Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

CoreFIR v8.6

Handbook

Table of Contents

Introduction

Core Overview

The finite impulse response (FIR) filter is one of the most essential building blocks in digital signal processing (DSP) systems. Digital filters have been used in many systems to remove unwanted noise, improve signal quality, or shape signal spectrum.

CoreFIR provides a configurable high performance multiplier-accumulator (MAC)-based FIR filter. The core is available as a register transfer level (RTL) code of the filter, both in Verilog and VHDL languages.

The core implements a range of filter types:

- Single-rate
  - Fully enumerated (parallel)
  - Folded (semi-parallel)
- Multi-rate
  - Polyphase interpolation
  - Polyphase decimation

The N-tap single rate FIR filter computes its sum-of-products output *y(k)* as explained in EQ 1:

$$y(k) = \sum_{j=0}^{N-1} x(k-j)c(j) \;= x(k)c(0) + x(k-1)c(1) + \cdots + x(k-N+1)c(N-1)$$

EQ 1

A series of coefficients *c(0), c(1), c(2), …, c(N-1)* are the filter impulse response. The coefficient values define whether the filter is a low-pass, band-pass, or high-pass filter. The fully enumerated filter type has as many physical MACs as filter taps. Such architecture provides the fastest input sample rate. The folded filter utilizes fewer physical MACs, taking advantage of a ratio between high clock rate and a slower input sample rate. It reuses the MACs over vacant clock intervals to process the input samples.

Figure 1 shows the single rate filter functional block diagram. The figure presents only a general notion of a digital filter computational component. An actual realization can be quite different. This handbook contains chapters dedicated to every filter type. Refer to the specific chapter for more information on each particular type.

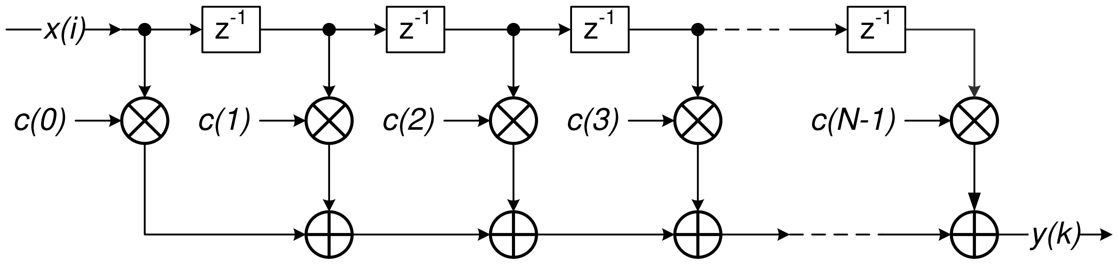

Figure 1 · Single-Rate FIR Filter Functional Block Diagram

CoreFIR supports the following three filter coefficient modes:

- Constant coefficient
- Multiple constant coefficient sets
- Reloadable coefficient

In Constant Coefficient mode, a single coefficient set can be programmed into field programmable gate array (FPGA) fabric. In Multiple Constant Coefficient mode, the multiple constant coefficient sets can be programmed, and the FPGA stores all of them. It is possible to switch between the pre-loaded sets at any moment during runtime, thus changing the filter impulse response. In Reloadable Coefficient mode, the coefficients can be reloaded at the runtime.

Internal filter processing takes place at full precision to reduce truncation or rounding noise and to avoid a risk of overflow. The filtered results are presented in full precision as well.

Figure 2 shows an example of using a FIR filter. Digital samples to be filtered enter the filter input and filtered samples appear at the filter output.

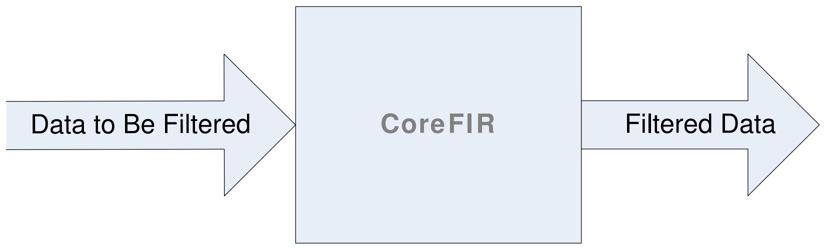

Figure 2 · CoreFIR Example Application

Key Features

CoreFIR supports various filter types: Fully Enumerated, folded, and polyphase interpolator. The key features for each type are listed in Table 1.

Table 1 Key Feature Support

Feature	Fully Enumerated	Folded	Interpolation
Number of filter coefficients	2 to 2N, where N is a number of physically available MAC’s	4 to 1024	2 – 1,024
Input data bit width	2 – 18	2 – 18	2 – 18
Coefficient bit width	2 – 18	2 – 18	2 – 18
Signed and unsigned data coefficients	Yes	Yes	Yes
Full precision output	Yes	Yes	Yes
Coefficient symmetry optimization	Yes	No	No
Constant coefficients and constant coefficient sets	Yes	Yes	Yes
Run-time reloadable coefficients	Yes	Yes	Yes
RAM-based coefficient storage	No	Yes	Yes
RAM-based data storage	No	Yes	Yes

Supported Families

The core supports RTG4™, SmartFusion®2, and IGLOO®2 FPGA families.

Core Version

This handbook applies to CoreFIR v8.6.

Utilization and Performance

CoreFIR has been implemented in the SmartFusion2 devices using speed grade-1. A summary of the implementation data is listed in Table 2.

Fully Enumerated Filter

Table 2 · Fully Enumerated CoreFIR M2S050 Device Utilization and Performance

Taps	Symmetry	Architecture	Configuration	Bit Width		Mathblocks	Resource Usage			Max. Data Rate (Msps)
				Coefficient	Data		Sequential	Combinatorial	Total	
24	No	Transposed	1	18	18	24	28	168	196	310
32	No	Transposed	1	18	18	32	40	385	425	310
48	No	Transposed	1	18	18	48	56	403	459	310
72	No	Transposed	1	18	18	72	84	637	721	286
16	No	Systolic	1	18	18	16	35	714	749	423
24	No	Systolic	1	18	18	24	51	1,020	1,071	423
48	No	Systolic	1	18	18	48	103	2,144	2,247	412
72	No	Systolic	1	18	18	72	155	3275	3,430	371

Note: Data in this table were achieved using typical synthesis settings. Synplify Frequency (MHz) was set to be 400. CoreFIR was configured for signed coefficients and data, a single set of constant coefficients, and RSTN pin was tied to Vcc
Layout settings were set as follows:
- Block creation enabled
- Timing-driven High Effort Layout

The maximum data rate shown also reflects the maximum clock rate. For example, the rate of 100 Msamples per second corresponds to the maximum clock rate of 100 MHz.

