SmartGen, FlashROM, Analog System Builder, and Flash Memory System Builder v7.1 User's Guide

Actel®

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 50200062-2

Release: February 2006

Trademarks

Table of Contents

What's New in SmartGen?

The SmartGen software generates a large variety of commonly used functions. You can create a workspace and generate structural netlists in EDIF, VHDL, and Verilog. Furthermore, you can generate VHDL and Verilog behavioral models for most parameterized functions (the behavioral models may be used in a simulation environment).

This help provides descriptions of cores that you can generate using the SmartGen software. For more information about instantiating specific cores refer to the *Actel HDL Coding Style Guide*.

The SmartGen environment enables you to create a workspace, and then import existing cores and create new ones.

SmartGen includes several new and updated features:

- The Analog System Builder (with new advanced memory configuration options)
- The Flash Memory System Builder
- Generated state information for your core - Enables you to see if your core has been updated since the last release.

SmartGen Workspace

You must open a workspace in SmartGen before you can create or modify a core. Workspaces enable you to create, modify, and import existing SmartGen cores. The workspace is a logical grouping for your cores; each workspace contains cores for a specific product family.

Cores

When you generate a SmartGen core, SmartGen creates a core file that includes a structural netlist, an optional VHDL or Verilog behavioral netlist, a log file that summarizes your core parameters, and a GEN file that includes all the parameters you selected when you generated the core. Do not manually edit any of your core files. Instead, use SmartGen to edit the parameters for a core and then regenerate it.

Fusion, with the Analog System and Flash Memory System Builders, creates special directories for each core. Your cores can only be saved in the workspace directory. Do not save other files in your workspace directory, or in sub-directories created by SmartGen; the files are over-written when you re-save the core.

Varieties

Varieties list different implementations of similar functions. For example, when you click Arithmetic in the Category tab, SmartGen displays the complete list of varieties available for Arithmetic. Click a column header to sort the list of varieties. Click the Adder function to display the list of core varieties for Adder.

Core Catalog

The top of the Categories tab in the Core Catalog window displays the device family for your workspace at the top. Categories vary according to your device family; select the latest families to see and test the newest cores available. Click a core category to expand the list and display the functions.

The Alphabetic tab displays an alphabetical list of all the functions available for your device family. Click a function to display a list of varieties. Click the column headers to sort your core varieties.

IP Core Catalog

The IP Core Catalog lists IP cores available from Actel and our solution partners. You can browse the list, double-click an IP core for more information, and download datasheets and application notes for any core if you are connected to the web. Contact Actel if you are interested in purchasing any of the cores listed in the catalog.

Core Variety View

The Core Variety View window displays the list of cores available for your device family. Click core categories and functions to narrow your list of core varieties. For example, click your device family for a complete list of all the core varieties for your family. Click the Arithmetic category to view a list of only Arithmetic core varieties, and click a specific function to narrow your list even further and view core varieties for the function.

The Generated State column displays information on the state of your core. It notifies you if you need to regenerate your core, if an update is available, or essential files are missing. See the SmartGen user interface for more information. A core in the Regenerate or Not Generated state is liable to cause your design to fail.

Log Window

The log window displays tips and instructions on the SmartGen flow. It may list errors and warnings that occur during core generation, as well as core parameters after you generate a core.

See Also

SmartGen user interface

Create a workspace

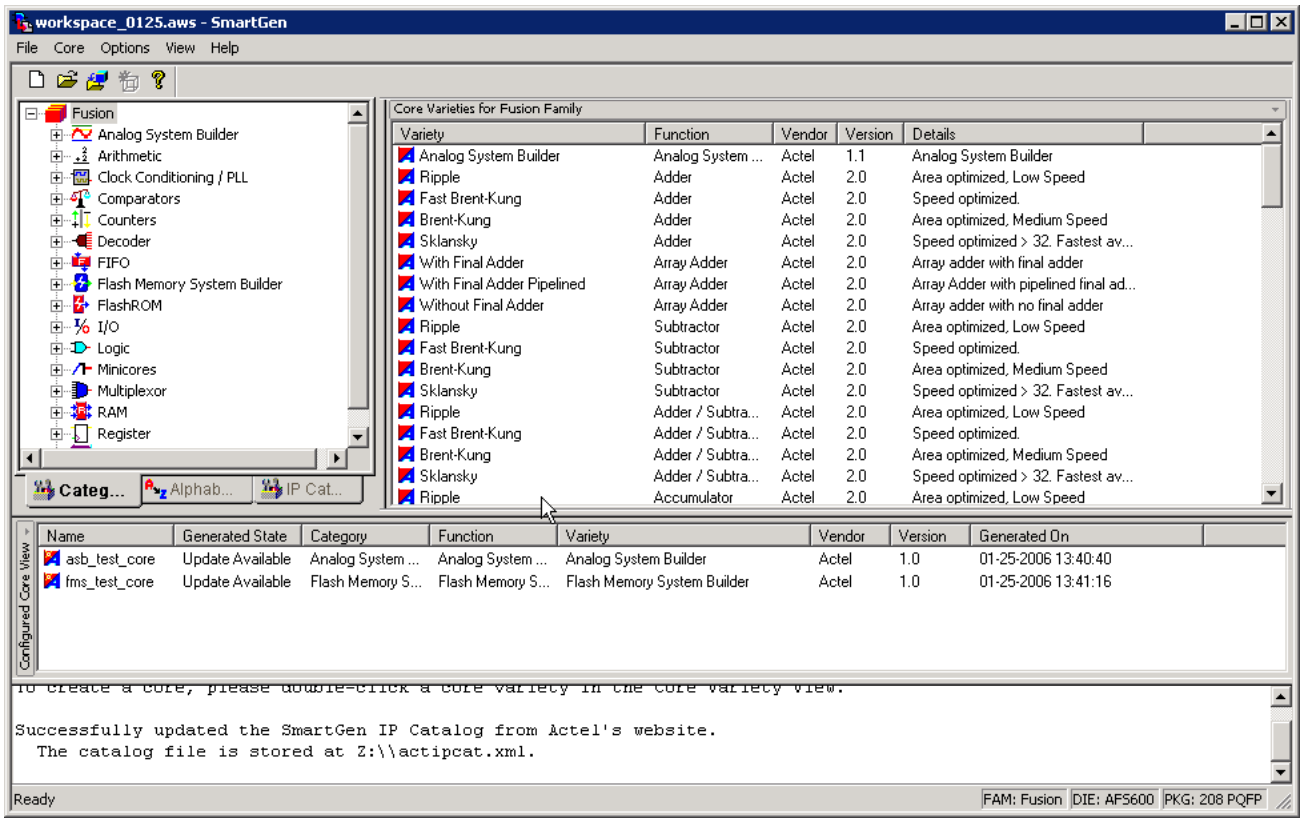
Import a legacy core

Create a new core in SmartGen

SmartGen User Interface

The SmartGen software generates a large variety of commonly used functions. You can generate structural netlists in EDIF, VHDL, and Verilog. Furthermore, you can generate VHDL and Verilog behavioral models for most parameterized functions (the behavioral models may be used in a simulation environment). SmartGen includes workspace and core-management features.

SmartGen is divided into the Core Catalog / Intellectual Property Catalog (IP Catalog), the Variety View window, the Configured Core View window, and the Log Window (as shown in the figure below).



SmartGen Main Window

Use this interface to browse your cores, review your configured cores, and select cores to create or modify.

The **Variety View** window displays a list of core varieties available for the core you selected in the Categories tab. For example, if you select the Arithmetic core type, and then select the Adder core, the Variety View window displays a list that includes Sklansky, Fast Brent-Kung, Ripple, etc.

Click a heading in the Variety View window to sort your list of cores. Cores are listed in Function order by default (Arithmetic cores, Clock Conditioning cores, Comparators, etc.). If you click a column heading they are sorted alphabetically by that column value (except for Version - then they are sorted numerically).

The **IP Catalog** displays a list of IP cores available for your device. You can access the complete core catalog library from within SmartGen. Cores are separated by type; select a type to display and double-click a core to display specific core information, or to download the core's datasheet. Click the headings in the Variety view to sort your cores alphabetically according to variety, function, vendor, etc.

The **Configured Core View** window displays a list of configured cores in your workspace. The configured cores appear each time you re-open your workspace; by default cores are listed in the order you generated them. Click a column heading to alphabetically by that column value (except for Version - then they are sorted numerically). The Configured Core view also displays information on the state of your core in the **Generated State** column:

- **Regenerate –** The core must be regenerated; the currently generated core is obsolete. This occurs if the core was generated using a previous version of SmartGen and the workspace contains a core generated from a newer file set. A core in the Regenerate state is liable to cause your design to fail.
- **Update available –** It is not mandatory that the core be regenerated but you may elect to in order to take advantage of a new feature or defect resolution. This state occurs when all cores of a given type have been generated successfully but the software has access to a newer version of that type.
- **Not Generated –** The core is missing its HDL file and must be regenerated. This may occur through user error or the import of very old cores. The core must be regenerated in order to create the proper HDL and associated files. A core in the Not Generated state is liable to cause your design to fail.

You can delete cores from your Configured Core View (and leave them on the disk), or you may delete them from your disk entirely. Select the core(s) you want to delete and press the **Delete** key, or right-click and choose **Remove from Workspace** or **Remove from Disk and Workspace**.

The **Log Window** displays information as you configure your cores in your workspace.

See Also

Import a legacy core

Create a Workspace

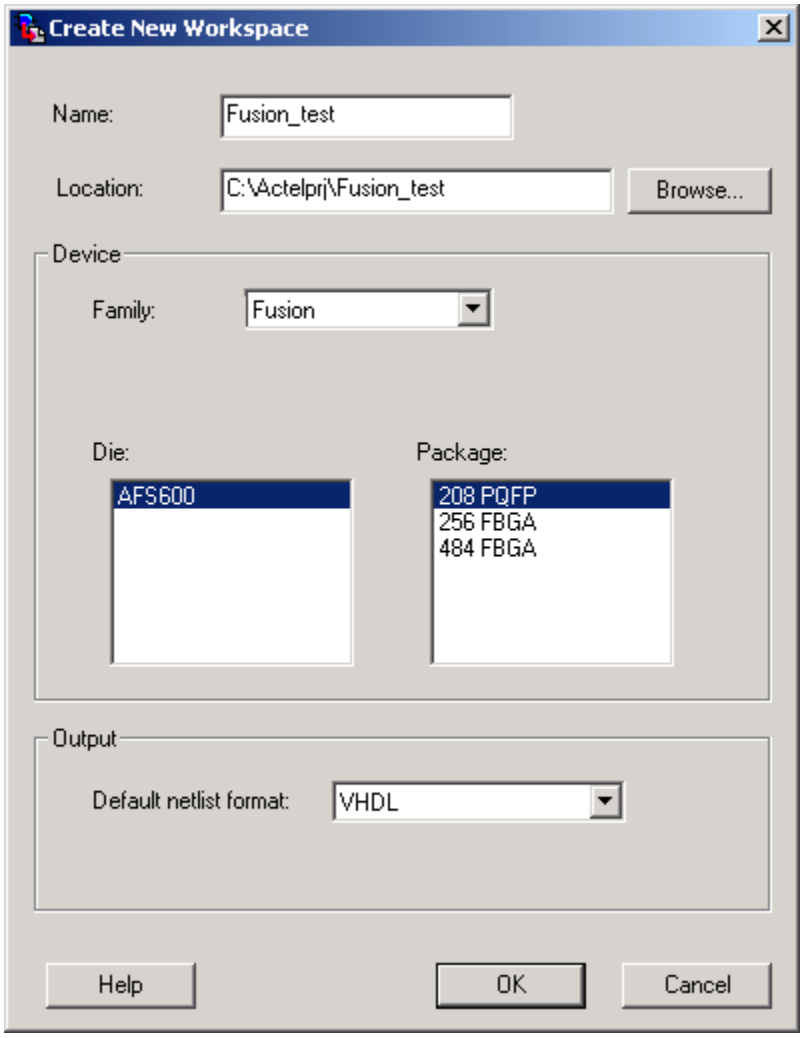
You must create a workspace to generate a core in SmartGen.

A workspace defines your family and the default directory in which you save your configured cores. If you wish, you may save your workspace in a different directory. The workspace is a logical grouping for your cores; each workspace contains cores for a specific product family.

To create a workspace:

1. Start SmartGen. When you open SmartGen it asks if you wish to create a workspace or open an existing workspace. You may choose to create a workspace or open an existing workspace. Select the check box to hide this dialog the next time you open SmartGen.

2. From the File menu, choose New Workspace, or click CTRL + N.
3. Specify the Workspace name, Workspace location (click the Browse button to navigate to or create a directory), Family, Die, Package and Default netlist format.



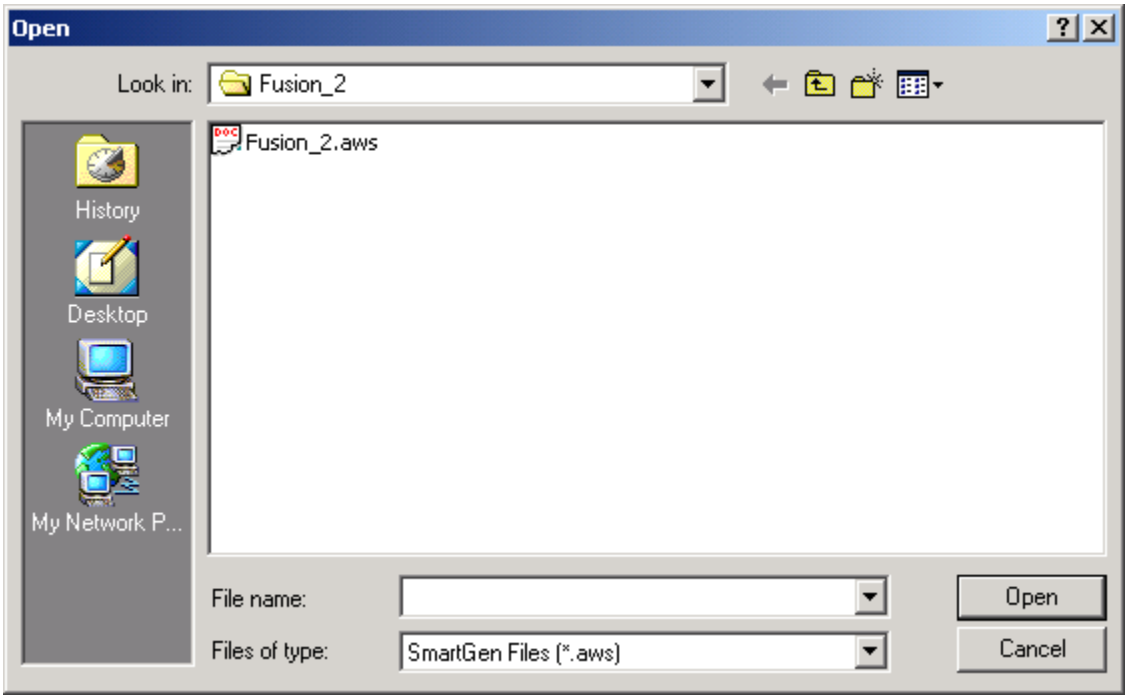
Create New Workspace Dialog Box

4. Click OK to create the new workspace. The new workspace appears in the SmartGen window. By default, the list of cores available in a new workspace is sorted by Categories. Click the Alphabetic tab to display the complete list of SmartGen cores.

Open a Workspace

You must open a workspace in SmartGen to generate cores. If you have never opened a workspace, you can create one. To open an existing workspace:

1. Start SmartGen.
2. From the File menu, choose Open Workspace, or click the Open Workspace button in the SmartGen toolbar. This opens the Open Workspace dialog box (as shown in the figure below).



3. Navigate to the workspace you wish to open and click Open.

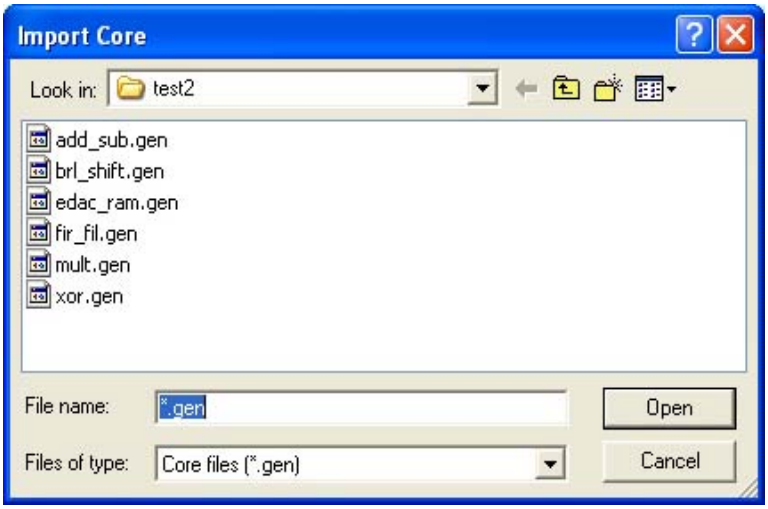
SmartGen audits the workspace for existing cores and their associated files (CXF, LOG, VHD/V, and other auxiliary files) each time you open it. SmartGen reports if any of the files associated with an existing core are missing. If a CXF (core configuration) file for a core is missing, SmartGen loads the core into the workspace with default parameters.

Import a SmartGen Core

You can import SmartGen cores into your current workspace.

To import a SmartGen core:

1. From the File menu, choose Import Core. This displays the Import Core dialog box (as shown in the figure below).



Import Core Dialog Box

2. Navigate to the core you wish to import and click Open. The core is added to your current workspace and remains in your workspace until you Remove it.

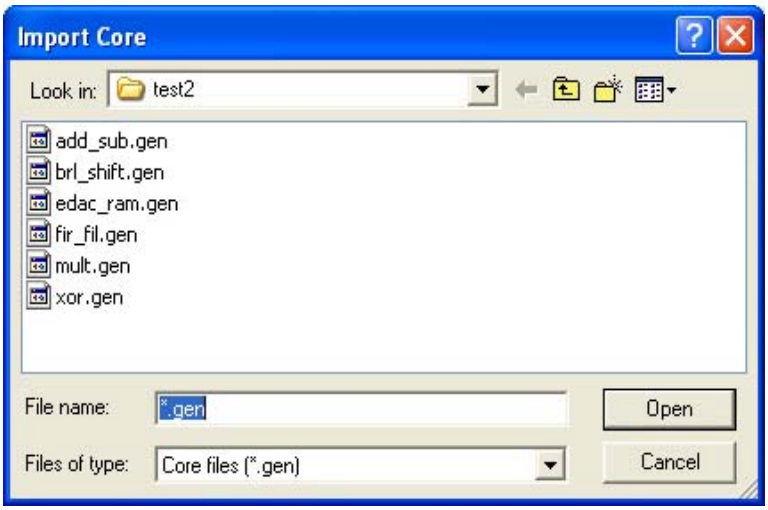
Note: You cannot import two cores with the same name. SmartGen is case-insensitive; "core_A" is equivalent to "core_a".

Import a Legacy Core

You can import SmartGen legacy cores into your current workspace. When you import a legacy core the import copies only the genfile into your workspace. You MUST regenerate the core to get a new netlist in SmartGen.

To import a legacy core:

1. From the File menu, choose Import Core. This displays the Import Core dialog box (as shown in the figure below).



Import Core Dialog Box

2. Navigate to your core you wish to import and click Open. The core is added to your current workspace and remains in your workspace until you Remove it.

   You can only import legacy cores that were generated for the family on which the workspace operates. If you import cores from other families, you get an error message, as shown in the figure below.

Import Core Error Message

Remove or Delete a Core

You can delete cores from your Configured Core View (and leave them on the disk), or you may delete them from your disk entirely. Select the core(s) you want to delete and press the **Delete** key, or right-click and choose **Remove from Workspace** or **Remove from Disk and Workspace**.

You do not need to save your workspace.

Removing a core from your workspace does not delete the core from your directory. You can import the core later if you wish. If you choose **Remove from Disk and Workspace** the core is deleted forever.

Save the Workspace

All changes to the workspace are saved automatically. You can save the workspace with a different name (useful if you want to create a copy of your workspace in a different directory).

To save your workspace, from the **File** menu, choose **Save Workspace As** and specify the **Name** and **Location**.

SmartGen Preferences

Use the Preferences dialog box to set the default directory for new workspaces. Whenever you try to Open a GEN file or generate a netlist, SmartGen uses the preferences you set here to generate your file.

Your default netlist output and family type are workspace settings.

