Sudan University of Science and Technology

College of Graduate Studies

A thesis submitted in partial fulfillment for the award of the degree of

M.Sc. in Computer and Network Engineering

Design of an Efficient Cyclic Redundancy Check-32 using Field Programmable Gate Array

تصميم إختبار التكرار الدوري فعال من النوع 32 باستخدام مصفوفة

البوابات المنطقية القابلة للبرمجة

Prepared by:

Mohamed Salah Eldin Abdulanbi

Supervisor:

Dr. Hisham Ahmed

SEPTEMPER 2018

الآية

قالَ تَعَالى:

صَدقَ الله العَظيم

DEDICATION

I dedicate this project report to Allah Almighty for his infinite mercy that he has granted to me through my studies, and I also dedicate this work to my family.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my supervisor Dr. Hisham Ahmed for his unlimited support, guidance and patience.

Besides that, I would like to thank my family and friends for their understanding and support toward me for completing this research.

ABSTRACT

The streaming of data over the networks is increasing day by day. the current applications, video web sites, Internet of things and Machine to Machine made new era of the data over the internet. Accordingly, the networking devices should be evolved to meet this change. the hardware and the software needs continuously to be upgraded and the research in this area cannot be stopped. Part of this research area is the cyclic redundancy check (CRC) which is the error detection technique that used for data integrity. The exist CRC32 that being implemented on the networking devices cannot meet the high speed requirements which is expected to 100 Gbps in the core, aggregation and backbone networking devices such as routers and switches. So the aim of this research is to perform a design of CRC32 capable to achieve a throughput equal to 100 Gbps. The design of the CRC32 performed using slicing by 16 algorithm that synthesized in Xilinx Virtex-7 Field Programmable Gate Array (FPGA) and simulated by Xilinx Isim. The result show that the achieved throughput is equal to 102.4 Gbps.

المستخلص

إنّ تدفق البيانات عبر الشبكات يتزايد يوما بعد يوم ، التطبيقات التي تعمل اليوم ، مواقع مشاهدة الفيديو على الانترنت ، انترنت الأشياء و اتصالات الآلات مع بعضها البعض كونت عهدا جديداً فيما يخص نقل البيانات على الانترنت. بذلك أصبحت الأجهزة العاملة في مجال الشبكات بحاجة إلى بحث و تطوير مستمر من ناحية العتاد و البرمجيات من ناحية أخرى لمواجهة هذا التغيير. من المجالات المتعلقة بالبحث و التطوير في مجال تطوير الشبكات هو تطوير الآلية التي يطبق بها اختبار التكرار الدوري المستخدم في اكتشاف الأخطاء والذي يضمن وصول البيانات بصورة سليمة الى المستقبل. اختبار التكرار الدوري من النوع 32 المستخدم حاليا لن يستطيع مجابهة تدفق البيانات المتوقع وصوله حتى 100 جيجابت / ثانية في أجهزة الشبكات مثل الراوترات و السويتشات العاملة بالمناطق المركزية ونقاط التجمع أو مراكز البيانات. لذلك يهدف البحث لتصميم اختبار التكرار الدوري ليكون قادرا على العمل مع تدفق بيانات يصل الى 100 جيجابت / ثانية. تم التصميم باستخدام خوارزمية التقسيم الى 16 والتي تم توليفها لتعمل على مصفوفة البوابات المنطقية القابلة للبرمجة من النوع زايلينكس – فيرتكس7. تم عمل المحاكاة على برنامج آي سيم . النتائج وضحت أن الخرج المنتج يصل الى 102.4 جيجابت / ثانية.

Table of Content

CHAPTER THREE: DESIGN AND SIMULATION

CHAPTER FOUR: RESULT DISCUSSION

CHAPTER FIVE: CONCULSION AND RECOMMENDATIONS

APPENDICES

List of Figures

List of Tables

List of Abbreviations

CLB	Configurable Logic Block
CMT	clock management tiles
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
FPGA	Field Programmable Gate Array
HDL	hardware description language
IoT	Internet of Things
IP	Internet Protocol
LUT	Look up Table
M2M	Machine to Machine
RAM	Random Access Memory
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLSI	Very Large Scale Integration
VoD	Video on Demand

CHAPTER ONE
INTRODUCTION

1.1 Overview

The internet and data communication considered one of the most powerful and important creations in all of human history. The internet already has had its impact on science, government, business, communication and humanity. The advances in telecommunications technologies and services facilitated the massive exchange of data among client devices, data center and clouds. novel applications are widely found that span across diverse fields as never before. Nevertheless, the Internet of Things (IoT) is a recent communication paradigm that envisions a near future, in which the objects of everyday life will be equipped with microcontrollers, transceivers for digital communication, and suitable protocol stacks that will make them able to communicate with one another and with users, becoming an integral part of the Internet [1]. The IoT concept, hence, aims at making the Internet even more immersive and pervasive. Furthermore, by enabling easy access and interaction with a wide variety of devices such as, for instance, home appliances, surveillance cameras, monitoring sensors, actuators, displays, vehicles, and so on, the IoT will foster the development of a number of applications that make use of the potentially enormous amount and variety of data generated by such objects to provide new services to citizens, companies, and public administrations. According to that, the amount of data in the backbones, datacenters and the core of networks will be increased. The

CRC32 (Cyclic Redundancy Check 32) which is the error detection and correction technique that used in the Ethernet will be affected by the huge increase of the data traffic. The existing throughput of CRC32 in the backbone and core devices cannot handle it, accordingly the throughput of the CRC32 should be increased as well.

On the other hand, the Internet of Things considered the backbone of smart cities, the smart city defined as “a city connecting the physical infrastructure, the information technology infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city” [2]. figure 1.1 illustrate the domain of IoT applications, Beside the IoT, Machine-to-Machine (M2M) which is refers to direct communication between devices using any communications channel, including wired and wireless. Machine-to-Machine communication can include industrial instrumentation, enabling a sensor or meter to communicate the data it records to application software that can use it. In addition to that, there is the concept of Big Data which is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy. According to these new technologies and applications the data that will be exchanged between devices will be very huge and the trend of the transferred data will be increased as illustrated in the next sections. According to that the networking devices should be capable to process and handle this huge traffic of data frames.

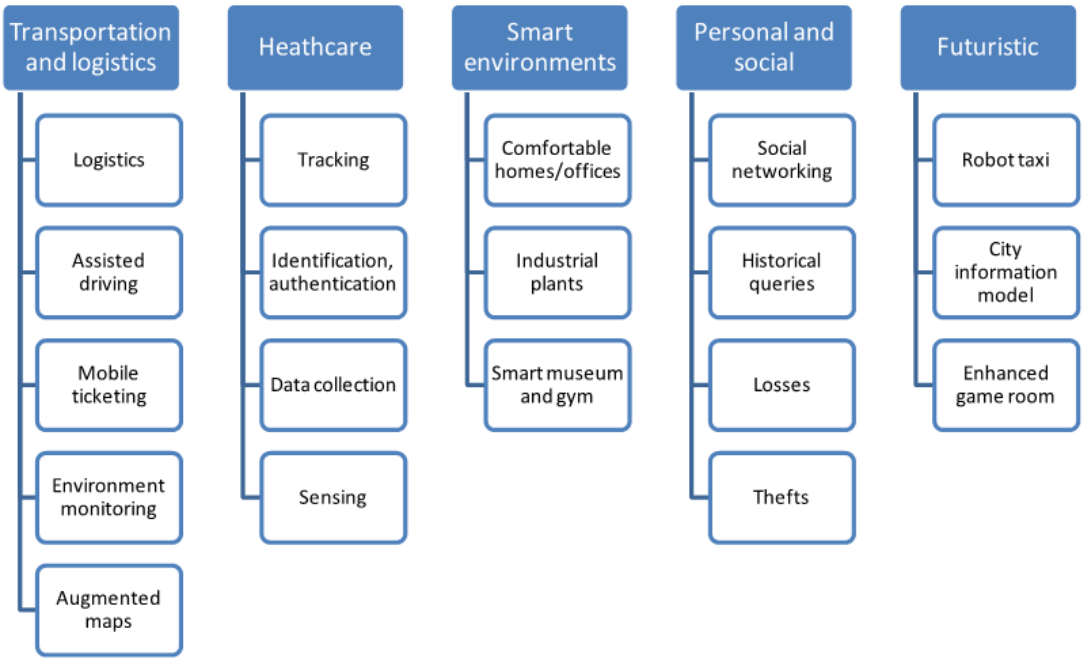


Figure 1.1: IoT application domain

1.2 Internet traffic trend

Total internet traffic has experienced dramatic growth in the past two decades. More than 20 years ago, in 1992, global internet networks carried approximately 100 GB of traffic per day. Ten years later, in 2002, global internet traffic amounted to 100 GB per seconds (GBps). In 2015, global internet traffic reached more than 20000 GBps [3]. Table 1.1 provides a view of historical benchmarks for total internet traffic.

By 2020 global IP traffic will reach 2.3 Zettabyte per year (Zettabyte is10^{21} bytes or 1 trillion Gigabytes), or 194 Exabyte (Exabytes is 10^{18} or 1 billion Gigabytes) per month. Per capita monthly IP traffic will reach 25 Gigabyte. The global fixed broadband speed will reach 47.7 Mbps by 2020. Not yet, by 2020 every minute a million minutes of video content will cross the network and the amount of video on demand (VoD) will be equivalent to 7.2 million DVD per month. Mobile data traffic will reach 30.6 Exabytes per month by

2020. Globally, IP traffic will reach 511 terabits per seconds (Tbps) in 2020, this is the equivalent of 142 million people streaming Internet high-definition (HD) video simultaneously, all day, every day [3].

Table 1.1: Historical Internet Context [3].

Year	Global internet traffic
1992	100 GB per day
1997	100 GB per hour
2002	100 GBps
2007	2000 GBps
2015	20235 GBps
2020	61386 GBps

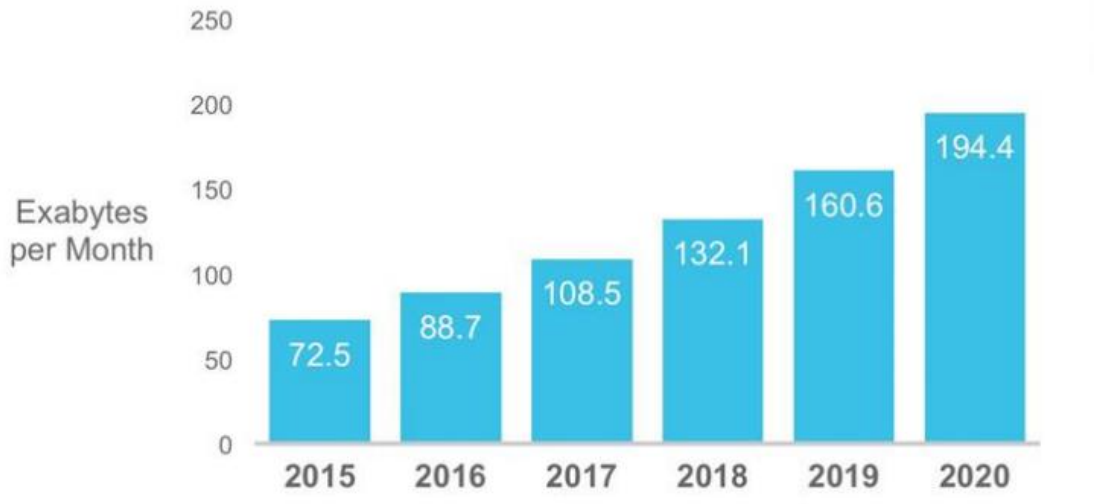


Figure 1.2: The globally forecasted IP traffic per month [3].

A growing number of M2M applications, such as video surveillance, healthcare monitoring, transportation, and package or asset tracking, are contributing in a major way to the growth of devices and connections. By 2020 too, M2M connections will be 46 percent of the total devices and connections. M2M connections will be the fastest-growing category [3]. The global M2M IP traffic will grow to 6.3 Exabyte by 2020, the amount of traffic is growing faster than the number of connection because of increase of deployment of video applications on M2M connections and increase use of applications, such as telemedicine and smart car navigation systems, which require greater bandwidth and lower latency [3].

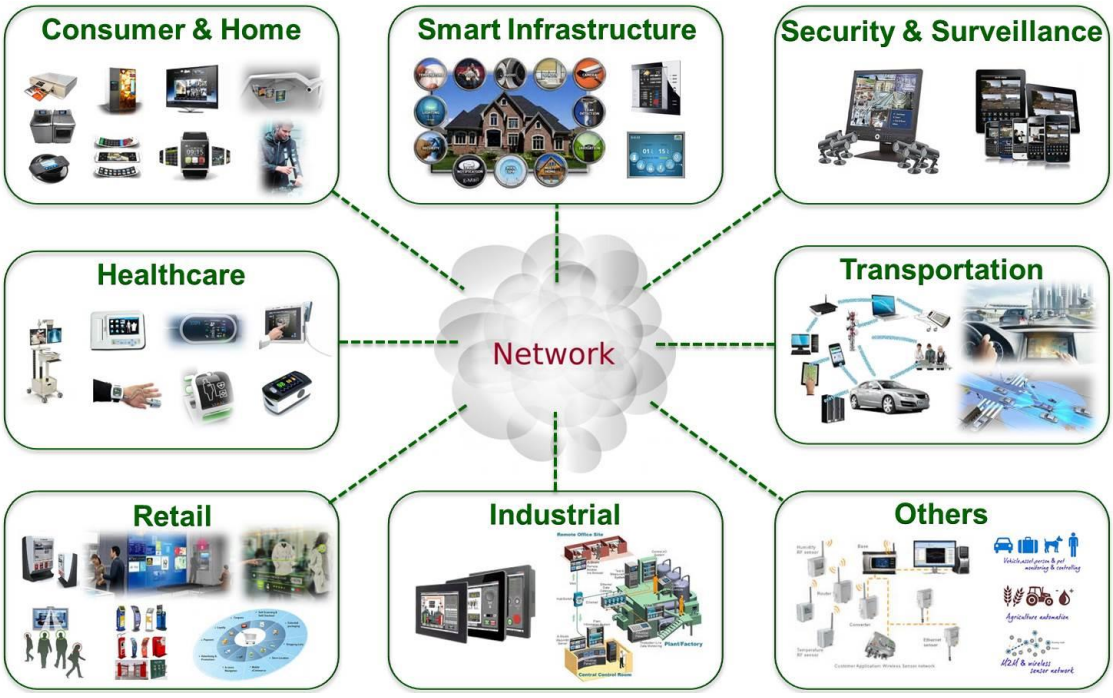


Figure 1.3: Internet of Things Applications

1.3 Problem Statement

The internet traffic continuously growing very fast, the Internet of Things and Machine to Machine devices will generate traffic which will be very high when aggregated at the core switches, with the high speed requirements, the CRC algorithm being implemented in the backbone, aggregation and distribution switches, data centers and core routers and switches should also be fast enough to meet the wire speed which is going to be 100 Gbps.

1.4 Objective

The main objective of this project is to enhance the throughput of CRC-32 to achieve 100 Gbps which can positively affect the operations of the core, distribution, and backbone switches and routers.

1.5 Methodology

The proposed method in this project is to design the CRC-32 for the Ethernet using VHDL (Very High Speed Integrated Circuit Hardware Description Language) and FPGA (Field Programmable Gate Array). the slicing by N algorithm selected to be enhanced and then simulated. The VHDL code for the algorithm will be written in a high efficiency way to reduce the area of the hardware, the delay and the power consumption will be considered in the design as well. The VHDL code synthesized based on a suitable FPGA device using Xilinx synthesis tool and simulated using Xilinx ISE simulator. The result of the simulation will be compared with the related

works in term of throughput, area utilization, number of CLBs and power consumption.

1.6 Thesis Outline

The purpose of this research is to enhance the processing of the CRC32 to meet the high speed requirements. In chapter two, the literature reviewed discussed. Chapter three, discuss the design and the simulation. In the design, the generation of lookup tables and the slicing by 16 algorithm discussed besides the design of the circuit on the FPGA using VHDL, in addition to that the needed stimulus to simulate the design specified. Chapter four discuss the simulation results in term of processing speed, power consumption and device utilization. finally, chapter five discuss the conclusion and the recommendation of this work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

The following review of related works covers different topics related to the design and the implementation of the CRC. The purpose of this reviewing of CRC design and implementation is to provide an overview of what has already been built and how these existing systems do or do not address portions of the problem statement.

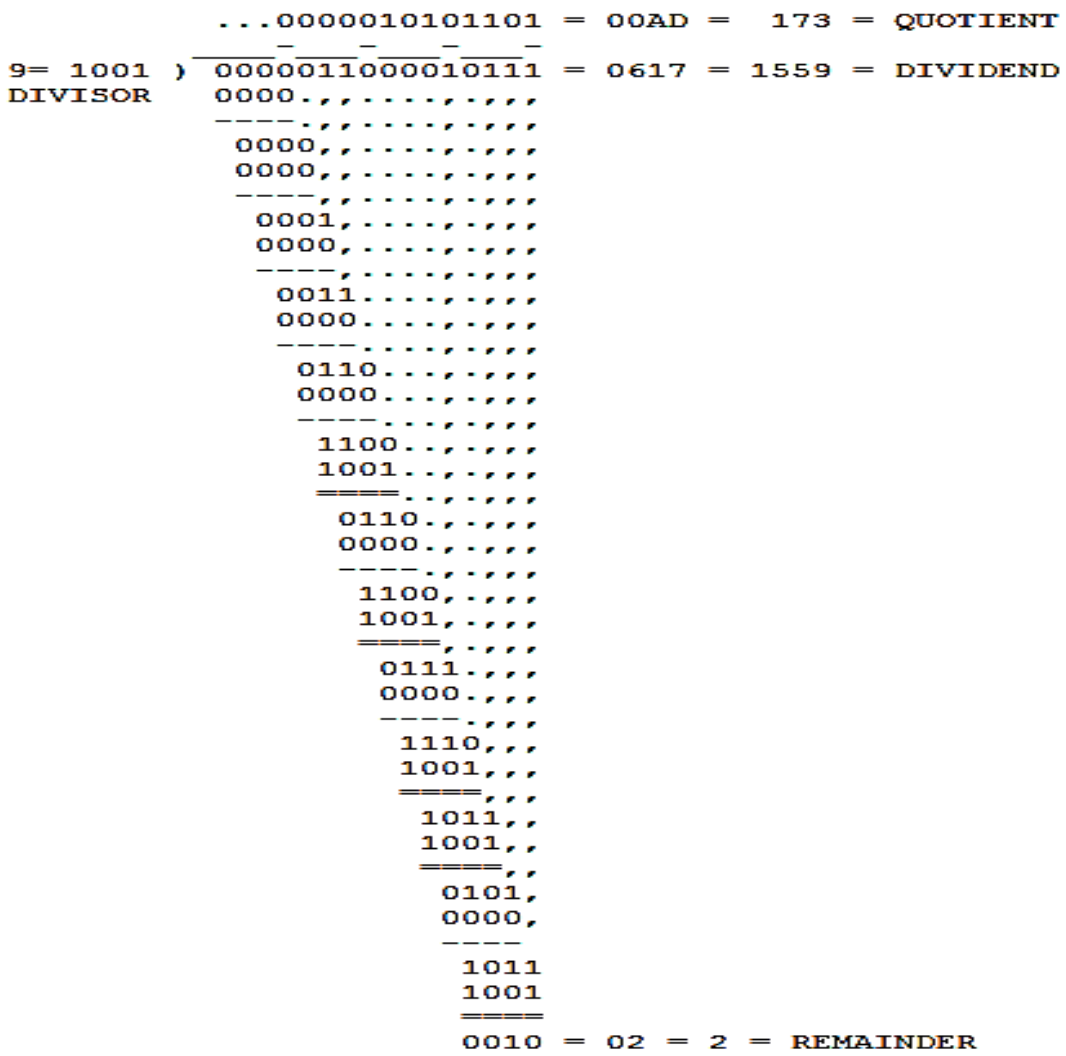
2.2 Error detection

The aim of an error detection technique is to enable the receiver of a message transmitted through a noisy (error-introducing) channel to determine whether the message has been corrupted. To do this, the transmitter constructs a value (called a checksum) that is a function of the message, and appends it to the message. The receiver can then use the same function to calculate the checksum of the received message and compare it with the appended checksum to see if the message was correctly received [5].

2.3 The Basic Idea Behind CRC Algorithms

The basic idea of CRC algorithms is simply to treat the message as an enormous binary number, to divide it by another fixed binary number, and to make the remainder from this division the checksum. Upon receipt of the message, the receiver can perform the same division and compare the remainder with the "checksum" (transmitted remainder) [5].

Example: Suppose the message consisted of the two bytes (6,23). These can be considered to be the hexadecimal number 0617 which can be considered to be the binary number 0000-0110-0001-0111. Suppose that we use a checksum register one-byte wide and use a constant divisor of 1001, then the checksum is the remainder after 0000-0110-0001 0111 is divided by 1001. While in this case, this calculation could obviously be performed using common garden variety 32-bit registers, in the general case this is messy. So instead, we will do the division using long division in binary:



Numerous varieties of cyclic redundancy checks have been incorporated into technical standards. Koopman and Chakravarty recommend selecting a polynomial according to the application requirements and the expected distribution of message lengths [13]. The design of the 32-bit polynomial most commonly used by standards bodies, CRC-32-IEEE, was the result of a joint effort for the Rome Laboratory and the Air Force Electronic Systems Division by Joseph Hammond, James Brown and Shyan-Shiang Liu of the Georgia Institute of Technology and Kenneth Brayer of the MITRE Corporation. The earliest known appearances of the 32-bit polynomial were in their 1975 publications: Technical Report 2956 by Brayer for MITRE, published in January and released for public dissemination through DTIC in August [14] and Hammond, Brown and Liu's report for the Rome Laboratory, published in May [15]. During December 1975, Brayer and Hammond presented their work in a paper at the IEEE National Telecommunications Conference: the IEEE CRC-32 polynomial is the generating polynomial of a Hamming code and was selected for its error detection performance [16]. The table below lists only the polynomials of the various algorithms in use [17].

Table 2.1: various types of CRC

Name	Uses	Polynomial
CRC-1	most hardware	0x1
CRC-3-GSM	mobile networks	0x3
CRC-4-ITU	ITU-T G.704	0x3
CRC-5-EPC	Gen 2 RFID	0x09
CRC-6-CDMA2000-A	mobile networks	0x27
CRC-8	DVB-S2	0xD5

CRC-10	ATM	0x233
CRC-11	FlexRay	0x385
CRC-14-DARC	Data Radio Channel	0x0805
CRC-15-CAN	CAN	0x4599
CRC-16-IBM	USB	0x8005
CRC-24-WCDMA	OS-9 RTOS.	0x800063
CRC-32	Ethernet	0x04C11DB7

2.4 Field Programmable Gate Array (FPGA)

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the designer after manufacturing – hence “field programmable”. The FPGA configuration is generally specified using a hardware language description (HDL).

FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. Figure 3.4 illustrate the relationship between logic blocks, I/O blocks, and programmable routing on an FPGA.

The following FPGA specification are important to be consider when designing an FPGA application.

- Number of configurable logic blocks
- Number of fixed function logic blocks, such as multipliers
- Size of memory resources, such as embedded block RAM

Contemporary field-programmable gate arrays (FPGAs) have large resources of logic gates and RAM blocks to implement complex digital computations. As FPGA designs employ very fast I/Os and bidirectional data buses, it becomes a challenge to verify correct timing of valid data within setup time and hold time. An FPGA can be used to solve any problem which is computable. This is trivially proven by the fact FPGA can be used to implement a soft microprocessor, such as the Xilinx MicroBlaze. Their advantage lies in that they are sometimes significantly faster for some applications because of their parallel nature and optimality in terms of the number of gates used for a certain process.

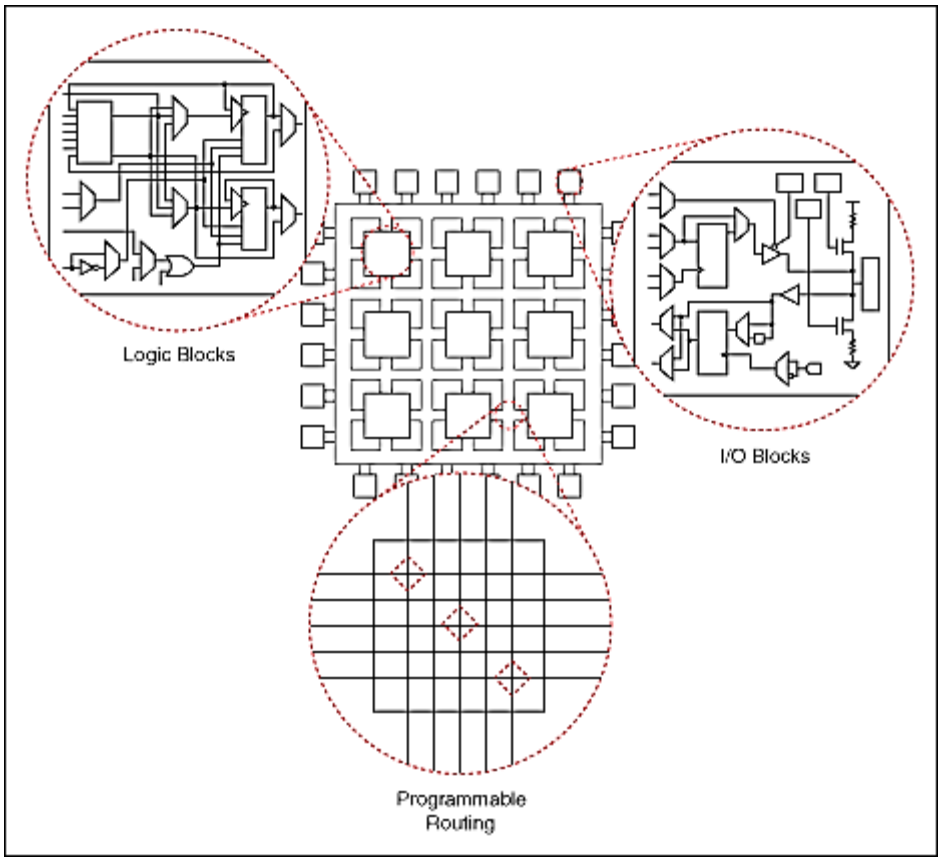


Figure 2.1: internal component of FPGA

2.5 Xilinx FPGA Virtex-7

Xilinx virtex-7 FPGA (xc7vx330t) selected to be used in this project. This type of FPGA optimized for highest system performance and capacity, consuming 50% less power than the previous generation devices [12]. It is belonging to 7 series FPGAs as called by Xilinx. Furthermore, the Lookup tables (LUTs) in 7 series FPGAs can be configured as either one 6-input LUT (64-bit ROMs) with one output, or as two 5-input registered in a flip-flop. Four such LUTs and their eight flip-flops as well as multiplexers and arithmetic carry logic form a slice, and two slices form a configurable logic block (CLB). Four of the eight flip-flops per slice (one per LUT) can optionally configured as latches. Between 25-50% of all slices can also use their LUTs as distributed 64-RAM or as 32-bit shift registers (SRL32) or as two SRL16s [12]. Each 7 series FPGA has up to 24 clock management tiles (CMTs), each consisting of one mixed-mode clock manager and phase-locked loop.

Table 2.2: Virtex-7 FPGA Feature Summary [12]

Device[1]	Logic Cells	Configurable Logic Blocks (CLBs)		DSP Slices[3]	Block RAM Blocks[4]			CMTs[5]	PCIe[6]	GTX	GTH	GTZ	XADC Blocks	Total I/O Banks[7]	Max User I/O[8]	SLRs[9]
		Slices[2]	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)									
XC7V585T	582,720	91,050	6,938	1,260	1,590	795	28,620	18	3	36	0	0	1	17	850	N/A
XC7V2000T	1,954,560	305,400	21,550	2,160	2,584	1,292	46,512	24	4	36	0	0	1	24	1,200	4
XC7VX330T	326,400	51,000	4,388	1,120	1,500	750	27,000	14	2	0	28	0	1	14	700	N/A
XC7VX415T	412,160	64,400	6,525	2,160	1,760	880	31,680	12	2	0	48	0	1	12	600	N/A
XC7VX485T	485,760	75,900	8,175	2,800	2,060	1,030	37,080	14	4	56	0	0	1	14	700	N/A
XC7VX550T	554,240	86,600	8,725	2,880	2,360	1,180	42,480	20	2	0	80	0	1	16	600	N/A
XC7VX690T	693,120	108,300	10,888	3,600	2,940	1,470	52,920	20	3	0	80	0	1	20	1,000	N/A
XC7VX980T	979,200	153,000	13,838	3,600	3,000	1,500	54,000	18	3	0	72	0	1	18	880	N/A
XC7VX1140T	1,139,200	178,000	17,700	3,360	3,760	1,880	67,680	24	4	0	96	0	1	22	1,100	4
XC7VH580T	580,480	90,700	8,850	1,680	1,880	940	33,840	12	2	0	48	8	1	12	600	2
XC7VH870T	876,160	136,900	13,275	2,520	2,820	1,410	50,760	18	3	0	72	16	1	13	650	3

2.6 Polynomial Arithmetic

The word you will hear all the time when dealing with CRC algorithms is the word "polynomial". A given CRC algorithm will be said to be using a particular polynomial, and CRC algorithms in general are said to be operating using polynomial arithmetic. What does this mean?

Instead of the divisor, dividend (message), quotient, and remainder (as described in the previous section) being viewed as positive integers, they are viewed as polynomials with binary coefficients. This is done by treating each number as a bit-string whose bits are the coefficients of a polynomial. For example, the ordinary number 23 (decimal) is 17 (hex) and 10111 binary and so it corresponds to the polynomial:

$$1 \times x^4 + 0 \times x^3 + 1 \times x^2 + 1 \times x^1 + 1 \times x^0 \qquad (2.1)$$

or, more simply:

$$x^4 + x^2 + x^1 + x^0 \qquad (2.2)$$

Using this technique, the message, and the divisor can be represented as polynomials and we can do all our arithmetic just as before, except that now it is all cluttered up with Xs. For example, suppose we wanted to multiply 1101 by 1011. We can do this simply by multiplying the polynomials:

$$(x^3 + x^2 + x^0)(x^3 + x^1 + x^0)$$
$$= (x^6 + x^4 + x^3 + x^5 + x^3 + x^2 + x^3 + x^1 + x^0) = x^6 + x^5 + x^4 + 3x^3 + x^2 + x^1 + x^0 \qquad (2.3)$$

At this point, to get the right answer, we have to pretend that x is 2 and propagate binary carries from the $3x^3$yielding

$$x^7 + x^3 + x^2 + x^1 + x^0 \quad (2.4)$$

It is just like ordinary arithmetic except that the base is abstracted and brought into all the calculations explicitly instead of being there implicitly. So what is the point? The point is that IF we pretend that we do not know what x is, we cannot perform the carries. We do not know that $3x^3$is the same as $x^3 + x^3$ because we do not know that x is 2. In this true polynomial arithmetic, the relationship between all the coefficients is unknown and so the coefficients of each power effectively become strongly typed; coefficients of $x^2$are effectively of a different type to coefficients of x^3. With the coefficients of each power nicely isolated, mathematicians came up with all sorts of different kinds of polynomial arithmetics simply by changing the rules about how coefficients work. Of these schemes, one in particular is relevant here, and that is a polynomial arithmetic where the coefficients are calculated MOD 2 and there is no carry; all coefficients must be either 0 or 1 and no carries are calculated. This is called "polynomial arithmetic mod 2". Thus, returning to the earlier example:

$$(x^3 + x^2 + x^0)(x^3 + x^1 + x^0)$$
$$= (x^6 + x^4 + x^3 + x^5 + x^3 + x^2 + x^3 + x^1 + x^0)$$
$$= x^6 + x^5 + x^4 + 3 \times x^3 + x^2 + x^1 + x^0 \quad (2.5)$$

Under the other arithmetic, the $3x^3$term was propagated using the carry mechanism using the knowledge that x=2. Under "polynomial arithmetic mod

2", we do not know what x is, there are no carries, and all coefficients have to be calculated mod 2. Thus, the result becomes:

$$= x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + x^0 \quad (2.6)$$

Thus polynomial arithmetic mod 2 is just binary arithmetic mod 2 with no carries. While polynomials provide useful mathematical machinery in more analytical approaches to CRC and error-correction algorithms, for the purposes of exposition they provide no extra insight and some encumbrance and have been discarded in the remainder of this document in favor of direct manipulation of the arithmetical system with which they are isomorphic: binary arithmetic with no carry [5].

2.7 CRC using linear feedback shift register

Simple linear feedback shift register can be used for hardware implementation of bit wide CRC – figure 2.3. While such a circuit is simple and can run at very high clock speeds, it suffers from the limitation that the stream must be bit-serial. This means that n clock cycles will be required to calculate the CRC values for an n-bit data stream [11].

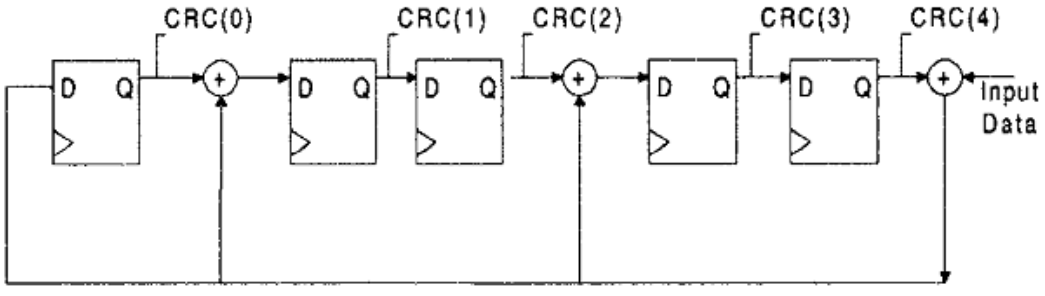


Figure 2.2: a bit wide CRC for the polynomial $1+X+X^3+X^5$ [11]

In many high speed data networking applications where data frames need to be processed at high speeds, this latency is intolerable and hence, implementation of CRC generation and checking on a parallel stream of data becomes desirable. So, for e.g., a byte-wide CRC generator can be implemented to generate CRC for an 8- bit wide data stream in one clock cycle as opposed to 8 clock cycles with a conventional CRC generator [11].

The author in [11] produce the CRC value operating on a multi-bit data stream, as would occur if the CRC were computed one bit at a time over the whole data stream. Based on this concept, an n-bit CRC generator can be described in a hardware description language (HDL). The technique that used in this work is based on symbolic simulation. The idea here is to use symbols instead of binary values and simulate a behavioral description of the circuit using these symbols. In the implementation of this technique a behavioral loop iteration scheme is implemented for the CRC-32 polynomial. The loop iteration scheme calculates the CRC value iteratively over a complete stream of data. The only input to the program is an integer generic representing the size of the data stream. After the symbolic simulation, using VHDL File I/O techniques, the program outputs a set of equations for each of the 32 register bits for the CRC-32 polynomial. The results show that the symbol based method is superior both in area and timing to the conventional method especially when the width of the CRC generator is large. Using the symbolic simulation technique, the authors was able to derive the Boolean expressions for the CRC registers for any data width. This further allowed them to achieve performance enhancements by removing redundancies in the expressions for the CRC registers [11].

2.8 Concurrent VLSI implementation of CRC

Sait and Hassan translate the byte-wise CRC algorithm that presented in a software implementation in [6] into VLSI chip. They prefer the hardware implementation for several reasons namely:

- Bit-wise software implementations are very slow.
- Byte-wise software implementations are normally based on table look-up methods. These algorithms have a large memory and considerable CPU time requirements.
- Hardware implementation is fast, simple and easy to realize.

Sait and Hassan used the hardware description languages to develop and implement this algorithm. The design implemented in small chip. The transmission of the first byte of the CRC can take place in 9^{th} clock pulse and the next byte is transmitted in the 10^{th} clock pulse. Thus, the number of clock cycle required to generate and transmit any CRC in the implementation is two more than the time required to calculate it [7].

On another hand, the authors in [8], designed 10G CRC-32 generation by using VLSI theory and technology. They said the design can be implemented by standard cells in 0.15 micron process technology, or using full custom design techniques in 0.18 micron process technology. The authors mentioned that, a key to achieve high speed is to have flip flop with low delay from clock to output and a high driving capability. The setup time should also be short. When doing a standard cell implementation, some flip flops should be parallelized. The CRC generator has been described in Mentor Graphics and VHDL. Synthesis was performed by Cadence Build gates and timing-driven place and route was made by Cadence Silicon Ensemble (SE). Instead of using

flip flops two stages of latches were used with XOR gates and inverters merged into the latches. The inverter in the latch is used as the last stage of the XOR gate. By doing this they reduce the logic depth and hence the total delay [8].

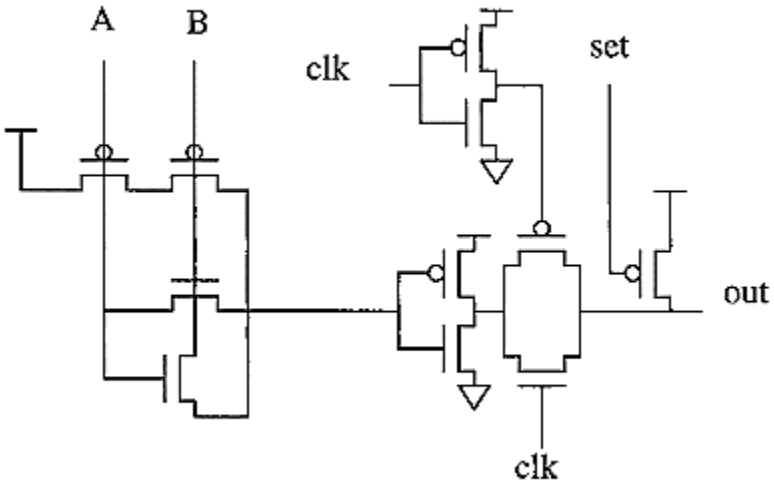


Figure 2.3: XOR merged into positive latch, with active low set signal [8]

The result of this work is 8.7 Gb/s CRC-32 throughput with maximum clock frequency of 1.09 GHz. The authors discussed that a clock frequency of 1.25 GHz is needed but cannot be performed by the technology they were used, so in their recommendation they mentioned that a 0.15 micron process can achieve 10 Gb/s [8].

2.9 FGPA Design architecture of CRC32 using lookup tables

The authors in [9] proposed a reconfigurable architecture of CRC-32. The novelty of this design is the ability to CRC engine to generate checksum

within a single clock cycle, thus the uniqueness of this design is its ability to calculate CRC in the same clock cycle in which the data is loaded. Only the final result is delayed by 1 clock cycle for latching the output result to prevent jitter in the CRC generated [9]. On the other hand, the authors pointed to the confusion that facing the developers about which CRC to use that is due to the numbers of CRCs that already exist. nevertheless, the CRC-32 itself has more than one type – table 2.2 illustrated the types of CRC-32. The system designed to be configurable and support run-time custom modification of polynomial and parameter settings by running a simple C-code remotely in the RISC processor. In the result of this work the variation of data bus width has considered, the test has performed on different FPGA boards and they proved the latency, power consumption and resources are varied based on the bus width modification [9].

The authors in [4], proposed a reduced lookup table algorithm for Ethernet CRC to achieve a throughput of 40 Gbps. The proposed algorithm reduces the memory size requirement to store pre-calculated values to 512 bytes while achieving 40 Gbps, the steps to compute Ethernet CRC for a given input stream of data from pre-computed tables are:

i. Let the pre-computed tables be LUT4, LUT3, LUT2 and LUT1 each containing 32 entries of 32 bytes each.
ii. Let t4, t3, t2, t1 be 32 bits registers.
iii. Initialize the 32 bits CRC register to 0 hex.
iv. From the 16 byte input b(x), XOR(b127…b96) with inverted previous CRC value, ~(s31… s0) to form (t431...t40); the remaining input bits(b95 ... b0) being t3, t2, t1.
v. For i= 31, …1, 0 and j = 4, 3, 2, 1 if $t_j[i]$= 1, look up the corresponding 32-bit value from $lut_j[i]$ and XOR the CRC register with it.

vi. Repeat steps iv to v until you reach the end of the message.

vii. Invert contents of CRC register to result in the CRC32 value corresponding to the input b(x).

Table 2.3: CRC-32 Types

Name	Width	Polynomial	Initialize	Input Reflection	Output Reflection	Xor out
CRC-32/Ethernet	32	0x04C11DB7	0xFFFFFFFF	True	True	0xFFFFFFFF
CRC-32/MPEG-2	32	0x04C11DB7	0xFFFFFFFF	False	False	0x00000000
CRC-32/BZIP2	32	0x04C11DB7	0xFFFFFFFF	False	False	0xFFFFFFFF
CRC-32C	32	0x1EDC6F41	0xFFFFFFFF	True	True	0xFFFFFFFF
CRC-32D	32	0xA833982B	0xFFFFFFFF	True	True	0xFFFFFFFF
CRC-32/POSIX	32	0x04C11DB7	0x00000000	False	False	0xFFFFFFFF
CRC-32Q	32	0x814141AB	0x00000000	False	False	0x00000000
JAMCRC	32	0x04C11DB7	0xFFFFFFFF	True	True	0x00000000
XFER	32	0x000000AF	0x00000000	False	False	0x00000000

The number of XOR operations corresponds to the number of nonzero bits in (t4, t3, t2, t1). The CRC module is designed with Verilog modeling with the signals as shown in figure 2.4, The signals required for the engine are described in table 2.3.

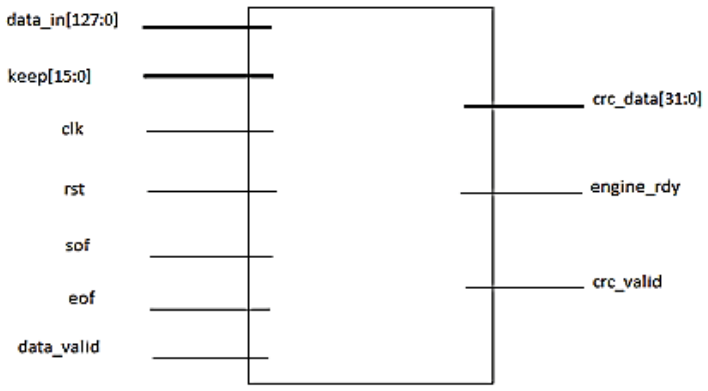


Figure 2.4: CRC module design

Table 2.4: CRC ENGINE SIGNAL DESCRIPTION

data_in (128 bits)	input	The data content in the Ethernet packet formed from destination and source MAC address and the data payload. It is processed 128 bits at a time.
clk	input	Signal of frequency 312.5Mhz based on which complete digital circuit is synchronized.
rst	input	This signal when it goes high resets the CRC Engine.
keep (16 bits)	input	This signal indicates which all bytes in the 16 Bytes (16 x 8 =128bits) data is valid. To keep "n" bytes from LSB, where n = 1, 2, .., 16 the encoding of signal is as follows: 16'hFFFF → keep 16bytes 16'h7FFF → keep 15 bytes 16'h3FFF → keep 14 bytes . . . 16'h00FF→ keep 8 bytes 16'h007F → keep 7 bytes 16'h003F→ keep 6 bytes 16'h001F→ keep 5 bytes 16'h000F→ keep 4 bytes 16'h0007→ keep 3 bytes 16'h0003→ keep 2 bytes 16'h0001→ keep 1 byte
data_valid	input	This indicates whether the current data_in available is valid or not.
sof	input	This is start of frame signal, which is a high pulse for one clock period.
eof	input	This is end of frame signal, which is a high pulse for one clock period.

crc_data (32 bits)	output	This is 32 bits CRC32 value computed for data_in between sof and eof
engine_rdy	output	This is a high signal when the crc engine is not computing anything and is ready to receive a start of frame and valid data_in to start computation.
crc_valid	output	This is a high pulse of one clock period when the CRC32 value is valid on the crc_data signal.

The CRC32 IP was written in Verilog and simulated in ModelSim SE PLUS 6.3g. The synthesis was done with Xilinx Synthesis Technology (XST) and implemented for XC6VHX380T-FF1923-3 FPGA. The functionality of the design is verified by varying the data packet content and message length. The simulation results show message data coming in data_in which starts with sof pulse and ends with eof pulse. This data between sof and eof is used for CRC computation resulting in crc_data, validity of which is asserted by the crc_valid pulse. From the simulation waveform, it is seen that for the data stream:

"0x31323334353637383132333435363738_35363738313233343132333435363738_35363738313233343132333435363738"

given between control signals sof and eof (asserted high for one clock period), the computed CRC is "0x81D0BB9C". At this instant, the crc_valid signal goes high indicating valid CRC. Also the engine_rdy signal goes high indicating that CRC engine is ready to process next set of data. With respect to the synthesis report, it can be seen that the proposed algorithm gives the desired results as per the speed requirement of achieving 40 Gbps throughput [4].

