Building High Performance FIR Filters Using KCM's

by Ken Chapman
Applications Specialist
Xilinx Ltd - UK

July 1996

Introduction

The implementation of digital filters with sample rates above just a few mega-Hertz are generally difficult and expensive to realise using standard digital signal processors. At this point the potential of distributed arithmetic and parallel processing performed in a Xilinx FPGA becomes the ideal solution. The re-programmable aspect of FPGA's permits optimum use of the available gates in the form of Constant (K) Coefficient Multipliers (KCM), whilst enabling the filter to be tuned or changed at any time. Filters employing fully parallel KCM's are ideal for sample rates exceeding 27 MHz with the following example able to operate above 50 MHz.

KCM's - The Key to High Performance

At the heart of a FIR filter lies the multiply and accumulate (MAC) function. In a traditional digital signal processor, the MAC instruction is implemented in dedicated hardware. However, the sequential use of this single MAC engine limits the sample rate of an FIR filter as each TAP will require a MAC instruction to be performed. From a logic perspective, the multiplier also has to be a fully operational unit with two inputs, (one for TAP-data and the other for coefficients), which is both large and complex.

In a fully parallel implementation of a FIR filter, each TAP has its own multiplier. This multiplier receives TAP-data, which it always multiplied by the SAME coefficient. Hence, these multipliers can be optimised to have a single input and are called Constant Coefficient Multipliers (KCM's).

KCM's are implemented in Xilinx devices using the Hybrid technique (see Figure 1). Small blocks of ROM, (or sometimes RAM), store 16 partial products relating to the fixed coefficient and then use a simple adder to combine these products. As a result, KCM's are less than one third of the size of full multipliers.
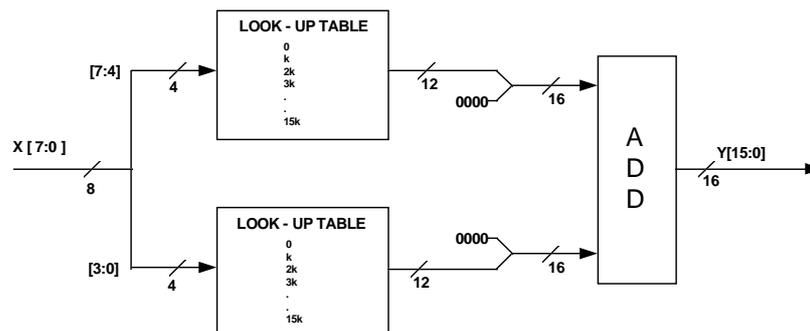

Fig 1. Constant Coefficient Multiplier (KCM)

Structures For FIR Filters

The traditional structure for a FIR filter, (see Figure 2), consists of a shift register, tap multipliers and then an adder tree to combine these results. All these elements can be implemented in the Xilinx FPGA devices. In particular the XC4000E family is register rich, ideal for KCM Hybrid multipliers, and contains in-built carry logic for addition.

The only potential problem is that of the lay-out on the device with respect to the adder tree where signals must propagate an increasingly greater distance for larger numbers of taps. Also, the adder tree only naturally scales for powers of two, (although the solution is obvious).
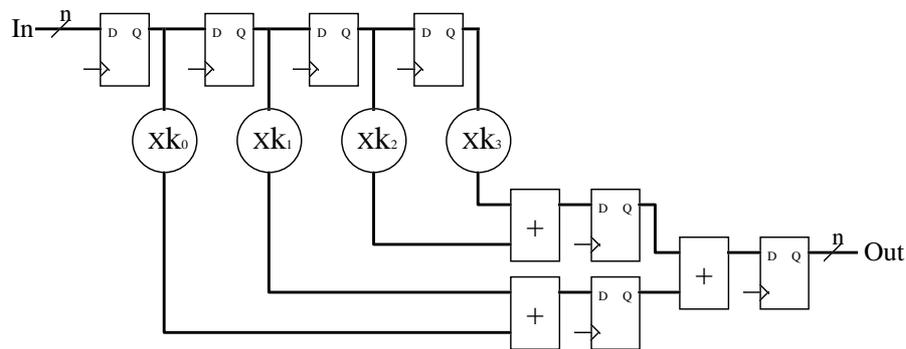
Fig 2. Traditional FIR structure employing tree of pipelined adders
An alternative implementation effectively inverts the structure, (see Figure 3). Utilising the same resources, the data samples are applied in parallel to all the tap multipliers via pipeline registers to reduce loading, (more or less can be used to meet desired performance). The results are then applied to a cascade chain of registered adders which combine the effect of accumulator and shift register. This structure provides a means for easily expanding the number of taps required in a filter since each "slice" is identical. With regards to device layout, the adder chain is now able to flow from tap to tap using local interconnect. Unfortunately, the distribution of the incoming data samples to all taps in parallel is now routing intensive, but this area has the advantage of being easily pipe-lined and more flexible than addition blocks.
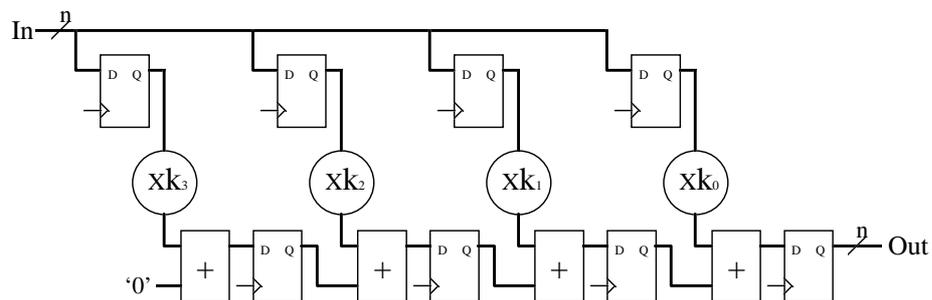
Fig 3. 'Inverse' FIR structure employing cascaded pipelined adders

Selecting a Style

Clearly there is a trade-off between the two styles. In the example design supplied, (10-BIT FIR, 8-TAP), the results are very nearly identical in both utilisation of the device and obtained performance. In general, a smaller filter would benefit from the traditional approach, whilst a larger filter would see advantage in the "inverse" approach. This argument becomes clear when very large filters are implemented across multiple devices. The cascadable nature of the "tap-slice" blocks making the inter-device connections very obvious.