To set or modify your General preferences:

From the **File** menu, choose **Preferences**. The Preferences dialog box (**General** tab) appears. To set the default directory for your workspaces, type the pathname, or click the browse button and navigate to your new default directory.

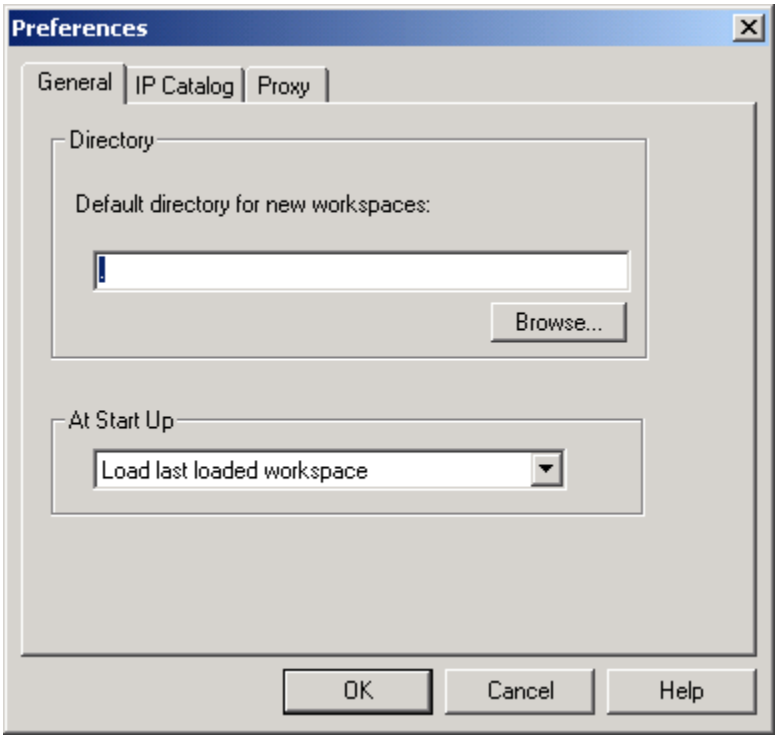
UNIX Only: Specify the location of your PDF reader and web browser.

In At Start Up, you can choose to **Load last loaded workspace (default)**, **Show [an] empty [work] environment**, or **Show the Welcome to SmartGen dialog box**.

To set your IP Catalog preferences:

Click the IP Catalog tab to specify when SmartGen checks for updates. The default is to check each time you open the tool. You can also specify the location of the IP Catalog info on your hard drive. Type the pathname, or click the Browse button and navigate to your new default directory.

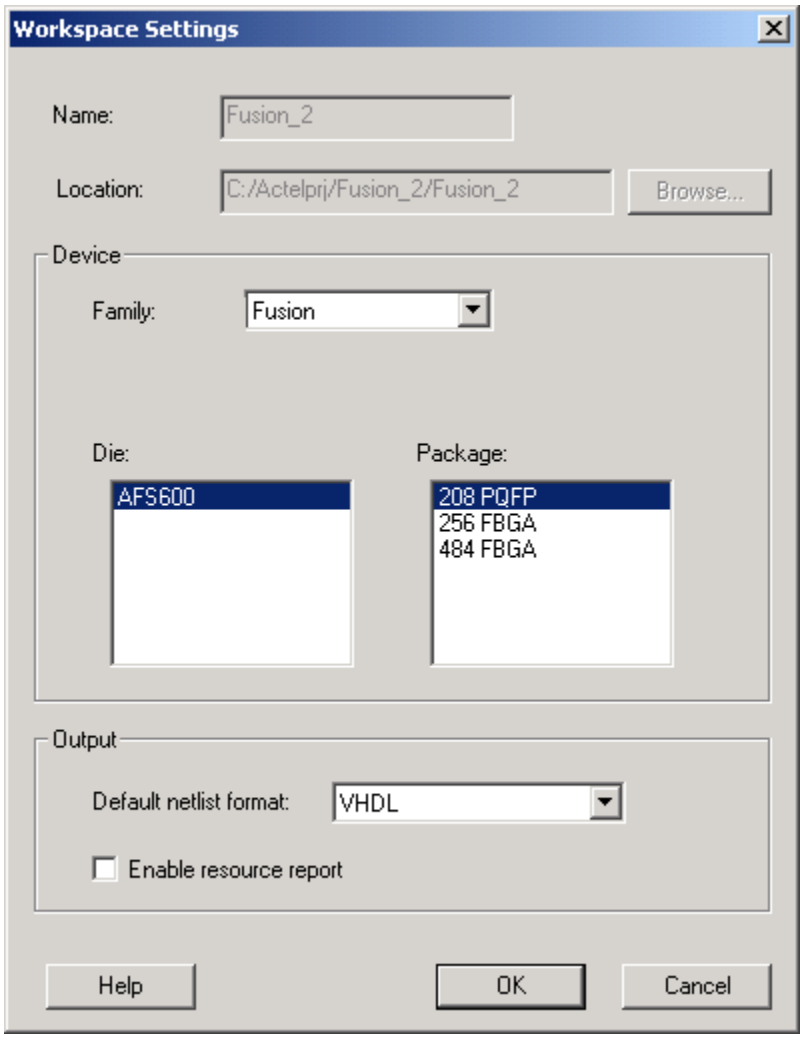
The Proxy tab enables you to set your internet proxy settings in SmartGen. This enables SmartGen to use an FTP connection to update some data files. If you use a proxy server, enter the server name in the Proxy field.



SmartGen Preferences Dialog Box

Workspace Settings

To change your workspace settings, from the **Options** menu, choose **Workspace settings**. If you wish to change your preferences, you can modify them in the File menu.



Workspace Settings Dialog Box

The Workspace Settings dialog box (above) specifies your device family, die, package, default output netlist format, enables or disables the Resource report (antifuse families only), and lists the device family you have selected for your workspace.

You may only change your Device Family if there are no cores in your workspace. If you wish to change families, delete all the cores from your workspace or create a new workspace.

Select the check box to enable the Resource report.

Antifuse families only: The Resource report calculates and displays the sequential, combinatorial, I/O, memory, and PLL resources used in the generated core as a percentage of the largest device in the family.

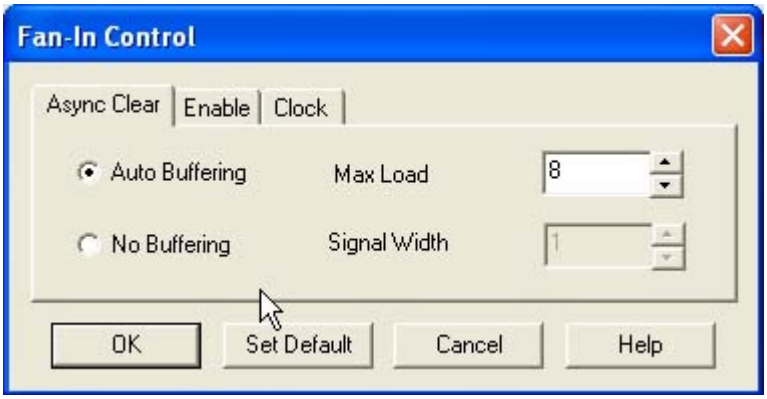
Generating Reports in SmartGen

As SmartGen generates a core it writes information to the Log window. The report contains information defining the core, and is divided into the following sections:

- **Core Parameters** - This section lists the options selected to build the core
- **Fan-in Control information** - This section defines the type of buffering for each control signal and the values used to distribute the total load
- (Optional) **Compile report** - Lists compile information related to the core

Fan-In Control

The Fan-In Control tool gives advanced users the ability to control the buffering of clocks, asynchronous presets and clears, and other control signals. This tool is optional because default buffering values are provided for all signals. The tool supports two types of buffering control, automatic and no buffering, which provide maximum buffering flexibility.



Fan-In Control Dialog Box

Using Fan-In Control

1. Set your core options.
2. Open the Fan-In Control dialog box and input your values. If you modify your core options after you set your fan-in values, you must check them to ensure that they are unaffected.

Auto Buffering

Automatic buffering inserts buffers as required, and provides ease of use for fanning out heavily-loaded signals. Automatic buffering is the default buffering option for most signals. The value defined for automatic buffering indicates the maximum loading on the

network for the given control signal. SmartGen provides a single input for the signal and automatically inserts buffers/inverters with this option. SmartGen also balances the loading as required.

No Buffering

The 'no buffering' option restricts SmartGen from inserting buffers. This allows designers to manually use global clock resources for control signals. This also provides the ability to enhance performance of control signals by performing a logic function and correcting for fan-in by duplicating logic external to the core. If the signal is to be driven by a clock resource, you must set the signal width on the clock to 1; a signal width value of one (1) causes all loads to be driven by a single input.

Fan-In Control Limitations

The Fan-In Control tool has the following limitations:

- The tool has been designed to be a slave to the primary core definition screen. Therefore, you should define exceptions to default values only after you have made all primary screen selections. Changing the main screen may affect the defined fan-in values. Information on modified fan-in will be provided in the Report window and should always be verified for correctness.
- The ability to perform no buffering on some control signals is limited to a single polarity because of hardware limitations. For example, ACT 2, ACT 3, 3200DX, MX, SX, SX-A, and eX limit asynchronous clears to Active Low only. Choosing Active High for this signal causes the No Buffering option to be unavailable. When this situation occurs, go back to the primary screen and change the active level for the given signal if no buffering is a must.
- Some control signals, such as the Count Enable signal are not included in the Fan-In Control tool because fan-out is corrected internally using AND and OR logic functions.

Use the Fan-In Control dialog box to specify Auto Buffering or No Buffering, Max Load, and Signal Width.

Create Cores

Create a New Core in SmartGen

You can create a new core in SmartGen only after you have created a workspace.

To create a new core in SmartGen:

1. Select a core type in the Categories tab (Arithmetic, Comparator, Converters, etc.). The Variety View window displays the list of configurable cores.
2. Double-click the core variety you wish to generate, or right-click and choose Create Core. The core configuration dialog box appears.

   The configuration dialog box varies depending on which core you select.
3. Set the parameters for your core and click Generate. The Generate Core dialog box appears. The Generate Core dialog box lists the names of all the configured cores already in your workspace.

4. Specify the name of your new core and click OK to continue. SmartGen saves your core and adds it to your workspace (in the Configured Core View Window). You cannot save your core anywhere but in your workspace.

   Your core remains in the workspace until you choose to **Remove** it.

Note: You cannot create two cores with the same name. SmartGen is case-insensitive; "core_A" is equivalent to "core_a".

Reconfigure an Existing Core in SmartGen

You can reconfigure your cores in SmartGen. To do so, you must first have added a core to your workspace.

To reconfigure a core in SmartGen:

1. Select the core in the Configured Core View window.
2. From the **Core** menu, choose **Modify Core**. The configuration dialog box opens to display the configuration options specified in the core. Your core configuration options vary depending on the device family you selected when you created your workspace.

SmartGen RAM Content Manager

The RAM Content Manager enables you to specify the contents of your memory so that you can avoid the simulation cycles required for initializing the memory, which reduces simulation runtime.

The SmartGen RAM core generator takes away much of the complexity required in the generation of large RAMs that utilize one or more RAM blocks on the device. SmartGen uses one or more memory blocks to generate a RAM matching your configuration. In addition, SmartGen also creates the surrounding cascading logic.

SmartGen cascades RAM blocks in three different ways.

- Cascaded deep (e.g. 2 blocks of 4096x1 to create a 8192x1)
- Cascaded wide (e.g. 2 blocks of 4096x1 to create a 4096x2)
- Cascaded wide and deep (e.g. 4 blocks of 4096x1 to create a 8192x2, in a 2 blocks width-wise by 2 blocks depth-wise configuration)

You specify memory content in terms of your total memory size. SmartGen must partition your memory file appropriately such that the right content goes to the right block RAM when multiple blocks are cascaded.

Supported Formats

Intel-Hex Record Format

A standard format created by Intel. Memory contents are stored in ASCII files using hexadecimal characters. Each file contains a series of records (lines of text) delimited by new line, '\n', characters and each record starts with a ':' character. For more information regarding this format, refer to the Intel-Hex Record Format Specification document available on the web (search **Intel Hexadecimal Object File** for several examples).

Motorola S-Record Format

This format uses ASCII files, hex characters, and records to specify memory content in much the same way that Intel-Hex does. Refer to the Motorola S-record description document for more information on this format (search **Motorola S-record description** for several examples). The RAM Content Manager uses only the S1 through S3 record types; the others are ignored.

The major difference between Intel-Hex and Motorola S is the record formats, and some extra error checking features that are incorporated into Motorola S.

In both formats, memory content is specified by providing a starting address and a data set. The upper bits of the data set are loaded into the starting address and leftovers overflow into the adjacent addresses until the entire data set has been used.

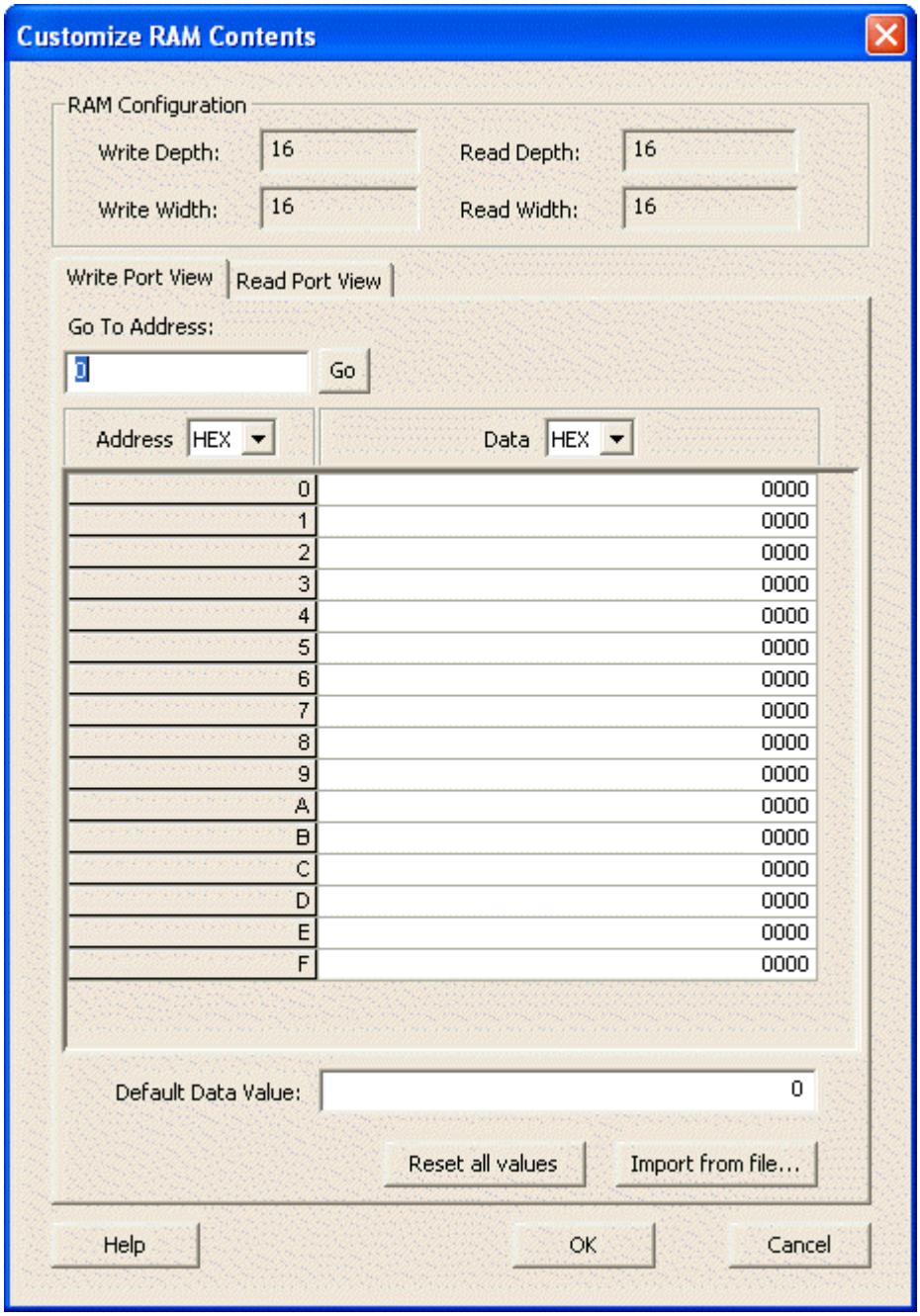
The Actel implementation of these formats interprets data sets in bytes. This means that if the memory width is 7 bits, every 8th bit in the data set is ignored. Or, if the data width is 9, two bytes are assigned to each memory address and the upper 7 bits of each 2-byte pair are ignored.

Using the RAM Content Manager

Using the RAM Content Manager is only possible if the device family supports the RAM Content Manager features (Axcelerator, ProASIC3, or ProASIC3E).

To open the RAM Content Manager:

1. From the **Options** menu, choose **Workspace Settings** and set your device family to **Axcelerator, ProASIC3,** or **ProASIC3E**. Click **OK**.
2. Click **RAM** in the Core Catalog to display the list of RAM types available for Axcelerator.
3. Double-click **Synchronous RAM** to create a new RAM block for Axcelerator. Specify your RAM settings (set your Read and Write Depth and Width), select the **Initialize RAM** checkbox, and then click **Customize RAM Content**. The RAM Content Manager appears, as shown in the figure below.



RAM Content Manager in SmartGen

RAM Configuration

Write Depth and Write Width - As specified in the RAM core generator dialog box (not editable).

Read Depth and Read Width - As specified in the RAM core generator dialog box (not editable).

Write Port View / Read Port View

Go To Address - Enables you to go to a specific address in the manager. Each memory block has many addresses; it is often difficult to scroll through and find a specific one. This task is simplified by enabling you to type in a specific address. The number display format (Hex, Bin, Dec) is controlled by the value you set in the drop-down menu above the Address column.

Address - The Address column lists the address of a memory location (you cannot specify the address of a memory location). The drop-down menu specifies the number root for your address list (hexadecimal, binary, or decimal).

Data - Enables you to control the data format and data value in the manager. Click the value to change it.

The RAM Content Manager enables you to Import or Export your files through either port.

Files are imported into whichever view you have selected. Importing files through the Write and Read ports with different aspect ratios results in completely different outcomes for your data.

Import from file (Write Port View)- Opens the Import Memory Content - Write Port View dialog box; enables you to select a memory content file (Intel-Hex, Motorola S) to load through the Write Port. During import, file extensions are set to *.hex for Intel-Hex files and *.s for Motorola S files.

Import from file (Read Port View) - Opens the Import Memory Content - Read Port View dialog box; enables you to select a memory content file (Intel-Hex, Motorola S) to load through the Read Port. During import, file extensions are set to *.hex for Intel-Hex files and *.s for Motorola S files.

Default Data Value - The value given to memory addresses that have not been explicitly initialized (by importing content or editing manually). When changed, all default values in the manager are updated to match the new value. The number display format (Hex, Bin, Dec) is controlled by the value you set in the drop-down menu above the Data column.

Reset All Values - Resets the Data values.

Help - Opens the RAM Content Manager online help.

OK- Closes the manager and saves all the changes made to the memory and its contents.

Cancel - Closes the manager, cancels all your changes in this instance of the manager, and returns the memory back to the state it held before the manager was opened.

MEMFILE (RAM Content Manager Output File)

Transfer of RAM data (from the RAM Content Manager) to test equipment is accomplished via MEM files. The contents of your RAM is first organized into the logical layer and then reorganized to fit the hardware layer. Then it is stored in MEM files that are read by other systems and used for testing.

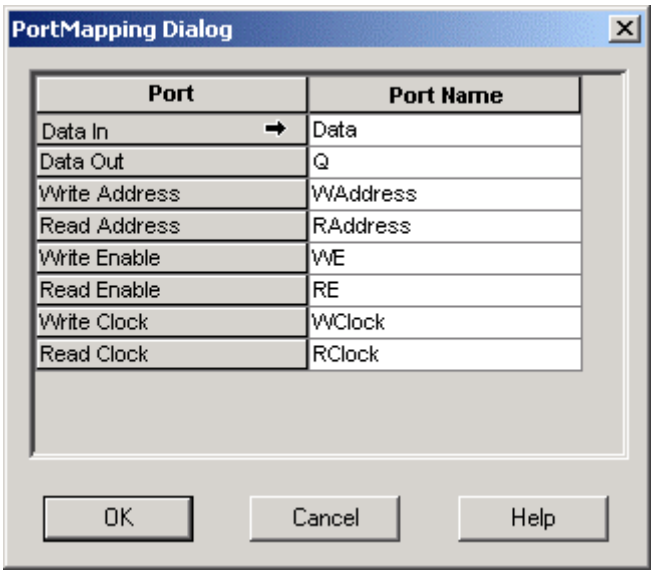
The MEM files are named according to the logical structure of RAM elements created by SmartGen. In this scheme the highest order RAM blocks are named CORE_R0C0.mem, where “R” stands for row and “C” stands for column. For multiple RAM blocks, the naming continues with CORE_R0C1, CORE_R0C2, CORE_R1C0, etc.