The author [10] focus on efficient method for polynomial division and CRC code implementation. Instead of working on one bit of the dividend at time, this method uses table look-ups to handle eight bits (one byte) at time. A detailed description of the computation of CRC bits using a CRC polynomial of degree 16 is given. Depending on the capabilities of the processor, either two tables of 256 8-bit bytes (each accessed once for each byte of the dividend) or one table of 256 16-bit words (accessed once for each byte of the dividend) can be used. With appropriate changes, the same technique can be applied to the computation of CRCs using CRC polynomials of degree 32 in which case four tables of 256 8-bit bytes are needed. These tables can also be organized as two tables of 256 Is-bit words or as one table of 256 32-bit double words. In all cases, each table is accessed once for each byte of the dividend [10]. The program fragment that follows computes two CRC bytes C_1 and C_0 from m data bytes stored in an array *A*. Two arrays (tables) f_1 and f_0 of 256 bytes each are used. The byte *T* is used to compute the relative address of the table elements to be looked up at each step. Note that the elements of the array *A* are not modified at all.

```
C1 = 0;
C0 = 0;
for (i = m-1; i != 0; i--)
(
     T = C1 xor A[i];
     C1 = C0 xor f1[T];
     C0 = f0 [T];
)
```

The program fragment can be used as part of a procedure whose arguments include the array *A* and its size *m*, and which returns $C_1$and C_0 as the value

computed by the procedure. In fact, if the procedure arguments include the arrays f_0 and f_1 then the same procedure can be used to compute CRC bytes for different CRC polynomials of degree 16. If m = n and the array A contains the transmitted data byte D_{n-1}, D_{n-2}, D_{n-3}, . . ., D_2, D_1, D_0, then $C_1 = B_1$ and $C_0 = B_0$ are the CRC bytes to be transmitted. If m = n + 2 and the array A contains the received bytes R_{n+1}, R_n, R_{n-1}, …, R_2, R_1, R_0, then the received data bytes R_{n+1}, R_n, R_{n-1}, …, R_2 are accepted as error free if and only if C_1 and C_0 are zero. Thus, the same procedure can be used at the transmitter and the receiver provided that the results returned by the procedure are interpreted appropriately [10]. In the program fragment just described, both table lookup operations use the same index T. Since many processors can fetch 16-bit operands in one memory cycle, it may be advantageous to combine the two byte tables f_1 and f_0 into a word table f containing 256 16-bit words. The high order byte of the table entry f [T] (the T-th word in the table f, where T is interpreted as an integer in the range 0 to 255) is f1[T], and the low-order byte is f0[T], so that a single table lookup of a 16-bit word fetches both f1 [T] and f0[T]. One other speed-up relies on the fact that many processors can Exclusive OR the contents of a 16-bit word in memory into a 16-bit register or a 16-bit word. Thus, a single instruction can be used to change both C_1 and C_0. The program fragment shown previously is easily modified to use a single table. In the modified version that follows, it is assumed that C_1 and C_0 are respectively the high-order and low-order bytes of a 16-bit register or word denoted by C.

```
C1 = 0;
for (i = m-1; i != 0; i--)
(
     T = C1 xor A[i];
     C1 = C0;
     C0 = 0;
     C = C xor f[T];
)
```

Both program fragments process the data bytes in the array A one at a time, and can be readily modified to process the data on-line, for example, by replacing the loop controlled by the *for* statement with a loop controlled by a *while* statement which reads and processes bytes until an end-of-file is encountered. Algorithms for the computation of 32 bit CRCs are very similar to the ones shown above.

CHAPTER THREE

DESIGN AND SIMULATION

3.1 System overview

In this project the computation of CRC32 performed by lookup table method. In this method the possible values of the CRC32 pre-computed and saved in lookup table. This pre-computed values called during the computation of the final CRC32 rather than compute it every time. Each byte has its own CRC32 value that will be added as an argument in logical operations that can find the final result. The efficiency of the system will be increased if more than one lookup tables work simultaneously which is the method that used in this project this method known as slicing by N. The System simulated based on FPGA and the related code written by VHDL.

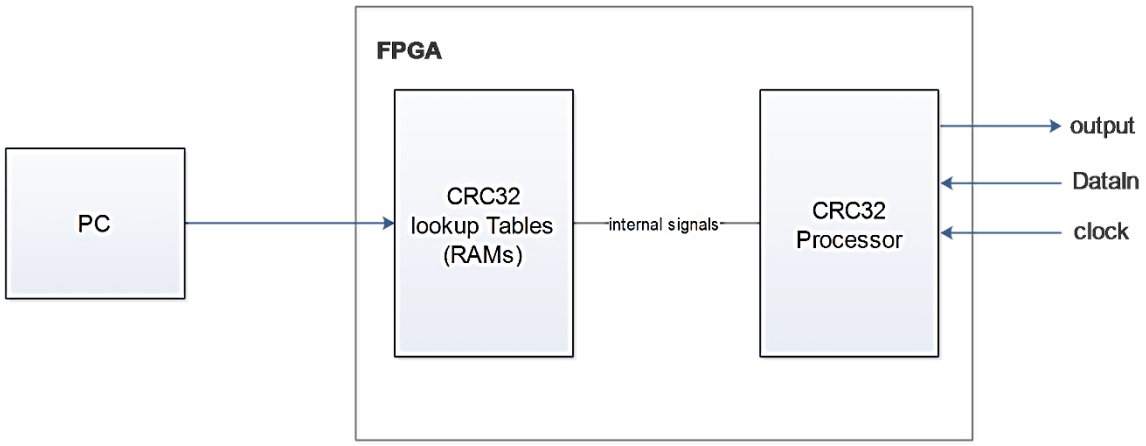


Figure 3.1: System Overview

Figure 3.1 show that a PC connected with FPGA, the role of this PC in the design is the generation of the pre-computed values of the lookup tables through Java program. After this values generated it will be transferred to the FPGA and saved inside the RAMs. This PC can be considered a micro-processor located outside the FPGA or inside it in the real implementation, but for simulation purposes the CRC values generated through software written in Java.

3.2 Generation of lookup tables

As mentioned earlier, lookup tables methodology can reduce the number of needed operations to find out the CRC32 for a stream of bytes. as long as the divisor (polynomial) is always fixed we can pre-compute the division for each possible byte and save the reminder in lookup table. This can be performed through the below code

```
for (int dividend = 0; dividend <= 255; dividend ++)
{
   int temp =  dividend; // temp loop over all possible values
   for (int bits=0; bits<8; bits++)// generate the CRC
          temp = (temp >>1)^((temp &1)*0x04C11DB7);
   crcTable[dividend] = temp; // save the CRC value in array
}
```

In this code, the dividend which include the possible values for the bytes (0x00 to 0xFF) saved in the temp variable in each iteration. Then the temp variable

processed in term of bits inside the second for loop. the equation behind the second for loop checkup the LSB of the temp, if its ‘1’ then the shifted right value of the temp will perform xor with the polynomial, this operation continues for the eight bits to find out the reminder of (dividend mod polynomial). This reminder saved in the lookup table (Array) that indexed by the dividend. Figure 3.2 show the flow chart of lookup table generation.

3.3 Slicing by 16 lookup tables

Section 3.2 illustrate how and why the lookup table generated. But as long as the design going to find out the CRC32 for 16 bytes (128 bit) in one operation that is mean 15 additional lookup tables should be generated as well. The generation of these tables depends on the first table that generated in section 3.2, the below code responsible for this issue.

```
for (int i=0; i<=255; i++)
{
crcTable1[i] = (crcTable0[i]>>8)^crcTable0[(int)(crcTable0[i] & 0xFF)];
crcTable2[i] = (crcTable1[i]>>8)^crcTable0[(int)(crcTable1[i] & 0xFF)];
crcTable3[i] = (crcTable2[i]>>8)^crcTable0[(int)(crcTable2[i] & 0xFF)];
crcTable4[i] = (crcTable3[i]>>8)^crcTable0[(int)(crcTable3[i] & 0xFF)];
crcTable5[i] = (crcTable4[i]>>8)^crcTable0[(int)(crcTable4[i] & 0xFF)];
crcTable6[i] = (crcTable5[i]>>8)^crcTable0[(int)(crcTable5[i] & 0xFF)];
crcTable7[i] = (crcTable6[i]>>8)^crcTable0[(int)(crcTable6[i] & 0xFF)];
crcTable8[i] = (crcTable7[i]>>8)^crcTable0[(int)(crcTable7[i] & 0xFF)];
crcTable9[i] = (crcTable8[i]>>8)^crcTable0[(int)(crcTable8[i] & 0xFF)];
crcTable10[i]= (crcTable9[i]>>8)^crcTable0[(int)(crcTable9[i] & 0xFF)];
crcTable11[i]=(crcTable10[i]>>8)^crcTable0[(int)(crcTable10[i] & 0xFF)];
crcTable12[i]=(crcTable11[i]>>8)^crcTable0[(int)(crcTable11[i] & 0xFF)];
crcTable13[i]=(crcTable12[i]>>8)^crcTable0[(int)(crcTable12[i] & 0xFF)];
crcTable14[i]=(crcTable13[i]>>8)^crcTable0[(int)(crcTable13[i] & 0xFF)];
crcTable15[i]=(crcTable14[i]>>8)^crcTable0[(int)(crcTable14[i]& 0xFF)];}
```

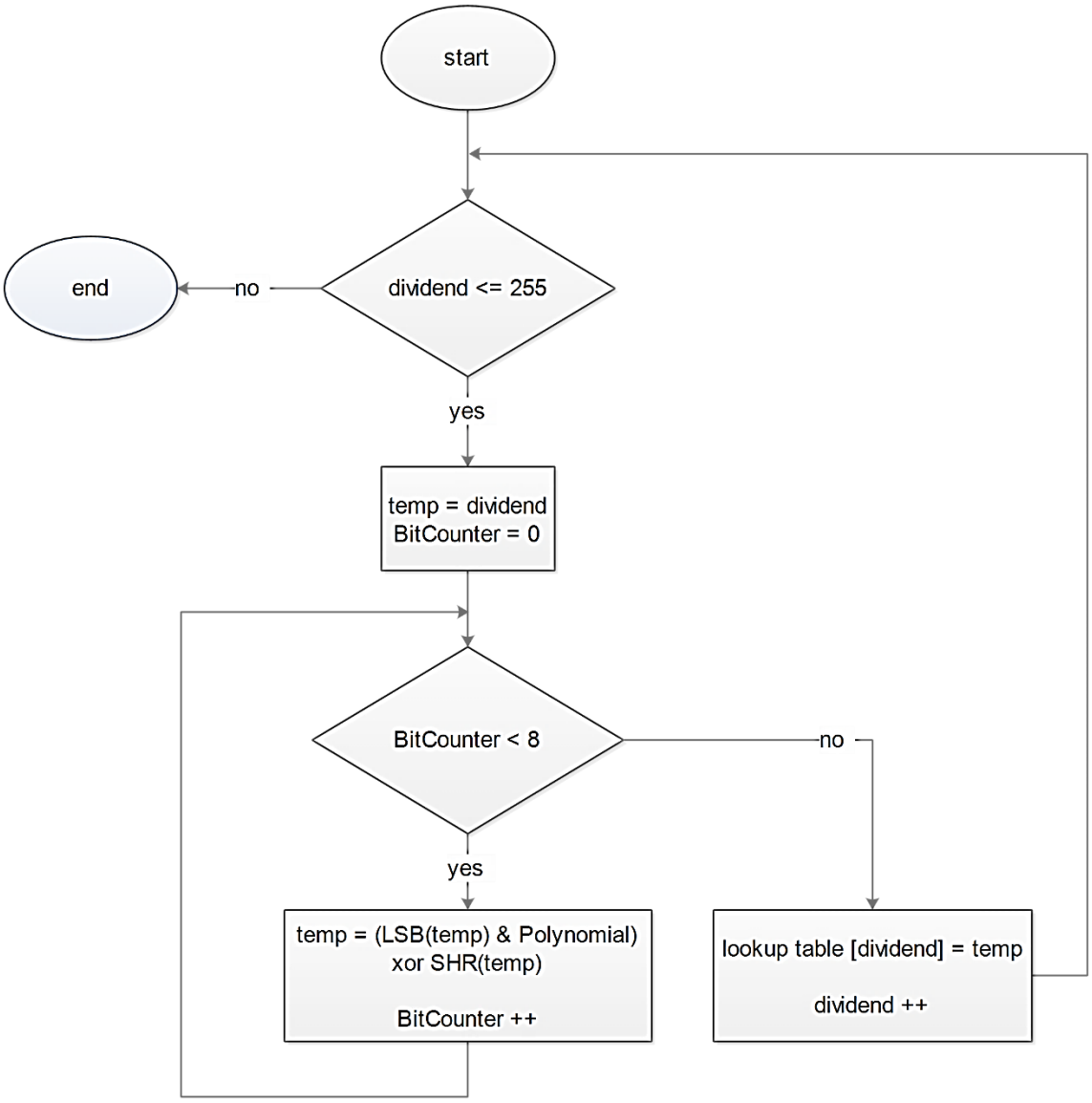


Figure 3.2: lookup table generation

In the previous code, the value for each byte on each lookup table depends on the preceding lookup table indexed with the same index. This preceding value shifted to the right by 8 bit and then xor with lookup table0 that indexed by the LSB byte from the preceding lookup table.

3.4 Ethernet CRC32 Polynomial Standard

Although the algorithm can support any 32-bit value of the polynomials, the design focus on the standard polynomial for CRC32 which already used in the Ethernet which is 0x04C11DB7. This value considered in the generation of the lookup tables and in the calculation of the final CRC32 value.

3.5 The Platform

The FPGA selected to be the platform to implement the Slicing by 16 CRC32 algorithm. FPGAs can work in parallel when compared with digital signal processors. In this case the algorithm needs to access 16 locations in different arrays or in different RAMs then after the data fetched from the memory xor dependent operation should be performed. Digital signal processors will need individual operations related to each line of the code a lot of pulses will be consumed to fetch the operation and to add the argument to it then to perform the operation itself. On the other hand, the FPGA can perform the calculation in parallel, it can access the 16 locations simultaneously and performing the operation in the same time thus saving considerable time and increase the throughput and the efficient of the system.

3.6 Design of Lookup tables (RAMs)

Lookup tables keeps the pre-calculated values that will be used in the computation of the CRC32, considering that during the operation of the CRC32 computation the processor will need to access the lookup tables for each byte of the input stream. In the high level language or in the object oriented programming the lookup tables could be saved in arrays. In this

design the implementation of the algorithm performed on the hardware rather than the software, accordingly hard RAMs will be needed. Figure 3.3 illustrate the timing waveform to read a data from external RAM, in this waveform a considerable time between RAM addressing and the validity of data is noted. This transfer time could be reduced if the RAMs created inside the FPGA.

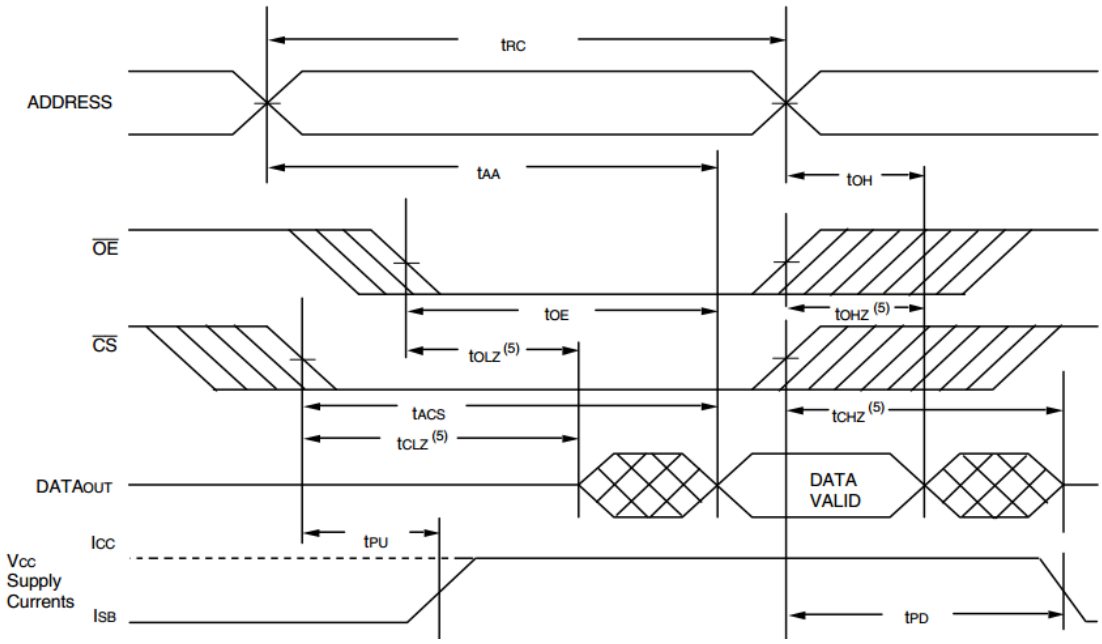


Figure 3.3: Timing waveform of Read cycle

3.7 CRC32 using Slicing by 16 – system architecture

The algorithm read 128 bit, the first 96 bit (from 127 to 32) sliced into 12 slices each with 8 bit. The value of each part will indexing the lookup tables starting from lookup table 0 till lookup table 11. The second group of input bits (from 31 to 0) should be xor with the preceding CRC32 output or with

initial value. After that, the output of the 16 lookup tables the system will xor them to find out the CRC32. This operation continues to the last 128 bit of the Ethernet frame. Figure 3.4 illustrate how slicing by 16 work to find the CRC32.

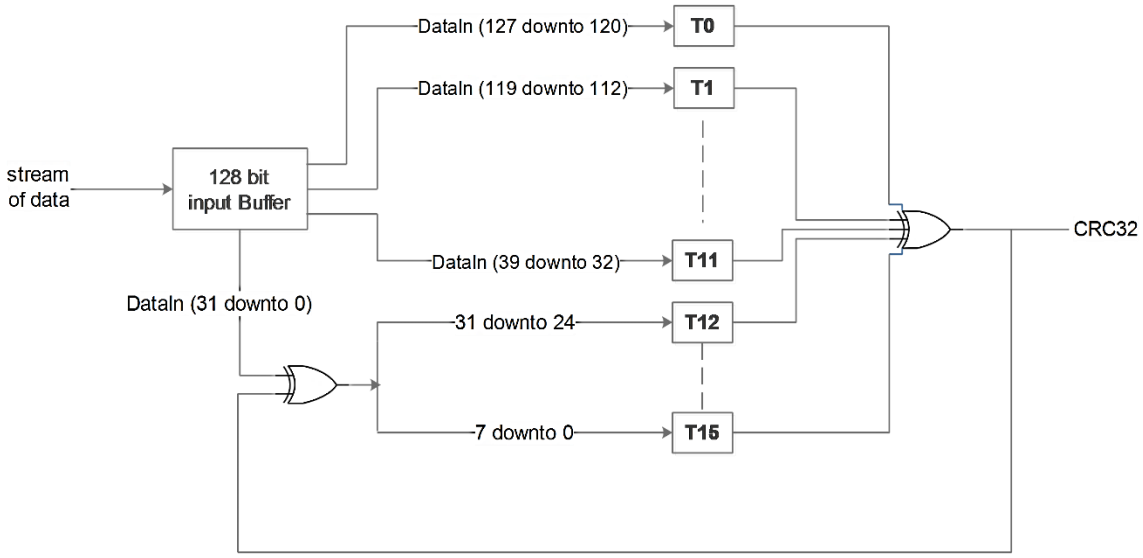


Figure 3.4: Block diagram of CRC32 using slicing by 16 algorithm

3.8 Implementation of lookup tables RAMs

As discussed in the design, the implementation of lookup table RAMs will be performed inside the FPGA that is in order to reduce the design area, decrease the power consumption and to increase the transferring speed between the FPGA and the RAMs. The sixteen RAMs created by VHDL code. New type called lookUpTable defined as array consist of 256 locations, each location has 28 bit. Then sixteen signal of this type has defined. During the initialization these memory locations loaded with the precomputed values. The below VHDL code responsible for creating the RAMs.

```
type lookUpTable is array (0 to 255) of std_logic_vector(27 downto 0);

signal crcTable0,crcTable1, crcTable2, crcTable3, crcTable4,crcTable5,crcTable6,
crcTable7,crcTable8, crcTable9,crcTable10, crcTable11, crcTable12, crcTable13,
crcTable14, crcTable15 : lookUpTable;
```

3.9 VHDL Design Entity

In VHDL the entity is the description of the ports of the circuit. In this project the entity consists of the following ports:

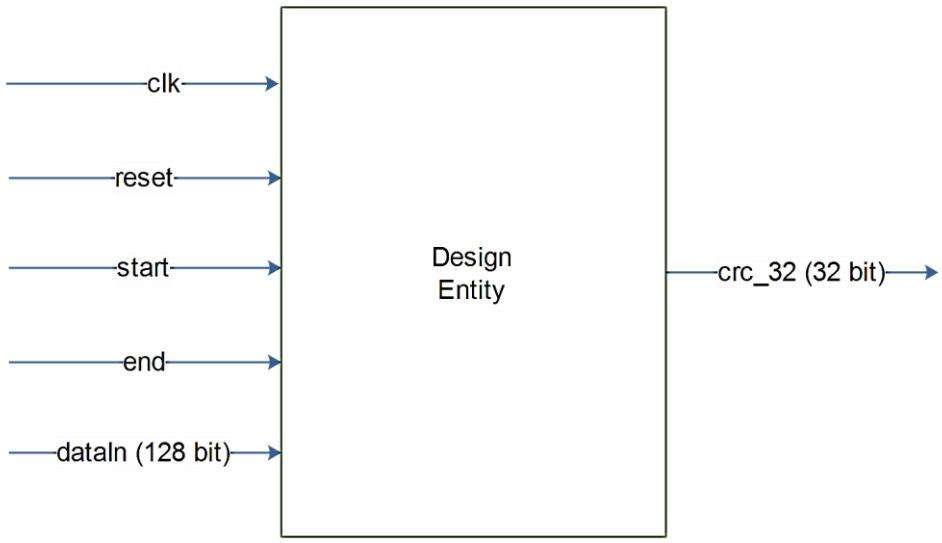


Figure 3.5: System design entity

- Clk: 1 bit input port, through it the circuit can receive its clock.
- Reset: 1 bit input port used to reset the circuit and the CRC32 register.
- Start: 1 bit input port, used to trigger the circuit of CRC32 computation.

- END: 1 bit input port, indicate the Ethernet frame reached the last bit and accordingly the computation of the CRC32 can be closed.
- DataIn: 128 bit port, should be connected to internal buffer. This port includes the data that needs CRC32 calculation.
- Crc_32: 32 bit output port, hold the result of CRC32 computation.

