

## Introduction

Many signal processing applications require a filter which can extract a narrow signal band from a wide band input. Satellite terminals, transmultiplexers, ultrasound equipment and spectrum analyzers are just a few examples. Traditionally this operation has been performed using analog electronics in the receiver front end which down converts the signal of interest to base band (0Hz) and filters out all other signals. The result is digitized by an A/D Converter operating at a low sample rate. The digitized data drives a DSP microprocessor, which extracts the information from the resulting signal. In this scenario, however, the maximum benefit of DSP is not being realized, since the nonlinear phase, drift and other problems of the analog circuitry are still present. Minimizing the amount of analog electronics by replacing them with DSP hardware alleviates these problems, but is not practical for all applications due to the throughput limitation of the microprocessor. This problem often cannot be solved merely by inserting a high speed A/D converter into the signal processing chain and replacing the downstream analog electronics with their digital counterparts. To extract a narrow band signal with such a brute force solution would often require an FIR filter with a thousand coefficients or more. A much better solution is to use several cascaded digital filter stages such that each stage partially filters and decimates the data, thus reducing the data rate and the subsequent processing speed required by each succeeding stage.

The Intersil HSP43220 is a two stage Decimating Digital filter or OOF. It is designed for applications requiring a high input sample rate and a low output rate (see Figure 1). The first stage performs the initial filtering and reduces the sample rate (decimates) by a factor of up to 1024. For this reason, it is called the High Order Decimation Filter, or HDF. The second stage is a sum of products FIR filter which provides the final spectral shaping and some additional decimation. The combination of the two results in a very narrow band low pass filter. A more complete description of the part can be found in the data sheet.

While the DDF is very efficient in this application, some difficulty arises when the application requires knowledge of the internal operation of the filter. The HDF is implemented using a set of registers with a "one out of N counter" controlling the Decimation Register. The FIR is implemented using dual port RAM for the data and coefficient memory. This architecture creates two subtle issues related to the relative start-up times of the state machines that drive the HDF and FIR. The first relates to the number of input samples required to produce an output sample, and the second relates to the input to output delay in the steady state condition. This Application Note will treat only the first case; DECIMATE™ can be used to resolve these issues on an experimental basis. This Application Note will allow the user to calculate the necessary pipeline delays on an empirical basis.

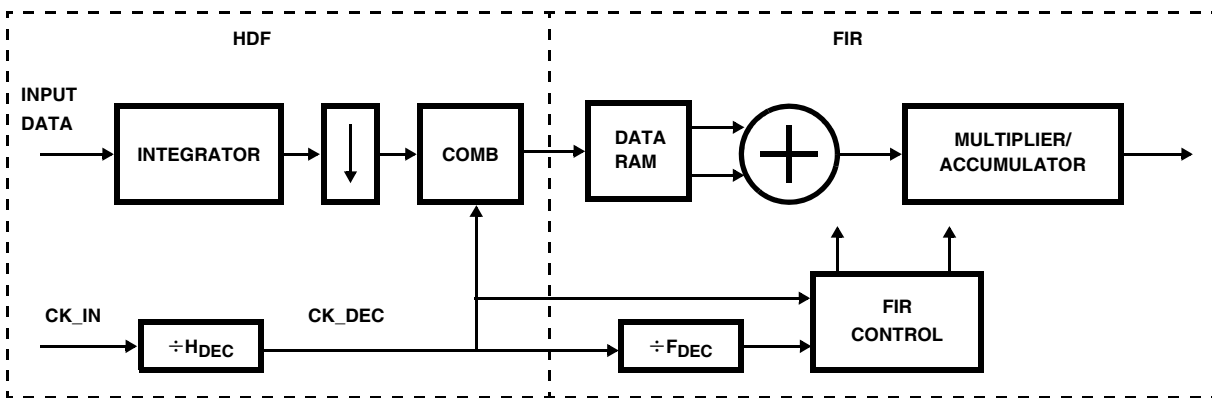


FIGURE 1. BLOCK DIAGRAM OF HSP43220 DECIMATING DIGITAL FILTER

This Application Note provides a way to predict:

1. The number of “invalid” (aka “transient”) data points that will emerge from the DDF (in “DATA\_RDY” cycles) from the time the DDF is started and “valid” data points are applied to the input (in “CK\_IN” cycles), for a given configuration of the device.
2. The minimum number of data points required at the input of the DDF (in “CK\_IN” cycles) in order to result in a desired number of “valid” data points out of the DDF (in “DATA\_RDY” cycles) for a given configuration of the device.
3. For an application where sampled data is “windowed” to the DDF, the exact number of input data points (in “CK\_IN” cycles) required in order to generate a required number of data points out of the DDF (in “DATA\_RDY” cycles) and to keep both decimation counters aligned (in phase) from one window to the next.

NOTE: In order to get DECIMATE to compile **ANY** filter specification, the ratio of the Input Sample Rate to the Output Rate **MUST** be an integer in the range of 1 to 16384 (the total decimation capability of the device). In this example the Input Rate is entered as 2000004Hz and the Output Rate as 166667Hz in order to satisfy this rule. ( $2000004/166667 = 12 = H_{DEC} \times F_{DEC}$ ). In order to determine the pipeline delays through the HSP43220 only four parameters need to be considered:

1. The HDF decimation factor ( $H_{DEC}$ ).
2. The FIR decimation factor ( $F_{DEC}$ ).
3. The number of taps that the FIR is programmed to implement.
4. The ratio of CK\_IN to FIR\_CK.

## Calculating The Number Of Transient Data Points

The DDF will output transient data until the following two conditions are met: (1) the HDF Section pipelines are filled, resulting in the HDF outputting “valid” data points to the FIR; (2) all taps in the FIR are filled with “valid” data. See Table 1 and Figure 2.

In the HDF the input data must first propagate through the Data Input Register and the five registers in the HDF Integrator Section (which takes six input points). It then must propagate through both the HDF Decimation Register and the HDF Output Register (which takes  $2 \times H_{DEC}$  more input points, since these two registers are clocked at the HDF decimated rate via “CK\_DEC”) before it enters the FIR Section. The First “valid” data point, therefore, doesn’t enter the FIR’s tapped pipeline until the  $[6 + 2(H_{DEC})]$  input point is applied.

Note that the five Comb Filter Registers don’t contribute to the HDF pipeline delay since they always contain a “prior” value, which is subtracted from a “current” value. This difference always “falls through” the comb filter after each “CK\_DEC” cycle.

## Caveat 1

- All examples in this Application Note assume that the DDF has already been put in “operational mode”, that is, the device has previously been “started” by the assertion of “STARTIN” in conjunction with **two** “CK\_IN” pulses according to the Start Timing Diagram in the Timing Waveforms Section of the HSP43220 Data Sheet.
- Important Point **the DDF does not process any data until the 3rd “CK\_IN” after “STARTIN”**.

Reference to the HDF Timing Diagram of Figure 2 may be helpful during the following discussions:

Since the FIR filter is implemented using a conventional “sum-of-products” architecture, the first “valid” output point will emerge from the DDF **only after** all 91 taps in the FIR pipeline are filled with “valid” input points **and** the FIR runs its calculation. Note that the FIR Section performs decimation by only running the “sum-of-products” calculation when required. In this example, the FIR only runs after every sixth point ( $F_{DEC} = 6$ ) that is passed to it from the HDF Section. It is important to remember, however, that **every** point that emerges from the HDF Section enters the FIRs tapped pipeline and can contribute to the result.

