Lecture 16

Introduction to VHDL Project

Important Notes

- **VHDL Project – Matrix Processing Core Design**
  - D = A x B + C
  - 32 x 32
  - 16-bit signed integer
- **Reference Design – Matrix Multiplier Core Design**
  - 4 x 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

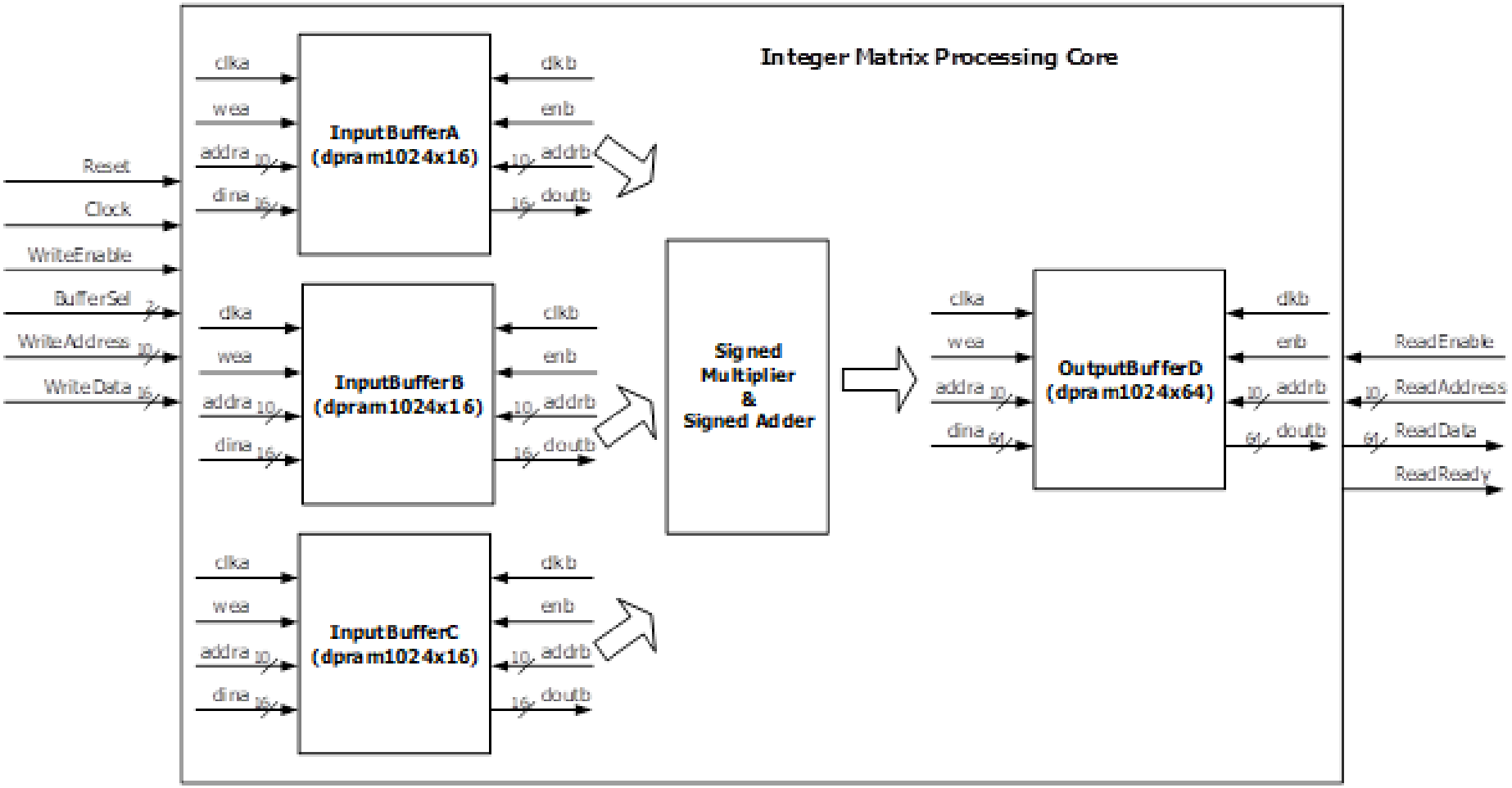
- **Design, Implementation and Verification (15%)**
  - It assesses VHDL code compilation, behavioural simulation, synthesis, coding quality and design efficiency.
- **10-page project report (15%)**
  - It should cover project design, synthesis and simulation, project discussion, conclusions and further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

Matrix Multiplication and Addition

$$\underset{A}{\begin{bmatrix} a11 & a12 & a13 & \dots & a132 \\ a21 & a22 & a23 & \dots & a232 \\ a31 & a32 & a33 & \dots & a332 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a321 & a322 & a323 & \dots & a3232 \end{bmatrix}} \times \underset{B}{\begin{bmatrix} b11 & b12 & b13 & \dots & b132 \\ b21 & b22 & b23 & \dots & b232 \\ b31 & b32 & b33 & \dots & b332 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b321 & b322 & b323 & \dots & b3232 \end{bmatrix}} + \underset{C}{\begin{bmatrix} c11 & c12 & c13 & \dots & c132 \\ c21 & c22 & c23 & \dots & c232 \\ c31 & c32 & c33 & \dots & c332 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c321 & c322 & c323 & \dots & c3232 \end{bmatrix}} = \underset{D}{\begin{bmatrix} d11 & d12 & d13 & \dots & d132 \\ d21 & d22 & d23 & \dots & d232 \\ d31 & d32 & d33 & \dots & d332 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d321 & d322 & d323 & \dots & d3232 \end{bmatrix}}$$

$$dij = ai1 \times b1j + ai2 \times b2j + ai3 \times b3j + \dots + ai32 \times b32j + cij;\ i = 1, 2, 3, \dots, 32;\ j = 1, 2, 3, \dots, 32$$

