

# IP-CORE INITIALIZATION

## • Instruction Memory

The reader is requested to adhere to the following specifications while generating the Memory Unit for instructions. Please note that this memory module should be generated before synthesizing the top *KGP\_RISC* module. The name of the module must be **SINGLE\_PORT\_ROM**.

- **Memory Type** must be *Single Port ROM*. **Enable 32-bit Address** must be *checked*.

Documents View  
IP Symbol

logiCORE **Block Memory Generator** xilinx.com:ip:blk\_mem\_gen:7.3

Memory Type: Single Port ROM

Clocking Opt: Single Port RAM  
Simple Dual Port RAM  
True Dual Port RAM  
Single Port ROM  
Dual Port ROM

Addressing Options  
☒ Enable 32-bit Address

ECC Options  
ECC Type: No ECC  
☐ Use Error Injection Pins: Single Bit Error Injection

Write Enable  
☐ Use Byte Write Enable  
Byte Size: 8 bits

Algorithm  
Defines the algorithm used to concatenate the block RAM primitives. See the datasheet for more information.  
☒ Minimum Area  
☐ Low Power  
☐ Fixed Primitives  
Primitive (Write Port A): 8kx2  
Actual Primitive(s) Used: 16kx2, 8kx2

IP Symbol Power Estimation

Datasheet < Back Page 2 of 6 Next > Generate Cancel Help

- Make sure that the **Read Width** is 32. Fill **Read Depth** as 65536.

Documents View  
IP Symbol

logiCORE **Block Memory Generator** xilinx.com:ip:blk\_mem\_gen:7.3

Port A Options

Memory Size  
Read Width: 32 Range: 32..1024  
Read Depth: 65536 Range: 2..9011200

Operating Mode  
☒ Write First  
☐ Read First  
☐ No Change

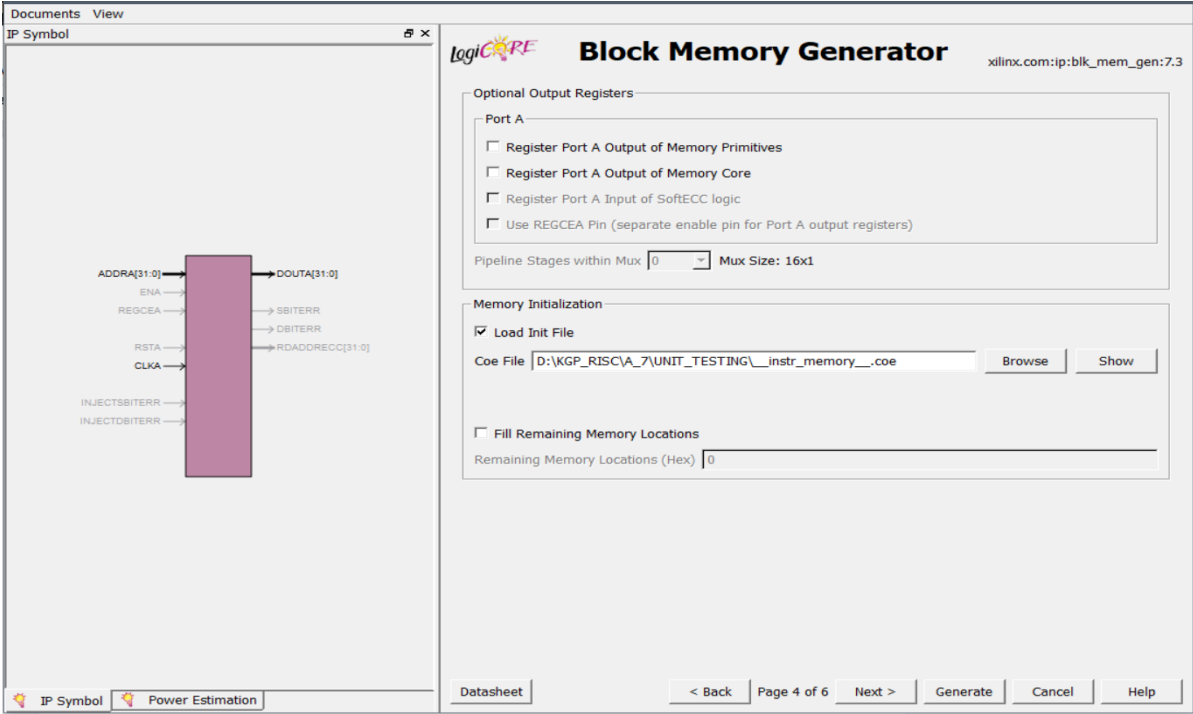
Enable  
☒ Always Enabled  
☐ Use ENA Pin

IP Symbol Power Estimation

Datasheet < Back Page 3 of 6 Next > Generate Cancel Help

- **Load Init File** should be *checked* and any of the initialized instruction memories can be loaded in the form of a .coe file, depending upon whether you are performing a unit test or the integrated test. Which .coe files have to be loaded for which tests will be clearly specified later. The reader can also load their own .coe files.

