Fusion Design Flow Tutorial

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 502-00064-0

Release: December 2005

Trademarks

Table of Contents

1

Fusion Design Flow Tutorial

Based upon the successful Actel ProASIC3/E architecture, the Actel Fusion devices integrate a configurable 12-bit successive approximation register (SAR) analog to digital converter (ADC) with frequencies up to 600 ksps. The flexible analog block supports metal-oxide semiconductor field-effect transistor (MOSFET) gate driver output and multiple analog inputs from -12 volts to +12 volts, with an optional prescaler, thus enabling direct connection and control of a wide variety of analog systems. You can use it to monitor voltage, enable a differential current monitor, or monitor temperature. The analog inputs and outputs, the ADC, and the related soft IP compose the Analog System.

The Actel Fusion Programmable System Chip (PSC) family is the only programmable logic solution that includes embedded Flash memory—up to 1 Mbyte per device. The Flash memory offers 60-nanosecond random access and a very fast 100 MHz access in read-ahead mode. The high performance Flash memory offers you a configurable data bus supporting x8, x16, and x32 bit widths. The memory also offers error correction circuitry (ECC) with single-bit error correct, and double-bit error-detect capabilities. Pseudo EEPROM can be achieved with an available endurance extender IP from Actel. Further, the Actel Fusion PSCs enable you to reconfigure analog block settings by simply downloading data from embedded Flash memory.

Fusion uses low static and dynamic power, and includes sleep and stand-by modes. The addition of both an RC oscillator and crystal oscillator circuit eliminates the need for expensive external clock sources. The low power features, combined with Fusion's Real-Time Counter, offer a wide variety of functionality: sleep, standby, periodic wake-up, and low-speed/power operations.

Fusion Design Flow Overview

The general Fusion design flow (Figure 1-1) starts when you create the Analog System and Flash Block System, then instantiate the sub-macros into the top-level netlist, run synthesis, then run place–and-route and simulation for each step.

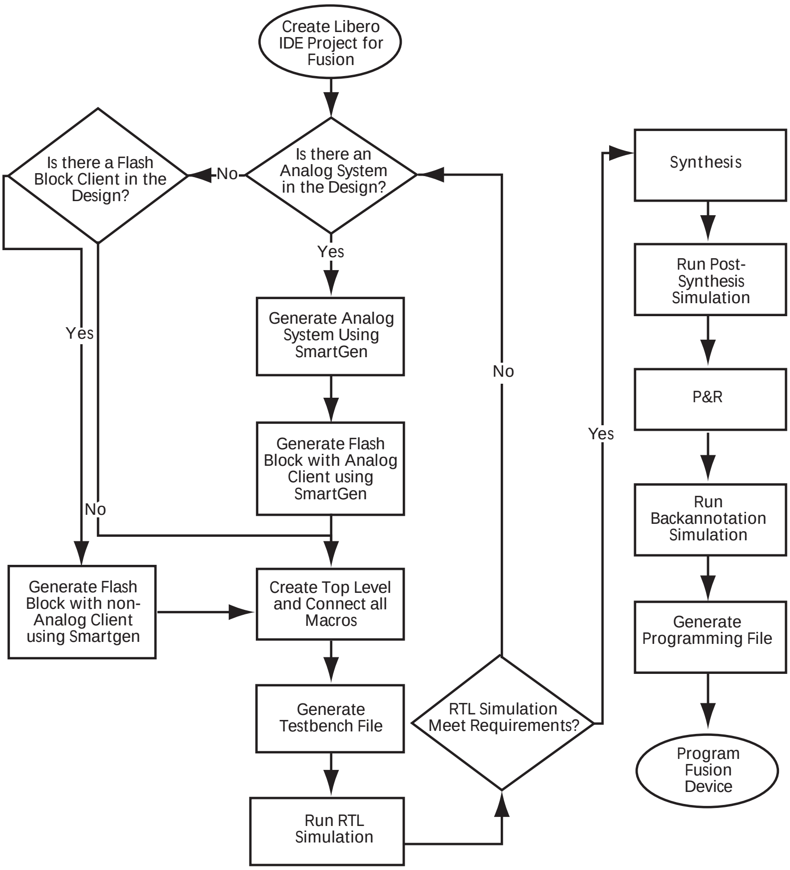

Figure 1-1. Fusion Design Flow Diagram

Application Overview

Power management is a term widely used in the industry to describe the act of managing the power-up and power-down behavior of electronic components. Power management features are often used to save power or to protect components during abnormal conditions.

The Fusion power management example described in this tutorial monitors the power supply voltage, the load side voltage, and the current provided from the supply side to load side. Based on preset voltage and current thresholds, a MOSFET between the supply side and load side is controlled by a Fusion gate driver to enable or disable power to the load side.

The power management example contains an Analog System (configured with voltage and current monitors), a 12-bit analog to digital converter (ADC), and a system frequency of 20 MHz (Figure 1-2 on page 7). The example monitors three analog signals (AV33V, AC33V, and AV33VLOAD) and configures multiple threshold flags. Additional logic blocks can be implemented in the Fusion FPGA fabric.

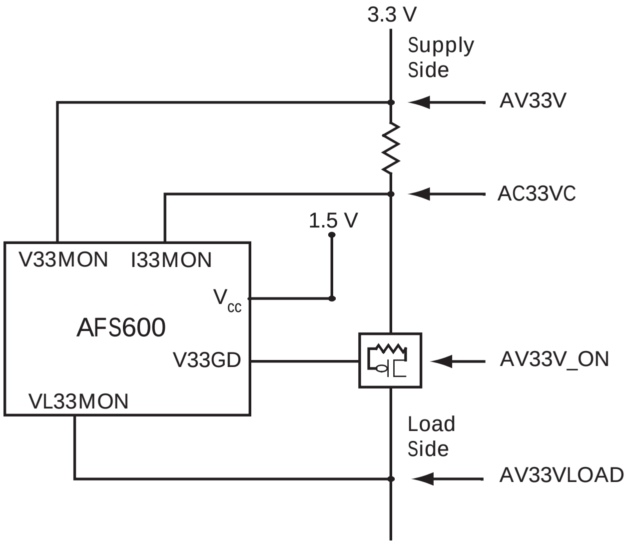

Figure 1-2. Power Management Example Block Diagram

Fusion Design Flow in Libero IDE

Step 1 – Initiate a Libero IDE Project

1. Invoke Libero IDE.
2. From the **File** menu, choose **New Project.**
3. Enter **PwrM** in Project Name field, select HDL type as VHDL (the sample Libero IDE project is in VHDL) or Verilog, then click **Next** (Figure 1-3).

