DSP : Designing for Optimal Result

1-Why DSP Slice Is So Important

In many digital signal processing (DSP) applications, it is much more advantageous to use DSP Slices instead of using the general purpose fabric resources of the FPGA. The main reason for this is that DSP Slices are specially optimized hardware blocks.

High Performance: XtremeDSP Slices in Virtex-4 FPGAs offer the highest performance by operating at 500 MHz. This can reach 256 GMACS (Giga Multiply-Accumulate per Second) with 18x18 multiply and 48-bit add/accumulator operations.

Resource Efficiency: When DSP Slices are used instead of traditional FPGA fabric resources, additional logic resources such as adder trees are not required, thus more resources can be allocated for other parts of the FPGA.

Low Power Consumption: Each XtremeDSP Slice consumes only 2.3 mW of power at 100 MHz, significantly reducing the total power requirement of the FPGA. In addition, static power consumption is reduced by 40% and dynamic power consumption by 50%.

High Efficiency with Cascade Structure: XtremeDSP Slices can be combined with cascade connections to create very precise filters and large multiplication operations. This enables more efficient DSP operations with less hardware resources.

2-Overview of DSP48 Slice

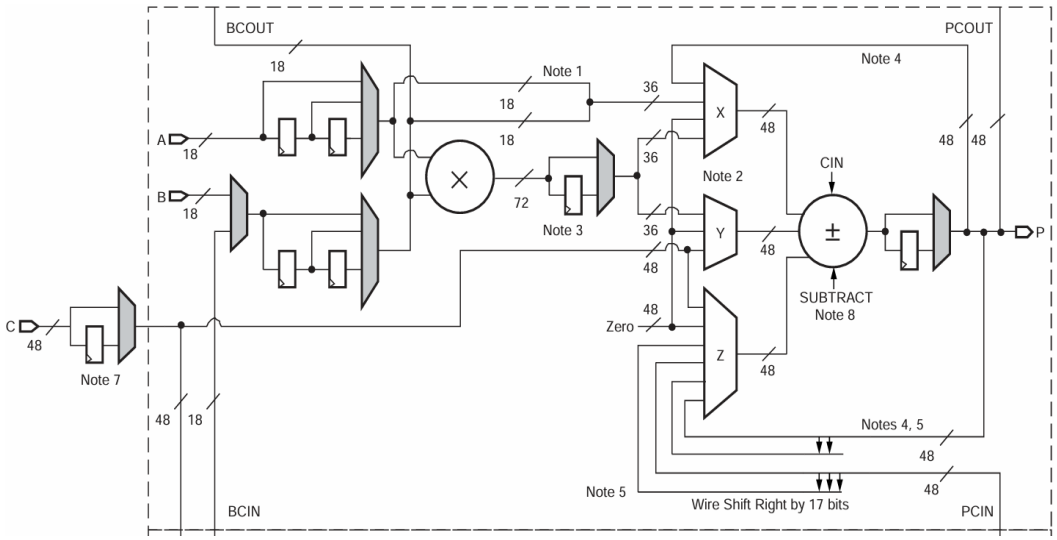

Figure 1: DSP48 SLICE

18-bit x 18-bit Multiplier Unit: Multiplies two 18-bit two's complement numbers, producing a 36-bit result, which can be sign-extended to 48 bits.
Three-Input 48-bit Adder/Subtractor: Processes multiplication results and performs addition, subtraction, or accumulation operations using three 48-bit inputs.
Dynamically Controlled Operating Modes: The DSP48 slice's function can be changed every clock cycle, enabling flexible signal processing solutions.

Cascade Data Paths:

- 18-bit B Bus: Allows input samples to propagate from one DSP slice to another.
- 48-bit P Bus: Enables partial results to be transferred between DSP slices.

Multi-Precision Multiplication Support: A 17-bit right shift feature allows wide multiplication operations to be performed in parallel or sequentially.

Symmetric Intelligent Rounding: Provides enhanced rounding support for applications requiring greater computational accuracy.

Performance-Boosting Pipelining: Pipeline options for input and control signals can be enabled via configuration bits, allowing the DSP slice to operate at maximum clock speed.

C Input Port: Used for multiply-add operations, three-input addition, or flexible rounding modes.