Folded Filter

The filter was realized on M2S150 device.

Table 3 Folded CoreFIR Device Utilization and Performance

Coefficients				Data Width	Folding Factor	Use RAM		Micro RAM Depth	Utilization					Maximal Clock Rate, MHz
Type	Number of Taps	Width	Number of Sets			For Coefficients	For Data		4LUT	DFF	R64X18	RAM1K18	MACC	
Constant	29	18	1	18	2	No	No	0	2271	3061	-	-	15	267
Constant	29	18	1	18	3	No	No	0	1776	2392	-	-	10	258
Constant	29	18	1	18	10	No	No	0	1102	1454	-	-	3	254
Constant	60	18	1	18	60	Yes	Yes	64	404	438	2	-	1	250
Constant	60	18	3	18	60	Yes	Yes	64	427	345	1	1	1	250
Reloadable	60	18	1	18	60	Yes	Yes	64	385	382	1	1	1	250
Constant	1024	18	1	18	100	Yes	Yes	0	1733	1778	-	2	11	250
Reloadable	1024	18	1	18	100	Yes	Yes	0	1547	1869	-	3	11	273

Note: *Data in this table were achieved using typical synthesis settings. Synplify Frequency (MHz) was set to be 300. CoreFIR was configured for signed coefficients and data, RSTN pin was tied to Vcc and REF pin was grounded*
Layout settings were set as follows:
- Block creation enabled
-Timing-driven High Effort Layout

Interpolation Filter

The filter was realized on M2S150 device

Table 4 Interpolation CoreFIR Device Utilization and Performance

Coefficients				Data Width	Interpolation Factor	Use RAM For Coefficients	Micro RAM Depth	Utilization					Maximal Clock Rate, MHz
Type	Number of Taps	Width	Number of Sets					4LUT	DFF	R64X18	RAM1K18	MACC	
Constant	16	18	1	18	4	No	0	359	700	-	-	4	420
Constant	16	18	1	18	8	No	0	305	477	-	-	2	418
Constant	16	18	4	18	4	No	0	600	916	-	-	4	385
Constant	16	18	4	18	8	No	0	529	708		-	2	348
Reloadable	16	18		18	4	No	0	671	1285			4	257
Reloadable	16	18		18	8	No	0	605	1036			2	267
Constant	128	18	4	18	16	Yes	0	863	1205	-	8	8	367
Constant	128	18	4	18	32	Yes	0	574	690	-	4	4	374
Constant	128	18	1	18	16	Yes	64	782	1178	8		8	250
Constant	128	18	4	18	16	Yes	64	863	1205	8		8	250
Reloadable	128	18		18	16	Yes	64	727	1236	8		8	250
Constant	1024	18	1	18	256	Yes	0	684	920		4	4	351
Constant	1024	18	4	18	256	Yes	0	766	935		4	4	321
Reloadable	1024	18		18	256	Yes	0	459	963		4	4	343

Note: *Data in this table were achieved using typical synthesis settings. Synplify Frequency (MHz) was set to be 400. CoreFIR was configured for signed coefficients and data, RSTN pin was tied to Vcc and REF pin was grounded*
Layout settings were set as follows:
- Block creation enabled
-Timing-driven High Effort Layout

Decimation Filter

The filter was realized on M2S150 device

Table 5 Decimation CoreFIR Device Utilization and Performance

Coefficients				Data Width	Decimation Factor	Use RAM		Micro RAM Depth	Utilization					Maximal Clock Rate, MHz
Type	Number of Taps	Width	Number of Sets			For Coefficients	For Data		4LUT	DFF	R64X18	RAM1K18	MACC	
Constant	16	18	1	18	4	No	No	0	388	977			5	425
Constant	16	18	4	18	4	No	No	0	591	1186			5	413
Reloadable	16	18		18	4	No	No	0	694	1524			5	294
Constant	128	18	1	18	16	Yes	Yes	64	1058	1390	15		9	250
Constant	128	18	4	18	16	Yes	Yes	64	1077	1404	15		9	250
Reloadable	128	18		18	16	Yes	Yes	64	1036	1440	15		9	250
Constant	1024	18	1	16	256	Yes	Yes	0	727	12936		4	5	319
Constant	1024	18	4	16	256	Yes	Yes	0	729	12942		4	5	319
Reloadable	1024	18		16	64	Yes	Yes	0	484	12962		4	5	330
Constant	1000	18	1	16	250	Yes	Yes	0	717	12645		4	5	313
Constant	1020	18	2	16	60	Yes	Yes	64	2291	2718	16	17	18	250
Constant	1024	18	3	16	256	Yes	Yes	0	729	12942		4	5	351
Reloadable	1000	18		16	250	Yes	Yes	0	488	12647		4	5	338

Note: Data in this table were achieved using typical synthesis settings. Synplify Frequency (MHz) was set to be 400. CoreFIR was configured for signed coefficients and data, RSTN pin was tied to Vcc and REF pin was grounded
Layout settings were set as follows:
- Block creation enabled
-Timing-driven High Effort Layout

Filter Types

Depending on a user configuration, CoreFIR generates one of the supported filter types.

Fully Enumerated Filter

The single rate parallel FIR filter provides the highest input sampling rate processing. The performance of the parallel filter expressed as the maximum input sample frequency equals the maximum clock rate, that is, the filter can process a sample per clock interval. Figure 1 on page 5 shows an accurate functional model of the fully enumerated filter. The filter utilizes as many MAC blocks as the number of coefficients also called the number of taps. The largest filter (in terms of number of coefficients) a particular FPGA device can carry is limited by the number of available MAC blocks.

Folded Filter

This one is a single-rate semi-parallel filter. In many practical cases, the filter input sample rate equals only a fraction of the FPGA clock rate. Then the processing power of the MAC block can be re-used to process more than a single filter tap. At every clock interval, the semi-parallel filter computes and accumulates several products of EQ 1 on page 5. In a few clocks before a next input sample comes in, all N products are accumulated.

The number PHY_TAPS of the products the filter engine computes at every clock interval is determined based on required number of taps N, input sample frequency, and the FPGA clock frequency as follows:

$$PHY_{TAPS} \geq \mathrm{N} * \frac{\text{Sample frequency}}{\text{Clock frequency}}$$

EQ 2

Figure 3 on page 14 shows the folded filter simplified functional block diagram. Input data samples come to a Data FIFO that collects a number of samples sufficient to compute PHY_TAPS products each clock interval. The Coefficient ROM stores N coefficients. The MAC engine has PHY_TAPS MACs, a delay line comprised of Z^{-1} blocks, and an output multiplexer. After filling in a delay line with necessary input samples, the MAC engine reads N data samples simultaneously with N coefficients to compute PHY_TAPS filtered results. After the computation is over, the PHY_TAPS results are collected in PHY_TAPS accumulators. The multiplexer (MUX) puts them out one by one.

