Serial Distributed Arithmetic FIR Filter

December 30, 1998 | Product Specification

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
E-mail: coregen@xilinx.com
URL: www.xilinx.com

Features

- Input data words from 4 to 32 bits
- Coefficient width from 4 to 24 bits
- Output data words from 2 to 31 bits
- Taps from 6 to 80 for symmetric filters
- Taps from 6 to 40 for non-symmetric filters
- Support for cascading multiple filters to create larger filters
- Support for symmetric, non-symmetric, and negative symmetry filters
- Full precision
- Scaleable output
- Input and output synchronization signals
- Registered output
- 2's complement or unsigned input data
- Uses SelectRAM™ for high density and performance
- Uses Fast Carry logic for high speed
- Drop-in modules for the XC4000E, EX, XL, XV, and Spartan families
- High performance and density guaranteed through Relational Placed Macro (RPM) mapping and placement technology
- Available in Xilinx CORE Generator

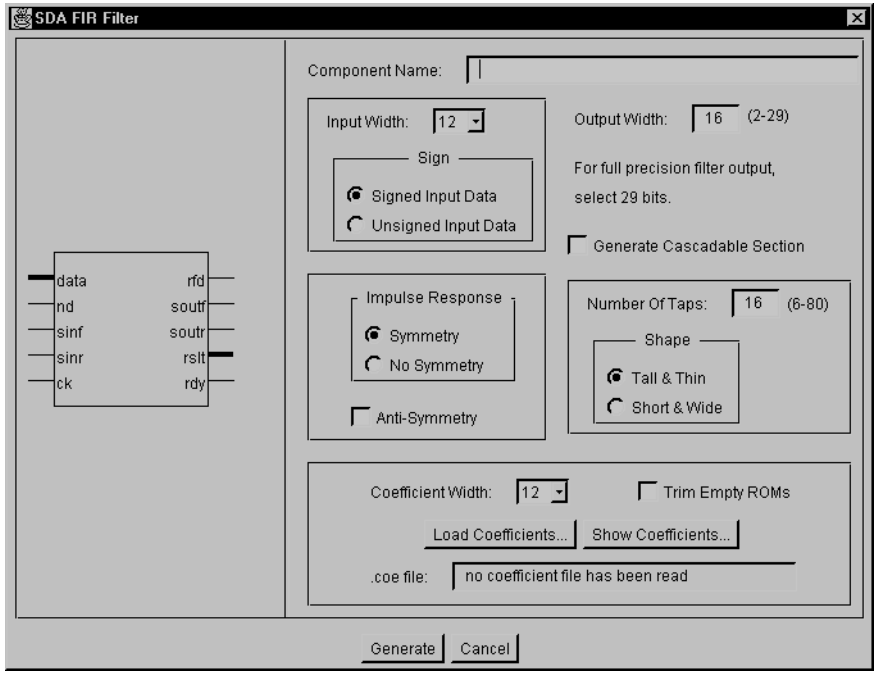

Figure 1: SDA FIR Filter Parameterization Window

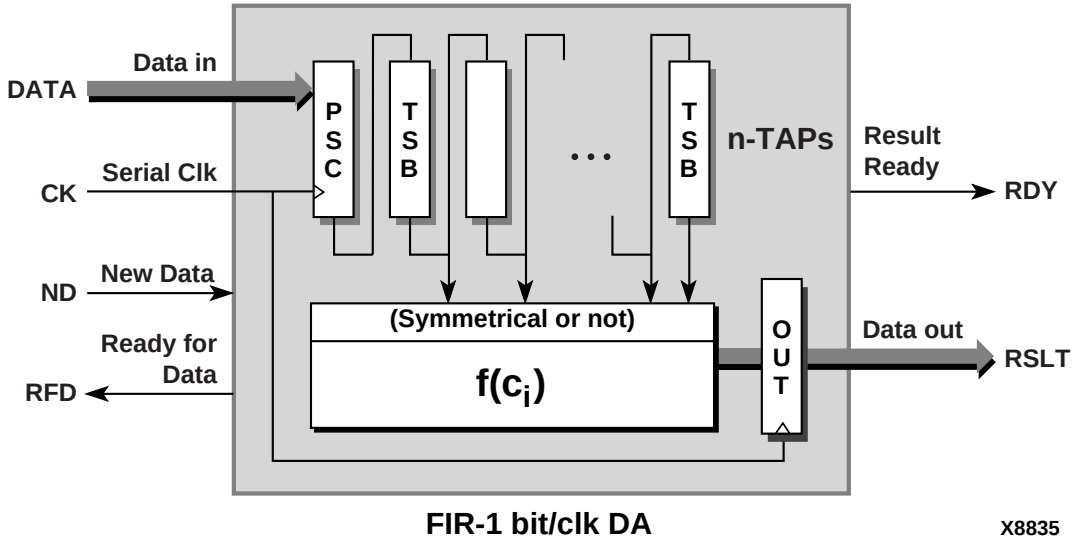

Figure 2: Serial Distributed Arithmetic FIR Filter Block Diagram

General Description

Field programmable gate arrays (FPGAs) using distributed arithmetic algorithms can implement large numbers of taps at MHz data sample rates, outperforming DSP processors by one or two orders of magnitude. A wide range of sample rates can be handled by applying the most efficient structure that just meets the required performance with the least number of FPGA configurable logic blocks (CLBs).

FIR filters are one of the most basic building blocks used in digital signal processing, taxing the performance that DSP hardware can deliver. Multiply accumulates must be performed at an ever-increasing rate and demands in the billions of MACs per second range are not uncommon.

Serial distributed arithmetic operates at relatively slow sample rates (3 to 15 million samples per second) but consumes few CLBs. Bits are processed serially, but all taps are processed in parallel. If higher sample rates are required, structures are available which process 2, 3, 4, or all of the bits in parallel. Multi-rate FIR filters can also be supported, resulting in higher sample rates with the same number of resources.

Functional Description

The FIR filter coefficients are formatted and stored in distributed ROM look-up tables when the module is generated, and the filter architecture is such that full precision is maintained through all stages of arithmetic processing. For example, a 10 tap symmetrical filter processing 8-bit input data and coefficients would yield a 19-bit result (the product of a signed and balanced 8-bit data sample and similar coefficient yields a 15-bit result, and the sum of 10 such results yields a 19-bit full precision output).

