Test Engine FPGA IP User Guide

Agilex™ 5 and Agilex™ 7 FPGAs

Updated for Quartus® Prime Design Suite: **24.2**

IP Version: **1.1.0**

Online Version
Send Feedback

817758
2024.07.08

Contents

1. About the Test Engine FPGA IP

1.1. Release Information

IP versions are the same as the Quartus® Prime Design Suite software versions up to v19.1. From Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

The IP version (X.Y.Z) number may change from one Quartus Prime software version to another. A change in:

- X indicates a major revision of the IP. If you update your Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 1.

Item	Description
IP Version	1.1.0
Quartus Prime	24.2
Release Date	2024.07.08

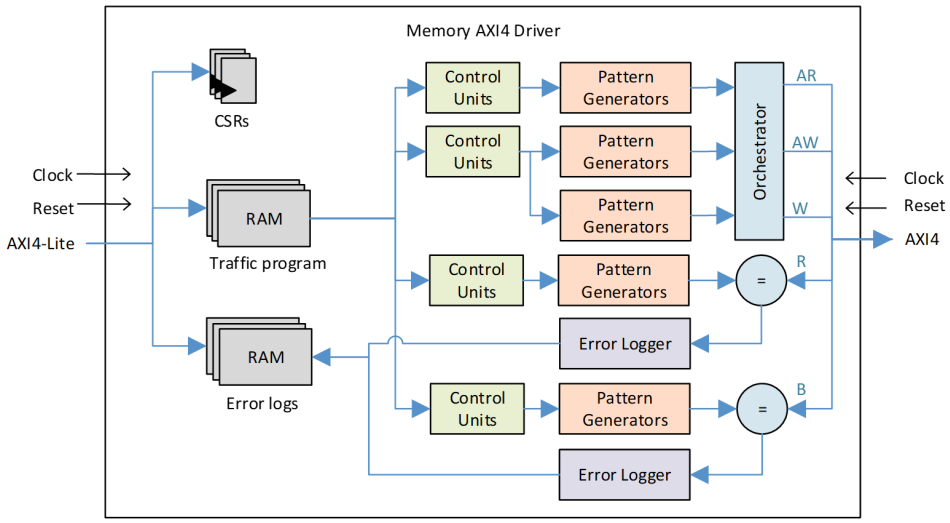
Note: This documentation is preliminary and subject to change.

2. Test Engine IP - Introduction

The Test Engine IP is a software-programmable AXI traffic generator that generates a configurable pattern of reads and writes to a programmable memory address range.

You can use the Test Engine IP to access multiple AXI interfaces and exercise the generated configurable patterns. This AXI traffic generator also monitors the data read from the memory to ensure that it matches the expected data; if the read data doesn't match the expected data, the Test Engine IP asserts a failure. All failures are captured within the error logs.

Figure 1. Test Engine IP's Memory AXI4 Driver Architecture



2.1. Test Engine IP Feature Support

The Test Engine IP is intended for use with Agilex™ 5 and Agilex 7 memory IPs.

In the Test Engine IP, you can easily describe traffic patterns using software, and have the flexibility to interact with other interfaces by creating a driver for each individual interface. This AXI traffic generator enables functionality and performance testing in both simulation and hardware – it is not necessary to recompile the design to try different traffic patterns during your testing. For hardware testing, System Console interaction is available with the Test Engine IP over a JTAG connection.

The Test Engine IP supports the following drivers: Memory AXI4, CSR AXI4-Lite, Memory Status, Memory Reset, and CAM AXI4-Stream. Refer to the following table for details on supported driver features.

Table 2. Driver Feature Support

Driver Type	Feature Support				
	Available in a Design Example?	Standalone Test Engine IP Supported?	Sideband Accessible?	Software Compiler Supported?	Hardware Reprogrammable?
Memory AXI4	X [1]	X	X	X	X
CSR AXI4-Lite	X [2]	X		X	
Memory Status	X [3]	X			
Memory Reset	X [3]	X			
CAM AXI4-Stream	X [4]				

Support key level:

- X = supported feature.
- Standalone = driver can be added to new instantiation of Test Engine IP variant support.
- Sideband Accessible = exported control and status registers (CSRs) support.
- Software Compiler = driver's traffic program redefinable support; a fixed program is used if no support is available.
- Hardware Reprogrammable = reprogram different traffic program support; traffic program loads at design compilation if software compiler is supported but no hardware reprogrammable support.

Note:
1. Available only in Agilex 5 and Agilex 7 M-Series EMIF, HBM2E and Memory Subsystem IP design examples.
2. Available only in HBM2E IP with AXI-Lite design examples.
3. Available only in Memory Subsystem IP design examples.
4. Available only in Memory Subsystem IP design examples. Cannot be instantiated as part of a standalone Test Engine IP variant.