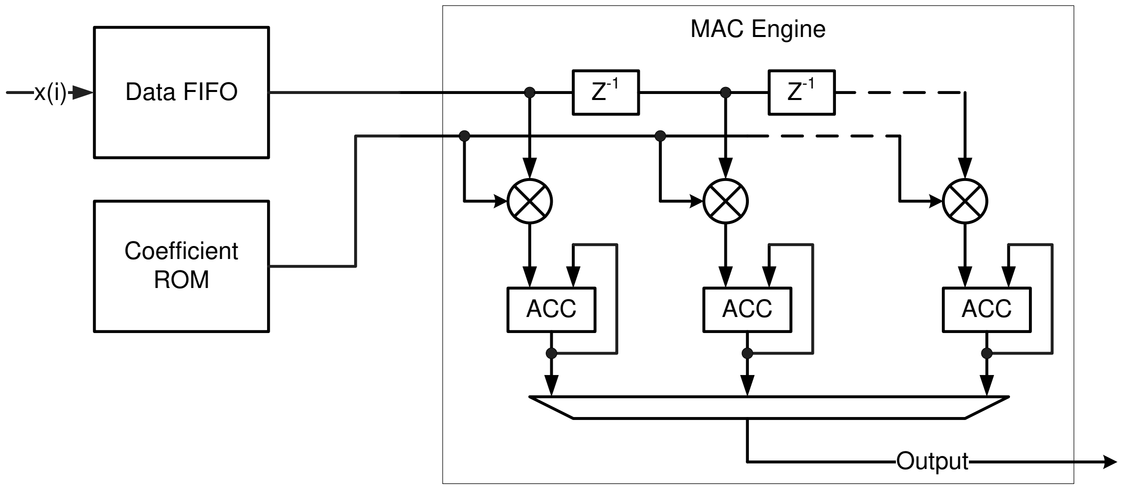

Figure 3 · Semi-Parallel FIR Filter Functional Block Diagram

$$\text{Maximum sample frequency} = \text{Clock frequency} * (\text{PHY\_TAPS})/\text{N}$$

EQ 3

There is no need to indicate whether the single rate filter is to be fully enumerated or folded. It is only required to indicate the clock, and input sample frequencies. If the clock to sample rate ratio is not less than 2, CoreFIR automatically generates the folded type. If the fully enumerated type is needed, the same value for the clock and sample frequencies must be entered.

Polyphase Interpolation Filter

The primary reason for interpolation is to increase the sampling rate at the output of one system so that another system operating at a higher sampling rate can input the signal.

Through calculations on existing data, interpolation fills in missing information between the samples of a signal. Interpolation increases a sample rate by an integer factor L.

The architecture calculates output using N/L multipliers, where N is a number of filter coefficients and L is an Interpolation factor, to get an overall computational saving of (N - N/L) compared to the straightforward implementation.

At every clock interval the architecture computes and accumulates several products, as shown in EQ 4.

$$Yp(k) = \sum_{j}^{N} X(k-j) * h(j),$$

EQ 4

Where, P = 0 to (L-1) and j = P, P+ (L-1), P+ 2*L – 1, 3*L – 1,…N

Figure 4 on page 13 provides an example of the polyphase interpolation filter architecture for a TAP of 16 and Interpolation factor of 4. The interpolator contains distributed coefficient ROM, one storage per physical filter tap. In Figure 4 on page 13, the first storage keeps coefficients h_0 to h_3, the second storage keeps coefficients h_4 to h_7, and so on.

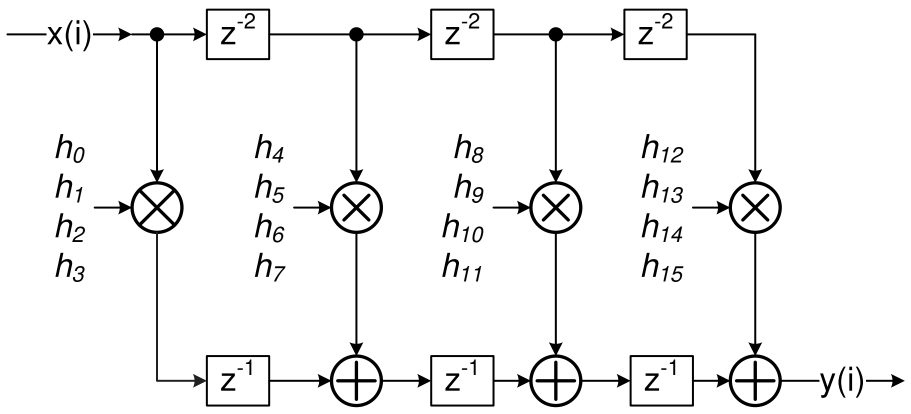

Figure 4 · 16-Tap Polyphase Interpolation Filter, L = 4

Polyphase Decimation Filter

The motivation for decimation is to reduce the cost of processing: the calculation to implement a DSP system, generally is proportional to the sampling rate, so the use of a lower sampling rate usually results in a cheaper implementation.

The decimation factor is simply the ratio of the input rate to the output rate. It is usually symbolized by "M", so input rate/output rate = M.

If a signal is defined by N samples, M:1 decimation is achieved by throwing away M – 1 samples after every sample kept In an M:1 decimator, the output data rate is 1/M times the input data rate, and M is the decimation factor.

The output Y is given by EQ 5.

$$Y(n) = \sum_{i}^{N} X(i) * h[nM - i]$$

EQ 5

Y (n) is discarded for n != 0, M, 2*M, 3*M etc.