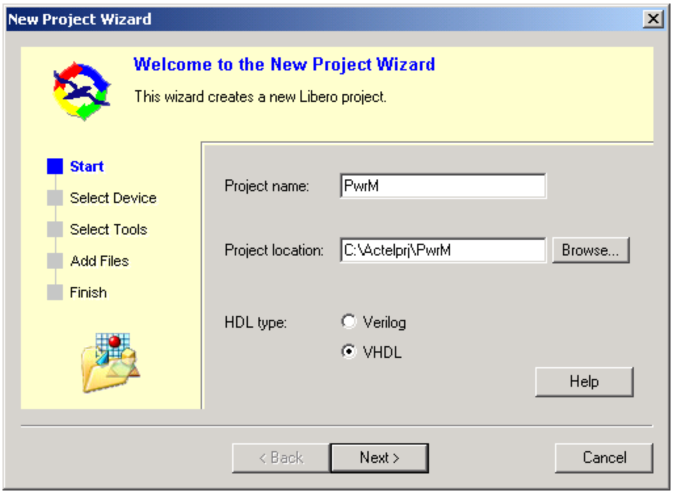

Figure 1-3. Create a Libero IDE Project

4. Select Family = **Fusion**, Die = **AFS600**, Package = **256 FBGA** (Figure 1-4).

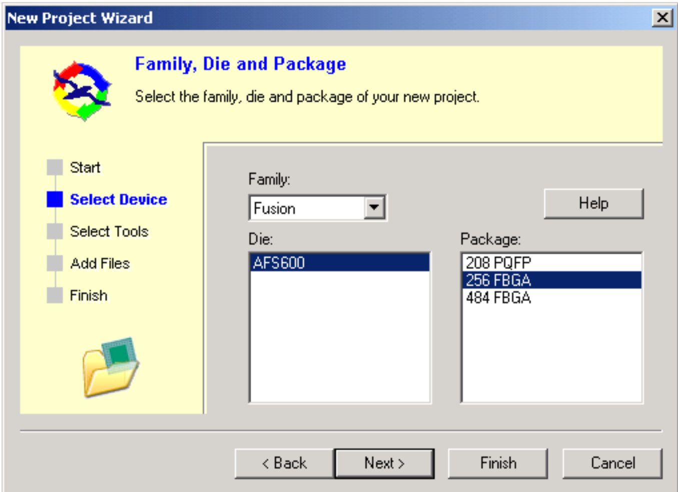

Figure 1-4. Select Fusion Device Family

5. Click **Finish**. Libero IDE creates a new project, as shown in Figure 1-5.

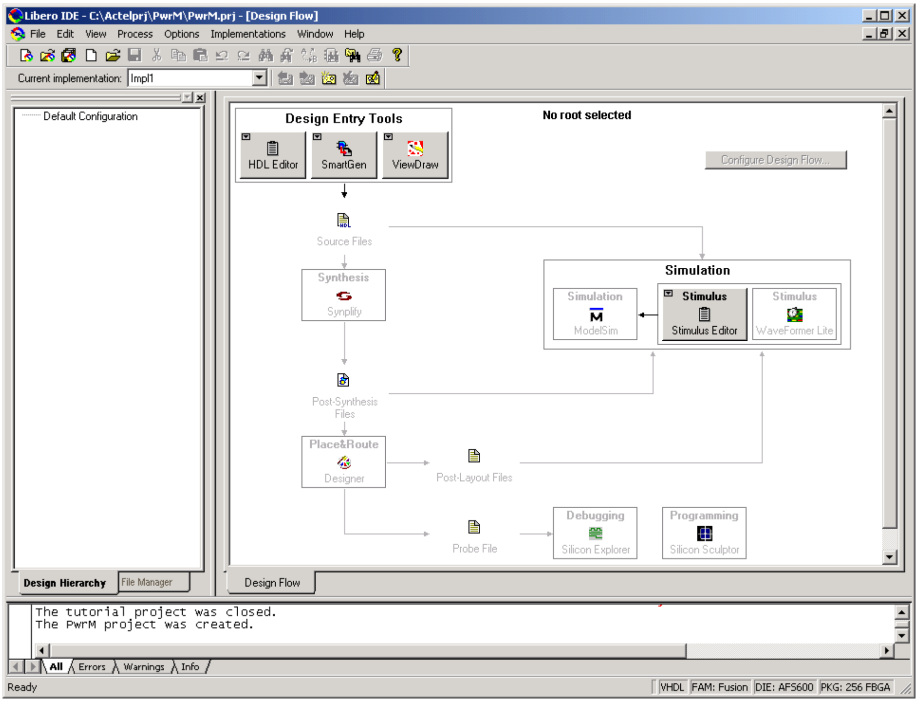

Figure 1-5. New Libero IDE Project Window

Note: If you are working on other Fusion projects, you must consider the following conditions.

- If your design does not include an Analog System or Flash Block client (i.e., if it only includes the regular FPGA macros, such as an AND gate, OR gate, NGMUX or SmartGen generated counters), follow the regular Libero IDE project design flow. For more information, refer to the Libero IDE Online Help.
- If your design only implements Flash Block related applications (such as an Initialization Client) but does not contain any portion of the Analog System, skip Step 2 and Step 3A and start from Step 3B directly.
- If your design uses the Fusion Analog System, follow Step 2.

Step 2 – Configure and Generate the Analog System in SmartGen

1. Invoke SmartGen in the Libero IDE Design Flow window.

2. Invoke Analog System Builder from Core Varieties for Fusion Family window (Figure 1-6)
3. Enter the following:
   - System Clock: 20.000 MHz
   - Resolution: 12 bits
4. Select **Current Monitor** from the Available Peripherals list.

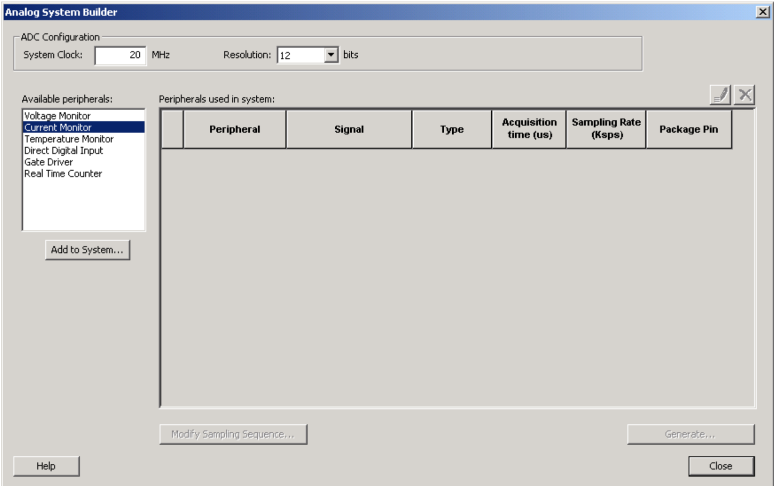

Figure 1-6. Analog System Builder Window

5. Click **Add to System** to configure the Current Monitor Peripheral.

In the Current Monitor configuration, you can create the Current Monitor for AC33V as well as Voltage Monitor for AV33V.

Note: The current channel must be used together with the adjacent voltage channel as a pair for Current Monitor. You can also use the adjacent voltage channel as a Voltage Monitor to monitor the voltage connected to this channel. Check the **Use Voltage Monitor** checkbox to enable the Voltage Monitor in the Current Monitor Configuration window. If you do not check this box, the voltage channel will still be used in the Current Monitor but cannot be used for any other voltage monitoring purpose. For configuring a pure Voltage Monitor, you can refer to Step 2, item 8.

6. Enter the following parameters, as shown in Figure 1-7 on page 13:
   - Digital filtering factor = 4
   - Acquisition time = 10.000 µs

- Enter the threshold flag values for the Current Monitor AC33V as shown in Table 1-1.