The data intended for the RAM is stored as ASCII 1s and 0s within the file. Each memory address occupies one line. Words from logical layer blocks are concatenated or split in order to make them fit efficiently within the hardware blocks. If the logical layer width is less than the hardware layer, two or more logical layer words are concatenated to form one hardware layer word. In this case, the lowest bits of the hardware word are made up of the lower address data bits from the logical layer. If the logical layer width is more than the hardware layer, the words are split, placing the lower bits in lower addresses.

If the logical layer words do not fit cleanly into the hardware layer words, the most significant bit of the hardware layer words is not used and defaulted to zero. This is also done when the logical layer width is 1 in order to avoid having left over memory at the end of the hardware block.

Port Mapping Dialog Box

You can use the Port Mapping function to specify the port naming for cores. Click the **Port Mapping** button to open the Port Mapping dialog box.

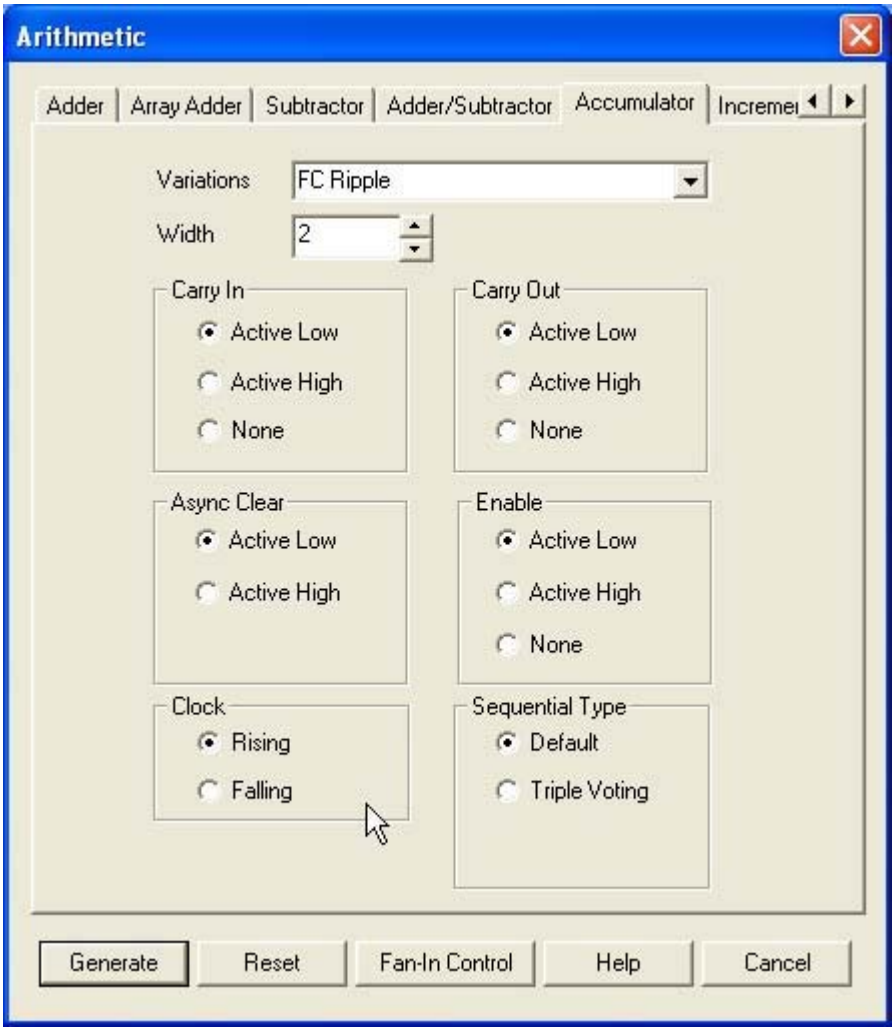


Port Mapping Dialog Box

The Port Mapping dialog box appears and displays the default port name values. Enter changes and click **OK** to submit, or click **Cancel** to return to the default values.

Fast Carry Chains (Axcelerator Only)

The Axcelerator family offers fast carry chain cores for a compact design of Arithmetic Macros and Counters. Fast-Carry cores for Axcelerator are available in the **Variations** drop-down menu.



Fast-Carry-Chain Cores in Variations Drop-Down Menu

You can generate FC cores via the SmartGen module generator or infer them with synthesis tools such as Synplicity Synplify or Synopsys Designware (DWACT). FC cores are always the most area-efficient way to implement these modules. They are also superior in performance for designs up to 32 bits, although some modules may be inferior beyond 32-bits (incrementers, for example). Though SmartGen offers both architecture types (FC cores and non-FC cores), Actel recommends you use FC cores to guarantee area efficiency.

In the SmartGen GUI, you can distinguish the FC cores from non-FC cores by the prefix “FC” or "Fast Carry" (e.g. “FC High Speed” versus “High Speed”). The GUI also lists a description of the macro in the Details column in the Variety View.

The core parameters used in the GEN file also use “FC” for distinction. For example the “High Speed” Adder using fast carry chains is specified by:

LPMTYPE:LPM_FC_ADD_SUB

LPM_HINT:FC_FADD

while the corresponding non-FC version is specified by:

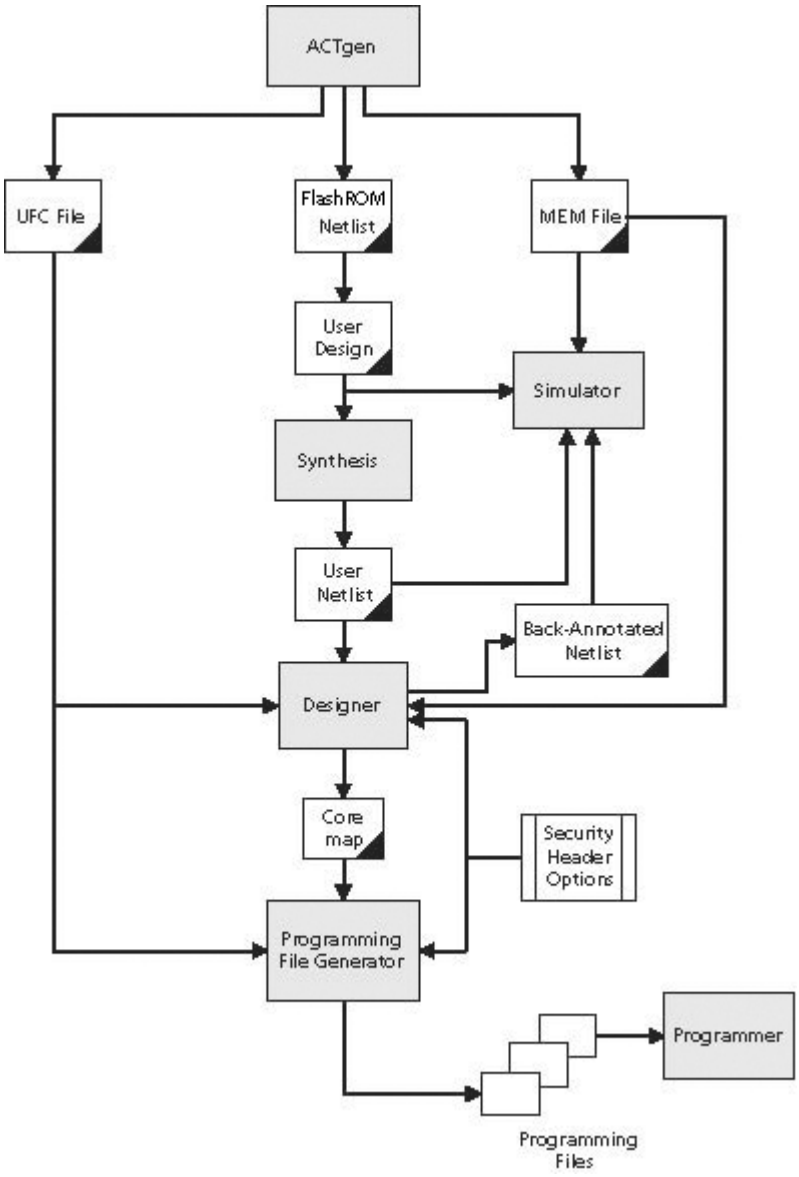
LPMTYPE:LPM_ADD_SUB

LPM_HINT:FADD

FlashROM

Welcome to FlashROM

FlashROM memory provides the security of stored data in addition to a 128-bit AES decryption core. You can read, modify, and write to the FlashROM using the JTAG interface; however, you can only read it from the FPGA core.



FlashROM Flow

Access FlashROM from the SmartGen tool. It opens a special FlashROM core generator that enables you to configure the FlashROM functionality.

Note: FlashROM is available only for ProASIC3 and ProASIC3E devices.

ProASIC3/E devices have a flexible programming option. The FlashROM and the FPGA core fabric can be programmed independently of each other, allowing the FlashROM to be updated without changing the FPGA core fabric. The following are just a few examples of possible applications for the FlashROM feature:

- Internet protocol (IP) addressing (wireless or fixed)
- System-calibration settings
- Device serialization and/or inventory control
- Subscription-based business models (e.g. set-top boxes)
- Secure key storage
- Asset management tracking
- Date stamping
- Version management

The FlashROM is programmed using the standard IEEE1532 JTAG programming interface. Pages can be individually programmed (erased and written) and on-chip AES decryption can be used selectively to load data securely into the FlashROM (such as application-based security keys stored in the FlashROM for a design). See the FlashPoint help or user's guide for information on how to program your FlashROM-enabled devices.

The FlashROM can selectively be read back either through the JTAG programming interface or via direct FPGA core addressing. Its contents can only be updated via the JTAG interface. A seven-bit address from the FPGA core defines which of the eight pages (3 MSBs of the addresss) is being read and which of the 16 bytes in the page (4 LSBs) are being read.

The FlashROM is physically organized as 8x128 bit blocks and logically organized as eight pages by 16 bytes. Only Flash FPGAs contain on-chip nonvolatile memory (NVM); Actel's Fusion and ProASIC3/E devices are the only FPGAs to support this feature.

You can assign specific regions of the FlashROM for specific purposes by floorplanning the FlashROM and assigning properties. The content of these regions can be modified during programming time if you assign a modifiable content property to a given region. If you do not want the FlashROM content to be modified, you can fix the content in SmartGen.

When you generate a new FlashROM file, the generator saves the following files for you to use throughout the design cycle:

- SmartGen GEN file
- **Netlist file** - use this file to instantiate your core, just as you would instantiate any other core in your design
- **UFC file** - User Flash configuration file; it contains all the configuration information regarding the FlashROM data content and is used for programming. You can export a core map file that contains the core programming information and use it along with

the UFC file to generate programming files. Designer software supports importing the UFC file and launching the programming file generator to merge the FPGA core map file and the FlashROM programming file.

- **MEM file** - FlashROM specific memory initialization file. The MEM file has 128 rows of eight bits, representing the contents of the FlashROM. SmartGen will default to 0s for any unspecified locations of the FlashROM memory. This file is used exclusively for simulation.

Use the FlashROM help to:

- Configure FlashROM in SmartGen
- Simulate Pre/Post Synthesis
- Synthesize
- Place-and-Route
- Run Back-Annotation and Timing Simulation
- Specify security settings
- Specify FlashROM content
- Generate a programming file

Create/Configure FlashROM in SmartGen

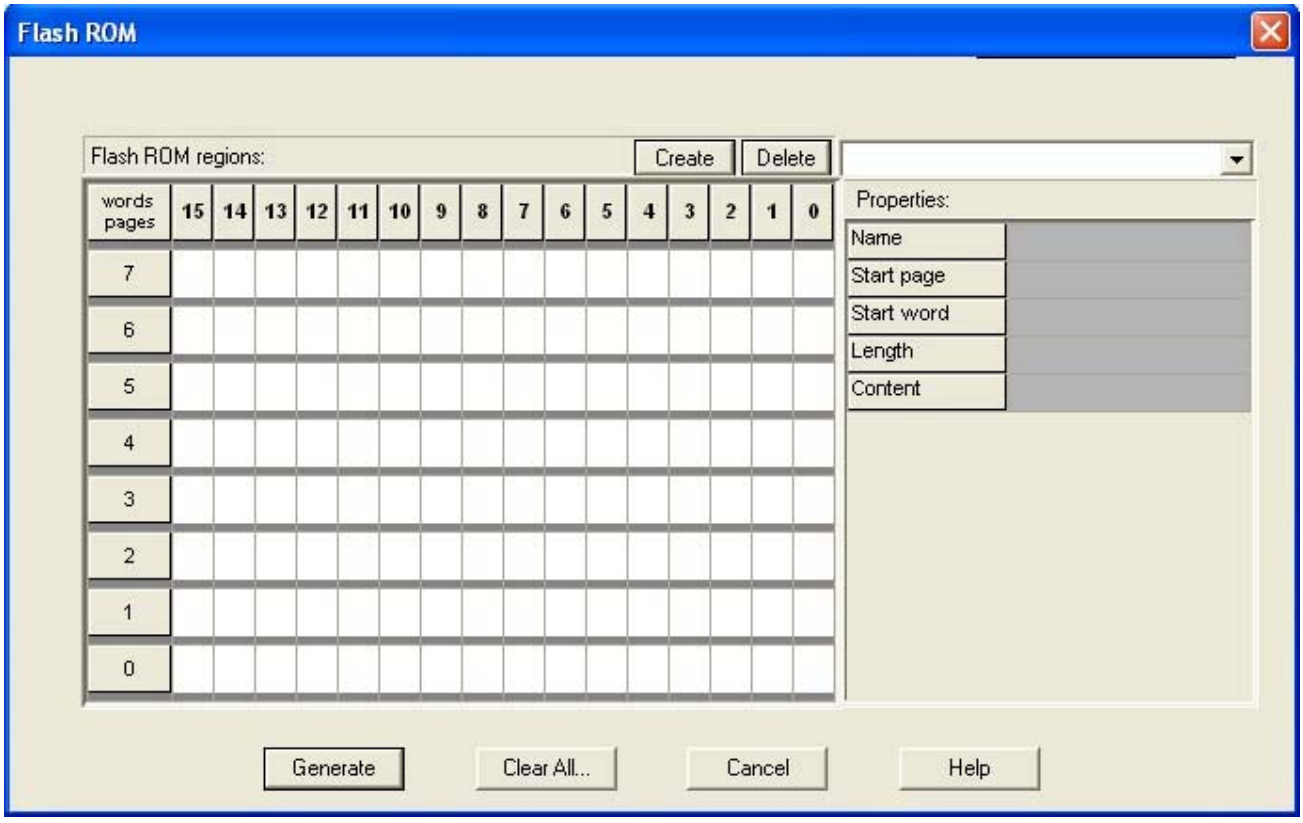
The FlashROM can be partitioned into regions and each region can be used for a specific purpose, like serial number storage, version number saving, etc.

Use the FlashROM core generator (in SmartGen) to create a region within a page, modify the region, and assign properties to that region.

The FlashROM user interface includes the Configuration Grid, a list of existing regions, and the Properties field. The Properties field includes region-specific information. You can assign values to the following properties:

- **Static Fixed Data** - Enables you to fix the data so that it cannot be changed during programming time. This option is useful when the you have fixed data stored in this region that is required for the operation of the design in the FPGA. Key storage is one example.
- **Modifiable Fixed Data** - Select this option when the data in a particular region is expected to be static data (such as a version number, which remains the same for a long duration, but could conceivably change in the future). This option enables you to identify this region so that you need not come back and change the value every time you enter new data.
- **Read from File** - This provides the customer the full flexibility of FlashROM usage. If you have a customized algorithm for generating the FlashROM data, you can specify this setting. You can then generate a file with data for as many devices as you wish to program and load that into FlashPoint programming file generation software to get programming files that include all the data. SmartGen optionally passes the location of the file where the data is stored, if you specify the file in SmartGen.

- **Auto Increment/Decrement** - This scenario is useful when you specify the contents of FlashROM for a large number of devices in a serial manner. You can specify the step value for the serial number and a maximum value for inventory control. During programming file generation, the actual number of devices to be programmed is specified and a start value is input to the software. Software generates the files to complete serial programming of the FlashROM.



FlashROM Core Generator

To create a new FlashROM in SmartGen:

1. In SmartGen, choose the **Flash ROM** in the **Function** tab. Double-click the **FlashROM** core in the **Variety View** window to start the core generator.
2. Click and drag the mouse to select words, then click the **Create** button. The core generator displays the new region properties in the **Properties** grid.

   You may also right-click a word and choose **Create** from the shortcut menu, or select a word and press the **Insert** key on your keyboard. You can copy and paste regions in FlashROM; to do so, right-click a word and choose **Copy**, then click an empty word, right-click, and choose **Paste**. If the region is not copied, the page does not have enough room. Try another page with more room.

3. Click in the **Properties** grid to modify a regions properties. **Start page**, **Start word**, and **Length** are read-only. The data you enter is verified and stored in the FlashROM as soon as you leave the Properties grid and select another FlashROM region.
4. Click **Generate** to generate a Netlist (you must specify EDIF, VHDL, or Verilog), SmartGen GEN file, UFC, and MEM file. The **Generate** button opens the **Save As** dialog. Specify a name, location, and file type, and click **Save**.

   After you generate the FlashROM netlist, you can instantiate the core similar to other SmartGen cores in your design.

To delete a FlashROM Region:

1. Click to select a region in the **Regions** window.
2. Click the **Delete** button in the core generator, press the **Delete** key on the keyboard, or right-click and choose **Delete** from the shortcut menu.
3. Click **OK**.

See the FlashPoint help or user's guide for information on how to program your FlashROM-enabled devices.

Modify Existing FlashROM Configuration

You can modify your existing FlashROM configuration the same way that you modify any configured core in SmartGen. To do so:

1. Open your SmartGen workspace that contains the configured FlashROM core.
2. Double-click the configured core (in the Configured Core View) to open the FlashROM configuration. The FlashROM core opens with all the settings you saved.
3. Modify the values you wish to change and click **Generate** to save your changes. **Generate** opens the **Save As** dialog. Save your new file with the same name if you wish to overwrite the old file.

   You cannot edit the configuration of an existing FlashROM GEN file, only the data. If you wish to change the configuration you must generate a new core.

Simulate Pre-/Post-Synthesis

FlashROM uses the MEM file for simulation.

The MEM file has 128 rows of eight bits, representing the contents of the FlashROM. SmartGen defaults to 0s for any unspecified locations of the FlashROM memory.

During simulation, employ the MEM file, which contains the memory content, along with the design netlist and testbench. The VITAL and Verilog simulation models accept the generics passed by the netlist, read the MEM file, and perform simulation with the data in the file.

In addition to using the MEM file from SmartGen, you may create a binary file with 128 rows of eight bits and save the file as a MEM file. Actel recommends using different names if you plan to generate multiple MEM files. During place-and-route in Designer, the software recognizes the generic property in the netlist and passes the MEM file links through to the output netlist.

Place-and-Route and FlashROM

There are no special instructions for place-and-route for the FlashROM. Run Layout in Designer to place-and-route your design.