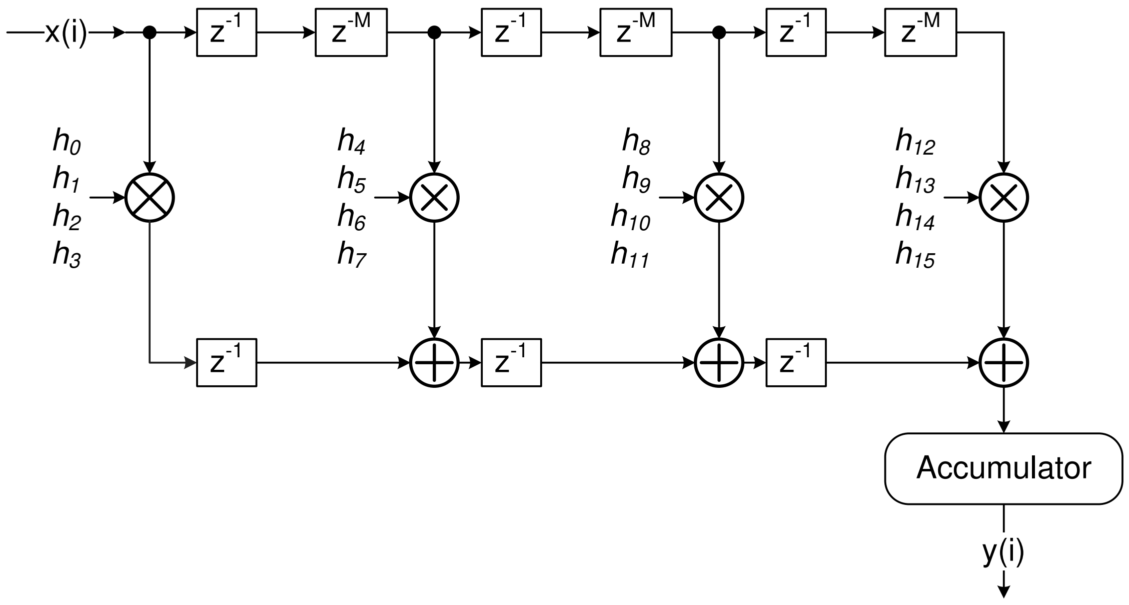

Figure 5 · 16 TAP Polyphase Decimation Filter, M = 4

Figure 5 shows an example of the polyphase decimation filter structure for a number of taps TAPS = 16 and a decimation factor M = 4. A decimated output sample is obtained at the accumulator. A valid output is available only at every M^{th} clock. The decimator contains distributed coefficient ROM, one storage per physical filter tap. In Figure 5, the first storage keeps coefficients h_0 to h_3, the second storage keeps coefficients h_4 to h_7, and so on.

Fully Enumerated Filter

Filter Description

This section describes how the fully enumerated CoreFIR filter implements hardware (HW) architectures. All the architectures realize EQ 1 on page 5.

Transposed Architecture

Figure 6 depicts the transposed HW structure. The architecture is functionally equivalent to the structure shown in Figure 1, but eliminates the need for a multi-input adder. Instead, it uses the pipelined two-input adder chain, which makes it highly beneficial for the HW implementation. It also offers low latency implementation.

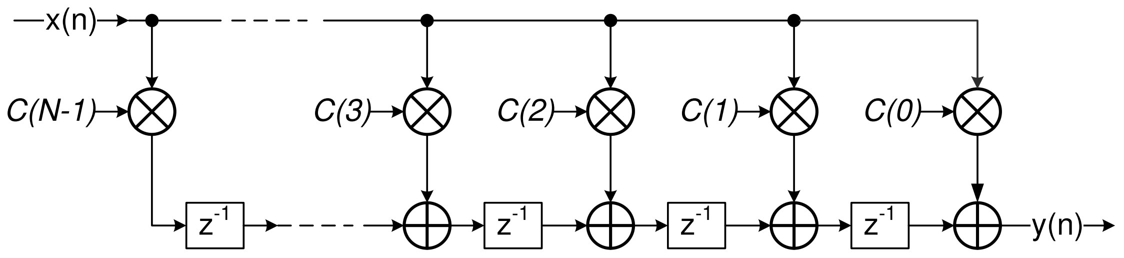

Figure 6 · Transposed FIR Filter Architecture

Systolic Architecture

The systolic architecture adds more pipelines to the Direct Form I structure of Figure 1. Depending on particular filter parameters, the systolic structure can offer better performance due to extensive pipelining. The structure is always used to implement FIR filters with symmetric impulse response. The systolic architecture is shown in Figure 7.

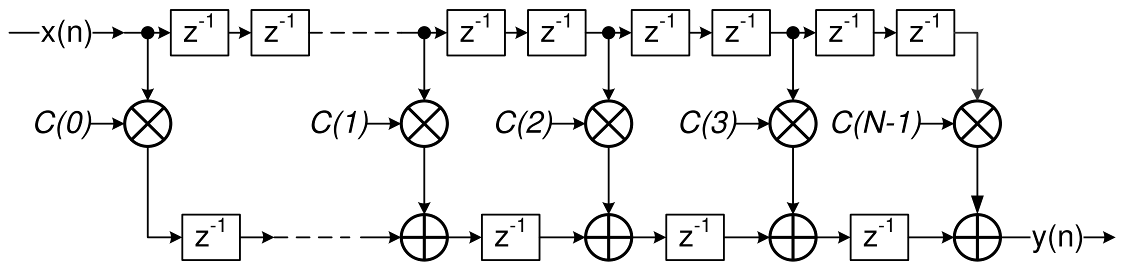

Figure 7 · Systolic FIR Filter Architecture

Fully Enumerated Filter Symmetry

Many FIR filters are symmetric. The fully enumerated filter exploits the filter symmetry to minimize the number of physical MACs required for the implementation. Figure 8 shows the impulse response for a 9-tap symmetric filter. In this filter *c(0) = c(8), c(1) = c(7), c(2) = c(6), and c(3) = c(5)*. When exploited, the symmetry can substantially reduce the number of multipliers. The symmetric FIR filter implementation first adds together the data samples that engage equal coefficients. Figure 9 shows the corresponding functional block diagram.

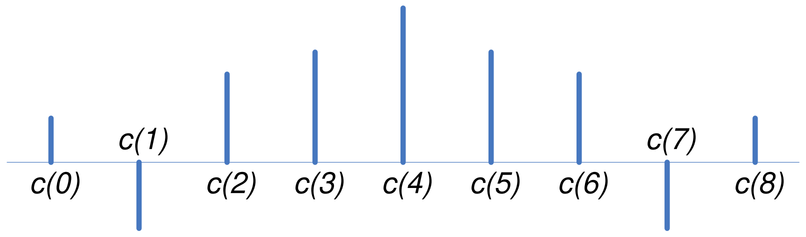

Figure 8 · Symmetric FIR Filter Impulse Response

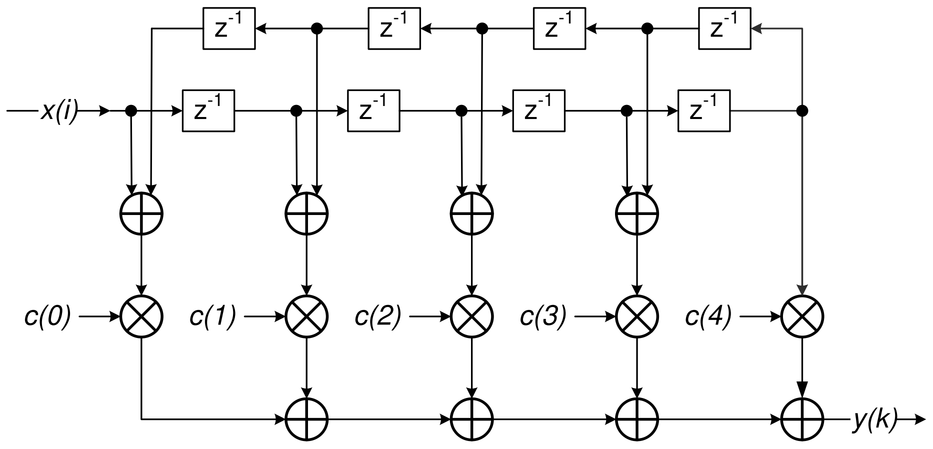

Figure 9 · Symmetric FIR Filter

There are a few types of symmetry. Figure 8 shows an example of an odd symmetry, where the number of taps is odd, and a central tap is unique. Even symmetry is applied to a filter with an even number of taps. The even symmetric filter does not have a unique center tap.

