APPLICATION NOTE

Xilinx has acquired the entire Philips CoolRunner Low Power CPLD Product Family. For more technical or sales information, please see: www.xilinx.com

XAPP 304

Probing internal nodes using XPLA software graphic simulator

Author: *B. Wade Baker, Philips Semiconductors*

1998 Jun 30

Probing internal nodes using XPLA software graphic simulator

XAPP 304

Author: B. Wade Baker, Philips Semiconductors

Many times, when using a simulator to explore the dynamics of your latest CPLD design, you would like to be able to see what is going on between the inputs and outputs. The Philips XPLA Software Simulator allows you to do just that by providing a method of displaying internal nodes directly as part of the waveform viewer. This application note describes how to accomplish this task.

XPLA Software allows you to do both functional and timing simulations. The functional simulation does not include part specific information since it is a generic check of your work and could be applied to any part in the Philips Coolrunner family. The timing simulation contains part specific information based upon the particular device you selected before your design was fitted.

The XPLA Simulator constructs a simulation of your design's logic based upon a binary netlist (.bin) file created by the simulator and an input stimulus (.scl) file created when you manipulate the voltage levels for the input signals displayed on the waveform viewer window. When the simulator is commanded to run in the functional mode it combines the .bin file with the .scl file and creates a net (.net) file. The net file generates the voltage levels of the outputs based upon your logic's design parameters. Figure 1 displays the .net file for the 3 bit counter demo design that comes with XPLA Software.

As you can see from Figure 1, there are not any internal nodes to check because this is the functional simulation file. Any signal you picked to display in the waveform viewer would already be present, therefore the idea of probing internal nodes does not apply to functional simulations.

For timing simulations the simulator combines the .scl and .bin files with a .mod file. The .mod file combines part specific information such as; package type, voltage level, speed, and density with design specific information such as how many levels of logic, PAL or PAL/PLA delay, and anything else that is applicable in accurately simulating the design. Figure 2 is the .mod file for the Demo design using a PZ3032–8 PLCC44. In Figure 3 we have a schematic representation of the internal workings of the Coolrunner family of parts. This schematic can be used to interpret the .mod file contents.

```
* pla2net.exe Created on:Mon Mar 03 12:54:10 1997
* Input File  : d:\xpla\example\demo\demo.pla
* Output File : d:\xpla\example\demo\demo.net
*
 NETSTART
*
CLOCK_P AND I(CLOCK) O(CLOCK_P)
RESET_P AND I(RESET) O(RESET_P)
bit0_N AND I(bit0_Q) O(bit0_N)
bit1_N AND I(bit1_Q) O(bit1_N)
PT0 AND I(CLOCK_P) O(PT0)
PT1 AND I(RESET_P) O(PT1)
PT2 AND I(bit0_N) O(PT2)
PT3 AND I(bit0_N, bit1_N) O(PT3)
bit0_AR  OR I(PT1) O(bit0_AR)
bit1_AR  OR I(PT1) O(bit1_AR)
bit2_AR  OR I(PT1) O(bit2_AR)
bit0_C  OR I(PT0) O(bit0_C)
bit1_C  OR I(PT0) O(bit1_C)
bit2_C  OR I(PT0) O(bit2_C)
bit0_T NOR I(GND) O(bit0_T)
bit1_T  OR I(PT2) O(bit1_T)
bit2_T  OR I(PT3) O(bit2_T)
bit0_Q TFFSR I(bit0_T, bit0_C, GND, bit0_AR) O(bit0_Q)
bit1_Q TFFSR I(bit1_T, bit1_C, GND, bit1_AR) O(bit1_Q)
bit2_Q TFFSR I(bit2_T, bit2_C, GND, bit2_AR) O(bit2_Q)
bit0 TRIBUF I(VCC, bit0_Q) O(bit0)
bit1 TRIBUF I(VCC, bit1_Q) O(bit1)
bit2 TRIBUF I(VCC, bit2_Q) O(bit2)
*
 NETEND
*
 NETIN VCC, GND, RESET, CLOCK
 NETOUT bit0, bit1, bit2
```