Given all this, then, it is a simple matter to calculate when the 1st “valid” point will emerge from the DDF. It will take  $(TAPS) \times (H_{DEC})$  input points to completely fill the FIR with “valid” data. We also determined that the 1st “valid” point doesn’t enter the FIR until the  $[6 + 2(H_{DEC})]$  input point is applied and that the FIR only runs its calculation every  $(H_{DEC}) \times (F_{DEC})$  input points. Therefore, the number of transient data points out of the DDF is given by:

$$DATA_{TRAN} = \left\lceil \frac{(TAPS \times H_{DEC}) + 6 + (2 \times H_{DEC})}{H_{DEC} \times F_{DEC}} \right\rceil$$

where  $\lceil x \rceil$  means round  $x$  up to the next integer. It follows that the first valid data point is  $DATA_{TRAN} + 1$ . Using our example, the first “valid” point out of the DDF is:

$$DATA_{TRAN} + 1 = \left\lceil \frac{(91 \times 2) + 6 + (2 \times 2)}{2 \times 6} \right\rceil + 1 = 17$$

and the previous 16 are the “invalid” or “transient” points that occur as the FIRs tapped pipeline is purged of data that was resident prior to the “valid” data points.

Predicting Output Data

TABLE 1. EXAMPLE OF A DECIMATE™ FILTER DESIGN SCREEN

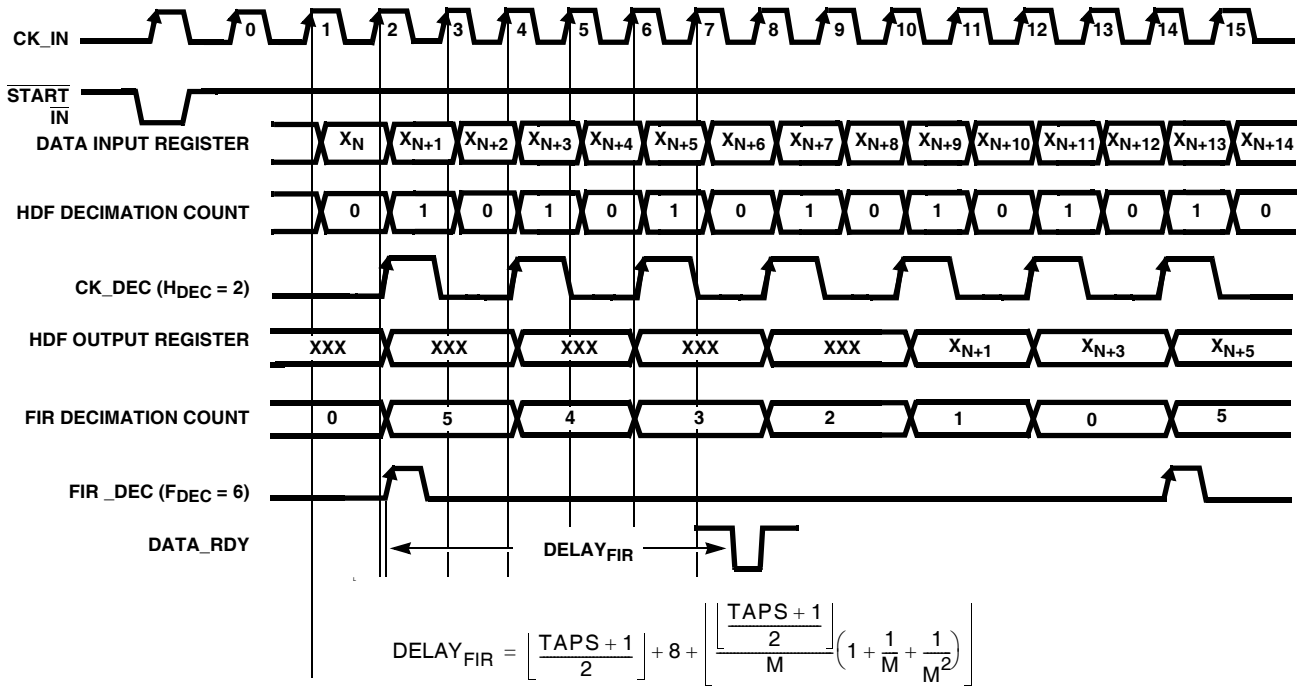
HSP 43220 DDF FILTER SPECIFICATION			
Filter File	41K666.DDF		
Input Sample Rate	2MHz	Design Mode	Manual
Output Rate	166.67kHz	Generate Report	Yes
Passband	41.666kHz	Display Response	Log
Transition Band	41.666kHz	Save Frequency Responses	No
Passband Atten	0.20dB	Save FIR Response	No
Stop Band Atten	80dB		
HDF Decimation	2		
HDF Order	3		
FIR Type	PRECOMP		
		FIR Input Rate	1MHz
		FIR Clock (Min)	10MHz
HDF Scale Factor	1	FIR Order	91
		FIR Decimation	6

