ES6010 – ECC Core

Introduction

The ES6010 core implements SECDED (Single Error Correction, Double Error Detection) functionality for memories blocks. It consists of an encoder and a decoder block. The encoder block generates the required redundancy information to be stored in the memory, needed to detect/correct payload data. On data retrieval, redundancy information is retrieved along with payload data and the combined set checked for errors. Single bit errors are corrected automatically. Detected/corrected errors can be signaled to a CPU for further action if needed through error flags provided.

Features

- ✓ Code generator produces optimal design given payload data width.
- ✓ Dual Error Detection can be optionally configured during core generation. Requires one extra redundancy bit.
- ✓ High speed operation. The design is purely combinational for highest integration flexibility
- ✓ Simple external interface
- ✓ Bypass modes built in for testability of redundancy bits
- ✓ Available in ASIC and FPGA Technologies.

Applications

- Data integrity protection for high-speed memory circuits.
    - DRAM
    - High density SRAM
- Mission critical devices
    - Aerospace
    - Automotive
    - Medical
    - Financial

Functional Description

Figure 1 depicts the schematic block diagram of the ECC IP core. It exposes an external interface and an interface to the underlying memory macro used for storage. External data (application payload) is augmented in width with redundancy information prior to storage. Upon read, the augmented data is checked for consistency. If the consistency checks pass the data is returned unmodified.

If the redundancy information indicates the presence of a single bit error, it will be automatically corrected and a signal (error_corrected) will be asserted to indicate such an event. The system designer can choose to perform further actions upon this signal assertion but is outside the scope of this unit. Examples of possible actions are:

- Interrupting the CPU to log the event

- Ask the CPU to refresh the offending memory location. Notice that in this case the address of the offending access should be externally latched in sync with the assertion of error_corrected

If the consistency checks indicate that they 2 error bits were detected (2 bits errors can be detected but not corrected) the condition will be signaled by the error_detected signal. As an example, the system designer could choose to interrupt the CPU and perform a system halt or any other appropriate action as a response to this event. Notice that dual error detection is a configuration option that requires one extra bit of redundancy information.

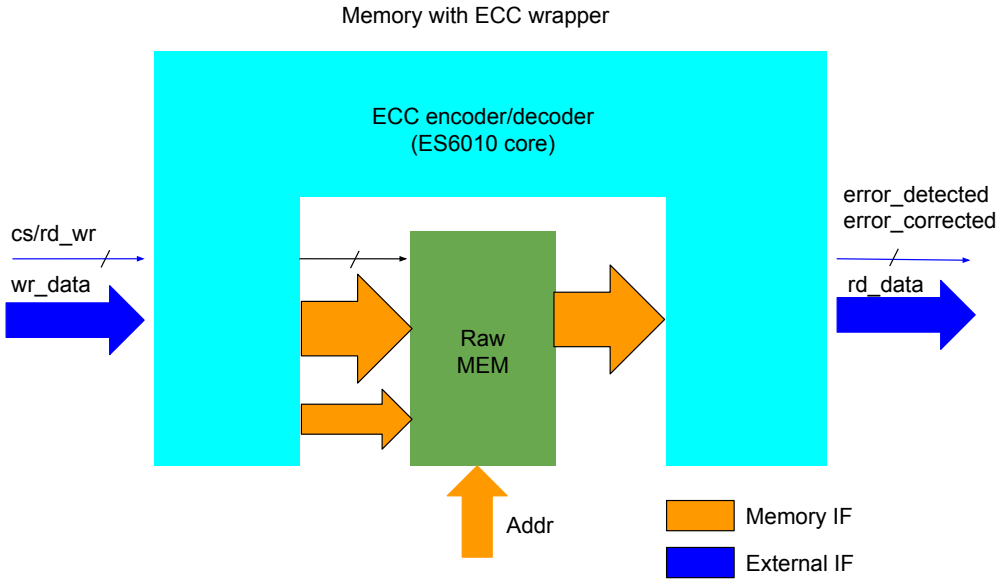

Figure 1. ES6010 Schematic Block Diagram

Note that the schematic block diagram in Figure 1 does not show test related signals (ecc_disable_*).