Matrix Processing Core

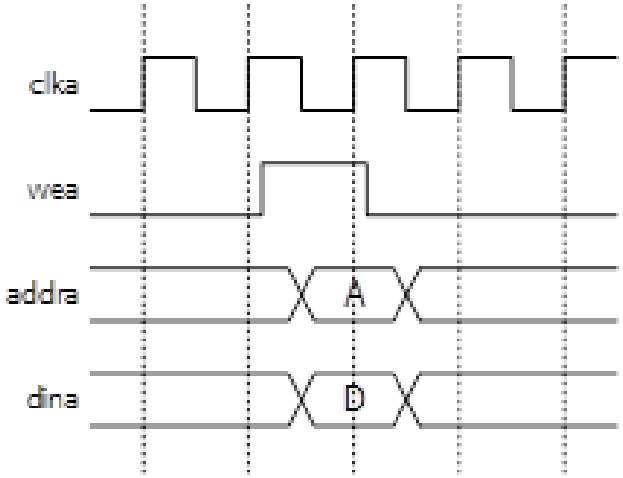


Buffer Data Organisation

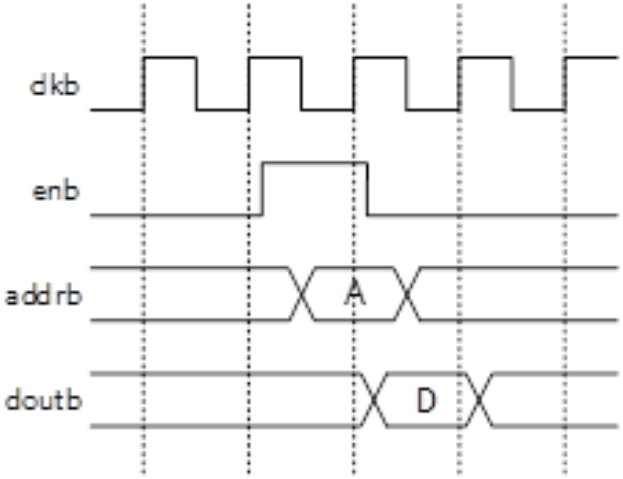
	InputBufferA (dpram1024x16)	InputBufferB (dpram1024x16)	OutputBufferC (dpram1024x64)	OutputBufferD (dpram1024x64)
0	a11	b11	c11	d11
1	a12	b12	c12	d12
2	a13	b13	c13	d13
⋮				
31	a132	b132	c132	d132
32	a21	b21	c21	d21
33	a22	b22	c22	d22
34	a23	b23	c23	d23
⋮				
63	a232	b232	c232	d232
64	a31	b31	c31	d31
65	a32	b32	c32	d32
66	a33	b33	c33	d33
⋮				
95	a332	b332	c332	d332
⋮				
992	a321	b321	c321	d321
993	a322	b322	c322	d322
994	a323	b323	c323	d323
⋮				
1023	a3232	b3232	c3232	d3232

Timing for Write to/Read from DPRAM

- **Write Operation**



- **Read Operation**



Referenece Design

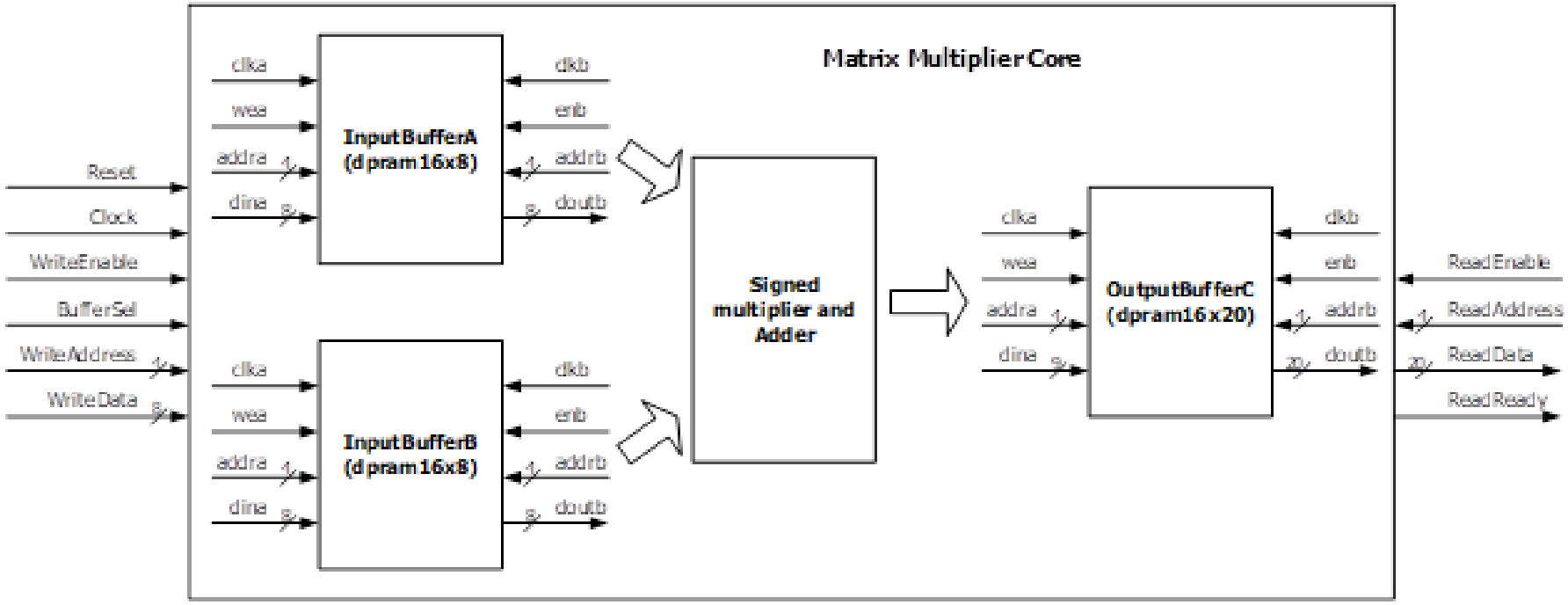
- **Matrix Multiplication**
  - Size of 4 x 4, 8 bits for each matrix elements of the input matrixes

