

ECP5 Product Families Update - Errata to Soft Error Detection (SED) Function –

Product Affected:

All ECP5™ product families (ECP5, ECP5-5G, ECP5 Automotive) are affected by this Product Bulletin.

Customers Affected:

Only customers with designs that use both Soft Error Detect (SED) function **and** Distributed RAM in the same design are affected.

Customers not using the SED function or using the SED function without Distributed RAM are not affected by this product bulletin.

Customers implementing new designs with SED function should follow the guidelines in this document to avoid using Distributed RAM in their design.

Description of Discrepancy from Data Sheet and TN1184:

Soft Error Detect (SED) is a hardened (but optional) function implemented in ECP5™ product families. When enabled, SED runs in the background while user logic is actively running in the FPGA, to detect Single Event Upsets (SEUs) in the configuration memory data stored in the FPGA. This hardened function reads the configuration memory data in the background, and compares it to a calculated Cyclic Redundancy Check (CRC) stored in the device to determine if there is any configuration memory state-change due to SEU. Upon detection of such error, the hardened SED sends a signal externally to a Controller so appropriate steps can be taken to mitigate the error.

TN1184 ([LatticeECP3, ECP5 and ECP5-5G Soft Error Detection \(SED\)/Correction \(SEC\) Usage Guide](#)) specifies that “EBR and distributed memory contents are ignored in CRC generation”.

It has been found that when distributed memory is used in the user design, the contents are not excluded in the CRC generation, causing the SED function to erroneously report an SEDERROR.

Planned ECP5 Fix:

Lattice intends to resolve this discrepancy in the next revision of the product in Q3 2017. Lattice will announce the next revision through the PCN process. For additional questions regarding existing product on the Affected Device list, please contact Lattice Technical Support team through your local Field Applications Engineers.

Effects to Current Customer Design:

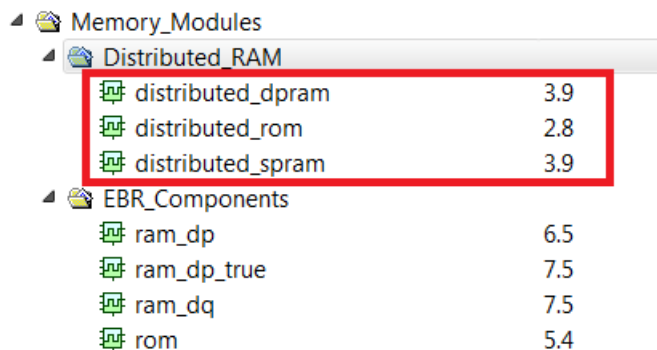
On the affected devices, customers using the SED function will need to ensure that Distributed RAM is not used in the design. If the current design is using Distributed RAM, it needs to be replaced with EBR memory.

There are 4 ways user instantiates a memory module:

1. *In Clarity Design*
2. *With PMI Module Generator*
3. *In Reveal Inserter*
4. *Inferred Memories*

1. Clarity Design:

When opening up the Catalog tab in Clarity Designer, under “Memory Modules” there are 2 types of memory that can be chosen, as shown below:

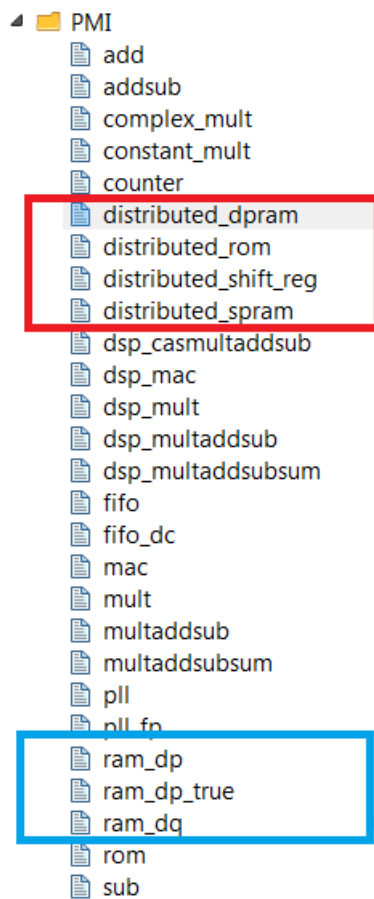


Memory_Modules	
Distributed_RAM	
distributed_dpram	3.9
distributed_rom	2.8
distributed_spram	3.9
EBR_Components	
ram_dp	6.5
ram_dp_true	7.5
ram_dq	7.5
rom	5.4

