# Brief Description of MATRIX\_MUL\_IP\_CORE

MATRIX\_MUL\_IP\_CORE is an IP core developed in VHDL. It can perform multiplication of two square Matrices of any size. The IP core is parameterizable.

The names of the Matrices to be multiplied by the IP Core are denoted as P Matrix and G Matrix. P matrix is on the right hand side of the multiplication sign (P \* G) while G matrix is on the left hand side. P matrix is loaded into the IP Core while G matrix is supplied externally during multiplication.

Before the multiplication is performed, P matrix must be loaded into the IP Core one element per clock cycle. After P matrix is loaded multiplication can commence. During multiplication the IP core expects to receive element from G matrix. The IP Core automatically generates the address for the element it requires from the G matrix. The IP core can perform the following matrix multiplication below.

1. P Matrix \* G Matrix
2. P Matrix \* G Matrix transposed
3. P Matrix transposed \* G Matrix, and
4. P Matrix transposed \* G Matrix transposed.

After multiplication has completed, the result can be offloaded from the IP core, one element per clock circle.

# Pinout

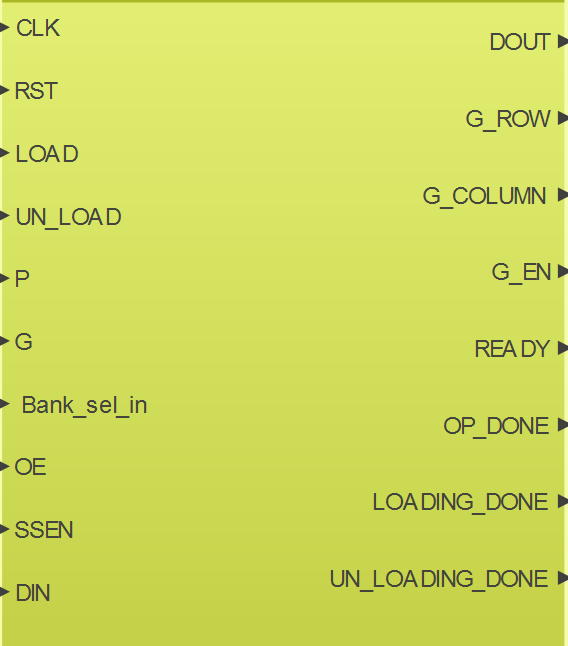


Figure 1: Core Schematic Symbol

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERICS** | | | |
| **Signal** | **type** | **Edit** | **Description** |
| COLUMN\_TOTAL | Integer | Can Edit | This generic Specifies the Dimension of the Matrix. Change the value to match the number of columns and rows your matrix will have. |
| OPCODE\_WIDTH | Integer | Do not Edit | This generic specifies the size of the OPCODE |
| ADDR\_WIDTH | Integer | Do not Edit | This generic specifies the size of the address |
| DATA\_WIDTH | Integer | Do not Edit | This generic specifies the size of the Data bus |
| DATA\_WIDE\_WIDTH | Integer | Do not Edit | This is used internally by the IP core. |
| **PORTS** | | | |
| **Signal** | **Direction** | **Optional** | **Description** |
| CLK | Input |  | Clock signal input - active when rising edge |
| RST | Input |  | Reset (synchronous) - active when high. Asserting RST synchronously with CLK resets only FSM in IP core, all other elements in IP Core remain intact. |
| LOAD | Input |  | LOAD - active when high. Asserting LOAD puts the IP core in the *loading state* and then data (for P Matrix) can be fed into the IP Core for storage in internal Block RAM. |
| UNLOAD | Input |  | UNLOAD - active when high. Asserting UNLOAD puts the IP core in the *unloading state* and the entire matrix data in internal Block RAM of the IP core is offloaded through DOUT |
| P | Input |  | P Matrix - Setting P to high will multiply P matrix transposed otherwise P matrix is multiplied in normal form. |
| G | Input |  | G Matrix - Setting G to high will multiply G matrix transposed otherwise G matrix is multiplied in normal form. |
| Bank\_sel\_in | Input |  | Bank select - When set to high the lower bank is selected, when set to low the upper bank is selected |
| DIN(DATA\_WIDTH-1 : 0) | Input |  | Data input Port - Serves as data input to IP Core. Max and optimal data width is 18 bits. This port is used to load P matrix data into the IP core before multiplication, and it is also used to supply G matrix data (one element per clock cycle) during Multiplication. |
| DOUT(DATA\_WIDTH-1 : 0) | Output |  | Data output Port - Serves as data output from IP Core. Max and optimal data width is 18 bits. Data is offloaded from IP Core, one element per clock cycle. |
| G\_ROW(ADDR\_WIDTH-1 : 0) | Output |  | G Matrix memory row address. This is generated automatically during multiplication. This should be used with G matrix memory row input. |
| G\_COLUMN(COLUMN\_TOTAL-1 : 0) | Output |  | G Matrix column address. This is generated automatically during multiplication. This should be used with G matrix memory column input. |
| G\_O\_EN | Output | Yes | G memory output enable. This signal should be used to enable the output of the G Matrix RAM Memory. If G matrix is not in separate RAM this signal can be ignored. |
| READY | Output |  | READY - This signal is set high when the IP core is ready to receive data for loading into the internal Block RAM. DIN should hold the first valid data at the time this signal is set high. It is also set high immediately the first data offloaded from the IP core is present in the output port DOUT. |
| OP\_DONE | Output |  | Operation done (Complete) - This signal is set high when the IP core has successfully multiplied P matrix by G matrix |
| LOADING\_DONE | Output |  | LOADING DONE (Complete) - This signal is set high when the IP core has finished loading the Block RAMs with P matrix data. |
| UNLOADING\_DONE | Output |  | UNLOADING DONE (Complete) - This signal is set high when the IP core has finished off-loading the Block RAMs. |

# IP Core Overview

The IP core is composed of 3 key sub components.

