

## Data sheet for Complex Number Multiplier Core

### Introduction:

Complex Number multiplier IP is fixed point core which multiplies two complex numbers

$$a = a_{\text{real}} + j a_{\text{image}} \quad \text{and}$$

$$b = b_{\text{real}} + j b_{\text{image}}$$

such that the product is

$$O = a * b = O_{\text{real}} + j O_{\text{image}}$$

where

$$O_{\text{real}} = a_{\text{real}} * b_{\text{real}} - a_{\text{image}} * b_{\text{image}}$$

-- eq(1)

$$O_{\text{image}} = a_{\text{real}} * b_{\text{image}} + a_{\text{image}} * b_{\text{real}}$$

--eq(2)

Complex number multiplication is commonly used in DSP application such as like FFT. It performs synchronous multiplication of two complex operands and produces result as a complex number.

### Functional description:

Complex multiplier is implemented using four multipliers, one adder and subtracter, as shown in eq(1) and (2). The multiplier, add and sub modules are implemented as fixed point arithmetic to get better precision of computation.

### Features:

- Fixed point arithmetic.
- Synchronous and pipelined design.
- Compatible, flexible and easy integration with other modules.

### Block Diagram:

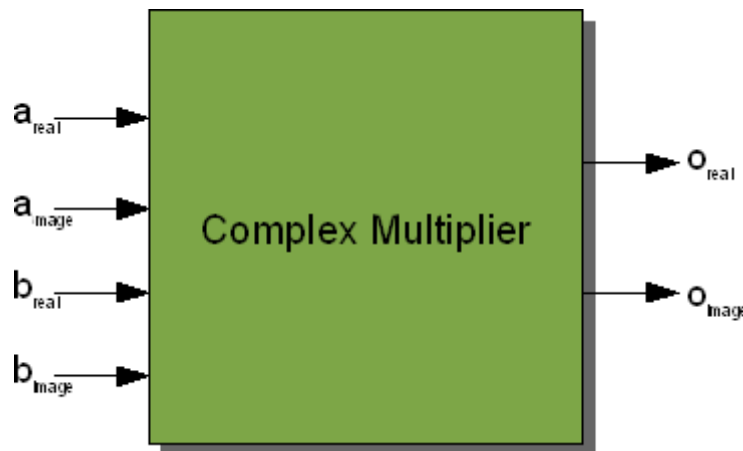


Figure 1: Complex Multiplier Block Diagram

**Architectural diagram:**

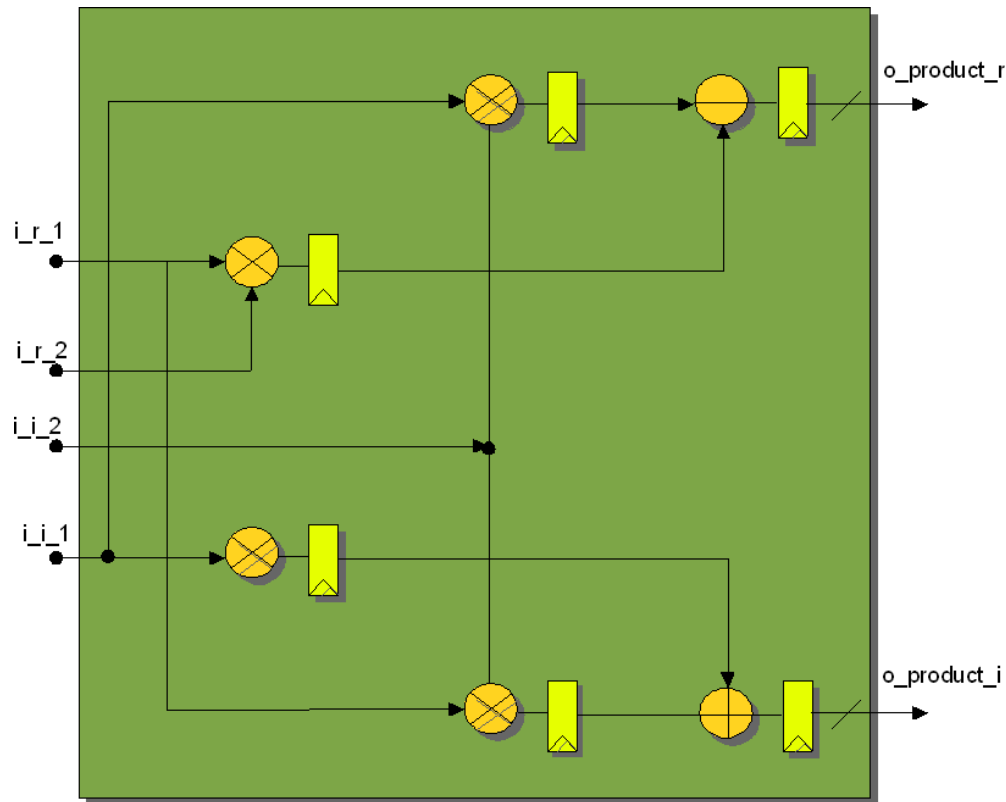


Figure 2: Complex Multiplier Architectural diagram.

**Description:**

1. Figure 1 shows block diagram of complex multiplier. It shows all inputs and outputs of the module.

Here  $a_{real}$ ,  $a_{image}$  and  $b_{real}$ ,  $b_{image}$  are two complex operands. After multiplication, the results is itself complex number with  $o_{real}$  and  $o_{image}$  as real and imaginary terms.

