White Paper

Soft Multipliers For DSP Applications

Introduction

New communication standards and high channel aggregation system requirements are pushing Digital Signal Processing (DSP) system performance requirements beyond the capabilities of digital signal processors. Altera's new Stratix field-programmable gate array (FPGA) family includes embedded DSP block multipliers and large numbers of shallow memories with huge I/O bandwidth, making them an excellent solution for these high-end DSP systems. The largest silicon consumer of these DSP systems are the multipliers required by finite impulse response (FIR) filters and other DSP functions, so an efficient implementation of multipliers is the key for cost effective solution of these applications.

Soft Multipliers, which are memory based multipliers, more than triple the already high cost effectiveness of FPGAs in DSP applications. Soft multipliers take advantage of the huge internal memories I/O bandwidth of Altera's Stratix family to increase the number of available multipliers for FIR implementation.

This article uses a terrestrial digital broadcasting subsystem application as a case study for the cost effectiveness of these new FPGA families for high bandwidth applications.

Case study - System definition

The terrestrial digital broadcasting subsystem is a FIR filter with the following requirements:

- 8 MHz input rate,
- 16 bit input resolution,
- 16 bit coefficient resolution, complex numbers arithmetic, 1024 taps, variable coefficients FIR Filter.

These requirements require 4,096 multiplications for each input received at 8 MHz clock rate. Therefore 32,768 Million Multiplications Per Second (MMPS) are required. The symmetric characteristics of the filter give some assistance:

- x(n-k) = x(n-4095+k) {k = 0 up to 2047}
- Y(n) = Sum[(x(n-i) + x(n-4095+i))*h(i)] {i = 0 up to 2047}

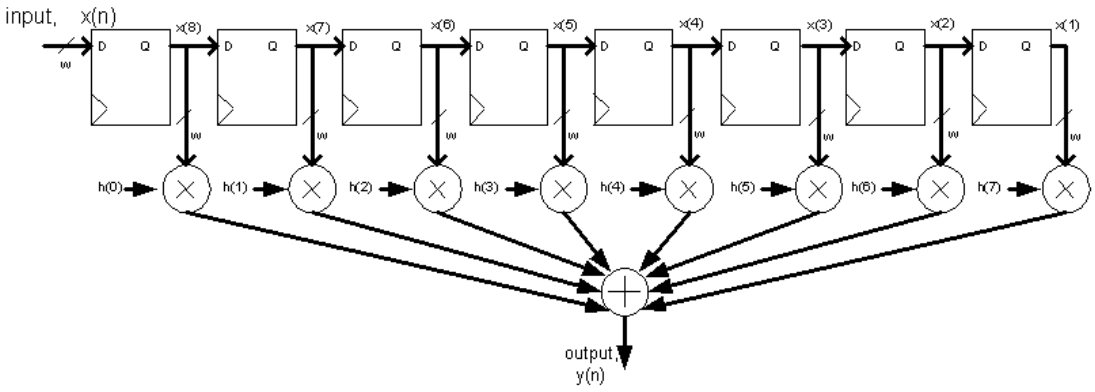
The symmetry of the filter makes it possible to reduce by a factor of 2 the number of multiplications. Nevertheless, 16,384 MMPS require a lot of computational power. Currently no single DSP processor has this level of computation power.

FIR Filter Implementation Options in Stratix Devices

The basic structure of a FIR filter (Figure 1) consists of a series of multiplications followed by additions. A FIR filter operation can be represented by the following equation:

$$y(n) = \sum_{i=0}^{L-1} x(n-i)h(i)$$

Figure 1: FIR Filter



The FIR filter in figure 1 is an example of a multiplier-intensive application. Most of the FIR filter silicon area is consumed by multipliers. The Altera® Stratix™ FPGA family has different resources that can be used for multiplier implementation:

1) DSP blocks: DSP blocks are dedicated blocks with customized multipliers.

2) Soft multipliers: multipliers can be constructed from memory blocks (detailed description is in the section 2.2). Soft multipliers use the Tri-Matrix™ memories of Stratix devices. There are 3 different sizes of memory block making up the Tri-Matrix™ memories in Stratix devices: M512 (32*18), M4K (128*36), and MRAM (4K*144). The configurations above are for the largest memory bandwidth option. Each memory has different configuration options (for example M512 has the following configurations: 512*1, 256*2, 128*4, 64*9, and 32*18).

3) Logic elements: The traditional method of implementing multipliers in a FPGA is with logic elements.

This article focuses on DSP block multipliers and soft multipliers. It is assumed that most logic elements will be consumed for other functions and for the FIR adder trees.

DSP Block Multipliers

Multipliers and their associated circuits (adders, subtracters, and accumulators) consume a significant portion of most DSP applications. Therefore, it makes sense to increase their performance/size efficiency by customization. The Stratix DSP blocks are dedicated and customized blocks of multipliers,

adders/subtracters, multiplexers and input and output registers. DSP blocks eliminate performance bottlenecks in DSP applications, provide predictable and reliable performance, and result in resource savings. Altera Stratix devices use DSP blocks together with soft multipliers to achieve the high data throughput necessary for computationally demanding applications. The DSP blocks have flexible functionality and flexible bit width, and deliver a predictable high-speed operation frequency that is not depended on availability of programmable logic routing resources. Figure 2 shows the DSP block structure.

The DSP block is composed of multipliers and 2 levels of adders/subtracters/accumulators. The DSP block is capable of operating in sum-of-multiplications mode, complex multiplication mode, independent multiplier mode, or 2 multiplier/accumulator mode. The DSP block supports, if needed, different clock sources for each multiplier, and has optional pipeline registers between the multipliers and the adders/subtractors to enable higher throughput.

