LOGIC
DEVICES INCORPORATED

LF3320
Polyphase M:1 Downsampling

M:1 POLYPHASE DOWNSAMPLING

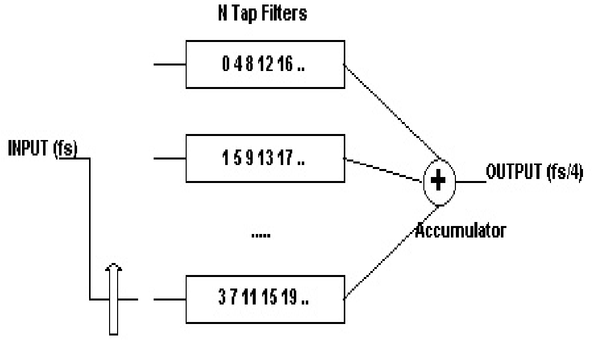
The objective of downsampling is a reduction in a signal’s sample rate. A reduction in sample rate results in a reduced bandwidth that the signal can reside within. Therefore, Bandlimiting filtering is required. Reducing the sample rate is sometimes accomplished by simply taking every Mth filter output sample. However, Polyphase downsampling allows us to effectively bandlimit using an expanded coefficient set AND reduce the sample rate efficiently within one LF3320. Note: 2 separate polyphase filters can be implemented within one LF3320 (in Dual Filter Mode).

When downsampling, the new Nyquist frequency (fs/2) is less than that of the original sample rate’s Nyquist frequency. One must ensure that no frequencies lie above the new fs/2 frequency in order to eliminate the possibility of aliasing. This necessitates a Bandlimiting filter with a cutoff frequency of fs/(2M). Therefore, 2:1 downsampling requires an fs/4 cutoff. Downsampling of 3:1 requires a fs/6 cutoff, etc. This Bandlimit filter is what we implement when performing polyphase filtering. Figure 2 illustrates the fs/8 filter we used for the 4:1 downsampling example in this note.

The Polyphase technique takes M phases of a relatively large filter coefficient set, and groups them into M separable filters – which are the M phases of the large coefficient set. Each unique filter processes the data in turn. Figure 1 illustrates the filter structure in our 4:1 polyphase downsampling implementation. Note the 4 phases.

The separate filters are actually implemented with our single FIR filter. Using multiple coefficient sets, we can make one hardware Filter appear as many more. In fact, the LF3320 has 256 coefficient sets – making it capable of performing downsampling of up to 256:1.