$$\underset{A}{\begin{bmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{bmatrix}} \times \underset{B}{\begin{bmatrix} b11 & b12 & b13 & b14 \\ b21 & b22 & b23 & b24 \\ b31 & b32 & b33 & b34 \\ b41 & b42 & b43 & b44 \end{bmatrix}} = \underset{C}{\begin{bmatrix} c11 & c12 & c13 & c14 \\ c21 & c22 & c23 & c24 \\ c31 & c32 & c33 & c34 \\ c41 & c42 & c43 & c44 \end{bmatrix}}$$

$$Cij = ai1 \times b1j + ai2 \times b2j + ai3 \times b3j + ai4 \times b4j;\ i = 1, 2, 3, 4;\ j = 1, 2, 3, 4$$

Referenece Design

- **Matrix Multiplier Core Design**



Referenece Design

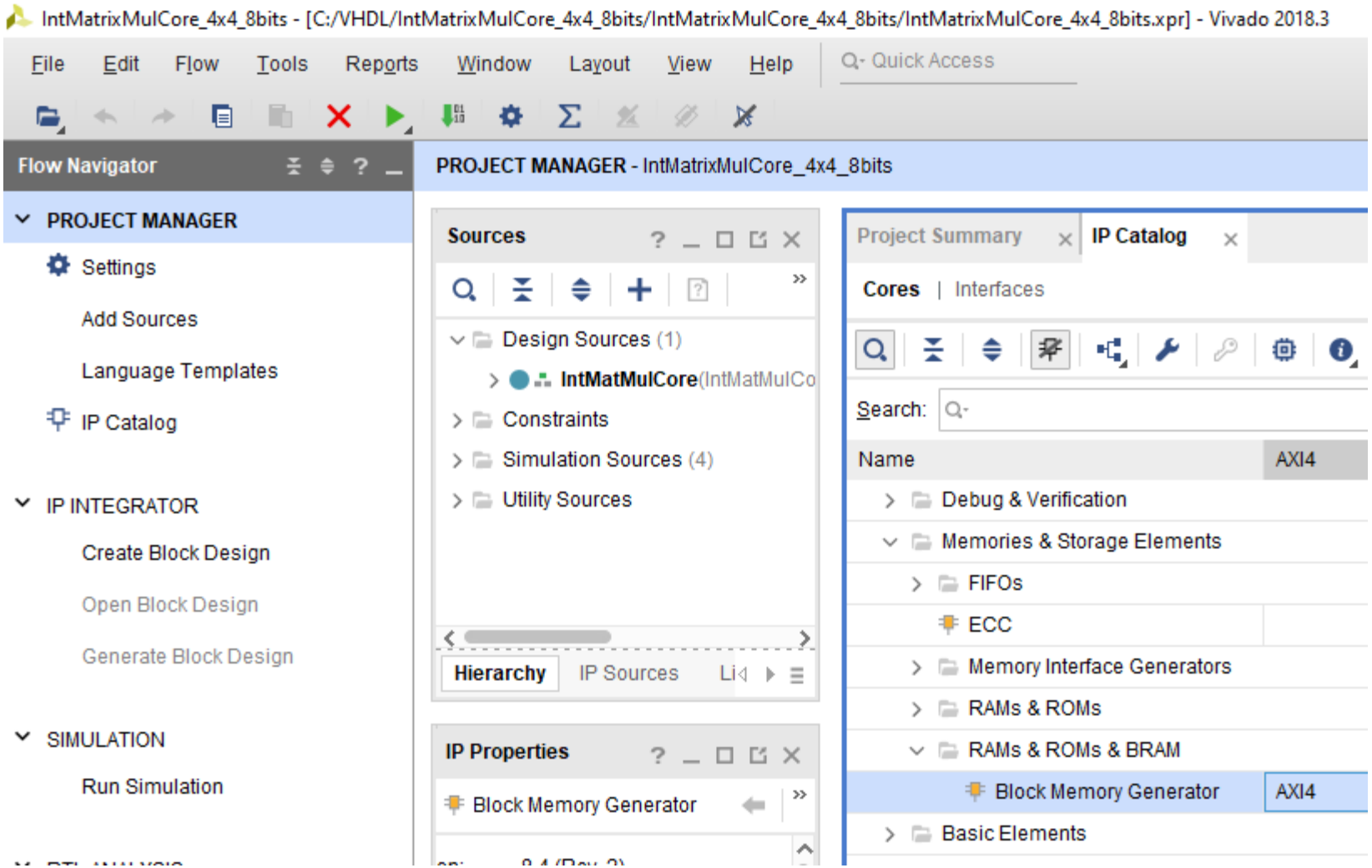
- **Buffer Organisation**

	InputBufferA (dpram16x8)
0	a11
1	a12
2	a13
3	a14
4	a21
5	a22
6	a23
7	a24
8	a31
9	a32
10	a33
11	a34
12	a41
13	a42
14	a43
15	a44

