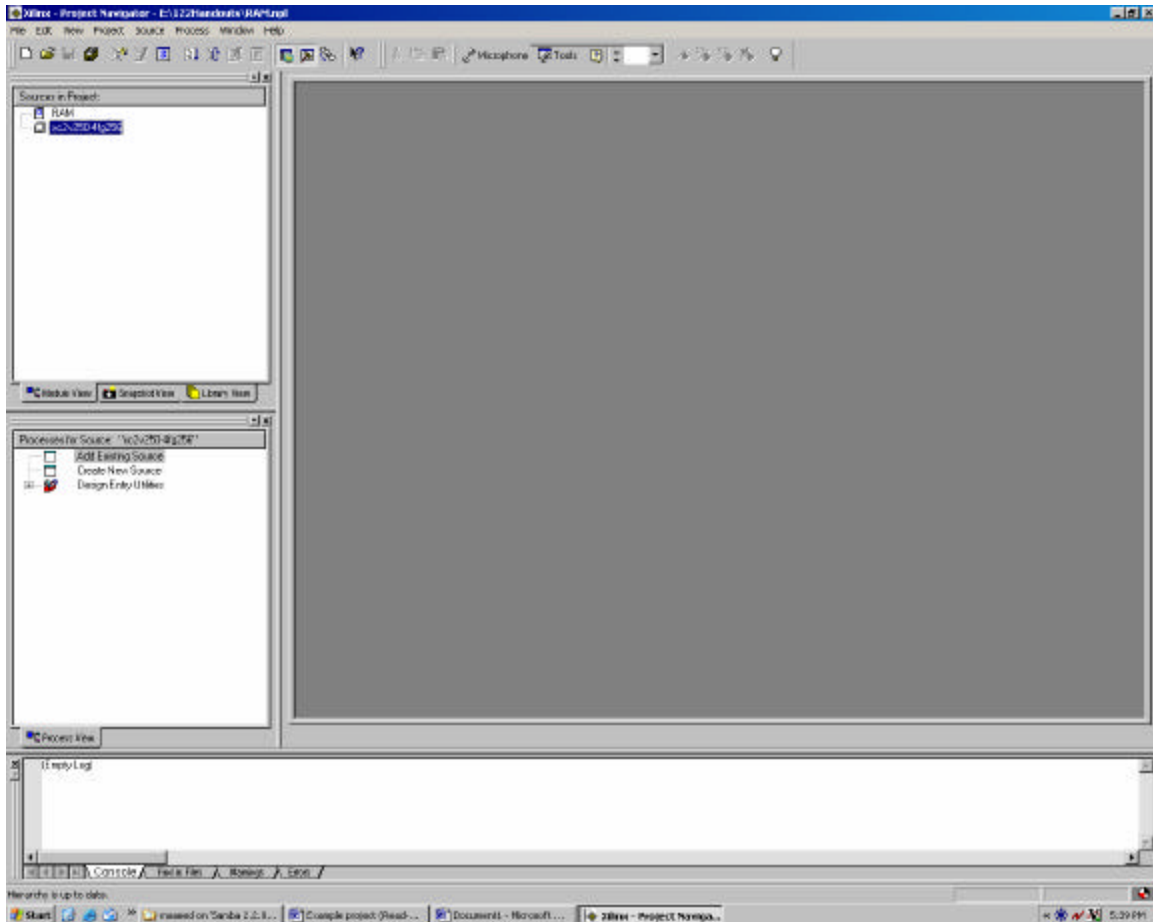


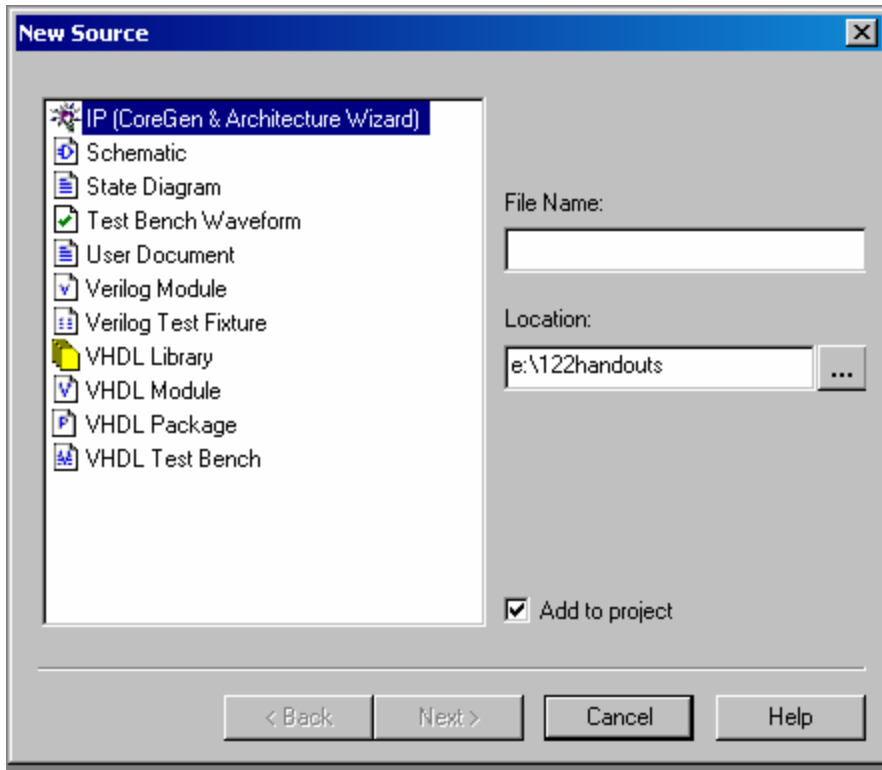
## LAB2: CORE GENERATOR

1. Create a new Verilog project in Xilinx ISE 6.2i, and name it RAM.



2. Go to “Project -> New Source”

You will see the following dialog box

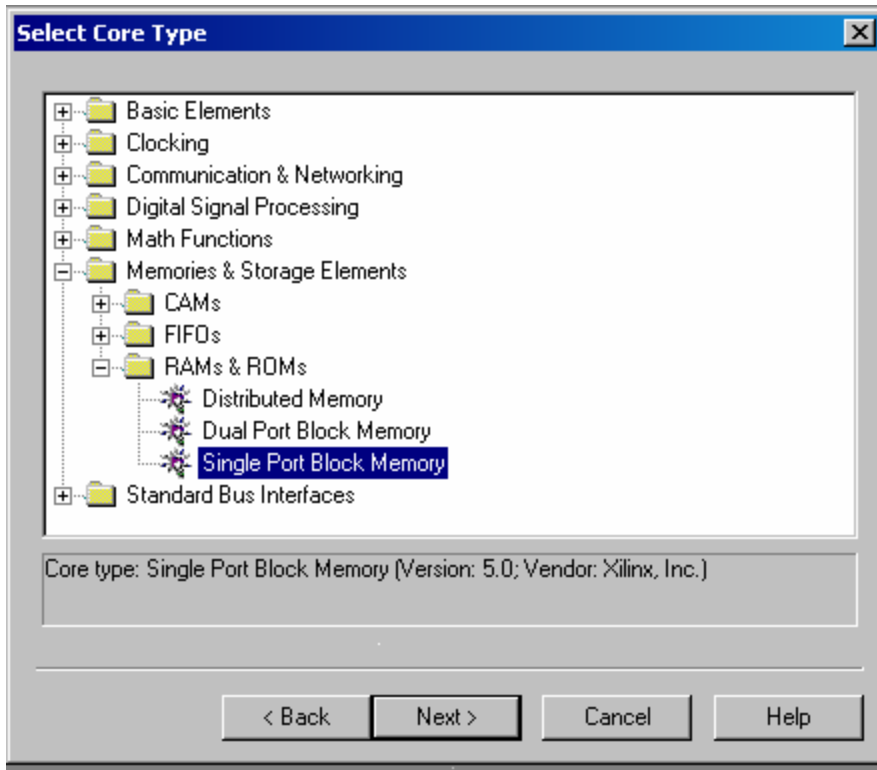


3. Select IP(CoreGen & Architecture Wizard).

Make sure that the “Add to project” checkbox is checked.

3. Give your file some name (SPRAM).

4.ClickNext.



5. From the “select core type” window, select Memories & Storage Elements->RAMs & ROMs->Single Port Block Memory

6. Click Next

7. Click Finish.

You will see the following Single Port Block Memory GUI:

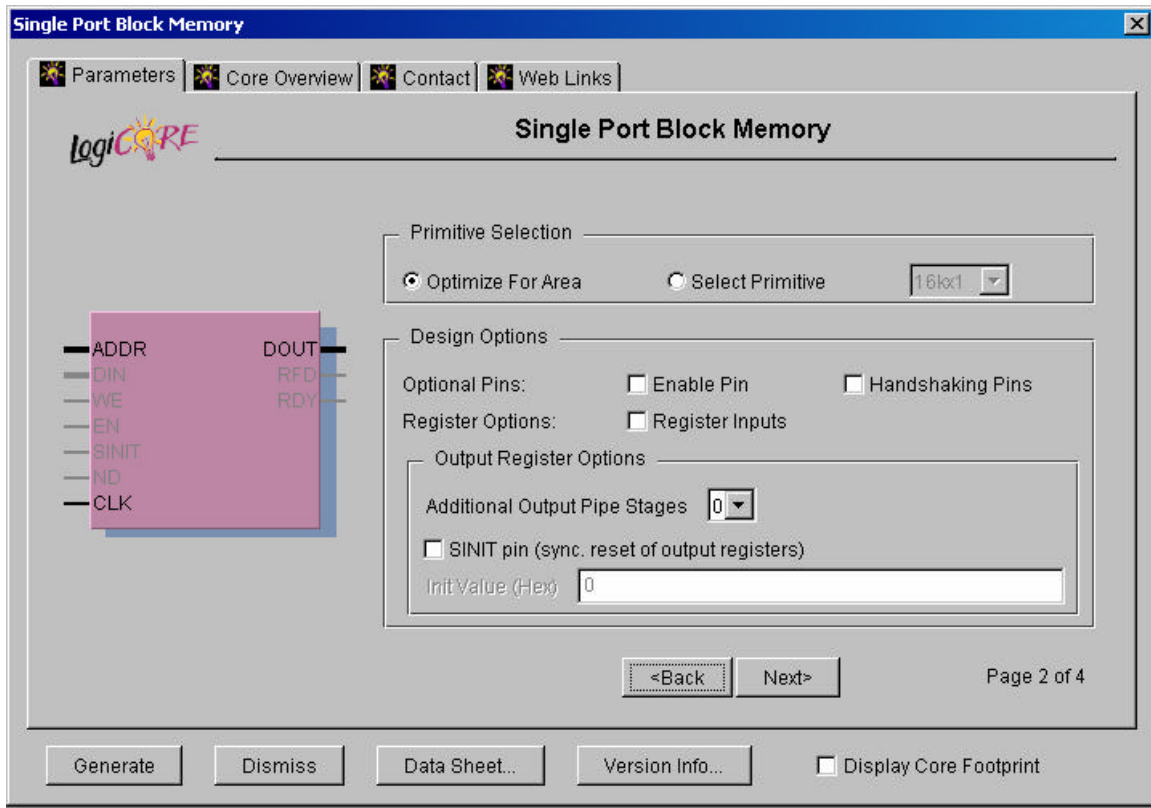
The screenshot shows the 'Single Port Block Memory' configuration window. At the top, there are tabs for 'Parameters', 'Core Overview', 'Contact', and 'Web Links'. The 'Parameters' tab is active. On the left, there is a diagram of the memory block with input/output ports: ADDR, DIN, WE, EN, SINIT, ND, CLK on the left; DOUT, RFD, RDY on the right. The main configuration area includes: 'Component Name' set to 'spram'; 'Port Configuration' with 'Read Only' selected; 'Memory Size' with 'Width' set to 8 (Valid Range 1..256) and 'Depth' set to 16 (Valid Range: 2..262144); and 'Write Mode' with 'Read After Write' selected. At the bottom, there are buttons for '<Back', 'Next>', 'Generate', 'Dismiss', 'Data Sheet...', 'Version Info...', and a checkbox for 'Display Core Footprint'. The page number 'Page 1 of 4' is displayed in the bottom right corner.