Analog System Builder

Welcome to the Analog System Builder

The Analog System Builder in SmartGen enables you to configure an entire analog system. You can:

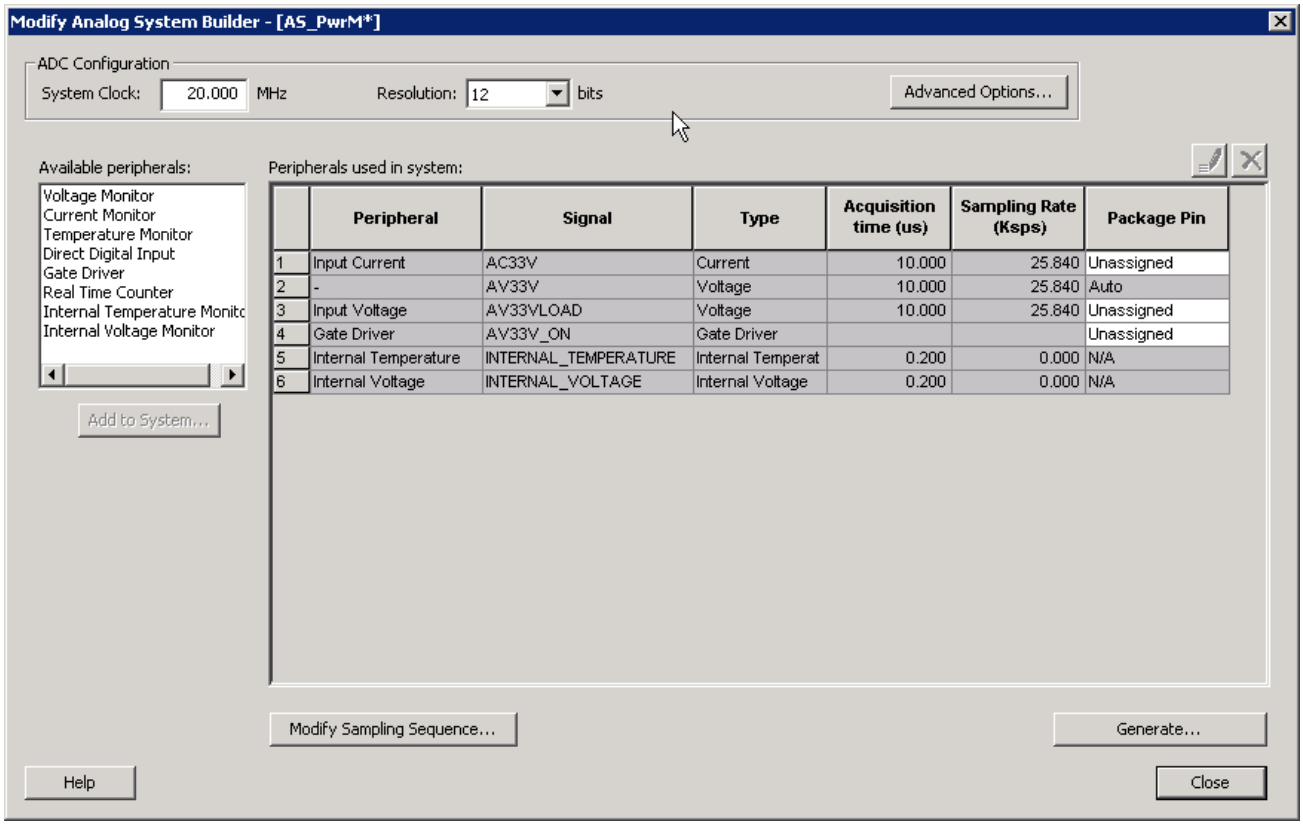
- Choose the number of Analog Input Channels to monitor
- Choose the type of each Input Channel
- Choose the number of Analog Output Channels
- Specify the placement of each channel
- Set Channel-specific options
- Sequence the channels in the required sampling order
- Define the operations on converted digital output from the ADC
- Specify the RTC settings.

The Analog System Builder enables you to create, configure, and place the following analog blocks (or "peripherals"):

- Voltage Monitor
- Current Monitor
- Temperature Monitor
- Direct Digital Input
- Gate Driver
- Real Time Counter
- Internal Temperature Monitor
- Internal Voltage Monitor

Analog System Builder Main Window

The Analog System Builder main window enables you to create and configure your analog system (as shown in the figure below).



Analog System Builder Dialog Box

The number of peripherals you can add to the system is limited by the size of your device.

The Analog to Digital Converter (ADC) Configuration sets your System Clock speed and resolution.

Click **Advanced Options** to set your Analog System configuration.

Available Peripherals lists all the analog peripherals you can add to your design. As you add peripherals, some resources are exhausted. The Analog System Builder disables peripherals in the list for which there are insufficient resources. If you want to add additional peripherals, select a larger device, or remove some existing peripherals from your design.

The **Peripherals used in system** grid lists specific information about each peripheral, including

- Peripheral - The type of the peripheral (such as Voltage Monitor, Temp. Monitor, etc.).
- Signal - Name you specified for the signal of your service in the service configuration dialog box.
- Type - Identifies channel type for the service.

- Acquisition Time - The required acquisition time for a given input channel. SmartGen takes the required acquisition times for all peripherals and computes the ADC clock frequency and the number of ADC clocks per sample and per peripheral, so that each peripheral meets or exceeds your required acquisition time.
- Sampling Rate (in µs) - The rate at which a given channel is sampled by the ADC. See the Modify Sampling Sequence topic for more information on setting your sampling rate. Sampling rate is affected by the following parameters:

  System Clock Frequency

  ADC Resolution

  Number of channels in the system

  Settling times for all channels

  Number of Flags specified per channel

  Operating Sample Sequence
- **Package Pin** - SmartGen automatically assigns a package pin for each channel in each peripheral added to the system. However, if you require a specific channel for a certain package pin (if you have board layout issues), you can choose a specific pin for that channel.
- **Real Time Counter** - You can configure the Real Time Counter so that it functions as a chronometer, allowing it to generate periodic alarms in conjunction with other peripherals (such as the Voltage monitor, etc.).
- **Modify Sampling Sequence** - Displays the Sample Sequencer. Since there are thirty analog input channels but only one ADC, the channels must be sequenced in the order in which they are to be sampled.

If the Analog System resources you build exceed the total system resources available for your device, SmartGen issues a warning. You cannot generate a system that exceeds your total system resources. The Analog System Builder also generates a warning if you have a port name conflict between two or more services. You cannot generate a system with port name conflicts.

When you click **Generate the system**, SmartGen creates HDL source files, memory (MEM) files, configuration files, and log files. They all appear in your SmartGen project folder under the <core_name> directory. Do not modify any of these generated files or store additional files in this folder. This folder will be recreated every time you overwrite the core.

Modify Sampling Sequence

Since there are 30 input channels but only one ADC, the channels must be sequenced in the required order for the system to function. There are 64 sequencing time slots available. You can run operations on the ADC in addition to sampling the channels.

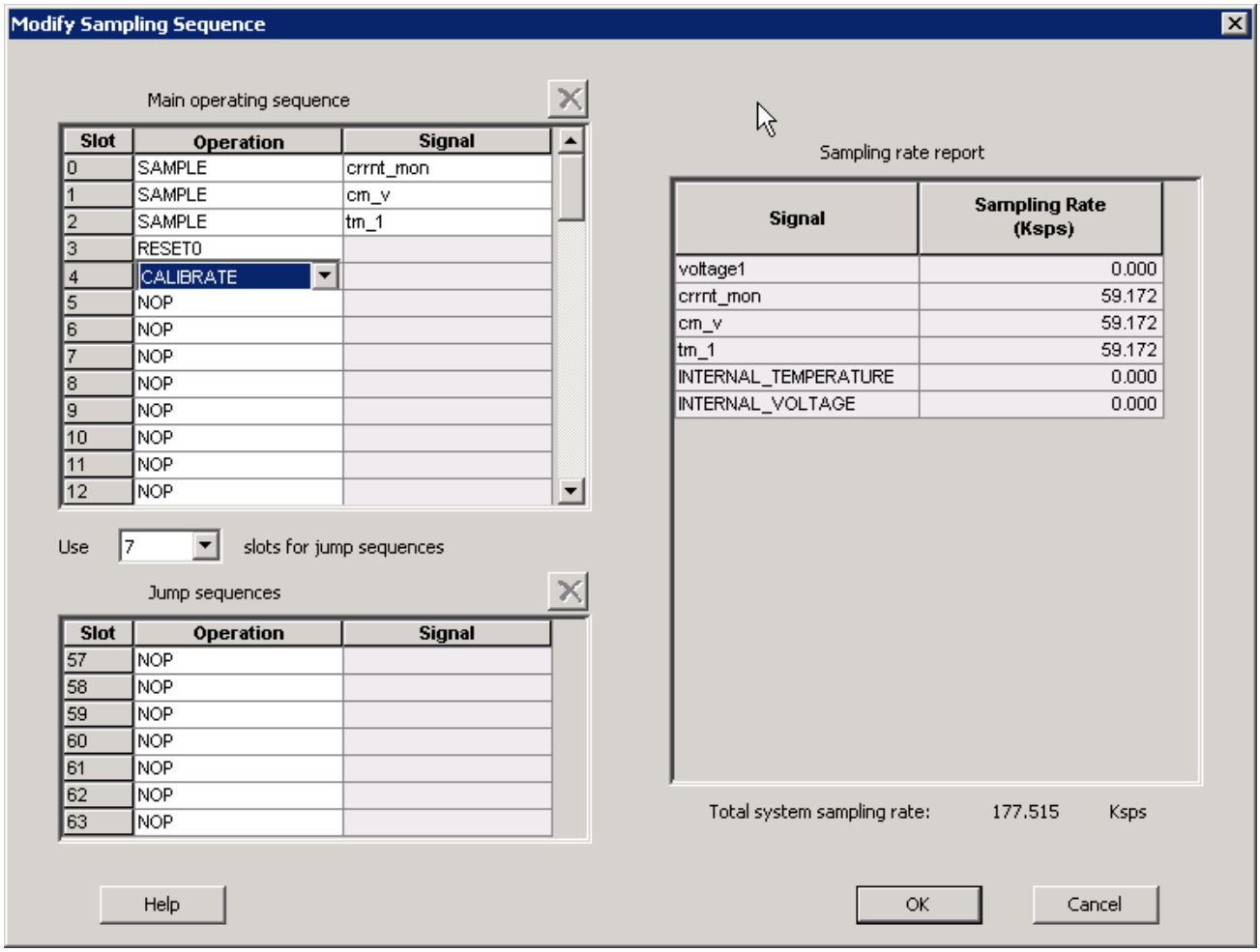
The sampling sequence specifies the Analog System's sampling order. For example, the sequence may be specified to sample channel 1 continuously, or it can be specified to sample channels 1 through 5, then repeat. In either case, the Sample Sequence Controller will drive the ADC signals to sample the channels in the specified sequence.

Your application and requirements dictate the sampling sequence. SmartGen enables you to either enter their sampling sequence manually or let SmartGen compute the sequence using your variables (the variables are explained below).

SmartGen provides you with two operating sequences: "main" and "jump". The main operating sequence operates the majority of the time in a constant loop. The jump sequence is used primarily for interrupt events and must be triggered by user logic.

The jump sequence is not a factor in the sampling rate calculations for the channels or the system.

Functions can be inserted into the sequence that do not perform any sampling, but are used to configure the ADC. These functions would probably affect the sampling rate of the system. The Modify Sampling Sequence dialog box is shown below.



Modify Sampling Sequence Dialog Box

Main Operating Sequence

This is the main sequence with SAMPLE, SAMPLE_RESET0, RESET0, or CALIBRATE as the last operation (so that the sequence repeats itself). You can go to a slot in the jump sequence, and return to slot 0 to restart the main operating sequence. The operations in the main sequence are:

- SAMPLE - Sample a channel that is added to the system

- SAMPLE_RESET0 - Sample a channel that has been added to the system and reset to Slot 0
- RESET0 - Reset to Slot 0
- CALIBRATE - Calibrate the analog-to-digital converter.

If you do not specify a terminating option in the main operating sequence (no RESET0) the sequence continues up to the 64th slot and resets to slot 0, with no operations on the intermediate slots (NOP).

SAMPLE_RESET0 completes the sample and reset action in the same time slot. SAMPLE and RESET0 require two time slots.

Jump Sequences

These are extra sampling options that are accessed via an external jump. There are two external jump modes:

- **Manual** - The system completes operations in the current slot, then waits for the signal to move to the next slot.
- **Auto Forwarding** - The system completes the operation in the jump slot and moves to the next slot, until the sequence reaches slot 63, at which point it begins sampling slot 0, or resets to slot 0 if it is a RESET0 operation.

If the operation is a STOP or POWERDOWN, the sequencer stops processing until another jump is initiated to a different slot. STOP stops the ADC from sampling. POWERDOWN forces ADC to power down. Returning from this state requires calibration of the ADC.

You can reserve the number of slots required for jump sequence. The slot with the smallest number used in the jump sequence determines how many slots can be made available to the main operating sequence.

Resulting sampling rate - The number of times that given channel gets sampled in a second, expressed in Ksps (thousands of samples per second).

Analog System Builder Output Files (from SmartGen)

SmartGen creates the following output files in your project directory when you generate an analog system.

HDL Source Files

<user_name>.vhd/.v – Top-level design that combines all the blocks together

<user_name>_assc_wrapper.vhd/.v – Analog system controller instantiation wrapper for this design

<user_name>_smev_wrapper.vhd/.v – Analog system data processing module instantiation

<user_name>_smtr_wrapper.vhd/.v – Analog system data processing module instantiation

<user_name>_assc_ram.vhd/v – Analog system controller RAM

<user_name>_smev_ram.vhd/v – Analog system data processing module RAM

<user_name>_smtr_ram.vhd/v – Analog system data processing module RAM

<workspace_directory/<common>/<Vhdl>/<Verilog>/assc.vhd/v – Analog sample sequence controller file; common file for all analog system cores.

<workspace_directory/<common>/<Vhdl>/<Verilog>/smev.vhd/v – Analog system data processing module file, common for all analog cores.

<workspace_directory/<common>/<Vhdl>/<Verilog>/smtr.vhd/v – Analog system data processing module file, common for all analog cores.

Memory Files

These memory files are used by the Flash Memory System (or any external microprocessor) to initialize the contents of the RAM and AB.

<user_name>_acm_ram.hex/ .s - Intel-hex or Motorola-S memory files for AB Hard IP

<user_name>_assc_ram.hex/ .s - Intel-hex or Motorola-S memory files for ASSC RAM

<user_name>_smev_ram.hex/ .s - Intel-hex or Motorola-S memory files for SMEV RAM

<user_name>_smtr_ram.hex/ .s - Intel-hex or Motorola-S memory files for SMTR RAM

The memory files below are used to initialize the RAM contents for simulation only. These files enable simulation of the Analog system in isolation (there is no need to connect the initialization circuitry).

<user_name>_acm_R0_C0.mem - Memory File for simulation for AB

<user_name>_assc_ram_R*_C*.mem - Memory Files for simulation for ASSC RAM

<user_name>_smev_ram_R*_C*.mem - Memory Files for simulation for SMEV RAM

<user_name>_smtr_ram_R*_C*.mem - Memory Files for simulation for SMTR RAM

Configuration Files

<user_name>.ncf – The embedded Flash configuration file used to communicate information from the Analog System to the Flash Memory system regarding the size of the Analog System Client and the location of the memory content.

<user_name>.cfg – This captures information about the settings that were specified for the system.

<user_name>.gen – The SmartGen GEN file. Enables SmartGen to open the system with your saved specifications.

<user_name>.cxf – The SmartGen Core Configuration file that contains information required by the Libero IDE for file management.

Log Files

The log file contains all the information SmartGen used to generate your system, as well as any messages related to conflicts or system resource limitations. The file is called <user_name>.log.

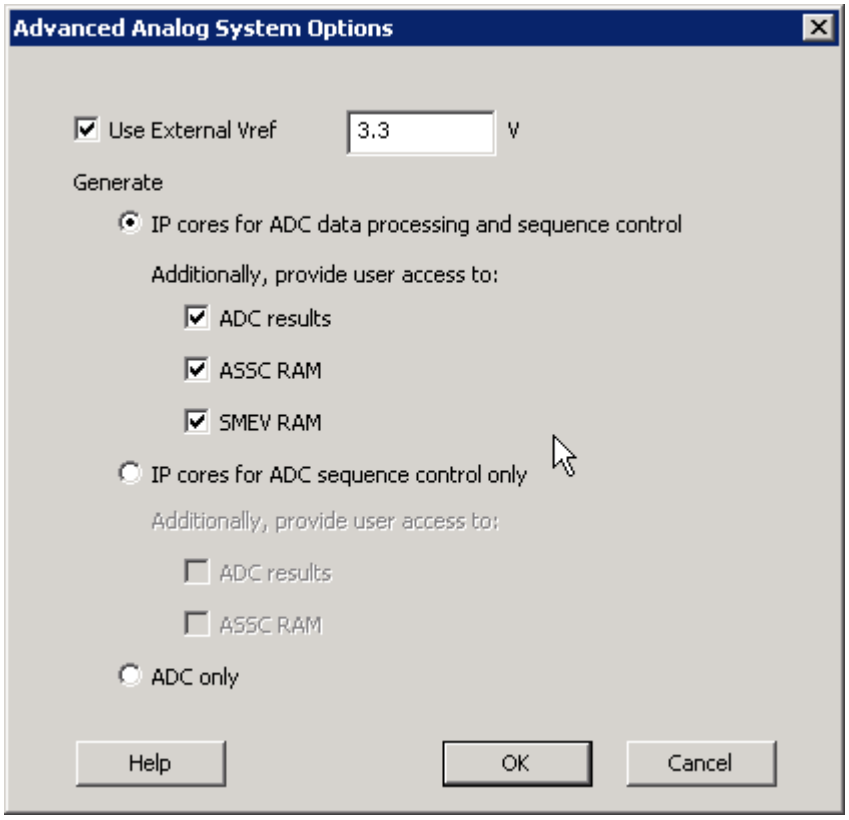
Analog System Builder Reference

The Analog System Builder introduces some terminology that is new to Actel. Here is a list of terms and acronyms that appear in the software and the help.

Term	Description
ADC	Analog-to-digital converter
ASSC	Analog sample sequence controller Soft IP - part of the analog system
Analog System	The complete system, including the analog block (AB) hard IP and one or more of ASSC, SMEV, and SMTR soft IPs.
SMEV	Data Evaluation Soft IP - part of the analog system
SMTR	Transition Soft IP - part of the analog system
AB	Analog block - The hard macro in the CAE library that includes the analog MUX and the ADC
Analog MUX	The 32 -1 MUX, select signals of which determine the channel being sampled by the analog to digital converter.
ACM	Analog Configuration MUX - stores configuration data related to analog channels (channel type, pre-scalar value, polarity, etc.)

ASB Advanced Options

The Advanced Options in the Analog System Builder enable you to set the external reference voltage and generate custom system configurations (as shown in the figure below). Some custom configurations (such as **ADC only**, and **IP cores for ADC sequence control only**) disable some of the functionality in the Analog System.



Analog System Builder Advanced Options Dialog Box

Setting your External VREF (external voltage) enables you to use a more accurate Analog to Digital Converter (ADC) reference voltage, allowing more accurate ADC conversions.

This VREF value specifies the voltage reference driven into the VREF interface of the analog block. If you do not enter a value then ASB uses an internal VREF of 2.56 V; applying an external VREF affects the threshold computations.

This implies that when modifying the External VREF, the legal range of thresholds for current and voltage may be altered.

Note: The legal threshold range for the temperature monitor is not based on VREF.

Therefore, it is possible that you could create a current or voltage peripheral, set up some flags with threshold values, and then decide to use and change the external VREF. This could lead to errors in the existing flag thresholds.

If you invalidate your flag thresholds by setting an external VREF, the ASB main window displays an icon notifying you of errors in the configured peripheral.

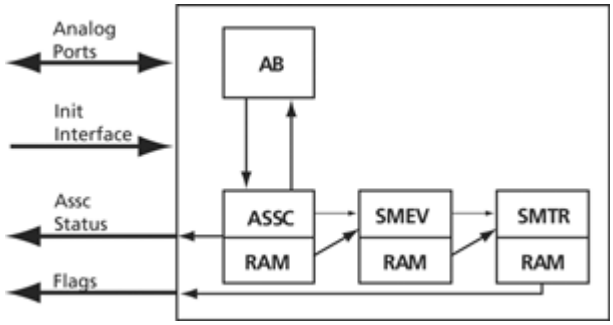
The ASB Advanced Options dialog box enables you to generate the following in your Analog Block:

- IP Cores for ADC data processing and sequence control
- IP Cores for ADC sequence control
- ADC only

Each of these is described below.

IP Cores for ADC Data Processing and Sequence Control

Enables all the Analog Block features: sequencing, flag generation, data averaging, and general ADC management. You can enable or disable access to ADC results, ASSC RAM, and SMEV RAM. ASSC RAM is responsible for setting the sample order in the ADC, and SMEV RAM evaluates the converted analog data. This option instantiates the Analog Block and the complete Analog System Controller (includes ASSC RAM, SMEV RAM, and SMTR RAM), as shown in the figure below.