Figure 1. Demo Design Net File

Xilinx has acquired the entire Philips CoolRunner Low Power CPLD Product Family. For more technical or sales information, please see: www.xilinx.com

```
*
* Created on:Wed Mar 05 10:39:56 1997
* Input File  : demo.jed
* Output File : demo.mod
*
 NETSTART
*
XCTA__4       AND           I(XPIN1__B)                                 O(XCTA__4)
bit2          XOUTBUF15     I(VCC, XFF__A0)                             O(bit2)
XFF__A0       XDFF3032_8    I(XSUM__A0, VCC, XCLK__0 , GND, XCTA__4)    O(XFF__A0)
XPALA0__0     AND           I(XNODEA2__B, XNODEA1__B)                   O(XPALA0__0)
XSUM__A0      OR            I(XPALA0__0)                                O(XSUM__A0)
bit1          XOUTBUF15     I(VCC, XFF__A1)                             O(bit1)
XFF__A1       XDFF3032_8    I(XSUM__A1, VCC, XCLK__0 , GND, XCTA__4)    O(XFF__A1)
XPALA1__0     AND           I(XNODEA2__B)                               O(XPALA1__0)
XSUM__A1      OR            I(XPALA1__0)                                O(XSUM__A1)
bit0          XOUTBUF15     I(VCC, XFF__A2)                             O(bit0)
XFF__A2       XDFF3032_8    I(XSUM__A2, VCC, XCLK__0 , GND, XCTA__4)    O(XFF__A2)
XSUM__A2      NOR           I(GND)                                      O(XSUM__A2)
XPIN1__B      XINBUF45      I(RESET)                                    O(XPIN1__B)
XNODEA1__B    XINBUF45      I(XFF__A1)                                  O(XNODEA1__B)
XNODEA2__B    XINBUF45      I(XFF__A2)                                  O(XNODEA2__B)
XCLK__0       XCKBUF        I(CLOCK)                                    O(XCLK__0 )
*
 NETEND
*
 NETIN RESET, CLOCK, VCC, GND
 NETOUT       bit2, bit1, bit0
*
```