8. Fill out the “Component Name”, “Width” and “Depth” text box accordingly.

9. Select a Port Configuration.

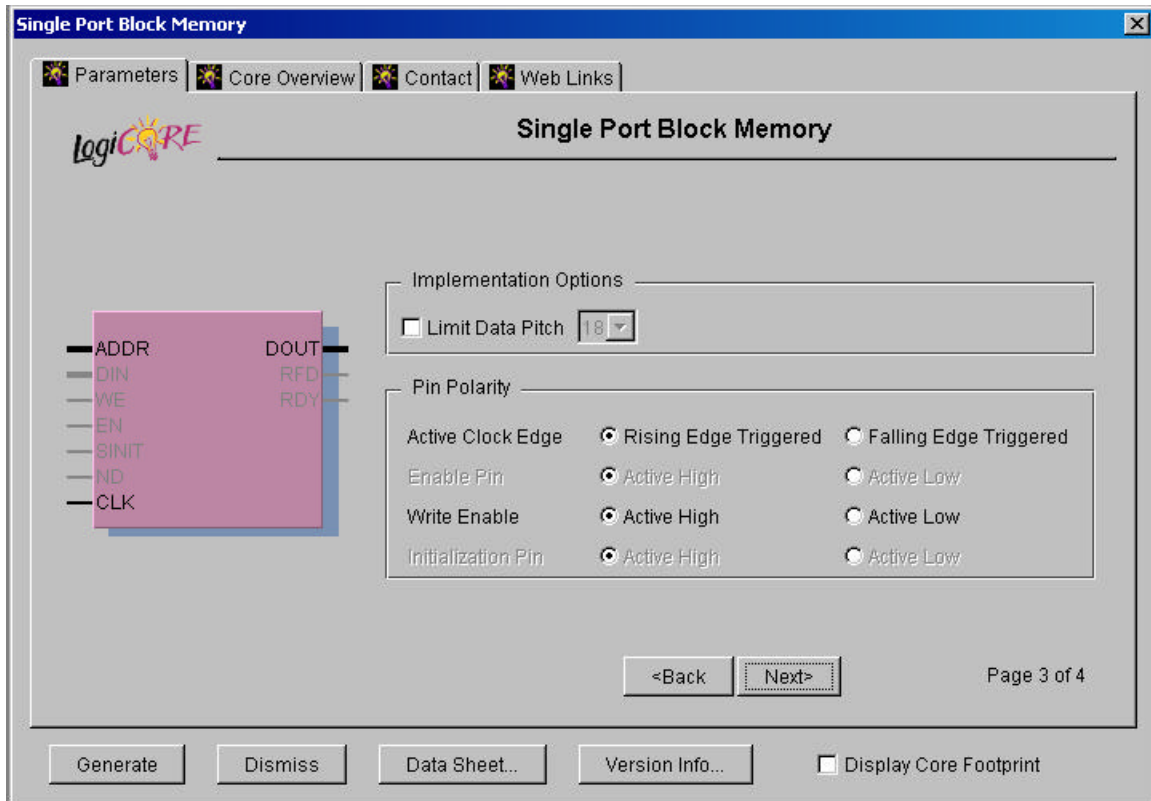
10. Click Next.