Figure 10 shows an 8-tap anti-symmetric filter impulse response. In this filter *c(0) = -c(7), c(1) = -c(6), c(2) = -c(5), and c(3) = -c(4)*. Similarly, odd anti-symmetric filters exist. The core exploits all four symmetry types to nearly double the tap number, given the same number of mathblocks utilized.

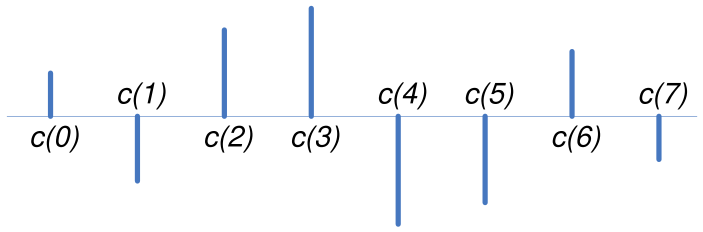

Figure 10 · Even Anti-Symmetric FIR Filter Impulse Response

All symmetric/anti-symmetric filters utilize the systolic architecture in Figure 7.

Reloadable Coefficient Mode

Figure 11 shows a functional block diagram for Reloadable Coefficient mode. This mode uses two memory pages: active and auxiliary. The active page comprises Storage registers connected to the multiplier coefficient inputs. The auxiliary page implements a multi-bit Shift register. Once a new coefficient vector is loaded in the auxiliary page, it can be copied in the active page within a single clock period. Then the auxiliary page is ready to accept another coefficient vector. Thus, the filter operation is not disturbed during coefficient reload.

New loadable coefficients arrive at COEFI input to be shifted in the multi-bit Shift register. Every new coefficient shifts the register contents one step to the left. The coefficients come in natural order, c(0), c(1),…,c(N-1). After the last coefficient arrives, the coefficients in the auxiliary page are distributed, as shown in Figure 11. On the COEF_ON signal the recently entered coefficients are loaded into the active page. From this time on, they are used as the filter coefficients.

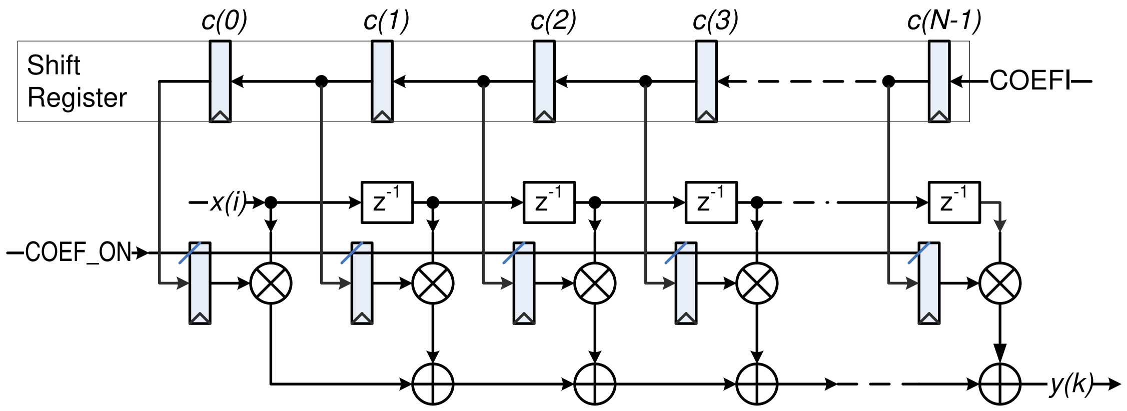

Figure 11 · Reloadable Coefficient Mode.

Fully Enumerated Interface Description

Parameters/Generics

The parallel fully enumerated CoreFIR RTL has parameters (Verilog) or generics (VHDL), described in Table 6. All the parameters and generics are positive integer types. The table shows the superset of the parameters used by all FIR filter types, while indicating which parameters the fully enumerated type does not utilize.

Table 6 · Fully Enumerated Filter Parameter/Generic Descriptions

Parameter Name	**Valid Range**	**Default**	**Description**
CFG_ARCH	1-4	1	Filter type: 1: Fully Enumerated 2: Folded 3: Polyphase Interpolator 4: Polyphase Decimator
TAPS	2-2N[1]	16	Number of taps. For a symmetric filter the valid range is 2 to 2*N (2 to 2*N-1 for odd symmetry filters), for the other filters the range is 2 to N.
COEF_TYPE	0-1	0	0: Constant coefficients (including multiple coefficient sets) 1: Reloadable coefficients
COEF_SETS	1-16	1	1: Single coefficient set 2-16: Multiple coefficient sets Valid when constant coefficient type is selected (COEF_TYPE==0)
COEF_SYMM	0-2	0	0: Not symmetric coefficients 1: Symmetric coefficients 2: Anti-symmetric coefficients
COEF_UNSIGN	0-1	0	0: Signed coefficients 1: Unsigned coefficients
COEF_WIDTH	2-18	12	Coefficient bit width. Signed coefficient width ranges from 2 to 18 bits; unsigned from 2 to 17 bits.
DATA_UNSIGN	0-1	0	0: Signed input data 1: Unsigned input data
DATA_WIDTH	2-18	12	Data bit width. Signed data width ranges from 2 to 18 bits; unsigned from 2 to 17 bits.
SYSTOLIC	0-1	0	• 0: Transposed architecture • 1: Systolic architecture Valid when non-symmetric filter is being implemented (COEF_SYMM==1). Otherwise the Systolic architecture is enforced.
INP_REG	0-1	1	Disable (0) or enable (1) input registers. Enabling the registers helps improving the filter speed.
FPGA_FAMILY	19, 24, 25	19	The target FPGA family: SmartFusion2 (19), IGLOO2 (24) or RTG4 (25)
DIE_SIZE	5 - 50	20	Die size. The parameter is automatically derived from FPGA device name set through the Libero® project settings dialog. If the Libero device selection changes, the core configuration interface must be invoked and the core needs to be regenerated.