Table 1-1. Current Monitor Values for AC33V

Flag Name	Flag Type	Threshold (A)	Assert Samples	De-Assert Samples
OVER1P0A	OVER	1.0	4	7
OVER1P5A	OVER	1.5	4	7
UNDER0P2A	UNDER	0.2	2	10
UNDER0P5A	UNDER	0.5	2	10

- Enter the threshold flag values for Voltage Monitor AV33V as shown in Table 1-2

Table 1-2. Voltage Monitor Values for AV33V

Flag Name	Flag Type	Threshold (A)	Assert Samples	De-Assert Samples
OVER3P6	OVER	3.6	4	7
OVER4P0	OVER	4.0	4	7
UNDER2P5	UNDER	2.5	2	10
UNDER3P3	UNDER	3.3	2	10

- External resistor = 0.100 Ohm
- Maximum voltage = 6.000 V

Note: For more information about these settings and parameters, refer to the SmartGen Online Help.

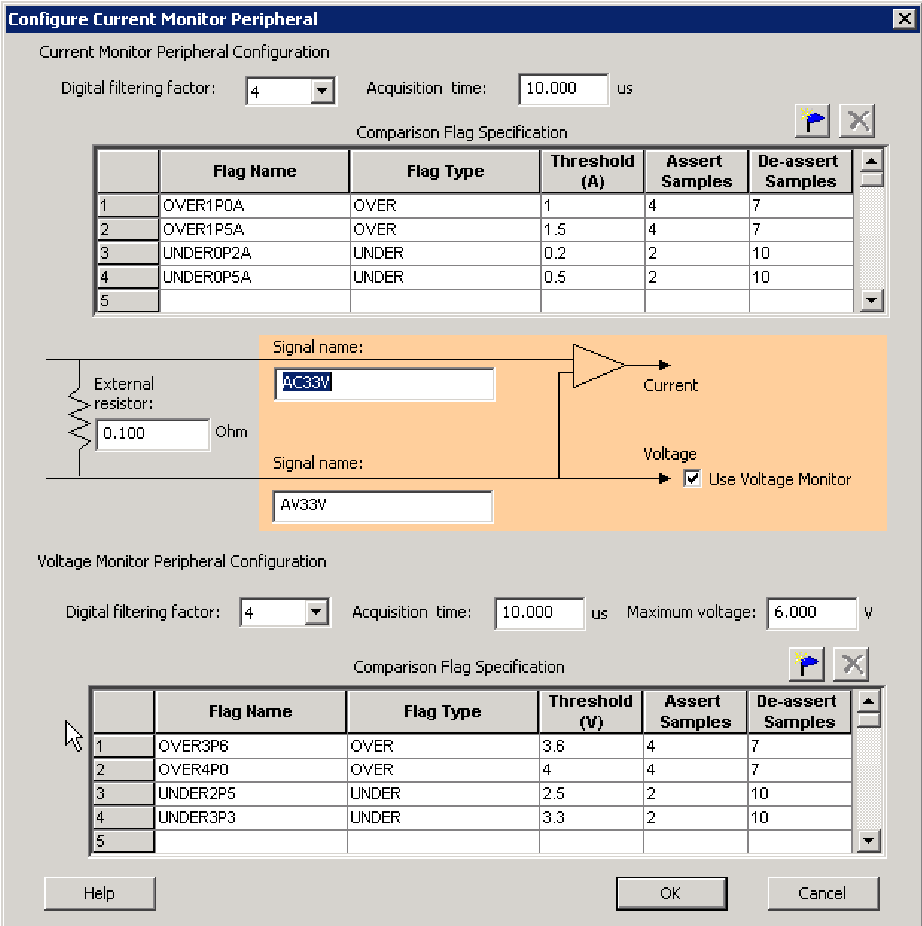

Figure 1-7. Current Monitor Configuration

7. Click **OK**. The Analog System Builder now lists the current monitor in your system peripherals (Figure 1-8)

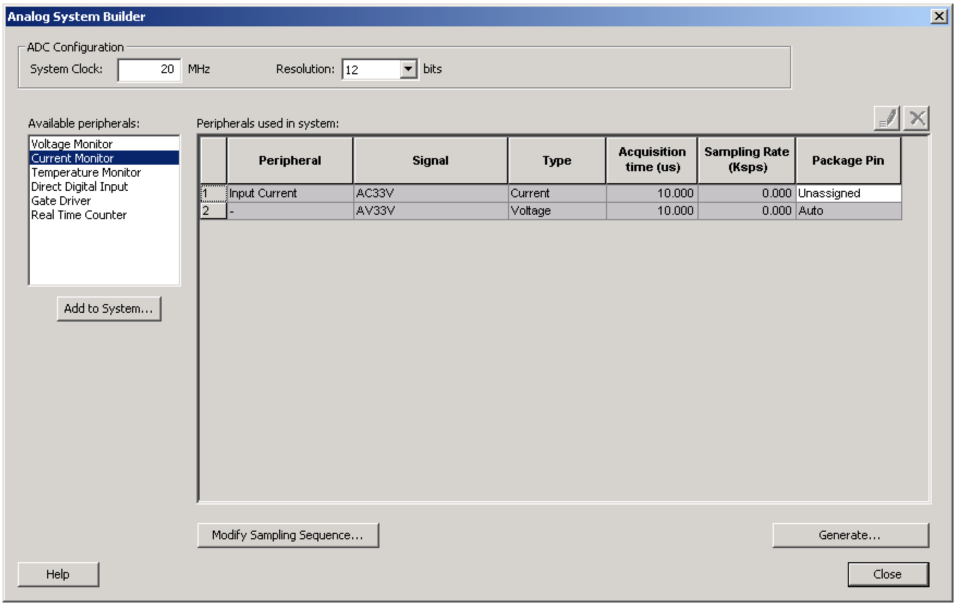

Figure 1-8. Analog System with Current Monitor Added

Note: You can select the package pin for current channel from Package Pin column. The associated voltage channel package pin is automatically selected by SmartGen after you select the current channel package pin. If you have another separate voltage channel, temperature channel, or gate driver, you can also select their package pin from the Package Pin column.

8. Select **Voltage Monitor** and configure the Voltage Monitor peripheral as shown in Figure 1-9 on page 15.

   In the Voltage Monitor configuration, create the Voltage Monitor for AV33VLOAD.

   - Digital filtering factor = 4
   - Acquisition time = 10.000 µs
   - Maximum voltage = 6.000 V

- Enter the threshold flag values for Voltage Monitor AV33V as shown in Table 1-3 and Figure 1-9.