DSP blocks supports bit widths of 8 multipliers 9*9, 4 multipliers 18*18, or one multiplier 36*36. The DSP block input register unit can load data and coefficients. Serial or parallel load modes are supported for both, data and coefficients.

Figure 2: DSP block structure

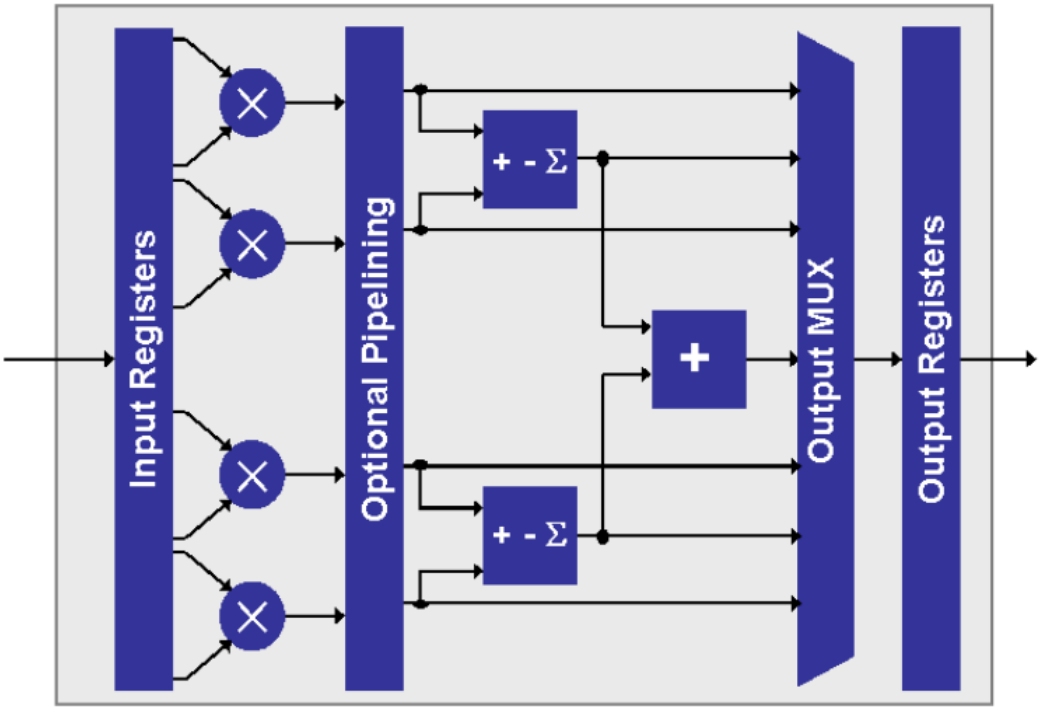


Figure 3 describes the different modes of input register operation. The input register modes give the flexibility needed to support efficiently different DSP applications input data streams:

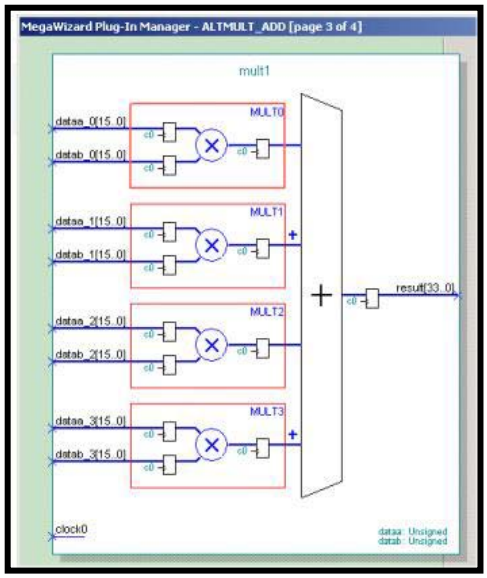
1) Data and coefficients can be parallel loaded into the multipliers input registers (shift registers cannot be used as a tap delay line).

2) Either data or coefficients use a tap delay line while the other is parallel loaded into the multiplier's input registers. This mode can be used to switch FIR filter coefficients with parallel loads while the data is shifted serially using a tap delay line.

3) Tap delay lines can be used for both data and coefficients.

Figure 3: Modes of operation for the input registers

Data and Coefficients Not Using Tap Delay Line Feature



Data OR Coefficients Using Tap Delay Line Feature

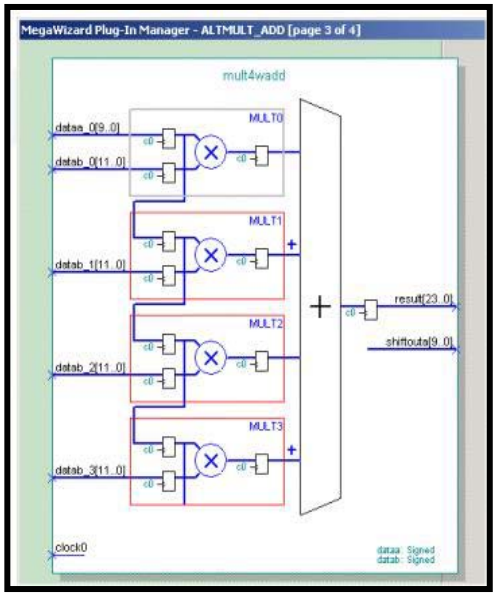
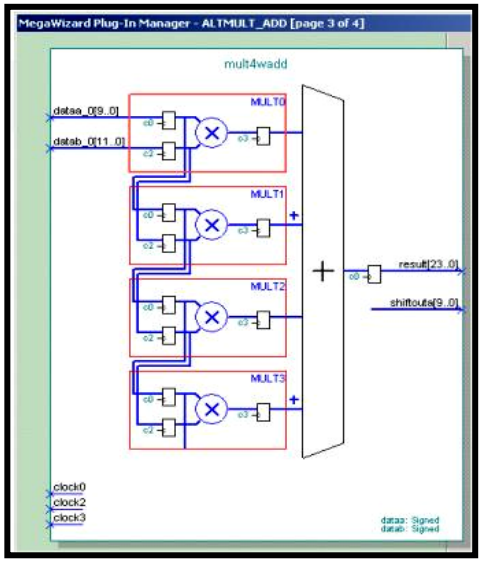