1. *N is a number of physically available MAC's.*

Parameter Name	Valid Range	Default	Description
COEF_RAM	0-1	0	Fully enumerated filter does not use the parameter
DATA_RAM	0-1	0	Fully enumerated filter does not use the parameter
SAMPLEID	0-1	0	Fully enumerated filter does not use the parameter
ID_WIDTH	1-10	5	Fully enumerated filter does not use the parameter
URAM_MAXDEPTH	4, 8, 16, 32, 64, 128, 256, 512	0	Fully enumerated filter does not use the parameter
L	2-512	2	Fully enumerated filter does not use the parameter
M	2-512	2	Fully enumerated filter does not use the parameter

Ports

The parallel FIR filter symbol is shown in Figure 12.

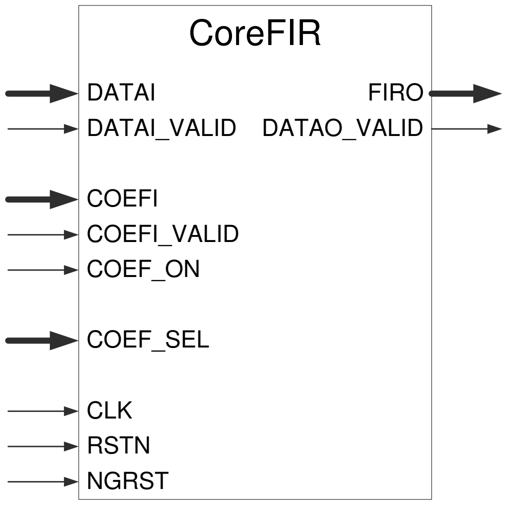

Figure 12 · Fully Enumerated Filter I/O Posts

The pinout of Figure 12 is a superset of all possible ports. In every configuration only a subset of these is used.

Table 7 Fully Enumerated Filter In/Out Signals

Signal	In/Out	Port Width Bits	Description
DATAI	In	Input data width, DATA_WIDTH	Input data to be filtered.
DATAI_VALID	In	1	Input data valid. Active high. When the signal is active, the input data sample is loaded into the FIR Filter.
COEFI	In	Coefficient bit width, COEF_WIDTH	Coefficient input. The coefficients are to be loaded sequentially, one by one. The coefficients are loaded in a temporary storage. They replace the current coefficients all at once on COEF_ON signal. This port is enabled only when reloadable coefficient mode is selected.
COEFI_VALID	In	1	Coefficient valid. Active high. When the signal is active, a coefficient is loaded into the MAC FIR filter. This port is enabled only when reloadable coefficient mode is selected.
COEF_ON	In	1	Coefficients on. Active high. Updates the filter coefficients. In Constant Coefficient mode (COEF_TYPE = 0), the signal replaces the existing filter coefficients with the new set of coefficients pointed by the input COEF_SEL. In Reloadable Coefficient mode (COEF_TYPE = 1), the signal makes the filter start using coefficients recently loaded in the auxiliary page.
COEF_SEL	In	4	Coefficient set selector. Identifies the pre-programmed fixed coefficient set to be activated and used as the filter coefficients. This port is enabled only when Multiple Coefficients mode is selected.
CLK	In	1	Clock. Rising edge active. The core master clock.
NGRST	In	1	Asynchronous reset. Active low. Resets all internal registers. The signal is expected to follow the FPGA power-on.
RSTN	In	1	Optional synchronous reset. Active low. Resets all internal registers.
FIRO	Out	DATA_WIDTH + COEF_WIDTH + ceiling($\log_2$TAPS)	Data output. The filtered data appear on this port. It is a full precision output. For example, at 12-bit data, 15-bit coefficients, and 150 taps, the output width = 12 + 15 + ceil(log2150) = 12 + 15 + 8 = 35 bits.
DATAO_VALID	Out	1	Output data valid. Active high. Indicates that a new output data sample is present at the FIRO port

Figure 13 and Figure 14 show examples of the core in Constant and Reloadable Coefficient modes. Figure 15 shows Multiple Coefficients mode.