Table 1-3. Voltage Monitor Values for AV33VLOAD

Flag Name	Flag Type	Threshold (A)	Assert Samples	De-Assert Samples
OVER3P3	OVER	3.3	4	7
OVER3P75	OVER	3.75	4	7
UNDER2P5	UNDER	2.5	2	10
UNDER3P0	UNDER	3.0	2	10

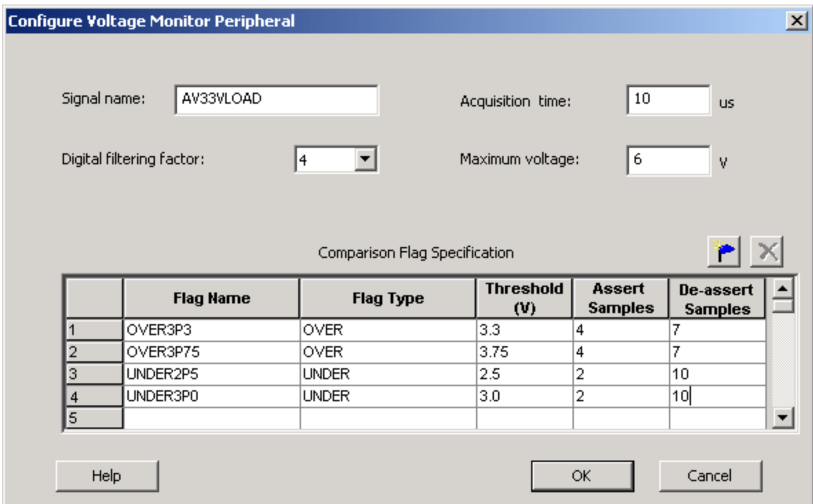

Figure 1-9. Voltage Monitor Configuration

9. Click **OK**. The ASB displays your new voltage monitor as shown in Figure 1-10.

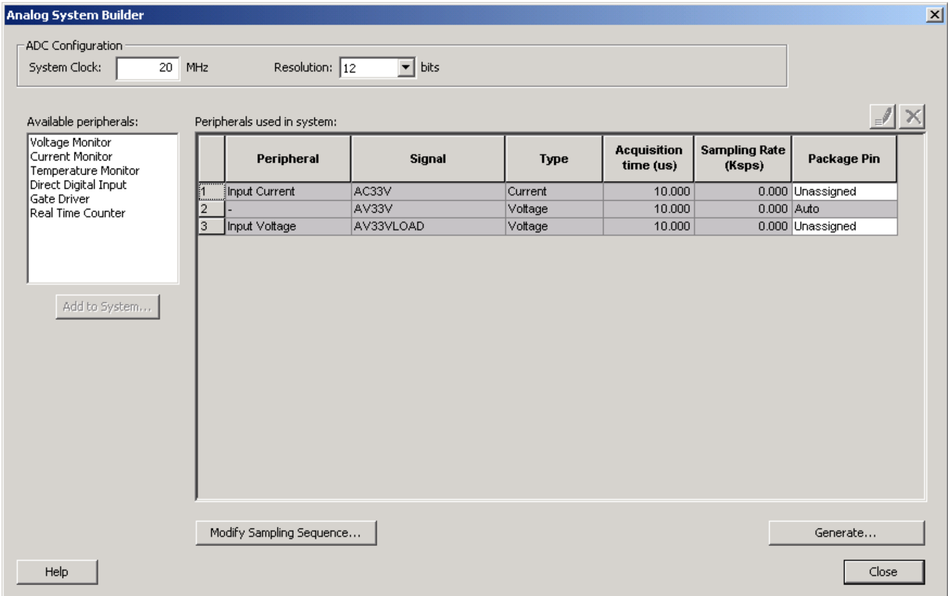

Figure 1-10. Analog System with Current/Voltage Monitors Added

10. After configuring the voltage and current monitors with multiple analog input channels, you need to define the sample sequencer parameters to allocate time slots for sampling different channels.

    - Click **Modify Sample Sequence** in the ASB window to change the sample sequence (Figure 1-11 on page 17).

    The operations available in Sample Sequencer are:

    - SAMPLE – Sample a channel that is added to the system
    - SAMPLE_RESET0 – Sample a channel and reset to Slot 0
    - RESET0 – Reset to Slot 0
    - CALIBRATE – Calibrate the ADC.
    - NOP – No operation.

    Set your sample sequence as shown in Figure 1-11. Set SLOT0 to SAMPLE on AV33V; set SLOT1 to SAMPLE on AC33V; and set SLOT2 to SAMPLE on AV33VLOAD, and set SLOT3 to RESET0.

    Click OK to return to the ASB main window.

Note: You can also use Jump sequences with jump triggers built in your HDL coding. To reserve the time slots for jump sequences, you can specify the number in "Use x slots for jump sequences". For more information on how to use jump sequences, refer to the SmartGen Online Help.

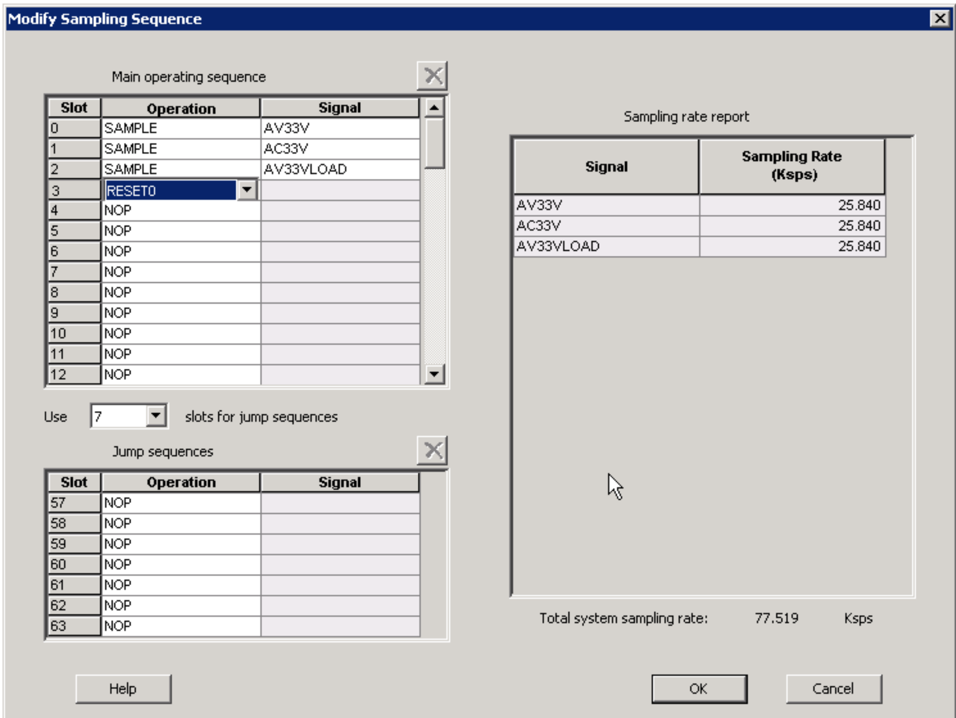

Figure 1-11. Modify Sampling Sequence

11. You must add a gate driver to control an external MOSFET pass transistor between the supply side and load side (Figure 1-12 on page 18). Based on the threshold values, you can create turn-on/off conditions for the gate driver in order to control the power supply to the load side.

    Select **Gate Driver** from the Available Peripherals list. Click **Add to System**.

    Enter the following Gate Driver parameters:

    - Gate Driver Polarity: Negative
    - Signal Name: AV33V_ON
    - Enable Signal Name: AV33V_ENABLE

- Source/Sink Current: 1.000 μA (default)

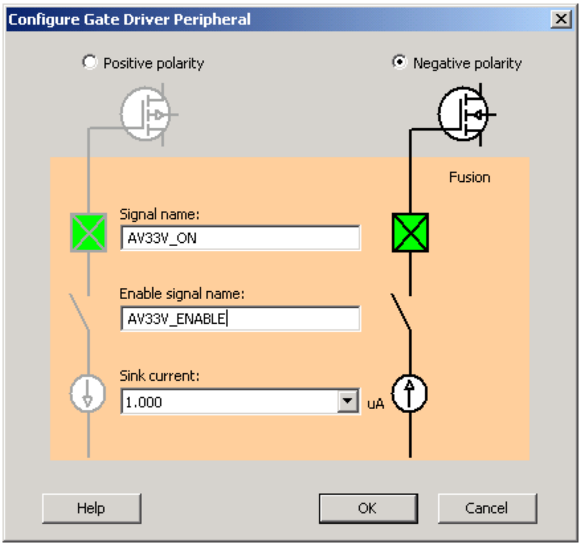

Figure 1-12. Gate Driver Configuration

12. Click **OK**. Your ASB is now configured with a Gate Driver, as shown in Figure 1-13.

For more information about Gate Driver settings, refer to the SmartGen Online Help.