Figure 3: Modes of operation for the input registers (Continued)

Data AND Coefficients Both Using Tap Delay Line Feature



The DSP blocks are composed from customized and silicon efficient multipliers and their associated adder/accumulator circuits. However, since DSP blocks are dedicated to the multiplier function, the percentage of DSP blocks out of the FPGA silicon area cannot be too high to maintain the FPGA flexibility. For arithmetic-intensive applications that require more multipliers than are offered by the DSP blocks, soft multipliers are an excellent complement to DSP blocks since they increase the number of multipliers while maintaining the high flexibility of the FPGA architecture.

Soft Multipliers

Soft multipliers are an extremely flexible alternative to using DSP blocks. Instead of implementing a combinatorial logic multiplier, they utilize a novel implementation based on a partial look-up table (LUT) implementation of the multiplication operation, where the LUT is implemented in the memory blocks in Stratix devices. Soft multipliers increase by a factor of between 2 and 15 the number of multipliers available on Stratix devices. With soft multipliers, Stratix devices are the most cost effective solution for multiplier-intensive applications.

By downloading different coefficient LUTs, different configurations of multipliers and adders are generated.

Figure 4 shows simple Soft Multiplier implemented with a M512 (32*18) RAM block. The 5 bits width input data is driving the address bus of a memory and pointing to a LUT location that has the 18 bit result. The LUT covers all the multiplication combinations of 5 bits of input data with 13 bits coefficient.

Figure 4: Simple soft multiplier

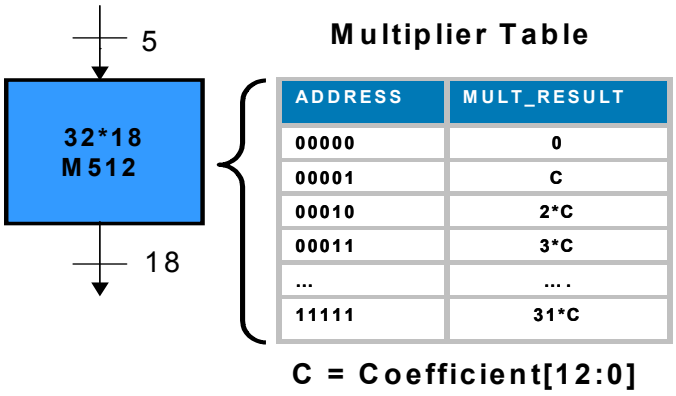
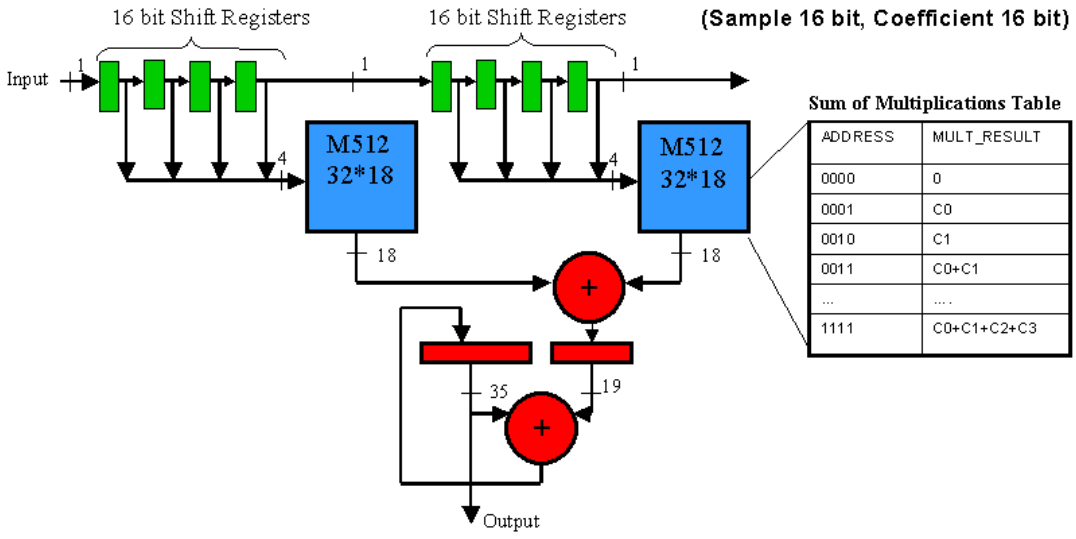


Figure 5 shows the sum-of-multiplications mode of operation. This is the mode of operation that is used for FIR designs. All possible combinations of a multiplicand summation are calculated and stored in the RAM block as a LUT. As a result, each address of the LUT represents a unique sum of multiplications result.