The core also provides a bypass path to allow direct access to the redundancy data field in memory for testing purposes.

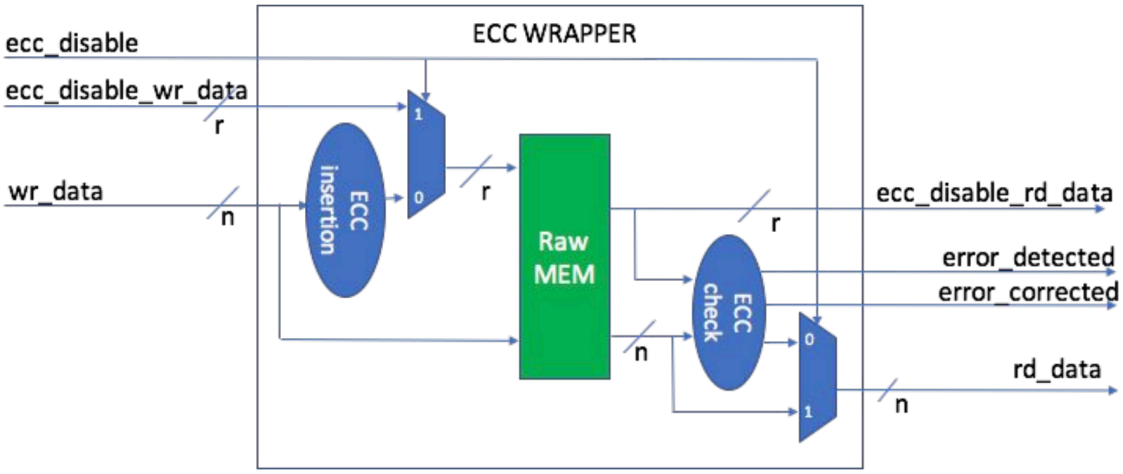

Figure 2. ES6010 bypass path

When ecc_disable is high, the user can write directly the redundancy bits on to memory and upon read, it will have access to the uncorrected data along with the raw redundancy bits previously stored. Ecc_disable is a dynamic signal that can be asserted/de-asserted on any read/write cycle.

Note that in Figure 2:

n = <*number of user/payload data bits*> = D_WDH
r = <*number of reduncancy bits*> = PRT_WDH + SECDEC

Parameters

The following parameters should be defined at *generation* time:

Name	*Valid Range*	*Description*
D_WDH	[5, 2036]	Payload/user data width
SECDEC	0, 1	1: SECDEC mode 0: SEC mode

Table 1.

After generation, the core will include other informative generated parameters that are used internally:

Name	Description
MEM_WDH	Data width including parity/redundancy. This is the total data width that the memory macro must support

Table 2.

MEM_DWH is generated as follows:

$$MEM\_DWH = D\_WDH + PRT\_WDH + SECDEC$$

Pin Description

External Interface:

Name	I/O	Width	Description
clk	Input	1	System Clock input (currently un-used)
wr_data	Input	D_WDH	Application write data (no redundancy included)
rd_wr (*)	Input	1	Choses read (low) vs. write cycle (high)
cs (*)	Input	1	Chip select to the memory
rd_data	Output	D_WDH	Application read data (corrected if applicable)
error_detected	Output	1	Flag that is held high as upon detecting a single or double bit error in the data (upon read). The flag stays high as long as the read data reflects that error. I.e. read cycle active and address/contents unchanged)
error_corrected	Output	1	Flag that is held high as upon correcting a single bit error in the data (upon read). The flag stays high as long as the read data reflects that error (I.e. read cycle active and address/contents unchanged)
ecc_disable	input	1	If high, ECC mode is disabled and external access to all data bits is provided to the external world through

Name	I/O	Width	Description
			ecc_disable_wr_data and ecc_disable_rd_data
ecc_disable_wr_data	input	PRT_WDH + SECDEC	If ecc_disable is high, provides the raw data to be written to memory, including redundancy field
ecc_disable_rd_data	Output	PRT_WDH + SECDEC	If ecc_disable is high, outputs the raw data written to memory previously, including redundancy field.

Table 3. ES6010 External Interface Pin Description

(*) These values are pass-thru as per current implementation to the memory block.

