



Lecture 16

Introduction to VHDL Project



Important Notes

- **VHDL Project – Matrix Processing Core Design**
 - $D = A \times B + C$
 - 32×32
 - 16-bit signed integer
- **Reference Design – Matrix Multiplier Core Design**
 - 4×4
 - 8-bit signed integer
- **Three 2-hour sessions of VHDL Project Workshop**
 - 09:00 – 11:00 on 27th April, 4th May and 11th May
 - No lecture, but you are strongly encouraged to attend
 - I will be there to help you, but don't do the design for you
- **Submission**
 - Moodle submission (See detail submission requirements in the project specification)
 - Deadline – **3:00pm, Friday, 21st May 2021**
- **Late submission**
 - A penalty of 5% per working day will be imposed for the late submission.



Project Marking Scheme

- **Design, Implementation and Verification (15%)**
 - It assesses VHDL code compilation, behavioural simulation, synthesis, coding quality and design efficiency.
- **10-page project report (15%)**
 - It should cover project design, synthesis and simulation, project discussion, conclusions and further work.



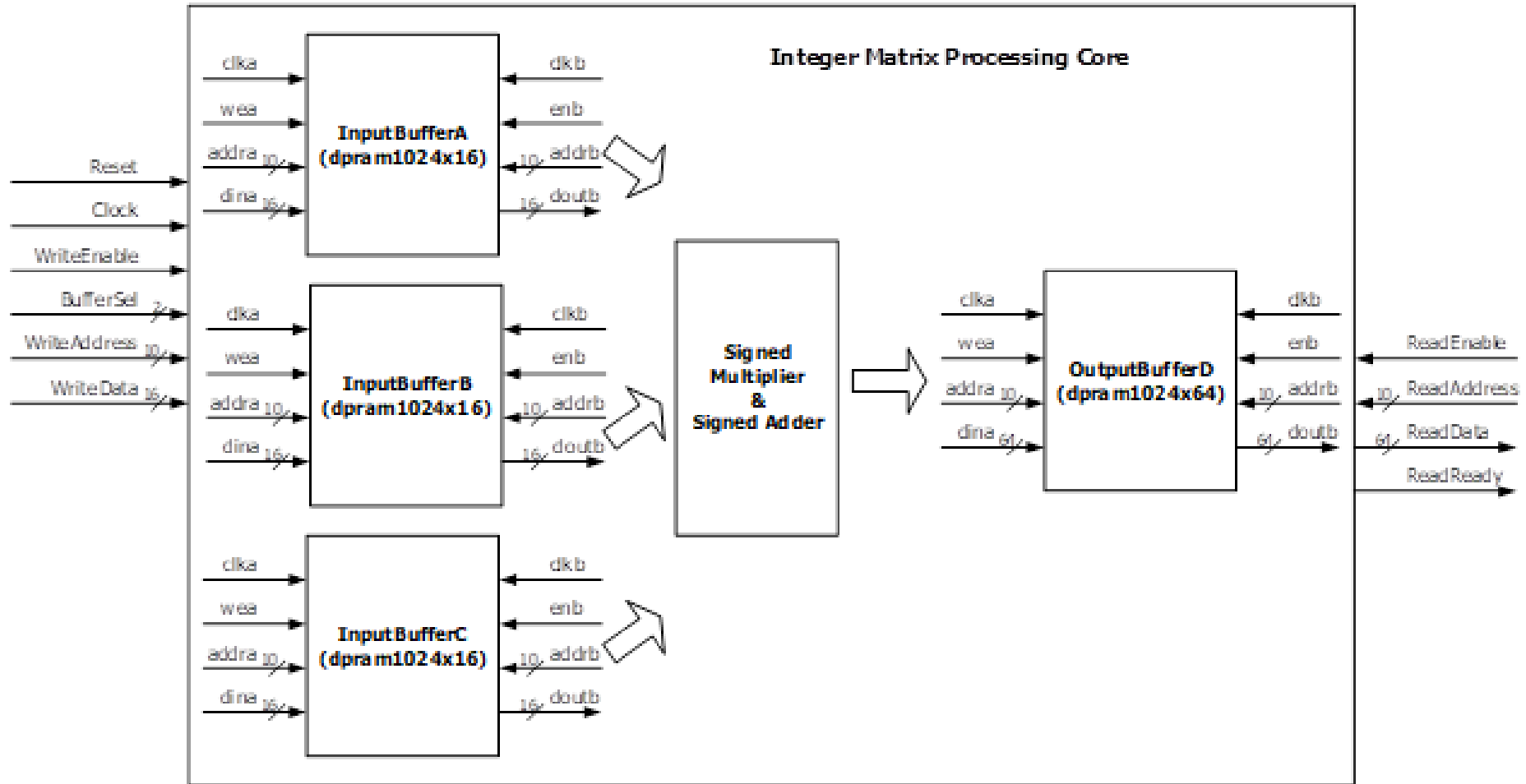
Matrix Multiplication and Addition

$$\begin{array}{c} \mathbf{A} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{132} \\ a_{21} & a_{22} & a_{23} & \dots & a_{232} \\ a_{31} & a_{32} & a_{33} & \dots & a_{332} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{321} & a_{322} & a_{323} & \dots & a_{3232} \end{bmatrix} \end{array} \times \begin{array}{c} \mathbf{B} \\ \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{132} \\ b_{21} & b_{22} & b_{23} & \dots & b_{232} \\ b_{31} & b_{32} & b_{33} & \dots & b_{332} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{321} & b_{322} & b_{323} & \dots & b_{3232} \end{bmatrix} \end{array} + \begin{array}{c} \mathbf{C} \\ \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{132} \\ c_{21} & c_{22} & c_{23} & \dots & c_{232} \\ c_{31} & c_{32} & c_{33} & \dots & c_{332} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{321} & c_{322} & c_{323} & \dots & c_{3232} \end{bmatrix} \end{array} = \begin{array}{c} \mathbf{D} \\ \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{132} \\ d_{21} & d_{22} & d_{23} & \dots & d_{232} \\ d_{31} & d_{32} & d_{33} & \dots & d_{332} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{321} & d_{322} & d_{323} & \dots & d_{3232} \end{bmatrix} \end{array}$$

$$d_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + a_{i3} \times b_{3j} + \dots + a_{i32} \times b_{32j} + c_{ij}; i = 1, 2, 3, \dots, 32; j = 1, 2, 3, \dots, 32$$



Matrix Processing Core





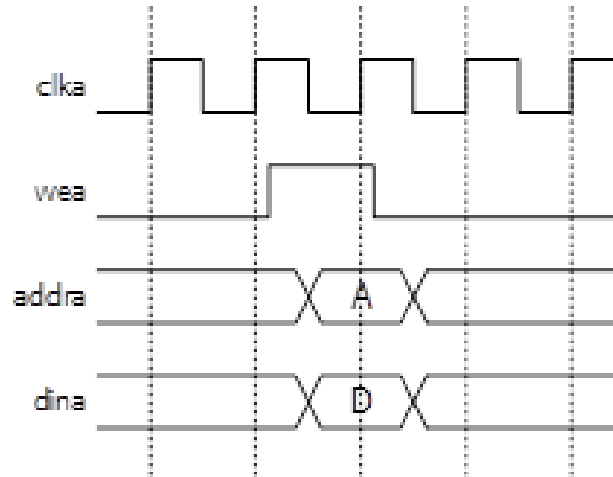
Buffer Data Organisation

InputBufferA (dpram 1024x16)		InputBufferB (dpram 1024x16)		OutputBufferC (dpram 1024x64)		OutputBufferD (dpram 1024x64)	
0	a11	0	b11	0	c11	0	d11
1	a12	1	b12	1	c12	1	d12
2	a13	2	b13	2	c13	2	d13
⋮		⋮		⋮		⋮	
31	a132	31	b132	31	c132	31	d132
32	a21	32	b21	32	c21	32	d21
33	a22	33	b22	32	c22	32	d22
34	a23	34	b23	34	c23	34	d23
⋮		⋮		⋮		⋮	
63	a232	63	b232	63	c232	63	d232
64	a31	64	b31	64	c31	64	d31
65	a32	65	b32	65	c32	65	d32
66	a33	66	b33	66	c33	66	d33
⋮		⋮		⋮		⋮	
95	a332	95	b332	95	c332	95	d332
⋮		⋮		⋮		⋮	
992	a321	992	b321	992	c321	992	d321
993	a322	993	b322	993	c322	993	d322
994	a323	994	b323	994	c323	994	d323
⋮		⋮		⋮		⋮	
1023	a3232	1023	b3232	1023	c3232	1023	d3232