System Diagram for IP Cores and ADC Data Processing and Sequence Control Options

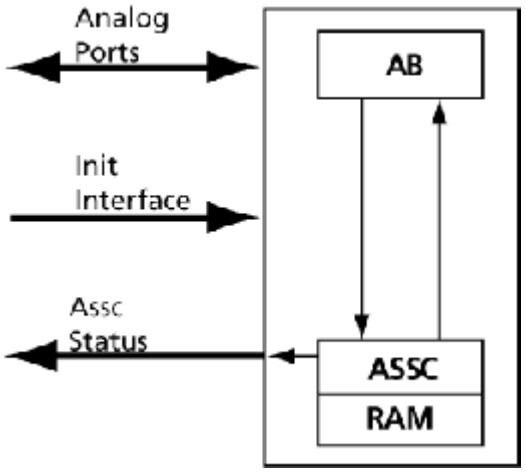
This option generates the following files:

- ACM MEM files
- ASSC IP, ASSC RAM, ASSC Wrappers, & ASSC MEM files
- SMEV IP, SMEV RAM, SMEV Wrappers & SMEV MEM files
- SMTR IP, SMTR RAM, SMTR Wrappers & SMTR MEM files

Enabling user access to ADC results, ASSC RAM, and SMEV RAM exposes additional interfaces and ports. See the help topics associated with each option for more information.

IP Cores for ADC Sequence Control

This configuration instantiates only the analog block model and the ASSC RAM. The data processing portions of the controller (SMEV and SMTR RAM) are omitted from the design (as shown in the figure below). If you select this option, you must process the ADC data directly from the ADC RESULT bus or the ASSC RAM.



System Diagram for IP Cores for ADC Sequence Control Only

This configuration disables flag generation for peripherals (the flag grid for peripherals); data averaging (Digital Filtering Factor and Initial Averaging value); SMEV RAM access; and the ability to specify the external resistor in the Current Monitor.

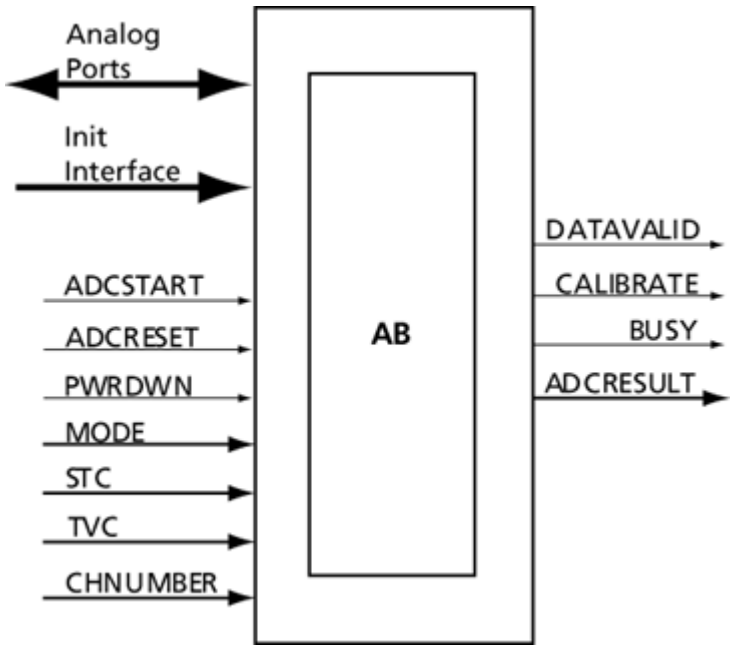
Note: You must explicitly choose to expose the ADC result and/or ASSC RAM data interfaces to gain access to the ADC data.

IP cores for ADC sequence control generates the following files:

- ACM MEM files
- ASSC IP, ASSC RAM, ASSC Wrappers, & ASSC MEM files

ADC Only

This configuration instantiates only the Analog Block model (as shown in the figure below). It omits the data processing, sequence controller, and the ADC management features. If you use this configuration you must completely manage the ADC and all related AB functionality.



System Diagram for ADC Only

This option disables sequencing (Sequencer dialog box); flag generation for peripherals (the flag grid for peripherals); data averaging (Digital Filtering Factor and Initial Averaging value); SMEV RAM access; SMEV RAM access; and the ability to specify the external resistor in the Current Monitor.

Without the ASSC, you must also manage some general ADC features. They are:

- ADC clock divider (derived from the system clock frequency)
- ADC resolution (resolution combo box on main screen)
- Acquisition time for peripherals (available on each peripheral)

The ADC only option generates ACM MEM files.

ASB Advanced Options – ASSC RAM

If you choose to enable user access to ASSC RAM in the ASB Advanced Options dialog box, then you can read the contents of the ASSC RAM.

When performing slot processing, the ASSC stores the raw ADC Result value and the digitally-scaled ADC value into this RAM. The address locations for these values are exported in the log file created by the Analog System Builder.

The ASSC stores its result on a per-slot basis. Thus, if your sequencer samples a single channel multiple times, then the samples for that channel will be in multiple locations of the ASSC RAM. However, each sample result could potentially be different since it is sampled at a different time.

The ASSC continually overwrites each slot location every time it processes a sequence slot.

Each data is stored as, a 12-bit value, so you must read two RAM locations to retrieve the value.

The exposed ASSC RAM ports are shown in the table below.

Port Name	Input/Output	Description
USER_ASSC_ADDR[8:0]	Input	**User RAM Address** - You can control these address signals and enable read access from the A-port of the 512x9 ASSC RAM. If unused, these signals must be tied to logic 0 or logic 1.
USER_ASSC_RD	Input	**User RAM Read Enable** - You can control the control signal and enable read access from the A-port of the 512x9 ASSC dual-port RAM (you must connect to the ASSC_RAM_DO_A[8:0] port for read data). If unused, this signal must be tied to logic 0. **Make sure that the ASSC_RAM_BUSY signal is inactive at logic 0 when this signal is first activated, otherwise, the data read from the A-port of the ASSC RAM will not be from the USER_ADDR[8:0] address.**
USER_ASSC_RAM_BUSY	Output	**ASSC RAM Busy** - This output signal indicates that either the Init/Config Soft IP block or the System Monitor Evaluation Phase State Machine Soft IP block is busy accessing the A-port of the ASSC RAM. This signal can be used by user logic outside the analog interface soft IP blocks or can be left unconnected if unused.
ASSC_RAM_WR_BUSY_B	Output	**ASSC Busy Writing** - This active-high signal is for user status monitoring and indicates that the ASSC block is busy writing to the B port of its dual-port RAM.
ASSC_RAM_DOUT	Input	Dout of Port A of the ASSC RAM

The ASSC memory content report looks like this:

```
*******************************************************************************
                        ASSC Memory Content Report
*******************************************************************************
Slot              Channel   Address      Bits                          Value
-------------------------------------------------------------------------------
   0                volt1
                                    3|   [08:00]|        Raw ADC Result [08:00]
                                    4|   [02:00]|        Raw ADC Result [11:09]
                                    5|   [08:00]|     Scaled ADC Result [08:00]
                                    6|   [08:00]|     Scaled ADC Result [17:09]
```

See Also

ASB Advanced Options

ASB Advanced Options - SMEV RAM

ASB Advanced Options - ADC results

ASB Advanced Options – SMEV RAM

If you choose to enable user access to SMEV RAM in the ASB Advanced Options dialog box, then you can read the contents of the SMEV RAM.

When processing channels the SMEV stores the digitally-filtered (i.e., averaged) ADC Result value into this RAM. The address locations for these values are exported in the log file created by Analog System Builder. Unlike the ASSC RAM, the SMEV stores its result on a per channel basis.

The SMEV will continually overwrite each channel location every time it processes a channel.

As with the ASSC RAM, the data is stored as a 12-bit value, so you must read two RAM locations to retrieve one value.

The exposed SMEV RAM ports are shown in the table below.

Port Name	Input/Output	Description
USER_EV_ADDR[8:0]	Input	**User RAM Address** - You can control these address signals and enable read access from the A-port of the 512x9 SMEV RAM. If unused, these signals must be tied to logic 0 or logic 1.
USER_EV_RD	Input	**User RAM Read Enable** - You can control the control signal and enable read access from the A-port of the 512x9 SMEV dual-port RAM (you must connect to the EV_RAM_DO_A[8:0] port for read data). If unused, this signal must be tied to logic 0. **Make sure that the USER_EV_RAM_BUSY signal is inactive at logic 0 while this signal is first activated, otherwise, the data read from the A-port of the SMEV RAM(s) will not be from the USER_EV_ADDR[EV_ASIZE-1:0] address.**
USER_EV_RAM_BUSY	Output	**SMEV RAM Busy** - This output signal indicates that either the Init/Config Soft IP block or the SMTR Soft IP block is busy accessing the A-port of the SMEV RAM. This signal can be used by user logic external to the analog interface soft IP blocks or can be left unconnected if unused.
EV_RAM_WR_BUSY_B	Output	**SMEV Busy Writing** - This active-high signal is for user status monitoring and indicates that the SMEV block is busy writing to the B port of its dual-port RAM.
SMEV_RAM_DOUT	Input	Dout of Port A of the SMEV RAM

The SMEV memory content report looks like this:

```
*******************************************************************************
                        SMEV Memory Content Report
*******************************************************************************
                  Channel    Address      Bits                          Value
-------------------------------------------------------------------------------
                    volt1
                                    75|    [08:00]|   Averaged ADC Result [08:00]
                                    76|    [02:00]|   Averaged ADC Result [11:09]
-------------------------------------------------------------------------------
```

See Also

ASB Advanced Options

ASB Advanced Options - ASSC RAM

ASB Advanced Options - ADC results

ASB Advanced Options – ADC Results

This option exposes the RESULT port from the AB. It represents the ADC result value that was currently sampled. The exposed port is described in the table below.

Port Name	Input/Output	Description
ADC_RESULT[11:0]	Output	**ADC Result** - These signals comprise the conversion result from the ADC. In 12-bit mode, the ADC uses all bits, in 10-bit mode, it uses bits 11:2, and in 8-bit mode, it uses bits 11:4; all unused bits are set to logic 0 values when in 10-bit or 8-bit mode. These inputs connect to the result_o[11:0] outputs from the ADC.

See Also

ASB Advanced Options

ASB Advanced Options - ASSC RAM

ASB Advanced Options - SMEV RAM

ASB – Calculating a Threshold

Scaling Factor is determined based upon the selected prescaler range. Refer to the table below for a list of prescaler range values.

The value of each LSB bit is determined by dividing VREF by 2^{12}. This calculation is always performed with 12-bit mode, the masking determines the resolution (see below).

Voltage Monitor

Calculated Threshold = (UserThreshold(V) * ScalingFactor * Value of each LSB)

Current Monitor

```
Calculated Threshold =
    (user threshold(A)*
    Resistor Value(Ohm)*
    Gain Applied by Current Monitor)*
    (Value of each LSB bit)
```

Gain = 10 for Current Monitor.

Then masking is performed; see below.

Temperature Monitor

The temperature value from the ADC is in degrees Kelvin. For example, using the internal VREF of 2.56 V and 10-bit resolution, each LSB of the ADC result = 1K. This is generalized for internal VREF to:

8-bit mode → 1LSB = 4K

10-bit mode → 1LSB = 1K

12-bit mode → 1LSB = .25K

For example:

8-bit mode → ADC LSB = 01001010b (74) = 74 * 4 = 296K

10-bit mode → ADC LSB = 0100101010b (298) = 298 * 1 = 298K

12-bit mode → ADC LSB = 000100101010 (298) = 298 * .25 = 74.5K

However, when an external VREF is used, then these ratios no longer apply. Thus a general formula for calculating this ratio is as follows:

@ 25C, 2.56 V

ADC_VALUE_AT_ROOM_TEMP = 748

KELVIN_VALUE_AT_ROOM_TEMP = 298.15

1. Get the value of each LSB in mV: (VREF / 4096) * 1000
2. ADC_VALUE_AT_ROOM_TEMP / (result from 1.)
3. KELVIN_VALUE_AT_ROOM_TEMP / (result from 2.)

This is then the ratio of Kelvin per LSB for any VREF in 12-bit resolution.

Thus,

Calculated Threshold = ((user Threshold(C) + 273.15) / KelvinRatioPerLSB)

Then masking is performed (see below).

Masking on Resolution

The calculated threshold is masked depending upon the resolution.

- 12-bit resolution – No bits are masked
- 10-bit resolution – Bits 0 & 1 are set to '0'
- 8-bit resolution – Bits [3:0] are set to '0'

Prescaler Range

The prescaler range is determined from the maximum input voltage field. See the table below to calculate your prescaler range.

Scaling Factor: Pad to ADC Input	LSB for 8-bit conversion (mV)	LSB for 10-bit conversion (mV)	LSB for 12-bit conversion (mV)	Full-Scale Voltage	Range Name (V)
0.15625	64	16	4	16.368	16
0.3125	32	8	2	8.184	8
0.625	16	4	1	4.092	4
1.25	8	2	0.5	2.046	2
2.5	4	1	0.25	1.023	1
5.0	2	0.5	0.125	0.5115	0.5
10.0	1	0.25	0.0625	0.25575	0.25
20.0	0.5	0.125	0.03125	0.127875	0.125

If the maximum input voltage is greater than the given range, it will select the higher range.

The corresponding ranges for negative polarity is the same.

Note: If the maximum input voltage is greater than 2.0 V and less than or equal to VREF, then no prescaler is used; in other words, the channel is a direct analog input.

ASB Channel Mapping

In SmartGen you must specify names for each peripheral you configure. For example, you can name your Voltage Monitor "MYVOLT" and it will be referred to throughout SmartGen (and in your netlists) as "MYVOLT".

However, the analog block (AB) names Analog Ports AV0, AV1, … , AC0, AC1, … AT0, AT1, etc. Also, each port is physically tied to a specific pin on the package. For example, AV0 is physical pin 99 on the package; AV1 is physical pin 103 on the package, etc.

The physical pin placement is communicated directly through the netlist. So, when Designer imports a netlist with an AB, it infers the pin placement from the netlist.

Thus, Smartgen has to map your logical channel into the physical channel on the AB.

Note: If you modify the die or package of a project the ASB automatically reverts all user-assigned pins to UNASSIGNED.

Mapping Requirements

The mapping rules are as follows:

- Voltage Monitors can be placed on AV, AC, or AT pads; Voltage Monitors placed on AT pads can only have prescalers in the 16 V and 4 V range. Refer to Appendix A for available ranges and range calculation.

- Current Monitors can be placed only on AC pads with the additional requirement that the adjacent AV pad is also available. This is due to the fact that a Current Monitor is an AC-AV pair.
- Temperature Monitors can be placed only on AT pads
- Gate Drivers can be placed only on AG pads
- Digital Inputs can be placed on AV, AC, or AT pads
- Internal channels do not have physical pin mappings, they reside purely on-chip.

The Fusion mapping resource requirements for the AFS600 are as follows:

- 10 AV
- 10 AC
- 10 AT
- 10 AG
- 1 Internal Voltage
- 1 Internal Temperature

The Fusion mapping resource requirements for the AFS250 are as follows:

- 6 AV
- 6 AC
- 6 AT
- 6 AG
- 1 Internal Voltage
- 1 Internal Temperature

Analog System Builder Peripherals

Current Monitor

The current monitor in Fusion measures current by measuring the differential voltage across a resistor between a pair of Voltage and Current Input channels. This peripheral requires two channels, one of type V and one of type C, and they must be on adjacent package pins.

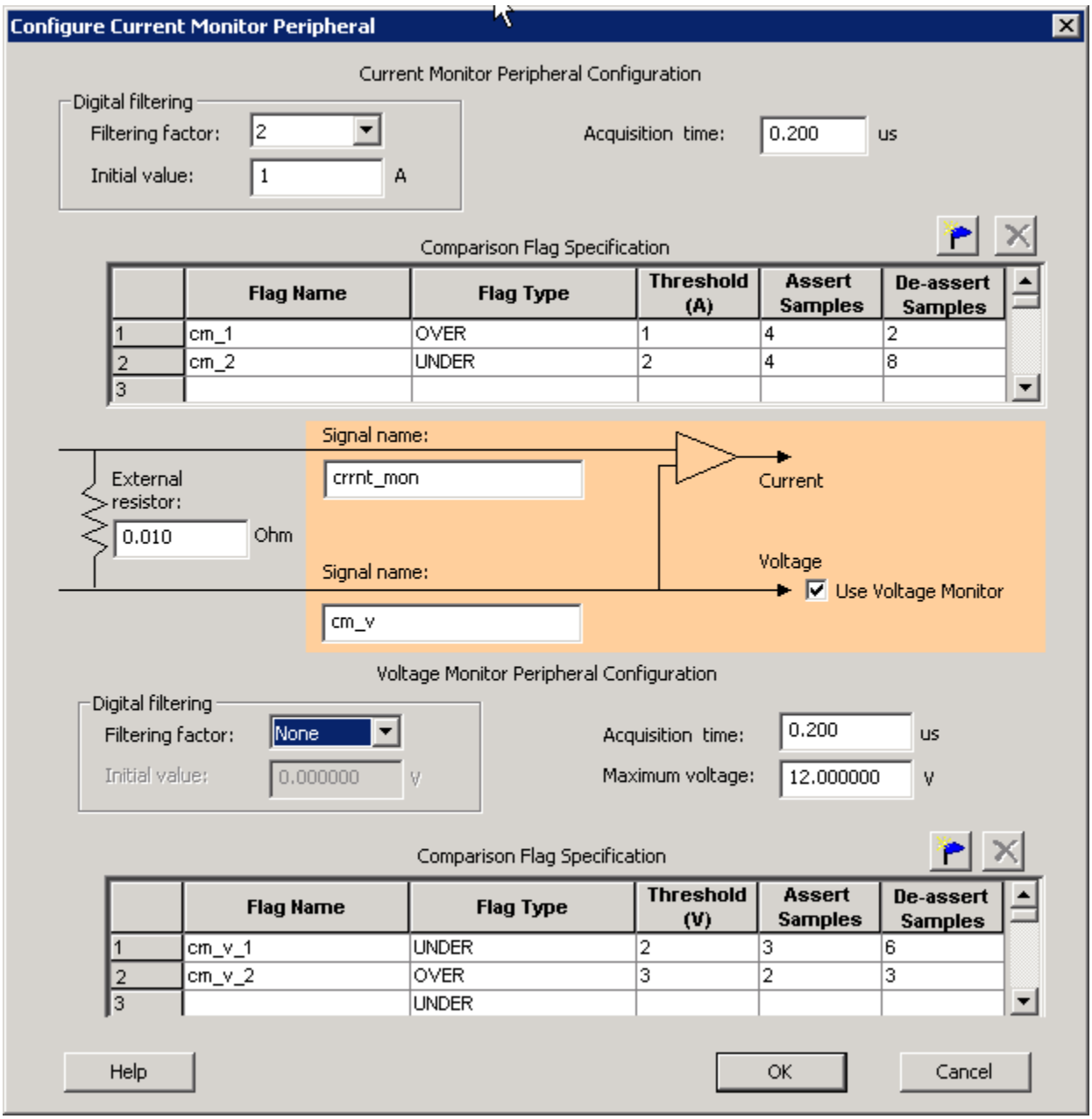
The differential voltage is multiplied by 10x before it is applied to the ADC; there is no pre-scaling on the differential voltage measurement. The difference in voltages must be less than the value of Vref, external or internal. You must choose an external resistor that satisfies this condition.

The differential amp gain in the current monitor is 10X. SmartGen assumes a series resistor because it is being used to measure current. The differential amplifier measures the potential/voltage drop (256 mV max if the reference is 2.56 V) across the resistor, which is proportional to the current flowing in the direction AV -> AC (I = (change in voltage)/resistance)).

The voltage channel in the pair can be used as a voltage monitor to measure the actual voltage that is connected to the Voltage channel.

See the Configuring Current, Voltage, and Temp peripherals for information on the digital filtering factor, Acquisition time, and Comparison Flag Specifications.

Controls unique to this peripheral are the External resistor and Maximum voltage.



Configure Current Monitor Peripheral Dialog Box

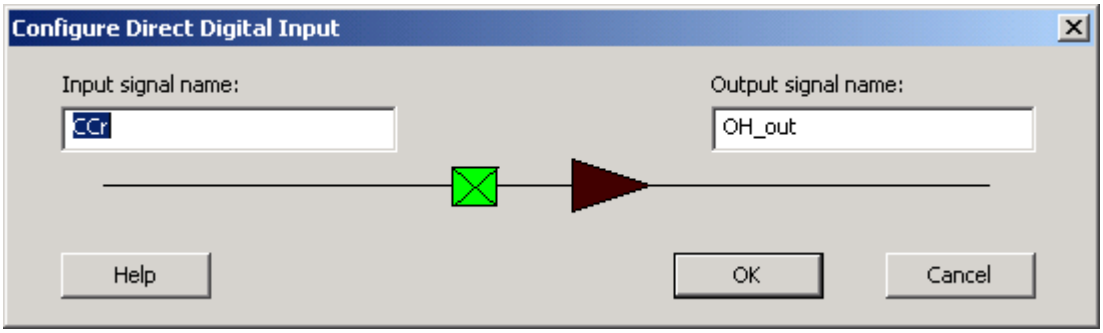
External Resistor - The value of the resistor that is connected across the Current-Voltage pair, external to the device. SmartGen uses this value to convert the thresholds into voltages.