3.10 VHDL implementation of CRC32 using the Slicing by 16

As illustrated in figure 3.5 the slicing by 16 algorithm needs to access 16 RAMs simultaneously to fetch the values of CRC32 for each 8 bit of the sliced DataIn which contains 128 bit. The implementation of the code in the VHDL performed concurrently, that is will support to achieve the operation of this algorithm in parallel. The sliced signals of the DataIn used to index the RAM, however in order to perform this the data of the type std_logic_vector of the DataIn should be converted into integer, this operation could be performed by the function that called conv_integer which is belongs to the package IEEE.STD_LOGIC_ARITH. In the code four zeros concatenated to the fetched data that is because the values of the CRC32 that saved in the RAMs consist of 28 bit and the design needs to keep the width of the registers to be 32 bit, this procedure reduces the design warnings. As illustrated in the next VHDL code, xor operation performed for these data to get the CRC32.

```
crc32Signal <=  x"0" & crcTable0(conv_integer(unsigned(DataIn(127 downto 120)))) xor
                x"0" & crcTable1(conv_integer(unsigned(DataIn(119 downto 112)))) xor
                x"0" & crcTable2(conv_integer(unsigned(DataIn(111 downto 104)))) xor
                x"0" & crcTable3(conv_integer(unsigned(DataIn(103 downto 96))))  xor

                x"0" & crcTable4(conv_integer(unsigned(DataIn(95 downto 88)))) xor
                x"0" & crcTable5(conv_integer(unsigned(DataIn(87 downto 80)))) xor
                x"0" & crcTable6(conv_integer(unsigned(DataIn(79 downto 72)))) xor
                x"0" & crcTable7(conv_integer(unsigned(DataIn(71 downto 64)))) xor

                x"0" & crcTable8(conv_integer(unsigned(DataIn(63 downto 56))))  xor
                x"0" & crcTable9(conv_integer(unsigned(DataIn(55 downto 48))))  xor
                x"0" & crcTable10(conv_integer(unsigned(DataIn(47 downto 40)))) xor
                x"0" & crcTable11(conv_integer(unsigned(DataIn(39 downto 32)))) xor

                x"0" & crcTable12(conv_integer(unsigned(PrevCrcXorDin(31 downto 24)))) xor
                x"0" & crcTable13(conv_integer(unsigned(PrevCrcXorDin(23 downto 16)))) xor
                x"0" & crcTable14(conv_integer(unsigned(PrevCrcXorDin(15 downto 8)))) xor
                x"0" & crcTable15(conv_integer(unsigned(PrevCrcXorDin(7 downto 0))));
```

slicing the least significant 32 bit. This was discussed in the design and figure 3.5, however, the below line of code illustrate how this operation performed.

```
PrevCrcXorDin <= crc32Signal xor DataIn (31 downto 0);
```

3.11 Software implementation of the lookup values generator

Actually, the implementation of the generator of the lookup values could be performed inside a co-processor working behind the FPGA. This processor responsible for the generation of these values and then transfer it to the RAMs on the FPGA during the initialization phase.

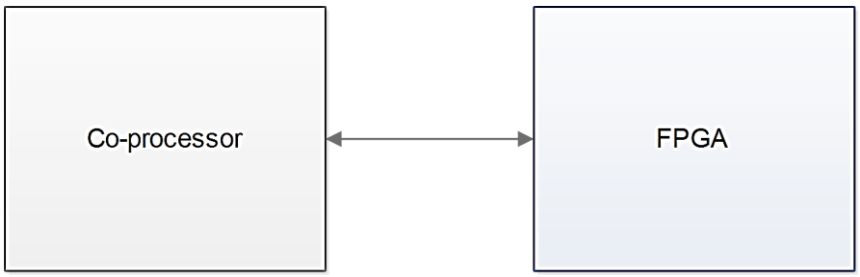


Figure 3.6: Co-processor for the generation of CRC32 lookup tables values

For simulation purposes, the generation of the value performed by java application, these values directly assigned to the internal RAMs on the FPGA.

3.12 Simulation and the simulation stimulus

Isim simulator version 14.5 used to find out the result of the design, Isim uses the VHDL to write the simulation code. However, Isim create a component for the unit under test. The component has the same ports like the design entity in the main design. furthermore, Isim automatically generate a signals corresponds to the entity ports, these signals used as a stimulus for the unit under test. Regarding the clock, the Isim automatically also generate constant time identify the period of the pulses. The constant time by default generated to be 10 ns but based on the design requirements the time modified to 1.25 ns which provide a frequency equal to 800 MHz. the simulation has two process working concurrently, the first process generates the clock pulses and the second process responsible for the simulation stimulus. Inside the process of the stimulus 128 bit randomly assigned to the DataIn signal in a period specified in simulation code, behind that the control signals such as start, end and reset also considered in the simulation code. Figure 3.7 illustrate how the control signals and the dataIn stimulated in the simulation.

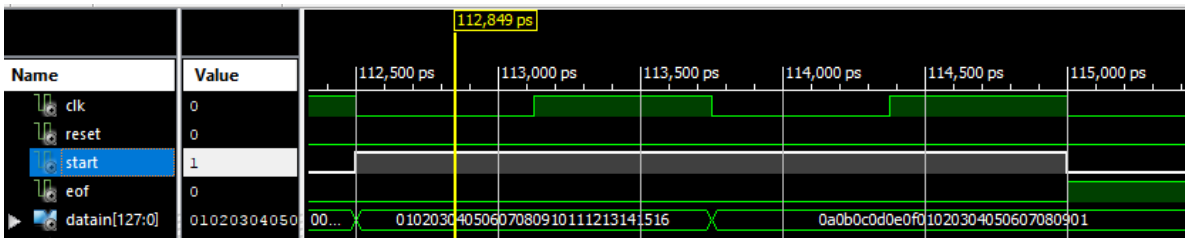


Figure 3.7: simulation of the signals and dataIn

CHAPTER FOUR

RESULT DISCUSSION

4.1 Design Throughput

The simulation of this design performed based on virtex-7 FPGA with 800 MHz oscillator , this can lead to process 128-bit of the data in 1.250 ns. That is mean the throughput of the CRC32 can reach up to 102.4 Gbps which is the targeted result, this throughput is higher than the throughputs that mentioned in the literature review. Regarding the analysis of the operations that performed in the time domain of the clock pulses, in figure 4.1 the highlighted combinational circuit responsible for performing xor operation between the previous result and the dataIn least significant bits. This circuit performed outside the process and independent of clock pulses. During the falling edge of the clock pulses the buffer response to send the data to the entity in the dataIn port. The time between the falling edge and the rising edge utilized to find out the output of the highlighted circuit. Nevertheless, to find out the final result the none-highlighted circuit needs to work, this circuit restricted to work only during the rising edge of the clock pulses. This procedure can guarantee that, the presence of the output of the highlighted circuit was ready. Accordingly, the none-highlighted circuit added to process with the related sensitivity list in VHDL code in order to perform its work during the high pulse. Figure 4.2 illustrate the relationship between the dataIn, the output of the CRC32 and the related pulse.

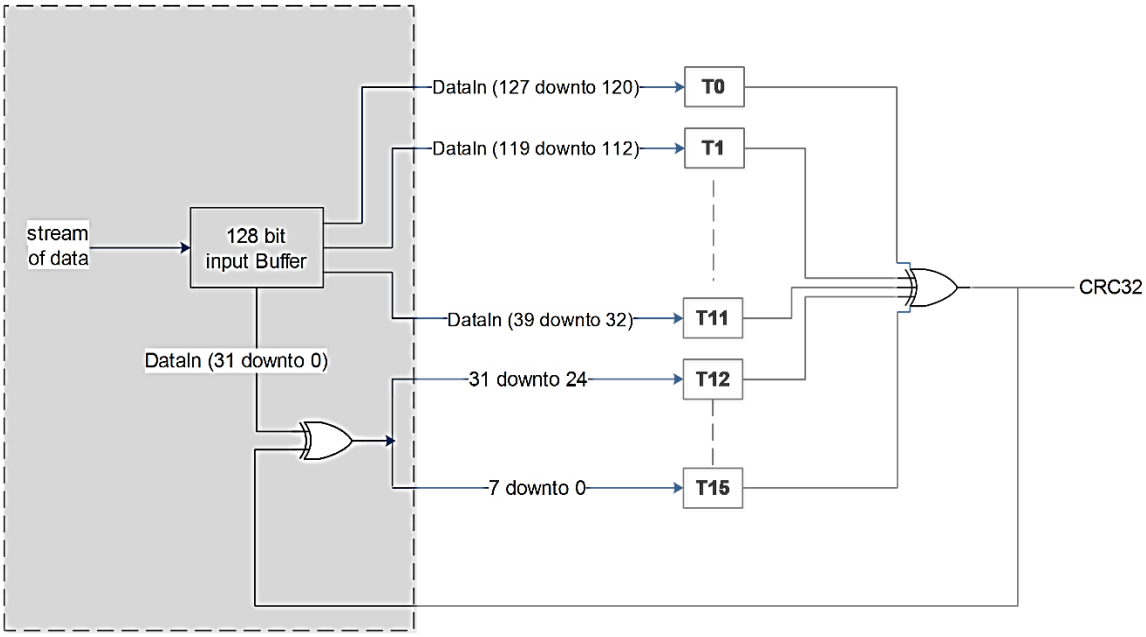


Figure 4.1: one clock to perform the processing

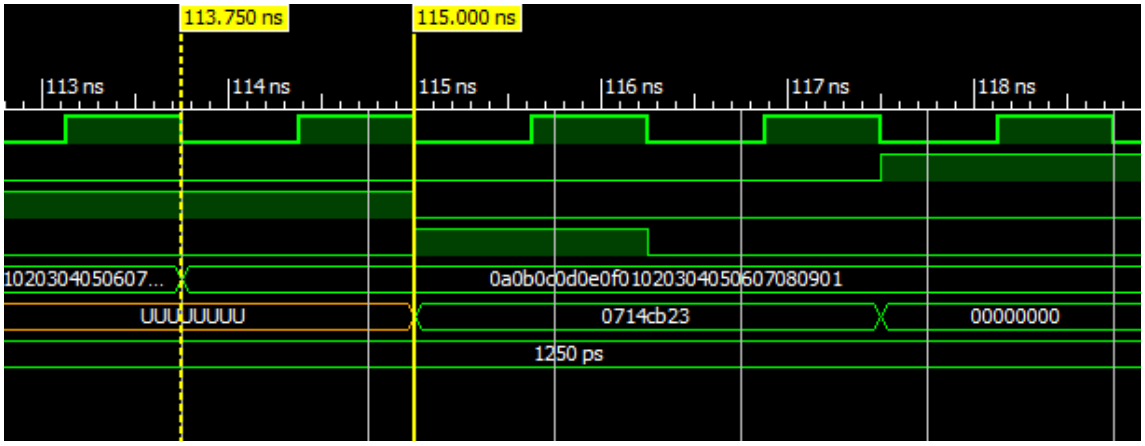


Figure 4.2: the relation between the dataIn and the CRC32 in term of needed number of clocks

4.2 FPGA Resource Utilization

Behind that, Figure 4.3 illustrate the utilized resources from the FPGA which could be considered very low and can allow to the system designer to merge it inside any system.

Table 4.1: utilization of FPGA resources

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	27	408000	0%
Number of Slice LUTs	550	204000	0%
Number of fully used LUT-FF pairs	27	550	4%
Number of bonded IOBs	164	600	27%
Number of BUFG/BUFGCTRL/BUFHCEs	2	200	1%

The utilization of the device compared with others work that illustrated in table 4.1, the comparison shows the number of Slices LUTs that used in this design is less than the number of Slices that performed in [4].

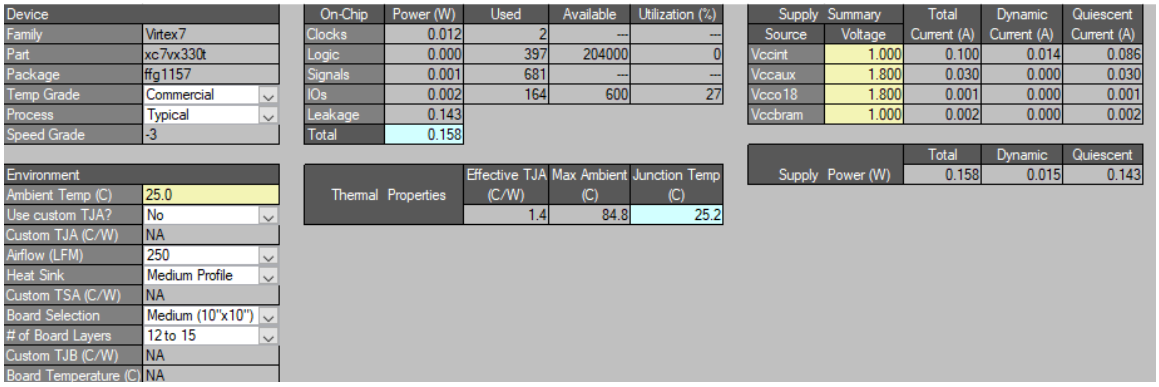
Table 4.2: device utilization in work [4]

Slice Register	Slice LUT	Logic
75	10164	10164

4.3 FPGA Power Consumption

On another hand, the total of the consumed power is 0.158W, the dynamic power consumption is only 0.015W. so most of the power consumption generated from the leakage or the quiescent which is around

0.143W. accordingly the power consumption is low and could be considered very low when compared to the works in the literature review.



Device	
Family	Virtex7
Part	xc7vx330t
Package	ffg1157
Temp Grade	Commercial
Process	Typical
Speed Grade	-3

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.012	2	---	---
Logic	0.000	397	204000	0
Signals	0.001	681	---	---
IOs	0.002	164	600	27
Leakage	0.143			
Total	0.158			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.100	0.014	0.086
Vccaux	1.800	0.030	0.000	0.030
Vcco18	1.800	0.001	0.000	0.001
Vccbram	1.000	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.158	0.015	0.143

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA
Board Temperature (C)	NA

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	1.4	84.8	25.2

Figure 4.3: System power consumption

CHAPTER FIVE
CONCOLUSION AND RECOMMENDATIONS

5.1 Conclusion

The demand on the data communication became very high, next year applications, the internet of things and machine to machine communications can affect the trend of the data consumption either from the user point of view or beside the companies, enterprises and data centers. This change affect directly on the networking devices in term of switching capability, processing, and finally the computation of the CRC32 for the error detection and correction which is main point that discussed in this project. The project aims to improve the processing capability of the CRC32 in order to meet the new requirements and speed. The targeted throughput is 100 Gbps and in order to achieve it the design built based on the FPGA. The FPGA selected due its concurrent behavior, it can perform more than one operation concurrently. Behind that, more than one algorithm has studied and the slicing by 16 algorithm selected, this algorithm reduces the computation of the CRC32 by creating lookup tables hold all the possible values of the CRC32. The code of the design and the simulation written by VHDL. The simulation test show that, the achieved throughput is equal to 102.4 Gbps.

5.2 Recommendations

One of the vital areas of the design is the buffering system, which is the system that feeds the design entity and provides the data that will be

processed. This research focused on the processing area more than the buffering system, so furthers research on this field will be needed. Behind that the trend of the data exchange will be increased, accordingly increasing the throughput of the CRC will be required as well.

References

[1] L. Atzori, A. Ieran, and G. Morabito, "The Internet of Things: A survey," *Computer Networks,* vol. 54, pp. 2787-2805, 2010.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE INTERNET OF THINGS JOURNAL,* vol. 1, pp. 22-32, 2014.

[3] Cisco, "The Zettabyte Era: Trends and Analysis," 2016.

[4] P. A. Anand and D. Bajarangbali, "Design of High Speed CRC Algorithm for Ethernet on FPGA Using Reduced Lookup Table Algorithm," presented at the IEEE Annual India Conference (INDICON), India 2016.

[5] R. N. Williams, *A PAINLESS GUIDE TO CRC ERROR DETECTION ALGORITHMS*, 1993.

[6] A. Perez, Wismer, and Becker, "Byte-wise CRC Calculation," *IEEE Micro,* 1983.

[7] S. M. Sait and W. Hasan, "Hardware Design and VLSI Implementation of Byte-Wise CRC Generator Chip," *IEEE Transaction on Consumer Electronics,* vol. 41, pp. 195-200, 1995.

[8] T. Henriksson, H. Eriksson, U. Nordqvist, P. Larsson-Edefors, and D. Liu, "VLSI implementation of CRC-32 for 10 Gigabit Ethernet," in *ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.01EX483)*, 2001, pp. 1215-1218 vol.3.

[9] J. Mitra and T. K. Nayak, "Reconfigurable Concurrent VLSI (FPGA) Design Architecture of CRC-32 for High-Speed Data Communication," in *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, 2015, pp. 112-117.

[10] E. H. Sibley and P. Editor, "Computation of Cyclic Redundancy Checks Via Table Look-up," *communications of the ACM,* vol. 32, pp. 1008-1013, 1988.

[11] R. Nair, G. Ryan, and F. Farzaneh, "A symbol based algorithm for hardware implementation of cyclic redundancy check (CRC)," in *VHDL International Users' Forum, 1997. Proceedings*, 1997, pp. 82-87.

[12] xilinx, "7 Series FPGAs Overview," 2012.

[13] *K*oopman, Ph*il*ip Chakravarty and Tridib*,* "Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks" *in 2004 The International Conference on Dependable Systems and Networks*

[14] *B*rayer, Kenne*th* "Evaluation of 32 Degree Polynomials in Error Detection on the SATIN IV Autovon Error Patterns" *in 2011 National Technical Information Service*

[15] Hammond, Joseph L., Jr. Brown, James E. Liu, Shyan-Shiang "Development of a Transmission Error Model and an Error Control Model" in 1975 *National Technical Information Service*

[16] Brayer, Kenneth; Hammond, Joseph L., Jr. "Evaluation of error detection polynomial performance on the AUTOVON channel" in 1975 *IEEE National Telecommunications Conference*

[17] Cook, Greg "Catalogue of parametrised CRC algorithms" 2016 .