1. DSP48 block

2. Block RAM (BRAM)

3. FSM unit.

## DSP48 block:

The Number of DSP48 block available in the IP core is a variable and it is equivalent to the number of Columns of the Matrix (The value specified on the *COLUMN\_TOTAL* generic). In this IP the DSP48 blocks are configured mainly for Multiplier and Accumulate operations (MACC). The data width is fixed at 18 bits and this is because for multiplication operation the input data width are as follows: (port A = 25 bits, and Port B = 18 bits). In other to maintain consistency DSP is fixed to operate with 18 bits data width for both inputs.

## Block RAM:

The Number of Block RAMs used in the IP core is a variable and it is equivalent to the number of Columns of the Matrix (The value specified on the *COLUMN\_TOTAL* generic). Actually the block RAM has a total width of 36 bits so one (1) block RAM can accommodate two (2) Columns.

The block RAM is a dual port Block RAM with separate *Read* and *Write* addresses. The address range of the block RAM is 10 bits (0 to 1023). In the block RAM each memory location is configured as 18 bit word but can accommodate 36 bits maximum. For this IP core 18 bits is the recommended size because of limitation from the DSP48 block.

The block RAM is divided into two (2) banks (upper and lower banks) and the user can specify which bank data should be loaded to, or offloaded from. This is very useful when performing matrix multiplication with result of previous operation.

## FSM Unit:

There is just one FSM unit that controls the entire IP core. The FSM unit is responsible for loading data into the BRAM and offloading data from the BRAM. It also generates, control signals for the DSP48 block and IP core as well as the flags for the IP core. The FSM can be controlled by the user based on the following control input pins.

1. **RST**
2. **LOAD**
3. **UN\_LOAD**
4. **P** and **G**

All input control signals to the IP core are active high. When the **RST** input is asserted only the FSM is placed in the reset state all other components remain intact. Typically before any operation is carried out with the IP core it is recommended to reset the FSM in the IP core using the RST input then before releasing the FSM, the desired state of the control input signals should be configured, after that the FSM can then be released from *reset\_state*. The FSM can be released from *reset\_state* by setting the RST input to low state.

When the **LOAD** input is asserted the FSM goes into the *Loading\_state*. ***Please note\*:*** *Assuming the IP core is in reset\_state already, The LOAD input should be set high before the RST input is set low, otherwise the FSM may not enter the loading\_state*. When the IP core is ready to start receiving data it will set the **READY** output signal to high state, at this point the user can start sending data to the IP core through the DIN input port. When the READY output signal is set, the first valid data should be present in DIN port.

When the UN\_LOAD input is asserted the FSM goes into the *unload\_state.* ***Please note\*****: The LOAD input must be set to low otherwise the FSM will not respond to the UN\_LOAD input. Assuming the IP core is in reset\_state already, make sure LOAD signal is set low then set UN\_LOAD high, after that release IP from reset\_state by setting RST to low.* When valid data from BRAM reaches the DOUT port, the IP core will set the READY output signal to high state, at this point the user can start receiving valid data from the **DOUT** output port. The READY output signal is set in the same clock cycle that valid data reaches **DOUT.**

**P** and **G** input signalsare used to specify what type of matrix multiplication the IP core should perform. The table below summarizes the configuration of PG and the resulting multiplication.

|  |  |  |
| --- | --- | --- |
| **P** | **G** | **Action** |
| 0 | 0 | P Matrix multiplied by G Matrix |
| 0 | 1 | P Matrix multiplied by transposed G Matrix |
| 1 | 0 | Transposed P Matrix Multiplied by G Matrix |
| 1 | 1 | Transposed P Matrix multiplied by Transposed G Matrix |

# Detailed Step by step setup and usage of IP Core

## LOAD (Input)

This is active high. When asserted the FSM in the IP Core goes into the loading state to Load P Matrix data into the IP core. As soon as the IP Core is ready to start receiving data the READY flag is set high and at that same clock circle the first data of P Matrix should be present in the DIN input port of the IP core. The order of data loading is row wise, first element of each row to last element of the row, then next row.

Before putting the IP core in the loading state the bank should be specified.

Signals that Influence LOAD input are: RST, Bank\_sel\_in

The following flags serve as feedback: READY, LOADING\_DONE

### Typical sequence of micro operations

**Step 1:**

RST => set high (at least 3 clock cycles)

LOAD => set high

Bank\_sel\_in: set to proper bank.

Upper bank => '0', Lower bank => '1' (These values are for Write address but are inverted for Read address)

**Step 2:**

RST => Set low

READY => wait until it is High, start sending data immediately

**Step 3:**

LOADING\_DONE: Becomes high when data completely loaded