You will see the following window



11. Click Next.

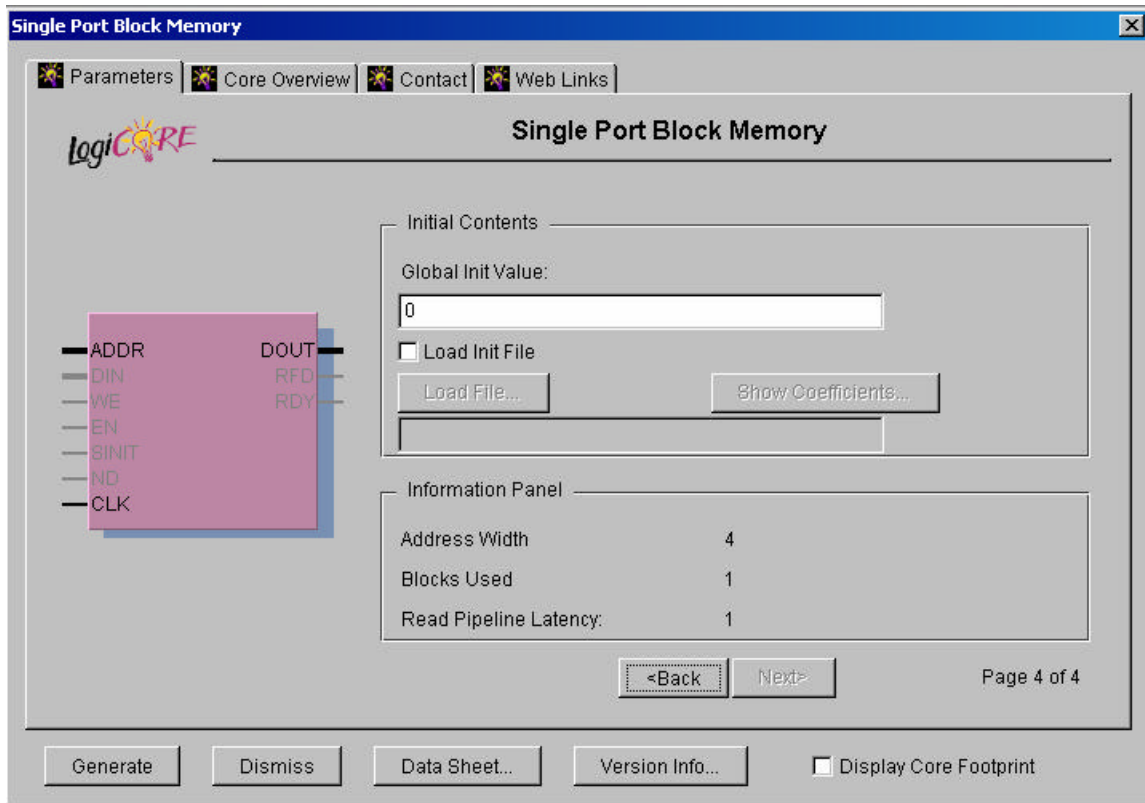
You will see the following window:



12. Select the desired pin polarities.

13. Click Next.

You will see the “initial content” portion of the GUI:



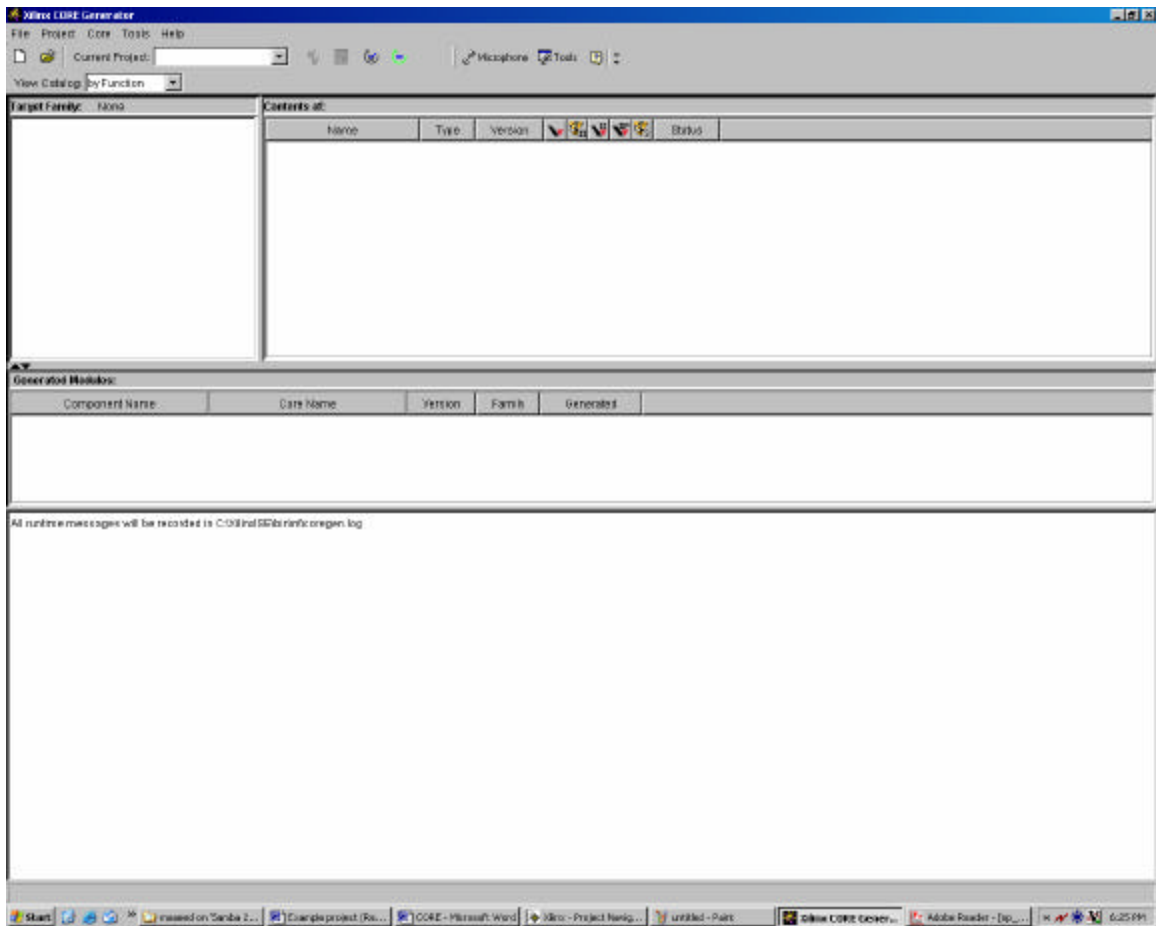
#### 14. Click Generate.

This will generate the memory with initial content “0” for all the addresses.

Now we will use the memory editor to make a COE file and load it in the memory to have some actual data in each memory address.