APPENDIX-A: VHDL CODE

```
1 ----------------------------------------------------------------------
-------------
2 -- Company: SUST_MSC
3 -- Engineer: Mohamed Salah
4 --
5 -- Create Date: 17:55:06 11/10/2017
6 -- Design Name: CRC32 - Slicing by 16
7 -- Module Name: crc32_slicing16 - Behavioral
8 -- Project Name: crc32 implementation using FPGA
15 -- Revision:
16 -- Revision 0.01 - File Created

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23
32
33 entity crc32_slicing16 is
34 Port ( clk : in STD_LOGIC;
35 reset : in STD_LOGIC;
36 start : in STD_LOGIC;
37 EOF : in STD_LOGIC;
38 DataIn : in STD_LOGIC_VECTOR (127 downto 0);
39 crc_32 : out STD_LOGIC_VECTOR (31 downto 0));
40 end crc32_slicing16;
41
42 architecture Behavioral of crc32_slicing16 is
43
44 type lookUpTable is array (0 to 255) of std_logic_vector(27 downto 0);


45 signal crcTable0,crcTable1, crcTable2, crcTable3,
crcTable4,crcTable5,crcTable6,
crcTable7, crcTable8, crcTable9,


46 crcTable10, crcTable11, crcTable12, crcTable13, crcTable14, crcTable15:
lookUpTable;
47
48 signal crc32Signal : std_logic_vector(31 downto 0) := (others => '0');


49 signal PrevCrcXorDin : std_logic_vector(31 downto 0) := (others => '0');
50
51 begin
52
53 PrevCrcXorDin <= crc32Signal xor DataIn(31 downto 0);
54
55
56 process(clk,start,reset,EOF,crc32signal)

57
58 begin
59
60 if reset = '1' then
```

```
61 crc_32 <= (others => '0');
62
63
64 elsif rising_edge(clk) and EOF /= '1' then
65
66 if start = '1' then
67
68 crc32Signal <= x"0" & crcTable0(conv_integer(unsigned(DataIn(127
downto 120)))) xor


69 x"0" & crcTable1(conv_integer(unsigned(DataIn(119
downto 112)))) xor


70 x"0" & crcTable2(conv_integer(unsigned(DataIn(111
downto 104)))) xor


71 x"0" & crcTable3(conv_integer(unsigned(DataIn(103
downto 96)))) xor
72
73 x"0" & crcTable4(conv_integer(unsigned(DataIn(95 downto
88)))) xor
74 x"0" & crcTable5(conv_integer(unsigned(DataIn(87 downto
80)))) xor
75 x"0" & crcTable6(conv_integer(unsigned(DataIn(79 downto
72)))) xor
76 x"0" & crcTable7(conv_integer(unsigned(DataIn(71 downto
64)))) xor
77
78 x"0" & crcTable8(conv_integer(unsigned(DataIn(63 downto
56)))) xor
79 x"0" & crcTable9(conv_integer(unsigned(DataIn(55 downto
48)))) xor
80 x"0" & crcTable10(conv_integer(unsigned(DataIn(47
downto 40)))) xor
81 x"0" & crcTable11(conv_integer(unsigned(DataIn(39
downto 32)))) xor
82
83 x"0" & crcTable12(conv_integer(unsigned(PrevCrcXorDin(
31 downto 24)))) xor
84 x"0" & crcTable13(conv_integer(unsigned(PrevCrcXorDin(
23 downto 16)))) xor
85 x"0" & crcTable14(conv_integer(unsigned(PrevCrcXorDin(
15 downto 8)))) xor
86 x"0" & crcTable15(conv_integer(unsigned(PrevCrcXorDin(7
downto 0))));
87
88
89 end if;
90 elsif rising_edge (EOF) then
91
92 crc_32 <= crc32Signal;
93
94 end if;
95 end process;
96
97 -- CRC32 Slicing by 16 Lookup Tables (16x RAM)
```

```
crcTable0 <=
(
 x"0000000"
,x"6233697"
,x"5c45641"
,x"3e760d6"
,x"20a97ed"
,x"429a17a"
,x"7cec1ac"
,x"1edf73b"
,x"4152fda"
,x"236194d"
,x"1d1799b"
,x"7f24f0c"
,x"61fb837"
,x"03c8ea0"
,x"3dbee76"
,x"5f8d8e1"
,x"1a864db"
,x"78b524c"
,x"46c329a"
,x"24f040d"
,x"3a2f336"
,x"581c5a1"
,x"666a577"
,x"04593e0"
,x"5bd4b01"
,x"39e7d96"
,x"0791d40"
,x"65a2bd7"
,x"7b7dcec"
,x"194ea7b"
,x"2738aad"
,x"450bc3a"
,x"350c9b6"     ,x"365c52d"     ,x"6163a0c"     ,x"0e8bccd"
,x"573ff21"     ,x"546f3ba"     ,x"7fbcd37"     ,x"30fdc1b"
,x"6949ff7"     ,x"4ab0481"     ,x"1d8fba0"     ,x"52cea8c"
,x"0b7a960"     ,x"2883216"     ,x"23f9b76"     ,x"0d4326d"
,x"15a5e5b"     ,x"16f52c0"     ,x"41cade1"     ,x"6f704fa"
,x"77968cc"     ,x"74c6457"     ,x"1e47500"     ,x"510642c"
,x"49e081a"     ,x"2b4bcb6"     ,x"7c74397"     ,x"33352bb"
,x"2bd3e8d"     ,x"4978a21"     ,x"4202341"     ,x"2dea580"
,x"745e66c"     ,x"770eaf7"     ,x"20315d6"     ,x"4fd9317"
,x"166d0fb"     ,x"153dc60"     ,x"3eee2ed"     ,x"71af3c1"
,x"281b02d"     ,x"0be2b5b"     ,x"5cdd47a"     ,x"139c556"
,x"4a286ba"     ,x"69d1dcc"     ,x"62ab4ac"     ,x"569796c"
,x"54f7181"     ,x"57a7d1a"     ,x"009823b"     ,x"34a4ffb"
,x"36c4716"     ,x"3594b8d"     ,x"4593e01"     ,x"0ad2f2d"
,x"08b27c0"     ,x"709f7b7"     ,x"27a0896"     ,x"68e19ba"
,x"6a81157"     ,x"12ac120"     ,x"19d6840"     ,x"763ee81"
,x"2f8ad6d"     ,x"2cda1f6"     ,x"7be5ed7"     ,x"140d816"
,x"4db9bfa"     ,x"4ee9761"     ,x"653a9ec"     ,x"2a7b8c0"
,x"73cfb2c"     ,x"503605a"     ,x"0709f7b"     ,x"4848e57"
,x"11fcdbb"     ,x"32056cd"     ,x"397ffad"     ,x"17c56b6"
,x"0f23a80"     ,x"0c7361b"     ,x"5b4c93a"     ,x"75f6021"
,x"6d10c17"     ,x"6e4008c"     ,x"04c11db"     ,x"4b800f7"
,x"5366cc1"     ,x"31cd86d"     ,x"66f274c"     ,x"29b3660"
,x"3155a56"     ,x"53feefa"     ,x"588479a"     ,x"376c15b"
,x"6ed82b7"     ,x"6d88e2c"     ,x"3ab710d"     ,x"555f7cc"
,x"0ceb420"     ,x"0fbb8bb"     ,x"2468636"     ,x"6b2971a"
,x"329d4f6"     ,x"1164f80"     ,x"465b0a1"     ,x"091a18d"
,x"50ae261"     ,x"7357917"     ,x"782d077"     ,x"791d401"
,x"4e7155a"     ,x"4d219c1"     ,x"1a1e6e0"     ,x"1b2e296"
,x"2c423cd"     ,x"2f12f56"     ,x"4c11db7"     ,x"2558240"
,x"123431b"     ,x"5f15ada"     ,x"2e22b20"     ,x"476b4d7"
,x"700758c"     ,x"3d26c4d"     ,x"1054bf6"     ,x"59b43ec"
,x"6a1936c"     ,x"0350c9b"     ,x"7267d61"     ,x"3b8757b"
,x"082a5fb"                     ,x"6cb8a5a"     ,x"05f15ad"
```

```
,x"67c233a"     ,x"0569796"     ,x"6c20861"
,x"384fbdb"     ,x"3b1f740"     ,x"72fff5a"
,x"5a7cd4c"     ,x"592c1d7"     ,x"10cc9cd"
,x"640ad9a"     ,x"47f36ec"     ,x"2eba91b"
,x"0639b0d"     ,x"25c007b"     ,x"4c89f8c"
,x"18e6c36"     ,x"1bb60ad"     ,x"09823b6"
,x"7ad5aa1"     ,x"798563a"     ,x"6bb1521"
,x"44a3a77"     ,x"3c8ea00"     ,x"55c75f7"
,x"2690ce0"     ,x"5ebdc97"     ,x"37f4360"
,x"639b0da"     ,x"60cbc41"     ,x"292b45b"
,x"01a864d"     ,x"02f8ad6"     ,x"4b182cc"
,x"3fde69b"     ,x"1c27ded"     ,x"756e21a"
,x"5ded00c"     ,x"7e14b7a"     ,x"175d48d"
,x"4332737"     ,x"4062bac"     ,x"48d0c6c"
,x"21011a0"     ,x"2251d3b"     ,x"2ae3afb"
,x"1f77176"     ,x"7ddc5da"     ,x"1495a2d"
,x"7d447e1"     ,x"1fef34d"     ,x"76a6cba"
,x"22c9f00"     ,x"219939b"     ,x"6879b81"
,x"40fa997"     ,x"43aa50c"     ,x"0a4ad16"
,x"7e8c941"     ,x"5d75237"     ,x"343cdc0"
,x"1cbffd6"     ,x"3f464a0"     ,x"560fb57");
,x"02608ed"     ,x"0130476"
,x"6053e7a"     ,x"63032e1"
,x"5e25eac"     ,x"130476d"
,x"3c1683b"     ,x"71371fa"
,x"2608edb"     ,x"4f4112c"
,x"443b84c"     ,x"2d727bb"
,x"7a4d89a"     ,x"33ad080"
,x"187ee0d"     ,x"519e617"
,x"06a1936"     ,x"6fe86c1"
,x"6492fa1"     ,x"0ddb056"
,x"5ae4f77"     ,x"52568b7"
,x"38d79e0"     ,x"3065e20"
,x"675a101"     ,x"0e13ef6"
```

```
crcTable1 <=    ,x"32d844a"     ,x"2d9a5f5"     ,x"0c5c734"     ,x"131e68b"
(
                ,x"1f421bf"     ,x"3e8437e"     ,x"21c62c1"     ,x"7d086fc"
 x"0000000"
                ,x"5768cde"     ,x"76aee1f"     ,x"69ecfa0"     ,x"3522b9d"
,x"482ad61"
                ,x"1734012"     ,x"36f22d3"     ,x"29b036c"     ,x"757e751"
,x"08761ad"
                ,x"5f1ed73"     ,x"7ed8fb2"     ,x"619ae0d"     ,x"3d54a30"
,x"405cccc"
                ,x"0fae2e5"     ,x"2e68024"     ,x"312a19b"     ,x"6de45a6"
,x"10ec35a"
                ,x"4784f84"     ,x"6642d45"     ,x"7900cfa"     ,x"25ce8c7"
,x"58c6e3b"
                ,x"07d8348"     ,x"261e189"     ,x"395c036"     ,x"659240b"
,x"189a2f7"
                ,x"4ff2e29"     ,x"6e34ce8"     ,x"7176d57"     ,x"2db896a"
,x"50b0f96"
                ,x"3e9a70b"     ,x"1f5c5ca"     ,x"001e475"     ,x"5cd0048"
,x"21d86b4"
                ,x"76b0a6a"     ,x"57768ab"     ,x"4834914"     ,x"14fad29"
,x"69f2bd5"
                ,x"36ec6a6"     ,x"172a467"     ,x"08685d8"     ,x"54a61e5"
,x"29ae719"
                ,x"7ec6bc7"     ,x"5f00906"     ,x"40428b9"     ,x"1c8cc84"
,x"6184a78"
                ,x"2e76451"     ,x"0fb0690"     ,x"10f272f"     ,x"4c3c312"
,x"31345ee"
                ,x"665c930"     ,x"479abf1"     ,x"58d8a4e"     ,x"0416e73"
,x"791e88f"
                ,x"26005fc"     ,x"07c673d"     ,x"1884682"     ,x"444a2bf"
,x"3942443"
                ,x"6e2a89d"     ,x"4feca5c"     ,x"50aebe3"     ,x"0c60fde"
,x"7168922"
                ,x"5cf2cd7"     ,x"7d34e16"     ,x"6276fa9"     ,x"3eb8b94"
,x"43b0d68"
                ,x"14d81b6"     ,x"351e377"     ,x"2a5c2c8"     ,x"76926f5"
,x"0b9a009"
                ,x"5484d7a"     ,x"7542fbb"     ,x"6a00e04"     ,x"36cea39"
,x"4bc6cc5"
                ,x"1cae01b"     ,x"3d682da"     ,x"222a365"     ,x"7ee4758"
,x"03ec1a4"
                ,x"4c1ef8d"     ,x"6dd8d4c"     ,x"729acf3"     ,x"2e548ce"
,x"535ce32"
                ,x"04342ec"     ,x"25f202d"     ,x"3ab0192"     ,x"667e5af"
,x"1b76353"
                ,x"4468e20"     ,x"65aece1"     ,x"7aecd5e"     ,x"2622963"
,x"5b2af9f"
                ,x"0c42341"     ,x"2d84180"     ,x"32c603f"     ,x"6e08402"
,x"13002fe"
                ,x"7d2aa63"     ,x"5cec8a2"     ,x"43ae91d"     ,x"1f60d20"
,x"6268bdc"
                ,x"3500702"     ,x"14c65c3"     ,x"0b8447c"     ,x"574a041"
,x"2a426bd"
                ,x"755cbce"     ,x"549a90f"     ,x"4bd88b0"     ,x"1716c8d"
,x"6a1ea71"
                ,x"3d766af"     ,x"1cb046e"     ,x"03f25d1"     ,x"5f3c1ec"
,x"2234710"
                ,x"6dc6939"     ,x"4c00bf8"     ,x"5342a47"     ,x"0f8ce7a"
,x"7284886"
                ,x"25ec458"     ,x"042a699"     ,x"1b68726"     ,x"47a631b"
,x"3aae5e7"
                ,x"65b0894"     ,x"4476a55"     ,x"5b34bea"     ,x"07fafd7"
,x"7af292b"
```

```
,x"4fd02b6"     ,x"10cefc5"     ,x"69ce33f"     ,x"07e4ba2"
,x"624a743"     ,x"58e42a4"     ,x"2992ff3"     ,x"4fce6c3"
,x"2a60a22"     ,x"18b8e68"     ,x"61b8292"     ,x"3ea6fe1"
,x"6a3c6ee"     ,x"5092309"     ,x"3108d04"     ,x"768c280"
,x"2216b8f"     ,x"438c582"     ,x"7922065"     ,x"36d0e4c"
,x"72a6419"     ,x"0ba68e3"     ,x"397eca9"     ,x"7efa32d"
,x"3a8c978"     ,x"4bfa42f"     ,x"71541c8"     ,x"2e4acbb"
,x"7ad05b4"     ,x"03d094e"     ,x"5cce43d"     ,x"66601da"
,x"32fa8d5"     ,x"53606d8"     ,x"14e495c"     ,x"263cd16"
,x"43921f7"     ,x"1b4abb9"     ,x"54b8590"     ,x"6e16077"
,x"0bb8c96"     ,x"5b16775"     ,x"1c928f1"     );
,x"4be405a"     ,x"133ca14"     ,x"4c22767"
,x"03ced3b"     ,x"6254336"     ,x"0408a06"
,x"537e2ad"     ,x"2a7ee57"     ,x"44546ca"
,x"1b54fcc"     ,x"6a2229b"     ,x"0c7ebab"
,x"5b08300"     ,x"2208ffa"     ,x"7d16289"
,x"1322e61"     ,x"72b806c"     ,x"353cfe8"
,x"21faa2b"     ,x"3a92d0d"     ,x"7560324"
,x"69d074a"     ,x"7ace1c1"     ,x"3d4ae45"
,x"298cb86"     ,x"32e4ca0"     ,x"6dfa1d3"
,x"61a66e7"     ,x"003c8ea"     ,x"25d0cb2"
,x"3116971"     ,x"481658b"     ,x"658c07e"
,x"793c410"     ,x"084a947"     ,x"2da6d1f"
,x"39608dc"     ,x"4060426"     ,x"1f7e955"
,x"714a5bd"     ,x"10d0bb0"     ,x"5754434"
,x"0022c9f"     ,x"58fa6d1"     ,x"17088f8"
,x"48081fe"     ,x"18a6a1d"     ,x"5f22599"
,x"0854d32"     ,x"508c77c"     ,x"0f92a0f"
,x"407e053"     ,x"21e4e5e"     ,x"47b876e"
```

```
crcTable2 <= (  ,x"3c09c1a"     ,x"02ce5f4"     ,x"29824c1"     ,x"7f41628"
 x"0000000"     ,x"01672fa"     ,x"3fa0b14"     ,x"14eca21"     ,x"422f8c8"
,x"3d6eee0"     ,x"46d41da"     ,x"158b8db"     ,x"535f901"     ,x"2b171b6"
,x"7adddc0"     ,x"7bbaf3a"     ,x"28e563b"     ,x"6e317e1"     ,x"1679f56"
,x"47b3320"     ,x"5191cf5"     ,x"6f5651b"     ,x"0709e9f"     ,x"51cac76"
,x"6d980ef"     ,x"6cff215"     ,x"5238bfb"     ,x"3a6707f"     ,x"6ca4296"
,x"50f6e0f"     ,x"2b4c135"     ,x"3b00285"     ,x"7dd435f"     ,x"468f159"
,x"1745d2f"     ,x"1622fd5"     ,x"066ec65"     ,x"40badbf"     ,x"7be1fb9"
,x"2a2b3cf"     ,x"7f1a6ab"     ,x"41ddf45"     ,x"6a91e70"     ,x"3c52c99"
,x"4313ab1"     ,x"427484b"     ,x"7cb31a5"     ,x"57ff090"     ,x"013c279"
,x"7e7d451"     ,x"05c7b6b"     ,x"569826a"     ,x"104c3b0"     ,x"760050a"
,x"39ce771"     ,x"38a958b"     ,x"6bf6c8a"     ,x"2d22d50"     ,x"4b6ebea"
,x"04a0991"     ,x"1282644"     ,x"2c45faa"     ,x"5a1ea23"     ,x"0cdd8ca"
,x"2e8ba5e"     ,x"2fec8a4"     ,x"112b14a"     ,x"67704c3"     ,x"31b362a"
,x"13e54be"     ,x"685fb84"     ,x"6617639"     ,x"20c37e3"     ,x"1b985e5"
,x"545679e"     ,x"5531564"     ,x"5b798d9"     ,x"1dad903"     ,x"26f6b05"
,x"693897e"     ,x"220d217"     ,x"1ccabf9"     ,x"3786acc"     ,x"6145825"
,x"1e04e0d"     ,x"1f63cf7"     ,x"21a4519"     ,x"0ae842c"     ,x"5c2b6c5"
,x"236a0ed"     ,x"58d0fd7"     ,x"0b8f6d6"     ,x"4d5b70c"     ,x"3513fbb"
,x"64d93cd"     ,x"65be137"     ,x"36e1836"     ,x"70359ec"     ,x"087d15b"
,x"59b7d2d"     ,x"4f952f8"     ,x"7152b16"     ,x"190d092"     ,x"4fce27b"
,x"739cee2"     ,x"72fbc18"     ,x"4c3c5f6"     ,x"2463e72"     ,x"72a0c9b"
,x"4ef2002"     ,x"3548f38"     ,x"2504c88"     ,x"63d0d52"     ,x"588bf54"
,x"0941322"     ,x"08261d8"     ,x"186a268"     ,x"5ebe3b2"     ,x"65e51b4"
,x"342fdc2"     ,x"611e8a6"     ,x"5fd9148"     ,x"749507d"     ,x"2256294"
,x"5d174bc"     ,x"5c70646"     ,x"62b7fa8"     ,x"49fbe9d"     ,x"1f38c74"
,x"6079a5c"     ,x"1bc3566"     ,x"489cc67"     ,x"0e48dbd"     ,x"540d71d"
,x"27ca97c"     ,x"26adb86"     ,x"75f2287"     ,x"332635d"     ,x"69639fd"
,x"1aa479c"     ,x"0c86849"     ,x"32411a7"     ,x"6804b07"     ,x"2ed0add"
,x"308f453"     ,x"31e86a9"     ,x"0f2ff47"     ,x"556a5e7"     ,x"13be43d"
,x"0de1ab3"     ,x"765b589"     ,x"441a42e"     ,x"12d96c7"     ,x"39957f2"
,x"4a52993"     ,x"4b35b69"     ,x"7974ace"     ,x"2fb7827"     ,x"04fb912"
,x"773c773"     ,x"7813834"     ,x"3ec79ee"     ,x"059cbe8"     ,x"4348a32"
                ,x"457d6d4"     ,x"03a970e"     ,x"38f2508"     ,x"7e264d2"
```

```
,x"171edac"     ,x"29d9442"     ,x"0295577"
,x"2a7034c"     ,x"14b7aa2"     ,x"3ffbb97"
,x"6dc306c"     ,x"3e9c96d"     ,x"78488b7"
,x"50ade8c"     ,x"03f278d"     ,x"4526657"
,x"7a86d43"     ,x"44414ad"     ,x"321a124"
,x"47e83a3"     ,x"792fa4d"     ,x"0f74fc4"
,x"005b083"     ,x"0e13d3e"     ,x"48c7ce4"
,x"3d35e63"     ,x"337d3de"     ,x"75a9204"
,x"4a09910"     ,x"74ce0fe"     ,x"5f821cb"
,x"77677f0"     ,x"49a0e1e"     ,x"62ecf2b"
,x"30d44d0"     ,x"638bdd1"     ,x"255fc0b"
,x"0dbaa30"     ,x"5ee5331"     ,x"18312eb"
,x"27919ff"     ,x"1956011"     ,x"7109b95"
,x"1aff71f"     ,x"2438ef1"     ,x"4c67575"
,x"5d4c43f"     ,x"4d0078f"     ,x"0bd4655"
,x"6022adf"     ,x"706e96f"     ,x"36ba8b5"
,x"091a3a1"     ,x"37dda4f"     ,x"1c91b7a"
,x"3474d41"     ,x"0ab34af"     ,x"21ff59a"
,x"73c7e61"     ,x"2098760"     ,x"664c6ba"
,x"4ea9081"     ,x"1df6980"     ,x"5b2285a"
,x"648234e"     ,x"5a45aa0"     );
,x"59ecdae"     ,x"672b440"
,x"1e5fe8e"     ,x"2c1ef29"
,x"233106e"     ,x"11701c9"
,x"1017333"     ,x"56c32e9"
,x"2d79dd3"     ,x"6badc09"
,x"6acaef3"     ,x"4186fc6"
,x"57a4013"     ,x"7ce8126"
,x"7d8f3dc"     ,x"3b5b206"
,x"40e1d3c"     ,x"0635ce6"
