AIN SHAMSUNIVERSITY - FACULTY OF ENGINEERING

International Credit Hours Engineering Programs -iCHEP

MAJOR TASK

Configurable CRC Module Implementation using VHDL

Electronic Design Automation (CSE215/CSE312)

Project Objective

The aim of this project is to design, implement and test a simple configurable CRC module using VHDL. The project main concern is the design of a synthesizable VHDL code for the CRC module and its test bench using design methodologies studied in class.

Project description

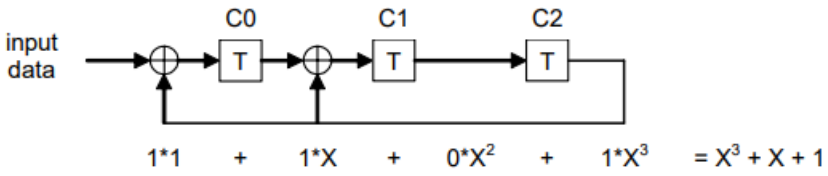
A Cyclic Redundancy Check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. There are many CRC polynomials available, used depending on the specific application.

The produced CRC Module should meet the following requirements:

1- It should support input data in the following ranges (8 bits, 16 bits, 32 bits)
2- It should support 3-bit CRC and 6-bit CRC calculations using the following polynomials:

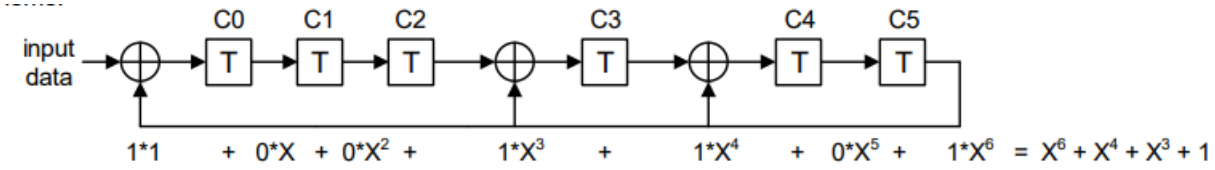
CRC Output	polynomial	initialization value
3-bits	*f(x)=1+x+x3*	"111"
6-bits	*f(x)= $x^6+x^4+ x^3+1$*	"010101"

3-bit CRC

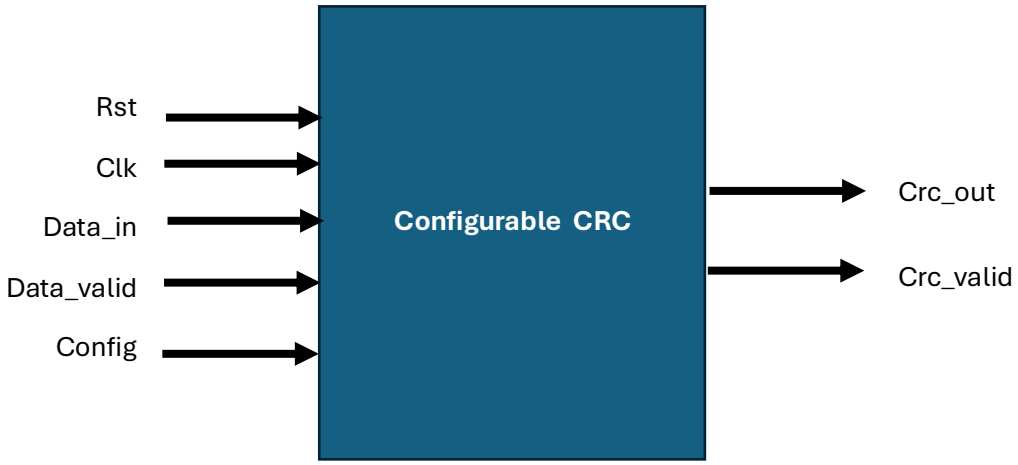


3 signals with delay w 2 x-or

6-bit CRC



Module Specification



Port name	Direction	Size	Description
Rst	input	1-bit	System Reset – Active high
Clk	input	1-bit	System Clock – Rising Edge
Data_in	input	32-bits	Input Data (LSB aligned)
Data_valid	input	1-bit	Indication that input data is valid at the input – Active high
Config	input	3-bits	Configuration commands 000, 100 – Not used 001 – 8-bits input-3 bits CRC 010 – 16-bits input-3 bits CRC 011 – 32-bits input-3 bits CRC 101 – 8-bits input-6 bits CRC 110 – 16-bits input-6 bits CRC 111 – 32-bits input-6 bits CRC
Crc_out	output	6-bits	Calculated CRC output (LSB aligned)
Crc_valid	output	1-bit	Indication that the calculated CRC is valid at the output