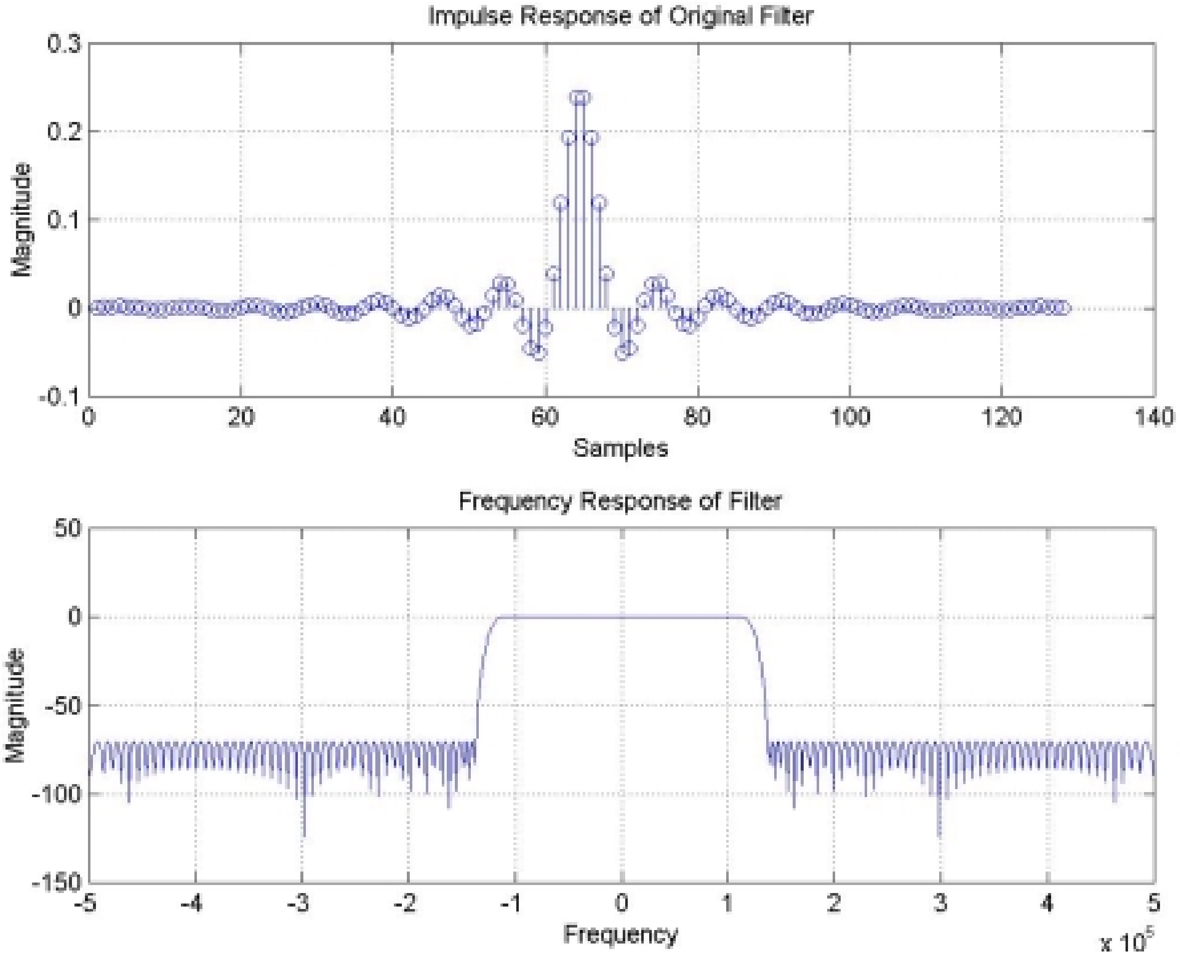
Figure 1. 4:1 Polyphase Implementation



Assuming we use the LF3320 in Single Filter mode to perform an M:1 downsampling filter, we ultimately implement a 32*M tap filter. The original filter is typically symmetric. Using the folded-back data path with M data samples reversed in sequence, along with accumulation, we can effectively implement a 2*16multipliers*M = 32*M tap filter. The value of this ‘Single Filter Mode’ implementation is in the relatively large filters that we can create.

We based this particular application note on 4:1 Downsampling. Therefore, our LF3320 will accommodate 32*4 = 128taps. Since we are taking advantage of symmetry in this 128tap filter – we will deal with only the first 64taps. We will take these 64 taps and break them into 4 groups (M=4) of 16 taps each. These 4 filter groups are the 4 phases of our original 128tap filter.

Figure 2. Downsampling Bandlimit Filter Coefficient Set and Frequency Response



We break the large coefficient set into 4 phases by taking coefficients 0, 4, 8, etc and placing them into the first phase. The second phase consists of coefficients 1, 5, 9, etc. Note: Filter A coefficient banks 0 through 7 receive the first 8 taps of the 16tap filter phase, and Filter B coefficient banks 0 through 7 receive the last 8 taps of the 16tap filter phase.

We load the 4 filter sets into 4 banks of our coefficient storage. As data is continuously streamed through the LF3320 at the original Fs CLK rate, we cycle through a new coefficient set every CLK cycle.

In 4 CLK cycles, we present the 4 phases of the coefficient set to the data, and accumulate all of the 4 filter results. So in 4 CLK cycles, we effectively implement the original 64tap filter and provide 1 output sample. This sequence of events continues indefinitely. (The internal accumulator sums the results of M cycles of filtering and provides an accumulated sum every M CLK cycles.)

The Interleave/Decimate (ID) registers need to be set to a length of M. This sets a distance between the data samples so that at any instance in time, the data and coefficients from a particular phase are aligned. We set Length to 4 for this example.

Tables 1 and 2 describe the appropriate configuration register and coefficient settings for this 4:1 downsampling example.

Toggling the TXFR pin reverses M samples of delayed input data on their way to the reverse ‘fold-back’ data path. This reversal of data properly aligns the data with their respective coefficient phase.

The timing and toggling sequence of CA (coefficient address), ACC (accumulator control), and TXFR (transfer control) is extremely important. Table 3 and Figure 3 illustrate the correct timing sequence for our 4:1 downsampling factor.