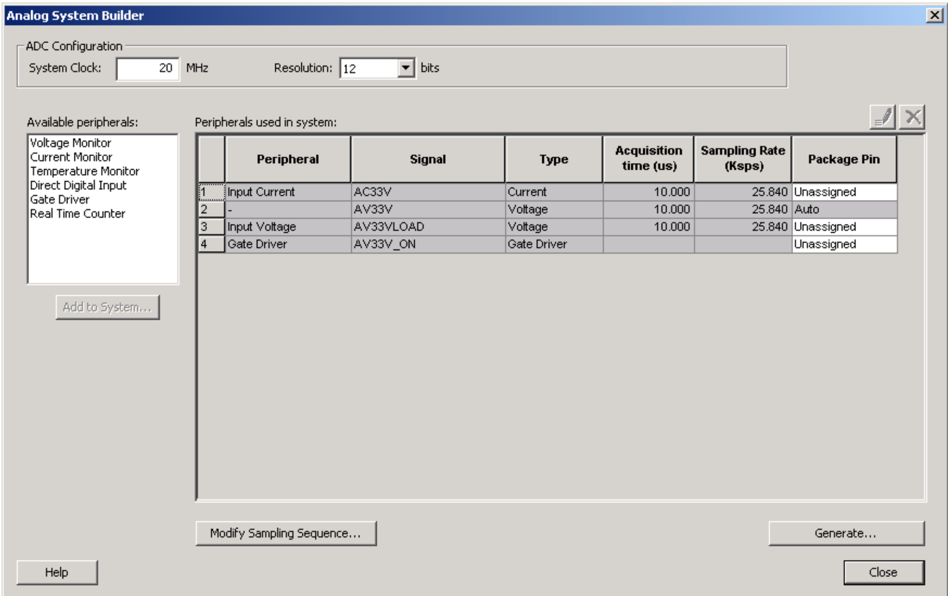

Figure 1-13. Analog System with Current/Voltage Monitors and Gate Driver Added

13. Click **Generate**. Enter the core name **AS_PwrM**, and click **OK** (Figure 1-14).

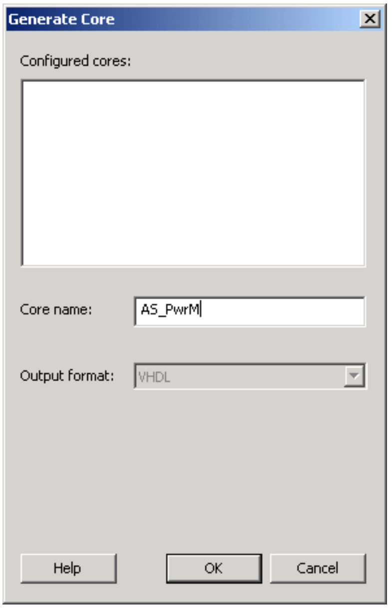

Figure 1-14. Generate AS_PwrM Core

14. Close the Analog System Builder.

The Analog System is saved in the Pwr_M Libero IDE project you created. You can reopen it for reconfiguration from the SmartGen Cores folder in Libero IDE File Manager tab or invoke it from the SmartGen GUI by double-clicking the core name in the Configured Core View window. After reconfiguration, generate the Analog System again. For detailed information about the available settings and specific tool usage, refer to the SmartGen Online Help.

Step 3A – Configure and Generate Flash Memory System with Analog System Client Using SmartGen

After you generate an Analog System, you must create a corresponding Analog System client in the Flash Memory. During Analog System generation, an NCF file is created to record the Analog System configuration and this file must be imported into the Flash Memory System generation tool (as shown in Figure 1-15 on page 21). For detailed instructions on creating a Flash Memory System, refer to the SmartGen Online Help.

1. Invoke SmartGen in the Libero IDE Design Flow window (skip this step if SmartGen is already open).

2. Invoke Flash Memory System builder from Core Varieties for Fusion Family window.
3. Select **Analog System** from the Available Client Types list, then click **Add to System**. The Add Analog System Client window appears, as shown in Figure 1-15.

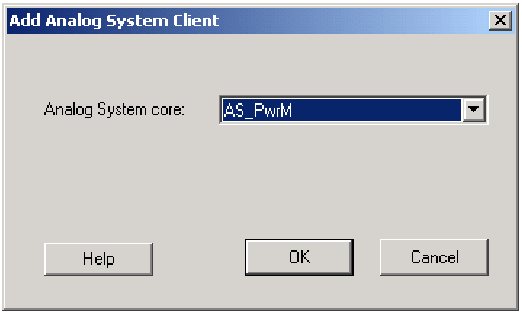

Figure 1-15. Add Analog System Client

4. Select **As_PwrM** from the Analog System core pull-down menu.

   Note: If the core you are looking for is not visible, make sure you pressed the **Generate** button in the Analog System builder.

5. Click **OK**. The Analog System Client is added to your Flash Memory System (Figure 1-16).

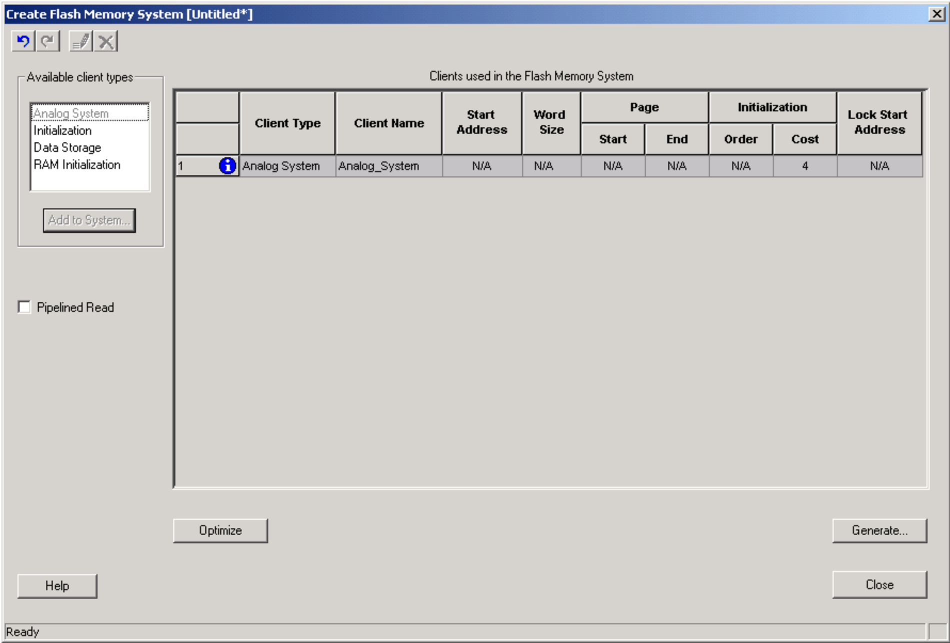

Figure 1-16. Flash Memory System with Analog System Client Added

6. Click **Generate** to complete the design.
7. Enter the Core name **nvm_sysm** and click **OK**. SmartGen creates all the Flash Block system netlist and memory files.
8. Close the Flash Memory System Builder.