Maximum voltage - The maximum anticipated voltage measured by this Voltage Monitor peripheral pad. The range is -12 V to +12 V (the voltage range is NOT bipolar). The ADC is capable of measuring a voltage range of 0 - Vref. For the Internal voltage

reference, this value is 2.56 V. SmartGen automatically configures the prescaler in the AB Analog Block for this peripheral to maximize the available voltage range. SmartGen also post-scales the digital result of ADC conversion so that it returns a result in your specified range.

Direct Digital Input

You can use the Direct Digital Input to configure the unused analog channels or peripherals as digital inputs. Specify the input and output signal name and add the inputs to your sampling sequence. The digital input can also be up to 12 V. but are limited to a frequency of 10 MHz.

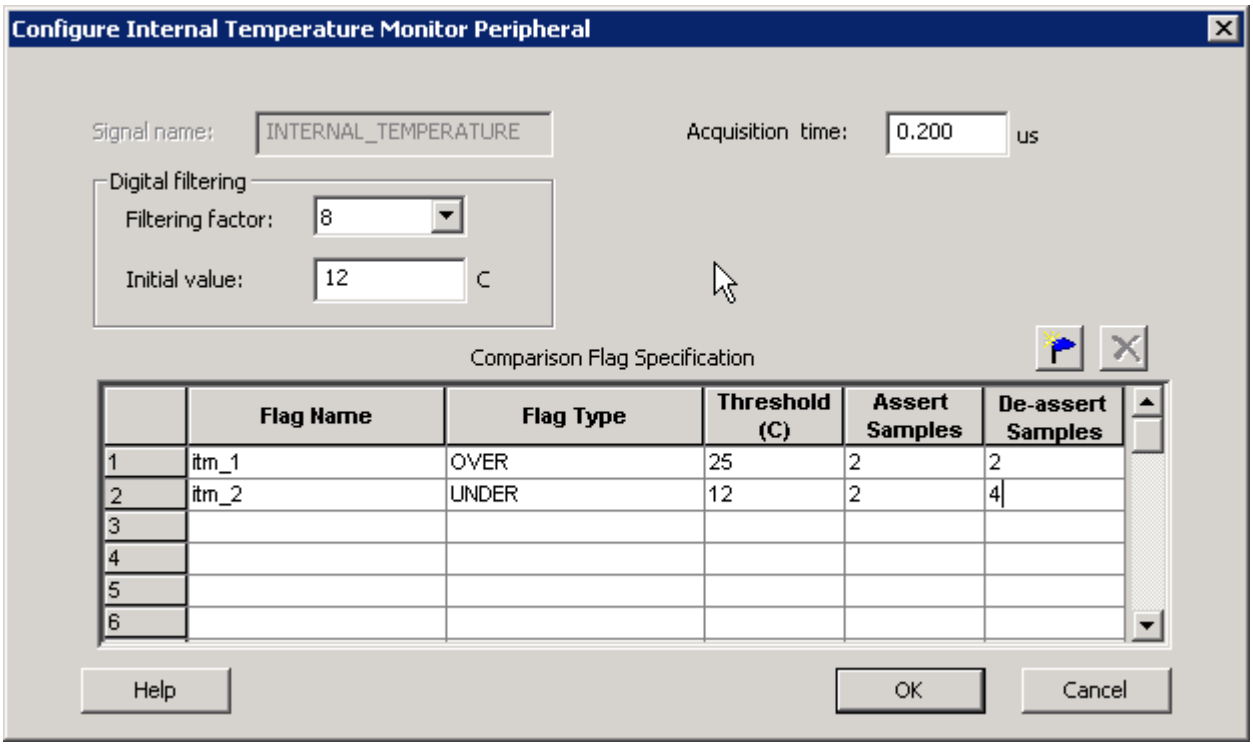


Configure Direct Digital Input Dialog Box

Internal Temperature Monitor

The internal temperature channel measures the internal die temperature (as shown in the figure below). The peripheral is mapped to logical channel 31 and does not have an associated physical pin.

See the Configuring Current, Voltage, and Temp Peripherals section for information on the digital filtering factor, Acquisition time, and Comparison Flag Specifications.

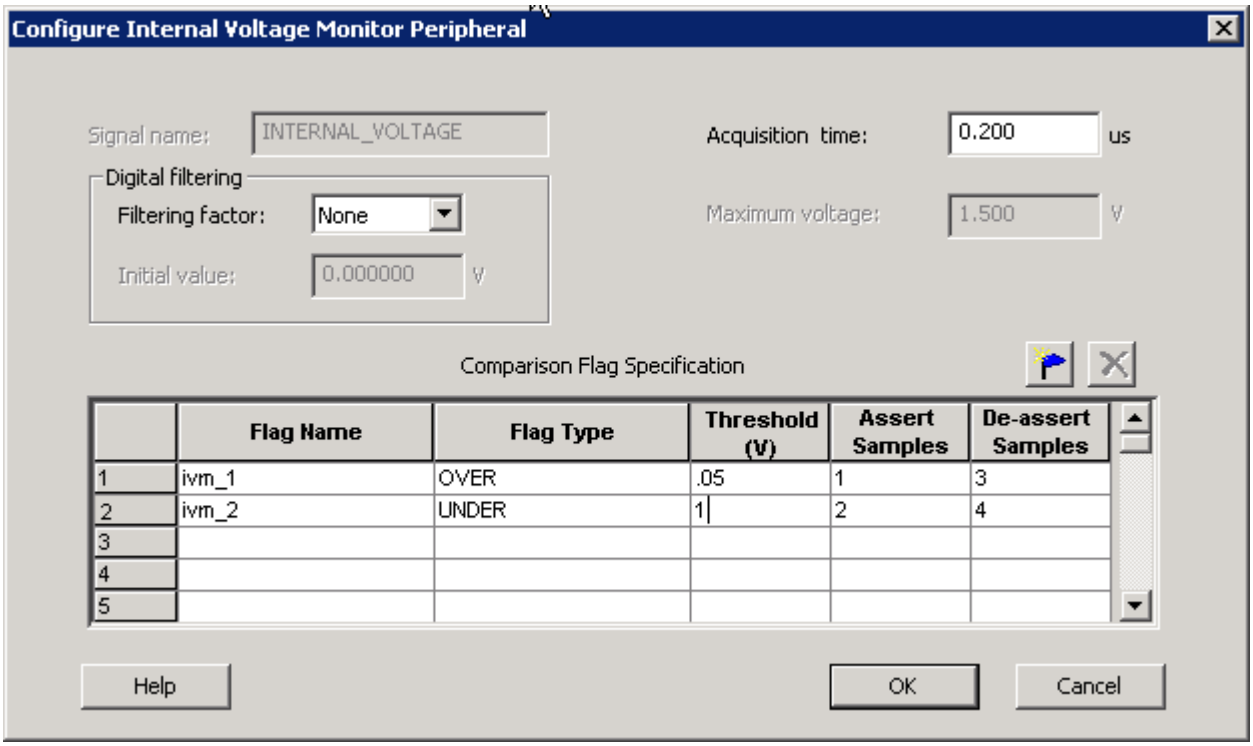


Internal Temperature Monitor Dialog Box

Internal Voltage Monitor

The internal voltage channel measures the internal 1.5 V power supply. The Maximum Input Voltage for this peripheral is fixed (as shown in the figure below). The peripheral is mapped to logic channel 0 and does not have an associated physical pin.

See the Configuring Current, Voltage, and Temp Peripherals section for information on the Digital filtering factor, Acquisition time, and Comparison Flag Specifications.

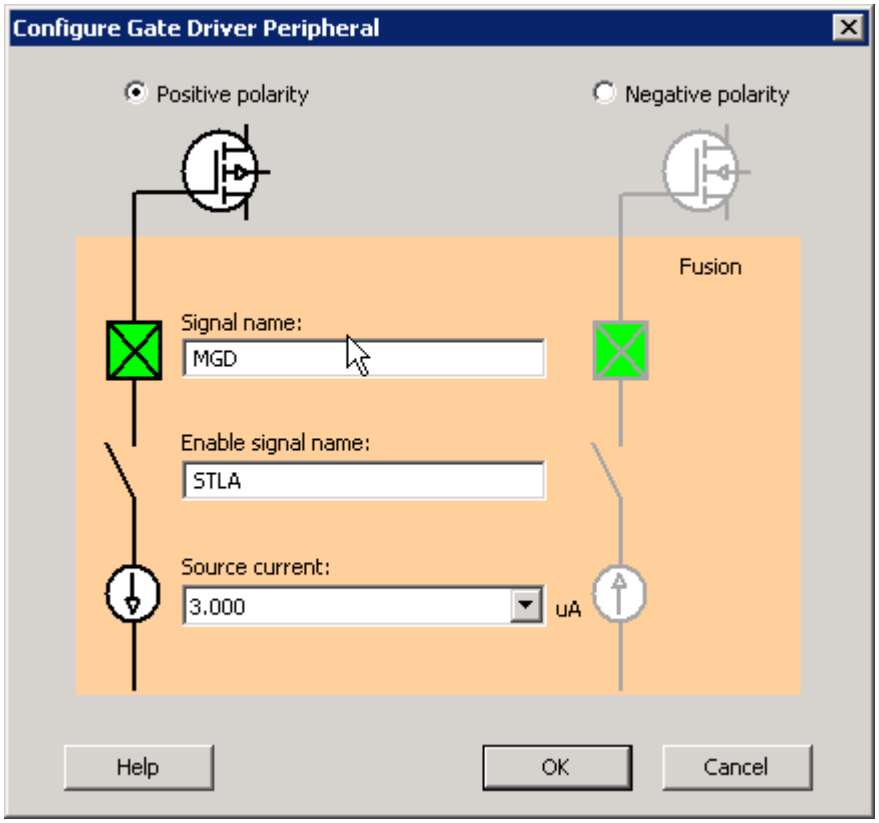


Internal Voltage Monitor Dialog Box

Gate Driver

The Gate Driver is the analog output coming from the analog system. You can use it to drive the gate of an external MOSFET on or off. The Gate Driver is designed to work with external MOSFETs as a configurable current sink or source.

The figure below shows the Gate Driver dialog box.



Output Gate Driver Dialog Box

Gate Driver Polarity

Indicates the type of MOSFET controlled by the gate driver (NMOS or PMOS).

PMOS controls positive voltage supply and NMOS controls negative voltage supply. The AG pad can work with either P-Channel (pulling down to ground) or N-Channel (pulling up to ground) devices.

Signal Name

The name of the signal assigned to this gate driver. It appears as the final port name in the generated system.

Enable Name

The name of the signal that enables the gate driver

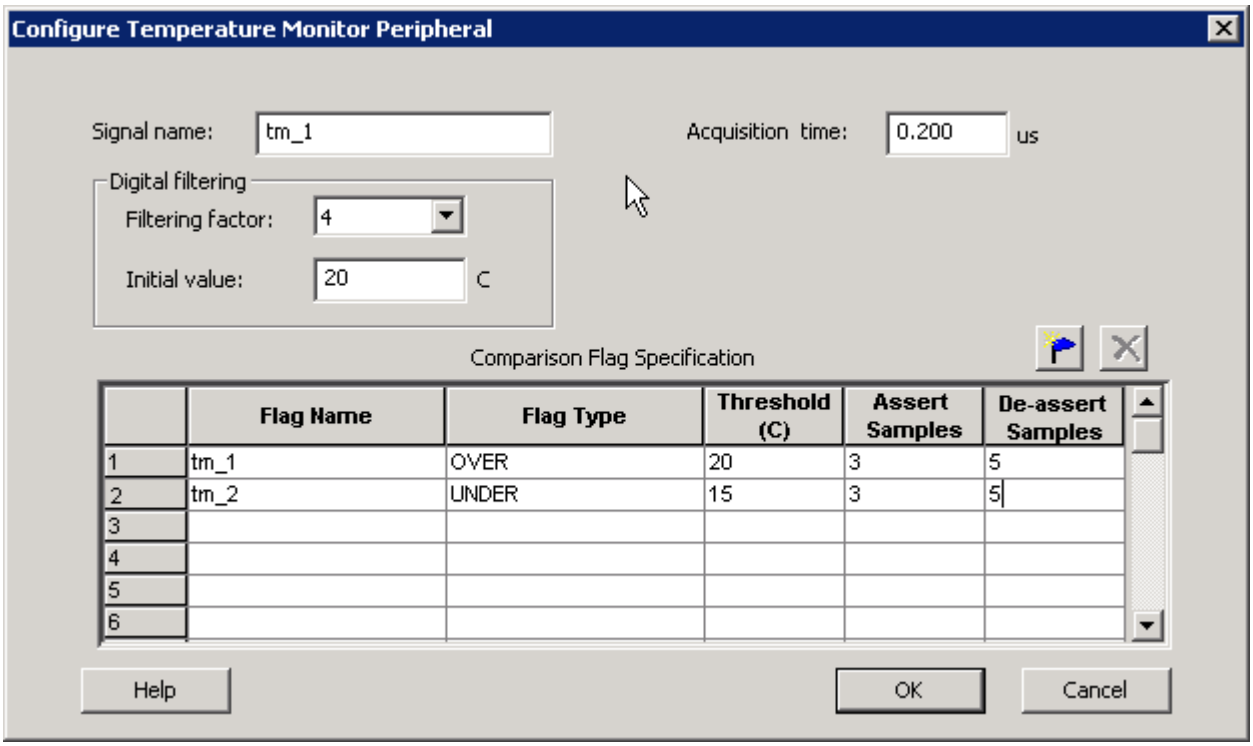
Source Current

The amount of current the device is expected to source. The Gate Driver supports four selectable current drive levels: 1 µA, 3 µA, 10 µA, and 30 µA.

Temperature Monitor

When used in conjunction with an external bipolar transistor, the Temperature Monitor is designed to measure temperature of an external location. A temperature monitor circuit can be very sensitive to system noise.

See the Configuring Current, Voltage, and Temp peripherals section for information on the Digital filtering factor, Acquisition time, and Comparison Flag Specifications.



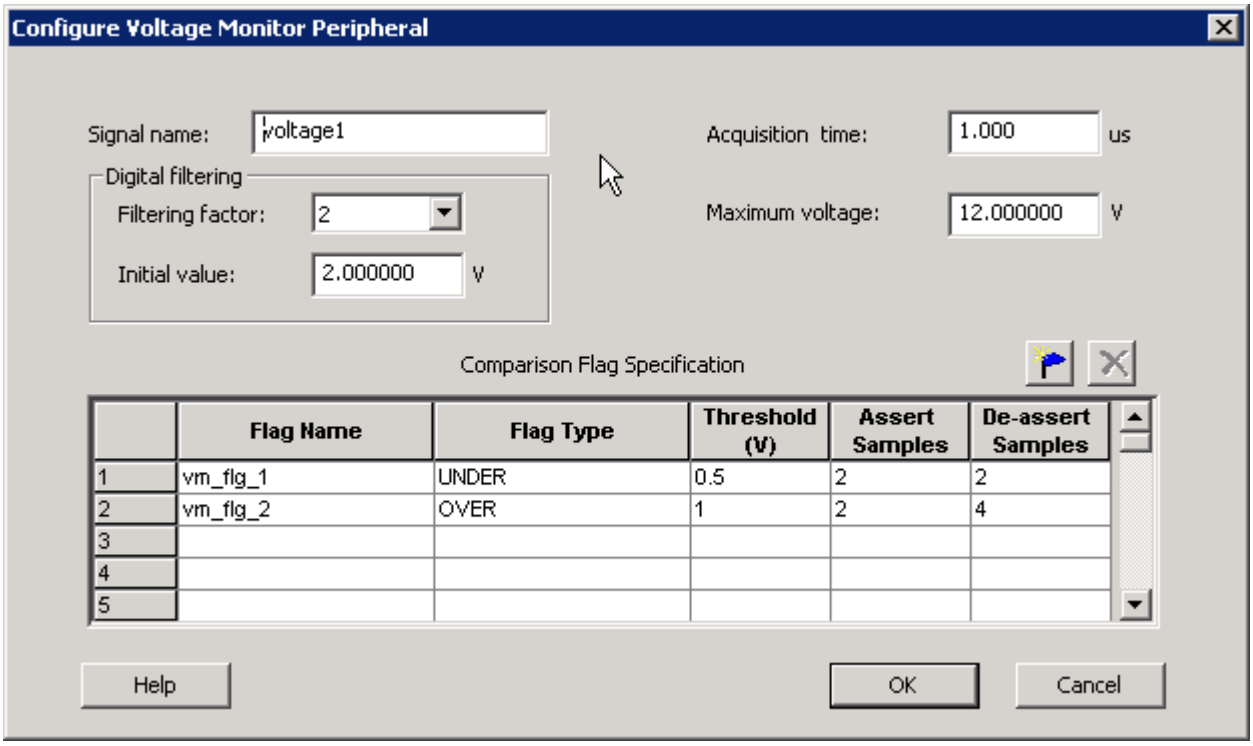
Configure Temperature Monitor Peripheral Dialog Box

Voltage Monitor

The Voltage Monitor contains a 2-channel analog multiplexer that allows an incoming analog signal to be routed directly to the analog-to-digital converter, or allows the signal to be routed to a prescaler circuit before being sent to the ADC.

See the Configuring Current, Voltage, and Temp peripherals section for information on the Digital filtering factor, Acquisition time, and Comparison Flag Specifications.

The only control unique to this peripheral is Maximum voltage (described below).



Configure Voltage Monitor Dialog Box

Maximum Voltage

The maximum anticipated voltage measured by this Voltage Monitor peripheral pad. The range is -12 V to +12 V (the voltage range is NOT bipolar). The ADC is capable of measuring a voltage range of 0 - Vref. For the Internal voltage reference, this value is 2.56 V. SmartGen automatically configures the prescaler in the AB Analog Block for this peripheral to maximize the available voltage range. SmartGen also post-scales the digital result of ADC conversion so that it returns a result in your specified range.

Also, an External VREF, affects the prescaler selection. The prescaler range selection is determined by

VoltValue = (your maximum input voltage / 2.56) * VREF

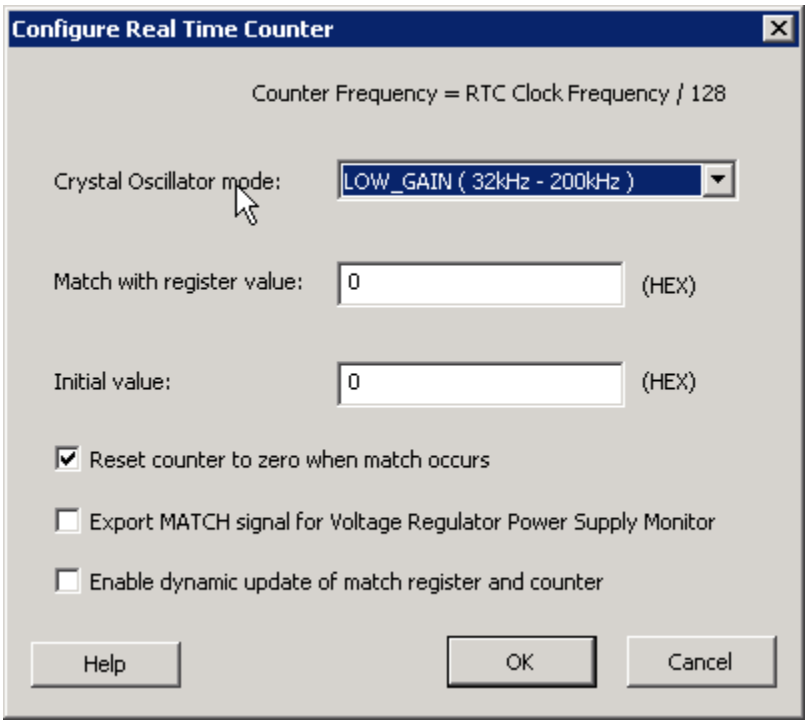
Real-Time Counter

The on-chip crystal oscillator circuit works with an off-chip crystal to generate a high-precision clock. It has an accuracy of 100 ppm (0.01%) and is capable of providing system clocks for Fusion peripherals and the other system clock networks, both on- and off-chip. When combined with the on-chip CCC/PLL blocks, a wide range of clock frequencies can be created to support various design requirements.