Figure 2. Demo Design Mod File

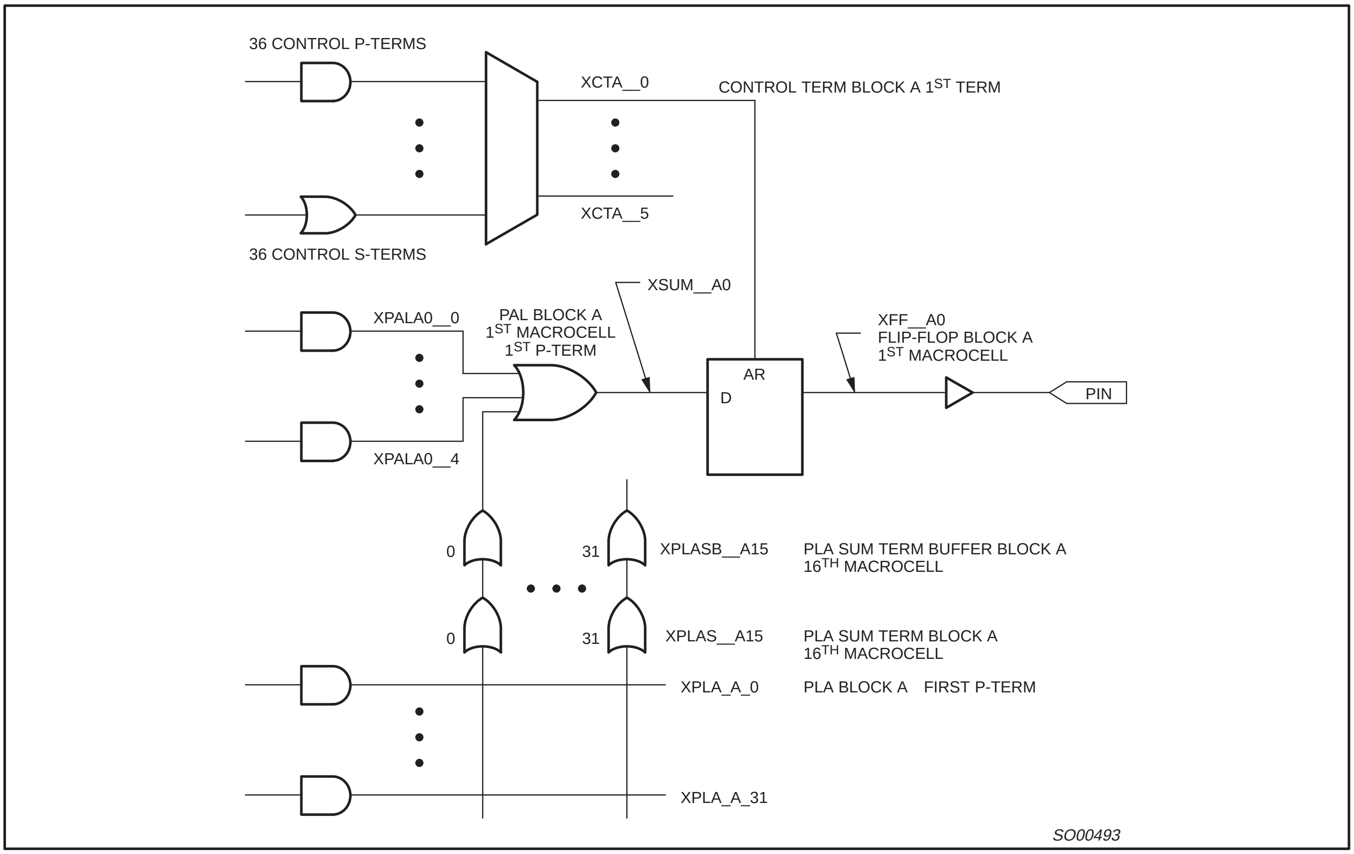

Figure 3. Internal Node Schematic

There are 36 control product terms and 36 control sum terms which feed the 6 Control terms per logic block. It is important to note that control terms can be **ONLY** sum terms **OR** product terms, **NOT BOTH**. For all of the names of internal nodes listed in Figure 3, the X at the beginning of the name represents XPLA. XCTA__0 through XCTA__5 , located at the top right in Figure 3, are those control terms. The CT is for Control Term and A__0 indicates logic block A, first control term. The middle left section of Figure 3 illustrates the 5 dedicated PAL terms available to each macrocell. They are labeled XPALA0__0 through XPALA0__4. The PAL in XPALA0__0 stands for Programmable Array Logic. The A0__0 means logic block A, first macrocell, first product term. XSUM__A0, located in the middle of the page, is the sum of the PAL structures with whatever PLA terms that may have been used by this particular macrocell. XFF__A0, at the middle right of the page, describes the output of the first flip-flop in logic block A. The PLA sum and sum term buffers are the next items down in the internal node schematic. PLA stands for Programmable Logic Array. There are 32 product terms and 32 sum terms available to each logic block. XPLASB__A15 means PLA Sum term Buffer, Block A, connecting to the 16th macrocell in the logic block. These buffers represent nodes that may be probed even though they do not actually exist in hardware. They were added in order to facilitate proper simulator function. XPLAS__A15 is PLA Sum term, Block A, 16th macrocell connection. The last part of the internal node schematic describes the PLA product term structures. XPLA_A_0 stands for PLA, Block A, 1st pterm.

Now that we have developed an understanding of the nomenclature and structure of the XPLA Software timing simulator, let's run the simulator and add an internal node. First, start XPLA Software and open the design **Demo.phd**. Choose the **PZ3032–8PLCC** device, set the *pin assignment* to **keep**, and *max P–term per equation* to **16**. Now fit the design. XPLA Software will automatically compile the design and then proceed to fit it into the selected device. Now invoke the timing simulator by pressing the **TimSim** radio button. Once the simulator window opens, move the cursor to the word *file* located in the top left corner of the window as shown in Figure 4. Click on the word *file* and then on the word *open*. A dialog box very much like Figure 5 will open.

