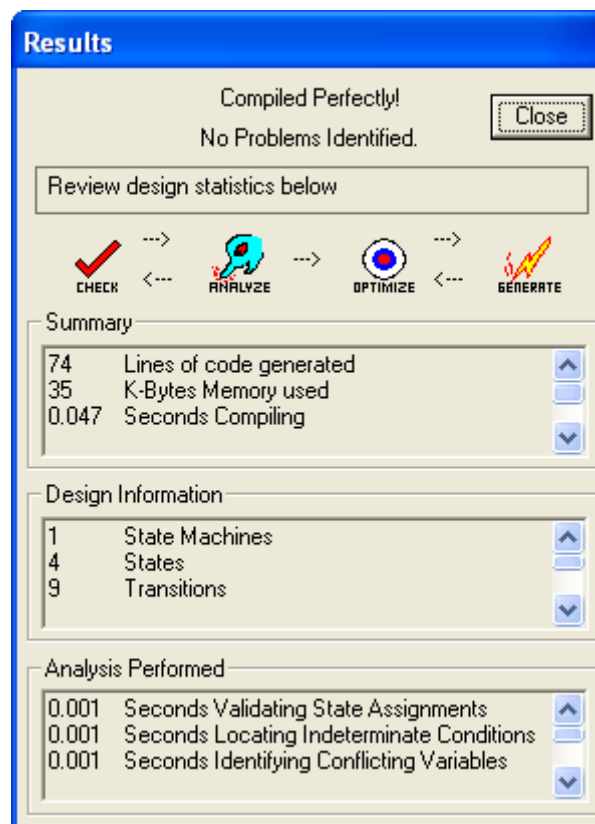
Select the target HDL:

ABEL HDL
 Verilog
VHDL

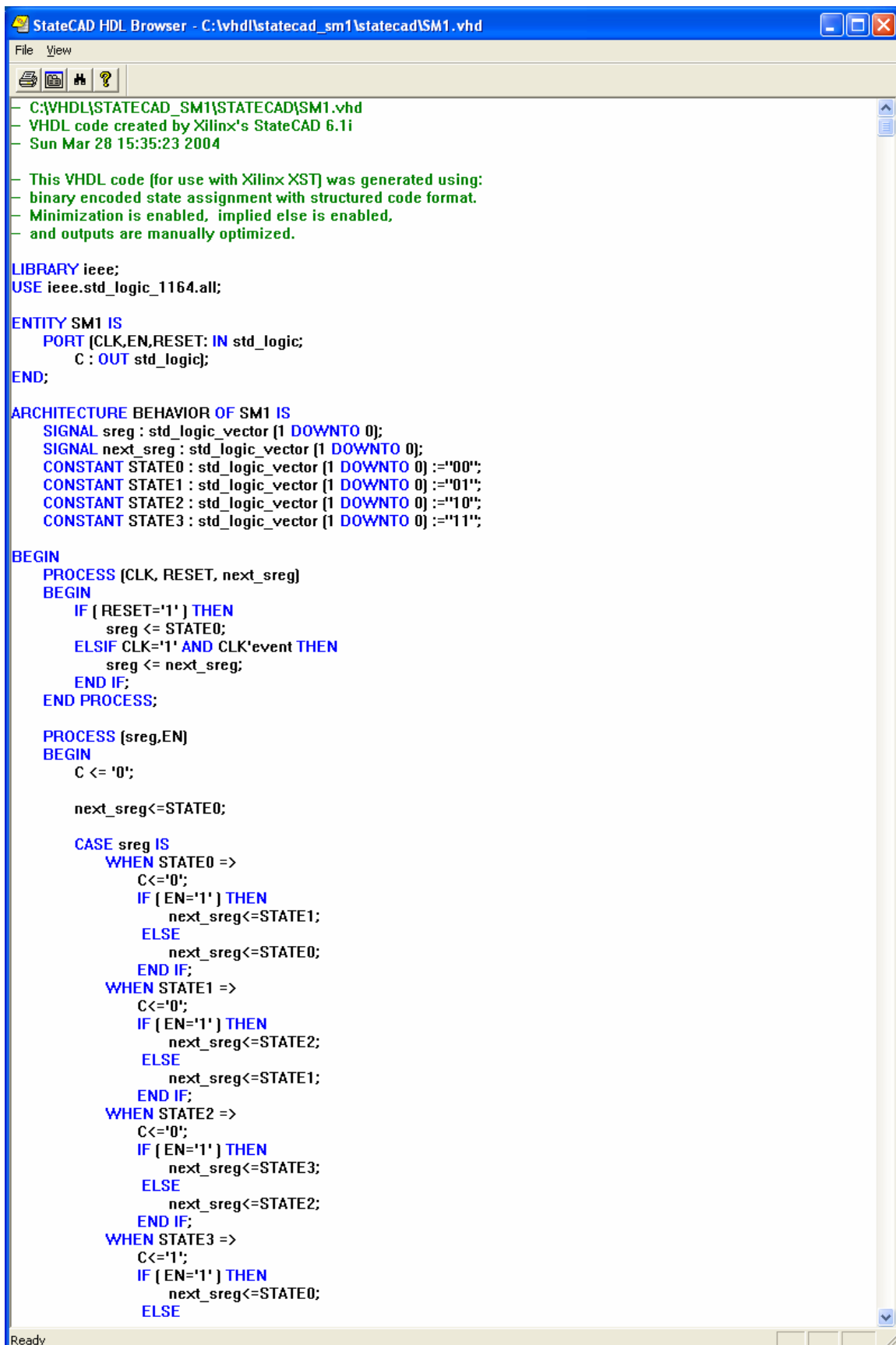


Click on **Next** to end the Optimize wizard.

Click on the **Generate HDL** button:



Click on Close and the HDL code generated is shown:



The screenshot shows a window titled "StateCAD HDL Browser - C:\vhdl\statecad_sm1\statecad\SM1.vhd". The window contains the following VHDL code:

```
File View
- C:\VHDL\STATECAD_SM1\STATECAD\SM1.vhd
- VHDL code created by Xilinx's StateCAD 6.1i
- Sun Mar 28 15:35:23 2004

- This VHDL code (for use with Xilinx XST) was generated using:
- binary encoded state assignment with structured code format.
- Minimization is enabled, implied else is enabled,
- and outputs are manually optimized.

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY SM1 IS
  PORT (CLK,EN,RESET: IN std_logic;
        C : OUT std_logic);
END;

ARCHITECTURE BEHAVIOR OF SM1 IS
  SIGNAL sreg : std_logic_vector (1 DOWNTO 0);
  SIGNAL next_sreg : std_logic_vector (1 DOWNTO 0);
  CONSTANT STATE0 : std_logic_vector (1 DOWNTO 0) := "00";
  CONSTANT STATE1 : std_logic_vector (1 DOWNTO 0) := "01";
  CONSTANT STATE2 : std_logic_vector (1 DOWNTO 0) := "10";
  CONSTANT STATE3 : std_logic_vector (1 DOWNTO 0) := "11";

BEGIN
  PROCESS (CLK, RESET, next_sreg)
  BEGIN
    IF ( RESET='1' ) THEN
      sreg <= STATE0;
    ELSIF CLK='1' AND CLK'event THEN
      sreg <= next_sreg;
    END IF;
  END PROCESS;

  PROCESS (sreg,EN)
  BEGIN
    C <= '0';

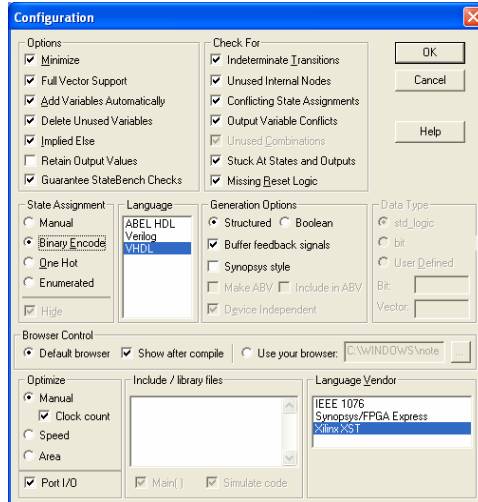
    next_sreg<=STATE0;

    CASE sreg IS
      WHEN STATE0 =>
        C<='0';
        IF ( EN='1' ) THEN
          next_sreg<=STATE1;
        ELSE
          next_sreg<=STATE0;
        END IF;
      WHEN STATE1 =>
        C<='0';
        IF ( EN='1' ) THEN
          next_sreg<=STATE2;
        ELSE
          next_sreg<=STATE1;
        END IF;
      WHEN STATE2 =>
        C<='0';
        IF ( EN='1' ) THEN
          next_sreg<=STATE3;
        ELSE
          next_sreg<=STATE2;
        END IF;
      WHEN STATE3 =>
        C<='1';
        IF ( EN='1' ) THEN
          next_sreg<=STATE0;
        ELSE