Figure 5: 8-Tap FIR filter implemented with soft multipliers operating in sum-of-multiplications mode



The input samples are serially shifted into a shift register. The shift register's taps drive the memory block's address buses. At each clock cycle, the sum of memory blocks outputs gives an intermediate sum-

of-multiplications result. The accumulator at the end of the adder tree gives the complete FIR filter result after n clock cycles (*n* is the resolution of the input sample.)

Figure 5 also shows the soft multipliers implementation of an 8 taps FIR filter. The sample data resolution is 16-bit and the coefficient resolution is also 16-bits. It takes 16 clocks to compute each filter output. In this example 2 M512 memory blocks are used to implement a full 8-tap FIR filter.

The performance of soft multipliers operating in sum-of-multiplications mode is determined by the length of the shift register (speed = system clk/n-bit sample). For filters that require high performance, the designer can use multiple memories and split the shift register into smaller shift registers. This technique uses more M512 RAM blocks, but Stratix devices have large numbers of TriMatrix memory blocks to accommodate this. Also, if the filter requires a larger coefficient than 16-bits, multiple M512 memory blocks should be used. The results of memory blocks are shifted-and-accumulated to add the partial products and obtain the final filter result.

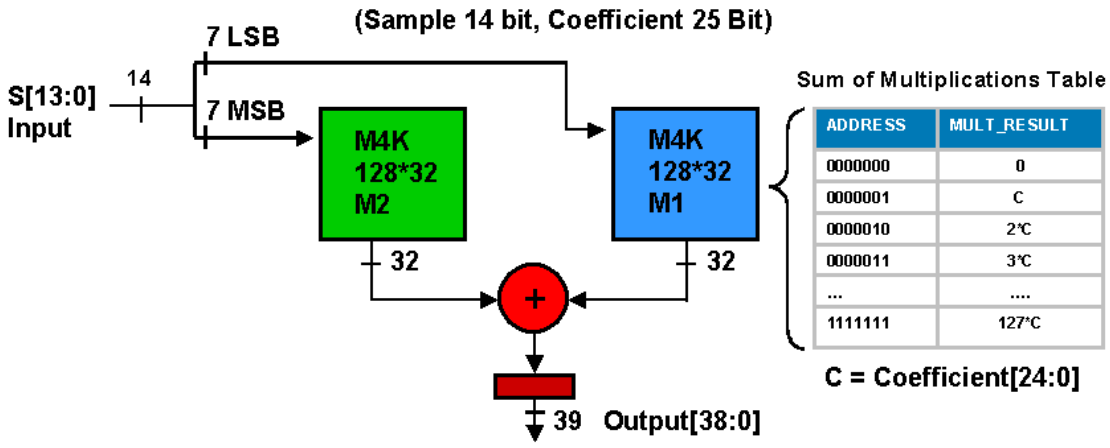
If needed, soft multipliers supporting variable coefficients can be used, without interrupting the multiplication process. The Tri-Matrix memory blocks of Stratix devices are dual-port memories. While one address and data register port is performing multiplications using one LUT address space, the other address and data registers port can be used to write a new coefficient LUT at another memory space. Once there is a need to switch to a different set of coefficients, only an address switch is needed. The coefficients switch can be done in one clock cycle. The main advantage of this variable coefficients method is maintaining high, uninterrupted throughput during the multiplication process.

Additional operation modes of Soft Multipliers:

Parallel Soft Multipliers

Parallel soft multipliers give a high throughput of results at each clock cycle. Figure 6 shows a parallel soft multipliers example.

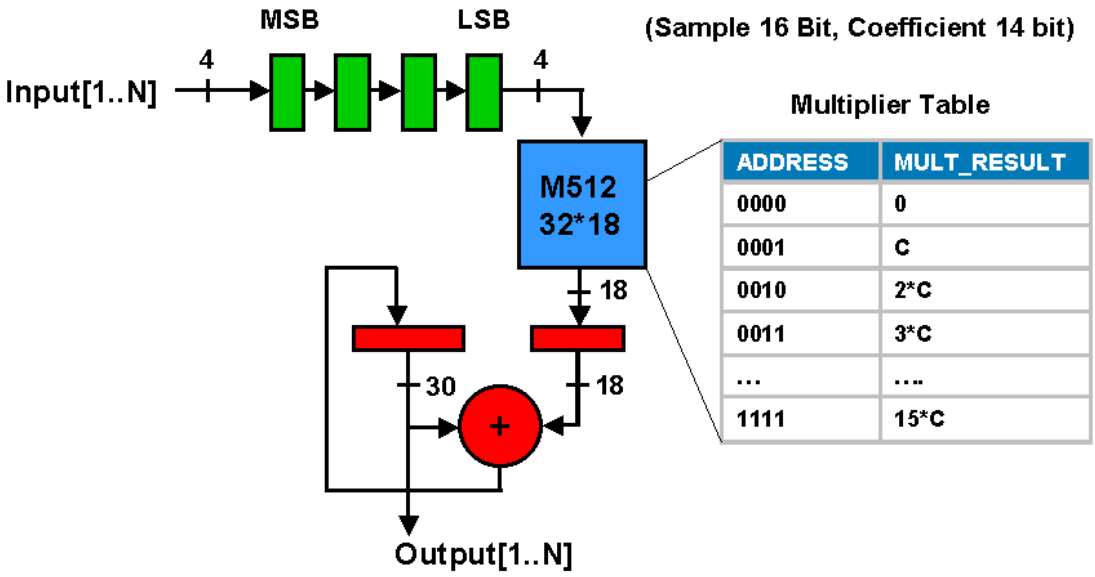
Figure 6: Parallel Soft Multiplier



Semi-Parallel Soft Multipliers

Semi-Parallel soft multipliers require fewer memory resources than parallel soft multipliers, but require multiple clock cycles to calculate each multiplication result. Figure 7 shows an example of a semi-parallel soft multiplier.