Toward the middle of Figure 5, near the bottom, you will see the word *Edit*. Click in the box located just to the left of the word. A check mark will appear in the box. Now move the cursor to the words *demo.mod* located near the top left portion of Figure 5 and click on them. *Demo.mod* will appear in the dialog window located just above the files box and the words *demo.mod* located in the files box will become highlighted. Now click the radio button labeled **Open** located near the bottom left of the dialog box. The contents of Figure 2 will appear on your computer screen. Select the node name **XPALA0__0,** located on the 12th line down from the top, by placing the cursor at the beginning of the word, holding the left mouse button down, and dragging the mouse to the right until the entire word is highlighted. Now, on the computer keyboard, press the **Control** key and the **C** key at the same time. This will copy the highlighted word into the Windows Clipboard. Now go back to the simulator window. Located near the top left portion of the window is the section called **Signals**. Under the word are two radio buttons labeled **+** and **–**. Figure 6 depicts the proper cursor location.

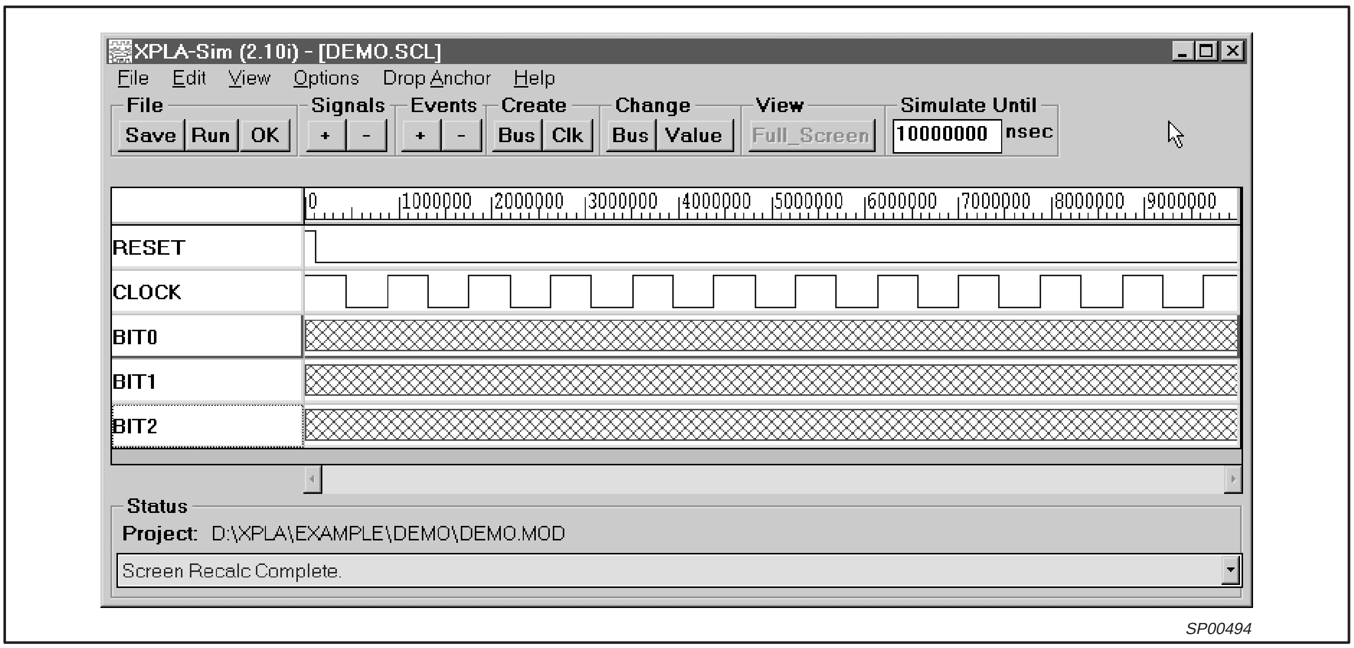

Figure 4. About to Click on File

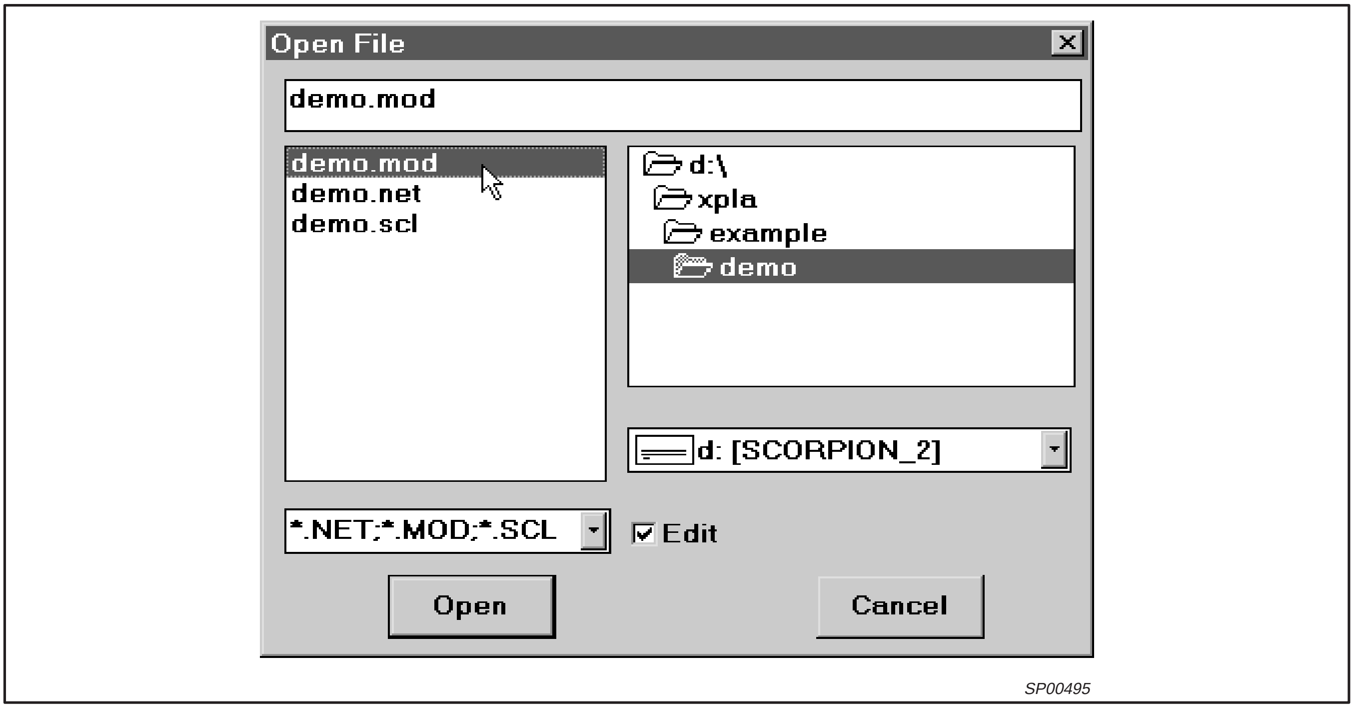

Figure 5. About to Edit .MOD File

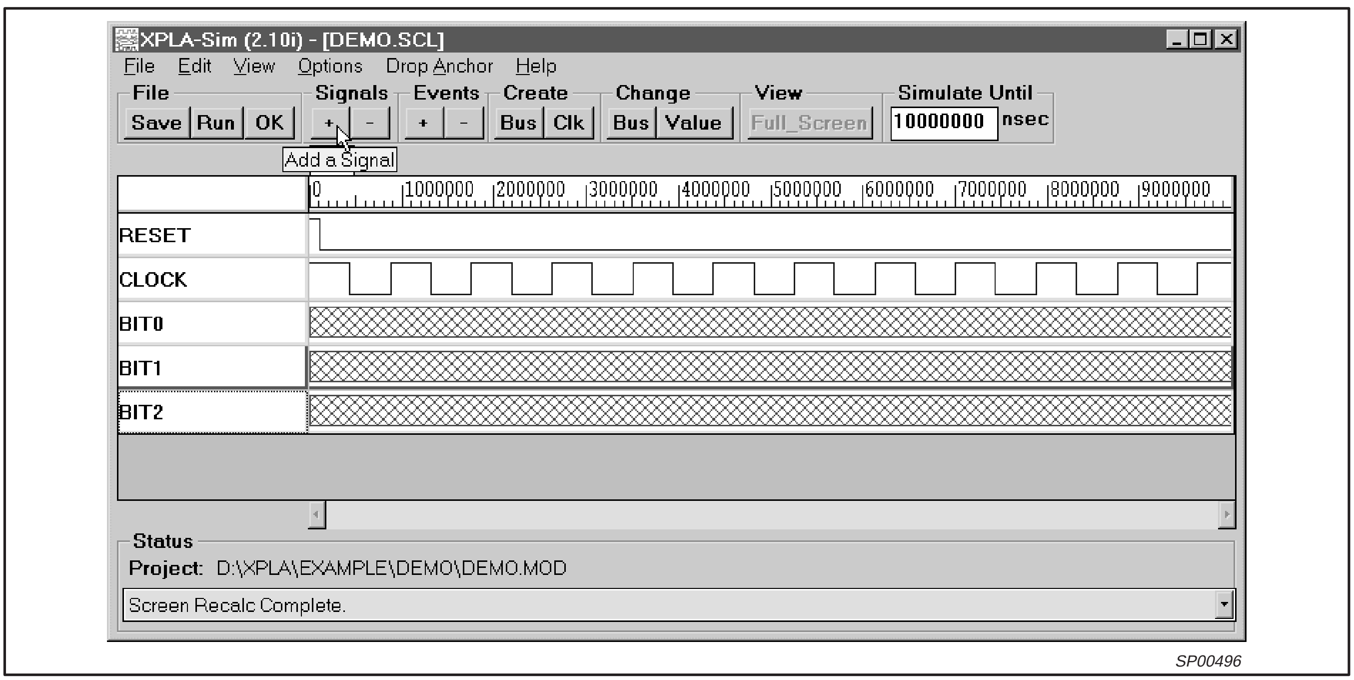

Figure 6. Adding Signals

Click the **Signals +** button. A dialog box will appear that looks like Figure 7.

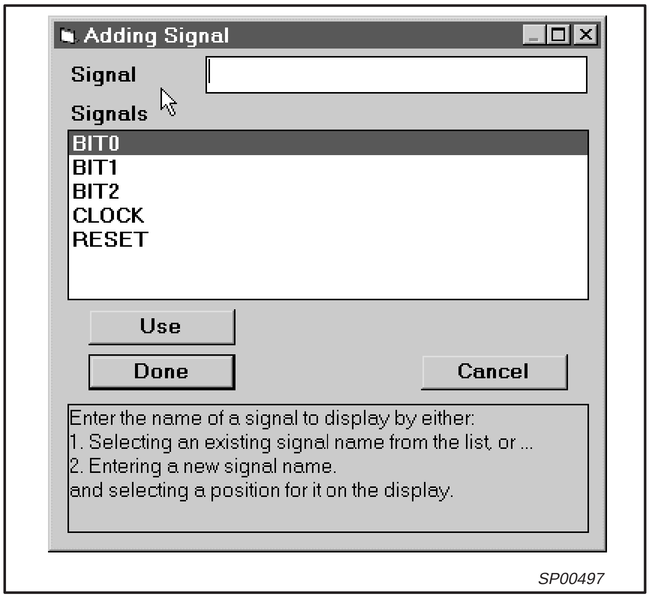

Figure 7. Adding Signal Dialog Box

Depress the **v** key while depressing and holding the **control** key. The signal **XPALA0__0** will appear in the dialog box as shown in Figure 8.

