CRC32 Generator and Verifier (CC-131)

January 26, 1998

Product Specification

CoreEl MicroSystems

CoreEl MicroSystems

46750 Fremont Blvd. #208
Fremont, CA 94538 USA
Phone: +1 510-770-2277
Fax: +1 510-770-2288
URL: www.coreel.com
E-mail: sales@coreel.com

Features

- Fully compatible with ITU-T Recommendation I.363 for AAL-5
- Single clock operation
- Separate blocks for CRC32 Generator and CRC32 Verifier
- Fully synchronous operation
- Performance up to 30 MHz giving a throughput of 960 Mbps
- Accepts 32 bit data per clock on which CRC32 is computed
- Fully synthesizable Register Transfer Level (RTL) VHDL source available extra

Applications

The CRC32 Generator and Verifier cores can be used in telecommunications and networking equipment including ATM, SONET and Ethernet systems.

General Description

The CRC32 core is fully compliant to the ITU-T recommendation I.363 for AAL-5. The Core includes CRC32 Generator and CRC32 Verifier modules. The CRC32 core is ideally suited to be designed into a Xilinx FPGA with other high level functions like Segmentation and Reassembly (SAR).

AllianceCORE™ Facts

Core Specifics		
Device Family	Spartan	XC4000XL
CLBs - Generator: CLBs - Verifier:	125 127	125 127
IOBs - Generator: IOBs - Verifier:	100[1] 69[1]	100[1] 69[1]
CLKIOBs - Generator: CLKIOBs - Verifier:	1 1	1 1
System Clock f_{max}	29 MHz	30 MHz
Device Features Used	None	

Supported Devices/Resources Remaining		
	I/O[1]	**CLBs**
XCS30-4 PQ208 (Gen) XCS30-4 PQ208 (Ver)	29 60	451 449
XC4013XL-2 PQ160 (Gen) XC4013XL-2 PQ160 (Ver)	29 60	451 449

Provided with Core	
Documentation	Product Brief Specification Document Test Bench Design Document Test Scripts
Design File Formats	VHDL compiled, EDIF netlist
Constraint Files	crcver.ucf crcgen.ucf
Verification Tool	Script Based Behavioral VHDL Test Bench
Schematic Symbols	None
Evaluation Model	Behavioral VHDL
Reference Designs & Application Notes	ITU-T I.363 Specification
Additional Items	None

Design Tool Requirements	
Xilinx Core Tools	Alliance 1.3
Entry/Verification Tool	Model Tech V-System

Support
Support provided by CoreEl Microsystems

Note:
1. Assuming all core signals are routed off-chip.

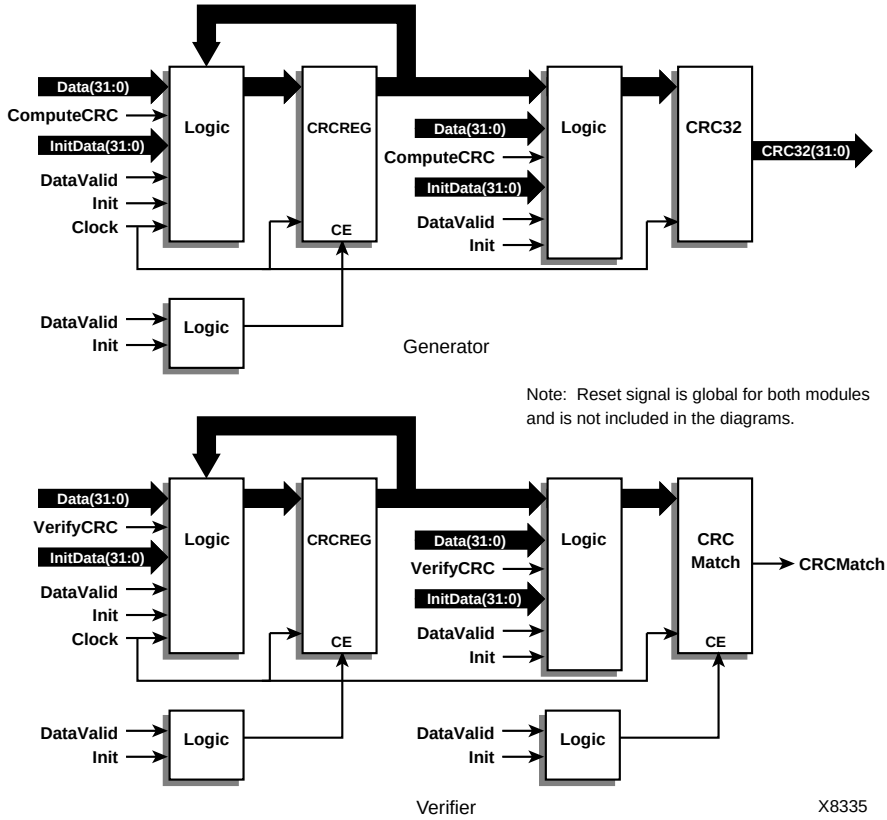

Figure 1: CRC32 Generator and Verifier Block Diagrams

The CRC32 core generates CRC32 remainder over incoming 32-bit data. The current CRC32 remainder value is always driven as an output from the CRC32 core. The CRC32 core supports resetting and preloading of the current CRC32 remainder. It enables the CRC32 core to be used to handle incoming data either as cell data or as packet data. It also supports verification of current CRC32 with a constant remainder as defined in ITU-T recommendation I.363.

Functional Description

The CRC32 Generator and Verifier modules are divided into blocks as shown in Figure 1. Operation of each module is described below.