,x"0752e1c"     ,x"6f0d598"
,x"3a3c0fc"     ,x"5263b78"
,x"5304982"     ,x"15d0858"
,x"6e6a762"     ,x"28be6b8"
```

```
crcTable3 <= (  ,x"5de0e16"     ,x"30db6c0"     ,x"1e5f3a2"     ,x"728fc03"
 x"0000000"     ,x"68abc93"     ,x"4974249"     ,x"2b14127"     ,x"0acbffd"
,x"1339183"     ,x"7b92d10"     ,x"5a4d3ca"     ,x"382d0a4"     ,x"19f2e7e"
,x"2672306"     ,x"07adddc"     ,x"6f0614f"     ,x"418242d"     ,x"2cb9cfb"
,x"354b285"     ,x"1494c5f"     ,x"7c3f0cc"     ,x"52bb5ae"     ,x"3f80d78"
,x"4ce460c"     ,x"21dfeda"     ,x"0f5bbb8"     ,x"67f072b"     ,x"462f9f1"
,x"5fdd78f"     ,x"32e6f59"     ,x"1c62a3b"     ,x"74c96a8"     ,x"5516872"
,x"6a9650a"     ,x"4b49bd0"     ,x"29298be"     ,x"08f6664"     ,x"605daf7"
,x"79af489"     ,x"5870a53"     ,x"3a1093d"     ,x"1bcf7e7"     ,x"7364b74"
,x"01eb777"     ,x"6d3b8d6"     ,x"43bfdb4"     ,x"2e84562"     ,x"1eb7770"
,x"12d26f4"     ,x"7e02955"     ,x"5086c37"     ,x"3dbd4e1"     ,x"0d8e6f3"
,x"2799471"     ,x"0646aab"     ,x"65cdeb2"     ,x"4412068"     ,x"38c5476"
,x"34a05f2"     ,x"157fb28"     ,x"76f4f31"     ,x"572b1eb"     ,x"2bfc5f5"
,x"4d0f17b"     ,x"20349ad"     ,x"0eb0ccf"     ,x"626036e"     ,x"525317c"
,x"5e360f8"     ,x"330d82e"     ,x"1d89d4c"     ,x"71592ed"     ,x"416a0ff"
,x"6b7d27d"     ,x"4aa2ca7"     ,x"28c2fc9"     ,x"091d113"     ,x"742127a"
,x"78443fe"     ,x"599bd24"     ,x"3bfbe4a"     ,x"1a24090"     ,x"67183f9"
,x"03d6eee"     ,x"6cd0fa1"     ,x"4254ac3"     ,x"2f6f215"     ,x"1f5c007"
,x"10eff6d"     ,x"7fe9e22"     ,x"516db40"     ,x"3c56396"     ,x"0c65184"
,x"25a4de8"     ,x"047b332"     ,x"64269c5"     ,x"45f971f"     ,x"392e301"
,x"369dc6b"     ,x"17422b1"     ,x"771f846"     ,x"56c069c"     ,x"2a17282"
,x"4f328e2"     ,x"2209034"     ,x"0c8d556"     ,x"638b419"     ,x"53b860b"
,x"5c0b961"     ,x"31301b7"     ,x"1fb44d5"     ,x"70b259a"     ,x"4081788"
,x"6940be4"     ,x"489f53e"     ,x"2aff650"     ,x"0b2088a"     ,x"75ca50d"
,x"7a79a67"     ,x"5ba64bd"     ,x"39c67d3"     ,x"1819909"     ,x"66f348e"
,x"023d999"     ,x"6eed638"     ,x"406935a"     ,x"2d52b8c"     ,x"1d6199e"
,x"110481a"     ,x"7dd47bb"     ,x"53502d9"     ,x"3e6ba0f"     ,x"0e5881d"
,x"244fa9f"     ,x"0590445"     ,x"661b05c"     ,x"47c4e86"     ,x"3b13a98"
,x"3776b1c"     ,x"16a95c6"     ,x"75221df"     ,x"54fdf05"     ,x"282ab1b"
,x"4ed9f95"     ,x"23e2743"     ,x"0d66221"     ,x"61b6d80"     ,x"5185f92"
```

```
,x"42bce11"     ,x"2f876c7"     ,x"01033a5"     ,x"6e052ea"
,x"77f7c94"     ,x"562824e"     ,x"3448120"     ,x"1597ffa"
,x"64ced17"     ,x"45113cd"     ,x"27710a3"     ,x"06aee79"
,x"1c8aee9"     ,x"705a148"     ,x"5ede42a"     ,x"33e5cfc"
,x"0fb3f6a"     ,x"63630cb"     ,x"4de75a9"     ,x"20dcd7f"
,x"3af8def"     ,x"1b27335"     ,x"78ac72c"     ,x"59739f6"
,x"29c1c6c"     ,x"081e2b6"     ,x"6b956af"     ,x"4a4a875"
,x"506e8e5"     ,x"3d55033"     ,x"13d1551"     ,x"7f01af0"
,x"4357966"     ,x"2e6c1b0"     ,x"00e84d2"     ,x"6c38b73"
,x"761cbe3"     ,x"57c3539"     ,x"35a3657"     ,x"147c88d"
,x"6525a60"     ,x"44fa4ba"     ,x"269a7d4"     ,x"074590e"
,x"191aaac"     ,x"71b163f"     ,x"5f3535d"     ,x"320eb8b"
,x"0a23b2f"     ,x"62887bc"     ,x"4c0c2de"     ,x"2137a08"
,x"3f689aa"     ,x"11eccc8"     ,x"794705b"     ,x"5898e81"
,x"2c51829"     ,x"02d5d4b"     ,x"6a7e1d8"     ,x"4ba1f02"
,x"55feca0"     ,x"379efce"     ,x"1641114"     ,x"7eead87"
,x"46c7d23"     ,x"24a7e4d"     ,x"0578097"     ,x"6dd3c04"
,x"738cfa6"     ,x"5d08ac4"     ,x"3033212"     );
,x"60b5e25"     ,x"4e31b47"     ,x"230a391"
,x"18f1ddb"     ,x"7b7a9c2"     ,x"5aa5718"
,x"0bc8c58"     ,x"6843841"     ,x"499c69b"
,x"3e83edd"     ,x"1007bbf"     ,x"7cd741e"
,x"2dbaf5e"     ,x"033ea3c"     ,x"6fee59d"
,x"5415bd7"     ,x"36758b9"     ,x"17aa663"
,x"472ca54"     ,x"254c93a"     ,x"04937e0"
,x"72678d1"     ,x"5ce3db3"     ,x"31d8565"
,x"615e952"     ,x"4fdac30"     ,x"22e14e6"
,x"1acc442"     ,x"7a91eb5"     ,x"5b4e06f"
,x"09f55c1"     ,x"69a8f36"     ,x"48771ec"
,x"3cbe744"     ,x"123a226"     ,x"7d3c369"
```

```
crcTable4 <=
(
 x"0000000"
,x"7274ef0"
,x"7cca68f"
,x"0ebe87f"
,x"61b7671"
,x"13c3881"
,x"1d7d0fe"
,x"6f09e0e"
,x"5b4d78d"
,x"293997d"
,x"2787102"
,x"55f3ff2"
,x"3afa1fc"
,x"488ef0c"
,x"4630773"
,x"3444983"
,x"2eb9475"
,x"5ccda85"
,x"52732fa"
,x"2007c0a"
,x"4f0e204"
,x"3d7acf4"
,x"33c448b"
,x"41b0a7b"
,x"75f43f8"
,x"0780d08"
,x"093e577"
,x"7b4ab87"
,x"1443589"
,x"6637b79"
,x"6889306"
,x"1afddf6"    ,x"22c6abb"    ,x"037e4de"    ,x"243a307"
,x"5d728ea"    ,x"50b244b"    ,x"710aa2e"    ,x"564edf7"
,x"2f0661a"    ,x"5e0cc34"    ,x"1e03420"    ,x"58f0588"
,x"21b8e65"    ,x"2c782c4"    ,x"6c77ad0"    ,x"2a84b78"
,x"53cc095"    ,x"4371cca"    ,x"62c92af"    ,x"1ec02fb"
,x"3cc5e9b"    ,x"310523a"    ,x"10bdc5f"    ,x"6cb4c0b"
,x"4eb106b"    ,x"3fbba45"    ,x"24f95dc"    ,x"620a474"
,x"400f814"    ,x"4dcf4b5"    ,x"568db2c"    ,x"107ea84"
,x"327b6e4"    ,x"798bd36"    ,x"5833353"    ,x"7f7748a"
,x"063ff67"    ,x"0bff3c6"    ,x"2a47da3"    ,x"0d03a7a"
,x"744b197"    ,x"0541bb9"    ,x"454e3ad"    ,x"03bd205"
,x"7af59e8"    ,x"7735549"    ,x"373ad5d"    ,x"71c9cf5"
,x"0881718"    ,x"183cb47"    ,x"3984522"    ,x"6b34103"
,x"6788916"    ,x"6a485b7"    ,x"4bf0bd2"    ,x"1940ff3"
,x"15fc7e6"    ,x"64f6dc8"    ,x"510d624"    ,x"17fe78c"
,x"1b42f99"    ,x"1682338"    ,x"23798d4"    ,x"658a97c"
,x"6936169"    ,x"0c7fece"    ,x"2dc70ab"    ,x"0a83772"
,x"73cbc9f"    ,x"7e0b03e"    ,x"5fb3e5b"    ,x"78f7982"
,x"01bf26f"    ,x"70b5841"    ,x"30ba055"    ,x"76491fd"
,x"0f01a10"    ,x"02c16b1"    ,x"42ceea5"    ,x"043df0d"
,x"7d754e0"    ,x"6dc88bf"    ,x"4c706da"    ,x"307968e"
,x"127caee"    ,x"1fbc64f"    ,x"3e0482a"    ,x"420d87e"
,x"600841e"    ,x"1102e30"    ,x"0a401a9"    ,x"4cb3001"
,x"6eb6c61"    ,x"63760c0"    ,x"7834f59"    ,x"3ec7ef1"
,x"1cc2291"    ,x"5732943"    ,x"768a726"    ,x"51ce0ff"
,x"2886b12"    ,x"25467b3"    ,x"04fe9d6"    ,x"23bae0f"
,x"5af25e2"    ,x"2bf8fcc"    ,x"6bf77d8"    ,x"2d04670"
,x"544cd9d"    ,x"598c13c"    ,x"1983928"    ,x"5f70880"
,x"263836d"    ,x"3685f32"    ,x"173d157"    ,x"18ffd9c"
,x"4931d63"    ,x"44f11c2"    ,x"6549fa7"    ,x"6a8b36c"
,x"3b45393"    ,x"4a4f9bd"    ,x"458d576"    ,x"6435b13"
,x"35fbbec"    ,x"383b74d"    ,x"37f9b86"    ,x"16415e3"
,x"478f51c"    ,x"7fb4251"    ,x"39473f9"    ,x"7948bed"
               ,x"0dc0ca1"    ,x"4b33d09"    ,x"0b3c51d"
```

```
,x"0582d62"     ,x"06fc9bc"     ,x"46f31a8"
,x"77f6392"     ,x"748874c"     ,x"3487f58"
,x"43b2a11"     ,x"7a36f33"     ,x"5b8e156"
,x"31c64e1"     ,x"08421c3"     ,x"29fafa6"
,x"3f78c9e"     ,x"3c06840"     ,x"27447d9"
,x"4d0c26e"     ,x"4e726b0"     ,x"5530929"
,x"2205c60"     ,x"40ccecf"     ,x"61740aa"
,x"5071290"     ,x"32b803f"     ,x"1300e5a"
,x"5ecfaef"     ,x"5db1e31"     ,x"1dbe625"
,x"2cbb41f"     ,x"2fc50c1"     ,x"6fca8d5"
,x"36469e9"     ,x"217b8be"     ,x"00c36db"
,x"4432719"     ,x"530f64e"     ,x"72b782b"
,x"4a8cf66"     ,x"49f2bb8"     ,x"7c09054"
,x"38f8196"     ,x"3b86548"     ,x"0e7dea4"
,x"57f1f98"     ,x"3538d37"     ,x"1480352"
,x"2585168"     ,x"474c3c7"     ,x"66f4da2"
,x"2b3b917"     ,x"2845dc9"     ,x"684a5dd"
,x"594f7e7"     ,x"5a31339"     ,x"1a3eb2d"
,x"6d0be64"     ,x"548fb46"     ,x"7537523"
,x"1f7f094"     ,x"26fb5b6"     ,x"0743bd3"
,x"11c18eb"     ,x"12bfc35"     ,x"09fd3ac"
,x"63b561b"     ,x"60cb2c5"     ,x"7b89d5c"
,x"0cbc815"     ,x"6e75aba"     ,x"4fcd4df"
,x"7ec86e5"     ,x"1c0144a"     ,x"3db9a2f"
,x"7076e9a"     ,x"7308a44"     ,x"3307250"
,x"020206a"     ,x"017c4b4"     ,x"4173ca0"
,x"674bfcd"     ,x"0fc2ccb"     ,x"2e7a2ae"
,x"153f13d"     ,x"7db623b"     ,x"5c0ec5e"
,x"1b81942"     ,x"3a39727"     ,x"52b0421"
,x"69f57b2"     ,x"484d9d7"     ,x"20c4ad1"
                                );
```

```
crcTable5 <=
(
 x"0000000"
,x"09f04f8"
,x"13e09f0"
,x"1a10d08"
,x"27c13e0"
,x"2e31718"
,x"3421a10"
,x"3dd1ee8"
,x"4f827c0"
,x"4672338"
,x"5c62e30"
,x"5592ac8"
,x"6843420"
,x"61b30d8"
,x"7ba3dd0"
,x"7253928"
,x"07274ef"
,x"0ed7017"
,x"14c7d1f"
,x"1d379e7"
,x"20e670f"
,x"29163f7"
,x"3306eff"
,x"3af6a07"
,x"48a532f"
,x"41557d7"
,x"5b45adf"
,x"52b5e27"
,x"6f640cf"
,x"6694437"
,x"7c8493f"
,x"7574dc7"    ,x"1c9d3bc"    ,x"1b23e9a"    ,x"2adae88"
,x"0e4e9de"    ,x"156d744"    ,x"0133392"    ,x"232aa70"
,x"07bed26"    ,x"0f7da4c"    ,x"08c376a"    ,x"1efb498"
,x"1dae02e"    ,x"068deb4"    ,x"3512982"    ,x"170b060"
,x"145e4d6"    ,x"3b5c05c"    ,x"3ce2d7a"    ,x"0d1bd68"
,x"298fa3e"    ,x"32ac4a4"    ,x"26f2072"    ,x"04eb990"
,x"207fec6"    ,x"28bc9ac"    ,x"2f0248a"    ,x"76b80b8"
,x"3a6f3ce"    ,x"214cd54"    ,x"5d51da2"    ,x"7f48440"
,x"339f736"    ,x"531f47c"    ,x"54a195a"    ,x"6558948"
,x"41cce1e"    ,x"5aef084"    ,x"4eb1452"    ,x"6ca8db0"
,x"483cae6"    ,x"40ffd8c"    ,x"47410aa"    ,x"5179358"
,x"522c7ee"    ,x"490f974"    ,x"7a90e42"    ,x"58897a0"
,x"5bdc316"    ,x"74de79c"    ,x"7360aba"    ,x"4299aa8"
,x"660ddfe"    ,x"7d2e364"    ,x"69707b2"    ,x"4b69e50"
,x"6ffd906"    ,x"673ee6c"    ,x"608034a"    ,x"3e1d397"
,x"75ed40e"    ,x"6ecea94"    ,x"15f4e8d"    ,x"37ed76f"
,x"7c1d0f6"    ,x"1bba753"    ,x"1c04a75"    ,x"2dfda67"
,x"0969d31"    ,x"124a3ab"    ,x"061477d"    ,x"240de9f"
,x"00999c9"    ,x"085aea3"    ,x"0fe4385"    ,x"19dc077"
,x"1a894c1"    ,x"01aaa5b"    ,x"3235d6d"    ,x"102c48f"
,x"1379039"    ,x"3c7b4b3"    ,x"3bc5995"    ,x"0a3c987"
,x"2ea8ed1"    ,x"358b04b"    ,x"21d549d"    ,x"03ccd7f"
,x"2758a29"    ,x"2f9bd43"    ,x"2825065"    ,x"719f457"
,x"3d48721"    ,x"266b9bb"    ,x"5a7694d"    ,x"786f0af"
,x"34b83d9"    ,x"5438093"    ,x"5386db5"    ,x"627fda7"
,x"46ebaf1"    ,x"5dc846b"    ,x"49960bd"    ,x"6b8f95f"
,x"4f1be09"    ,x"47d8963"    ,x"4066445"    ,x"565e7b7"
,x"550b301"    ,x"4e28d9b"    ,x"7db7aad"    ,x"5fae34f"
,x"5cfb7f9"    ,x"73f9373"    ,x"7447e55"    ,x"45bee47"
,x"612a911"    ,x"7a0978b"    ,x"6e5735d"    ,x"4c4eabf"
,x"68dade9"    ,x"6019a83"    ,x"67a77a5"    ,x"3774ea6"
,x"72ca0e1"    ,x"69e9e7b"    ,x"393a778"    ,x"3e84a5e"
,x"7b3a419"    ,x"12d3a62"    ,x"30ca380"
```

```
,x"2494756"     ,x"4bf0799"     ,x"7ee2e1b"     ,x"18ef3e5"
,x"2d643ae"     ,x"4200361"     ,x"7712ae3"     ,x"111f71d"
,x"10b5d46"     ,x"25a74c4"     ,x"4ac340b"     ,x"634ce35"
,x"19459be"     ,x"2c5703c"     ,x"43330f3"     ,x"6abcacd"
,x"03554b6"     ,x"3647d34"     ,x"5923dfb"     ,x"70ac7c5"
,x"0aa504e"     ,x"3fb79cc"     ,x"50d3903"     ,x"795c33d"
,x"78f6966"     ,x"0266724"     ,x"2be9d1a"     ,x"448ddd5"
,x"7106d9e"     ,x"0b963dc"     ,x"22199e2"     ,x"4d7d92d"
,x"6b16096"     ,x"1186ed4"     ,x"38094ea"     ,x"576d425"
,x"62e646e"     ,x"1876a2c"     ,x"31f9012"     ,x"5e9d0dd"
,x"5f37a86"     ,x"6a25304"     ,x"0c28efa"     );
,x"56c7e7e"     ,x"63d57fc"     ,x"05d8a02"
,x"4cd7376"     ,x"79c5af4"     ,x"1fc870a"
,x"452778e"     ,x"7035e0c"     ,x"16383f2"
,x"3053a49"     ,x"4de40e4"     ,x"646bada"
,x"39a3eb1"     ,x"441441c"     ,x"6d9be22"
,x"23b33b9"     ,x"5e04914"     ,x"778b32a"
,x"2a43741"     ,x"57f4dec"     ,x"7e7b7d2"
,x"17929a9"     ,x"228002b"     ,x"43aa93a"
,x"1e62d51"     ,x"2b704d3"     ,x"4a5adc2"
,x"0472059"     ,x"31609db"     ,x"59ba432"
,x"0d824a1"     ,x"3890d23"     ,x"2cce9f5"
,x"7fd1d89"     ,x"05413cb"     ,x"253ed0d"
,x"7621971"     ,x"0cb1733"     ,x"3f2e005"
,x"6c31479"     ,x"16a1a3b"     ,x"36de4fd"
,x"65c1081"     ,x"1f51ec3"     ,x"0b0fa15"
,x"5810e69"     ,x"6d027eb"     ,x"02ffeed"
,x"51e0a91"     ,x"64f2313"     ,x"504a0ca"
```

```
crcTable6 <= (  ,x"707bdc5"     ,x"300d38d"     ,x"7b96144"     ,x"28c3472"
 x"0000000"     ,x"38a2ead"     ,x"7145d5a"     ,x"334f22c"     ,x"7aa81db"
,x"48d9368"     ,x"013e09f"     ,x"399ce32"     ,x"7207cfb"     ,x"32712b3"
,x"0991dbf"     ,x"49e73f7"     ,x"027c13e"     ,x"3adef93"     ,x"7339c64"
,x"4148ed7"     ,x"08afd20"     ,x"4aa5256"     ,x"03421a1"     ,x"3be0f0c"
,x"1323b7e"     ,x"4076e48"     ,x"0bedc81"     ,x"4b9b2c9"     ,x"04f827c"
,x"5bfa816"     ,x"121dbe1"     ,x"4334fe9"     ,x"0ad3c1e"     ,x"4c21114"
,x"1ab26c1"     ,x"5ac4889"     ,x"115fa40"     ,x"420af76"     ,x"0d69fc3"
,x"526b5a9"     ,x"1b8c65e"     ,x"5986928"     ,x"1061adf"     ,x"45b0cab"
,x"26476fc"     ,x"5355536"     ,x"18ce7ff"     ,x"58b89b7"     ,x"17db902"
,x"6e9e594"     ,x"2779663"     ,x"5017497"     ,x"19f0760"     ,x"5f02a6a"
,x"2fd6b43"     ,x"6fa050b"     ,x"243b7c2"     ,x"5129408"     ,x"1e4a4bd"
,x"670f82b"     ,x"2ee8bdc"     ,x"6ce24aa"     ,x"250575d"     ,x"56937d5"
,x"3564d82"     ,x"66318b4"     ,x"2daaa7d"     ,x"6ddc435"     ,x"22bf480"
,x"7dbdeea"     ,x"345ad1d"     ,x"6573915"     ,x"2c94ae2"     ,x"6a667e8"
,x"3cf503d"     ,x"7c83e75"     ,x"3718cbc"     ,x"644d98a"     ,x"2b2e93f"
,x"742c355"     ,x"3dcb0a2"     ,x"7fc1fd4"     ,x"3626c23"     ,x"63f7a57"
,x"4c8edf8"     ,x"75123ca"     ,x"3e89103"     ,x"7efff4b"     ,x"319cffe"
,x"0457e90"     ,x"4db0d67"     ,x"765026b"     ,x"3fb719c"     ,x"7945c96"
,x"451f047"     ,x"0569e0f"     ,x"4ef2cc6"     ,x"776e2f4"     ,x"380d241"
,x"0dc632f"     ,x"44210d8"     ,x"062bfae"     ,x"4fccc59"     ,x"70d4129"
,x"5fad686"     ,x"0cf83b0"     ,x"4763179"     ,x"0715f31"     ,x"4876f84"
,x"17745ee"     ,x"5e93619"     ,x"0fba211"     ,x"465d1e6"     ,x"00afcec"
,x"563cb39"     ,x"164a571"     ,x"5dd17b8"     ,x"0e8428e"     ,x"41e723b"
,x"1ee5851"     ,x"5702ba6"     ,x"15084d0"     ,x"5cef727"     ,x"093e153"
,x"6ac9b04"     ,x"1fdb8ce"     ,x"5440a07"     ,x"143644f"     ,x"5b554fa"
,x"221086c"     ,x"6bf7b9b"     ,x"1c9996f"     ,x"557ea98"     ,x"138c792"
,x"63586bb"     ,x"232e8f3"     ,x"68b5a3a"     ,x"1da79f0"     ,x"52c4945"
,x"2b815d3"     ,x"6266624"     ,x"206c952"     ,x"698baa5"     ,x"1a1da2d"
,x"79ea07a"     ,x"2abf54c"     ,x"6124785"     ,x"21529cd"     ,x"6e31978"
,x"3133312"     ,x"78d40e5"     ,x"29fd4ed"     ,x"601a71a"     ,x"26e8a10"
```

```
,x"67a04c7"     ,x"27d6a8f"     ,x"6c4d846"     ,x"195fb8c"
,x"2f797af"     ,x"669e458"     ,x"2494b2e"     ,x"6d738d9"
,x"7d12206"     ,x"2e47730"     ,x"65dc5f9"     ,x"25aabb1"
,x"35cb16e"     ,x"7c2c299"     ,x"2d05691"     ,x"64e2566"
,x"7483fb9"     ,x"34f51f1"     ,x"7f6e338"     ,x"2c3b60e"
,x"3c5acd1"     ,x"75bdf26"     ,x"37b7050"     ,x"7e503a7"
,x"05c62e3"     ,x"3d64c4e"     ,x"76ffe87"     ,x"36890cf"
,x"4d1f18b"     ,x"0684342"     ,x"3e26def"     ,x"77c1e18"
,x"0c57f5c"     ,x"4e5d02a"     ,x"07ba3dd"     ,x"3f18d70"
,x"448ec34"     ,x"0f15efd"     ,x"4f630b5"     );
,x"16e599d"     ,x"47ccd95"     ,x"0e2be62"
,x"5e3caf5"     ,x"15a783c"     ,x"46f2d0a"
,x"1f74422"     ,x"5d7eb54"     ,x"14998a3"
,x"57ad74a"     ,x"1c36583"     ,x"5c40bcb"
,x"238141f"     ,x"54ef6eb"     ,x"1d0851c"
,x"6b58777"     ,x"20c35be"     ,x"55d1674"
,x"2a109a0"     ,x"681a6d6"     ,x"21fd521"