You have generated AS_PwrM (Analog System) and nvm_sysm (Flash Memory System), as shown in Figure 1-17.

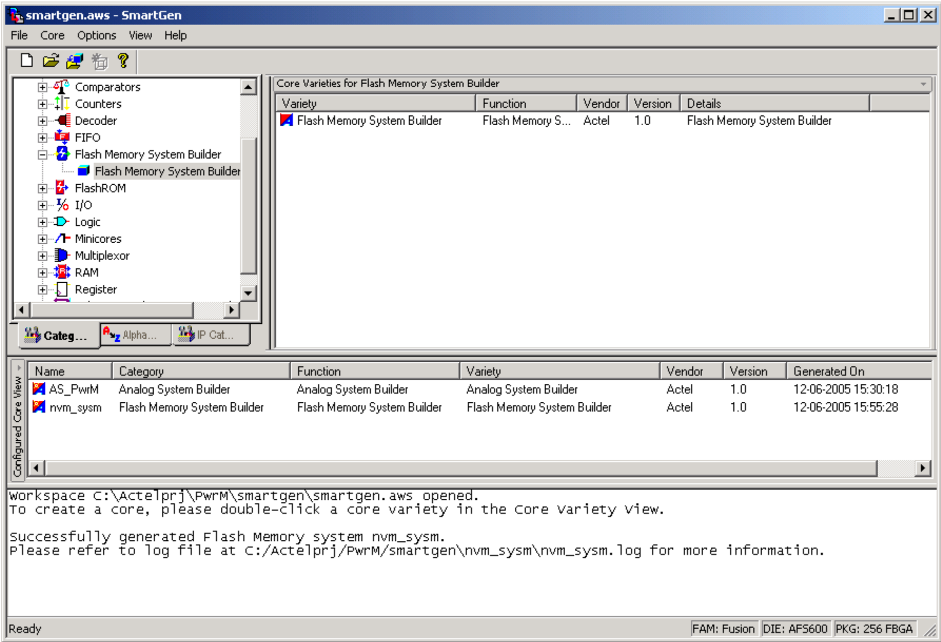

Figure 1-17. SmartGen with Analog System and Flash Memory System Generated

9. Close SmartGen.

   Note: In this particular design, after generating the Flash Memory System, you can jump to Step 4. If you are working on other Fusion projects with additional Flash Memory System clients, proceed to Step 3B.

Step 3B – Configure and Generate Flash Memory System Using SmartGen

Note: Use this section if you are working on other Fusion projects with additional Flash Memory System clients.

1. Invoke the Flash Memory System builder from SmartGen to add an Initialization Client or Data Storage Client into the Flash Memory System.
2. To add an Initialization Client or Data Storage Client into Flash Memory System, select the target client from Available Client Types window, then click **Add**.

3. Enter your parameters (as shown in Figure 1-18 on page 24). The Memory Content File is the content for the Embedded Flash to initialize this client. It can be specified in one of the supported memory formats (Binary, Intel_Hex, MotorolaS, Simple_Hex). You can manually generate a sample memory file following one of the supported file formats.

For more information on how to add an Initialization Client or Data Storage Client, refer to the SmartGen Online Help.

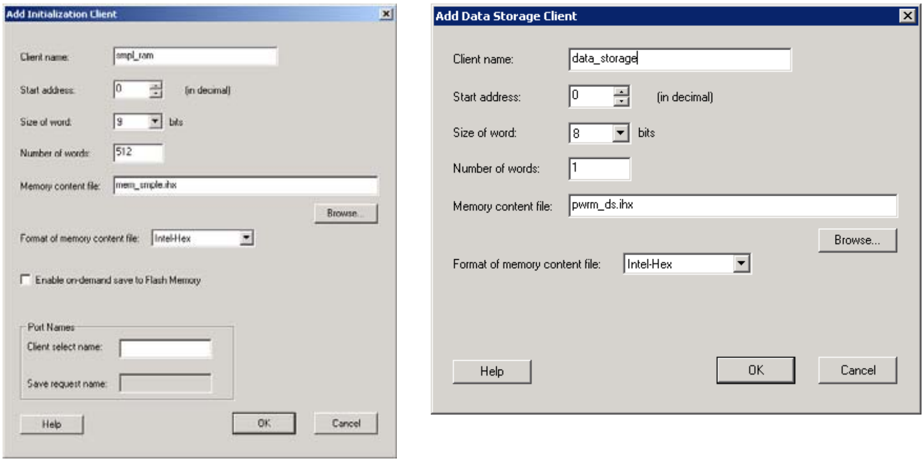

Figure 1-18. Add Initialization/Data Storage Client Dialog Boxes

Step 4 – Create Top Level Netlist

You must create a top-level VHDL (the sample project is in VHDL) or Verilog netlist to instantiate and connect all the sub-blocks, just as in any hierarchical HDL designs. The complete top-level VHDL code is in the sample Libero IDE project, under the *\hdl* folder as *Power_Management.vhd*.

To import the VHDL file, from the **File** menu, choose **Import**. Then navigate to the *Power_Management.vhd* file and click Import. Libero IDE imports the top-level netlist.

After you import or create the top-level netlist, right-click the new file in the Design Hierarchy window in the Libero IDE and select **Set As Root**.

The sample top-level VHDL code below shows the basic architecture of the top level. You will see Analog System and Flash Block system are instantiated and the gate driver turn-on/off conditions are coded.

```
***********************************************************************

library ieee;

use ieee.std_logic_1164.all;
```

```
library fusion;
entity Power_Management is --- Top level entity
port(...);
end Power_Management;
architecture DEF_ARCH of Power_Management is
component AS_PwrM is---Analog System generated by SmartGen
     port(...
        INIT_ADDR : in std_logic_vector(8 downto 0); INIT_DATA :
        in std_logic_vector(8 downto 0); INIT_DONE, INIT_ACM_WEN,
        INIT_ASSC_WEN, INIT_EV_WEN, INIT_TR_WEN : in std_logic
          ... );
end component;
component nvm_sysm is --- Flash Memory System generated by SmartGen
    port(INIT_CLK, SYS_RESET, INIT_POWER_UP : in std_logic;
        INIT_DONE : out std_logic; INIT_DATA :
        out std_logic_vector(8 downto 0); INIT_ADDR :
        out std_logic_vector(8 downto 0); INIT_ACM_WEN,
        INIT_ASSC_WEN, INIT_EV_WEN, INIT_TR_WEN : out std_logic);
end component;
begin
    INIT_DONE<=INIT_DONE_net;

    nvm_system_inst:nvm_sysm---Flash Memory System instantiation
    port map (
    INIT_CLK=>SYS_CLK,
    SYS_RESET=>SYS_RESET,
    INIT_POWER_UP=>INIT_POWER_UP,
    INIT_DONE=>INIT_DONE_net,
    INIT_DATA=>INIT_DATA_to_INIT_DATA,
    INIT_ADDR=>INIT_ADDR_to_INIT_ADDR,
    INIT_ACM_WEN=>INIT_ACM_WEN,
    INIT_ASSC_WEN=>INIT_ASSC_WEN,
    INIT_EV_WEN=>INIT_EV_WEN,
```

```
    INIT_TR_WEN=>INIT_TR_WEN
    );
analog_system_inst AS_PwrM --- Analog System Instantiation
    port map (
    ...
    INIT_DATA => INIT_DATA_to_INIT_DATA,
    INIT_ADDR => INIT_ADDR_to_INIT_ADDR,
    INIT_DONE => INIT_DONE_net,
    INIT_ACM_WEN => INIT_ACM_WEN,
    INIT_ASSC_WEN => INIT_ASSC_WEN,
    INIT_EV_WEN => INIT_EV_WEN,
    INIT_TR_WEN => INIT_TR_WEN
    );
AV33V_ON_ON <= (not AV33V_UNDER2P5) and (not AV33V_OVER4P0) and (not
AC33V_OVER1P5A) and (not AV33VLOAD_OVER3P75); ---Gate Driver Turn-ON/OFF
condition
end DEF_ARCH;
***********************************************************************
```