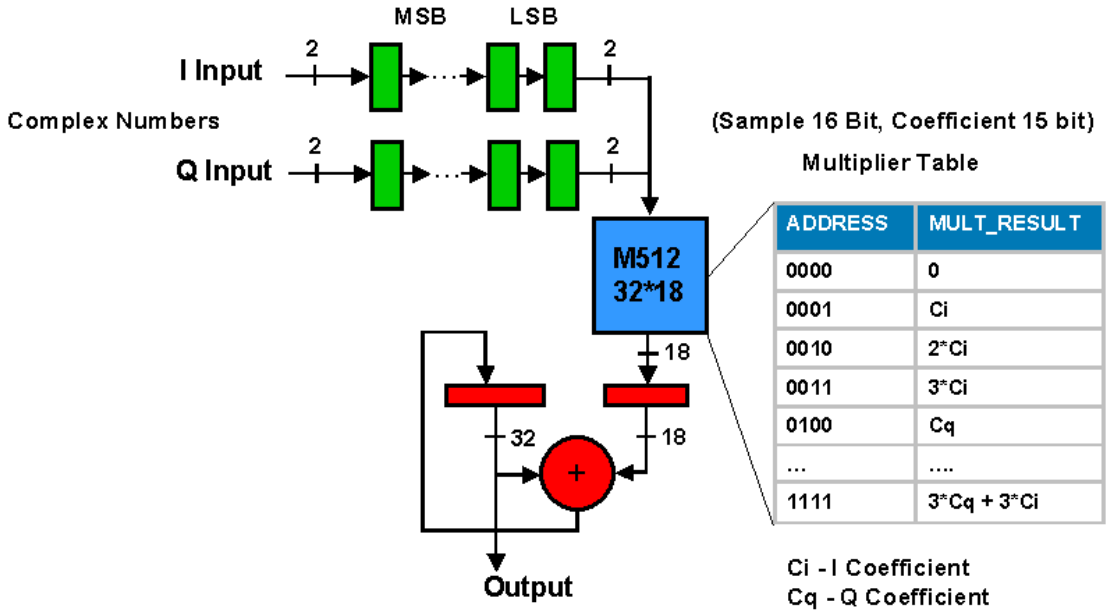
Figure 7: Semi-Parallel Soft Multiplier



Hybrid Soft Multipliers

Hybrid soft multipliers are ideal for multiplying complex numbers. They are suitable for FFT and NCO implementations. A hybrid soft multiplier is shown in Figure 8.

Figure 8: Hybrid Soft Multipliers



The different soft multiplier modes of operation give an optimized solution for different DSP applications. The combination of flexible functionality and high computational power enables soft multipliers to provide the most cost effective solutions for a wide range of computationally intensive and bandwidth-hungry DSP applications.

Logic Element-Based Multipliers

Another alternative multiplier architecture is the traditional FPGA implementation method using logic elements. Logic elements are the smallest and the most flexible building blocks of FPGAs. They are composed mainly of 16 bits of memory (16*1 configuration) with a 4 bit address bus and a flip-flop on the output. The logic element based multiplier is implemented as an array of logic elements, each logic element implementing a small portion of the multiplier function.

Since logic elements are the most flexible function resource in an FPGA, it is generally best to use DSP blocks and then memory blocks for multipliers before consuming logic elements. Logic elements are needed for the adder trees of DSP blocks and soft multipliers-based FIR filters and other system functions. Because of this, logic elements should generally be the last resort for multiplier implementation.

Table 1: Stratix Multiplier Implementation Options

Category	DSP Blocks	Soft Multipliers	Logic Elements
Fixed Coefficient Multiplication	Yes	Yes	Yes
Variable Coefficient Multiplication	Yes	Yes(1)	Yes(2)
Efficiency of Silicon Usage	High	High	Medium
Speed (up to)	> 300 MHz	> 300 MHz	> 300 MHz
Function Flexibility	Low	High	Very High

(1) Soft Multipliers Are Useful for Cases of Variable Coefficients When the Coefficients Update Rate Requirement Gives Enough Time to Load New Coefficients
(2) Not Size Efficient

Table 1 shows that all Stratix multiplier implementation options support both fixed and variable coefficients. It is possible to use variable coefficients for soft multipliers when the update rate requirement allows enough time to load new coefficients (For example, consider: 16 clocks period for 16 coefficients LUT size). It is possible to implement variable coefficient, logic element-based multipliers, but it is the least resource-efficient method compared to using DSP blocks or soft multipliers. Also, at high logic element utilization levels the routing resources are scarce which has a negative impact on the maximum system clock frequency.

In all cases, multiplier speed depends on parameters such as sample and coefficient bit width, pipeline levels, device speed grade, and availability of routing resources. The function flexibility is low for DSP blocks since they can be used only as multipliers and for few functions other than multipliers. The flexibility increases with memory blocks, and the highest level of function flexibility is achieved with logic elements. Logic elements are designed to be as flexible as possible.