,x"62c9ac8"     ,x"2952801"     ,x"6924649"
,x"30a2f61"     ,x"618bb69"     ,x"286c89e"
,x"787bc09"     ,x"33e0ec0"     ,x"60b5bf6"
,x"39332de"     ,x"7b39da8"     ,x"32dee5f"
,x"71ea1b6"     ,x"3a7137f"     ,x"7a07d37"
,x"4948f1b"     ,x"72a8017"     ,x"3b4f3e0"
,x"0191c73"     ,x"4a0aeba"     ,x"7396088"
,x"40d92a4"     ,x"02d3dd2"     ,x"4b34e25"
,x"08001cc"     ,x"439b305"     ,x"03edd4d"
,x"5a6b465"     ,x"0b4206d"     ,x"42a539a"
,x"12b270d"     ,x"59295c4"     ,x"0a7c0f2"
,x"53fa9da"     ,x"11f06ac"     ,x"581755b"
,x"1b23ab2"     ,x"50b887b"     ,x"10ce633"
,x"6f0f9e7"     ,x"1861b13"     ,x"51868e4"
```

```
crcTable7 <= (  ,x"58fd93a"     ,x"37d7188"     ,x"1e8b331"     ,x"71a1b83"
 x"0000000"     ,x"46f21a9"     ,x"15c783d"     ,x"0084ba2"     ,x"2b8f07a"
,x"1e0f893"     ,x"64e281c"     ,x"0bc80ae"     ,x"32f5fac"     ,x"35808e9"
,x"3c1f126"     ,x"7aed08f"     ,x"51e6b57"     ,x"2cfa73f"     ,x"179015c"
,x"22109b5"     ,x"091b26d"     ,x"4fe93c4"     ,x"0eeae8a"     ,x"099f9cf"
,x"783e24c"     ,x"1714afe"     ,x"6df9a71"     ,x"10e5619"     ,x"3beedc1"
,x"6631adf"     ,x"350434b"     ,x"73f62e2"     ,x"4acbde0"     ,x"25e1552"
,x"442136a"     ,x"2b0bbd8"     ,x"12364da"     ,x"54c4573"     ,x"07f1ce7"
,x"5a2ebf9"     ,x"7125021"     ,x"0c39c49"     ,x"76d4cc6"     ,x"19fe474"
,x"685fff7"     ,x"6f2a8b2"     ,x"2e295fc"     ,x"68db455"     ,x"43d0f8d"
,x"7650764"     ,x"4d3a107"     ,x"3026d6f"     ,x"1b2d6b7"     ,x"5ddf71e"
,x"5440ed1"     ,x"5335994"     ,x"6a08696"     ,x"0522e24"     ,x"7fcfeab"
,x"4a4f642"     ,x"6144d9a"     ,x"7407e05"     ,x"2732791"     ,x"61c0638"
,x"1061dbb"     ,x"7f4b509"     ,x"56177b0"     ,x"393df02"     ,x"246c9b4"
,x"0e6e528"     ,x"5d5bcbc"     ,x"4818f23"     ,x"63134fb"     ,x"3a63127"
,x"2c7ec9d"     ,x"435442f"     ,x"7a69b2d"     ,x"7d1cc68"     ,x"1873892"
,x"327140e"     ,x"197afd6"     ,x"64663be"     ,x"5f0c5dd"     ,x"067c001"
,x"489c481"     ,x"0775745"     ,x"4676a0b"     ,x"4103d4e"     ,x"5c52bf8"
,x"5693c12"     ,x"2565ef0"     ,x"5879298"     ,x"7372940"     ,x"425d36b"
,x"74835a7"     ,x"3b6a663"     ,x"0257961"     ,x"6d7d1d3"     ,x"604dade"
,x"6a8cd34"     ,x"41876ec"     ,x"1c581f2"     ,x"4f6d866"     ,x"7e4224d"
,x"30a26cd"     ,x"5f88e7f"     ,x"3e48847"     ,x"51620f5"     ,x"4c33643"
,x"2eade5e"     ,x"7d987ca"     ,x"20470d4"     ,x"0b4cb0c"     ,x"523ced0"
,x"0cbd7eb"     ,x"6397f59"     ,x"5aaa05b"     ,x"154339f"     ,x"702c765"
,x"12b2f78"     ,x"39b94a0"     ,x"44a58c8"     ,x"3753a2a"     ,x"6e23ff6"
,x"20c3b76"     ,x"27b6c33"     ,x"66b517d"     ,x"295c2b9"     ,x"340d40f"
,x"3ecc3e5"     ,x"05a6586"     ,x"78ba9ee"     ,x"53b1236"     ,x"2a02c9c"
,x"1cdca50"     ,x"1ba9d15"     ,x"2294217"     ,x"4dbeaa5"     ,x"0812529"
,x"02d32c3"     ,x"29d891b"     ,x"3c9ba84"     ,x"6fae310"     ,x"161ddba"
```

```
,x"6cf0d35"     ,x"2319ef1"     ,x"621a3bf"     ,x"656f4fa"
,x"72ff5a6"     ,x"0109744"     ,x"7c15b2c"     ,x"571e0f4"
,x"50efc13"     ,x"1f06fd7"     ,x"263b0d5"     ,x"4911867"
,x"4ee0480"     ,x"65ebf58"     ,x"3834846"     ,x"6b011d2"
,x"14cef79"     ,x"7be47cb"     ,x"1a241f3"     ,x"750e941"
,x"0ac17ea"     ,x"59f4e7e"     ,x"042b960"     ,x"2f202b8"
,x"28d1e5f"     ,x"47fb6ed"     ,x"7ec69ef"     ,x"312fa2b"
,x"36de6cc"     ,x"1dd5d14"     ,x"60c917c"     ,x"133f39e"
,x"04af2c2"     ,x"03da587"     ,x"42d98c9"     ,x"0d30b0d"
,x"1aa0a51"     ,x"21cac32"     ,x"5cd605a"     ,x"77ddb82"
,x"38b03e4"     ,x"3fc54a1"     ,x"06f8ba3"     ,x"69d2311"
,x"26bfb77"     ,x"0db40af"     ,x"18f7330"     ,x"4bc2aa4"
,x"7c9108e"     ,x"13bb83c"     ,x"3ae7a85"     ,x"55cd237"
,x"629e81d"     ,x"31ab189"     ,x"24e8216"     ,x"0fe39ce"
,x"408e1a8"     ,x"2fa491a"     ,x"1699618"     ,x"11ec15d"
,x"5e8193b"     ,x"758a2e3"     ,x"0896e8b"     ,x"33fc8e8"
,x"2d77bd9"     ,x"6b85a70"     ,x"2a8673e"     ,x"2df307b"
,x"337834a"     ,x"49953c5"     ,x"3489fad"     ,x"1f82475"
,x"1168aff"     ,x"579ab56"     ,x"6ea7454"     ,x"018dce6"
,x"0f6726c"     ,x"365ad6e"     ,x"70a8cc7"     ,x"239d553"
,x"5549995"     ,x"28555fd"     ,x"52b8572"     ,x"3d92dc0"
,x"4b46106"     ,x"0a45c48"     ,x"4cb7de1"     ,x"67bc639"
,x"69568b3"     ,x"144a4db"     ,x"3f41f03"     ,x"79b3eaa"
,x"7759020"     ,x"4e64f22"     ,x"214e790"     ,x"5ba371f"
,x"452842e"     ,x"506b7b1"     ,x"035ee25"     ,x"45acf8c"
,x"5b27cbd"     ,x"727be04"     ,x"1d516b6"     );
,x"7937508"     ,x"6c74697"     ,x"477fd4f"
,x"6738d9b"     ,x"5e05299"     ,x"59705dc"
,x"3d16662"     ,x"400aa0a"     ,x"7b60c69"
```

```
crcTable8 <= (  ,x"7b192c4"     ,x"6e11ee7"     ,x"5108a82"     ,x"44006a1"
 x"0000000"     ,x"13e6b86"     ,x"06ee7a5"     ,x"39f73c0"     ,x"2cfffe3"
,x"68ff942"     ,x"5cd45c8"     ,x"27cd70c"     ,x"134ebd0"     ,x"0ddcf4a"
,x"49dc9eb"     ,x"342bc8a"     ,x"4f32e4e"     ,x"7bb1292"     ,x"6523608"
,x"21230a9"     ,x"1508c23"     ,x"218b0ff"     ,x"5a9223b"     ,x"4f9ae18"
,x"0b9a8b9"     ,x"7df7561"     ,x"49749bd"     ,x"326db79"     ,x"276575a"
,x"63651fb"     ,x"574ed71"     ,x"6857914"     ,x"7d5f537"     ,x"06467f3"
,x"4246152"     ,x"3fb1433"     ,x"00a8056"     ,x"15a0c75"     ,x"6eb9eb1"
,x"2ab9810"     ,x"1e9249a"     ,x"2a11846"     ,x"3483cdc"     ,x"43161fe"
,x"1735172"     ,x"766ddd8"     ,x"42ee104"     ,x"5c7c59e"     ,x"2be98bc"
,x"7fca830"     ,x"4be14ba"     ,x"63cd1ad"     ,x"76c5d8e"     ,x"0aca815"
,x"5ee9899"     ,x"231edf8"     ,x"0b328ef"     ,x"1e3a4cc"     ,x"6235157"
,x"36161db"     ,x"023dd51"     ,x"36be18d"     ,x"3f19465"     ,x"488c947"
,x"1caf9cb"     ,x"6ac2413"     ,x"5e418cf"     ,x"57e6d27"     ,x"2073005"
,x"7450089"     ,x"407bc03"     ,x"7f62866"     ,x"6a6a445"     ,x"01500ac"
,x"5573020"     ,x"2884541"     ,x"179d124"     ,x"0295d07"     ,x"69af9ee"
,x"3d8c962"     ,x"09a75e8"     ,x"3d24934"     ,x"23b6dae"     ,x"542308c"
,x"2e6a2e4"     ,x"6158caa"     ,x"55db076"     ,x"4b494ec"     ,x"3cdc9ce"
,x"4695ba6"     ,x"72be72c"     ,x"74f80df"     ,x"61f0cfc"     ,x"1dff967"
,x"67b6b0f"     ,x"1a41e6e"     ,x"1c0799d"     ,x"090f5be"     ,x"7500025"
,x"0f4924d"     ,x"3b62ec7"     ,x"0fe121b"     ,x"282c517"     ,x"5fb9835"
,x"25f0a5d"     ,x"539d785"     ,x"671eb59"     ,x"40d3c55"     ,x"3746177"
,x"4d0f31f"     ,x"7924f95"     ,x"463dbf0"     ,x"53357d3"     ,x"16651de"
,x"6c2c3b6"     ,x"11db6d7"     ,x"2ec22b2"     ,x"3bcae91"     ,x"7e9a89c"
,x"04d3af4"     ,x"30f867e"     ,x"047baa2"     ,x"1ae9e38"     ,x"6d7c31a"
,x"395f396"     ,x"5807f3c"     ,x"6c843e0"     ,x"721677a"     ,x"0583a58"
,x"51a0ad4"     ,x"658b65e"     ,x"4da7349"     ,x"58aff6a"     ,x"24a0af1"
,x"7083a7d"     ,x"0d74f1c"     ,x"2558a0b"     ,x"3050628"     ,x"4c5f3b3"
,x"187c33f"     ,x"2c57fb5"     ,x"18d4369"     ,x"1173681"     ,x"66e6ba3"
,x"32c5b2f"     ,x"44a86f7"     ,x"702ba2b"     ,x"798cfc3"     ,x"0e192e1"
,x"5a3a26d"
```

```
,x"2f3a248"     ,x"108b67b"     ,x"2408aa7"     ,x"03c5dab"
,x"47c5b0a"     ,x"3a32e6b"     ,x"052ba0e"     ,x"102362d"
,x"7a49268"     ,x"52cd729"     ,x"6dd434c"     ,x"78dcf6f"
,x"12b6b2a"     ,x"73ee780"     ,x"476db5c"     ,x"59fffc6"
,x"3395b83"     ,x"1b11ec2"     ,x"2f9221e"     ,x"3100684"
,x"5b6a2c1"     ,x"269d7a0"     ,x"0eb12b7"     ,x"1bb9e94"
,x"71d3ad1"     ,x"4e62ee2"     ,x"664ebf5"     ,x"73467d6"
,x"192c393"     ,x"6f41e4b"     ,x"5bc2297"     ,x"526577f"
,x"380f33a"     ,x"07be709"     ,x"333dbd5"     ,x"3a9ae3d"
,x"50f0a78"     ,x"2d07f19"     ,x"121eb7c"     ,x"071675f"
,x"1fc2436"     ,x"45f865b"     ,x"7ae123e"     ,x"6fe9e1d"
,x"773dd74"     ,x"64db6f2"     ,x"5058a2e"     ,x"4ecaeb4"
,x"561eddd"     ,x"0c24fb0"     ,x"38a736c"     ,x"26357f6"
,x"3ee149f"     ,x"629d101"     ,x"19843c5"     ,x"0c8cfe6"
,x"1458c8f"     ,x"0a62843"     ,x"717ba87"     ,x"64736a4"
,x"7ca75cd"     ,x"2b418ea"     ,x"3e494c9"     ,x"455060d"
,x"5d84564"     ,x"43be1a8"     ,x"56b6d8b"     ,x"2daff4f"
,x"357bc26"     ,x"69079b8"     ,x"7795d22"     );
,x"08f7544"     ,x"01f80fa"     ,x"1f6a460"
,x"6008c06"     ,x"20db053"     ,x"35d3c70"
,x"412bcaf"     ,x"4824911"     ,x"5d2c532"
,x"29d45ed"     ,x"75a8073"     ,x"7c0f59b"
,x"036ddfd"     ,x"1d57931"     ,x"14f0cd9"
,x"6b924bf"     ,x"3c74998"     ,x"297c5bb"
,x"4ab1416"     ,x"548b0da"     ,x"4183cf9"
,x"224ed54"     ,x"7e328ca"     ,x"60a0c50"
,x"31a86d2"     ,x"16cd188"     ,x"085f512"
,x"5957f90"     ,x"37ee121"     ,x"22e6d02"
,x"7874f39"     ,x"5f11863"     ,x"4a19440"
                ,x"4cf73e5"     ,x"6b3a4e9"
```

```
crcTable9 <= (  ,x"701f24f"     ,x"7b70ba3"     ,x"7f01a86"     ,x"746e36a"
 x"0000000"     ,x"6ab3d0c"     ,x"212d4df"     ,x"7c6ced4"     ,x"2e33c16"
,x"3634ad4"     ,x"5c877d8"     ,x"1719e0b"     ,x"4a58400"     ,x"18076c2"
,x"6c695a8"     ,x"06da8a4"     ,x"0db5148"     ,x"1005b7c"     ,x"1b6a290"
,x"5a5df7c"     ,x"30ee270"     ,x"3b81b9c"     ,x"26311a8"     ,x"2d5e844"
,x"40f103f"     ,x"6706c44"     ,x"61dc4e0"     ,x"3c9deeb"     ,x"7703738"
,x"76c5aeb"     ,x"5132690"     ,x"57e8e34"     ,x"0aa943f"     ,x"4137dec"
,x"2c98597"     ,x"0b6f9ec"     ,x"562e3e7"     ,x"50f4b43"     ,x"5b9b2af"
,x"1aacf43"     ,x"3d5b338"     ,x"601a933"     ,x"66c0197"     ,x"6daf87b"
,x"19c1b11"     ,x"27f7c7b"     ,x"3a4764f"     ,x"3128fa3"     ,x"37f2707"
,x"2ff51c5"     ,x"11c36af"     ,x"0c73c9b"     ,x"071c577"     ,x"01c6dd3"
,x"75a8eb9"     ,x"4b9e9d3"     ,x"16df3d8"     ,x"5d41a0b"     ,x"347fca1"
,x"439c46d"     ,x"7daa307"     ,x"20eb90c"     ,x"6b750df"     ,x"024b675"
,x"5930b2e"     ,x"7ec7755"     ,x"7ab6670"     ,x"71d9f9c"     ,x"5816909"
,x"6f041fa"     ,x"48f3d81"     ,x"4c82ca4"     ,x"47ed548"     ,x"6e223dd"
,x"3559e86"     ,x"12ae2fd"     ,x"4fef8f6"     ,x"1db0a34"     ,x"748ec9e"
,x"036d452"     ,x"249a829"     ,x"79db222"     ,x"2b840e0"     ,x"42ba64a"
,x"3383622"     ,x"3e3676a"     ,x"2386d5e"     ,x"28e94b2"     ,x"18e7936"
,x"05b7cf6"     ,x"0802dbe"     ,x"15b278a"     ,x"1edde66"     ,x"2ed33e2"
,x"5fea38a"     ,x"525f2c2"     ,x"0f1e8c9"     ,x"448011a"     ,x"2dbe7b0"
,x"69de95e"     ,x"646b816"     ,x"392a21d"     ,x"72b4bce"     ,x"1b8ad64"
,x"737261d"     ,x"5485a66"     ,x"6377d61"     ,x"681848d"     ,x"41d7218"
,x"4546cc9"     ,x"62b10b2"     ,x"55437b5"     ,x"5e2ce59"     ,x"77e38cc"
,x"1f1b3b5"     ,x"38ecfce"     ,x"65ad5c5"     ,x"0471125"     ,x"6d4f78f"
,x"292f961"     ,x"0ed851a"     ,x"5399f11"     ,x"3245bf1"     ,x"5b7bd5b"
,x"2a42d33"     ,x"1474a59"     ,x"09c406d"     ,x"02ab981"     ,x"0126227"
,x"1c767e7"     ,x"224008d"     ,x"3ff0ab9"     ,x"349f355"     ,x"37128f3"
,x"462b89b"     ,x"781dff1"     ,x"255c5fa"     ,x"6ec2c29"     ,x"07fca83"
                ,x"4e29525"     ,x"1368f2e"     ,x"58f66fd"     ,x"31c8057"
                ,x"4d44177"     ,x"4935052"     ,x"425a9be"     ,x"6b95f2b"
```

```
,x"5da15ff"     ,x"56cec13"     ,x"613cb14"     ,x"6a532f8"
,x"470dabc"     ,x"0c9336f"     ,x"51d2964"     ,x"300ed84"
,x"7139068"     ,x"3aa79bb"     ,x"67e63b0"     ,x"063a750"
,x"2b64f14"     ,x"200b6f8"     ,x"3dbbccc"     ,x"36d4520"
,x"1d505c0"     ,x"163fc2c"     ,x"0b8f618"     ,x"00e0ff4"
,x"1e3d192"     ,x"4c62350"     ,x"112395b"     ,x"5abd088"
,x"2809b46"     ,x"7a56984"     ,x"271738f"     ,x"6c89a5c"
,x"725443a"     ,x"793bdd6"     ,x"7d4acf3"     ,x"762551f"
,x"4460eee"     ,x"4f0f702"     ,x"4b7e627"     ,x"4011fcb"
,x"5ecc1ad"     ,x"155287e"     ,x"4813275"     ,x"1a4c0b7"
,x"68f8b79"     ,x"23662aa"     ,x"7e278a1"     ,x"2c78a63"
,x"32a5405"     ,x"39cade9"     ,x"247a7dd"     ,x"2f15e31"
,x"0491ed1"     ,x"0ffe73d"     ,x"124ed09"     ,x"19214e5"
,x"53790e5"     ,x"55a3841"     ,x"08e224a"     ,x"437cb99"
,x"654da31"     ,x"6397295"     ,x"3ed689e"     ,x"754814d"
,x"3f1054d"     ,x"6251f46"     ,x"648b7e2"     ,x"6fe4e0e"
,x"0924f99"     ,x"5465592"     ,x"52bfd36"     ,x"59d04da"
,x"13880da"     ,x"0e38aee"     ,x"0557302"     ,x"038dba6"
,x"25bca0e"     ,x"380c03a"     ,x"33639d6"     ,x"35b9172"
,x"7fe1572"     ,x"22a0f79"     ,x"693e6aa"     );
,x"49d5fa6"     ,x"14945ad"     ,x"5f0ac7e"
,x"4ab8bf4"     ,x"4ec9ad1"     ,x"45a633d"
,x"7c8c120"     ,x"78fd005"     ,x"73929e9"
,x"26d1e5c"     ,x"7b90457"     ,x"29cf695"
,x"10e5488"     ,x"4da4e83"     ,x"1ffbc41"
,x"0a49bcb"     ,x"17f91ff"     ,x"1c96813"
,x"3c7d11f"     ,x"21cdb2b"     ,x"2aa22c7"
,x"6620e63"     ,x"3b61468"     ,x"70ffdbb"
,x"50144b7"     ,x"0d55ebc"     ,x"46cb76f"
,x"60fa6c7"     ,x"57081c0"     ,x"5c6782c"
```

```
crcTable10  <=  ,x"7d80dad"     ,x"7f33e92"     ,x"004a584"     ,x"7a558ec"
(
                ,x"59b2e44"     ,x"2b467e7"     ,x"247866d"     ,x"568cfce"
 x"0000000"
                ,x"45a30e3"     ,x"3757940"     ,x"38698ca"     ,x"4a9d169"
,x"1c11ea7"
                ,x"4ad74ed"     ,x"1365aa9"     ,x"6c1c1bf"     ,x"6eaf280"
,x"3823d4e"
                ,x"56c6a4a"     ,x"0f7440e"     ,x"700df18"     ,x"72bec27"
,x"24323e9"
                ,x"72f49a3"     ,x"0d8d2b5"     ,x"543fcf1"     ,x"26cb552"
,x"7047a9c"
                ,x"6ee5704"     ,x"119cc12"     ,x"482e256"     ,x"3adabf5"
,x"6c5643b"
                ,x"3a90e71"     ,x"35aeffb"     ,x"475a658"     ,x"1ee881c"
,x"48647d2"
                ,x"26810d6"     ,x"29bf15c"     ,x"5b4b8ff"     ,x"02f96bb"
,x"5475975"
                ,x"02b333f"     ,x"7dca829"     ,x"7f79b16"     ,x"1b1a56a"
,x"78ace57"
                ,x"1ea2d98"     ,x"61db68e"     ,x"63685b1"     ,x"070bbcd"
,x"64bd0f0"
                ,x"327baba"     ,x"45e9567"     ,x"371dcc4"     ,x"2339824"
,x"408f319"
                ,x"2e6a41d"     ,x"59f8bc0"     ,x"2b0c263"     ,x"3f28683"
,x"5c9edbe"
                ,x"0a587f4"     ,x"7521ce2"     ,x"0f3e18a"     ,x"6b5dff6"
,x"08eb4cb"
                ,x"1649953"     ,x"6930245"     ,x"132ff2d"     ,x"774c151"
,x"14faa6c"
                ,x"423c026"     ,x"4d021ac"     ,x"3ff680f"     ,x"537e2b8"
,x"30c8985"
                ,x"5e2de81"     ,x"5113f0b"     ,x"23e76a8"     ,x"4f6fc1f"
,x"2cd9722"
                ,x"7a1fd68"     ,x"056667e"     ,x"07d5541"     ,x"63b6b3d"
,x"697a7c1"
                ,x"660e3cf"     ,x"19778d9"     ,x"1bc4be6"     ,x"7fa759a"
,x"756b966"
                ,x"23ad32c"     ,x"3d45b30"     ,x"4fb1293"     ,x"5b95673"
,x"5159a8f"
                ,x"3fbcd8b"     ,x"2154597"     ,x"53a0c34"     ,x"47848d4"
,x"4d48428"
                ,x"1b8ee62"     ,x"64f7574"     ,x"7792fdd"     ,x"13f11a1"
,x"193dd5d"
                ,x"079f0c5"     ,x"78e6bd3"     ,x"6b8317a"     ,x"0fe0f06"
,x"052c3fa"
                ,x"53ea9b0"     ,x"5cd483a"     ,x"2e20199"     ,x"2bd2cef"
,x"211e013"
                ,x"4ffb717"     ,x"40c569d"     ,x"3231f3e"     ,x"37c3248"
,x"3d0feb4"
                ,x"6bc94fe"     ,x"14b0fe8"     ,x"1603cd7"     ,x"72602ab"