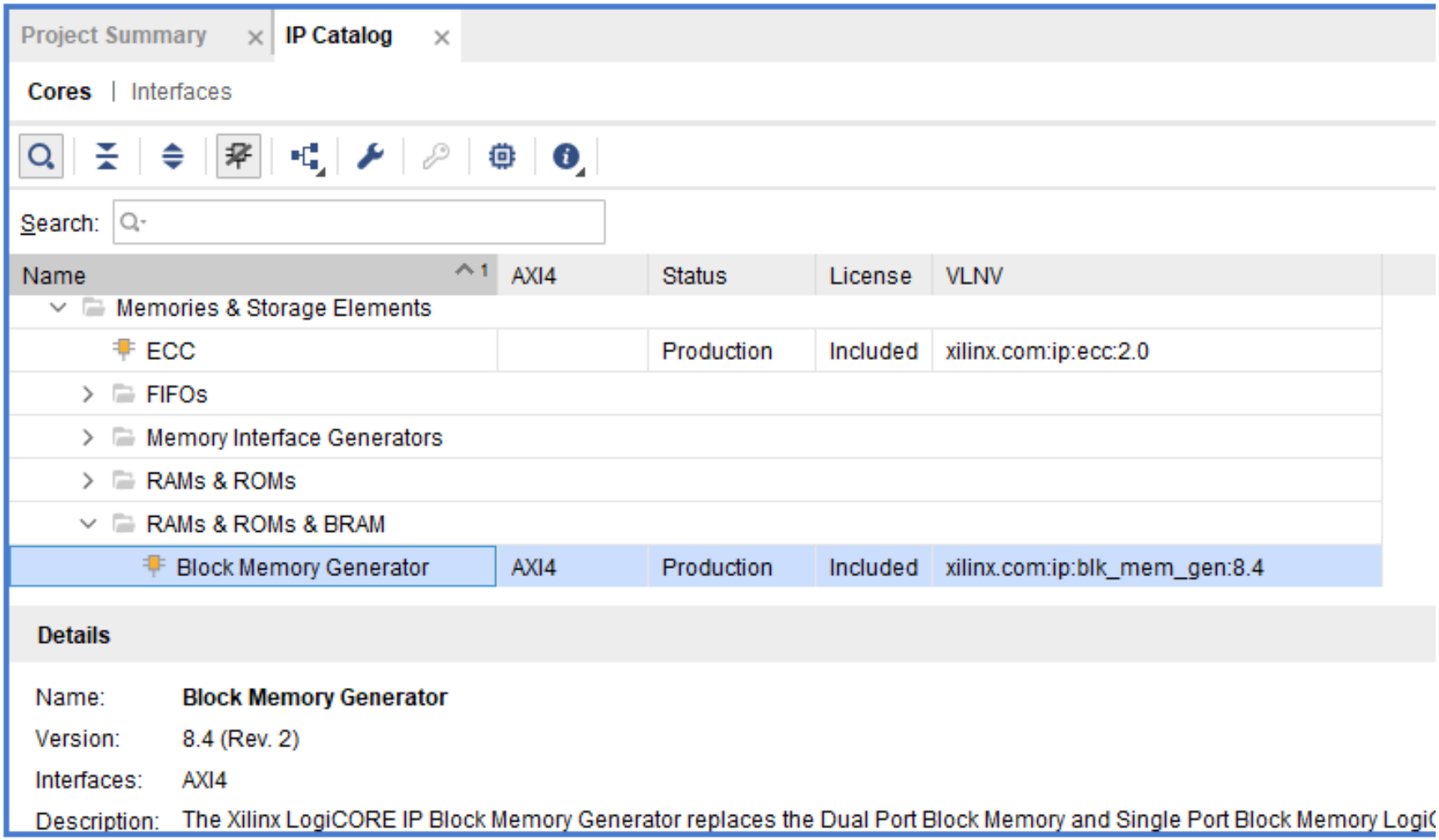
	InputBufferB (dpram16x8)
0	b11
1	b12
2	b13
3	b14
4	b21
5	b22
6	b23
7	b24
8	b31
9	b32
10	b33
11	b34
12	b41
13	b42
14	b43
15	b44

	OutputBufferC (dpram16x20)
0	c11
1	c12
2	c13
3	c14
4	c21
5	c22
6	c23
7	c24
8	c31
9	c32
10	c33
11	c34
12	c41
13	c42
14	c43
15	c44