#### 15. Open the CORE GENERATOR utility (Start-> All Programs -> Xilinx ISE 6-> Accessories->CORE GENERATOR).

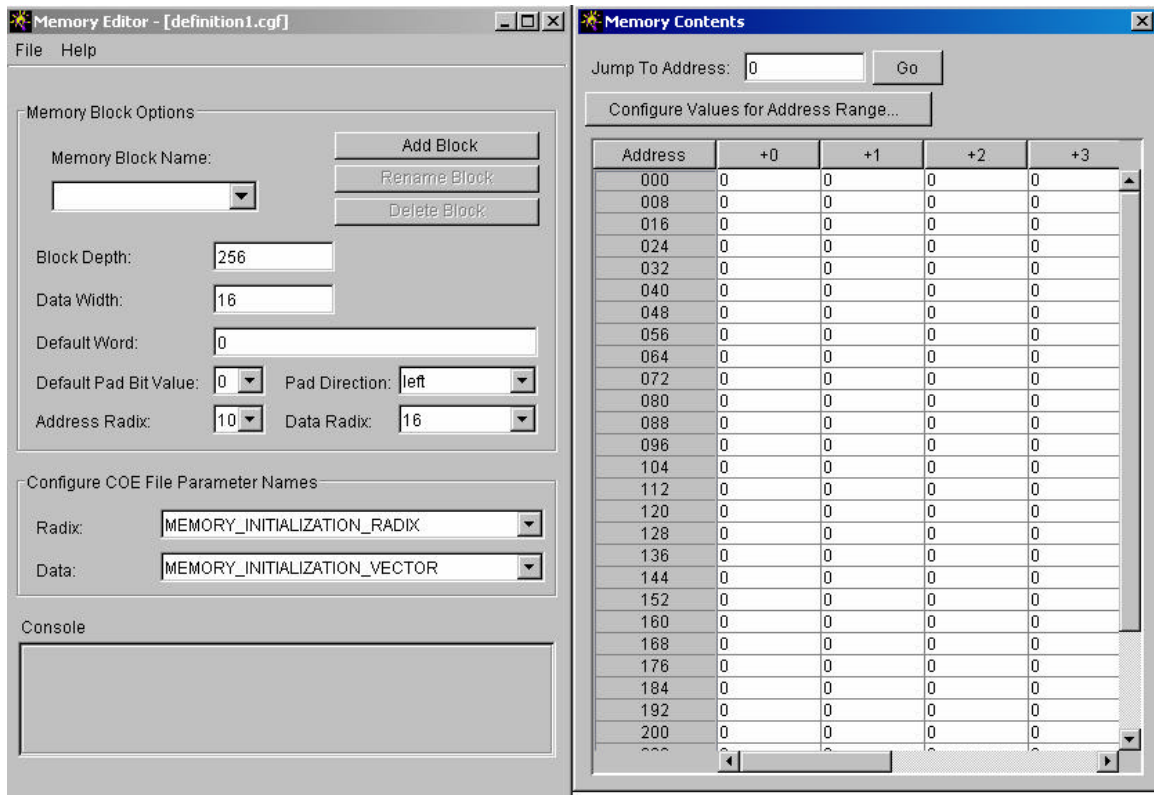
This gives you the following window:



## 16. Goto Tools->Memory Editor.

A memory editor GUI will pop up.