2. After multiplication real part of complex number is  $(a_{real} * b_{real}) - (a_{image} * b_{image})$  and imaginary part is  $(a_{image} * b_{real}) + (a_{real} * b_{image})$

3. Figure.2 shows architecture of complex multiplier IP core.
4. It is having 4 fixed point multipliers and one fixed point adder and subtractor.
5. Inputs ports  $i_{r_1}$ ,  $i_{r_2}$ ,  $i_{i_1}$  and  $i_{i_2}$  are real and imaginary parts of two input complex numbers and output real and imaginary values are  $o_{product_r}$ , and  $o_{product_i}$  respectively.
6. Table. 1 show all parameters used in the core.
7. Figure.3 shows schematic diagram of complex number multiplier. It shows all input and output ports along with their data width.

- 8. Table.2 shows signal definition table of the complex number multiplier core.
- 9. Table.3 shows gives frequency of operation and

slice count of the design when it targeted on virtex-4 and virtex-5 devices.

### Complex Multiplier Parameter Table

This table describes the general Complex Multiplier parameters:

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
WL	Integer	Represents width of the each data symbol.
IWL1	Integer	Number of bits allocated to represent integer part of first complex number operand. WL-IWL1 bits are allocated to represent fractional part of the number.
IWL2	Integer	Number of bits allocated to represent integer part of second operand. WL-IWL2 bits are provided to represent fractional part of second complex number.

Table 1: Complex Number Multiplier Parameter table.

### Schematic Symbol

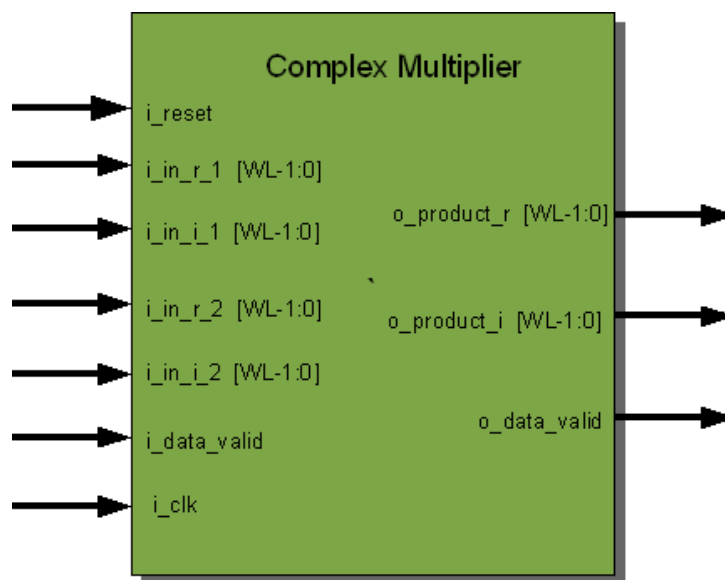


Figure 3: Complex Multiplier Schematic Diagram

### Signal definition table:

<b>Signal</b>	<b>Direction</b>	<b>Data width</b>	<b>Description</b>
<i>i_clk</i>	IN	1	Complex Multiplier is single clock system and all I/Os and internal multiplier processor is in synchronous with it.
<i>i_reset</i>	IN	1	This is one bit input signal which resets all registers and counters.

<b>Signal</b>	<b>Direction</b>	<b>Data width</b>	<b>Description</b>
i_in_r_1	IN	WL	This is the port of WL width through which real part of the first operand is fed through.
i_in_i_1	IN	WL	This is the port of WL width through which imaginary part of the first operand is fed through.
i_in_r_2	IN	WL	This is the port of WL width through which real part of the second operand is fed through.
i_in_i_2	IN	WL	This is the port of WL width through which imaginary part of the second operand is fed through.
i_data_valid	IN	1	This is single bit input signal which indicates the validity of input data all input ports at <b>i_in_r_1</b> , <b>i_in_i_1</b> , <b>i_in_r_2</b> and <b>i_in_i_2</b> as long as it is enabled.
o_product_r	OUT	1	This is WL bit input. It mentions about the length of the decoded data symbols. All the code lengths defined in the DVB-S2 and DVB-S standard are supported.
o_product_i	OUT	1	This signal indicates the starting of input received data block. As long as this signal is high, indicates that data presented at input port <b>data_in</b> is valid received block data.
o_data_valid	OUT	1	This is single bit output signal which indicates the validity of the multiplied output at output ports <b>o_product_r</b> , <b>o_product_i</b> .

Table 2: Complex Multiplier signal definition table.

#### Performance:

<b>Family</b>	<b>Device</b>	<b>Slice Register count</b>	<b>Slice LUT count</b>	<b>Frequency (MHz)</b>
Virtex-4	lx15-12sf363	518	472	600
Virtex-5	lx30-3ff324	518	472	684

Table 3: Complex Number Multiplier Core performance table.

#### Verification:

The Complex multiplier core module has been verified with following approaches:

- Exhaustive Functional/Timing simulation.
- Results compared with MATLAB functions and complex multiplier -C reference code functionality.

#### Deliverables:

- Verilog RTL source code

- The IP core test environment developed in verilog HDL (test benches).
- Synthesis and Simulation scripts.
- Detailed user documentation, including RTL source code documentation.

#### Applications:

Complex multiplier finds extensive applications in areas like digital signal processing, communication and etc. It is one of basic building block of digital system