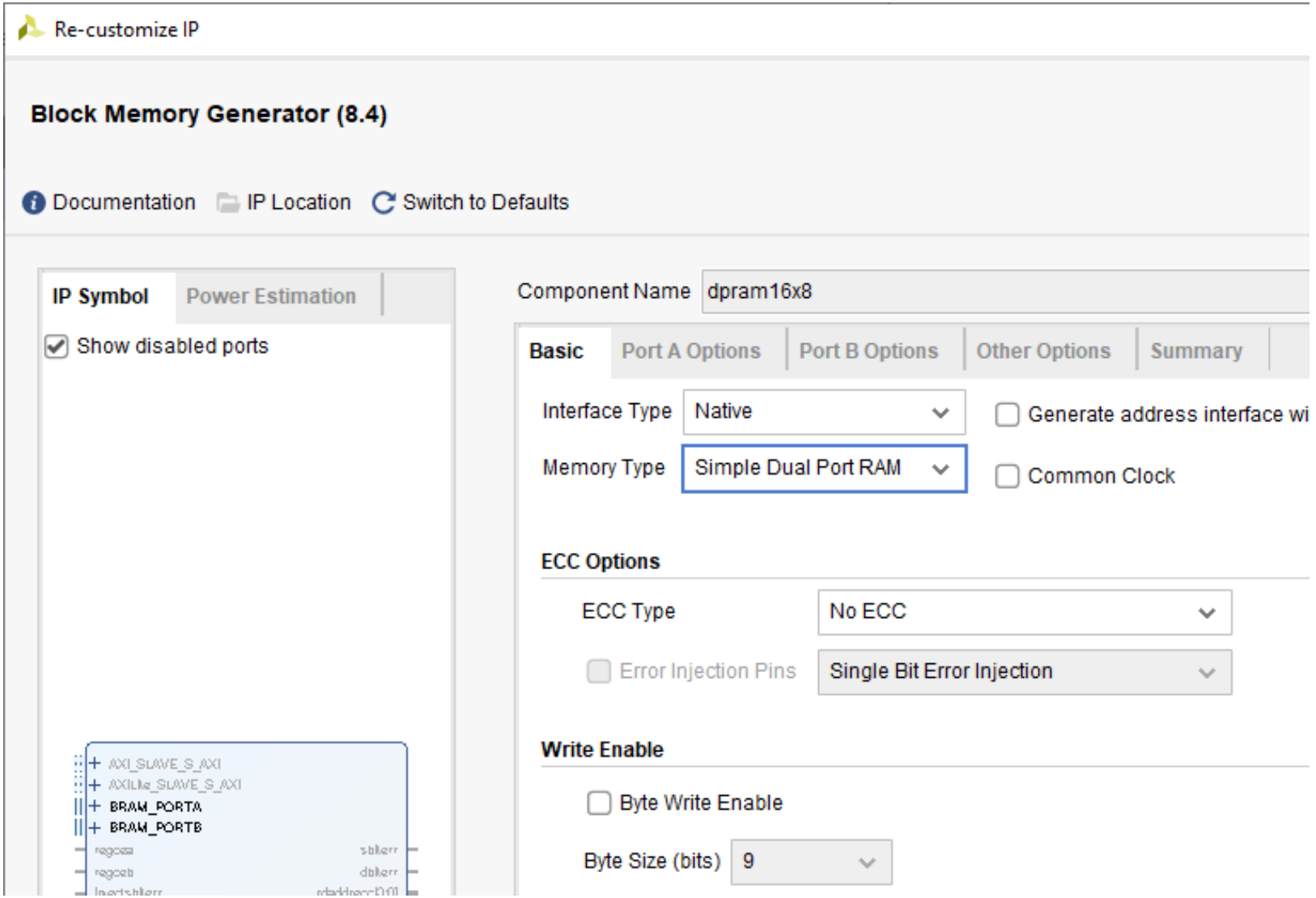
IP Catalog to Generate Dual Port RAM



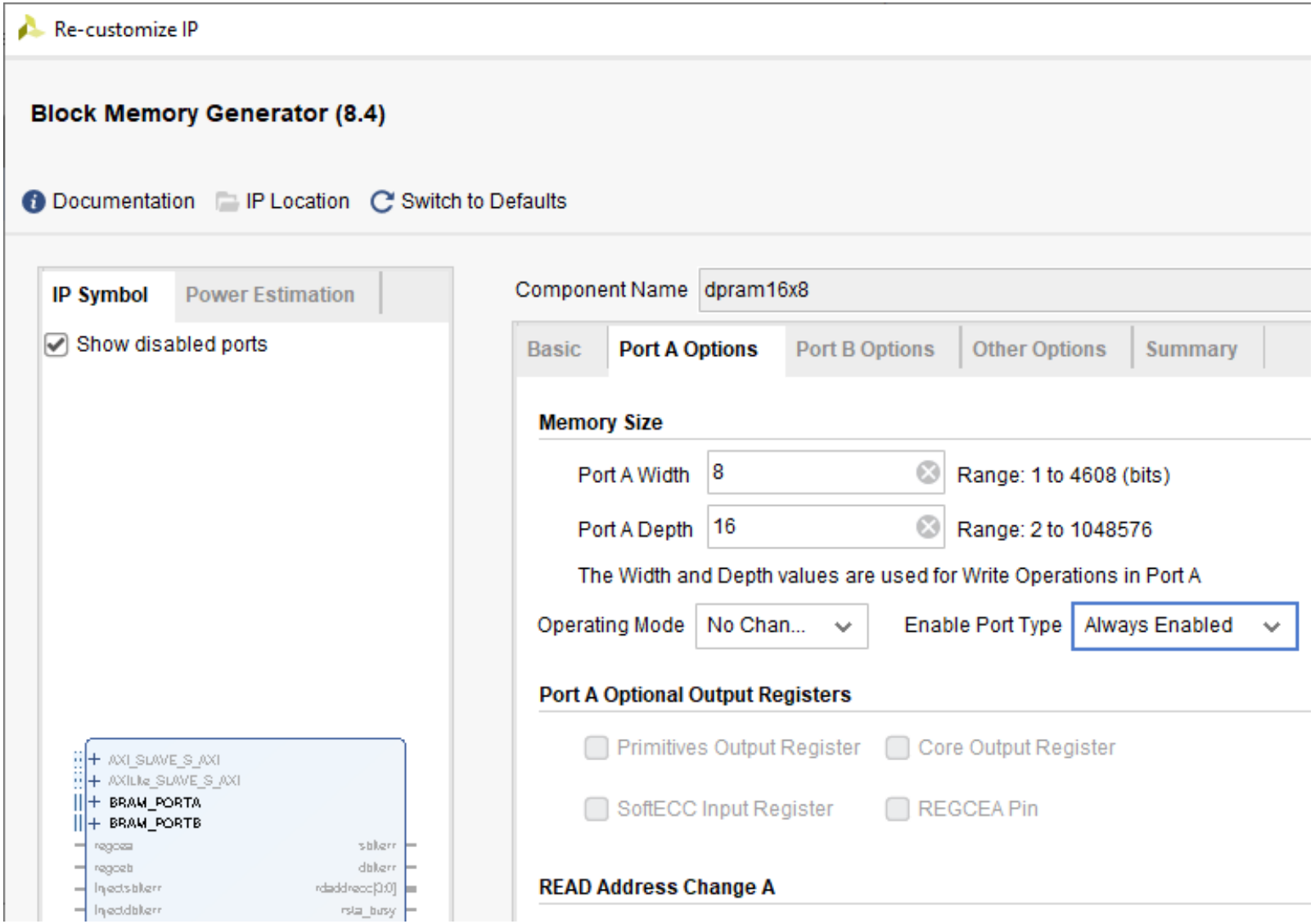