The maximum output resolution is automatically calculated and displayed on the filter parameterization window after the balance of the other filter parameters have been entered. The user can choose a full precision output or select fewer output bits if desired. In the above example, the 19-bit output could be scaled to only 12-bits if desired, resulting in the 7 LSB bits being trimmed from the filter's full precision output. Note that truncation does not significantly reduce the amount of logic consumed by the filter since full precision is always maintained internally regardless of the output bus width. The important point is that no noise is introduced into the system as a result of arithmetic operations or truncation.

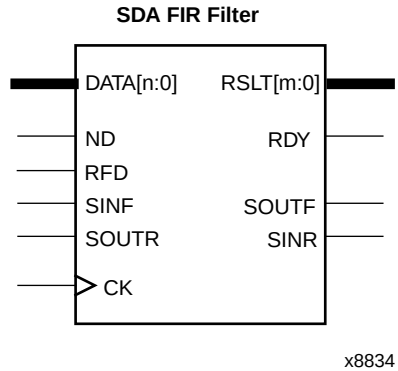

Figure 3: Core Schematic Symbol

The SDA FIR filter processes data in a bit-serial fashion and therefore requires either N or N+1 clock cycles to process N-bit sample data. (Neither the number of taps nor the bit-width of the coefficients are factors in the number of clock cycles necessary to process input data of a given bit-width.) Filters with non-symmetric impulse responses require N clock cycles to process N-bit sample data, whereas symmetric filters require an additional cycle. The user must therefore provide a clock that is at least N, or N+1, times the data sample rate.

Hand-shake Control

The filter requires a continuous clock operating at a frequency that is at least N, or N+1, times the sample rate (assuming N-bit sample data.) The filter indicates that it has finished processing the previous data sample by asserting its ready-for-data (RFD) output. A new sample may be loaded into the filter by asserting the new-data (ND) input. It is critical that ND only be asserted when RFD is High or corruption will occur within the filter. On the cycle after ND is asserted, RFD goes Low indicating that the filter is now processing the new sample. After N, or N+1, clock cycles RFD will return High indicating that the filter is ready to receive the next sample.

The filter indicates that a valid result is present on the result (RSLT) output port by asserting the result-ready (RDY) output. Valid filter outputs are stable for one clock cycle only; the cycle on which RDY is asserted. At all other times the filter's RSLT output is undefined.

Latency and Delay

The first N, or N+1, clock cycles after a piece of sample data has been loaded into the filter are used to serially shift the sample from the initial parallel-to-serial converter into the first timeskew buffer. See figure 2. It is not until the sample is shifted from the 1st timeskew buffer into the 2nd timeskew buffer that its value contributes to the filter's output.