The Memory AXI4 driver exercises the AXI4 interface of a memory IP by generating traffic and error-checking responses. This driver supports out-of-order responses over any range of IDs, can perform ALU operations on selective bits within the address bus by using address fields, and allows you to enable or disable certain features to preserve the maximum achievable clock frequency.

When error-checking responses while using the Memory AXI4 driver, the error logger captures detailed logs for every error triggered. An error is triggered if the driver receives a different BID, receives a BRESP not equivalent to the expected BRESP, or if RLAST is not asserted at the correct data beat. Apart from the error logger, there are dedicated counters for each error condition. These error counters can capture higher quantities of errors than the error logger. You can also enable the stop-on-error feature to stop traffic generation when an error condition occurs.

Related Information

- CSR AXI4-Lite Driver Registers on page 48
- Memory Status Driver Registers on page 49
- Memory Reset Driver Registers on page 49
- CAM AXI4-Stream Driver Registers on page 49

3. Parameterizing the Test Engine IP

This chapter contains parameter descriptions for the Test Engine IP.

You can parameterize your Test Engine IP using the Test Engine IP parameter editor. The parameter editor consists of two tabs, **Drivers** and **Remote Access**. You can create your desired configuration using the parameters on these tabs.

3.1. Parameters

You can configure the IP to run all drivers after a reset.

Figure 2. Group: Parameters

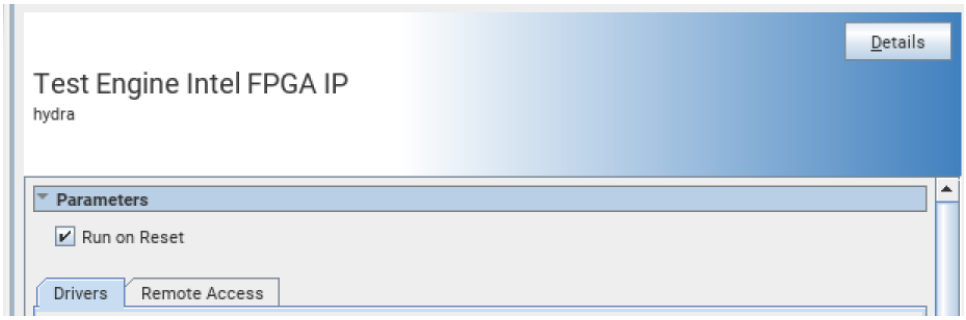


Table 3. Group: Parameters

Display Name	Description
Run on Reset	Check this parameter to automatically run all drivers after a reset is applied and released. It is recommended to enable this parameter for simulations. To run traffic when this parameter is disabled, access to the control and status registers (CSR) is required. Refer to *Global Control Registers* for the `ctrl_status_0` register's drivers_run field and the driver_run_bit mask registers. You should use the System Console library for hardware if this parameter is disabled. Refer to *Test Engine IP - Software* for information on the System Console library and testengine_run function.

Related Information

Global Control Registers on page 30

3.2. Drivers Parameters

The parameters of the **Drivers** tab allow you to select the number of drivers that you want to implement, the address and data configurations, and the AXI ports settings.

Figure 3. Group: Drivers

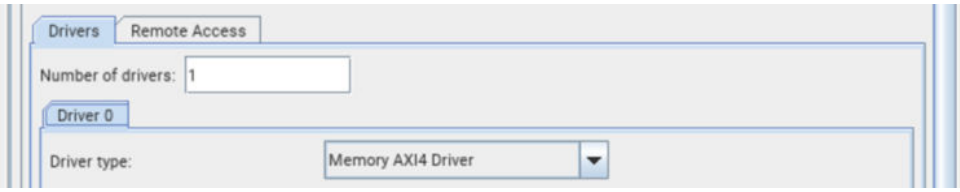


Table 4. Group: Drivers

Display Name	Description
Number of drivers	Specifies the number of drivers. The supported range for this parameter is from 1 to 50.
Driver type	Specifies the driver type. You can choose from several driver options: • CSR AXI4-Lite driver • Memory AIX4 driver • Memory status driver • Memory reset driver • CAM AXI4-stream driver

3.2.1. CSR AXI4-Lite Driver Parameters

Figure 4. Group:Drivers / AXI4-Lite Ports

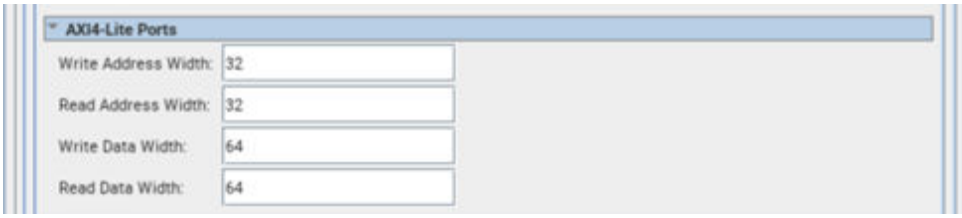


Table 5.