Calculating the Number of Input Points Required

The number of input points required to generate a given number of output points is a function of two variables: (1) the number of input points required to fill the HDF and FIR pipelines which simultaneously purges the pipeline of any “invalid” or “transient” data; (2) the subsequent number of points required to generate N output samples. There are two types of systems to consider, which will be called type A and type B.

For type A systems, CK\_IN stops after the minimum number of inputs is given and FIR\_CK continues to run. The number of input points is given as POINTS, which is defined below. The position of the final output sample relative to the last input sample that resulted in the FIR running the calculation can also be determined using the equation for DELAY\_FIR given below.

For type B systems, CK\_IN and FIR\_CK run continuously; the number of input points is calculated by the number of input samples required to satisfy condition 1 above and the number of inputs that are incidentally input during FIR calculation. It is assumed that an answer given entirely in CK\_IN cycles would be of more use than an answer comprised of CK\_IN and FIR\_CK cycles.



WHERE: “DELAY<sub>FIR</sub>” IS MEASURED IN FIR\_CK CYCLES  
 “TAPS” IS THE NUMBER OF TAPS IN THE FIR SECTION  
 “M” IS THE RATIO:  $\frac{FIR\_CK}{CK\_IN} H_{DEC}$   
 “H<sub>DEC</sub>” IS THE HDF DECIMATION RATE  
 [X] INDICATES THAT X IS ROUNDED DOWN TO THE NEXT INTEGER

FIGURE 2. HDF TIMING DIAGRAM

In the example referring to the HDF Timing Diagram of Figure 2, the very first “valid” input point doesn’t enter the FIR’s tapped pipeline until the **10th** input point is clocked in ( $10 = 6 + 2 \times H_{DEC}$ ). The 10th “CK\_IN” cycle, however, does not coincide with the FIR running a calculation since the FIR’s decimation counter is still one count away from its terminal count.

Remember that this counter began its first decimation cycle following the **2nd** input point and that it is being clocked at the HDF decimation rate by “CK\_DEC”. Therefore, the FIR decimation counter will reach its terminal count after 12 more input points are applied. The 2nd FIR calculation cycle will, therefore, occur after the **14th** input point is applied. The Timing Diagram of Figure 2 illustrates this scenario. Since the total decimation of the DDF, in this example, is  $H_{DEC} \times F_{DEC} = 2 \times 6 = 12$ , an FIR calculation cycle will begin following every 12 input points. Thus, in this example, The DDF produces output points after input points 2, 14, 26, 38 and so on.

For type A systems a formula for determining the minimum number of input points required to produce a desired number of output points is given in the following equation:

$$POINTS = (N-1)(H_{DEC})(F_{DEC}) + H_{DEC}$$

where POINTS represents the total number of input points (in CK\_IN cycles) required in order to produce N output points (in DATA\_RDY cycles) from the DDF.

The delay from the last input sample to the final output sample in FIR\_CKs is:

$$DELAY_{FIR} = \left\lfloor \frac{TAPS + 1}{2} \right\rfloor + 8 +$$

$$\left\lceil \left\lfloor \frac{TAPS + 1}{M} \right\rfloor \times \left( 1 + \frac{1}{M} + \frac{1}{M^2} \right) \right\rceil$$

where:

$$M = \frac{FIR\_CK}{CK\_IN} \times H_{DEC}$$

For type B systems the total number of input samples required for N outputs is:

$$SAM_{TOTAL} = POINTS + DELAY_{FIR} \times (CK\_IN/FIR\_CK)$$

The term in parentheses converts  $DELAY_{FIR}$  from FIR\_CK to CK\_IN cycles.