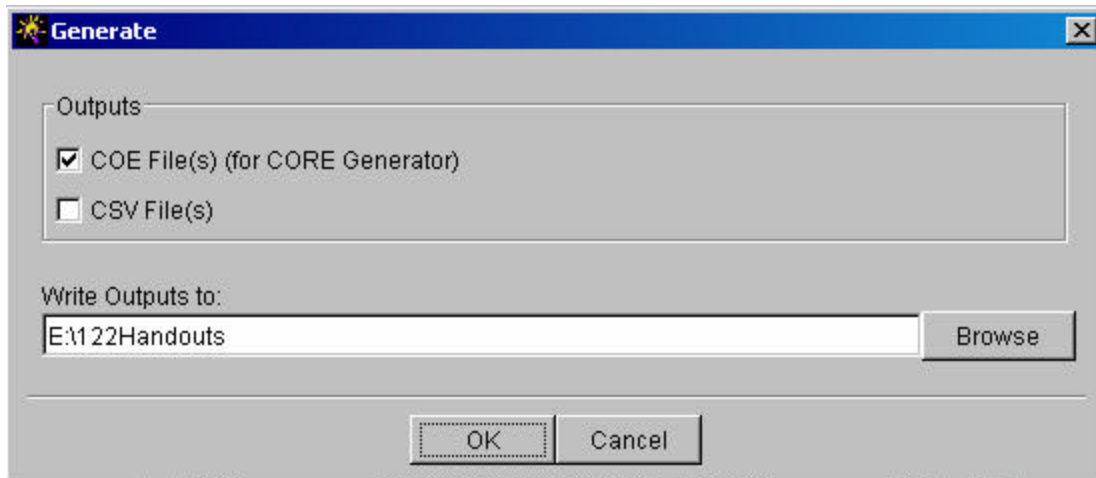


17. Click Add Block.

18. Give your block a name (spram).

19. Click OK.



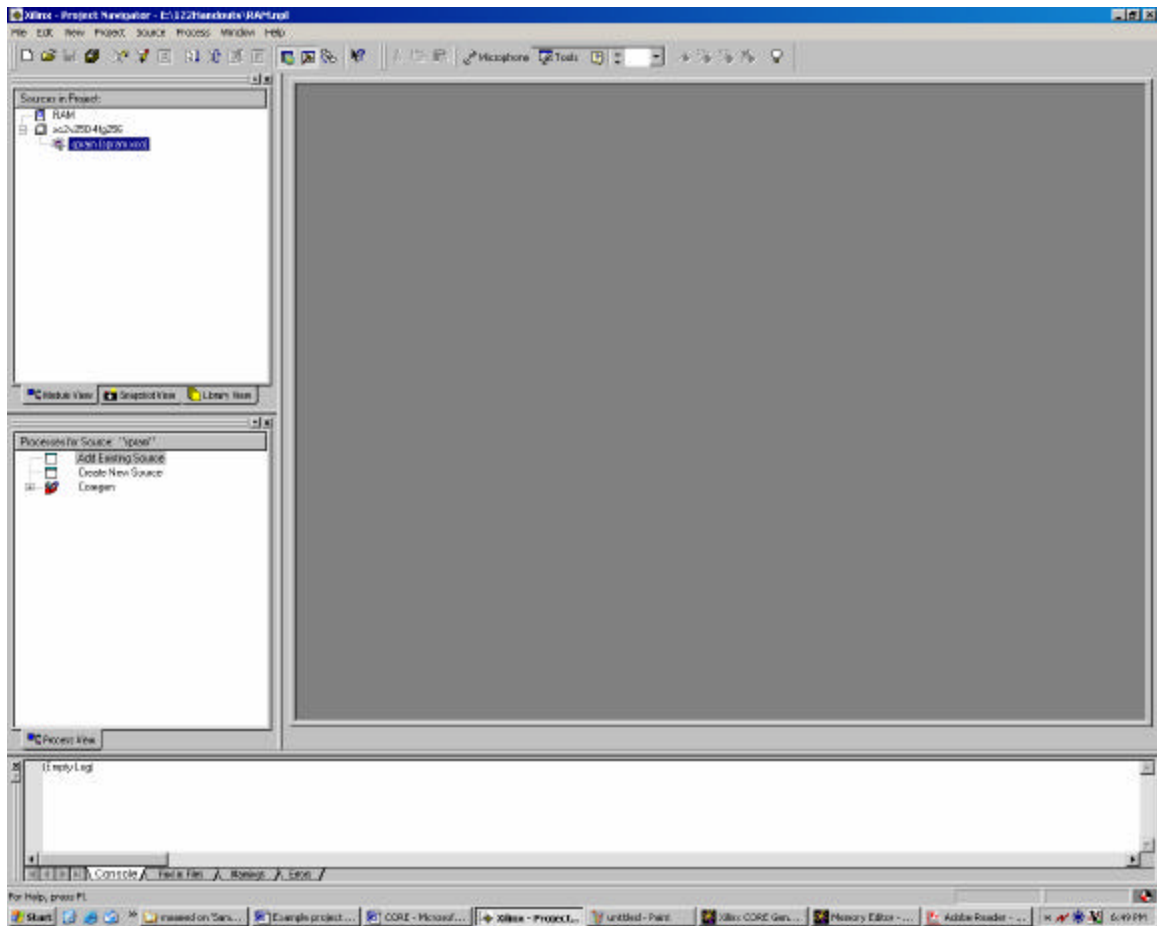


25. Select the COE file Option.

26. Click OK.

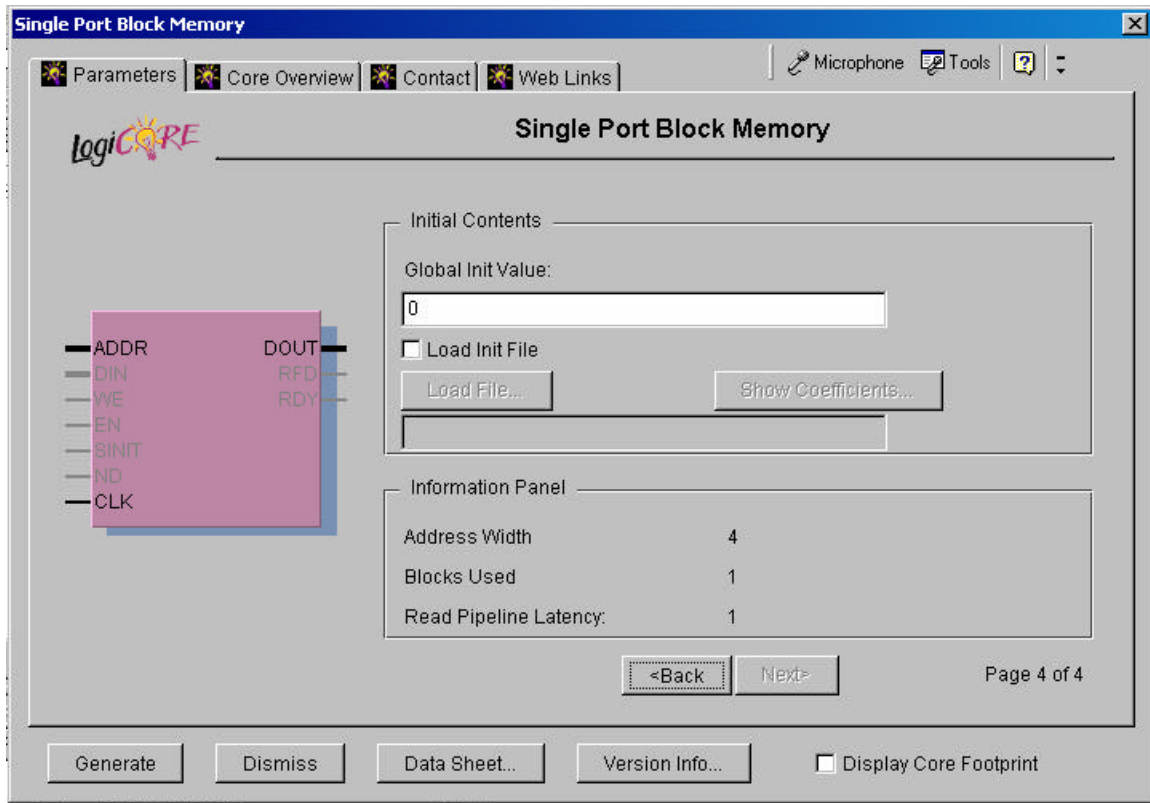
This will generate the .COE file for you.