The Real-Time Counter inside the Analog Block has the following features:

- The MATCH signal on the output of the system asserts when the value in the counter matches the value specified in the match register. Also, there is an optional output RTCPSMMATCH that is triggered on match. The RTCPSMMATCH signal must be connected to the RTCPSM macro so that the Voltage Regulator activates when the Match signal is asserted.
- If you use the RTC, RTCCLK must be driven by the External Crystal Oscillator driving the Fusion device and the mode of the Crystal oscillator must be controlled by the RTC.

The figure below shows the Real Time Counter dialog box.



Real Time Counter Dialog Box

Crystal Oscillator Mode

RC network - Oscillator is configured to work with an external resistor-capacitor network.

LOW_GAIN / MEDIUM_GAIN / HIGH_GAIN - Oscillator is configured to support an external crystal or ceramic resonator. The difference is the supported crystal frequency or resonator, as shown in the table below.

Mode	Recommended Capacitor	Frequency Range (MHz)
LOW_GAIN	100 pF	0.032 to 0.20
MEDIUM_GAIN	100 pF	0.21 to 2.0
HIGH_GAIN	15 pF	2.1 to 20.0

Match with Register Value

The MATCH signal asserts when the counter is equal to this register value (40-bit binary or 10-bit hex).

Initial Value

You may specify a different counter start value. The default is zero. This also is a 40 binary or a 10 bit hex value.

Reset count to zero when match occurs- Resets the counter once the match occurs. This can be disabled for an application that measures elapsed time.

Export Match signal for Voltage Regulator Power Supply Monitor - Asserts the RTCPSMMATCH to activate the Voltage Regulator Power Supply Monitor (VRPSM).

Enable dynamic update of match register and counter - This option provides access to the ACM Address and Data signals to update the values of counter and match register. Exercise caution: The ACM address bus is shared by the Analog System and the Analog system configuration could be overwritten if it accesses an incorrect address.

Configuring Current, Temperature, and Voltage Peripherals

The Current, Temperature, and Voltage peripherals are all configured the same way. Also, the effects of averaging are the same for all peripherals in the Analog System Builder.

Minor variations, such as Maximum Voltage in the Voltage monitor, are explained in the help topic for that peripheral.

Signal name is the name of the signal as you want it to appear in the main Analog System Builder dialog box.

Digital filtering

Once the ADC finishes converting the Analog Signal to a digital value, it filters (averages) the resulting digital output. Digital filtering is a single-pole low-pass filter built in soft gates; you can use it to improve the signal-to-noise ratio. If the ADC input data is very erratic, the filtering will smooth out the input and reduce the noise.

The filtered value is calculated using the following equation:

$Filtering\_result_n = filtering\_result_{n\_1} + (ADC\_Result_n / filtering\_factor) – (filtering\_result_{n\_1} / filtering\_factor)$

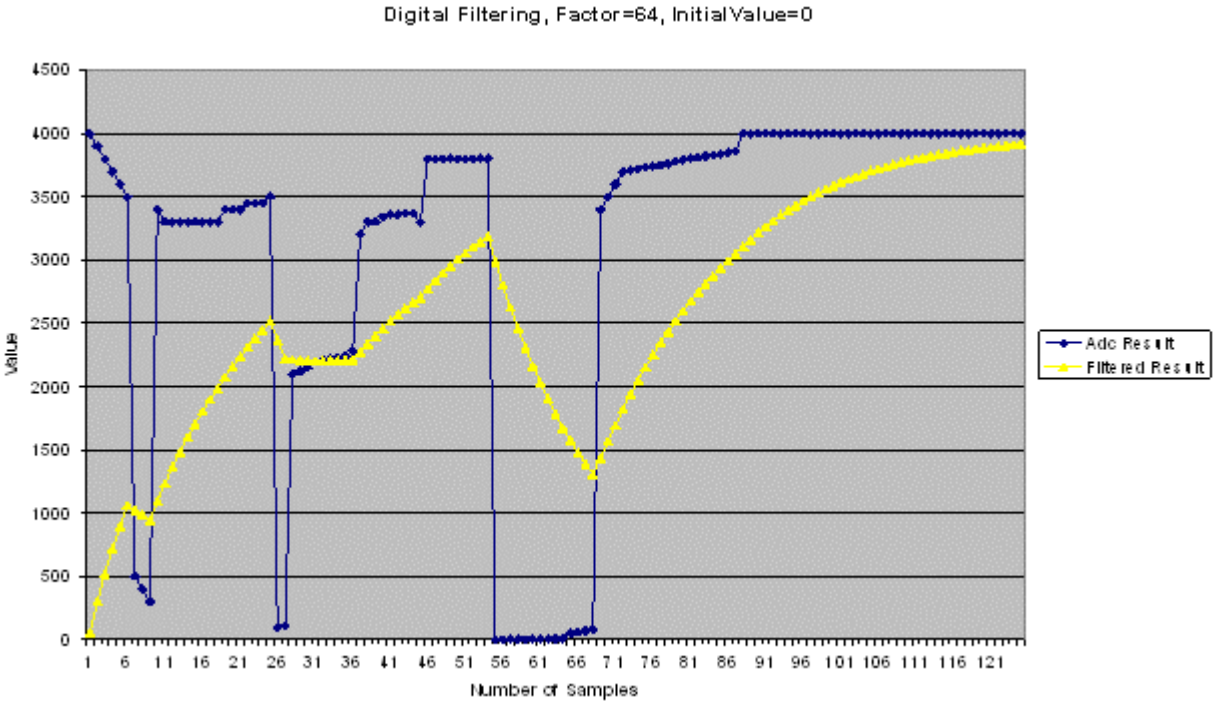
If the Digital filtering factor is set to 1 it is ignored.

Initial Filtering Value - The initial filtering value enables you to specify the starting value for the averaging function (Filtering Result[0]). This enables you to 'seed' your filtering function so that there are no erroneous values produced during the beginning of operation.

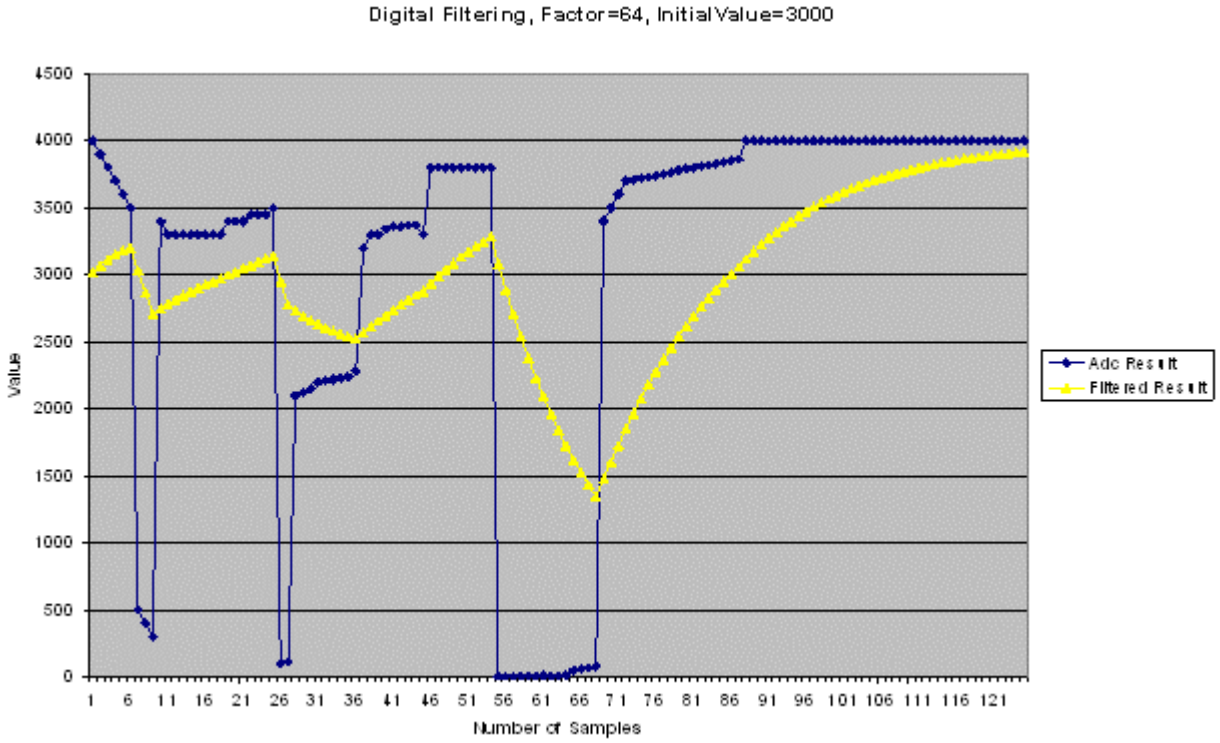
If you do not use an initial filtering value, the filtering function always starts with FilteringResult[0] = 0, thereby skewing the results towards 0 during the first range of samples.

Range - The Initial value range for the digital filter is identical to the threshold range for the peripheral.

The system instantiates the logic required to perform averaging as soon as there is at least one channel in the system that requires averaging. There is no extra logic penalty for averaging the other channels of the system. See the figures below for a graphical representation of the effect of digital filtering on a signal.



Effect of Averaging on Voltage, Initial Digital Filtering Value = 0



Effect of Averaging on Voltage, Initial Digital Filtering Value = 3000

Acquisition Time

The required settling/sampling time for this channel. It is the amount of time the Sample and Hold circuit in the ADC charges the capacitor with the input analog signal.

Comparison Flag Specification

Once the software calculates the average, you can further process the result by comparing the result to a given threshold and deciding under what conditions to assert the comparison flags.

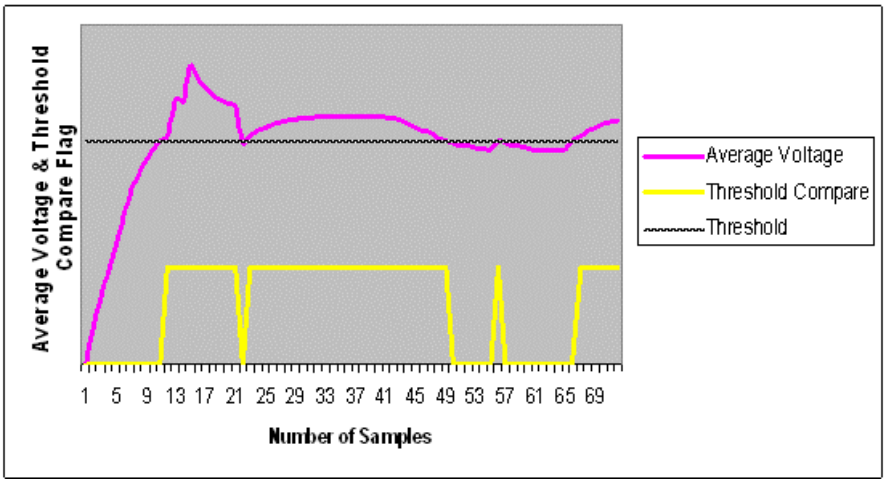
Select a flag and click the **Delete** button to delete it.

Click the **Add Flag** button to add more flags to the system.

Flag Name - This is the name of the flag. This will appear as the output port name in the final output. It will be prefixed with the signal name with which it is associated, to group the input and outputs together.

Flag Type - You can choose to assert the flag when the signal is either under a given threshold or over a given threshold.

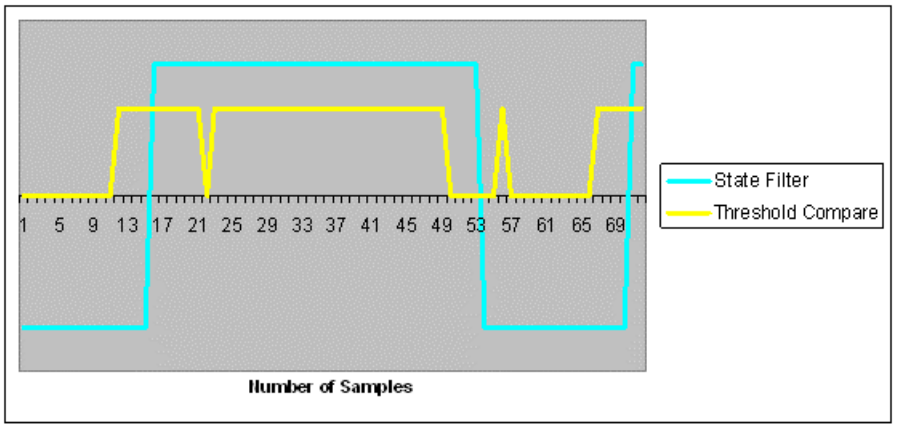
Threshold - Threshold value. The figure below shows the effect of the threshold on a given signal.



Threshold Comparison

Assert Samples - The number of consecutive samples on this channel that reach or exceed the threshold for the flag to assert. This can be a glitch removal feature. If it is set to 1, the final flag is identical to the comparison result.

For example, if your Assert Sample value is 3.0 V, the channel must reach or exceed 3.0 V five times in a row on this channel for the flag to assert. If your voltage values are less than 3.0 V, the flag will not assert. The figure below shows the effect of glitch removal on a given signal.



Glitch Removal

De-assert Samples - The number of consecutive samples of this channel that are not above the threshold required for the flag to de-assert once it has been asserted. This is a glitch removal feature. If this value is set to 1, the final flag is identical to the comparison result.

For example, if your de-assert Sample value is 3.3 V, you must have 10 consecutive samples that go below 3.3 V in order for the flag to de-assert

Acquisition Time

Acquisition Time is the time that it takes for the ADC to acquire and convert an analog signal to a digital value. The conversion time is a sum of:

(Sample Time) + (Calibration Time) + (Charge Distribution Time) + (Synchronization Time)

The Sample Time specifies how long an analog input signal must be sampled to obtain a proper value. The sample time is a calculated by:

(2 + sample time control) * ADC Clock Period

And the ADC Clock Period is calculated by:

(4 * (1 + clock divider setting)) / System Clock Period

The ADC Clock Period has a maximum possible frequency of 10Mhz.

Based upon the sample time requirement specified by the user, SmartGen automatically computes values for the sample time control (STC), clock divider setting (TVC), and ADC Clock Period.

The goal of SmartGen is to meet the minimum Sample Time requirement with the highest possible ADC Clock frequency, which implies a low TVC value and high STC value.

Sampling Rate in Analog System Builder

For ADC's that perform one sample per conversion, the throughput rate is also referred to as the sampling rate – this is the case for Fusion. Sampling Rate or frequency is the rate at which the ADC acquires or samples the analog input and converts it to digital data. It is specified as samples per second (S/s) or Hertz (Hz).

The sampling rate is typically the inverse of the ADC Conversion Time. For example, an ADC that takes 10 microseconds to acquire and convert an analog signal to a digital value will be able to generate about 100,000 samples per second. In the case where only a single channel is being sampled, the channel's sampling rate is equal to the total system sampling rate.

However, in the case where the sampling sequence contains multiple channels and where a channel may be sampled multiple times in relation to another channel, the sampling rate computation becomes more involved.

The following sections describe the sampling rate computation for SmartGen. SmartGen enables you to specify the minimum sampling time and minimum sampling rate. SmartGen then reports the total system sampling rate and the actual sampling rate of each channel.

General Formula for Sampling Rate

The general formula for sampling rate is as follows:

Total Sampling Rate = Total # of Samples / Total Conversion Time of all Samples

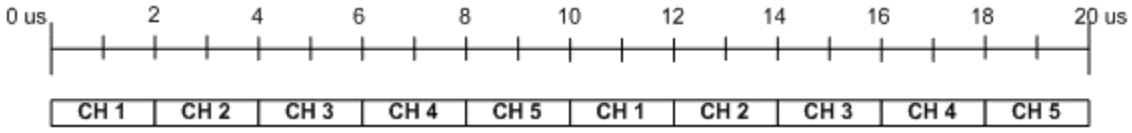
Channel Sampling Rate =
((Total Number of Samples for Channel) /

(Total Number of all Samples)) *
(Total Sampling Rate)

The method for arriving at this formula is explained in the examples below.

Example: Equal Weight and Equal Conversion Time

All Channels have a conversion time of 2 µs, as shown in the figure below.



Equal Weight and Equal Conversion Time

In this case we have 10 samples which take a total of 20 µs.
Thus, our total system sampling rate is: 10 / 20 µs = 500 kS / s

Channel1 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

Channel2 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

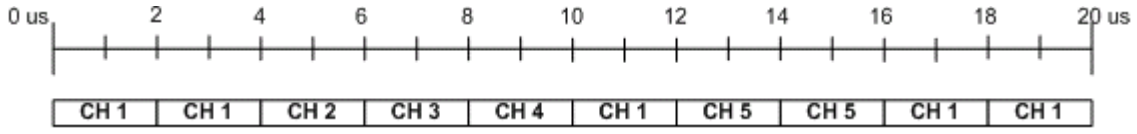
Channel3 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

Channel4 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

Channel5 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

Example: Unequal Weight and Equal Conversion Time

In this example, the channels are not equally weighted in the sampling sequence,as shown in the figure below.



Unequal Weight and Equal Conversion Time

In this case we have 10 samples that take a total of 20 µs, giving us a total system sampling rate of 500 kS/s (as above). However, the individual channel sampling rates are different.

Channel1 Sampling Rate: 5 µs / 10 µs = .5 * 500 kS /s = 250 kS /s

Channel2 Sampling Rate: 1 µs / 10 µs = .1 * 500 kS /s = 50 kS /s

Channel3 Sampling Rate: 1 µs / 10 µs = .1 * 500 kS /s = 50 kS /s

Channel4 Sampling Rate: 1 µs / 10 µs = .1 * 500 kS /s = 50 kS /s

Channel5 Sampling Rate: 2 µs / 10 µs = .20 * 500 kS /s = 100 kS /s

Example: Unequal Weight and Unequal Conversion Time

In this example, channels have different conversion times and are not equally weighted in the sampling sequence, as shown in the figure below.

Unequal Weight and Unequal Conversion Time

In this case we have 12 samples in 20 µs, giving us a total system sampling rate of 600 kS/s.

Channel1 Sampling Rate: 7 / 12 = .583 * 600 kS /s = 349 kS /s

Channel2 Sampling Rate: 2 / 12 = .166 * 600 kS /s = 99.6 kS /s

Channel3 Sampling Rate: 1 / 12 = .083 * 600 kS /s = 49.8 kS /s

Channel4 Sampling Rate: 1 / 12 = .083 * 600 kS /s = 49.8 kS /s

Channel5 Sampling Rate: 1 / 12 = .083 * 600 kS /s = 49.8 kS /s

Wait States and the Analog System Controller

The Analog System Controller automatically inserts wait states to prevent collisions during processing. The wait states are inserted to ensure this condition:

Current conversion time + ASSC processing time > = previous sample's (SMEV + SMTR) processing time

Thus a wait state is calculated as:

(Current conversion time + ASSC processing time) – previous sample's (SMEV + SMTR) processing time

This extra time is inserted into the sample rate calculation, thereby decreasing the total sample rate. Furthermore, when wait states are inserted, the calculated sample rate is only an approximation.

Sequencing in Fusion

The ADC has a strobe signal per temperature or current channel that must be asserted to initiate the conversion process. This strobe signal has strict timing pulse-width requirements; the strobe signal must stay on and then off for certain periods of time.

The requirement is that a strobe signal must stay low for a minimum of 5 µs after a high-to-low transition.

Furthermore, there is a high time requirement which is:

- 5 µs high for current strobe
- 10 µs high for temperature strobe

The ASSC generates these strobe signals; it has no concept of elapsed time, past samples, or future samples. A strobe pulse is activated for the entire ASSC processing time for a particular channel, and deactivated as soon as it moves to another channel. Thus, this requirement must be accounted for by you and/or SmartGen during the sequencing.