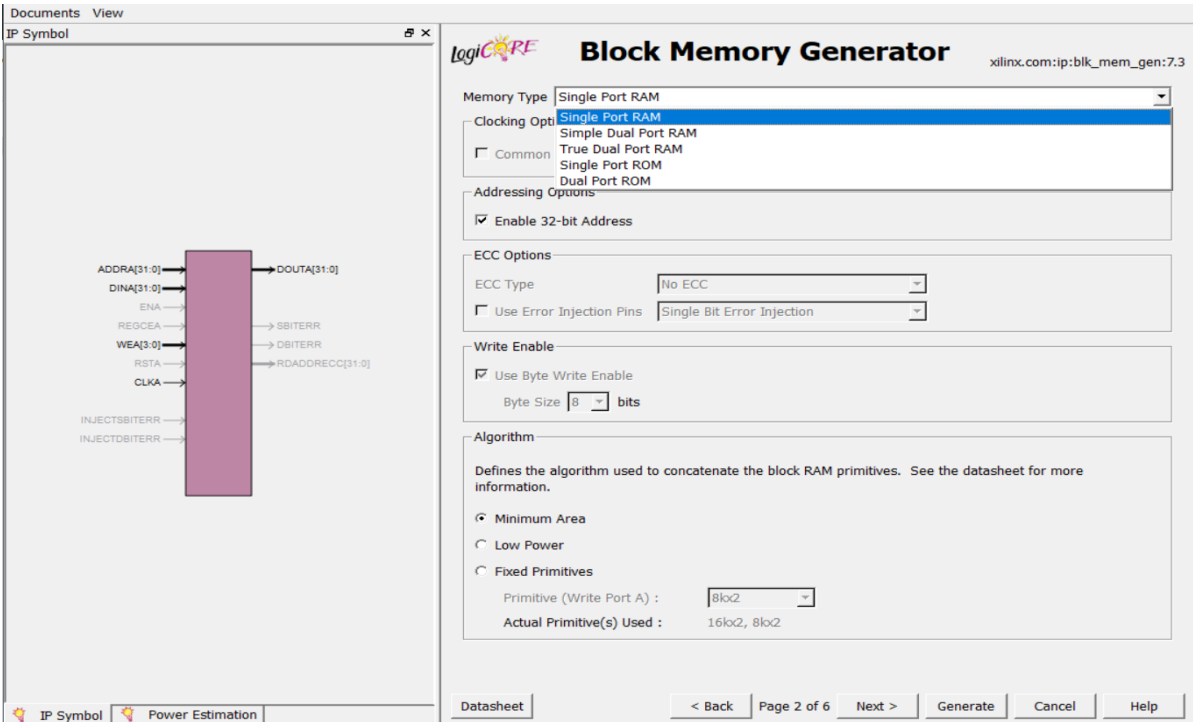
Even if you do not load the .coe file at the time of generation, you can always go back to the block memory generator in order to load the file by simply double-clicking on the ip-core in the hierarchy.