It is now possible to compute the minimum number of input points necessary to apply to the DDF in order to get a desired number of “valid” output points out of it. The final goal in this example is to apply a 1024 point FFT to the filtered data in order to determine the spectral content of the applied signal. Obviously, any “invalid” output points contained in the 1024 point transformation must be eliminated, since they will contaminate the results.

Therefore, in order to get exactly 1024 “valid” output points, the exact number of input points must be applied to the DDF in order to “purge” the “invalid” points **and** produce the 1024

desired points. Note that the 16 “invalid” points must still be stripped off so that they don’t make it into the FFT.

Using the previous example, then, the number of “invalid” output points is given by:

$$DATA_{TRAN} = \left\lceil \frac{(TAPS \times H_{DEC}) + 6 + (2 \times H_{DEC})}{H_{DEC} \times F_{DEC}} \right\rceil =$$

$$\left\lceil \frac{(91 \times 2) + 6 + (2 \times 2)}{2 \times 6} \right\rceil = 16$$

Adding the desired number of “valid” points to this yields:

$$16 + 1024 = 1040.$$

The minimum number of input points required to produce 1040 output points is:

$$POINTS = (n - 1) \times (H_{DEC}) \times (F_{DEC}) + H_{DEC} =$$

$$(1040 - 1) \times (2) \times (6) + 2 = 12,470$$

In type A systems the delay from the last input to the final output is:

$$DELAY = \left\lfloor \frac{(TAPS + 1)}{2} \right\rfloor + 8 +$$

$$\left\lceil \left\lfloor \frac{(TAPS + 1)}{M} \right\rfloor \left( 1 + \frac{1}{M} + \frac{1}{M^2} \right) \right\rceil$$

$$DELAY = \left\lfloor \frac{(91 + 1)}{2} \right\rfloor + 8 +$$

$$\left\lceil \left\lfloor \frac{(91 + 1)}{\left(\frac{10}{2} \times 2\right)} \right\rfloor \left( 1 + \frac{1}{\left(\frac{10}{2} \times 2\right)} + \frac{1}{\left(\frac{10}{2} \times 2\right)^2} \right) \right\rceil$$

$$DELAY = 46 + 8 + \left\lfloor 4.6(1.11) \right\rfloor = 59 \text{ FIR\_CKs}$$

For a type B system the number of input samples is:

$$SAM_{TOTAL} = POINTS + DELAY_{FIR} \times (CK\_IN/FIR\_CK) =$$

$$\left\lceil 12,470 + 59 \times (2 \div 10) \right\rceil = 12,482 \text{ CK\_INs}$$

### Exact Input Samples

The data acquisition scenario can now be expanded to include a series of “data sampling windows” for type A systems. When a “data sampling window” is “opened”, “CK\_IN” cycles, along with “valid” data are applied to the DDF. The idea is to “open” each “window” such that it encompasses the desired portion of the applied signal, for the required amount of time to generate the appropriate number of “valid” output points for the subsequent FFT. In practice, one quickly discovers that the “window” must be extended, slightly, in order to allow both decimation counters to “realign”, such that they are always in the same state as each window opens. The consequence of not realigning the

## Application Note 9403

decimation counters in this situation is that the total number of output points produced by each window will vary by one, from window to window, as the position of the windows “walk” with respect to the state of the two counters. To solve this problem, it is necessary to increase the number of applied input points to be evenly divisible by the total decimation rate ( $H_{DEC}$ )  $\times$  ( $F_{DEC}$ ).

