Welcome to this tutorial on the Enhanced Filter Co-Processor (EFCOP). The EFCOP provides powerful co-processing of complex filtering tasks, tasks that run concurrently with core processing and minimize core overhead.

In this tutorial, we’ll review the basic features of EFCOP, including its functional blocks and memory organization. We’ll also review the different EFCOP filter modes and take a detailed look at processing of the two main communication filter types: the FIR filter and the IIR filter.
Let’s begin with a basic overview of the EFCOP. The EFCOP provides a cost-effective and power-efficient co-processor to accelerate filtering tasks concurrently with software running on the DSP core. Examples include echo cancellation and correlation. Several modes of operation, modified by scaling, decimation, and multi-channel options, offer the programmer considerable flexibility.

The EFCOP interfaces with the DSP core via the peripheral module bus.

The DSP core and EFCOP share data via X and Y data memory. For the DSP56311, the shared memory size is 10K words.

DMA channels can also be used to transfer input and output data between the DSP core and the EFCOP.
Here, we see a functional view of the EFCOP. Basic operations are determined by the bit settings in the Control Status Register, FCSR. You can modify the operating mode using the bits in the EFCOP Decimation Channel Count Register (FDCH) and the EFCOP ALU Control Register (FACR).

The Peripheral Module Bus (PMB) interface provides control and status registers, buffering of the internal bus, address decoding and generation, and control of the handshake signals required for DMA and interrupt operations. This block also generates interrupt and DMA trigger signals whenever data transfer is required.

The Filter Data Memory (FDM) bank is X memory that stores data taps for filter processing by the EFCOP. The data input FIFO (FDIR) writes to it, and the EFCOP address generation logic generates its addresses. The data taps are read sequentially from the FDM into the MAC. The FDM is shared with the lowest locations of the on-chip internal X memory. For the DSP56L307, this is the 4K lowest locations and for the DSP56311, this is the 10K lowest memory locations.

The Filter Multiplier and Accumulator (FMAC) machine performs a 24-bit x 24-bit multiplication with accumulation in a 56-bit accumulator. The FMAC operates a pipeline: the multiplication is performed in one clock cycle, and the accumulation occurs in the following clock cycle.

Throughput is one MAC result per clock cycle. The two MAC operands are read from the Filter Data Memory and from the Filter Coefficient Memory. The full 56-bit width of the accumulator stores intermediate results during the filter calculations. When operating in sixteen-bit arithmetic mode, the FMAC performs a 16-bit x 16-bit multiplication with accumulation into a 40-bit accumulator. The result from the FMAC is stored in the EFCOP output buffer, FDOR.

The Filter Coefficient Memory (FCM) bank is Y memory that stores filter coefficients for filter processing by the EFCOP. The DSP56300 core writes to it, and the EFCOP address generation logic generates its addresses. The filter coefficients are read sequentially from the FCM into the MAC. Note that the core only has write access to the FCM. The FCM is shared with the lowest locations of the on-chip internal Y memory. In the DSP56L307, this is the 4K lowest locations.
Here is a detailed view of EFCOP memory access. On-chip memory for the DSP56311 totals 128K of 24-bit words. Program memory can be configured with up to 96K, and X and Y data memory are configured with up to 48K each within the total 128K available. Off-chip memory can total up to 4MB for each type of memory.

You configure the memory by setting control bits in the core's operating mode register (OMR). In any configuration, the lowest 10K words of both X and Y data memory are shared by the EFCOP and the core and are unavailable to the DMA controller. Note that simultaneous access by the core and the EFCOP to a memory module block is not permitted. It is the user’s responsibility to prevent such simultaneous access.
The EFCOP supports two basic filter types used in communication algorithms, the Finite Impulse Response (FIR) filter and the Infinite Impulse Response (IIR) filter.

The FIR is a non-recursive digital filter that doesn’t contain poles. A main advantage of the FIR filter is that it is inherently stable. Also, it filters with a precise linear phase characteristic that can be designed and implemented.

