**ID1000500C**

**CONVOLUTION IP-CORE USER MANUAL**

# DESCRIPTION

The convolution task between two vectors *u* and *v* is a mathematical operation that represents the area of superposition under the points when *v* spans the same length as *u*.

## CONFIGURABLE FEATURES

|  |  |
| --- | --- |
| **Software configurations** | **Description** |
| Shape | Convolution type FULL or SAME |
| sizeX | Memory size X |
| sizeY | Memory size Y |

## TYPICAL APPLICATION



Figure 1.1 IP Dummy connected to a host

# CONTENTS

[1. DESCRIPTION 1](#_Toc167486164)

[1.1. CONFIGURABLE FEATURES 1](#_Toc167486165)

[1.2. TYPICAL APPLICATION 1](#_Toc167486166)

[2. CONTENTS 2](#_Toc167486167)

[2.1. List of figures **¡Error! Marcador no definido.**](#_Toc167486168)

[2.2. List of tables 2](#_Toc167486169)

[3. INPUT/OUTPUT SIGNAL DESCRIPTION 3](#_Toc167486170)

[4. THEORY OF OPERATION 5](#_Toc167486171)

[5. AIP interface registers and memories description 5](#_Toc167486172)

[5.1. Status register 5](#_Toc167486173)

[5.1. Configuration register 6](#_Toc167486174)

[5.2. Input data memory X 6](#_Toc167486175)

[5.1. Input data memory Y 6](#_Toc167486176)

[5.2. Output data memory Z 6](#_Toc167486177)

[6. PYTHON DRIVER 7](#_Toc167486178)

[7. C DRIVER 7](#_Toc167486179)

## List of tables

[Table 1 IP Dummy input/output signal description 3](#_Toc140488110)

# INPUT/OUTPUT SIGNAL DESCRIPTION

Table 1 IP Dummy input/output signal description

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | Bitwidth | Direction | Description |
| **General signals** | | | |
| clk | 1 | Input | System clock |
| rst\_a | 1 | Input | Asynchronous system reset, low active |
| en\_s | 1 | Input | Enables the IP Core functionality |
| **AIP Interface** | | | |
| data\_in | 32 | Input | Input data for configuration and processing |
| data\_out | 32 | Output | Output data for processing results and status |
| conf\_dbus | 5 | Input | Selects the bus configuration to determine the information flow from/to the IP Core |
| write | 1 | Input | Write indication, data from the data\_in bus will be written into the AIP Interface according to the conf\_dbus value |
| read | 1 | Input | Read indication, data from the AIP Interface will be read according to the conf\_dbus value. The data\_out bus shows the new data read. |
| start | 1 | Input | Initializes the IP Core process |
| int\_req | 1 | Output | Interruption request. It notifies certain events according to the configurated interruption bits. |
| **Core signals** | | | |
| memX\_addr | 5 | output | Data memory address X |
| dataX | 8 | Input | Data X |
| sizeX | 5 | Input | Size of X |
| memY\_addr | 5 | output | Data memory address Y |
| dataY | 8 | Input | Data Y |
| sizeX | 5 | Input | Size of Y |
| memZ\_addr | 6 | output | Data memory address z |
| dataZ | 16 | Input | Data Z |
| Start | 1 | Input | Convolution start signal |
| Shape | 1 | Input | Convolution type FULL=0 , SAME =1 |
| Busy | 1 | output | Status of co-processor 1=bussy, 0= available |
| Done | 1 | output | Co-processor task status 1= completed, 0= in process |

# THEORY OF OPERATION

The convolution core receives as input the data from two inputs memories X and Y with configurable size and the type of convolution FULL or SAME, after of receiving the start command, according to the data obtained from test performed, it has an area of 88 register, and works at maximum frequency of 106.25 MHz. The operation of two memories X of size 10 and Z of size 5, the convolution takes 230 clock cycles.

# AIP interface registers and memories description

## Status register

Config: STATUS

Size: 32 bits

Mode: Read/Write.

This register is divided in 3 sections, see Figure 5.1:

* **Status Bits**: These bits indicate the current state of the core.
* **Interruption Flags:** These bits are used to generate an interruption request in the *int\_req* signal of the AIP interface.
* **Mask Bits**: Each one of these bits can enable of disable the interruption flags.



Figure 5.1 IP Dummy status register

Bits 31:24 – Reserved, must be kept cleared.

Bits 23:17 – Reserved Mask Bits for future use and must be kept cleared.

Bit 16 – **MSK:** mask bit for the DN (Done) interruption flag. If it is required to enable the DN interruption flag, this bit must be written to 1.

Bits 15:9 – Reserved Status Bits for future use and are read as 0.

Bit 8 – **BSY**: status bit “**Busy**”.

Reading this bit indicates the current IP Dummy state:

0: The IP Dummy is not busy and ready to start a new process.

1: The IP Dummy is busy, and it is not available for a new process.

Bits 7:1 – Reserved Interrupt/clear flags for future use and must be kept cleared.

Bit 0 – **DN**: interrupt/clear flag “**Done**”

Reading this bit indicates if the IP Dummy has generated an interruption:

0: interruption not generated.

1: the IP Dummy has successfully finished its processing.

Writing this bit to 1 will clear the interruption flag DN.

## Configuration register

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | | 2 | | 1 | 0 |
| - Y/Sizes[10:6] - X/Sizes[5:1]-Shape[0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | | w | | w | w |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ZY | ZY | ZY | ZY | ZY | ZY | ZX | ZX | | ZX | | ZX | Shape |

Tabla 1 Convolution configuration register..

Bit 0- It is reserved for FULL or SAME convolution operation type.

1: SAME

0: FULL

Bits 5:1- Reserved for the size of memory X.

Bits 10:6- Reserved for the size of memory Y.

## Input data memory X

Config: memX

Size: Nx32 bits

Mode: Write

This is the X memory used for the convolution operation between the X and Y memory.

## Input data memory Y

Config: memY

Size: Nx32 bits

Mode: Write

This is the Y memory used for the convolution operation between the X and Y memory.

## Output data memory Z

Config: memZ

Size: Nx32 bits

Mode: Read

This is the Z memory used to store the result of the convolution operation between X and Y memory.

# PYTHON DRIVER

# C DRIVER