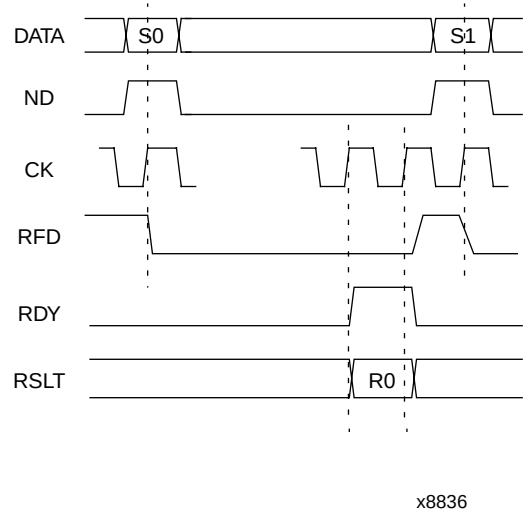

Figure 4: Handshaking Timing Diagram

The diagram shows timing waveforms for a 6-tap, non-symmetric filter processing 6-bit sample data.

1. Filter idle, waiting for a new data sample to arrive
2. Data sample S_n loaded into filter's parallel-to-serial converter.
3. Data sample S_n shifted out of parallel-to-serial converter into 1st timeskew buffer
4. Filter idle, waiting for new data sample to arrive

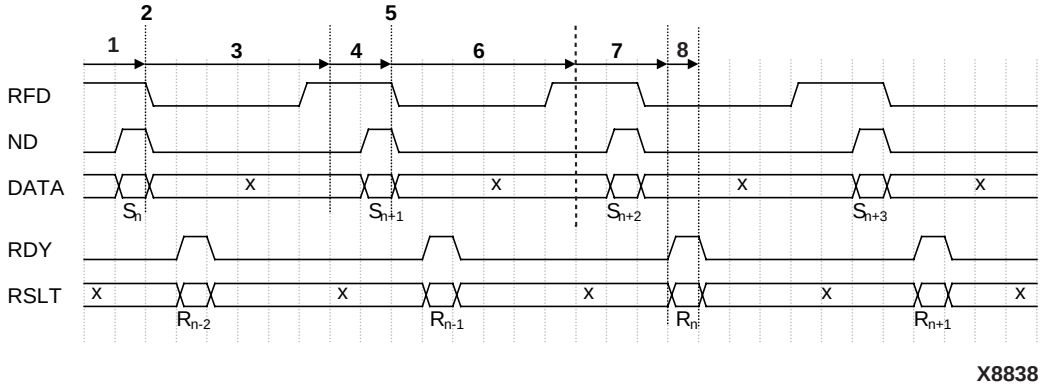

Figure 5: Timing Waveform for 6-tap Non-Symmetric Filter Processing 6-bit Sample Data

5. Next data sample loaded into filter
6. Data sample Sn shifted out of 1st timeskew into 2nd timeskew buffer as the data is transferred, its value contributes to the filter's result
7. Once all data bits have been processed, the newly calculated result propagates through internal look-up table and adder tree latency
8. Valid result available at filter output

The filter's internal latency (due to look-up table and adder tree delay) can be found in Table 1.

Assuming that the filter's clock is free running, the following assumptions may be made about the filter's operation:

- The RFD flag will always go High N (non-symmetric) or N + 1 (symmetric) cycles after ND is asserted.

Interpretation: The filter will indicate it is ready to receive a new piece of sample data, N or N + 1 cycles after the previous piece of sample data was loaded.

- The RDY flag will always go High N + L (non-symmetric) or N + L + 1 (symmetric) cycles after ND is asserted.

Interpretation: The filter will generate a new result N + L, or N + L + 1, clock cycles after a new piece of sample data has been loaded.

Pinout

Port names for the schematic symbol are shown in Figure 3 and described in Table 2.