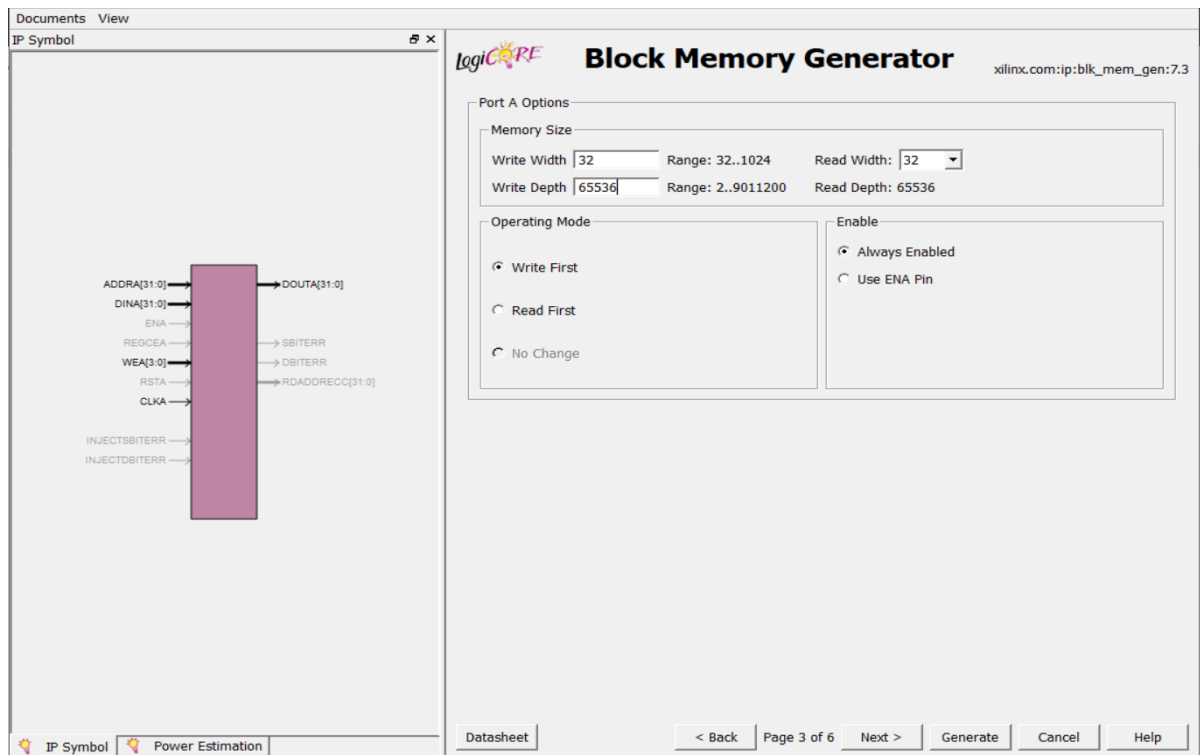
## • Data Memory

The reader is requested to adhere to the following specifications while generating the Memory Unit for data or block-RAM. Please note that this memory module should be generated before synthesizing the top *KGP\_RISC* module. The name of the module must be **SINGLE\_PORT\_RAM**.

- **Memory Type** must be *Single Port RAM*. **Enable 32-bit Address** must be *checked*.

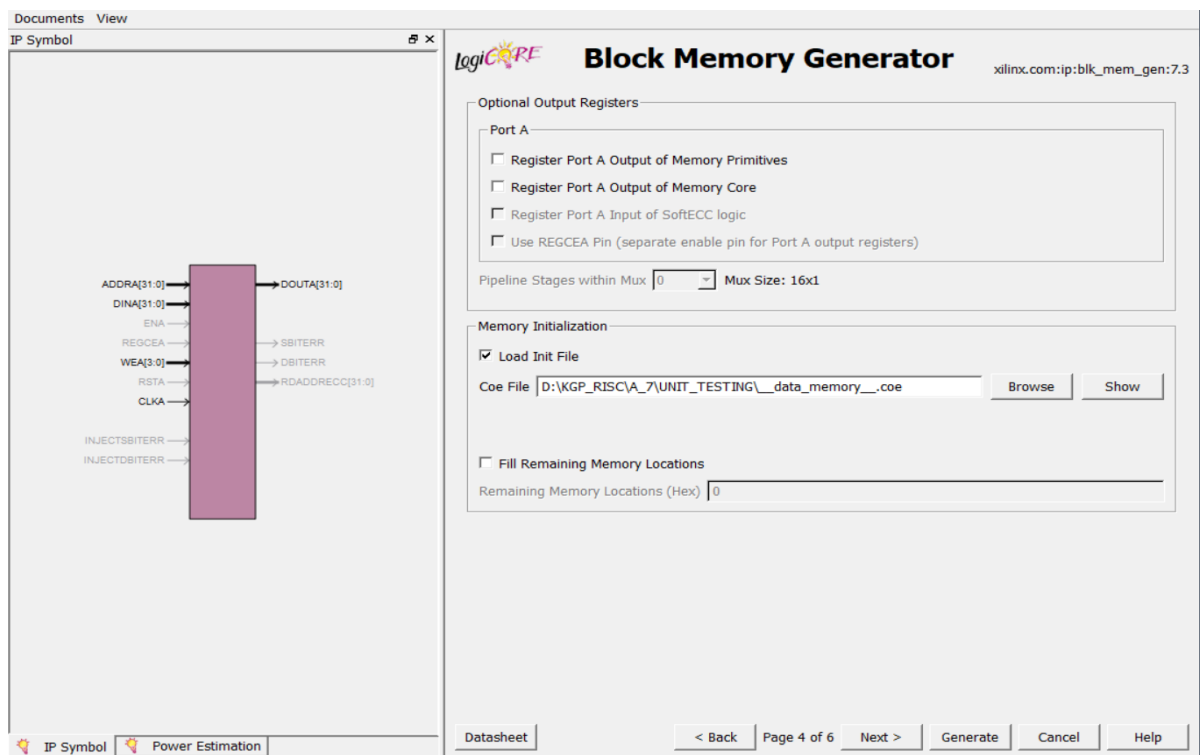


- Make sure that the **Write Width** and **Read Width** is 32. Fill **Write Depth** as 65536.