Independent Reset and Clock Enable: Separate reset and clock enable signals for both control and data registers.

I/O Registers: Ensure maximum clock performance and highest sample rates with no additional area cost.

OPMODE Multiplexers: A control mechanism used to select and modify different DSP48 slice functions.

3-Basic Mathematical Functions in DSP48 Slice

1. Addition and Subtraction

The DSP48 slice contains an adder/subtractor unit that can perform addition and subtraction operations.
The general operation formula is: Output = Z ± (X + Y +CIN)

Inputs X, Y and Z are selected from internal multiplexers.CIN (carry input) is the carry input. The SUBTRACT input determines whether the operation will be addition (0) or subtraction (1).

2.Accumulator

The DSP48 slice can perform an accumulator operation by adding the result of the previous operation to the new input value.

The output equation is as follows: Output = Output + A:B + C

3.Multiply-Accumulate - MAC

It can perform a MAC operation by multiplying two 18-bit numbers and adding the result to a value.
Output equation: Output = P + A:B + C
For large-scale calculations, multiple DSP48 slices can be combined with cascade connections.

4.Multiplexers

The DSP48 slice contains three internal multiplexers to route input data, 3:1 Y multiplexer, 4:1 X multiplexer, 6:1 Z multiplexer. OPMODE signals determine how the multiplexers select inputs. These multiplexers' outputs are the input of Adder/Substract block.

5. Barrel Shifter

DSP48 slice supports dynamic bit shifting operations (barrel shifter). It is used especially for pre-multiply scaling operations.

6. Counter

DSP48 slice can perform up/down counting operations. The output of the counter can be used with an additional accumulator.

7. Multiplication

DSP48 slice supports high-speed multiplication operations in hardware. 18x18 bit multiplication operation produces 36-bit result and can be extended to 48 bit with sign extension. Optimized for low latency and high-speed calculations.

8. Division

Division can be accomplished by iterative subtraction or multiplication.

Two methods:

Division by Subtraction: The division is subtracted from the number being divided by repetition.

Division by Multiplication: The division is sped up by multiplying the number being divided by a reciprocal value.

9. Square Root Calculation

DSP48 slice can perform square root calculation in an iterative method. Used in applications that require precise numerical operations.

10. Square Root of Sum of Squares

This operation is widely used in signal processing and Euclidean distance calculations. The sum of squares is calculated with the MAC operation and then the square root operation is applied.

4-MAC FIR Filters

1.Single-Multiplier MAC FIR Filter

The single-multiplier MAC FIR filter is one of the most basic FIR filter structures and multiplies and adds coefficients with data samples using only one multiplier and one accumulator. This structure makes efficient use of hardware resources while increasing processing time and making calculations sequential.

Architectural Structure and Working Principle

Multiply and add operations are performed using the DSP48 slice of the Virtex-4 FPGA.

Only one multiplication and one addition are performed in each clock cycle.

Using the delay line, previous input samples are stored by shifting and kept in a queue to be used in the next calculation of the FIR filter.

The result is added to the previous multiplication sum and the MAC (Multiply-Accumulate) operation is performed.

Mathematical Equation

The general FIR filter equation is as follows: $$y_n = \sum_{i=0}^{N-1} x_{n-i} \, h_i$$

Dual-Multiplier MAC FIR Filter

The double-multiplier MAC FIR filter uses two DSP48 slices, unlike the single-multiplier structure, and speeds up the filtering process by performing two multiplication and addition operations at the same time. It is especially advantageous at high sampling rates or large-scale filters.

Two DSP48 slices work in parallel and perform two multiplication operations at the same time. The input data samples are divided into even and odd indices and sent to two separate multiplication units. Two multiplication and two addition operations are performed in each clock cycle. The results are combined to obtain the final FIR filter output.

Symmetric MAC FIR Filter

Symmetric MAC FIR Filter is a method that optimizes the computational process by taking advantage of the symmetrical feature of FIR filter coefficients. By using this method on Virtex-4 FPGA, the sampling rate can be doubled. Without changing the clock frequency, the processing time is shortened and less hardware resources are used thanks to the symmetrical filter coefficients.

$$(X0\times C0)+(Xn\times Cn)\rightarrow(X0+Xn)\times C0,\ \text{if } C0=Cn$$

This transformation allows the input values to be added first and the result to be produced with a single multiplication operation instead of two separate multiplication operations.