Dual Port RAM Generation



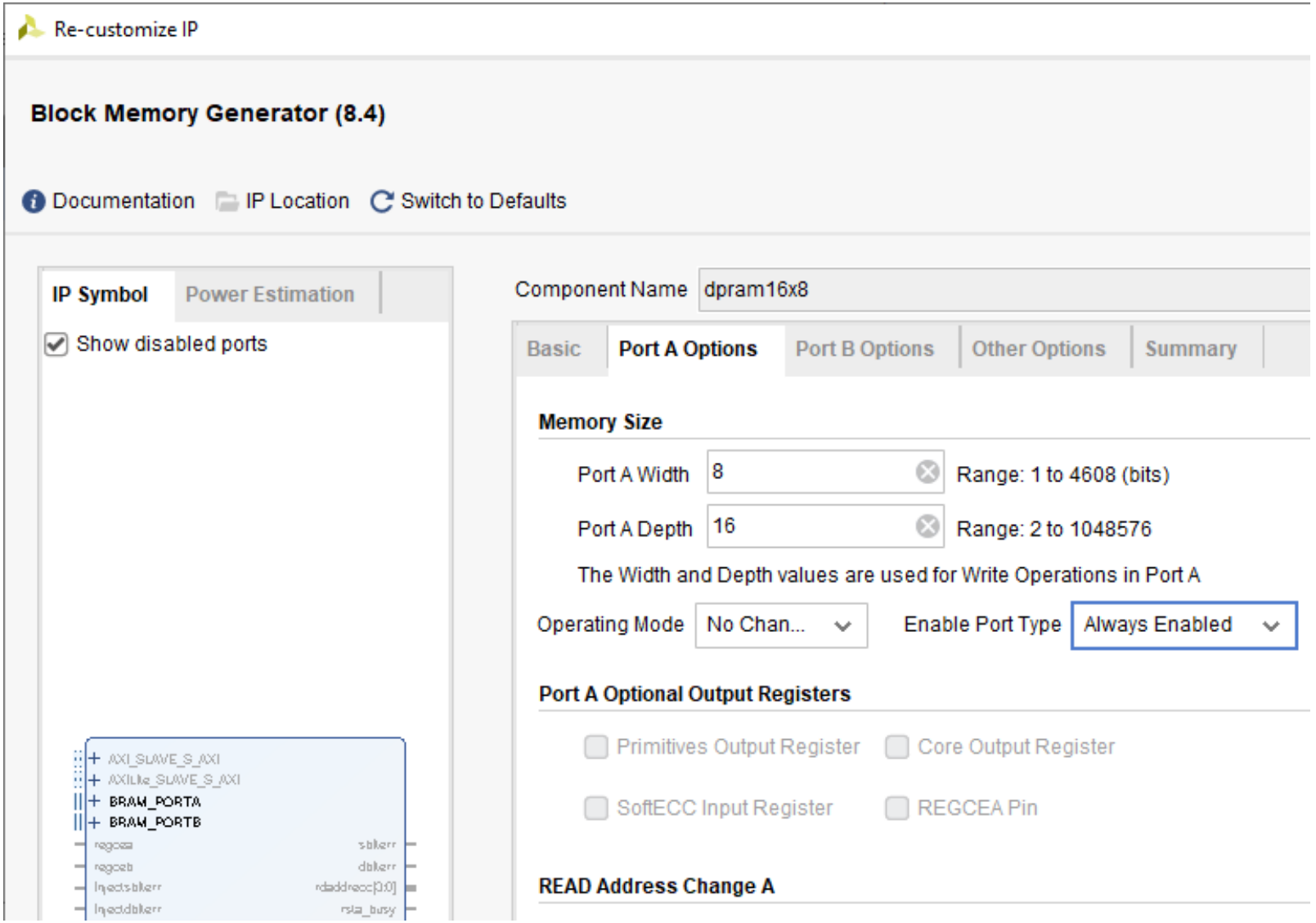
Dual Port RAM Generation (Basic)