Connectivity Between the Analog System and Flash Memory System

The following Analog System and Flash Memory ports are connected in the Power_Management.vhd file included in the tutorial.

If you do not use the Power_Management.vhd file provided in the example, you must connect the following Analog and Flash memory system ports manually, according to the example used in code above. The ports are listed in Table 1-4.

Table 1-4. Internal Ports Between Analog System and Flash Memory System

Flash Memory System (from)	Analog System (to)	Size
INIT_ADDR	INIT_ADDR	9-11 bits
INIT_DATA	INIT_DATA	9-11 bits
INIT_DONE	INIT_DONE	1 bit
INIT_ACM_WEN	INIT_ACM_WEN	1 bit
INIT_ASSC_WEN	INIT_ASSC_WEN	1 bit
INIT_EV_WEN	INIT_EV_WEN	1 bit
INIT_TR_WEN	INIT_TR_WEN	1 bit

Gate Driver Turn-ON/OFF Conditions

If you write your own top-level code, you must add the code below (substituting your own threshold flag names). If you use the *Power_Management.vhd* file you do not have to add any additional code.

Here is an example of the turn-on/off condition for the gate driver based on the thresholds set in analog system configuration in Step 2. Using these conditions, you can control the power supply to the load as well as protect the load if voltage or current is off the limit.

If 2.5 V < AV33V < 4.0 V, then the gate driver enable signal (AV33V_ON_ON) is HIGH and the gate driver (AV33V_ON) will be ON.

If AC33V > 1.5 A, then the gate driver enable signal (AV33V_ON_ON) is LOW and the gate driver (AV33V_ON) will be OFF.

If AV33VLOAD > 3.75 V, then the gate driver enable signal (AV33V_ON_ON) is LOW and the gate driver (AV33V_ON) will be OFF.

Sample VHDL code:

```
AV33V_ON_ON_int <= (not AV33V_UNDER2P5_int) and (not AV5V_OVER4_int) and
(not AC33V_OVER1P5A_int) and (not AV33VLOAD_OVER3P75_int);
```

You can customize the gate driver turn-on/off conditions based on your threshold values.

Step 5 – Create Testbench

You must have a testbench file in order to run simulation. You can either create one manually, or you can import the sample testbench file (*tb_new_pwr_1.vhd*) included in the sample Libero IDE project.

Actel has a special function in the Fusion library that enables you to manually code a testbench for analog signals. Manually coding a testbench that includes analog signals requires that you use the special function; an example is included in the testbench file *tb_new_pwr_1.vhd* in the sample Libero IDE project.

Analog signals must be pre-processed before they can be used in digital simulation. The drive_analog_input function enables our simulation tool to simulate the analog signals from the Analog System Builder.

Here is sample VHDL code that includes the drive_analog_input function for the analog signals (there are no other special simulation requirements for Fusion besides these analog input):

```
convert_analoginput_4_simulation : process
begin
wait on <analog_input_name>;
drive_analog_input ( real (<analog_input_name>),
<converted_digital_value> );
end process convert_analoginput_4_simulation;
```

To import the testbench file from the sample project, from the **File** menu, choose **Import**, and select **Stimulus Files** from the dropdown menu. You must navigate to the *stimulus* directory in the sample project to import the file *tb_new_pwr_1.vhd*. Click **Import** to continue.

Step 6 – Run RTL Simulation

After generating the Analog System, Flash Memory System, and top-level netlist and creating the testbench, you can proceed with RTL simulation. The top-level netlist must be Set As Root before you can run simulation. To do so, right-click the top-level netlist in the **Design Hierarchy** window in the Libero IDE and choose **Set As Root** from the Right-click menu.

You must associate a stimulus before you can run simulation. To do so, right-click the top-level netlist in the Libero IDE Design Hierarchy window and choose **Organize Stimulus**.

Select the stimulus file in your project (tb_new_pwr_1.vhd is the filename in the example project) and click the Add button to add it to your projects' Associated files list. You are now ready to run RTL Simulation.

1. Invoke Model*Sim*® from the Libero IDE project.
2. Run the simulation for 6 ms.
3. Verify the threshold flags with respect to the analog input signals in the testbench file (Figure 1-19 on page 29).

If RTL simulation does not meet your requirements, you may need to change Analog System parameters. If so, redo steps 2 to 6. Refer to the SmartGen Cores Reference guide for more information on how adjusting parameters will change the results. Once the simulation passes, continue to step 7. For more information regarding simulation, refer to the Libero IDE online help.

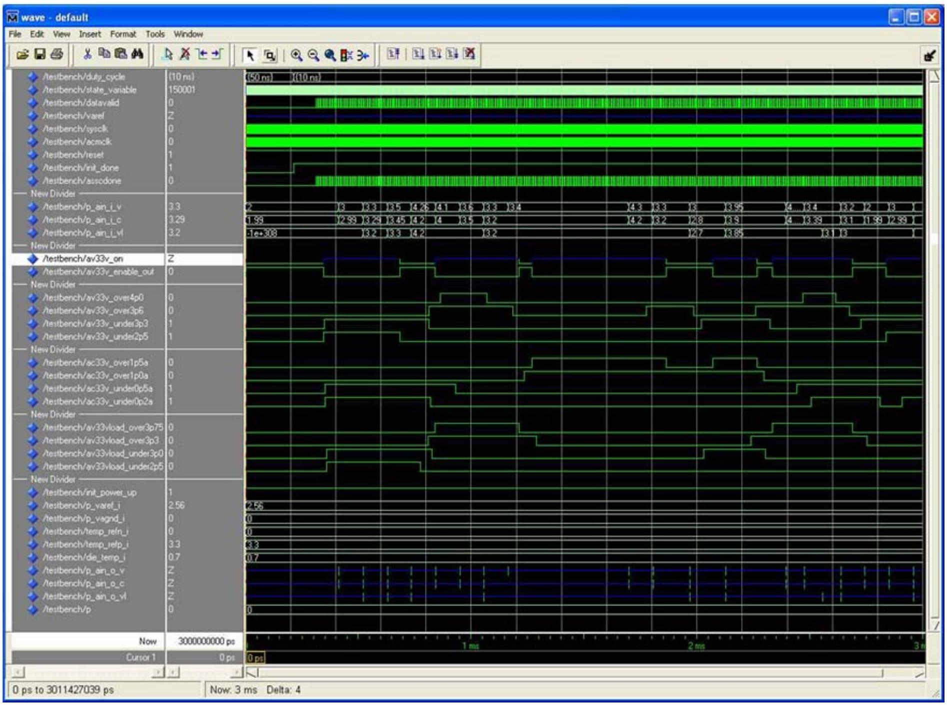

Figure 1-19. RTL Simulation Result

Updating the Top Level Netlist When Sub Macros Changed

If you made any changes to the Analog System (step 2) or Flash Memory System (step 3), then you need to make the following corresponding changes:

1. If you changed the Analog System parameters configuration (no changes to signal names or threshold flag names), the Flash Memory System must be regenerated by loading the NCF from the latest Analog System. No changes are required in the Top level netlist for this change in the Analog System.
2. If you changed the Analog System signal names and/or threshold flag names, the Flash Memory System must be regenerated by loading the NCF from the latest Analog System. Also, you must alter the top-level netlist for this change in the Analog System.
3. If you changed the client name and/or parameters of the Flash Memory System client, the corresponding netlist is changed. You must regenerate the Flash Memory System and change the corresponding port names in the top-level netlist.