For a design that uses SED function, the “distributed_xxx” (shown in red box) should **not** be used; EBR memory should be used instead. If the design is already using Distributed RAM, a new memory should be created with EBR to replace the existing one that is using Distributed RAM.

2. PMI Module Generator:

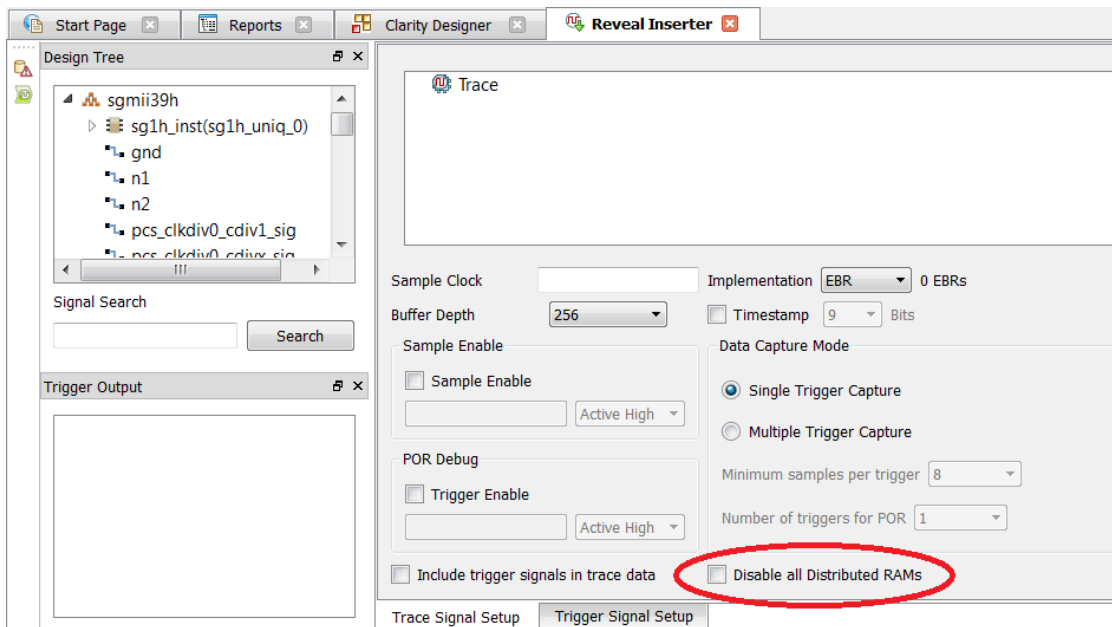
Similarly, if the memory is created in the PMI Module Generator, there are “distributed_xxx” memory selections in the list, as shown below:



For a design that uses SED function, the “distributed_xxx” (shown in red box) should **not** be used; EBR (shown in blue box) should be used instead. If the design is already using Distributed RAM, a new memory should be created with EBR to replace the existing one that is using Distributed RAM.

3. Reveal Inserter:

Reveal “Capture” needs memory to store the data to be displayed. When Reveal Inserter is being added, the following dialog screen can be seen:



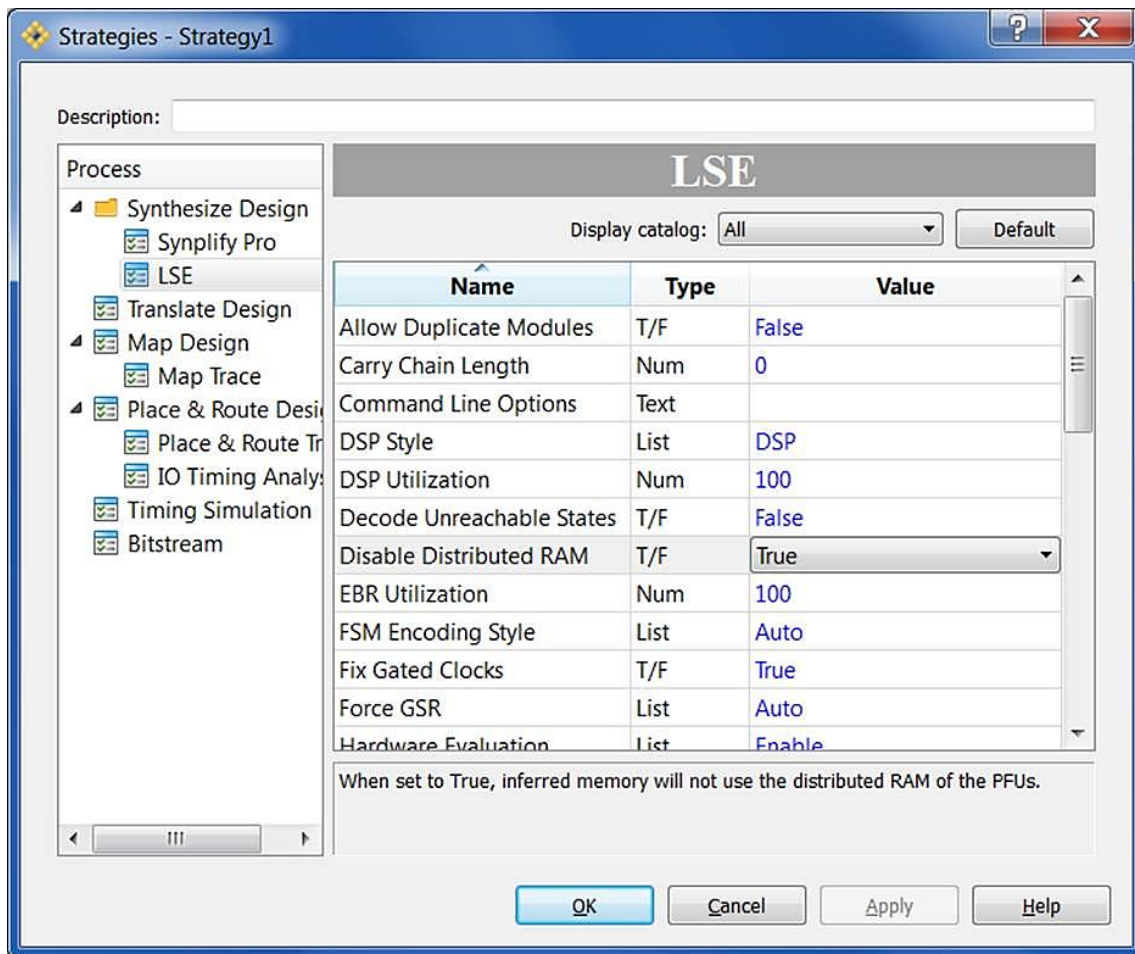
For a design that uses SED function, the “Disable all Distributed RAMs” box needs to be checked. If the existing Reveal module did not have Distributed RAM disabled, Reveal Inserter needs to be generated again to replace the existing Reveal module.

4. Inferred Memories:

Lattice Diamond Design Software allows memories to be inferred. Inferred memories could be mapped into Distributed RAM, which would cause SED to not function correctly.

To avoid the use of Distributed RAM when Diamond Software infers memories, if Lattice Synthesis Engine (LSE) is selected as the synthesis tool in the design, customer can disable the use of Distributed RAM.