Now go back to your project in the project navigator.

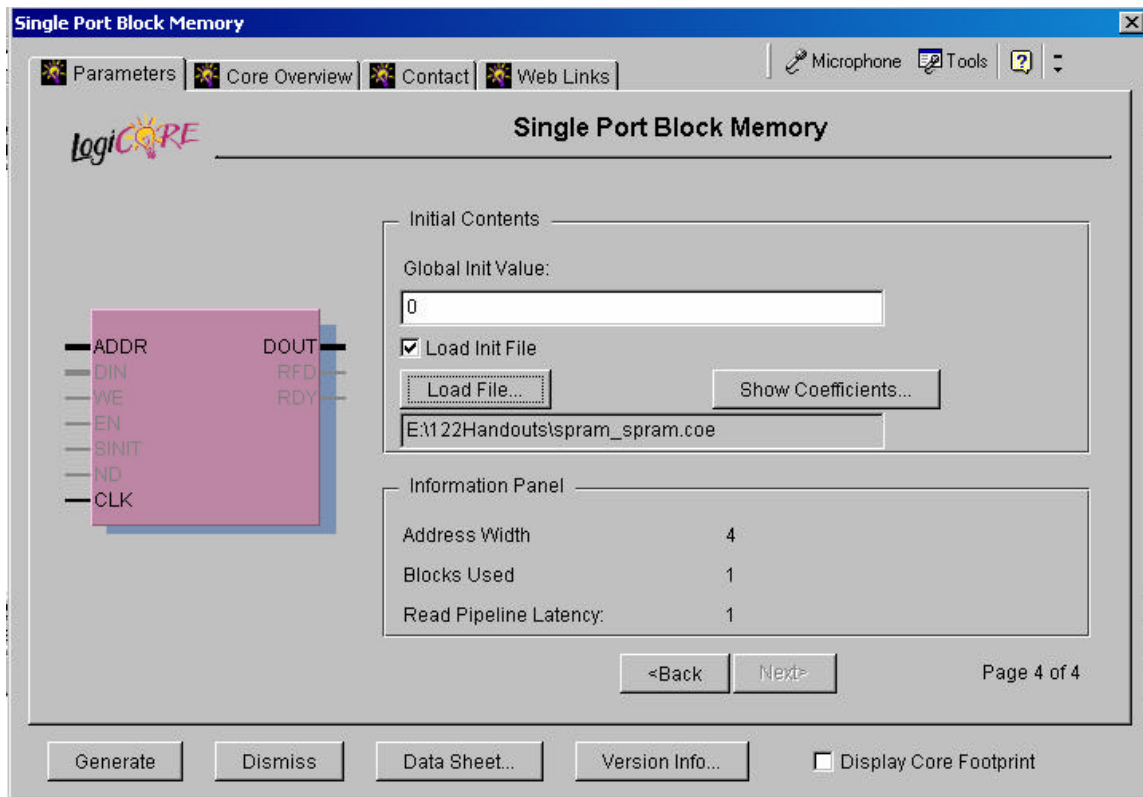


27. Double click on your CORE (spram).

You should see the CORE GUI.



28. Check the “Load Init File” checkbox and select the .COE file that you generated earlier.



29. Click Generate.

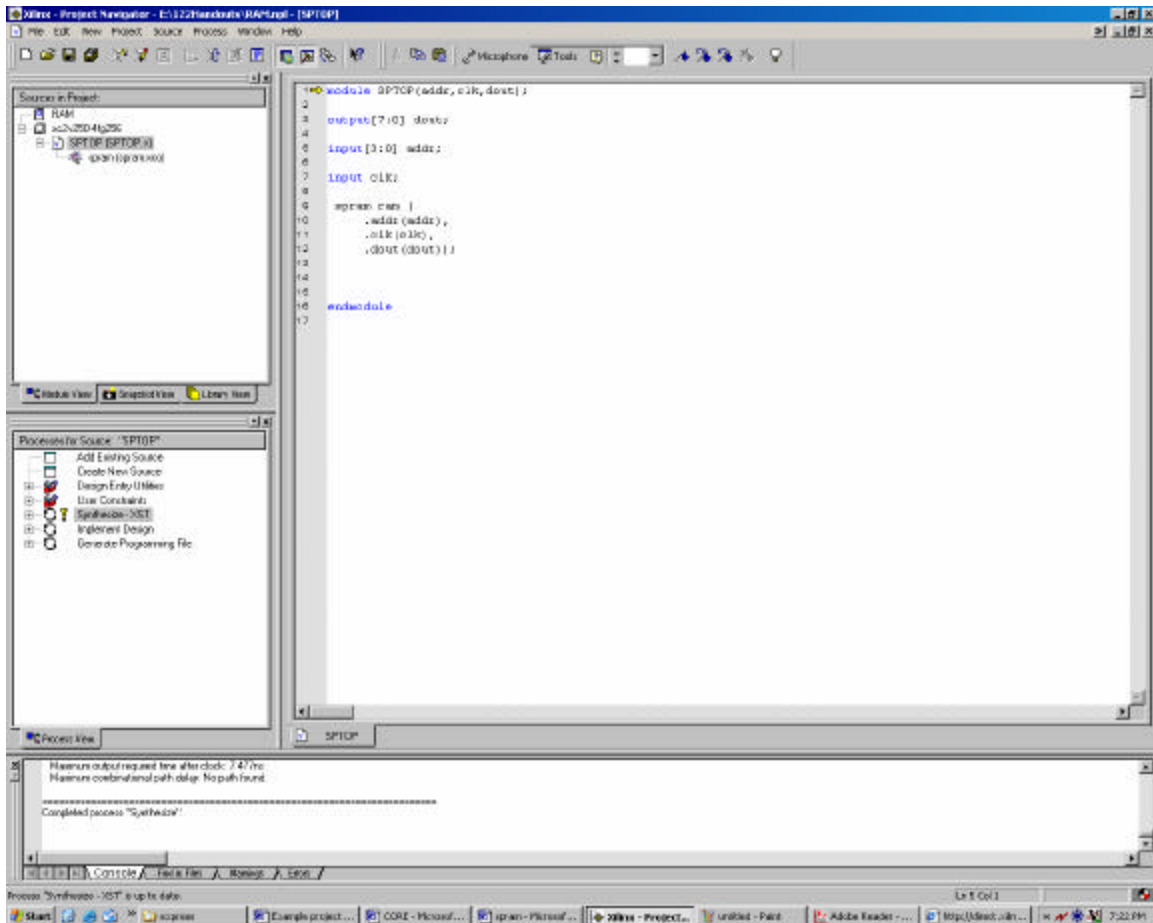
Now we will write the top module in verilog and instantiate this memory block.