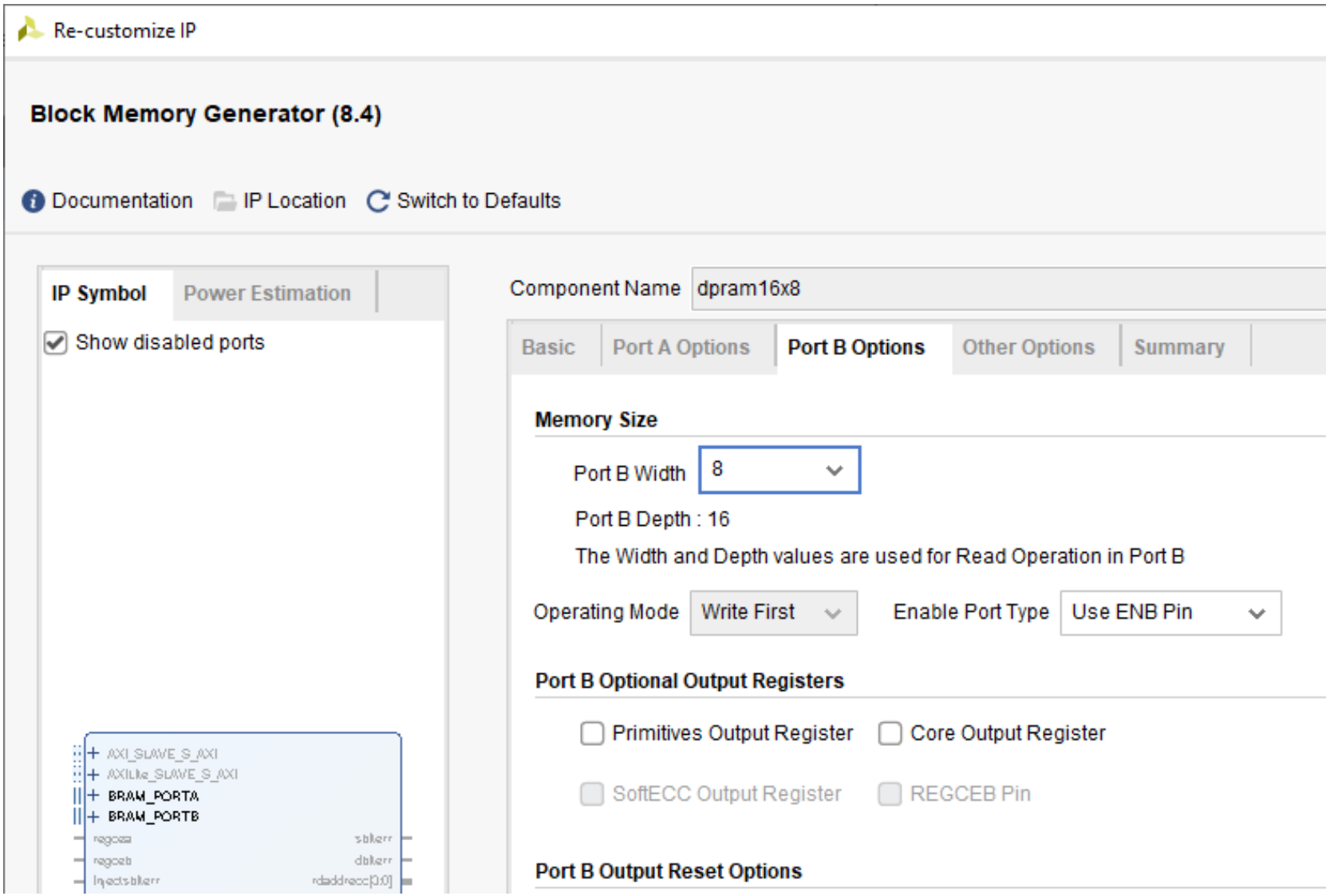
Dual Port RAM Generation (Basic)