Table 2: Stratix Device Family

Device	Logic Elements	DSP Blocks	32*18 M512 Blocks	128*36 M4K Blocks	18*18 Hard Multipliers	16*16 Soft Multipliers	Total
EP1S10	10,570	6	94	60	**24**	**53**	**77**
EP1S20	18,460	10	194	82	**40**	**89**	**129**
EP1S25	25,660	10	224	138	**40**	**125**	**165**
EP1S30	32,470	12	295	171	**48**	**158**	**206**
EP1S40	41,250	14	384	183	**56**	**187**	**243**
EP1S60	57,120	18	574	292	**72**	**289**	**361**
EP1S80	79,040	22	767	364	**88**	**373**	**461**
EP1S120	114,140	28	1118	520	**112**	**539**	**651**

Table 2 shows the resources of different Stratix devices (logic elements, DSP blocks, M512 and M4K memories). Based on the DSP block, M512 block, and M4K block information, the number of Hard and soft multipliers were calculated. M-RAM (4K*144) memory blocks (The third size of Tri-Matrix memory block) are not presented in table 2 as they are not used to create soft multipliers in our case study.

1024 taps FIR Filter Implementation for Terrestrial Digital Broadcasting Application: Implementation Strategy

As we mention previously we will use the case study of a terrestrial digital broadcasting application to demonstrate the efficacy of implementing multiplier-intensive functions in Stratix devices.

Implementation

Using the filter's symmetric characteristic the number of required multiplications per second was reduced to 16.384 billion. Since each two 16-bit input samples are added together to reduce the number of multiplications, the multiplier input resolution is increased to 17 bit. The input data of the soft multipliers is serial, and therefore for simplification reasons, the system clock frequency is set to 136 MHz (17 bits * 8 MHz input sample rate). A system clock frequency of 136 MHz enables completion the soft multiplier serial multiplication in 17 clocks. 136 MHz is considered a low clock frequency for Stratix devices. It is possible to increase the clock frequency up to more than 250 MHz, depending on routing resources and availability, and reduce the number of multipliers needed and potentially fit into a smaller device. The low speed system was chosen in this article to simplify the system description.

121 multipliers are needed to implement the filter. The reason: 16,384 MMPS / 136 MHz < 121 multipliers.

A 25% reduction in the number of multipliers can be achieved by using a canonical representation of complex multiplications. It uses 3 multipliers instead of the 4 multipliers required using the standard complex multiplication method. This paper does not cover this size-reduction method.

Table 2 shows the Stratix device family, their DSP blocks, hard multipliers, and their soft multiplier resources. The EP1S20 device, with a total of 129 multipliers (DSP blocks and soft multipliers) can do the job for this implementation.

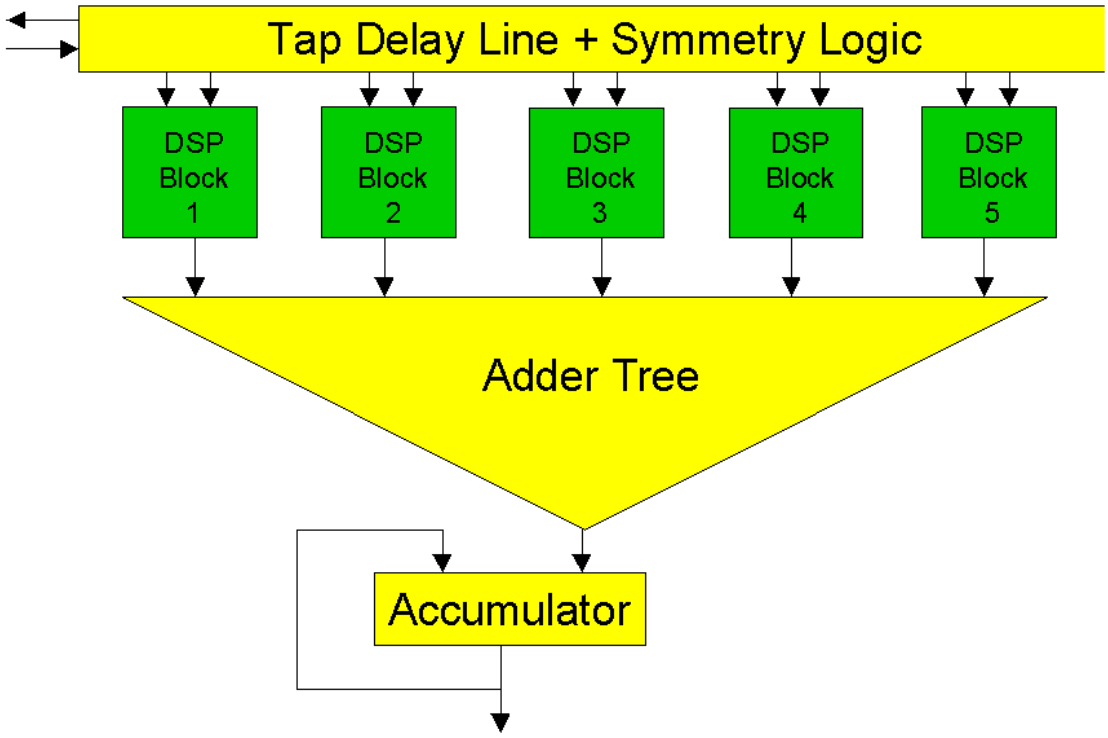
Since the filter is a complex FIR filter, it will be divided into two sections, I and Q.

Data	a+jb	
Coefficient	C+jD	
	(a+jb)*(C+jD) = (aC-bD) + j(bC+aD)	
I section	(aC-bD)	{2*512 =1024 Multiplications}
Q section	(bC+aD)	{2*512 =1024 Multiplications}

Each section is constructed from DSP blocks and soft multipliers (subdivided into M512 and M4K based soft multipliers as illustrated in figure 12). First the DSP block multipliers are utilized. There are 10 DSP blocks in the EP1S20 device. 5 DSP block multipliers are used for each section I and Q.