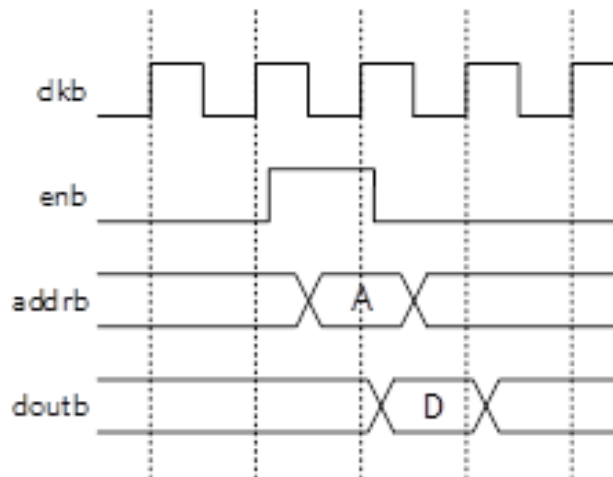


Timing for Write to/Read from DPRAM

■ Write Operation



■ Read Operation





Referenece Design

■ Matrix Multiplication

- Size of 4 x 4, 8 bits for each matrix elements of the input matrixes

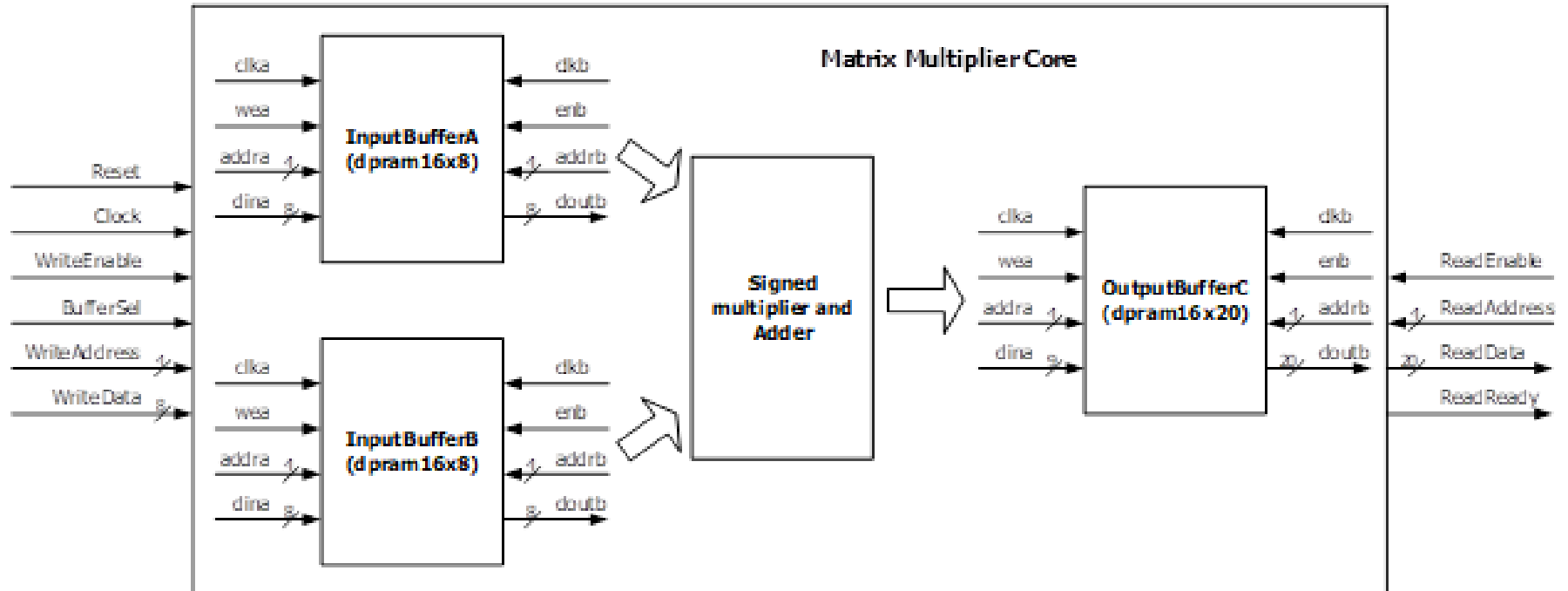
$$\begin{array}{c} \mathbf{A} \\ \left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array} \right] \end{array} \times \begin{array}{c} \mathbf{B} \\ \left[\begin{array}{cccc} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{array} \right] \end{array} = \begin{array}{c} \mathbf{C} \\ \left[\begin{array}{cccc} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{array} \right] \end{array}$$

$$C_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + a_{i3} \times b_{3j} + a_{i4} \times b_{4j}; i = 1, 2, 3, 4; j = 1, 2, 3, 4$$



