**7 Series DSP48E1 Slice**

1. Overview

FPGAs are efficent for digital signal processing(DSP) applications Because they can implement custom, fully parallel algorithms. DSP applications use many binary multipliers and accumulators that are best implemented in dedicated DSP slices

**metin, diyagram, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure : Basic DSP48E1 Slice Functionality

* 25 x 18 two’s-complement multiplier;  
   Multiplier: A **multiplier** is a digital circuit that multiplies two numbers. In this diagram, a **25 × 18 multiplier** takes inputs
* 48-bit accumulator

Reccommendations for using DSP48E1 slices include:

* Use signed values in HDL sources
* Use Pipeline for performance and low power
* Use CLB for small multipliers,adders and counters (in VHDLwiz Blog Pages)
* If Whatever I want, Dont the other things (Set USE\_MULT to NONE when using only the adder/logic unit to save power)

1. Description and Specifics

The DSP48E1 slice supports many independent functions. These functions include **multiply, multiply accumulate (MACC), multiply add, three-input add, barrel shift, widebus multiplexing, magnitude comparator, bitwise logic functions, pattern detect, and wide counter.** The architecture also supports cascading multiple DSP48E1 slices to form wide math functions, DSP filters, and complex arithmetic without the use of general FPGA logic.

metin, diyagram, plan, şematik içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure : 7 Series FPGA DSP48E1 Slice

* The DSP48E1 Slice Consist of a multiplier followed by an accumulator.
* **At least three pipeline registers** are required for full-speed **multiplication** and **multiply-accumulate (MACC)** operations.
* The **multiply operation** produces two partial products, which are added in the second stage.
* **Add/Sub and Logic Unit** operations require **at least two pipeline registers** for full-speed operation.
* **Cascade capabilities** enable **efficient high-speed pipelined filters**, replacing adder trees with adder cascades.
* **Multiplexers** are controlled dynamically using **OPMODE, ALUMODE, and CARRYINSEL** for flexibility.

**HOW TO USE ITS?**

You can check Table 2-7,8,9,10 of DSP Slice User guide

* 25-pre-adder with D register to enhance the capabilities of the A path.
* INMODE control supports balanced pipelining (Multiply between add operations)
* Logic unit mode selectable by ALUMODE
* **DSP48E1 includes** a **two-input multiplier**, multiplexers, and a **three-input adder/subtractor/accumulator**.
* The **multiplier takes** an **18-bit and a 25-bit operand** (two’s complement) and produces a **43-bit result**, which is **sign-extended to 48 bits** before addition.
* When **multiplication is used**, the **adder functions as a two-input adder**.
* If **multiplication is bypassed (USE\_MULT = NONE)**, the **adder can handle three 48-bit inputs**.
* **SIMD mode** supports **dual 24-bit or quad 12-bit arithmetic** with **CARRYOUT bits**.
* **Cascading DSP48E1 slices** (using **ACOUT, BCOUT, PCOUT, etc.**) improves performance in **FIR filters** and **reduces power consumption**.
* The **C input port** allows **three-input addition**, **multiply-add operations**, and **symmetric rounding**.
* **Multi-precision arithmetic** is supported via a **17-bit right shift**, enabling **larger multipliers**.
* **Pipeline registers** improve **throughput** and **reduce latency**.
* The **pattern detector** supports **convergent rounding, overflow/underflow detection, and auto-reset counters**.

1. Input Ports

* **INMODE[3:0]** controls **how the A and D inputs affect the multiplier.**
* When **USE\_DPORT = TRUE**, **the pre-adder is enabled, allowing A and D to be processed together.**
* When **USE\_DPORT = FALSE**, **only the A registers are used, and D is ignored.**
* The **A input can be set to zero, or arithmetic operations (addition/subtraction) can be performed on A and D.**
* **Negative values (the negated version of A2 or A1) can be used in multiplication operations.**
* **INMODF[4]** selects the Multiplier B port

1. Output Ports

* All the output ports except **ACout** and **BCout** are reset by **RSTP** and enabled by **CEP**
* The DSP48E1 slices use **CARRYOUT** and **CARRYCASCOUT** to cascade carry signals, enabling larger arithmetic operations.
* **CARRYCASCOUT** is related to **CARRYOUT**[3] but not identical, and it feeds back into the same slice.
* In **SIMD** mode (or when using certain operations), **CARRYOUT**[3] isn’t used; alternate valid signals apply.
* For a 96-bit multiply-accumulate (MACC) operation spanning two slices, both **MULTSIGNOUT** and **CARRYCASCOUT** are combined, with specific settings (MACC\_EXTEND) required.
* **MULTSIGNOUT**, representing the multiplier’s most significant bit, is cascaded via **MULTSIGNIN** for building extended accumulators.
* Pattern detectors (**PATTERNDETECT**/**PATTERNBDETECT**) and dedicated overflow/underflow outputs monitor arithmetic results for specific patterns and errors.

1. Design Considerations

This chapter describes some design to use to achieve higher performance,low power and lower resources in a particular design To achieve maximum performance when using the DSP48E1 slice, the design needs to be fully pipelined.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure : Key Comparisons

Adder Tree

* First Stage: Add (A+B) and (C+D)
* Second Stage: Add The results from first stage

Adder Cascade

* First Stage: Add (A+ B)
* Second Stage : First Stage + C
* Third Stage: Second Stage + D