Control Logic and Memory Usage

- Dual-Port Block RAM is used to store data and coefficients in the FIR filter.
- Distributed RAM such as SRL16E (16-bit shift register) is suitable for small-scale filters.
- Control logic provides addressing to manage the FIR filter delay line

Rounding and Truncation

- Since the filter output has more bits than the input, it must be reduced to the desired bit width.
- Truncation can create unwanted DC shifts.
- Symmetric rounding makes the output more precise.
- The DSP48 slice can perform this operation without using additional logic resources with its built-in rounding feature.

Chapter 5 of the document describes the implementation of high-performance, parallel, full-precision FIR (Finite Impulse Response) filters using the DSP48 slice in a Virtex-4 FPGA. The chapter emphasizes how the Virtex-4 architecture allows for flexible and optimized FIR filter designs tailored to specific applications.

5-Overview of Parallel FIR Filters

FIR filters are commonly used in digital signal processing, and there are various implementation techniques available. In high-performance applications, a fully parallel FIR filter structure is preferred, as it processes data at a high rate. The chapter discusses how the Virtex-4 DSP48 slice enhances multiplication and arithmetic operations, making it well-suited for parallel FIR filters.

The choice of architecture depends on two key factors:

1. **Sample Rate (Fs)** – The rate at which input data is processed.
2. **Number of Coefficients (N)** – The number of filter coefficients that define the FIR filter's response.

As both factors increase, the filter architecture needs to become more parallel, incorporating more multiply-and-add operations to keep up with the data stream.

Parallel FIR Filter Architectures

The chapter discusses two main parallel FIR filter architectures:

1. **Transposed FIR Filter**
    - Uses a pipelined adder chain, which makes it easier to integrate with DSP48 slices.
    - The input data is broadcast to all multipliers simultaneously.
    - Coefficients are ordered from right to left, with the first coefficient (h0) positioned on the right.
    - This structure is beneficial for low-latency applications as the latency never exceeds three clock cycles.
2. **Systolic FIR Filter**
    - Considered an optimal solution for parallel FIR implementations.
    - The input data passes through a series of registers that act as a data buffer.
    - Each register sends a sample to a multiplier, where it is multiplied by a coefficient.
    - Unlike the transposed FIR filter, the coefficients are aligned from left to right.
    - The adder chain stores accumulated products, forming the final result.
    - No external logic is required, making it highly efficient.

Advantages and Disadvantages

Transposed FIR Filter

Advantages:

- Low latency (typically three clock cycles).
- Efficient mapping to DSP48 slices.
- No external FPGA logic required, maximizing performance.

Disadvantage:

- Performance may be limited if there are too many filter taps due to high fanout.

Systolic FIR Filter

Advantages:

- High performance due to dedicated routing within DSP48 slices.
- Efficient mapping to DSP hardware.
- Fully utilizes available FPGA resources without needing external logic.

Disadvantage:

- Higher latency compared to the transposed FIR filter, as latency depends on the number of coefficients.

Performance Considerations

The chapter also discusses performance metrics, such as:

- The maximum input sample rate, which depends on the clock speed.
- The impact of filter coefficient symmetry in reducing hardware requirements.
- The benefits of using FPGA-specific DSP slices for optimizing FIR filter implementations.

6-Semi-Parallel FIR Filter Implementations

Digital signal processing (DSP) engineers can optimise performance and hardware resource utilisation by using different filtering techniques. Semi-parallel FIR filters are used to achieve medium to high sampling rates when sufficient clock cycles are available.

The chosen architecture of an FIR filter depends on the amount of data to be processed and the available clock cycles. The two basic parameters are as follows:

Sampling Rate (Fs) : Determines how fast the input data is processed.

Number of Coefficients (N) : How many coefficients (coefficients) the filter has.

In semi-parallel filters, the number of multipliers (M) is also an important factor. The more multipliers are used, the better the filter performance. The processing time of the filter is calculated by dividing the total number of coefficients by the number of multipliers:

Result per clock cycle = N / M

This formula is used to calculate how many multipliers are required to increase the efficiency of a quasi-parallel FIR filter.