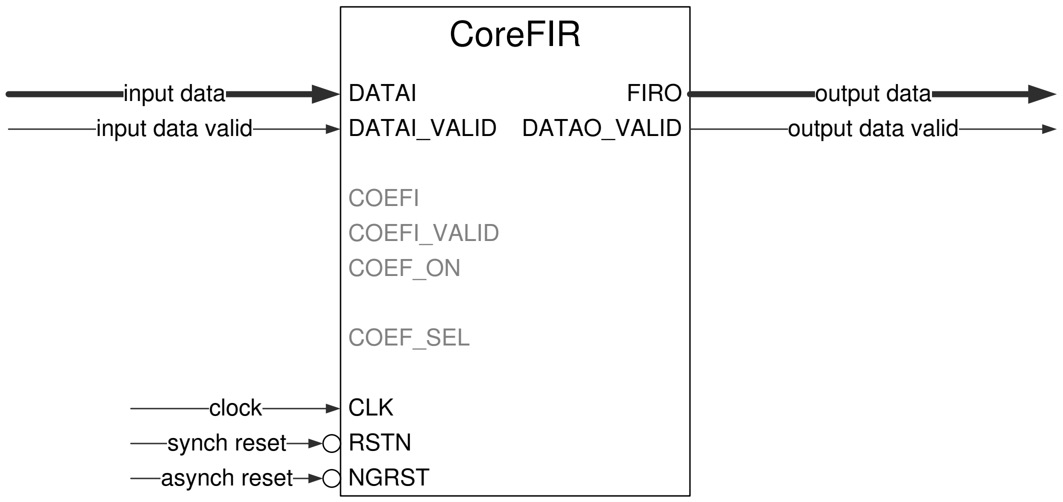

Figure 13 · Parallel Filter Constant Coefficient Configuration

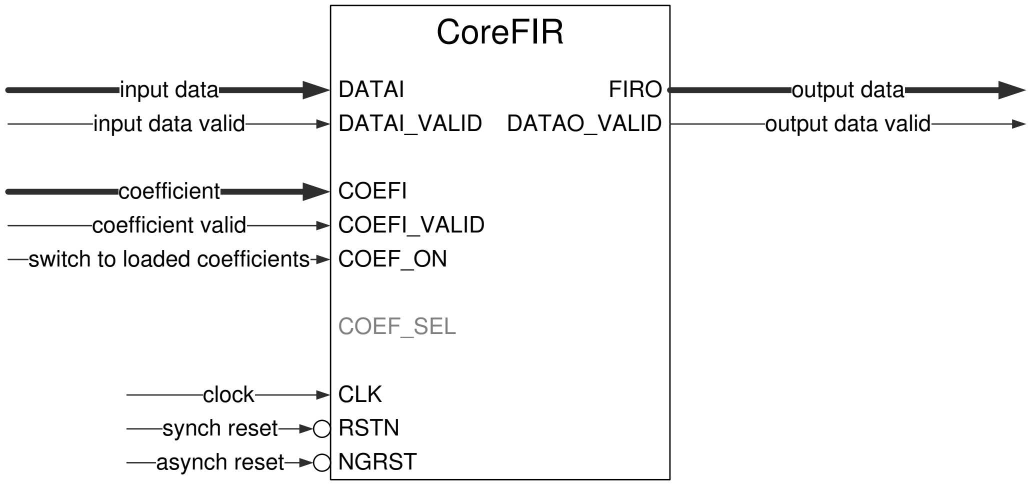

Figure 14 · Parallel Filter Reloadable Coefficient Configuration

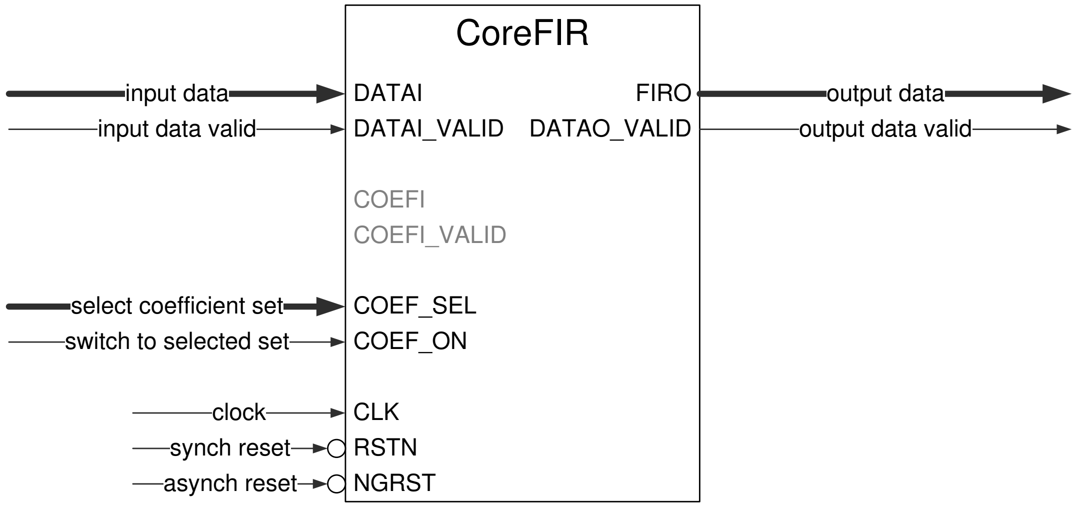

Figure 15 · Parallel Filter Multiple Coefficients Configuration

Fully Enumerated Filter Implementation Details

Data Path Bit Width

The core supports signed and unsigned data and coefficients. The core can be configured accordingly. The output data is unsigned if both input data and coefficients are unsigned. Otherwise the output data is signed. Input and output data, as well as the coefficients are integers in two's complement format.

The core supports signed data and coefficients of 2 to 18 bits. For the unsigned data and coefficients, the width is limited to 17 bits. With symmetric filter implementation, the maximum data width is reduced to 17 bits for signed data, and to 16 bits for unsigned data.

Internal filter processing takes place at full precision to reduce truncation/rounding noise and avoid risk of overflow. The filter output data are presented in full precision as well. If the data and coefficient bit widths are DATA_WIDTH and COEF_WIDTH, and the number of filter taps = TAPS, the full precision output bit width is given by EQ 6:

$$\text{Output Bit Width} = \text{DATA\_WIDTH} + \text{COEF\_WIDTH} + \text{ceiling}\ (\log_2 \text{TAPS})$$

EQ 6

Maximum Number of Taps

As the fully enumerated filter utilizes as many MACs as the number of taps, the maximum number of taps depends on the number of available mathblocks, which in turn is defined by your selection of a particular FPGA device. Symmetric and anti-symmetric filters can implement twice as many taps as the number of physically available MACs. Note, in the odd-symmetry filters, the maximum number of taps is one less. The core configuration window automatically limits the number of taps depending on the device selection.

Multiple Coefficients Mode

In this mode, the filter can switch between *k* pre-configured coefficient sets. Figure 16 shows a single filter tap in this mode. Other taps are organized and behave similarly. The COEF_SEL input controls a MUX, that is, selects one of the coefficient sets, but the coefficients are not propagated to the filter yet. This only takes place when the COEF_ON signal is issued, which loads the newly selected coefficients in the Pipeline registers.

To improve switching characteristics, issue the COEF_ON signal at least four clock cycles later after changing the COEF_SEL signal.

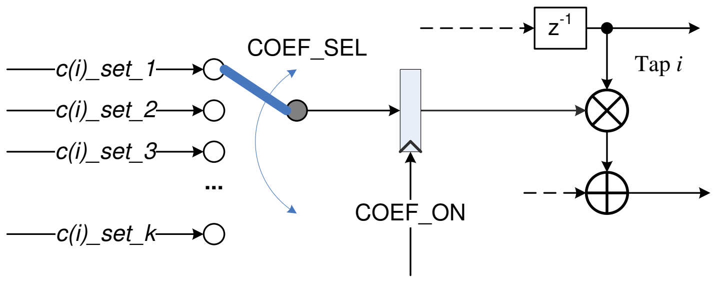

Figure 16 · A Parallel Filter Tap in Multiple Coefficients Mode

Input Registers

The core inputs that present extensive load for the input signal sources are optionally registered, so that the user circuitry does not face extensive fan-out. When the parameter INP_REG is set as 1, the core infers a pair of registers on the following inputs: DATAI, DATAI_VALID, COEFI, COEFI_VALID, COEF_SEL, and COEF_ON. Figure 17 shows an example of the input register inference. The pairs of registers are used to enable the synthesis tool (Synplify) to infer replicated register instances and to contain the user input fanout within the optimal limit. To achieve the best timing results, the global syn_replicate attribute of Synplify should be used.

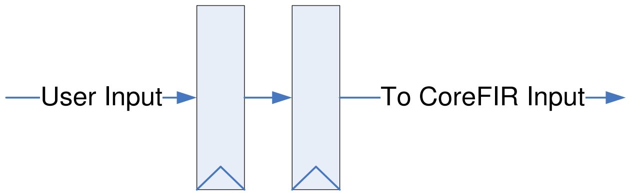

Figure 17 · Optional Input Registers

Inter-Column/Row Pipelines

CoreFIR implements several HW architectures, depending on the user configuration. All of them utilize the transposed architecture shown in Figure 6.