Core Generator Parameters

The CORE Generator parameterization window for this macro is shown in Figure 6. The parameters are as follows:

- **Component Name:** Enter a name for the output files generated for this module.
- **Input Width**: Select an input bit width from the pull-down menu. The valid range is 4-32.
- **Output Width**: Full arithmetic precision is carried throughout the filter, however you may truncate the output bit-width to a smaller value. The valid range depends on other filter parameters.
- **Generate Cascadeable Section**: Allows this module to be combined with other filter modules to form larger filters. Check this box when constructing a filter for cascading purposes.
- **Impulse Response:** Select whether the filter should have a symmetric or non-symmetric impulse response.
- **Anti-Symmetry:** Select whether the filter should have an anti-symmetric (also known as negative-symmetry) impulse response. This control is only enabled if a symmetric impulse response is selected.
- **Number Of Taps:** Choose the number of taps required for the filter. For symmetric filters the valid range is 6 to 80 taps. For non-symmetrical filters the range is 6 to 40.
- **Shape:** For symmetric filters with between 27 and 40 taps two different shape floorplans are available. The rectangle of CLBs occupied by the filter can be tall-and-thin or short-and-wide.
- **Coefficient Width:** Specify the bit-width necessary for the filter's largest coefficient. The valid range is 4-24.
- **Trim Empty ROMs:** Since not all coefficients will require all 'coefficient width' bits to express them, this option allows the filter's internal look-up tables to be optimized to the smallest possible dimensions given the actual coefficient values.
- **Load Coefficients:** Get the filter coefficients from a specified file.
- **Show Coefficients**: Display the coefficients after they have been loaded.
- **.coe File**: Displays the name of the coefficient file. This field is read-only.

Specifying Filter Coefficients

Not all of the parameters necessary to completely specify a filter's characteristics can be entered on the parameterization window. Specifically the filter coefficients must be loaded from a text file (known as a COE file) described below. In addition, all the parameters visible on the parameterization window may be assigned in the COE file. COE files may take any root filename but must end with the extension ".COE".

Table 1: Latency Table

Non-Symmetric		Symmetric	
# Taps	Latency (L)	# Taps	Latency (L)
3-10	3	6-20	4
11-20	4	21-40	5
21-40	5	41-80	6

Table 2: CORE Signal Pinout

Signal	Signal Direction	Description
DATA[n:0]	Input	PARALLEL DATA INPUT – N-bit wide sample data
ND	Input	NEW DATA - When asserted, the value on the DATA port is captured and internal processing begins. Must not be asserted when RFD is Low.
RFD	Output	READY FOR DATA – Active High. Indicates when the last bit of the previously loaded sample is about to be processed and new data may be loaded into the filter.
SINF	Input	SERIAL CASCADE DATA IN, FORWARD - Used when cascading multiple filters. Leave open-circuit when not cascading.
SOUTR	Output	SERIAL CASCADE DATA OUT, REVERSE - Used when cascading multiple filters. Leave open-circuit when not cascading.
CK	Input	BIT RATE CLOCK - With the exception of asynchronous control inputs (where applicable), control and data inputs are captured, and new output data formed on rising clock transitions.
RSLT[m:0]	Output	RESULT – Parallel filter output. The result of summing the products of the coefficients and the recorded sample data.
RDY	Output	RESULT READY – Active High. Indicates that valid output data is present on the RSLT port.
SOUTF	Output	SERIAL CASCADE DATA OUT, FORWARD - Used when cascading multiple filters. Leave open-circuit when not cascading.
SINR	Input	SERIAL CASCADE DATA IN, REVERSE - Used when cascading multiple filters. Leave open-circuit when not cascading.

To select and load a COE file, press the “Load Coefficients...” button on the filter parameterization window and choose the desired file from the dialog-box. To review the filter’s coefficient-values at any time, press the “Show Coefficients...” button. Any field on the parameterization window that is assigned a value in the COE file will lose its previous value when the COE file is loaded. Changing a parameter value that was previously loaded from a COE file causes the COE file’s name to be highlighted in red, indicating that the settings have been changed since the file was loaded.

For a detailed description of COE file syntax, please refer to the Xilinx CORE Generator User Guide. The list of COE file keywords supported by the Serial Distributed Arithmetic FIR Filter are shown in the parameter File Information table at the end of this datasheet. An example COE file is shown below.

```
component_name = section1;
antisymmetry = false;
trim_empty_roms = false;
short_wide_floorplan = true;
input_width = 8;
coef_width = 8;
number_of_taps = 10;
cascade = true;
symmetry = true;
signed_input_data = true;
output_width = 19;
coefdata = 1,2,3,4,5;
```

Example COE File For SDA FIR Filter

When specifying filter coefficients in a COE file, the radix is assumed to be 10. You may however specify COEFDATA values in hexadecimal by adding the command RADIX=16.

Filters with non-symmetric impulse responses require as many coefficients specified in their COE file as they have taps. I.e. a 10-tap non-symmetric filter requires10 coefficients, C_0 - C_9.