Note that MEM_D_WDH = D_WDH + <number of redundancy bits>
Where <number of redundancy bits> is computed by the core generator at generation time as PRT_WDH + SECDEC (see Table 5 and **Error! Reference source not found.**)

Interface to local Memory

Name	I/O	Width	Description
mem_wr_data	Output	MEM_D_WDH	Effective memory write data (including redundancy bits) going to memory
mem_rd_wr	Output	1	Memory handshake line (high = rd, low = write)
mem_cs	Output	1	Memory chip-select line (high = selected, low = deselected)
mem_rd_data	Input	MEM_D_WDH	Effective memory read data (including redundancy bits) coming out from memory

Table 4. ES6010 Memory Interface Pin Description

Table 5 shows the number of extra bits required for redundancy as a function of user/application usable data width for SECDED case and for SEC case. Note that SECDED is a configuration/generation time option. If chosen it adds one extra redundancy bit to the total requirement vs SEC case (SECDEC=0)

Data Bits (D_WDH)	Redundancy Bits (SECDEC=1)	Redundancy Bits (SECDEC=0)
5..11	5	4
12..26	6	5

Data Bits (D_WDH)	Redundancy Bits (SECDEC=1)	Redundancy Bits (SECDEC=0)
27..57	7	6
58..120	8	7
121..247	9	8
248..502	10	9
503..1013	11	10
1014..2036	12	11

Table 5. ES6010 Redundancy Bit size as a function of Data Width

Figure 3 shows a write cycle followed by a read cycle to the same address where the contents of the memory has been altered/corrupted by 1 bit by the test bench prior to the read cycle (backdoor). This is explicit in the value of mem_rd_data differing from mem_wr_data in bit 12. We can also observe that both error_detected and error_corrected signals are asserted during the read cycle and that the final rd_data is actually correctly matching the written data during the write cycle (wr_data)

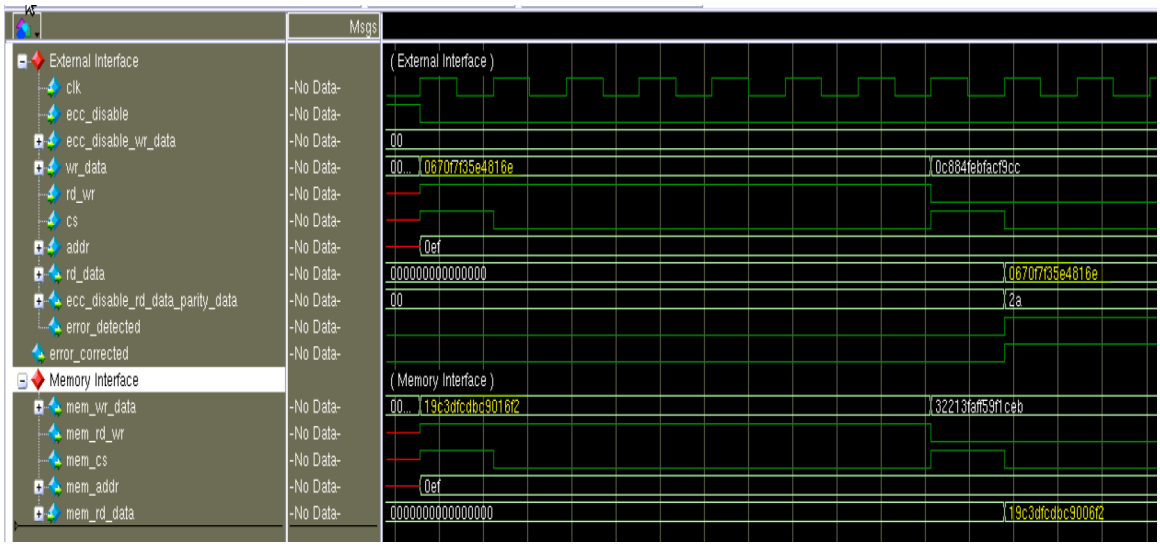

Figure 3. SECDEC=0, MEM_D_WDH=64, One bit corrupted in Memory (bit 12)

Figure 4 shows a write cycle followed by a read cycle to the same address where the contents of the memory has been altered/corrupted by 2 bits by the test bench prior to the read cycle (backdoor). This is explicit in the value of mem_rd_data differing from

mem_wr_data in bits 29 and 59. We can also observe that error_detected is asserted but not error_corrected during the read cycle and that the final rd_data does not match the written data during the write cycle (wr_data) as double error bits are detectable but not correctable.