We must satisfy the following conditions to meet the requirement:

- A channel that requires a strobe cannot be sampled again until 5 µs has elapsed
- A channel that requires a strobe must have a conversion time + ASSC processing time >= 5 µs

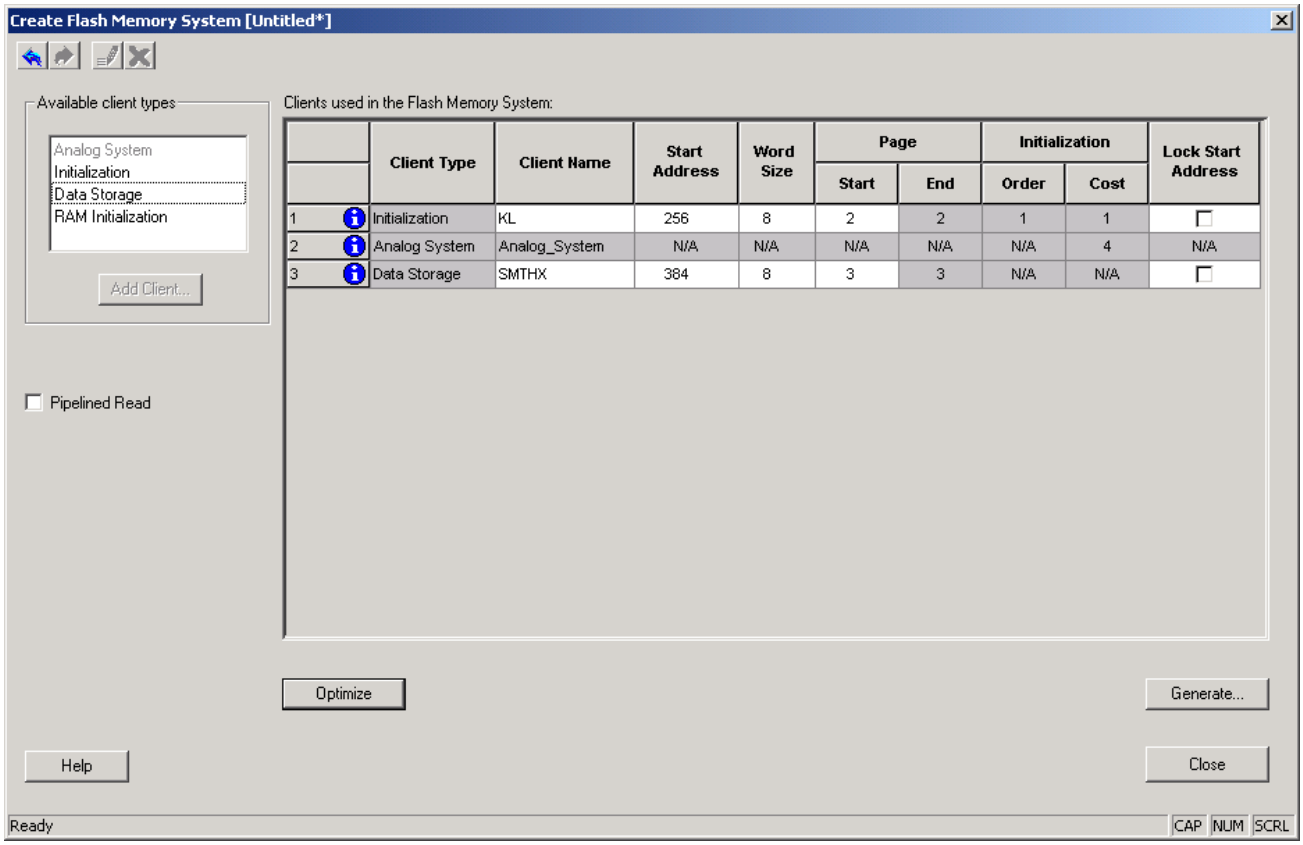
The second condition is automatically handled because you enter a MINIMUM required sampling time. SmartGen can then ensure a minimum 5 µs conversion time for temperature and current channels by adjusting the STC and ADC clock periods.

Flash Memory System Builder

Welcome to the Flash Memory System Builder

The Flash Memory System Builder in SmartGen enables you to configure the entire flash memory block. You can:

- Add analog system initialization data
- Add initialization clients
- Add Fusion RAM clients that require initialization
- Partition non-volatile memory for data access
- Specify the sizes of the partitions
- Specify the memory contents for partitions



Flash Memory Block Builder

The Flash Memory Block Builder supports the following clients:

- Analog System
- Initialization
- Data Storage
- Ram Initialization

Each client spans a minimum of one page (128 bytes) and can go up to 2048 pages, based on the number of free pages available. The analog system itself does not take any of the regular pages; it is stored entirely in the reserved pages.

The System Grid in the GUI provides an overview of the system.

Client Type - The type of the client that is added to the system.

Client Name - The name of the client. It must be unique across the system.

Start Address - The decimal or HEX address at which the client starts. It must be on a page boundary. Clients cannot have overlapping start addresses.

Word Size -

Page Start - This is the page on which the start address begins or ends. You can modify either the start address or the start page.

Page End- This is computed based on the word size and the number of words in the client.

Initialization Order - The order in which the clients' required initialization is written with the data from the Flash Memory System. If an analog system is present, it will be initialized first. If any clients must save their data to the Flash Memory System, the save order is the same as the Initialization Order.

Initialization Cost - The number of Initialization Enables used by a given client. An Analog System client has an Init Cost of 4. A RAM client has an Init Cost equal to the number of RAM blocks. A regular initialization client has an Init Cost of 1. The data storage client has no Init Cost.

The total init cost must be less than or equal to 64.

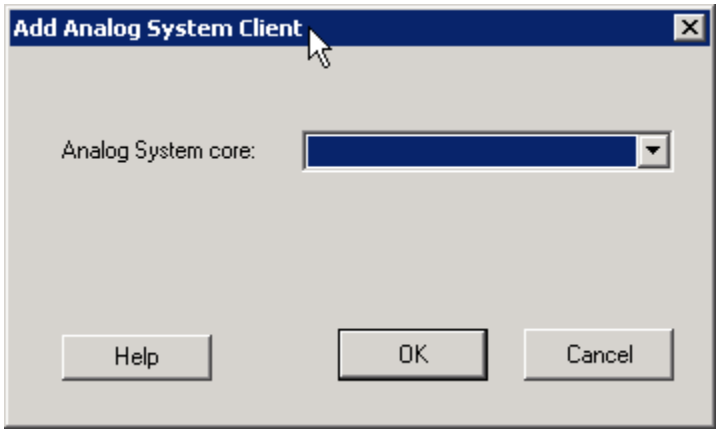
Lock Start Address - You can specify this option if you do not want the system to change your start address for any reason.

Optimize - Click this button to resolve the conflicts on overlapping base addresses for clients. This operation will not modify the base addresses for any clients that have their base addresses locked. It also de-fragments the memory.

Analog System Client

You can load the configuration file generated by the Analog System Builder into the Flash Memory System Builder. Once loaded, all the analog system components can be initialized by the Flash Memory System at start up.

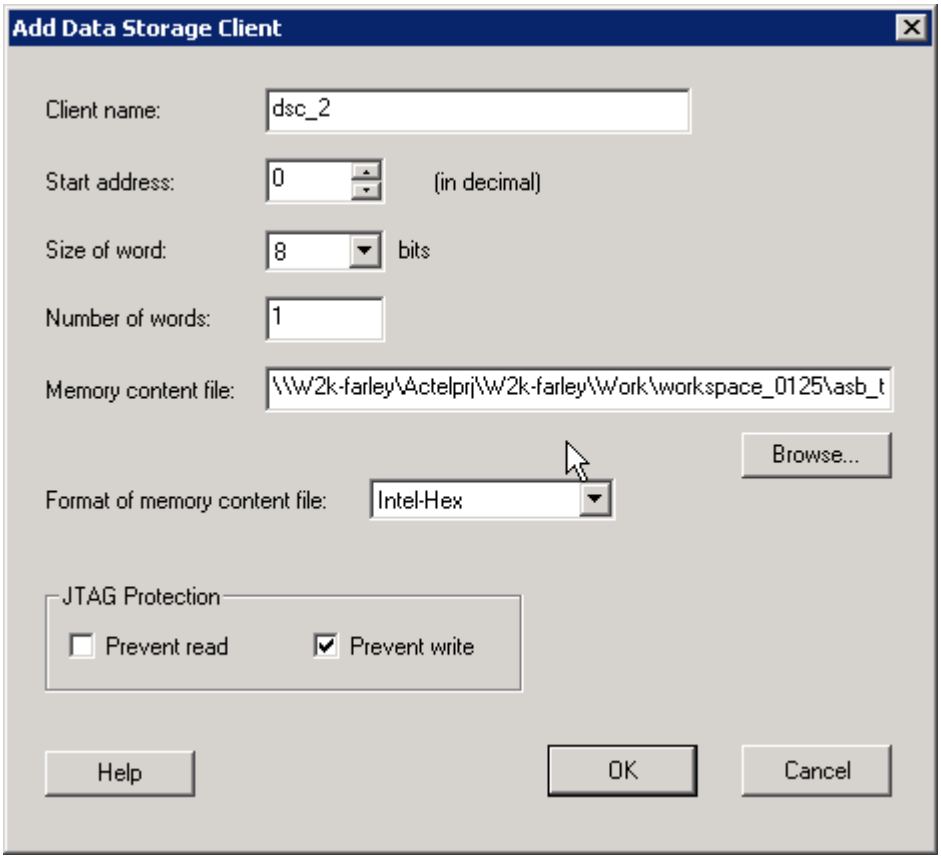
The Add Analog System Client opens with a blank field for the Configuration file. The Modify Analog System Client opens with the field filled in. Click the dropdown menu and select your Analog System core from the list.



Add Analog System Client Dialog Box

Data Storage Client

The Data Storage Client enables you to create a partition in the Flash Memory System and specify the memory content for that partition. You can access the partition directly via the Flash Memory System busses. The Add Data Storage Client dialog box opens with blank fields; the Modify Data Storage Client displays any values you have already set.



Add Data Storage Client Dialog Box

Client Name - Name of the client; the value you enter is attached before the select and enable names to group all the control signals for that client.

Start Address (in decimal) - Sets the starting address for the Data Storage Client.

Size of Word- Word size, in bits, of the initialized client; can be either 8, 16 or 32.

Number of Words - Can be anywhere from 1-262144 for 8 bit words, 1–131072 for 16-bit words, and 1-65536 for 32-bit words.

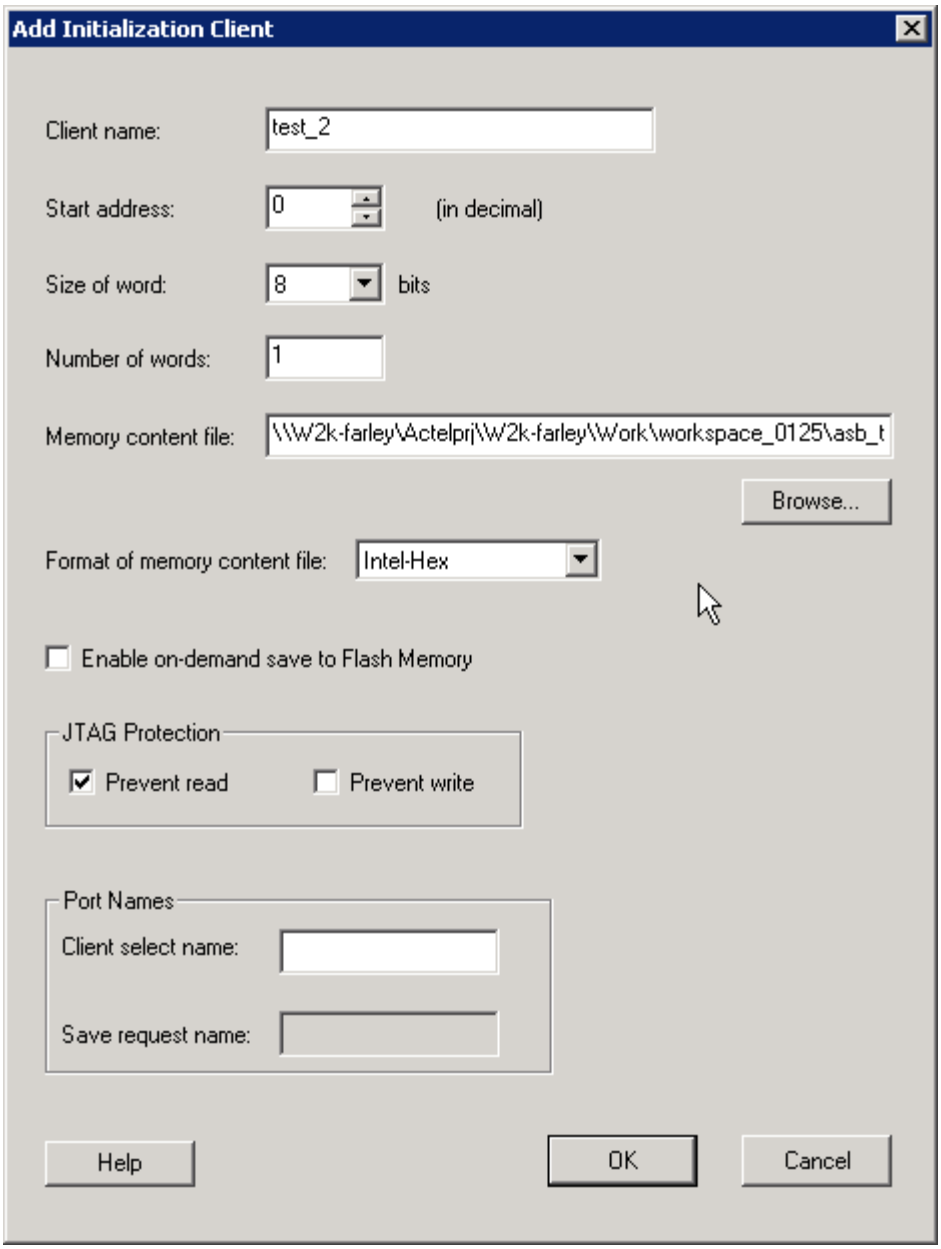
Memory Content files - The content for the Flash Memory Block to initialize this client. Must be specified in one of the supported memory formats.

Format of memory content file - Set your memory file format for the Flash Memory System Builder.

JTAG Protection - Protect your JTAG from read and write with these options.

Initialization Client

The Flash Memory System Builder initializes all the clients with the data stored in the Flash Memory when the system is powered-up. You must raise the power signal to High to power up the system. You can use the Initialization Client to set initial values of the RAM/FIFO (such as a table or list of filter coefficients, MAC addresses, etc.) and ROM emulation.



Add Initialization Client Dialog Box

Client Name - Name of the client; the value you enter is attached before the select and enable names to group all the control signals for that client.

Start address - Starting address for the initialization client.

Size of word- The word size in bits (8 or 9) of the initialized client. With a 9-bit word size, each word takes 2 bytes of Flash Memory System data.

Number of Words - May be anywhere from 1-262144 for 8-bit words or 1-131072 for 9-bit words.

Format of memory content file - The content for the Flash Memory System Builder to initialize this client; specified in one of the supported memory formats.

Enable on demand save to Flash Memory- Enables the content of Flash Memory System to be stored in the Flash Memory. For timing diagrams and signals operations, refer to Save data to FMS in the SmartGen Cores Reference Guide.

JTAG Protection - Protect your JTAG from read and write with these options.

Client Select and Save Request port names - Port names for the client chip select for initialization and save request for save-back. The port names are prefixed with the client name to group control signals.

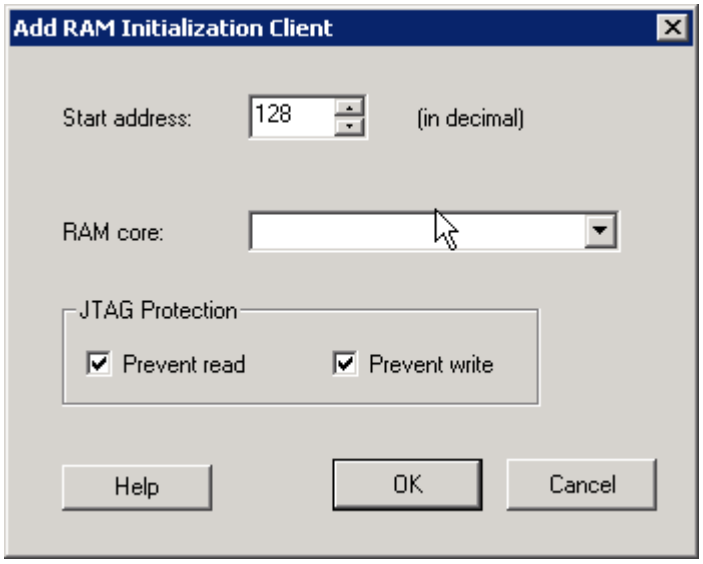
RAM Initialization Client

The RAM Initialization client is a special type of Initialization client. When you generate a RAM with Initialization from SmartGen for Fusion, the data for the RAM can be loaded into the Flash Memory System Builder such that at power-up the data is loaded into the RAM from the Flash memory.

The difference between this and the regular Initialization Client is that the cascading of multiple RAM blocks is handled automatically. The RAM Initialization client takes as many initialization clients as there are RAM blocks in the cascaded RAM.

JTAG Protection protects your JTAG chain from reading and/or writing if you select the checkboxes.

You can also specify a start address and lock the start address if you wish to always use this address for initializing this client.



RAM Initialization Client Dialog Box

Flash Memory System Output Files (from SmartGen)

When you click **Generate the complete system,** SmartGen generates the following files; they are saved in your SmartGen project directory.

HDL Source Files

<user_name>.vhd/.v – Top level design that combines all the blocks together

<user_name>_init_wrapper.vhd/.v – Initialization and configuration instantiation wrapper for this design

<Vhdl>/<Verilog>/address_gen.vhd/v - Soft IP

<Vhdl>/<Verilog>/init_sm.vhd/v – Soft IP

<Vhdl>/<Verilog>/nvm_ctl.vhd/v – Soft IP

<Vhdl>/<Verilog>/save_sm.vhd/v – Soft IP

<Vhdl>/<Verilog>/user_clk_sel.vhd/v -Soft IP

<Vhdl>/<Verilog>/user_ctl.vhd/v – Soft IP

<Vhdl>/<Verilog>/user_valid.vhd/v – Soft IP

<Vhdl>/<Verilog>/numbits.vhd/v – package file

Memory Files

<user_name>.mem - Non-volatile memory file

<user_name>.efc - Contains all the Embedded Flash Memory partitions defined by each client of the Flash Memory System. FlashPoint uses the EFC file to create the STAPL file that programs the embedded Flash memory.

SmartGen and the Flash Memory System Builder create only one EFC file per Embedded Flash Memory System. A Fusion device may contain from 1-4 embedded Flash memories. You must use the FlashPoint user interface to associate each embedded Flash memory in the design with an EFC file.

Configuration Files

<user_name>.cfg – Captures information about the settings that were specified for the system.

<user_name>.gen – SmartGen project file; stores information about your Flash Memory System so that you can save your settings.

<user_name>.cxf – SmartGen core configuration file that contains information required by Libero IDE for file management.

Log Files

<user_name>.log

Memory File Formats in Flash Memory System Builder

The following memory file formats are available in the Flash Memory System Builder.

Actel BINARY

The simplest memory format. Each memfile contains as many rows as there are words. Each row is one word, where the number of binary digits equals the word size in bits. This format has a very strict syntax. The word size and number of rows must match exactly. The file extension is MEM; for example, file1.mem.

Example: Depth 6, Width is 8

```
01010011
11111111
01010101
11100010
10101010
11110000
```

INTEL-HEX

Industry standard file. Extensions are HEX and IHX. For example, file2.hex or file3.ihx.

MOTOROLA S

Industry standard file. File extension is S, such as file4.s

Actel-HEX

A simple address/data pair format. All the addresses that have content are specified. Addresses with no content specified will be initialized to zeroes. The file extension is AHX, such as filex.ahx. The format is:

```
AA:DDDDD
```

The data size must match the word size. Example: Depth 6, Width is 8

```
00:FF
11:BB
```

All other addresses will be zeroes.

For more information about Actel's products, call 888-99-ACTEL or visit our Web site at http://www.actel.com

Actel Corporation • 2061 Stierlin Court • Mountain View, CA USA 94043
U.S. Toll Free Line: 888-99-ACTEL • Customer Service: 650-318-4200 • Customer Service FAX: 650-318-2440
Customer Applications Center: 800.262.1060 • Customer Applications FAX: 650.318.4600

Actel Europe Ltd. • Dunlop House, Riverside Way • Camberley, Surrey GU15 3YL • United Kingdom
Tel: +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
Tel: +81.03.3445.7671 • Fax: +81.03.3445.7668

Actel Hong Kong International • Suite 2114 • Two Pacific Place • 88 Queensway, Admiralty • Hong Kong

50200062-2