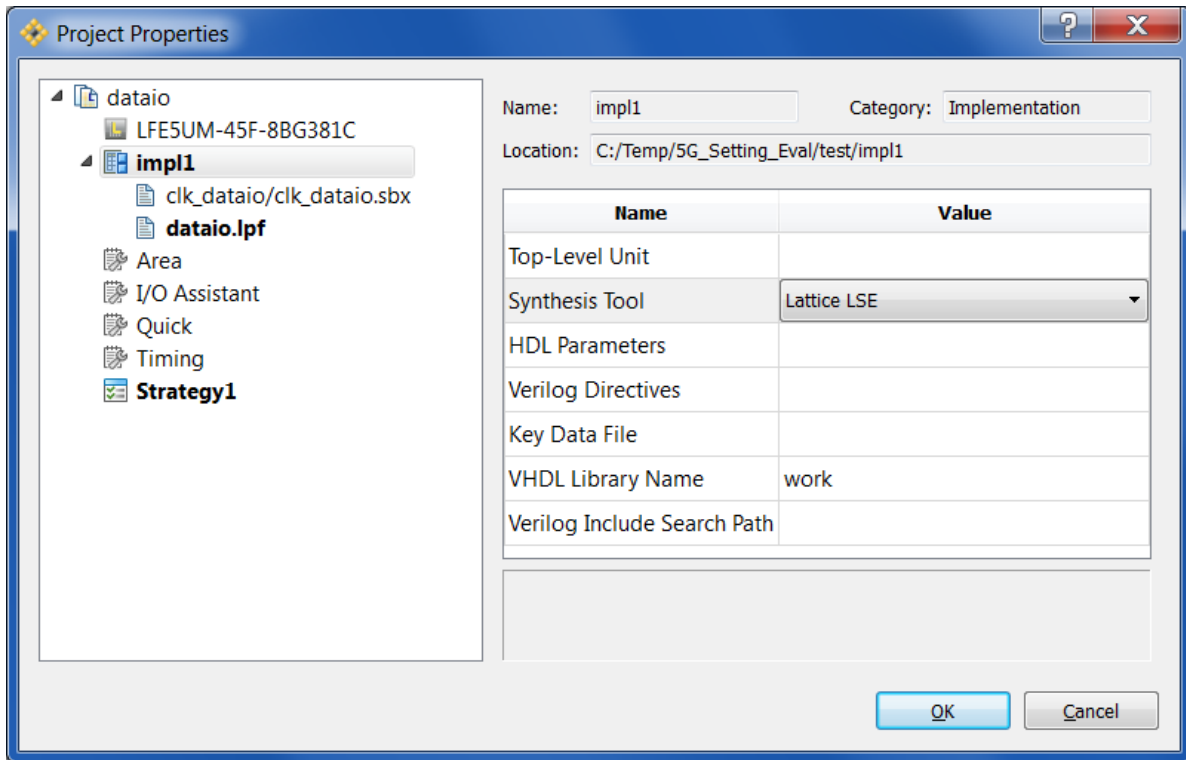
Go to Project pull-down menu, and select “Active Strategy -> LSE Settings”. The following window will pop up:



Set the “Disable Distributed RAM” setting to “True”, and re-compile the design. This will exclude the use of Distributed RAM in the design.

For designs using Synplify Pro, the customer will need to change the synthesis engine to LSE.

Go to Project pull-down menu, and select “Active Implementation -> Select Synthesis Tool...”. The following window will pop up:



Set the “Synthesis Tool” setting to “Lattice LSE”. Changing synthesis tool will require the modules generated in Clarity Designer to be re-configured. Once re-configured, customer can re-compile the design, and the Distributed RAM will be excluded.

Note: In addition to the 4 methods described above, Distributed RAM can also get instantiated if customer design is using a Lattice IP Core that utilizes Distributed RAM. Since some IP Cores are designed with Distributed RAM generated via PMI Module Generator, disabling Distributed RAM in LSE will not prevent such IP Cores from using Distributed RAM. Please contact Lattice Technical Support if you encounter this issue.

Customer Acknowledgment:

By using the devices in the Affected Devices list, the customer acknowledges that SED does not function correctly when the design uses Distributed RAM and has been provided with instructions on how to avoid using Distributed RAM in the design.

Lattice Semiconductor Home Page: <http://www.latticesemi.com>
 Application & Literature hotline: 1-800-LATTICE

Date	Revision	Description of Revision
April 14, 2017	1.0	Initial Release