Referenece Design

■ Matrix Multiplier Core Design





Referenece Design

■ Buffer Organisation

InputBufferA (dpram16x8)		InputBufferB (dpram16x8)		OutputBufferC (dpram16x20)	
0	a11	0	b11	0	c11
1	a12	1	b12	1	c12
2	a13	2	b13	2	c13
3	a14	3	b14	3	c14
4	a21	4	b21	4	c21
5	a22	5	b22	5	c22
6	a23	6	b23	6	c23
7	a24	7	b24	7	c24
8	a31	8	b31	8	c31
9	a32	9	b32	9	c32
10	a33	10	b33	10	c33
11	a34	11	b34	11	c34
12	a41	12	b41	12	c41
13	a42	13	b42	13	c42
14	a43	14	b43	14	c43
15	a44	15	b44	15	c44



IP Catalog to Generate Dual Port RAM

IntMatrixMulCore_4x4_8bits - [C:/VHDL/IntMatrixMulCore_4x4_8bits/IntMatrixMulCore_4x4_8bits.xpr] - Vivado 2018.3

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access

Flow Navigator

PROJECT MANAGER - IntMatrixMulCore_4x4_8bits

PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION

- Run Simulation

Sources

- Design Sources (1)
 - IntMatMulCore(IntMatMulCo
- Constraints
- Simulation Sources (4)
- Utility Sources

Hierarchy IP Sources Li > >>

IP Properties

- Block Memory Generator

Project Summary x IP Catalog x

Cores | Interfaces

Search: Q-

Name	AXI4
Debug & Verification	
Memories & Storage Elements	
FIFOs	
ECC	
Memory Interface Generators	
RAMs & ROMs	
RAMs & ROMs & BRAM	
Block Memory Generator	AXI4
Basic Elements	



Dual Port RAM Generation

Project Summary x IP Catalog x

Cores | Interfaces

Q

≡

≡

✖

⚙

🔑

⚙

ℹ

Search:

Name	AXI4	Status	License	VLNV
Memories & Storage Elements				
ECC		Production	Included	xilinx.com:ip:ecc:2.0
FIFOs				
Memory Interface Generators				
RAMs & ROMs				
RAMs & ROMs & BRAM				
Block Memory Generator	AXI4	Production	Included	xilinx.com:ip:blk_mem_gen:8.4

Details

Name:

Block Memory Generator

Version:

8.4 (Rev. 2)

Interfaces:

AXI4

Description:

The Xilinx LogiCORE IP Block Memory Generator replaces the Dual Port Block Memory and Single Port Block Memory LogiCORE IP blocks.



Dual Port RAM Generation (Basic)

Re-customize IP

Block Memory Generator (8.4)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol | Power Estimation

☒ Show disabled ports

Component Name: dpram16x8

Basic | Port A Options | Port B Options | Other Options | Summary

Interface Type: Native ☐ Generate address interface with data bus

Memory Type: Simple Dual Port RAM ☐ Common Clock

ECC Options

ECC Type: No ECC

☐ Error Injection Pins: Single Bit Error Injection

Write Enable

☐ Byte Write Enable

Byte Size (bits): 9

+ AXI_SLAVE_S_AXI

+ AXI4_SLAVE_S_AXI

+ BRAM_PORTA

+ BRAM_PORTB

reg0a2 sbkerr

reg0a1 dbkerr

reg0a0 m0akerr000





VHDL



Dual Port RAM Generation (Basic)

Re-customize IP

Block Memory Generator (8.4)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol | Power Estimation

☒ Show disabled ports

Component Name: dpram16x8

Basic | Port A Options | **Port B Options** | Other Options | Summary

Memory Size

Port B Width:

Port B Depth: 16

The Width and Depth values are used for Read Operation in Port B

Operating Mode: Enable Port Type:

Port B Optional Output Registers

☐ Primitives Output Register ☐ Core Output Register

☐ SoftECC Output Register ☐ REGCEB Pin

Port B Output Reset Options

AXI_SLAVE_S_AXI

AXILite_SLAVE_S_AXI

BRAM_PORTA

BRAM_PORTB

regceb sbkerr

regceb dbkerr

injectsbkerr rdaddress[3:0]



Dual Port RAM Generation (.VHO)

File Explorer interface showing the directory structure for Dual Port RAM Generation (.VHO).