5 DSP blocks have 20 18*18 multipliers. For each 8 MHz sample, the 20 multipliers are able to perform 340 (20*17) multiplications. Figure 9 shows the DSP block FIR section, composed of the input tap delay line and the symmetric logic that can be implemented with logic element registers or Stratix memory. The tap delay line is feeding data and coefficients into the DSP blocks. The adder tree is adding all the intermediate results of the DSP block multipliers. The accumulator is accumulating all the intermediate results. The adder tree and the accumulator are implemented with logic elements.

Figure 9: DSP blocks segment of the FIR filter



The DSP blocks contribute 40 multipliers out of the 121 needed for the FIR implementation. Soft multipliers can more than triple the total number of EP1S20 multipliers. Figure 10 shows 91 M512 soft multipliers, each one of them performing 4 multiplications simultaneously and giving 364 multiplications (4*91). Figure 11 shows 40 M4K soft multipliers, each one of them performing 8 multiplications simultaneously for a total 320 multiplications (8*40). Together the 340 DSP block multiplications combined with 364 M512 multiplications and 320 M4K multiplications gives a total of 1024 multiplications for each segment (I & Q). The function of the tap delay line, symmetry logic, adder tree, and accumulator is similar to their function of the DSP blocks section shown in Figure 9.

Figure 10: M512 based soft multiplier segment of the FIR filter

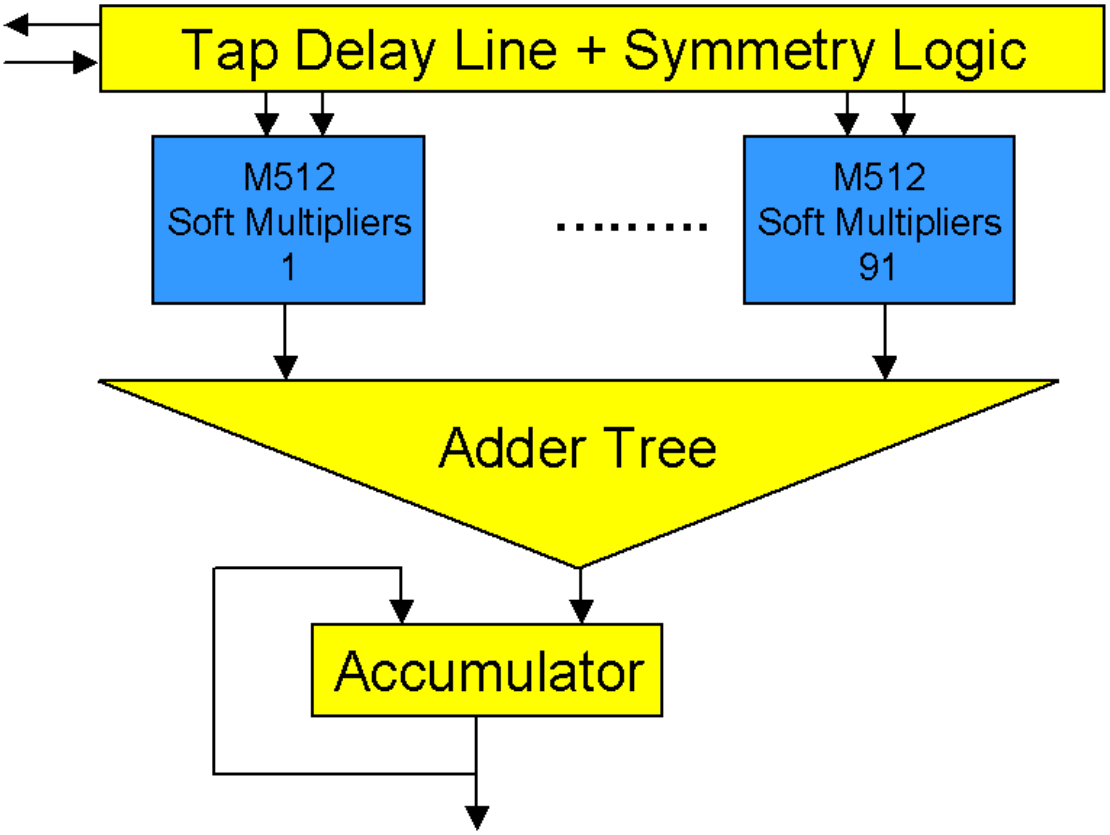
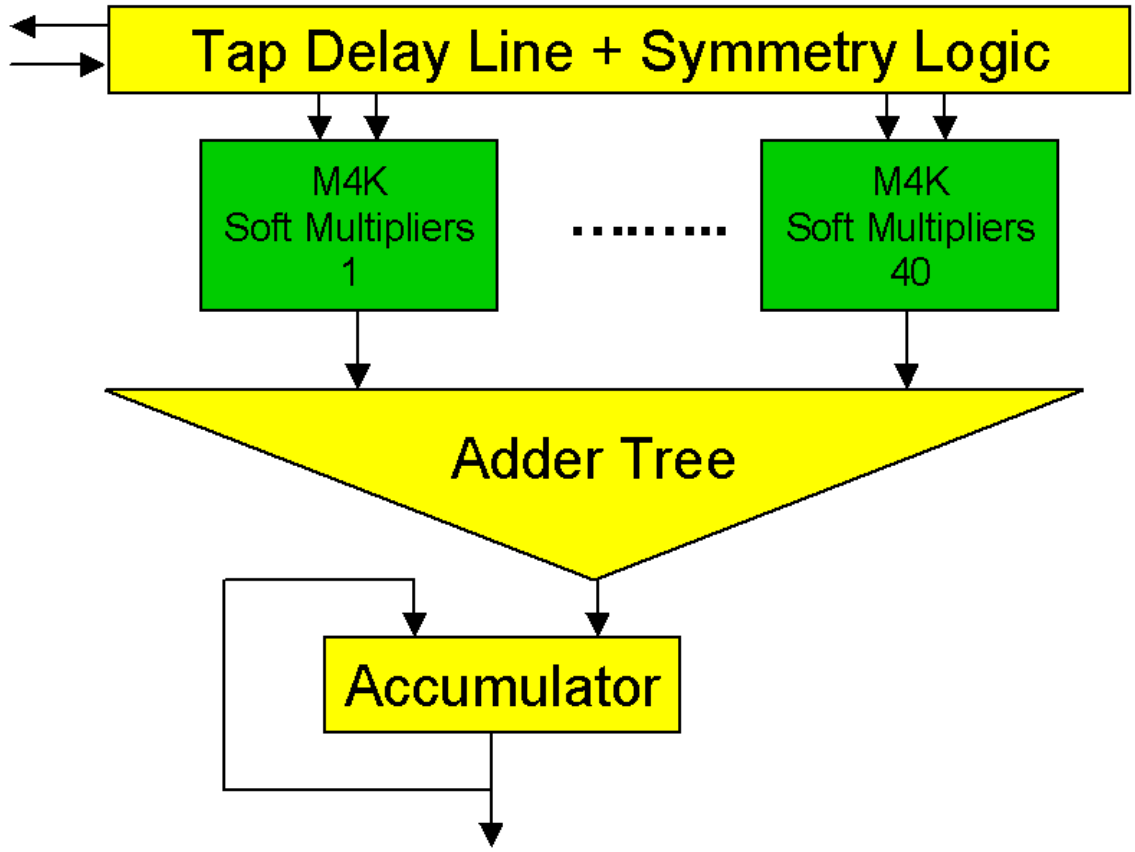