Filters with symmetric impulse responses require half as many coefficients specified in their COE file as they have taps. I.e. a 10-tap symmetric filter requires only 5 coefficients, C_0 - C_4. Coefficients C_5 - C_9 are assigned the values of coefficients C_4 - C_0 when anti-symmetry is not selected, and the negated values of coefficients C_4 - C_0 when anti-symmetry is selected.

Filter Output Bit-Width

The filter calculates the number of bits required at its RSLT output port on the basis of the parameters provided by the user. The width of the filter’s RSLT port will always be sufficient to express the largest magnitude result that may ever arise during the filter’s operation. However, the bit-width

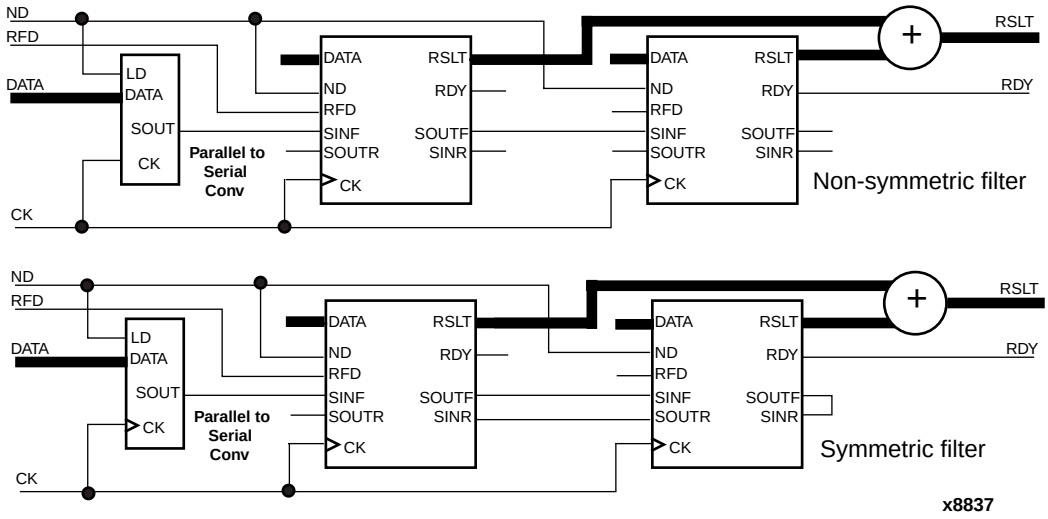

Figure 6: Cascading Several Filter Modules to Create Larger Filters

calculation is conservative and a RSLT port with more bits than necessary may be created.

The process used to determine the filter's output port bit-width is:

Largest Coefficient Value (Given the coefficient bit-width provided by the user)

$2^{\text{Coefficient Bit Width} - 1}$

Largest Sample Value (Given the data bit-width provided by the user)

$2^{(\text{Data Bit Width1})} - 1$ for signed data,
$2^{\text{Data Bit Width}} - 1$ for signed data

Largest Filter Output Value (Given the number of taps provided by the user)

Largest Sample Value * *Largest Coefficient Value* * Number Of Taps

Number Of Bits Required At Output Port

$\text{Log}_2(\textit{Largest Filter Output Value}) + 2$ ...Rounded down to the previous whole number

Given that not all coefficients will be as large as the *Largest Coefficient Value* calculated above, the filter's RSLT output port may, at times, be wider than necessary. Given a specific set of coefficients, the ideal output port bit-width may be found manually by substituting the sum of the magnitudes of the coefficients for the product of the *Largest Coefficient Value* and the Number Of Taps. That is,

Largest Sample Value (Given the data bit-width provided by the user)

$2^{(\text{Data Bit Width1})} - 1$ for signed data,
$2^{\text{Data Bit Width}} - 1$ for signed data

Largest Filter Output Value (Given the coefficients provided by the user)

Largest Sample Value * $(|C_0| + |C_1| + |C_2| + ... + |C_n|)$

Number Of Bits Required At Output Port

$\text{Log}_2(\textit{Largest Filter Output Value}) + 2$ ...Rounded-down to the previous whole number

The difference between the conservative bit-width calculated by the Core Generator and the ideal bit-width calculated manually represents the number of unnecessary MSBs that are included in the filter's output port. These bits may safely be dropped from the filter's output, and a small amount of logic trimming may occur as a result.