30. Create a new verilog source file in the project navigator with the appropriate input and output ports (for spram your input is ADDR, CLK and output is DOUT).

31. To instantiate the spram open the “.veo “, which was created when you generated the CORE and copy.

32. Highlight the inserted code in .veo from  
 “//----- Begin cut here for INSTANTIATION TEMPLATE ---  
 //”  
 to  
 “// INST\_TAG\_END ----- End INSTANTIATION Template”

33. Copy and paste this code in your top level verilog source code file.



34. Synthesis your code.

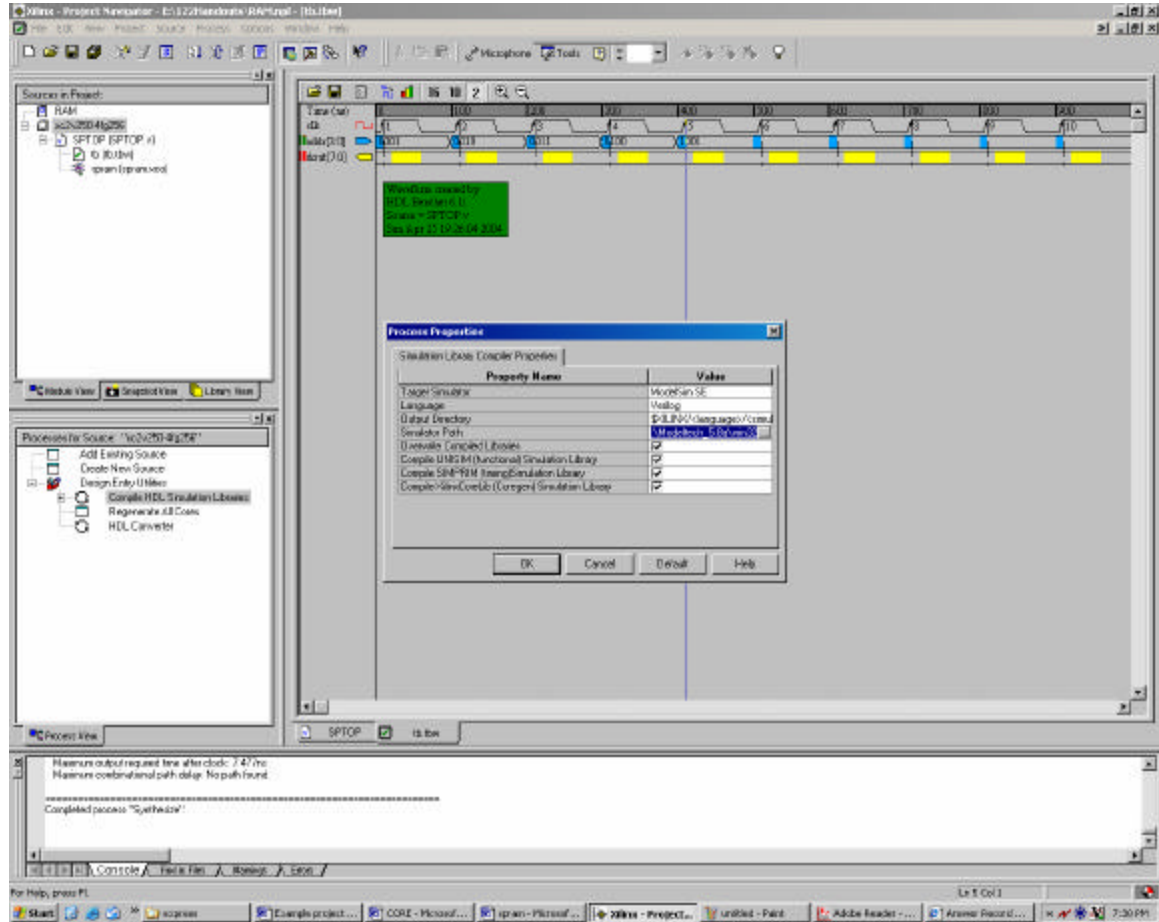
35. Now simulate your code.

Note: Since you are using a CORE model sim will need the xilinxcorelib. If the libraries are not compiled follow the direction below to compile the libraries from the project GUI.

#### Running CompXLib from the Project Navigator GUI (6.1i or later)

1. Click the appropriate device (e.g., xcv50-6bg256) in the Sources in Project Window.
2. Right-click Compile HDL Simulation Libraries in the Processes for Source Window.
3. Select Properties.

4. Set the applicable Simulation Library Compiler Properties (C:\Modeltech\_5.8a\win32).



5. Click OK.

6. Double-click Compile HDL Simulation Libraries in the Processes for Source Window.