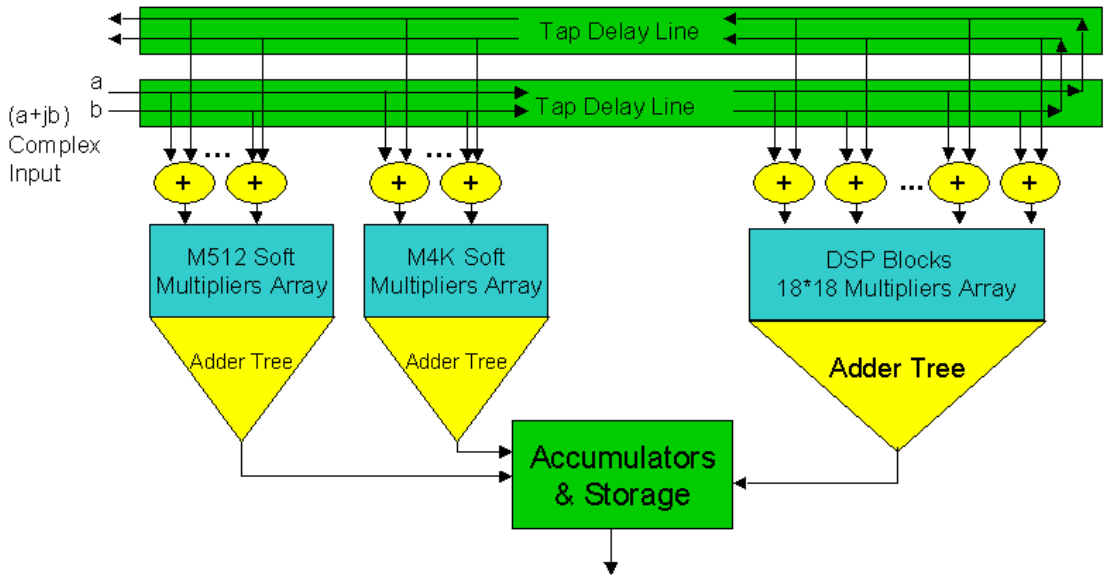


Figure 11: M4K based soft multiplier segment of the FIR filter



The 3 different segments of each section (I & Q) are cascaded together as shown in figure 12. The tap delay lines are cascaded together and the output of each adder tree is combined and accumulated together at the accumulator and storage unit to give the final result of each FIR sample point after 17 clocks (136 MHz clock).

Figure 12: 1024 taps complex FIR architecture (I or Q section)



Conclusion

The digital broadcasting 1024-tap complex FIR filter application that this article use as a case study demonstrates the unmatched computation power of Stratix devices. The parallel processing performance level that Stratix devices achieve in this application is unachievable by any other single chip FPGA of similar density, or any single DSP processor currently available. Stratix devices are well positioned to address the challenges of high-end and high-bandwidth applications.

Soft multipliers more than triple the number of multipliers available at Altera Stratix device for implementation of the digital broadcasting 1024 tap complex FIR filter. Using the same technique, users can translate the large number of M4K and M512 memory blocks offered by Stratix devices into a large number of soft multipliers to achieve higher levels of cost effectiveness compared with other FPGA families. This cost efficiency gives the Stratix family a clear advantage for any multiplier-intensive application.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com