As an example, take a symmetric 6-tap filter, with coefficients 1,-8,12,12,-8 and 1. The center coefficients require 5 bits to express, but the other coefficients require only 3 and 2 bits. Assuming 6-bit sample data, the Core Generator conservatively calculates that this filter requires a 13-bit RSLT port. Given the specific coefficients however, only 12 bits are actually necessary. In this application therefore, the MSB of the filter's output may be discarded.

Cascading Multiple SDA FIR Filters

Filters requiring numbers of taps greater than the limit imposed by this module can be constructed by 'cascading' several filter modules and summing the results of each to form a final output. Cascading filter modules is performed using the SINF and SOUTF pins in the case of non-symmetric filters, and also the SINR and SOUTR pins in the case of symmetric filters. See Figure 4.

In cascade-mode, each filter module receives its sample data bit-serially from its neighbor. Setting a filter's cascade

option suppresses the generation of the internal parallel-to-serial converter and the module's parallel DATA port becomes redundant. Therefore, the first filter in the chain should receive its sample data from an external parallel to serial converter. Serial data is passed along the chain of filter modules via the SINF-SOUTF links. If the composite filter is required to have a non-symmetric impulse response then the SOUTF pin of the last module in the chain may be left open circuit.

If the composite filter is required to have a symmetric impulse response, then the SOUTF pin of the last filter in the chain should be attached to its own SINR pin. This has the effect of sending the emerging sample data back along the filter chain in the reverse direction. The data propagates back along the chain of filters via the SOUTR-SINR links. The SOUTR pin on the first filter module may be left open-circuit.

Since the SDA FIR filter module's latency is related to the number of taps it implements, to ensure that results appear on the outputs of each of the filter modules at the same instant, make all cascaded sections the same number of taps. Only the last filter in the chain may have an odd number of taps, i.e. when the total number of taps is desired to be odd, give each filter module an even number of taps, except the last one which may be odd. (The same number of taps as the other sections minus 1.)

Core Resource Utilization

Table 2 shows the approximate number of CLBs required for various example bit widths. Table 3 shows the maximum speed for data sample rates using XC4000E-1 devices. The sample rate is calculated by dividing the system clock by the number of bits in the data word, N for non-symmetrical FIR filters, or N+1 for symmetric filters. Sample rates are independent of the number of taps, but a small increase in delay per clock is a function of the width of the coefficients due to the increase in the width for the adders.

Ordering Information

This macro comes free with the Xilinx CORE Generator. For additional information contact your local Xilinx sales representative, or e-mail requests to dsp@xilinx.com.

Table 3: CLB Utilization for Example SDA FIR Filter Implementations

Taps	Symmetry	CLB Counts for Example Data Word Widths[1]							
		5-bit	8-bit	10-bit	12-bit	14-bit	16-bit	18-bit	20-bit
8	Symmetrical		33	36	39	42	45	52	55
	Non-symmetrical		46	54	59	64	69	77	85
16	Symmetrical	53	61	69	71	76	81	96	102
	Non-symmetrical		80	95	104	112	123	138	
24	Symmetrical	80	89	101	108	116	127	146	154
	Non-symmetrical		101	114	127	140	153	174	187
32	Symmetrical	93	107	118	126	137	148	175	182
40	Symmetrical	116	138	154	165	179	191	226	239
48	Symmetrical		158	173	187	202	217	246	261
64	Symmetrical		197	215	233	250	268	305	323
80	Symmetrical		236	257	278	299	320	364	385

Note:

1. Coefficient width is equal to the word width.

Table 4: SDA FIR Filter Data Sample Rates in MHz Using XC4000E-1

Symmetry	5-bit	8-bit	10-bit	12-bit	14-bit	16-bit	18-bit	20-bit
Symmetrical	13.3	8.9	7.3	6.2	5.3	4.7	4.2	3.8
Non-symmetrical	16.0	10.0	8.0	6.7	5.7	5.0	4.4	4.0

Parameter File Information

Parameter Name	Type	Notes
Component_Name	String	
Input_Width	Integer	4 - 32
Output_Width	Integer	
Coef_Width	Integer	4 - 24
Number_Of_Taps	Integer	6 - 80
Symmetry	Boolean	True/False
Antisymmetry	Boolean	True/False
Trim_Empty_ROMs	Boolean	True/False
Cascade	Boolean	True/False
Signed_Input_Data	Boolean	True/False
Short_Wide_Floorplan	Boolean	True/False
CoefData{n} 0<=n<=39	Integer	
Radix*	Integer	10 or 16

*COE file only.