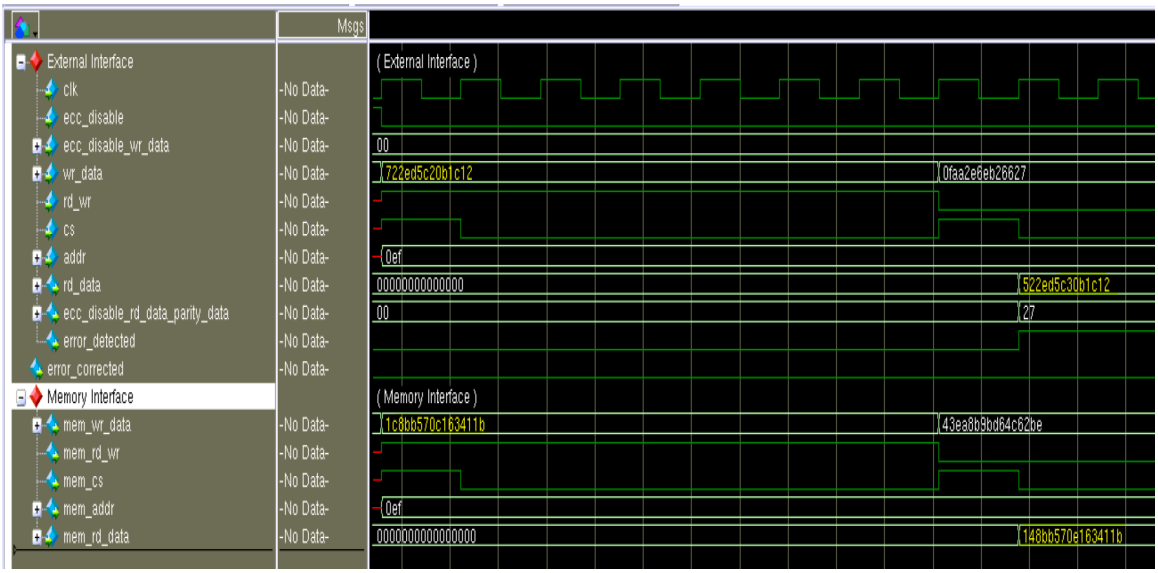

Figure 4. SECDEC=1, MEM_D_WDH=64, Two bits corrupted in Memory (bit 29 and bit 59)

Synthesis Summary

TSMC 28nm HPC technology library

- Synthesis results are based on WLM
- Voltage: 0.81 V
-Temperatute: 0 C
- HVT Cells alone for lowest leakage (higher speed possible w/o this constraint)

SECDEC=0 (SEC case)

Data Bits	*Area (um^2)*	*Area (Gae)*	*# Instances*	*Max delay (ns)*
128	1,603	5,850	1,641	0.65
256	3,095	11,297	3,195	0.75
512	5,519	20,146	6,018	0.84

SECDEC=1 (SECDEC case)

Data Bits	Area (um^2)	Area (Gae)	# Instances	Max delay (ns)
128	1,567	5,719	1,700	0.70
256	2,982	10,885	3,333	0.80
512	5,806	21,191	6,391	0.89

Deliverables

- ✓ Synthesizable Verilog RTL source code
- ✓ Simulation scripts
- ✓ Self-checking Test environment
  - ➢ Test-bench
  - ➢ Test-vectors
  - ➢ Expected results
- ✓ Synthesis scripts
- ✓ User Documentation

Sales Representatives

For pricing information or customization/integration services:

Esencia Technologies Inc.
3945 Freedom Circle Suite 360
Santa Clara, CA 95054
Tel: (408) 736-8284
Web: www.esenciatech.com
E-mail: sales@esenciatech.com

About Esencia

Esencia Technologies Inc. provides IP cores and design services for the semiconductor industry.