Report outlines

The project report will include a detail description of the design problem, background information, the design approach and detailed design information, test scenarios, cases and results, test data, conclusions and recommendations.

The report should contain the following sections:

1. Introduction: Introduction to CRC. How is it calculated? Why is it used in digital designs?
2. Literature Review: Survey of different approaches of implementing CRC modules using VHDL.
3. System Design: Detailed description of the module design and implementation steps, including block diagram, data flow, state machine implementation, etc.
4. System Simulation: Detailed discussion about test strategy, test cases, test results, validation of results, etc.
5. VHDL Code: Complete VHDL Code of the system
6. Simulation Waveforms: Complete Simulation waveforms for all testcases
7. Implementation Reports: Synthesis and Implementation Reports of Area, power and Timing with no violations.
8. Discussion: Comment on the design challenges and report any known bugs

Milestone #1:

A working VHDL design for CRC-3bits for 8-bit input using the polynomial *f(x)=1+x+x3 and initialization value "111".*

Milestone #2:

A working VHDL design for a configurable CRC-3bits for 8/16/32-bit input using the polynomial *f(x)=1+x+x3 and initialization value "111".*

Final Delivery:

A configurable CRC Module that meets the following requirements:

1- Supports input data in the following ranges (8 bits, 16 bits, 32 bits)
2- Supports 3-bit CRC and 6-bit CRC calculations using the following polynomials:

CRC Output	polynomial	initialization value
3-bits	*f(x)=1+x+x3*	"111"
6-bits	*f(x)= $x^6+x^4+ x^3+1$*	"010101"

Evaluation Criteria

1. **90% and above:** Your report must be of outstanding quality and fully meet the requirements of the project specifications stated. You must show independent thinking and apply this to your work showing originality and consideration of key issues. There must be evidence of wider reading on the subject. In addition, your report should:
 illustrate a professional ability of designing digital logic using VHDL,
 express a deep understanding of the simulation and synthesis of logic circuits,
 and applying, masterly, the learned knowledge in the report.

2. **76% - 89%:** Your project must be of good quality and meet the requirements of the project specification stated. You must demonstrate some originality in your work and show this by applying new learning to the key issues of the project. There must be evidence of wider reading on the subject. In addition, your report should:
 illustrate a Good ability of designing digital logic using VHDL
 express a very Good understanding of the simulation and synthesis of logic circuits,
 and applying most of the learned knowledge in the report.

3. **67% - 76%:** Your project must be comprehensive and meet all of the requirements stated by the project specification. You must show a good understanding of the key concepts and be able to apply them to solve the problem set by the project specs. There must be enough depth to your work to provide evidence of wider reading. In addition, your report should:
 illustrate a moderate ability of designing digital logic using VHDL,
 express a good understanding of the simulation and synthesis of logic circuits,
 and applying most of the learned knowledge, correctly, in the report.

4. **50% - 67%:** Your work must be of a standard that meets the requirements stated by the project specification. You must show a reasonable level of understanding of the key concepts and principles and you must have applied this knowledge to the project problem. There should be some evidence of wider reading. In addition, your report should:
 illustrate a fair ability of designing digital logic using VHDL,
 express a fair understanding of the simulation and synthesis of logic circuits,
 and applying some of the learned knowledge, correctly, in the report.

5. **Below 50%:** Your project is of poor quality and does not meet the requirements stated by the project specification. There is a lack of understanding of key concepts and knowledge and no evidence of wider reading. In addition, your report would:
 Illustrate an inability of designing digital logic using VHDL,
 Failed to complete the basics of the simulation and synthesis of logic circuits,

 Failed to apply correctly the learned knowledge for proposing a valid solution.