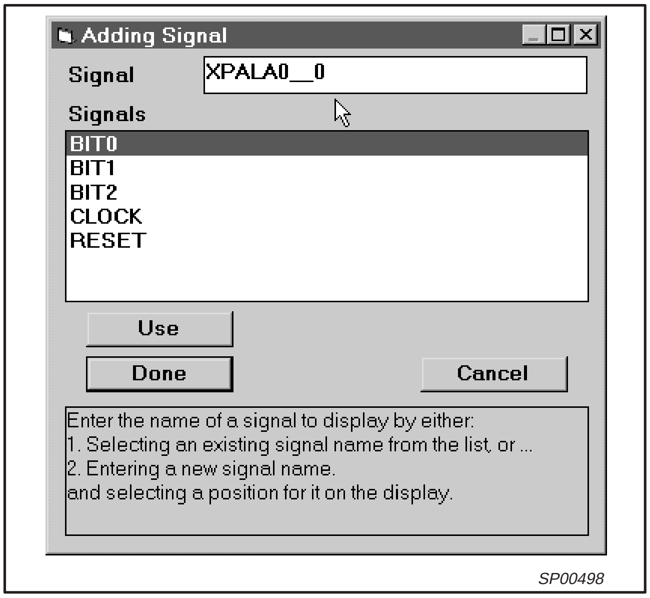

Figure 8. Adding A Signal

Click the **Done** radio button located near the bottom left of the dialog box. The signal **XPALA0__0** will now appear in the simulator waveform viewer window. You can position this signal anywhere within the simulator window by moving the cursor over the signal name and clicking and holding the left mouse button while you drag the signal to a different location. Click the **OK** radio button and then the **Run** button. You should now see a simulation very similar to the one in Figure 9.

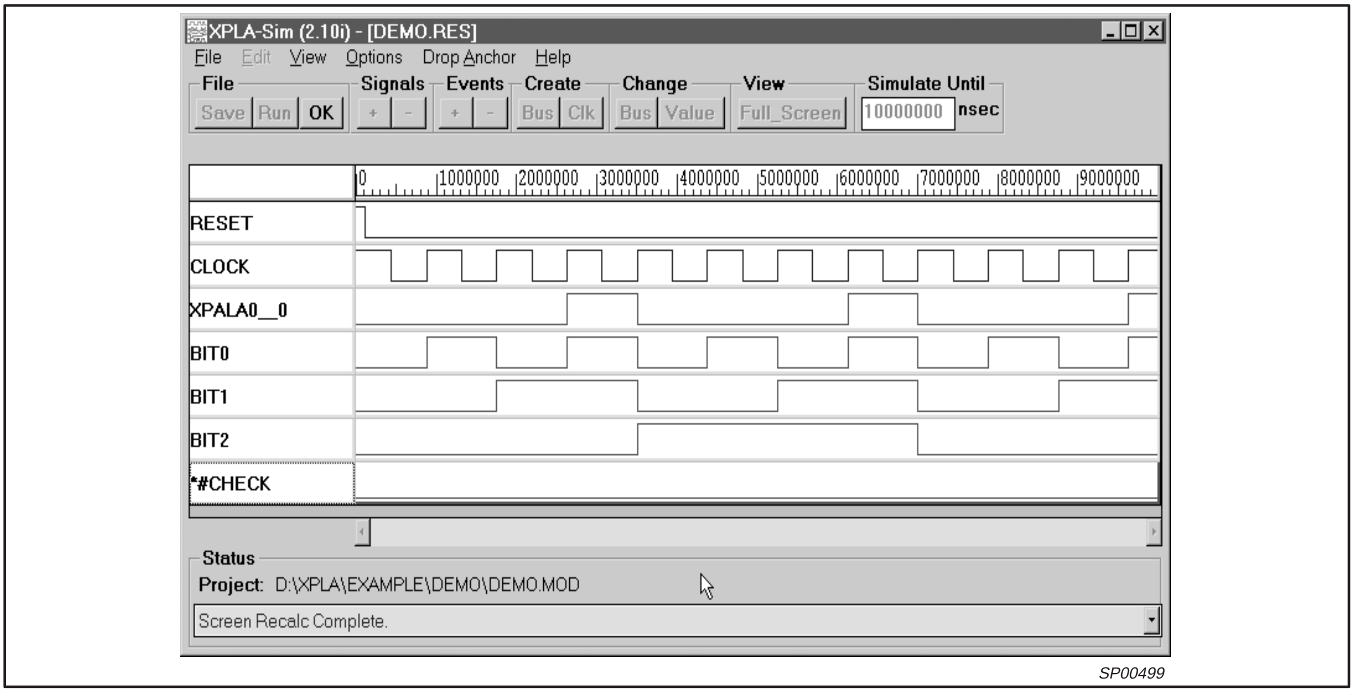

Figure 9. Simulation with Internal Node Signal Added

The internal signal levels over time for **XPALA0__0** are now displayed. In this manner you can add as many internal node signals as you require to fully understand the operation of your design.

NOTES

Probing internal nodes using XPLA software graphic simulator

AN063

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

print code
Date of release: 07-98
Document order number: 9397 750 04151

Let's make things better.

Philips
Semiconductors

PHILIPS