   Note: The Client Name is prefixed before the select and enable signal names to group all the control signals for that client together.

Step 7 – Synthesis

Invoke Synplify® from the Libero IDE Design Hierarchy window or click the Synthesis icon in the Design Flow window (Figure 1-20).

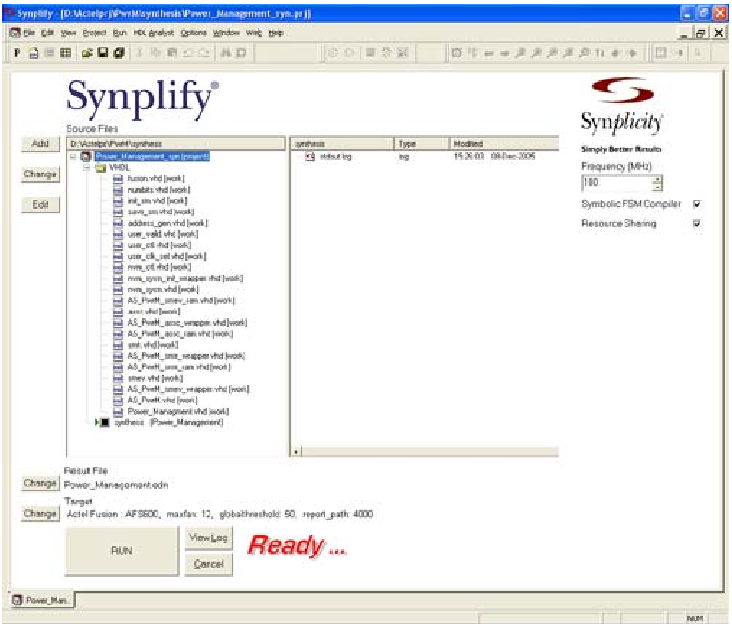

Figure 1-20. Synthesis

Accept the default implementation, make sure AFS600 is selected for this particular design, then click **Run** to execute synthesis. For more information about Synthesis, refer to the Libero IDE Online Help.

Step 8 – Post-Synthesis Simulation

After synthesis, click the Simulation icon in the Libero IDE to continue with post-synthesis. Verify the threshold flags with respect to the analog input signals in the testbench file. For more information about Post-Synthesis Simulation, refer to the Libero IDE Online Help.

Step 9 – Place-and-Route

Click Place-and-Route to invoke Designer from the Libero IDE Design Flow window, then follow the instructions in the Device Selection Wizard to select the appropriate device. For this particular design, select **AFS600**, **256FBGA**, –2 speed, and a Die Voltage of 1.5 V (Figure 1-21).

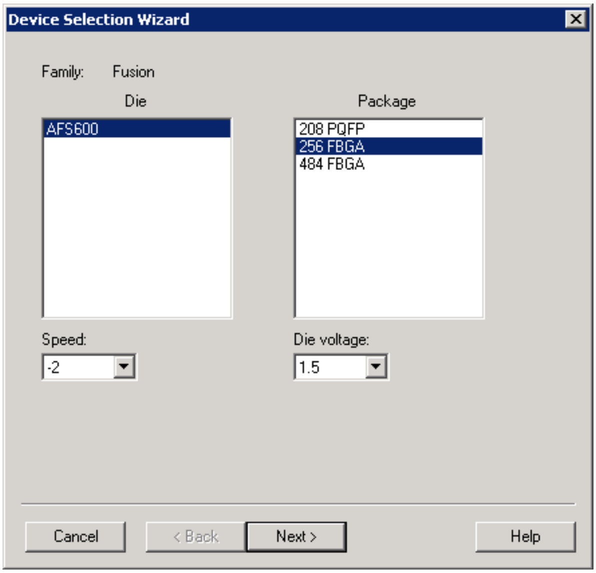

Figure 1-21. Place-and-Route

After Compile and Layout are complete, click the Back-Annotate icon to generate a back-annotated VHDL netlist and timing information. For more information about place-and-route, refer to the Libero IDE online help.

Step 10 – Back-Annotation Simulation

After Back-Annotation, you can continue with the post-layout simulation by clicking the Simulation icon in the Libero IDE Design Flow window. Verify the threshold flags with respect to the analog input signals in the testbench file. For more information about Back-Annotation Simulation, refer to Libero IDE online help.

2
Special Notes

Log file shows the system configuration.

The Log file records all System parameter settings after configuring and generating systems from SmartGen. This gives advanced users the capability to explore details of the system and provides guidance for fine-tuning the system.

Pre-defined Soft IP and Applets speed up the design process.

Pre-defined Soft IP (included with SmartGen) can be implemented in a particular application. This reduces the development time. You also have the flexibility to implement your own IP (such as Core8051 to replace the IP that is bundled with SmartGen). Any user-defined application module, or Applets, may be archived and re-used in other applications. For example, you could archive the GEN file of the power management system and reuse it for your next project.

If you have multiple versions of the Analog System, make sure the corresponding Flash Memory System is used for the project.

As stated in step 4, make sure that the Analog and the Flash Memory System used in the top level match each other if you have generated several versions of the Analog System and Flash Memory System.

A

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call **650.318.4480**
From Southeast and Southwest U.S.A., call **650. 318.4480**
From South Central U.S.A., call **650.318.4434**
From Northwest U.S.A., call **650.318.4434**
From Canada, call **650.318.4480**
From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**
From Japan, call **650.318.4743**
From the rest of the world, call **650.318.4743**
Fax, from anywhere in the world **650.318.8044**

Actel Customer Technical Support Center

Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support

Visit the Actel Customer Support website (www.actel.com/custsup/search.html) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's home page, at www.actel.com.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/contact/offices/index.html.

For more information about Actel's products, visit our website at http://www.actel.com

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 USA*
Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd. *• Dunlop House, Riverside Way • Camberley, Surrey GU15 3YL • United Kingdom*
Phone +44 (0) 1276 401 450 • Fax +44 (0) 1276 401 490

Actel Japan *• EXOS Ebisu Bldg. 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 Fax +81.03.3445.7668 • www.jp.actel.com

Actel Hong Kong *• Suite 2114, Two Pacific Place • 88 Queensway, Admiralty Hong Kong*
Phone +852 2185 6460 Fax +852 2185 6488 • www.actel.com.cn

502-00064-0/12.05