Four-Multiplier, Distributed-RAM-Based, Semi-Parallel FIR Filter

This architecture uses the DSP48 slice to create a four-factor FIR filter. Distributed RAM is used in this design and the basic components are as follows:

Data Buffers: Memory areas where input data is stored.

Coefficient Memory: Areas where the coefficients of the FIR filter are stored.

Control Logic & Address Sequencing: They are control units that regulate the flow of data during operation.

This filter structure can operate with low latency and high clock speeds because it takes advantage of the internal routing mechanisms offered by DSP48 slices.

Three-Multiplier, Block RAM-Based, Semi-Parallel FIR Filter

Alternatively, a block RAM-based approach can be used to implement the FIR filter. Here, the memory unit in which data and coefficients are stored is block RAM instead of distributed RAM.

Advantages:

Can work with larger coefficient sets (e.g., filters with 300 coefficients).

Can use dual-port RAM to optimise data storage and access speed.

Disadvantages:

It can perform fewer operations per clock cycle.

May consume more hardware resources.

Other Semi-Parallel FIR Filter Structures

According to different application requirements, various semi-parallel FIR filters based on systolic, transposed, or block RAM can be built. The transposed FIR filter offers lower resource utilisation, while the systolic FIR filter can provide higher performance and more regular data flow.

The use of block RAM becomes advantageous above a certain number of coefficients.

Semi-parallel FIR filters consume less FPGA resources, especially for very large coefficient numbers.

Rounding

The number of bits at the output of the FIR filter is much larger than at the input and must therefore be reduced to a certain width. Truncation by simply selecting the most significant bits (MSB) creates an unwanted DC shift.

In the two's complement number system:

Negative numbers become more negative

Positive numbers also shift to the negative direction

Symmetric rounding is used to improve this situation. This method reduces DC drift by rounding positive numbers up and negative numbers down.

Rounding with DSP48 Slice:

For positive numbers: Add 0.10000... to the number, then truncate.

For negative numbers: Add 0.01111... to the number, then truncate.

The DSP48 slice supports rounding with the C input port and Carry-In port.

7-Multi-Channel FIR Filters

The main advantage of multi-channel FIR filters is that they efficiently utilise FPGA resources by sharing the same filtering structure for multiple input channels.

In conventional FIR filters, a separate FIR filter is used for each channel.
In multi-channel FIR filters, all input channels are processed sequentially by running a single FIR filter at a higher clock rate.

This structure reduces the total hardware resources used almost as much as the number of input channels.

Top Level Structure

The chapter describes how a six-channel and eight-coefficient (8-tap) FIR filter is designed.

The components used in this design are as follows:

6-to-1 multiplexer → Converts separate input streams into a single interleaved stream.

SRL16E delay lines → Hold data samples and organise the processing sequence.

DSP48 slices → Performs multiplication and addition operations.

DSP48 Slice Usage

DSP48 blocks enable efficient implementation of multi-channel FIR filters.

Each DSP48 slice contains multiplier, adder and delay registers, which are the basic components of a classical FIR filter.

Input streams (B bus) and output streams (P bus) are directly connected, enabling processing without going beyond the general FPGA logic.

Up to 500 million samples/second processing speed.

Converting Separate Input Streams into a Sequential Stream

In multichannel FIR filter design, data from different channels are processed in a time-multiplexed manner.

The 6 input channels are processed sequentially by a single FIR filter over many clock cycles.

A high-speed 6-to-1 multiplexer is used to perform this operation.

In each clock cycle, a different input channel is selected and converted to an interleaved stream using the SRL16E shift register.

Coefficient RAM

The coefficients are stored in the SRL16E memory.

The same coefficient set can be used for all channels or different coefficient sets can be defined for each channel.

When adaptive filtering is required, the coefficients can be updated by storing them in RAM structures.

Control Logic

The control logic is very important for the correct operation of the multichannel FIR filter:

When coefficient RAM blocks are used, the loading signal is active for six clock cycles.

The clock frequency operates at 6 times the clock frequency of the input signal.

The start delay must be completed before the filtering process starts.