Display Name	Description
Write Address Width	Specifies width of AWADDR port of the AXI4-Lite bus. The supported range for this parameter is from 1 to 64. Default value is 32.
Read Address Width	Specifies width of ARADDR port of the AXI4-Lite bus. The supported range for this parameter is from 1 to 64. Default value is 32.
Write Data Width	Specifies width of WDATA port of the AXI4-Lite bus. The supported values for this parameter are 32 and 64. Default value is 32.
Read Data Width	Specifies width of RDATA port of the AXI4-Lite bus. The supported values for this parameter are 32 and 64. Default value is 32.

3.2.2. Memory AXI4 Driver Parameters

Figure 5. Group: Drivers / Address Generation

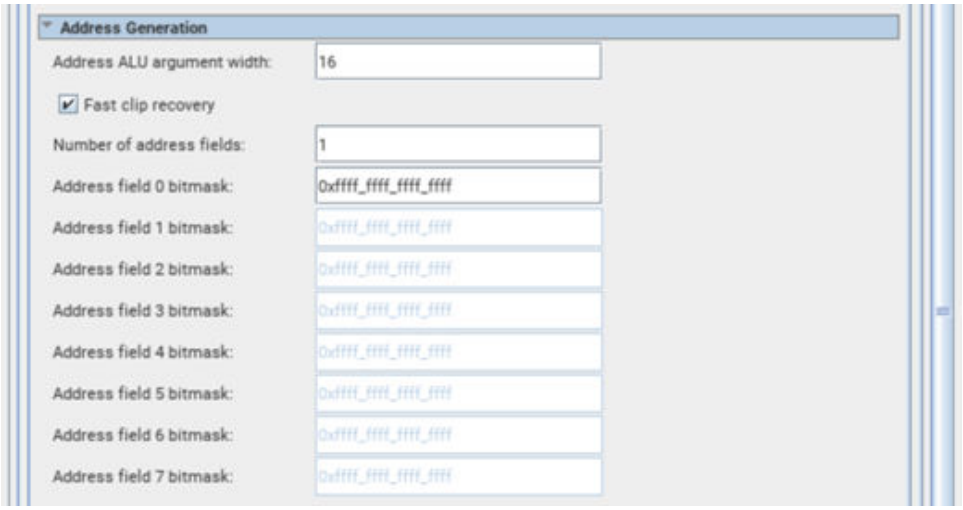


Table 6. Group: Drivers / Address Generation

Display Name	Description
Address ALU argument width	Specifies the width of the Address ALU. This affects the number of bits the Address ALU can change. Increasing this will allow wider ranges of address sweeping per command, but could degrade Fmax. The supported range for this parameter is from 1 to 44. Default value is 16.
Fast clip recovery	If enabled, address will reset to lower bound for 1 cycle when it exceeds upper bound, else it will reset to lower bound for 2 cycles – where lower bound is the minimum value of the defined address range, and upper bound is the maximum value of the defined address range. Enabling this can reduce latency for looped addresses, but can also degrade Fmax.

continued...

Display Name	Description
Number of address fields	Address fields allow updating selective bits (for example, field) within the address. The supported range for this parameter is from 1 to 8. Default value is 1.
Address field 0 bitmask	Update selective bits (for example, field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 1 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 2 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 3 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 4 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 5 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 6 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.
Address field 7 bitmask	Update selective bits (i.e. field) within the address by specifying a bitmask for the field. The main usage of address fields is to selectively choose which bits to affect. The supported range for this parameter is 0x0000_0000_0000_0000 to 0xffff_ffff_ffff_ffff. Default value is 0xffff_ffff_ffff_ffff.

Figure 6. Group: Drivers / Data Generation

Table 7. Group: Drivers / Data Generation

Display Name	Description
Number of DQ generators	Memory DQ pins can be driven by distinct pattern generators. If there are more DQ pins than generators, the patterns will be replicated. If there are fewer DQ pins than generators, the unused generators are discarded. The supported values for this parameter are: 1, 2, 4, 8, 16. Default value is 2.
Number of DM generators	Memory DM pins can be driven by distinct pattern generators. If there are more DM pins than generators, the patterns will be replicated. If there are fewer DM pins than generators, the unused generators are discarded. The supported values for this parameter are: 1, 2, 4, 8. Default value is 2.
AXI-data / Memory-DQ width ratio	Ratio between AXI data width and Memory DQ width, which affects the unroll factor of DQ and DM generators. (Memory-DQ Width Ratio = Data Width ÷ DQ Width).