Generator Operation

The CRC32 Generator computes and outputs the CRC32 for a Data packet. The Reset signal sets the CRC32 outputs 0.

Before starting CRC32 computation over a packet, the CRC register is programmed with the Initial value. According to the ITU I.363 specification the initial value is FFFFFFFF (hex). Assert Init for one clock cycle and drive InitData with FFFFFFFF (hex). From the next clock onwards, the data can be fed along with the qualifier DataValid. DataValid is asserted before the data is fed.

The current value of CRC32 is always available at the CRC32 output bus. Once the whole packet is fed, the ComputeCRC signal must be asserted for one clock cycle in order to get the final CRC32 which conforms to ITU I.363 specification, and which will be transmitted with the packet.

Verifier Operation

The CRC32 Verifier computes the CRC32 for the entire packet received and compares that with a constant value defined in the ITU I.363 specification. It generates a CRCMatch signal if there is a match. Reset signal deasserts the CRCMatch output.

Before starting CRC32 verification over a packet, the CRC register must be programmed with the Initial value FFFFFFFF (hex) as per ITU specification. Assert Init signal for one clock cycle and drive InitData with FFFFFFFF (hex). From the next clock onwards, the data can be fed along with the qualifier DataValid. DataValid is the signal which is to be asserted before the Data is fed.

Once the whole packet is fed, VerifyCRC must be asserted for one clock cycle in order to check for a CRC32 match. If it matches the constant value specified in I.363, the CRC-Match output signal will be driven high for one clock cycle.

Core Modifications

Normally, modifications are not possible by the user since the core is provided in a Xilinx netlist format. CoreEl can perform special modifications for additional charge. However source code is available for additional cost where the customer can make modifications. Contact CoreEl Microsystems for more information.

Pinout

The pinout is not fixed to any specific device I/O. Signal names for each module are provided in the block diagrams shown in Figure 1, and described in Table 1.

Verification Methods

This core has been used in larger ASICs and is silicon proven. The FPGA verification was done by back annotating the implementation and simulating in a Model technology V-System environment.

The test bench was written in VHDL with very powerful scripting capabilities and several scripts have been written for verifying the implementation. Additional tests can be added to the testbench by writing new scripts.

Recommended Design Experience

Knowledge of error correction in network systems is needed. User should be familiar with HDL design methodology including FPGA targeting. Using the testbench requires familiarity with V-System of Model Technology.

Available Support Products

CoreEl offers a test bench for verifying the core along with a complete line of ATM Core Cells for Xilinx FPGAs that perform the following functions:

- UTOPIA Interface
- Cell Delineation
- Cell Assembly
- CRC-10

Table 1: Core Signal Pinout.

Signal	Signal Direction	Description
CR32 Generator Signals		
Data(31:0)	Input	When *DataValid* is asserted, *Data* should be next dword of AAL5 packet over which CRC32 is computed.
ComputeCRC	Input	When sampled active, *CRC* output is inverted; should be asserted along with last payload to find CRC remainder to be transmitted as last dword of AAL5 trailer.
InitData(31:0)	Input	When *Init* is asserted, *InitData* should be partial CRC result for AAL5 packet processed so far.
DataValid	Input	Indicates input *Data* is valid and CRC32 to be computed for value on *Data*.
Init	Input	Loads *CRC* output with input *InitData*; asserted at start of new cell.
Clock	Input	Used to sample all other inputs; uses FPGA CLKIOB pin.
Reset	Input	Resets outputs to 0; asserted on power-up/reset.
CRC32(31:0)	Output	Indicates current CRC remainder of AAL5 packet being processed. During AAL5 packet processing, *CRC32* should be sampled and value stored in memory at each packet change. When same AAL5 packet processing resumes, stored CRC value should be driven onto *InitData*, and *Init* pulse should be generated to maintain CRC consistency over AAL5 packet.
CRC32 Verifier Signals		
Data(31:0)	Input	When *DataValid* is asserted, *Data* should be next dword of AAL5 packet over which CRC32 is computed.
VerifyCRC	Input	When sampled active, *CRC* output is compared with an ITU I.363 defined constant; asserted with AAL5 trailer to find if AAL5 packet was received without errors.

Signal	Signal Direction	Description
InitData(31:0)	Input	When *Init* is asserted, *InitData* should be partial CRC result for AAL5 packet processed so far.
DataValid	Input	Indicates input *Data* is valid and CRC32 to be computed for value on *Data*.
Init	Input	Loads *CRC* output with input *InitData*; asserted at start of new cell.
Clock	Input	Used to sample all other inputs; uses FPGA CLKIOB pin.
Reset	Input	Resets outputs to 0; asserted on power-up/reset.
CRCMatch	Output	Asserted in response to *VerifyCRC* if current CRC remainder matches an ITU I.363 defined constant.

Ordering Information

For information on this or other products mentioned in this specification, contact CoreEl Microsystems directly from the information provided on the front page.

Related Information

The ATM Forum

The ATM Forum publishes specifications regarding ATM. For more information, contact them as follows:

ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1 650-949-6700
Fax: +1 650-949-6705
E-mail: info@atmforum.com
URL: www.atmforum.com

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

For general Xilinx literature, contact:

Phone: +1 800-231-3386 (inside the US)
+1 408-879-5017 (outside the US)
E-mail: literature@xilinx.com

For AllianceCORE™ specific information, contact:

Phone: +1 408-879-5381
E-mail: alliancecore@xilinx.com
URL: www.xilinx.com/products/logicore/alliance/tblpart.htm