NOTE: Tie SHENA/SHENB LOW. Also, tie Filter A and Filter B common control pins together (CAA/CAB, TXFRA/TXFRB, ACCA/ACCB, etc..)

Table 1. Configuration Loading

CFA	ADDRESS	400
	DATA	00D
CFA	ADDRESS	800
	DATA	000
	DATA	000
	DATA	000
	DATA	000
CFA	ADDRESS	C00
	DATA	001
	DATA	0F8
	DATA	0FF
	DATA	007
CFA	ADDRESS	200
	DATA	006
CFA	ADDRESS	201
	DATA	006
CFA	ADDRESS	202
	DATA	006
CFA	ADDRESS	203
	DATA	006
CFA	ADDRESS	204
	DATA	000
CFA	ADDRESS	205
	DATA	000

Table 2. Coefficient Loading

CFA	ADDRESS	000	CFB	ADDRESS	000
	DATA	004		DATA	FF4
	DATA	008		DATA	FF6
	DATA	00D		DATA	FFD
	DATA	010		DATA	008
	DATA	00F		DATA	00F
	DATA	00A		DATA	00E
	DATA	001		DATA	005
	DATA	FF9		DATA	FF7
CFA	ADDRESS	001	CFB	ADDRESS	001
	DATA	008		DATA	FC7
	DATA	00A		DATA	049
	DATA	FF6		DATA	FA3
	DATA	00E		DATA	079
	DATA	FED		DATA	F5E
	DATA	01A		DATA	0E7
	DATA	FDE		DATA	E87
	DATA	02C		DATA	3CF
CFA	ADDRESS	002	CFB	ADDRESS	002
	DATA	00D		DATA	FD9
	DATA	001		DATA	036
	DATA	FFD		DATA	FB6
	DATA	005		DATA	067
	DATA	FF8		DATA	F6C
	DATA	00D		DATA	0E3
	DATA	FEC		DATA	E60
	DATA	01C		DATA	635
CFA	ADDRESS	003	CFB	ADDRESS	003
	DATA	010		DATA	004
	DATA	FF9		DATA	001
	DATA	008		DATA	FF8
	DATA	FF7		DATA	013
	DATA	009		DATA	FD9
	DATA	FF7		DATA	04E
	DATA	009		DATA	F49
	DATA	FF9		DATA	7A7

Table 3. Control Sequence

DIN	SHENA/B	TXFRA/B	CA/B	ACCA/B	DOUT*
DIN(n)	0	0	0	1	XX
DIN(n+1)	0	1	3	1	DOUT(n)
DIN(n+2)	0	1	2	1	DOUT(n)
DIN(n+3)	0	1	1	0	DOUT(n)
DIN(n+4)	0	0	0	1	DOUT(n)
DIN(n+5)	0	1	3	1	DOUT(n+1)
DIN(n+6)	0	1	2	1	DOUT(n+1)
DIN(n+7)	0	1	1	0	DOUT(n+1)
	0	0	0	1	DOUT(n+1)
	.	.	.	.	.

NOTE: As previously stated, the first phase (coefficients 0, 4, 8, etc. of the original set) are loaded into Address 0 in the coefficient memory. The second phase (coefficients 1, 5, 9, etc. of the original set) are loaded into Address 1 in the coefficient memory. This is done for all 4 phases of the original coefficient set. The coefficient Address (CA) in Table 3 refers to addressing a certain phase of the original coefficient set. For example, CA=3 addresses the final phase of the original set, which are coefficients 3, 7, 11, etc.

Bit Weighting and Coefficient Scaling

Since the maximum coefficient value for the downsampling polyphase filter is just less than 0.25, we have scaled the coefficients up by 2^2 to fit as much precision within our 12bit coefficient storage as possible. For example, the coefficient 0.2391 will be represented by the 12bit hex value of 7A7h. So a gain of 2^-2 (shift of 2) must be applied to the filter output data. We will perform this shift in the RSL section.

Since there are 11 bits below the radix point in our coefficients, we would typically load an 11 (B hex) into the Select register. The required coefficient scaling of 2^-2 adds 2 shifts to the 11, and we end up with a shift of 11+2=13 (D hex). We therefore load 13(D hex) into Address 400h (Filter A Select Register).

Figure 3. Control Waveforms