The Infinite Impulse Response (IIR) filter is a recursive digital filter that contains both poles and zeros. Often, a frequency response can be approximated using an IIR filter with considerably less computational complexity.
Filter Modes

• FIR Filters
  • FIR machine with real taps
  • FIR machine with complex taps
  • Complex FIR machine generating pure real/imaginary outputs alternately
  • Magnitude (calculate the square of each input sample)
  • A 4-bit decimation factor in FIR filters
    • Provides a decimation ratio up to 16
  • Adaptive FIR filter with true least mean square (LMS) coefficient updates
  • Adaptive FIR filter with delayed LMS coefficient

• IIR Filter Options
  • Direct form 1 (DFI) Infinite Impulse Response (IIR) filter
  • Direct form 2 (DFII) IIR filter
  • Three scaling factors (1, 8, 16) for IIR output

As shown here, the EFCOP supports a variety of filter modes, some of which are optimized for cellular base station applications. In base transceiver stations, EFCOP can perform complex matched filtering to maximize the signal-to-noise ratio within an equalizer. In a transcoder base station or mobile switching center, the EFCOP can perform all types of FIR and IIR filtering within a vocoder, as well as LMS-type echo cancellation.

The EFCOP supports up to 4K taps and 4K coefficients in any combination of number and length of filters; for example, 8 filters of length 512, or 16 filters of length 256. It can perform either 24-bit or 16-bit precision arithmetic with full support for saturation arithmetic.
The EFCOP is very easy to use. This table lists the registers that configure and operate the EFCOP.

Three registers support data transfers.
- The FDIR is a 4-word deep, 24-bit wide FIFO used for DSP-to-EFCOP data transfers.
- The FKIR is a 24-bit write-only register used for DSP-to-EFCOP data transfers in adaptive mode.
- The FDOR is a 4-bit read-only register used for EFCOP-to-DSP data transfers.

The FCNT is a read/write register that selects the filter length.

The DSP56300 core uses the FCSR and the FACR to control and monitor EFCOP operations. The core uses the FCSR, a 24-bit read/write register, to control the main operation modes of the EFCOP and monitors its status. It uses the FACR, a read/write register, to control the main operation modes of the EFCOP ALU.

The FDBA is a 16-bit read/write register used as an address pointer to the EFCOP FDM bank. The FCBA is a 16-bit read/write counter register used as an address pointer to the EFCOP FCM bank.

The FDCH is a read/write register that selects the number of channels used in multi-channel mode. For FIR filter mode, this register also selects the decimation ratio.

---

### EFCOP Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Register Name</th>
<th>Main Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDIR</td>
<td>Filter Data Input Register</td>
<td>DSP-to-EFCOP data transfers</td>
</tr>
<tr>
<td>FKIR</td>
<td>Filter K-Constant Input Register</td>
<td>DSP-to-EFCOP data transfers in adaptive mode</td>
</tr>
<tr>
<td>FDOR</td>
<td>Filter Data Output Register</td>
<td>EFCOP-to-DSP data transfers</td>
</tr>
<tr>
<td>FCNT</td>
<td>Filter Count Register</td>
<td>Selects filter length</td>
</tr>
<tr>
<td>FCSR</td>
<td>Filter Control Status Register</td>
<td>Used by core to control EFCOP modes and monitor status</td>
</tr>
<tr>
<td>FACR</td>
<td>Filter ALU Control Register</td>
<td>Used by core to control EFCOP ALU modes</td>
</tr>
<tr>
<td>FDBA</td>
<td>EFCOP Data Base Address</td>
<td>Address pointer to EFCOP FDM bank</td>
</tr>
<tr>
<td>FCBA</td>
<td>EFCOP Coefficient Base Address</td>
<td>Address pointer to EFCOP FCM bank</td>
</tr>
<tr>
<td>FDCH</td>
<td>Decimation/Channel Count Register</td>
<td>Selects number of channels and FIR filter decimation ratio</td>
</tr>
</tbody>
</table>
Let’s check your understanding of the material presented so far with a couple of questions. What is the main interface between the EFCOP and the DSP core? Click on your choice.

Answer: The Peripheral Module Bus is the main interface between the EFCOP and the DSP core. Note that DMA channels can also be used to transfer data between the EFCOP and the DSP core.
Question

Which statement best describes the IIR filter type? Click on your choice.

a) A non-recursive digital filter that includes poles and zeros.
b) A non-recursive digital filter that is inherently stable.
c) A recursive digital filter that includes poles and zeros.
d) A recursive digital filter that filters with a precise linear phase characteristic.

Which statement best describes the IIR filter type? Click on your choice.