The hard MACs on a chip are organized into physical columns or rows. Within a column or row, the adder chain runs on a dedicated resource thus providing excellent performance characteristics. When a filter utilizes more than one hard MAC physical column or row, the long data path between the columns introduces an extended propagation delay. To eliminate this critical path, CoreFIR automatically infers optimal number of fabric pipeline registers in the inter-column or row sections of the adder chain. Simultaneously, it infers fabric registers in other data paths, which are necessary to preserve the correct functionality of the filter.

Figure 18 on page 23 presents an example of the two fabric inter-column or row registers in the adder chain balanced by a pair of the data bus registers. The added registers are shaded in Figure 18.

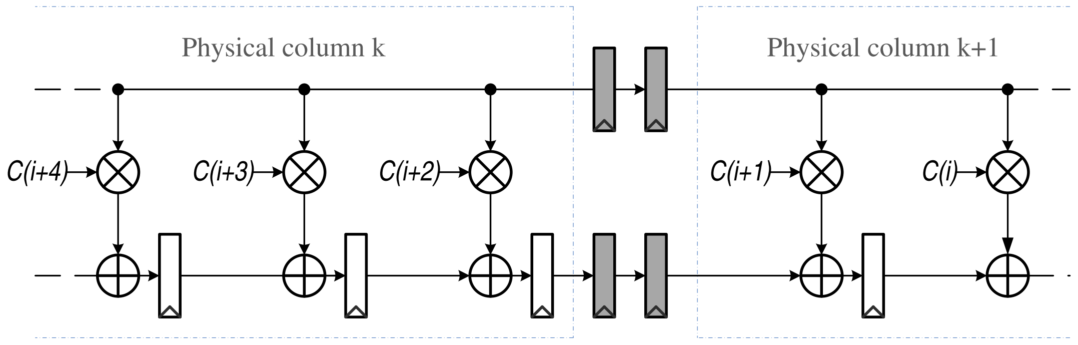

Figure 18 · Transposed Architecture with Inter-Column Fabric Registers

Fully Enumerated Filter Latencies

The filter imposes the following two latency types:

- Pipeline Latency
- Transition Latency, which is proportionate to the number of filter taps

The overall latency is a sum of these two latency types.

Pipeline Latency

This latency accounts for a time period between a valid input and valid output samples. Figure 19 shows the latency when the input registers are disabled. If these are enabled, the pipeline latency adds up to two clock cycles. The DATAO_VALID flag marks the valid output samples.

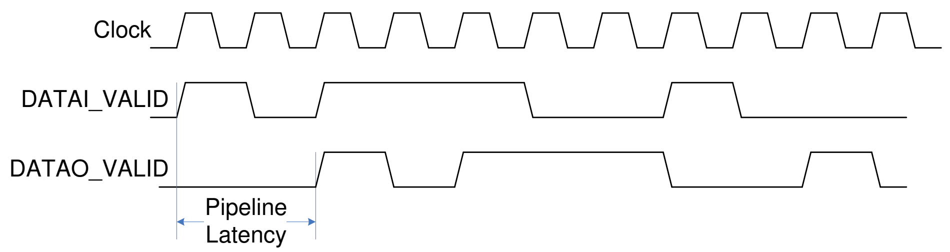

Figure 19 · Pipeline Latency

The flag is not of particular use, when the valid data to be filtered are coming at every clock period (Figure 20). It quickly becomes permanently active.

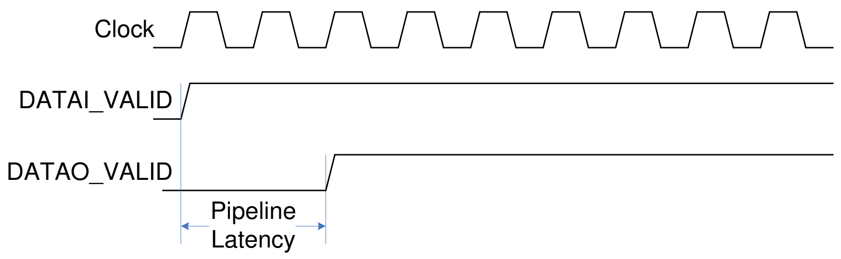

Figure 20 · Pipeline Latency with No Breaks in the Input Data

Transition Latency

The FIR filter starts producing valid output samples after its delay line is filled with input data samples. Until then, there is not enough data to be entered in EQ 1. In other words, after reset, when the filter delay line becomes empty, the first valid output will be available only when the filter receives N data samples. After that, every subsequent input sample will cause the filter to generate a fresh output sample. The reset is not the only event causing the transitional delay. The same applies to the filter coefficient modification or update that takes place on the COEF_ON signal.

Figure 21 shows an example of a 20-tap transposed filter. This filter must collect 20 input samples to satisfy EQ 1. The transition latency depends on the number of taps (TAPS), filter architecture, and symmetry.

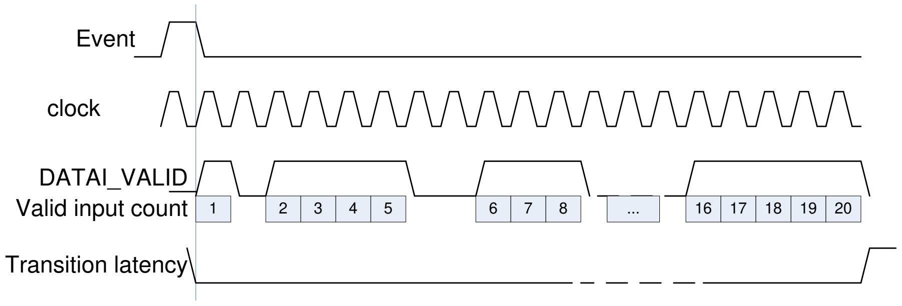

Figure 21 · Transition Latency of a 20-Tap Filter

CoreFIR generates the DATAO_VALID flag that accounts for the pipeline and transition latencies.

In many cases there is no need for the filtered sample recipient to know precisely when the valid filtered data starts. It is sufficient to know that after some initial warm-up time the filter generates the valid data. Table 8 reflects the maximum transition latency values expressed in number of valid input samples required to fill in the filter delay line.

Table 8 Maximum Transition Latency Values

Transposed	Systolic	Symmetric
TAPS + 12	2 * TAPS + 12	ceiling(1.5 * TAPS) + 12

If the precise transition latency is not of concern, it it might be convenient to let the filter run for the time indicated in Table 8 plus the pipeline latency. Every DATAO_VALID flag marks the valid filtered data sample.