Using the same example, it was previously determined that it requires 12,470 input points to produce exactly 1024 “valid” output points. In order to keep both decimation counters aligned from window to window, simply increase this number until it is evenly divisible by  $(H_{DEC})(F_{DEC}) = (2)(6) = 12$ .

$$\lceil 12,470 \div 12 \rceil = \lceil 1039.17 \rceil = 1040$$

The new number of input points =  $1040 \times 12 = 12,480$

Note that, again, it is still the responsibility of the downstream hardware or software to “strip off” the 16 “invalid” points so that they don't make it into the FFT.

### Examples

Several examples of various configurations are shown below, along with the relevant equations used to calculate  $DELAY_{FIR}$ :

$$\begin{array}{ll} \#TAPS = 91 & H_{DEC} = 1 \\ FIR\_CK = 20M & CK\_IN = 2M \end{array}$$

$$\begin{aligned} DELAY_{FIR} &= \left\lfloor \frac{91+1}{2} \right\rfloor + 8 + \left\lfloor \frac{\left\lfloor \frac{91+1}{2} \right\rfloor}{10} \right\rfloor \left( 1 + \frac{1}{10} + \frac{1}{100} \right) \\ &= 46 + 8 + 5 \Rightarrow 59 \left( \frac{1}{20m} \right) = 2.95 \mu s \end{aligned}$$

#### EXAMPLE 1.

$$\begin{array}{ll} \#TAPS = 119 & H_{DEC} = 1 \\ FIR\_CK = 20M & CK\_IN = 2M \end{array}$$

$$\begin{aligned} DELAY_{FIR} &= \left\lfloor \frac{119+1}{2} \right\rfloor + 8 + \left\lfloor \frac{\left\lfloor \frac{119+1}{2} \right\rfloor}{10} \right\rfloor \left( 1 + \frac{1}{10} + \frac{1}{100} \right) \\ &= 60 + 8 + 6 \Rightarrow 74 \left( \frac{1}{20m} \right) = 3.70 \mu s \end{aligned}$$

#### EXAMPLE 2.

$$\begin{array}{ll} \#TAPS = 79 & H_{DEC} = 2 \\ FIR\_CK = 20M & CK\_IN = 2M \end{array}$$

$$\begin{aligned} DELAY_{FIR} &= \left\lfloor \frac{79+1}{2} \right\rfloor + 8 + \left\lfloor \frac{\left\lfloor \frac{79+1}{2} \right\rfloor}{20} \right\rfloor \left( 1 + \frac{1}{20} + \frac{1}{400} \right) \\ &= 40 + 8 + 2 \Rightarrow 50 \left( \frac{1}{20m} \right) = 2.5 \mu s \end{aligned}$$

#### EXAMPLE 2.

$$\begin{array}{ll} \#TAPS = 91 & H_{DEC} = 4 \\ FIR\_CK = 20M & CK\_IN = 2M \end{array}$$

$$\begin{aligned} DELAY_{FIR} &= \left\lfloor \frac{91+1}{2} \right\rfloor + 8 + \left\lfloor \frac{\left\lfloor \frac{91+1}{2} \right\rfloor}{40} \right\rfloor \left( 1 + \frac{1}{40} + \frac{1}{1600} \right) \\ &= 46 + 8 + 1 \Rightarrow 55 \left( \frac{1}{20m} \right) = 2.75 \mu s \end{aligned}$$

#### EXAMPLE 4.

### References

For Intersil documents available on the internet, see web site [www.Intersil.com/](http://www.Intersil.com/)

- [1] Hogenauer, E.B., “An Economical Class of Digital Filter for Decimation and Interpolation,” IEEE Trans. on ASSP, Vol. ASSP-29, No. 2, pp 155-162, April 1981.
- [2] HSP43220 *Data Sheet* of Intersil Corporation.
- [3] Riley, C.A., et. al., “High Decimation Digital Filters,” Proceedings of ICASSP 91, Vol. 3, pp. 1613-1616, Toronto, May, 1991.
- [4] Sabin, W.E. and Shoenike, E.O., “Single-Sideband Systems and Circuits,” McGraw-Hill, New York, 1987, Chapter 7.