Figure 7. Group: Drivers / AXI4 Ports

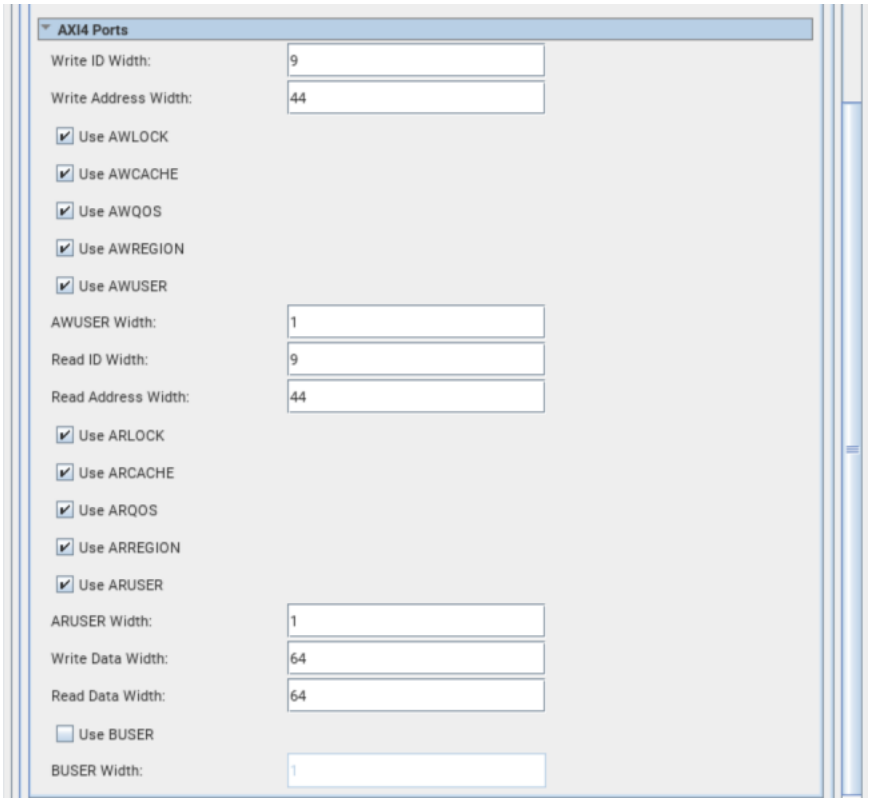


Table 8. Group: Drivers / AXI4 Ports

Display Name	Description
Write ID Width	Specifies widths of AWID and BID ports of the AXI4 bus. The supported range for this parameter is from 1 to 9. Default value is 7.
Write Address Width	Specifies width of AWADDR port of the AXI4 bus. The supported range for this parameter is from 4 to 64. Default value is 31.
Use AWLOCK	Use AWLOCK port on the AXI4 bus.
Use AWCACHE	Use AWCACHE port on the AXI4 bus.
Use AWQOS	Use AWQOS port on the AXI4 bus.
Use AWREGION	Use AWREGION port on the AXI4 bus.
Use AWUSER	Use AWUSER port on the AXI4 bus.
AWUSER Width	Specifies width of AWUSER port of the AXI4 bus. The supported range for this parameter is from 1 to 64. Default value is 8.
Read ID Width	Specifies widths of ARID and RID ports of the AXI4 bus. The supported range for this parameter is from 1 to 9. Default value is 7.

continued...

Display Name	Description
Read Address Width	Specifies width of ARADDR port of the AXI4 bus. The supported range for this parameter is from 4 to 64. Default value is 31.
Use ARLOCK	Use ARLOCK port on the AXI4 bus.
Use ARCACHE	Use ARCACHE port on the AXI4 bus.
Use ARQOS	Use ARQOS port on the AXI4 bus.
Use ARREGION	Use ARREGION port on the AXI4 bus.
Use ARUSER	Use ARUSER port on the AXI4 bus.
ARUSER Width	Specifies width of ARUSER port of the AXI4 bus. The supported range for this parameter is from 1 to 64. Default value is 8.
Write Data Width	Specifies width of write data. 2^N bits are mapped to the WDATA port and the remaining bits, up to a maximum of 64, are mapped to WUSER port of the AXI4 Bus. N is an integer ranging from 3 to 10. The supported ranges for this parameter are from 8 to 192, 256 to 320, 512 to 576, and 1024 to 1088.
Read Data Width	Specifies width of read data. 2^N bits are mapped to the RDATA port and the remaining bits, up to a maximum of 64, are mapped to RUSER port of the AXI4 Bus. N is an integer ranging from 3 to 10. The supported ranges for this parameter are from 8 to 192, 256 to 320, 512 to 576, and 1024 to 1088.
Use BUSER	Use BUSER port on the AXI4 bus.
BUSER Width	Specifies width of BUSER port of the AXI4 bus. The supported range for this parameter is from 1 to 64. Default value is 8.