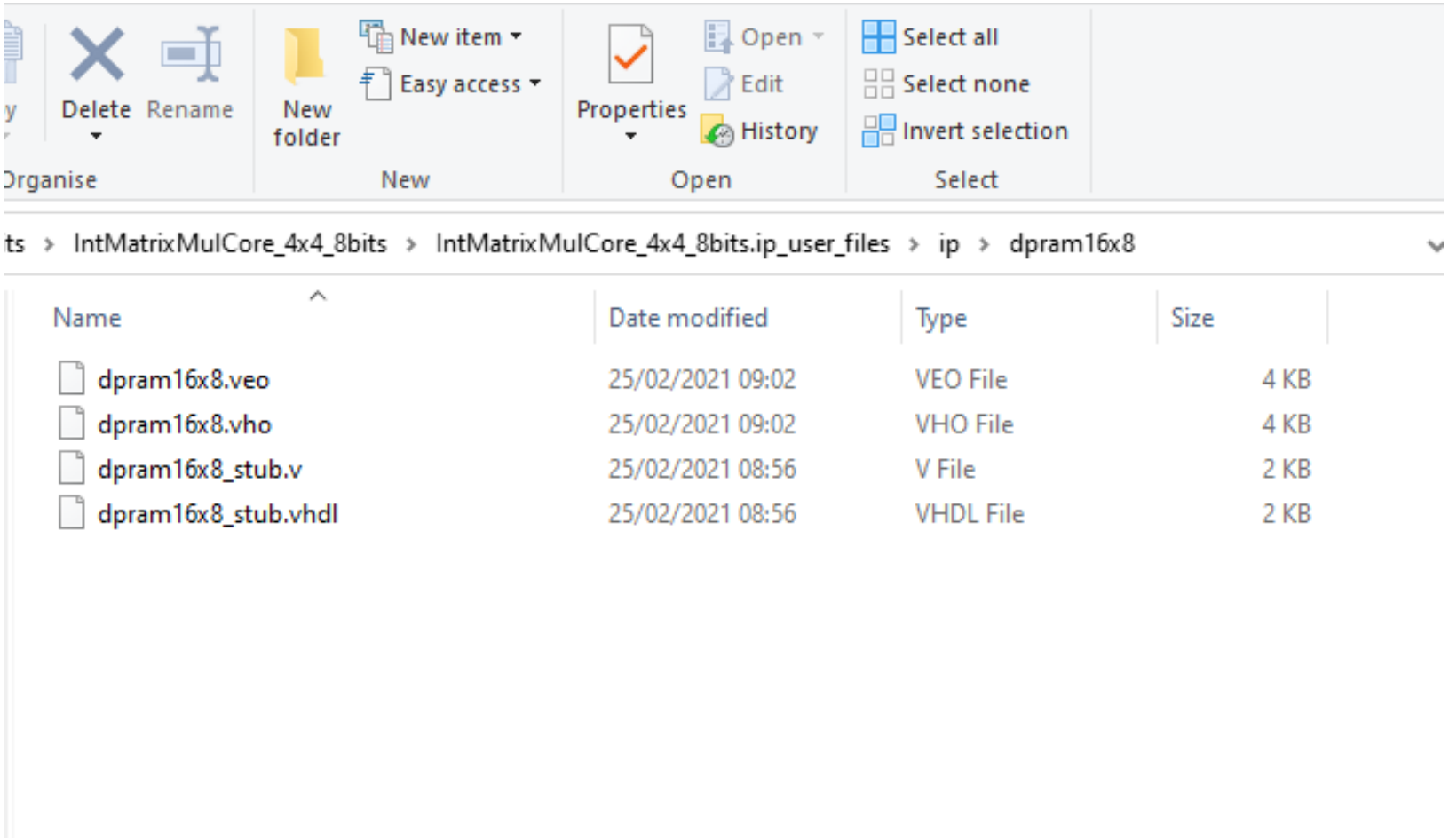
Dual Port RAM Generation (Basic)



Dual Port RAM Generation (Basic)



Dual Port RAM Generation (.VHO)



Dual Port RAM Generation (.VHO)

```
52  -- The following code must appear in the VHDL architecture header.
53
54  ------------ Begin Cut here for COMPONENT Declaration ------ COMP_TAG
55  COMPONENT dpram16x8
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      clkb : IN STD_LOGIC;
62      enb : IN STD_LOGIC;
63      addrb : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
64      doutb : 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 : dpram16x8
74    PORT MAP (
75      clka => clka,
76      wea => wea,
77      addra => addra,
78      dina => dina,
79      clkb => clkb,
80      enb => enb,
81      addrb => addrb,
82      doutb => doutb
83    );
84  -- INST_TAG_END ------ End INSTANTIATION Template ---------
```

Matlab Code to Generate Test Data

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

File Edit Search View Encoding Language Settings Tools Macro Rur

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

File Edit Search View E

InputA.txt | InputB.txt

1	'05'
2	'01'
3	'01'
4	'06'
5	'04'
6	'07'
7	'06'
8	'07'
9	'00'
10	'03'
11	'02'
12	'04'
13	'03'
14	'05'
15	'06'
16	'00'

C:\VHDL\IntMatrixMulCore

File Edit Search View E

InputA.txt | InputB.txt

1	'02'
2	'05'
3	'01'
4	'08'
5	'09'
6	'09'
7	'09'
8	'08'
9	'01'
10	'00'
11	'00'
12	'00'
13	'08'
14	'04'
15	'07'
16	'03'

C:\VHDL\IntMatrixMulCore_4x4_8bits\Ou

File Edit Search View Encoding Lan

InputA.txt | InputB.txt | OutputC_m

1	'00044'
2	'0003A'
3	'00038'
4	'00042'
5	'00085'
6	'0006F'
7	'00074'
8	'0006D'
9	'0003D'
10	'0002B'
11	'00037'
12	'00024'
13	'00039'
14	'0003C'
15	'00030'
16	'00040'

Result Data

C:\VHDL\IntMatrixMulCore_4x4_8bits\IntMatrixMulCore_4x4_8bits\IntMatrixMulCore_4x4_8bits.sim\sim_1\behav\xsim\OutputC.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

InputA.txt | InputB.txt | OutputC_matlab.txt | OutputC.txt

```
Results
Data from Matlab     Data from Simulation
00044                000044                TRUE
0003A                00003A                TRUE
00038                000038                TRUE
00042                000042                TRUE
00085                000085                TRUE
0006F                00006F                TRUE
00074                000074                TRUE
0006D                00006D                TRUE
0003D                00003D                TRUE
0002B                00002B                TRUE
00037                000037                TRUE
00024                000024                TRUE
00039                000039                TRUE
0003C                00003C                TRUE
00030                000030                TRUE
00040                000040                TRUE
```