Building Filters

Design Entry
The supplied examples of traditional and "inverse" approach are VIEW-logic based schematics of a 10-BIT, 8-TAP FIR filter. The design methology is very similar regardless of design entry tool, (including VHDL).

There are only four basic logic blocks:-

(i) IO-STUFF contains all the input and output signals. The clock should employ a global clock buffer. Data signals can benefit from the I/O Registers on the device. Some care should be taken with the pipe-line stage this introduces when cascading a filter across multiple devices, but in general should be used to aid system performance both inside and outside the device, (i.e. zero hold time requirement).

(ii) FD10 macros are simply 10-BIT pipe-line registers. Although these will normally be arranged as vertical columns in the silicon, it is worth keeping this flexible to aid density, (good for filling gaps!).

(iii) ADD10R are registered 10-BIT adders. The XC4000E contains in-built carry logic which should be employed for both density and performance advantage. The designer must decide what size these adders should be as they determine the internal resolution of the filter. In this example 10-BITs are maintained throughout by truncating the 20-BIT results of the multipliers.

(iv) KCM10R symbols represent the Constant Coefficient Multipliers. A utility program called KCM-GEN (DOS-based) greatly simplifies the generation of this critical logic. The symbols (or VHDL instantiations) are simply "Black Boxes" referencing the XNF (Xilinx Netlist) generated by KCM-GEN. The program is able to generate pipe-lined and combinatorial versions of 8,10 and 16-BIT KCM's which are optimum in density and performance. The generated macros are also Relationally Placed Macros (RPM) which locks in the topology of the logic.

Processing the Design

Once the basic design has been implemented, KCM-GEN is used for each TAP coefficient. The program asks for the type of multiplier required, and of course the value of the coefficient. Execution of the program is only a few seconds for each KCM generated. The complete design is then ready for compilation. This can be performed totally automatically, however the very structured nature of an FIR filter means that use of the XACT Floorplanner is very appropriate.

The packing density can be improved by using every corner of the device, and performance can also be increased by typically 5 to 10% in this example. Floorplanning is normally considered to be a very time consuming process, but in this case it is actually a quicker route to device bitstream than automatic tools. The reasons for this are the highly structured design and the intensive use of RPM's (KCM's and the adders). This means that the designer only has to place as many blocks as there are symbols on the schematic, and NOT the hundreds of elements of which they are formed. Floorplanning these examples took approximately 10 minutes each, which was less time than the automatic tools took to place the design. The automatic router is still used.

Changing Coefficients

It may be necessary to "tune" a filter in a system, or even have multiple filter settings. Here the SRAM technology of the XC4000E can be exploited by reconfiguring the part multiple times either by exchecker cable during development, or by peripheral/master modes during system operation. In all these cases, the changes to the filter only lie in the coefficients with the actual structure of the design remaining unchanged. Hence, only the KCM-GEN utility needs to be re-executed and the design recompiled. Here, again, more time can be saved during compilation. The coefficients contained in KCM macros are stored as partial products in ROM elements. These not only permit ANY values to be programmed into the same logic, but because ROM elements are implemented in the function generators of the CLB's (Configurable Logic Blocks), all the placement and routing of the design into the FPGA remains unchanged. During the re-processing of the design, no new work needs to be performed either manually or automatically to place and route the design (PPR guide option), resulting in a rapid iteration of the filter and IDENTICAL timing.

Further Considerations

The following points are indications of aspects to consider in other FIR filter designs:-

(i) Lower Performance - The slowest XC4000E device is the -4 (4ns) part which yields 43.3 MHz for the 10-BIT, 8-TAP example implemented totally automatically. For lower sample rates, pipe-line stages can be removed to save logic and filter latency. As soon as multiple
clock cycles can be employed to process each data sample, a different technique should be considered to increase density, (i.e. the use of a 50MHz clock to process 5MHz data).

(ii) Symmetrical Filters - In symmetrical filters, pairs of coefficients are identical and using two multipliers with the same value becomes inefficient The solution is to add data and then multiply once, ie. $y = ax + bx = (a+b)x$. Both traditional and "inverse" structures can exploit this concept.

(iii) Resolution - As mentioned previously, the internal resolution of the filter will be set by the adders as the KCM's will always yield a full result, (unused logic subsequently trimmed out by XACT). In many filters, twos complement values are used, and it is often overlooked that a bit of resolution is wasted in practical applications. In the 10-BIT example, the largest result from any multiplier will occur when both data and coefficient are the maximum *negative* value 512. Only in this case will the MSB of the result carry useful information, (i.e. the sign flag). For all other values of data and coefficient, the MSB of the result will be identical to MSB-1. In practice, the signals applied to the filter will not extend to the maximum negative value especially as the maximum positive value is 511, and hence the MSB of each multiplier can safely be ignored. This means that smaller adders can actually yield the same resolution, or an additional bit of resolution gained by adjusting the connection point to the multiplier output.

Summary

The XC4000E provides an ideal solution to high performance FIR filter implementation with the inherent advantage of re-programming. The KCM Hybrid multiplier technique yields the density and performance required and via KCM-GEN provides a simplified design and development process.