Note: For more information on AXI signals, refer to the *External Memory Interfaces Agilex 7 M-Series FPGA IP User Guide* or *External Memory Interfaces (EMIF) IP User Guide: Agilex 5 FPGAs and SoCs*, as appropriate, or to the *High Bandwidth Memory (HBM2E) Interface Agilex 7 M-Series FPGA IP User Guide*.

Related Information

- External Memory Interfaces Agilex 7 M-Series FPGA IP User Guide
- External Memory Interfaces (EMIF) IP User Guide: Agilex 5 FPGAs and SoCs
- High Bandwidth Memory (HBM2E) Interface Agilex 7 M-Series FPGA IP User Guide

3.2.3. Memory Status Driver Parameters

Figure 8. Group: Drivers / General

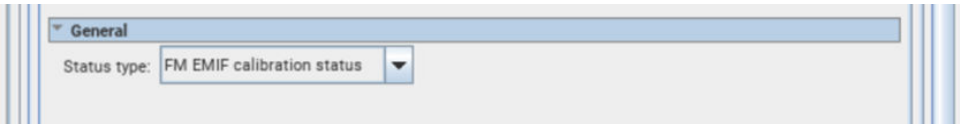


Table 9. Group: Drivers / General

Display Name	Description
Status type	Specifies the status interface type of the memory IP. The supported value for this parameter is FM EMIF calibration status.

3.2.4. Memory Reset Driver Parameters

Figure 9. Group: Drivers / General

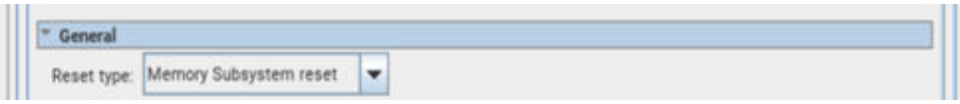


Table 10. Group: Drivers / General

Display Name	Description
Reset type	Specifies the reset interface type of the memory IP. The supported value for this parameter is Memory Subsystem reset.

3.2.5. CAM AXI4-Stream Driver Parameters

CAM AXI4-Stream driver parameters are available only in Memory Subsystem IP design example, and cannot be instantiated as part of a standalone Test Engine IP variant. Refer to the *Agilex 7 F-Series and I-Series FPGA Memory Subsystem IP User Guide* or *Agilex 7 F-Series and I-Series FPGA Memory Subsystem IP Release Notes* for more information on CAM AXI4-Stream driver parameters.

Related Information

- Agilex™ 7 F-Series and I-Series FPGA Memory Subsystem IP User Guide
- Agilex™ 7 F-Series and I-Series FPGA Memory Subsystem IP Release Notes

3.3. Remote Access Parameters

Figure 10. Group: Remote Access

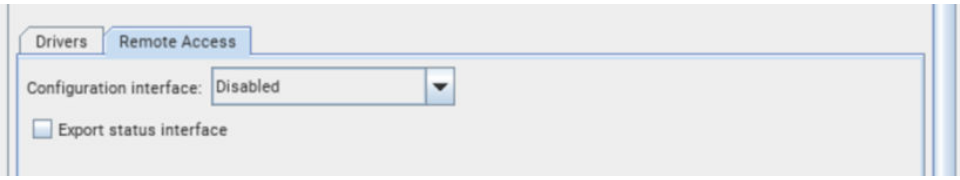


Table 11. Group: Remote Access

Display Name	Description
Configuration interface	The configuration interface enables access to Control and Status Registers (CSRs). Possible values are: **Disabled** – No CSR bus is exported. **Export** – Sideband AXI-Lite CSR buses for the Global CSR and individual drivers are exported. **Remote Access via JTAG** – A JTAG remote interface is generated internally. You can interact with the Test Engine through the System Console.
Export status interface	Status interface indicates the pass/fail status of all drivers. Exported signals are: **status_done** is asserted if all drivers are done. **status_error** is asserted if any driver reports an error.