Navigation path: ts > IntMatrixMulCore_4x4_8bits > IntMatrixMulCore_4x4_8bits.ip_user_files > ip > dpram16x8

Name	Date modified	Type	Size
dpram16x8.veo	25/02/2021 09:02	VEO File	4 KB
dpram16x8.vho	25/02/2021 09:02	VHO File	4 KB
dpram16x8_stub.v	25/02/2021 08:56	V File	2 KB
dpram16x8_stub.vhdl	25/02/2021 08:56	VHDL File	2 KB



Dual Port RAM Generation (.VHO)

```
51
52 -- The following code must appear in the VHDL architecture header.
53
54 ----- Begin Cut here for COMPONENT Declaration ----- COMP_TAG
55 COMPONENT dpraml6x8
56 PORT (
57     clka : IN STD_LOGIC;
58     wea  : IN STD_LOGIC_VECTOR(0 DOWNTO 0);
59     addra : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
60     dina  : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
61     clk b : IN STD_LOGIC;
62     enb   : IN STD_LOGIC;
63     addr b : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
64     dout b : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
65 );
66 END COMPONENT;
67 -- COMP_TAG_END ----- End COMPONENT Declaration -----
68
69 -- The following code must appear in the VHDL architecture
70 -- body. Substitute your own instance name and net names.
71
72 ----- Begin Cut here for INSTANTIATION Template ----- INST_TAG
73 your_instance_name : dpraml6x8
74 PORT MAP (
75     clka => clka,
76     wea  => wea,
77     addra => addra,
78     dina  => dina,
79     clk b => clk b,
80     enb   => enb,
81     addr b => addr b,
82     dout b => dout b
83 );
84 -- INST_TAG_END ----- End INSTANTIATION Template -----
```



Matlab Code to Generate Test Data

 C:\VHDL\IntMatrixMulCore_4x4_8bits\RandomMatrixMulTest_4x4_8bits.m - Notepad

File Edit Search View Encoding Language Settings Tools Macro Run



RandomMatrixMulTest_4x4_8bits.m

```
1
2   clc
3   A = randi([0 9], 4, 4);
4   B = randi([0 9], 4, 4);
5   C = A*B;
6
7   AHex = dec2hex(A', 2);
8   BHex = dec2hex(B', 2);
9   CHex = dec2hex(C', 5);
10  % end
```



Test Data from Matlab

C:\VHDL\IntMatrixMulCore C:\VHDL\IntMatrixMulCore C:\VHDL\IntMatrixMulCore_4x4_8bits\Our

File Edit Search View E File Edit Search View E File Edit Search View Encoding Lan

InputA.txt InputB.txt InputA.txt InputB.txt InputA.txt InputB.txt OutputC_m

1	'05'	1	'02'	1	'00044'
2	'01'	2	'05'	2	'0003A'
3	'01'	3	'01'	3	'00038'
4	'06'	4	'08'	4	'00042'
5	'04'	5	'09'	5	'00085'
6	'07'	6	'09'	6	'0006F'
7	'06'	7	'09'	7	'00074'
8	'07'	8	'08'	8	'0006D'
9	'00'	9	'01'	9	'0003D'
10	'03'	10	'00'	10	'0002B'
11	'02'	11	'00'	11	'00037'
12	'04'	12	'00'	12	'00024'
13	'03'	13	'08'	13	'00039'
14	'05'	14	'04'	14	'0003C'
15	'06'	15	'07'	15	'00030'
16	'00'	16	'03'	16	'00040'



Result Data

```
C:\VHDL\IntMatrixMulCore_4x4_8bits\IntMatrixMulCore_4x4_8bits\IntMatrixMulCore_4x4_8bits.sim\sim_1\behav\sim_1\OutputC.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
[Icons]
InputA.txt x InputB.txt x OutputC_matlab.txt x OutputC.txt x
1 Results
2 Data from Matlab      Data from Simulation
3 00044                000044                TRUE
4 0003A                00003A                TRUE
5 00038                000038                TRUE
6 00042                000042                TRUE
7 00085                000085                TRUE
8 0006F                00006F                TRUE
9 00074                000074                TRUE
10 0006D               00006D                TRUE
11 0003D               00003D                TRUE
12 0002B               00002B                TRUE
13 00037               000037                TRUE
14 00024               000024                TRUE
15 00039               000039                TRUE
16 0003C               00003C                TRUE
17 00030               000030                TRUE
18 00040               000040                TRUE
```