- **Load Init File** should be *checked* and any of the initialized data memories can be loaded in the form of a .coe file, depending upon which test you are performing. Which .coe files have to be loaded for which tests will be clearly specified later. The reader can also load their own .coe files.

Even if you do not load the .coe file at the time of generation, you can always go back to the block memory generator in order to load the file by simply double-clicking on the ip-core in the hierarchy.



## • Initializing Single Port ROM

The reader will have to load the initial instruction memory into the *Single Port ROM* before simulating the *KGP-RISC* Processor. There are multiple testbenches written for unit testing and for integrated testing that require a suitable .coe file to be first loaded into the ROM before simulation.

### Unit Testing

The testbenches that require a .coe file loaded into the ROM before simulating are as follows. Next to every testbench, the corresponding .coe file that has to be loaded is also mentioned.

**TestBench\_Datapath.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/UNIT\_TESTING/TestBench\_Datapath.v)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/UNIT\\_TESTING/\\_\\_instr\\_memory\\_\\_.coe](#)

**TestBench\_InstructionMemory.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/UNIT\_TESTING/TestBench\_InstructionMemory.v)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/UNIT\\_TESTING/\\_\\_instr\\_memory\\_\\_.coe](#)

### Integrated Testing

The testbenches that require a .coe file loaded into the ROM before simulating are as follows. Next to every testbench, the corresponding .coe file that has to be loaded is also mentioned.

**TestBench\_BinaryToDecimal.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/INTEGRATED\_TESTING/BINTODEC/TestBench\_BinaryToDecimal.v)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/BINTODEC/\\_\\_instr\\_memory\\_\\_.coe](#)

**TestBench\_ComputeGCD.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/INTEGRATED\_TESTING/GCD/TestBench\_ComputeGCD.v)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/GCD/\\_\\_instr\\_memory\\_\\_.coe](#)

## • Initializing Single Port RAM

The reader will have to load the initial instruction memory into the *Single Port RAM* before simulating the *KGP-RISC* Processor. There are multiple testbenches written for unit testing and for integrated testing that require a suitable .coe file to be first loaded into the RAM before simulation.

### Unit Testing

The testbenches that require a .coe file loaded into the RAM before simulating are as follows. Next to every testbench, the corresponding .coe file that has to be loaded is also mentioned.

**TestBench\_Datapath.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/UNIT\_TESTING/TestBench\_Datapath.coe)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/UNIT\\_TESTING/\\_\\_data\\_memory\\_\\_.coe](#)

**TestBench\_DataMemory.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/UNIT\_TESTING/TestBench\_DataMemory.coe)

Location: [A7\\_19CS10044\\_19CS30053/KGP\\_RISC/UNIT\\_TESTING/\\_\\_data\\_memory\\_\\_.coe](#)

### Integrated Testing

The testbenches that require a .coe file loaded into the ROM before simulating are as follows. Next to every testbench, the corresponding .coe file that has to be loaded is also mentioned.

Note that the input arguments for the algorithm implemented in the assembly code are taken from the data memory. So different test cases will correspond to different initializations of the data memory. Hence depending on the test case ID, the suitable .coe file should be loaded into the RAM.

**TestBench\_BinaryToDecimal.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/INTEGRATED\_TESTING/BINTODEC/TestBench\_BinaryToDecimal.v)

Location:

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/BINTODEC/\\_\\_test\\_case\\_01\\_\\_.coe](#)

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/BINTODEC/\\_\\_test\\_case\\_02\\_\\_.coe](#)

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/BINTODEC/\\_\\_test\\_case\\_03\\_\\_.coe](#)

**TestBench\_ComputeGCD.v** (A7\_19CS10044\_19CS30053/KGP\_RISC/INTEGRATED\_TESTING/GCD/TestBench\_ComputeGCD.v)

Location:

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/GCD/\\_\\_test\\_case\\_01\\_\\_.coe](#)

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/GCD/\\_\\_test\\_case\\_02\\_\\_.coe](#)

[A7\\_19CS10044\\_19CS30053/KGP\\_RISC/INTEGRATED\\_TESTING/GCD/\\_\\_test\\_case\\_03\\_\\_.coe](#)