```

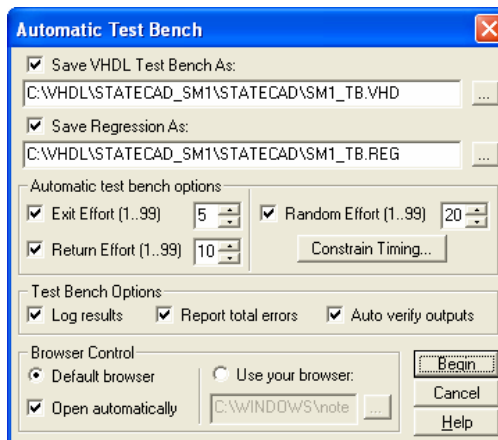
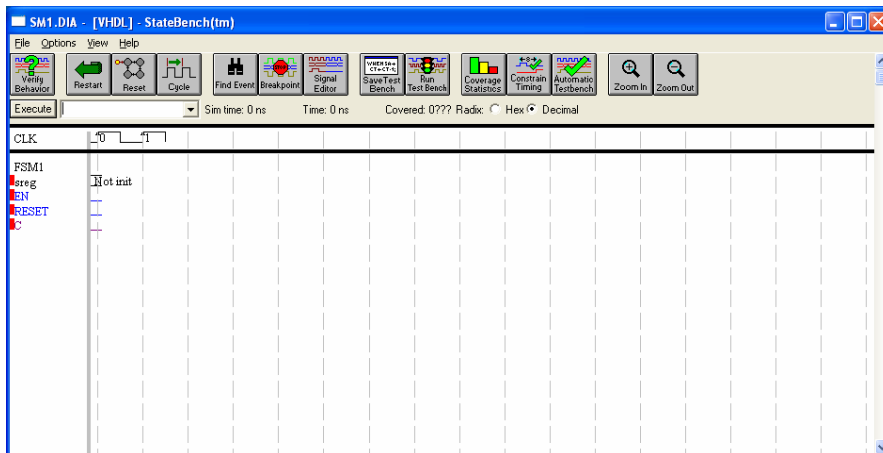
Ready

You can change the generated VHDL by **Options => Configuration**



Click OK or Cancel.

Click on the **State Bench** button then click on **Automatic Test Bench**



Click **Begin**

```

StateCAD HDL Browser - C:\WHDL\STATECAD_SM1\STATECAD\SM1_TB.VHD
File View
C:\WHDL\STATECAD_SM1\STATECAD\SM1_TB.VHD
VHDL testbench created by
Xilinx's StateBench 1.01
Sun Mar 28 15:43:27 2004

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY ieee;
USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE STD.TEXTIO.ALL;

ENTITY testbench IS
END testbench;

ARCHITECTURE testbench_arch OF testbench IS
FILE RESULTS: TEXT IS OUT "results.txt";
COMPONENT SM1
PORT [CLK,EN,RESET: IN std_logic;
C : OUT std_logic];
END COMPONENT;

SIGNAL CLK,EN,RESET: std_logic := '0';
SIGNAL C : std_logic := '0';

BEGIN

UUT : SM1 PORT MAP (
CLK=>CLK,
EN=>EN,
RESET=>RESET,
C->C);

PROCESS
VARIABLE TX_OUT : LINE;
VARIABLE TX_ERROR : INTEGER := 0;

PROCEDURE CHECK_C(
next_C : std_logic
) IS BEGIN
IF (C /= next_C) THEN
write[TX_OUT,string'(
** Error, C=)];
write[TX_OUT, C];
write[TX_OUT, string(' Expected = ');
write[TX_OUT, next_C];
write[TX_OUT, string('**)];
writeline(results, TX_OUT);
TX_ERROR := TX_ERROR + 1;
END IF;
ASSERT (C=next_C) REPORT
"Error, C has incorrect value"
SEVERITY ERROR;

END;

BEGIN

-----
CLK <= '0'; -- Initialize clock inactive
EN <= '0';
RESET <= '1';
WAIT FOR 15 ns;
-----

CLK <= '1'; -- Clock 0 Time 15 ns
WAIT FOR 20 ns;
CHECK_C('0');
WAIT FOR 30 ns;
CLK <= '0'; -- inactive clock edge
WAIT FOR 35 ns;

```

Ready