Answer: The IIR filter type is a recursive digital filter that includes poles and zeros. Note that the FIR filter type is a non-recursive digital filter that is inherently stable and doesn’t include poles.
FIR Filter Type

- To select the FIR filter type, clear the FLT bit.
- Real mode performs FIR type filtering with real data.
  - Adaptive
  - Multichannel
  - Complex
  - Alternating complex
  - Magnitude mode
- Number of samples needed for initialization:
  - Number of filter coefficients minus one.
- The initialization mode is configured with the FPRC bit.
  - FPRC = 1, initialization mode is disabled, first sample is the first value written to FDIR
  - FPRC = 0, initialization mode is enabled, first sample is next value written to FDIR

Next, let’s take a detailed look at the two filter types beginning with the FIR filter. As noted earlier, the EFCOP operating mode is configured by setting bits in the status control register (FCSR). To select the FIR filter type, clear the FLT bit. The FIR filter provides four operating modes: real, complex, alternating complex, and magnitude mode.

Before the first sample is processed, the EFCOP filter must be initialized. This means that the input samples must be loaded into the FDM. The number of samples needed to initialize the filter is the number of filter coefficients minus one.

The initialization mode is selected by clearing the FPRC bit. If this bit is set, initialization is disabled and the EFCOP assumes that the core wrote the initial input values to the FDM before the EFCOP was enabled. In this case, the first value written to the FDIR is the first sample to be filtered.

If the FPRC bit is clear, initialization mode is enabled and the EFCOP initializes the FDM by receiving samples through the FDIR. After the samples are loaded, the next value written to the FDIR is the first sample to be filtered.
The FIR filter processing is shown here.

The EFCOP takes an input, X(n), from the FDIR, and saves the input while shifting the previous inputs down in the FDM.

It then multiplies each input in the FDM by the corresponding coefficient, Bn, stored in the FCM.

Finally, it accumulates the multiplication result. It then places the accumulation result, W(n), in the FDOR. The EFCOP performs this function for each sample input to the FDIR.
IIR Filter Processing

Next, let's look at the IIR filtering processing. You select this filter type by setting the FLT bit in the FCSR.

Note that initialization is always disabled with the IIR filter type and the FPRC bit is ignored. Therefore, the DSP56300 core must write the initial input values to shared memory before the EFCOP is enabled. The first value written to FDIR is always the first sample to be filtered.

The EFCOP multiplies each previous output value in the FDM by the corresponding coefficient, An, stored in the FCM.

It then accumulates the multiplication results, W(n); adds the input, X(n), from the FDIR; places the accumulation result, Y(n), in the FDOR; and saves the output while shifting the previous outputs down in the FDM. The EFCOP performs this function for each sample input to the FDIR. To process a complete IIR filter, the EFCOP requires a FIR filter session followed by an IIR filter session.
Implementing an IIR Filter

\[ Y(n) = \sum_{i=0}^{N} B_i x(n-i) + \sum_{j=1}^{M} A_j y(n-j) \]

Here, we see the difference equation for an IIR filter. \(x(n)\) is the filter input at time \(n\) and \(Y(n)\) is the filter output at time \(n\).

\(B_i\) represents the feed-forward filter coefficients and \(N\) is the number of feed-forward filter coefficients minus one.

\(A_j\) represents the feed-back filter coefficients and \(M\) is the number of feed-back filter coefficients.

Also shown is an example IIR filter design. This particular IIR design implements a butterworth lowpass filter with \(M = N = 3\) and a cut-off frequency of 0.8 \(W_n\), where \(W_n\) is half the sampling rate. The filter coefficients for these design parameters was determined using MATLAB. The coefficients both before and after scaling by 8 are shown.
The chart shows the output results for this particular IIR filter design using the input shown. The output is close to the frequency spectrum of the input for all frequency values less than 0.8Wn, where Wn is half the sampling rate. However, since the output is processed through the lowpass IIR filter, the frequency spectrum of the output is greatly attenuated for frequency values greater than 0.8Wn.
**Question**

What is the number of samples required to initialize an FIR filter? Click on your choice.

- a) Number of filter coefficients minus one
- b) Number of filter coefficients
- c) Number of filter coefficients plus one
- d) None, initialization is disabled

Here’s another question for you. What is the number of samples required to initialize an FIR filter?

Answer: The number of samples required to initialize an FIR filter is the number of filter coefficients minus one.
This concludes our overview of the DSP56300 family. In this tutorial, we reviewed the basic features of EFCOP, including its functional blocks and memory organization. We looked at how to configure the different EFCOP filter modes. We also took a detailed look at the processing steps for the FIR and IIR filter types.