,x"11d6996"
                ,x"77d8a59"     ,x"08a114f"     ,x"0a12270"     ,x"6e71c0c"
,x"0dc7731"
                ,x"5b01d7b"     ,x"2c932a6"     ,x"5e67b05"     ,x"4a43fe5"
,x"29f54d8"
                ,x"47103dc"     ,x"3082c01"     ,x"42765a2"     ,x"5652142"
,x"35e4a7f"
                ,x"6322035"     ,x"1c5bb23"     ,x"664464b"     ,x"0227837"
,x"619130a"
```

```
,x"1e36690"     ,x"1c855af"     ,x"63fceb9"     ,x"7099410"
,x"3a04579"     ,x"48f0cda"     ,x"47ced50"     ,x"353a4f3"
,x"2615bde"     ,x"54e127d"     ,x"5bdf3f7"     ,x"292ba54"
,x"0acccfc"     ,x"70d3194"     ,x"0faaa82"     ,x"0d199bd"
,x"16dd25b"     ,x"6cc2f33"     ,x"13bb425"     ,x"110871a"
,x"32ef1b2"     ,x"401b811"     ,x"37897cc"     ,x"457de6f"
,x"2efef15"     ,x"5c0a6b6"     ,x"2b9896b"     ,x"596c0c8"
,x"7a8b660"     ,x"783855f"     ,x"0741e49"     ,x"7d5e321"
,x"669a8c7"     ,x"6429bf8"     ,x"1b500ee"     ,x"614fd86"
,x"42a8b2e"     ,x"305c28d"     ,x"3f62307"     ,x"4d96aa4"
,x"5eb9589"     ,x"2c4dc2a"     ,x"2373da0"     ,x"5187403"
,x"51cd187"     ,x"087ffc3"     ,x"77064d5"     ,x"75b57ea"
,x"4ddcf20"     ,x"146e164"     ,x"6b17a72"     ,x"69a494d"
,x"69eecc9"     ,x"16977df"     ,x"4f2599b"     ,x"3dd1038"
,x"75ff26e"     ,x"0a86978"     ,x"533473c"     ,x"21c0e9f"
,x"218ab1b"     ,x"2eb4a91"     ,x"5c40332"     ,x"05f2d76"
,x"3d9b5bc"     ,x"32a5436"     ,x"4051d95"     ,x"19e33d1"
,x"19a9655"     ,x"66d0d43"     ,x"6463e7c"     );
,x"05b88f2"     ,x"7ac13e4"     ,x"78720db"
,x"2961fd0"     ,x"5ef300d"     ,x"2c079ae"
,x"3570177"     ,x"42e2eaa"     ,x"3016709"
,x"114229e"     ,x"6e3b988"     ,x"14244e0"
,x"0d53c39"     ,x"722a72f"     ,x"0835a47"
,x"592654c"     ,x"56184c6"     ,x"24ecd65"
,x"4537beb"     ,x"4a09a61"     ,x"38fd3c2"
,x"6105802"     ,x"1e7c314"     ,x"1ccf02b"
,x"7d146a5"     ,x"026ddb3"     ,x"00dee8c"
,x"38b7646"     ,x"265fe5a"     ,x"54ab7f9"
,x"24a68e1"     ,x"3a4e0fd"     ,x"48ba95e"
,x"0094b08"     ,x"7fed01e"     ,x"6c88ab7"
```

```
crcTable11  <=  ,x"7aa82ac"     ,x"0aadc13"     ,x"6cb8eff"     ,x"7285a02"
(
                ,x"4ae9faf"     ,x"0df0431"     ,x"5cf93fc"     ,x"1be0862"
 x"0000000"
                ,x"2d37d8b"     ,x"6a2e615"     ,x"3b271d8"     ,x"7c3ea46"
,x"67de224"
                ,x"10864b9"     ,x"5a6fb16"     ,x"3c7a9fa"     ,x"4c7f745"
,x"579ff27"
                ,x"775869d"     ,x"3db1932"     ,x"5ba4bde"     ,x"2ba1561"
,x"3041d03"
                ,x"4719b9e"     ,x"210c972"     ,x"6be56dd"     ,x"2cfcd43"
,x"371c521"
                ,x"20c79ba"     ,x"46d2b56"     ,x"0c3b4f9"     ,x"4b22f67"
,x"50c2705"
                ,x"279a198"     ,x"7693655"     ,x"318adcb"     ,x"7b63264"
,x"6083a06"
                ,x"40443bc"     ,x"114d471"     ,x"5654fef"     ,x"1cbd040"
,x"075d822"
                ,x"7005ebf"     ,x"1610c53"     ,x"66152ec"     ,x"42192e4"
,x"6e38a42"
                ,x"17dbc9b"     ,x"71cee77"     ,x"01cb0c8"     ,x"25c70c0"
,x"09e6866"
                ,x"7ebeefb"     ,x"418f374"     ,x"06968ea"     ,x"1586dc3"
,x"39a7565"
                ,x"1960cdf"     ,x"2651150"     ,x"6148ace"     ,x"7258fe7"
,x"5e79741"
                ,x"29211dc"     ,x"4f34330"     ,x"51097cd"     ,x"75057c5"
,x"5924f63"
                ,x"4eff3f8"     ,x"28ea114"     ,x"36d75e9"     ,x"12db5e1"
,x"3efad47"
                ,x"49a2bda"     ,x"18abc17"     ,x"5fb2789"     ,x"229a8e2"
,x"0ebb044"
                ,x"2e7c9fe"     ,x"7f75e33"     ,x"386c5ad"     ,x"4544ac6"
,x"6965260"
                ,x"1e3d4fd"     ,x"7828611"     ,x"082d8ae"     ,x"2c218a6"
,x"4452feb"
                ,x"79e36d9"     ,x"1ff6435"     ,x"6ff3a8a"     ,x"4bffa82"
,x"238cdcf"
                ,x"54d4b52"     ,x"2fb7936"     ,x"68ae2a8"     ,x"7bbe781"
,x"13cd0cc"
                ,x"330a976"     ,x"4869b12"     ,x"0f7008c"     ,x"1c605a5"
,x"74132e8"
                ,x"034b475"     ,x"655e699"     ,x"3f31d8f"     ,x"1b3dd87"
,x"734eaca"
                ,x"6495651"     ,x"02804bd"     ,x"58effab"     ,x"7ce3fa3"
,x"14908ee"
                ,x"63c8e73"     ,x"32c19be"     ,x"75d8220"     ,x"4ca22a0"
,x"24d15ed"
                ,x"0416c57"     ,x"551fb9a"     ,x"1206004"     ,x"2b7c084"
,x"430f7c9"
                ,x"3457154"     ,x"52423b8"     ,x"2247d07"     ,x"064bd0f"
,x"2a6a5a9"
                ,x"5389370"     ,x"359c19c"     ,x"4599f23"     ,x"6195f2b"
,x"4db478d"
                ,x"3aec110"     ,x"05ddc9f"     ,x"42c4701"     ,x"51d4228"
,x"7df5a8e"
                ,x"5d32334"     ,x"6203ebb"     ,x"251a525"     ,x"360a00c"
,x"1a2b8aa"
                ,x"6d73e37"     ,x"0b66cdb"     ,x"155b826"     ,x"315782e"
,x"1d76088"
```

```
,x"5689a0a"     ,x"268c4b5"     ,x"4099659"     ,x"1af6d4f"
,x"66c8709"     ,x"21d1c97"     ,x"70d8b5a"     ,x"37c10c4"
,x"011652d"     ,x"460feb3"     ,x"170697e"     ,x"501f2e0"
,x"687374d"     ,x"764e3b0"     ,x"105b15c"     ,x"605efe3"
,x"0fad569"     ,x"1190194"     ,x"7785378"     ,x"0780dc7"
,x"3fec86a"     ,x"78f53f4"     ,x"47c4e7b"     ,x"00dd5e5"
,x"5832a4e"     ,x"1f2b1d0"     ,x"201ac5f"     ,x"67037c1"
,x"5f6f26c"     ,x"2f6acd3"     ,x"497fe3f"     ,x"5742ac2"
,x"38b1048"     ,x"48b4ef7"     ,x"2ea1c1b"     ,x"309c8e6"
,x"08f0d4b"     ,x"4fe96d5"     ,x"1ee0118"     ,x"59f9a86"
,x"6f2ef6f"     ,x"28374f1"     ,x"793e33c"     ,x"3e278a2"
,x"529f65d"     ,x"18769f2"     ,x"7e63b1e"     ,x"0e665a1"
,x"3541479"     ,x"7fa8bd6"     ,x"19bd93a"     ,x"69b8785"
,x"050097a"     ,x"6315b96"     ,x"29fc439"     ,x"6ee5fa7"
,x"62deb5e"     ,x"04cb9b2"     ,x"4e2261d"     ,x"093bd83"
,x"658337c"     ,x"348a4b1"     ,x"7393f2f"     ,x"397a080"
,x"025d158"     ,x"5354695"     ,x"144dd0b"     ,x"5ea42a4"
,x"321cc5b"     ,x"5409eb7"     ,x"240c008"     );
,x"55c2e7f"     ,x"33d7c93"     ,x"43d222c"
,x"3ca7c1f"     ,x"0396190"     ,x"448fa0e"
,x"5b79e3b"     ,x"64483b4"     ,x"235182a"
,x"6b38338"     ,x"0d2d1d4"     ,x"1310529"
,x"0ce611c"     ,x"6af33f0"     ,x"74ce70d"
,x"0bbb93e"     ,x"5ab2ef3"     ,x"1dab56d"
,x"6c65b1a"     ,x"3d6ccd7"     ,x"7a75749"
,x"5c24619"     ,x"3a314f5"     ,x"4a34a4a"
,x"3bfa43d"     ,x"5def6d1"     ,x"2dea86e"
,x"16cd9b6"     ,x"6daebd2"     ,x"2ab704c"
,x"7113b92"     ,x"0a709f6"     ,x"4d69268"
,x"4152691"     ,x"274747d"     ,x"7d28f6b"
```

```
crcTable12  <=  ,x"26da27f"     ,x"4db44fe"     ,x"6825412"     ,x"034b293"
(
                ,x"33181c6"     ,x"5876747"     ,x"7de77ab"     ,x"168912a"
 x"0000000"
                ,x"40ea1f3"     ,x"663038c"     ,x"14a9d84"     ,x"28cf5e1"
,x"15c23b9"
                ,x"552824a"     ,x"73f2035"     ,x"016be3d"     ,x"3d0d658"
,x"2b84772"
                ,x"6b6e681"     ,x"19f7889"     ,x"3f2daf6"     ,x"5443c77"
,x"3e464cb"
                ,x"7eac538"     ,x"0c35b30"     ,x"2aef94f"     ,x"4181fce"
,x"5708ee4"
                ,x"17e2f17"     ,x"3273ffb"     ,x"591d97a"     ,x"7fc7b05"
,x"42cad5d"
                ,x"0220cae"     ,x"27b1c42"     ,x"4cdfac3"     ,x"6a058bc"
,x"7c8c996"
                ,x"3c66865"     ,x"4eff66d"     ,x"7299e08"     ,x"33ef112"
,x"694ea2f"
                ,x"29a4bdc"     ,x"5b3d5d4"     ,x"675bdb1"     ,x"262d2ab"
,x"36326a7"
                ,x"76d8754"     ,x"657b11f"     ,x"0e1579e"     ,x"186b660"
,x"23f051e"
                ,x"631a4ed"     ,x"70b92a6"     ,x"1bd7427"     ,x"0da95d9"
,x"1db61d5"
                ,x"5d5c026"     ,x"2fc5e2e"     ,x"25910ec"     ,x"64e7ff6"
,x"087426c"
                ,x"489e39f"     ,x"3a07d97"     ,x"3053355"     ,x"7125c4f"
,x"613a843"
                ,x"21d09b0"     ,x"044195c"     ,x"6f2ffdd"     ,x"4f63884"
,x"74f8bfa"
                ,x"3412a09"     ,x"1183ae5"     ,x"7aedc64"     ,x"5aa1b3d"
,x"4abef31"
                ,x"0a54ec2"     ,x"78cd0ca"     ,x"44ab8af"     ,x"05dd7b5"
,x"5f7cc88"
                ,x"1f96d7b"     ,x"6d0f373"     ,x"5169b16"     ,x"101f40c"
,x"6c64d4e"
                ,x"2c8ecbd"     ,x"53497b8"     ,x"3827139"     ,x"2e590c7"
,x"79a6ef7"
                ,x"394cf04"     ,x"468b401"     ,x"2de5280"     ,x"3b9b37e"
,x"47e0a3c"
                ,x"070abcf"     ,x"75935c7"     ,x"13a364b"     ,x"52d5951"
,x"5222985"
                ,x"12c8876"     ,x"605167e"     ,x"06615f2"     ,x"4717ae8"
,x"3b6c3aa"
                ,x"7b86259"     ,x"5e172b5"     ,x"3579434"     ,x"7951e23"
,x"2eae013"
                ,x"6e441e0"     ,x"4bd510c"     ,x"20bb78d"     ,x"6c93d9a"
,x"10e84d8"
                ,x"500252b"     ,x"229bb23"     ,x"1efd346"     ,x"5f8bc5c"
,x"052a761"
                ,x"45c0692"     ,x"375989a"     ,x"0b3f0ff"     ,x"4a49fe5"
,x"5a56be9"
                ,x"1abca1a"     ,x"091fc51"     ,x"6271ad0"     ,x"740fb2e"
,x"4f94850"
                ,x"0f7e9a3"     ,x"1cddfe8"     ,x"77b3969"     ,x"61cd897"
,x"71d2c9b"
                ,x"3138d68"     ,x"43a1360"     ,x"49f5da2"     ,x"08832b8"
,x"6410f22"
                ,x"24faed1"     ,x"56630d9"     ,x"5c37e1b"     ,x"1d41101"
,x"0d5e50d"

,x"189c6b4"
```

```
,x"23075ca"     ,x"486934b"     ,x"6df83a7"     ,x"0696526"
,x"36c5673"     ,x"5dab0f2"     ,x"783a01e"     ,x"135469f"
,x"69b9afb"     ,x"63ed439"     ,x"1174a31"     ,x"2d12254"
,x"7c7b942"     ,x"762f780"     ,x"04b6988"     ,x"38d01ed"
,x"423dd89"     ,x"2953b08"     ,x"3af0d43"     ,x"519ebc2"
,x"57ffe30"     ,x"3c918b1"     ,x"2f32efa"     ,x"445c87b"
,x"3eb141f"     ,x"02d7c7a"     ,x"704e272"     ,x"7a1acb0"
,x"2b737a6"     ,x"1715fc3"     ,x"658c1cb"     ,x"6fd8f09"
,x"153536d"     ,x"7e5b5ec"     ,x"5bca500"     ,x"30a4381"
,x"00f70d4"     ,x"6b99655"     ,x"4e086b9"     ,x"2566038"
,x"73050e1"     ,x"55df29e"     ,x"2746c96"     ,x"1b204f3"
,x"66c7358"     ,x"401d127"     ,x"3284f2f"     ,x"0ee274a"
,x"5881793"     ,x"2a1899b"     ,x"0cc2be4"     ,x"67acd65"
,x"4d4342a"     ,x"3fdaa22"     ,x"190085d"     ,x"726eedc"
,x"240de05"     ,x"019cee9"     ,x"6af2868"     ,x"4c28a17"
,x"31cfdbc"     ,x"145ed50"     ,x"7f30bd1"     ,x"59ea9ae"
,x"0f89977"     ,x"7d1077f"     ,x"4176f1a"     );
,x"1a4bace"     ,x"68d24c6"     ,x"54b4ca3"
,x"4537646"     ,x"569400d"     ,x"3dfa68c"
,x"50f55ff"     ,x"43563b4"     ,x"2838535"
,x"6eb3134"     ,x"1c2af3c"     ,x"167e1fe"
,x"7b7128d"     ,x"09e8c85"     ,x"03bc247"
,x"123f8a2"     ,x"37ae84e"     ,x"5cc0ecf"
,x"07fdb1b"     ,x"226cbf7"     ,x"4902d76"
,x"39bbfd0"     ,x"4b221d8"     ,x"77449bd"
,x"2c79c69"     ,x"5ee0261"     ,x"6286a04"
,x"1f61daf"     ,x"60a66aa"     ,x"0bc802b"
,x"0aa3e16"     ,x"7564513"     ,x"1e0a392"
,x"34e5add"     ,x"467c4d5"     ,x"204c759"
,x"2127964"     ,x"53be76c"     ,x"358e4e0"
```

```
crcTable13  <=  ,x"1a38b42"     ,x"3471684"     ,x"46ab0ce"     ,x"68e2d08"
(
                ,x"5ad7ef6"     ,x"749e330"     ,x"064457a"     ,x"280d8bc"
 x"0000000"
                ,x"37b4dc1"     ,x"2d8c683"     ,x"6cac0c7"     ,x"711fd0f"
,x"40ef5b4"
                ,x"775b875"     ,x"6d63337"     ,x"2c43573"     ,x"31f08bb"
,x"19fd007"
                ,x"2e49dc6"     ,x"6f69b82"     ,x"75510c0"     ,x"5b18d06"
,x"59125b3"
                ,x"6ea6872"     ,x"2f86e36"     ,x"35be574"     ,x"1bf78b2"
,x"33fa00e"
                ,x"044edcf"     ,x"7694b85"     ,x"58dd643"     ,x"42e5d01"
,x"73155ba"
                ,x"44a187b"     ,x"367be31"     ,x"18323f7"     ,x"020a8b5"
,x"2a07009"
                ,x"1db3dc8"     ,x"5c93b8c"     ,x"4120644"     ,x"46f0c6b"
,x"6ae85bd"
                ,x"5d5c87c"     ,x"1c7ce38"     ,x"01cf3f0"     ,x"061f9df"
,x"67f401c"
                ,x"5040ddd"     ,x"456eb8b"     ,x"6b2764d"     ,x"5f0dc6c"
,x"271b5a8"
                ,x"10af869"     ,x"0581e3f"     ,x"2bc83f9"     ,x"1fe29d8"
,x"7e0901b"
                ,x"49bddda"     ,x"089db9e"     ,x"72da64a"     ,x"750ac65"
,x"3ee65af"
                ,x"095286e"     ,x"4872e2a"     ,x"32353fe"     ,x"35e59d1"
,x"540e012"
                ,x"63badd3"     ,x"1160b99"     ,x"3f2965f"     ,x"6cf7c62"
,x"14e15a6"
                ,x"2355867"     ,x"518fe2d"     ,x"7fc63eb"     ,x"2c189d6"
,x"4df3015"
                ,x"7a47dd4"     ,x"3b67b90"     ,x"26d4658"     ,x"2104c77"
,x"0d1c5a1"
                ,x"3aa8860"     ,x"7b88e24"     ,x"663b3ec"     ,x"61eb9c3"
,x"57cbb57"
                ,x"607f696"     ,x"229ab97"     ,x"0cd3651"     ,x"38f9c70"
,x"1724ee3"
                ,x"2090322"     ,x"6275e23"     ,x"4c3c3e5"     ,x"78169c4"
,x"4e36b50"
                ,x"7982691"     ,x"38a20d5"     ,x"152e656"     ,x"12fec79"
,x"0ed9ee4"
                ,x"396d325"     ,x"784d561"     ,x"55c13e2"     ,x"52119cd"
,x"6431b59"
                ,x"5385698"     ,x"215f0d2"     ,x"0f16d14"     ,x"0b03c7e"
,x"24deeed"
                ,x"136a32c"     ,x"61b0566"     ,x"4ff98a0"     ,x"4bec9ca"
,x"7dccb5e"
                ,x"4a7869f"     ,x"0b580db"     ,x"16ebd13"     ,x"113b73c"
,x"3d23eea"
                ,x"0a9732b"     ,x"4bb756f"     ,x"56048a7"     ,x"51d4288"
,x"303fb4b"
                ,x"078b68a"     ,x"12a50dc"     ,x"3cecd1a"     ,x"08c673b"
,x"70d0eff"
                ,x"476433e"     ,x"524a568"     ,x"7c038ae"     ,x"482928f"
,x"29c2b4c"
                ,x"1e7668d"     ,x"5f560c9"     ,x"2511d1d"     ,x"22c1732"
,x"692def8"
                ,x"5e99339"     ,x"1fb957d"     ,x"65fe8a9"     ,x"622e286"
,x"03c5b45"
,x"432aef1"
```

```
,x"3b3c735"     ,x"1575af3"     ,x"67afcb9"     ,x"49e617f"
,x"7bd3281"     ,x"559af47"     ,x"274090d"     ,x"09094cb"
,x"76cf720"     ,x"0c88af4"     ,x"4da8cb0"     ,x"501b178"
,x"3620294"     ,x"4c67f40"     ,x"0d47904"     ,x"10f44cc"
,x"6f32727"     ,x"417bae1"     ,x"5455cb7"     ,x"7a1c171"
,x"2fdd293"     ,x"0194f55"     ,x"14ba903"     ,x"3af34c5"
,x"453572e"     ,x"5886ae6"     ,x"19a6ca2"     ,x"63e1176"
,x"05da29a"     ,x"1869f52"     ,x"5949916"     ,x"230e4c2"
,x"5cc8729"     ,x"7281aef"     ,x"005bca5"     ,x"2e12163"
,x"1c2729d"     ,x"326ef5b"     ,x"40b4911"     ,x"6efd4d7"
,x"71441aa"     ,x"6b7cae8"     ,x"2a5ccac"     ,x"37ef164"
,x"31ab41e"     ,x"2b93f5c"     ,x"6ab3918"     ,x"77004d0"
,x"68b91ad"     ,x"29997e9"     ,x"33a1cab"     ,x"1de816d"
,x"2856419"     ,x"697625d"     ,x"734e91f"     ,x"5d074d9"
,x"42be1a4"     ,x"30647ee"     ,x"1e2da28"     ,x"041516a"
,x"0251410"     ,x"708b25a"     ,x"5ec2f9c"     ,x"44fa4de"
,x"5b431a3"     ,x"1a637e7"     ,x"07d0a2f"     );
,x"1bac417"     ,x"5a8c253"     ,x"473ff9b"
,x"16b01b6"     ,x"039e7e0"     ,x"2dd7a26"
,x"565f402"     ,x"4371254"     ,x"6d38f92"
,x"0f4d1b1"     ,x"4e6d7f5"     ,x"342aa21"
,x"4fa2405"     ,x"0e82241"     ,x"74c5f95"
,x"254a1b8"     ,x"57907f2"     ,x"79d9a34"
,x"65a540c"     ,x"177f246"     ,x"3936f80"
,x"3cb71bf"     ,x"7d977fb"     ,x"6024a33"
,x"7c5840b"     ,x"3d7824f"     ,x"20cbf87"
,x"268fafd"     ,x"646a7fc"     ,x"4a23a3a"
,x"6660f49"     ,x"2485248"     ,x"0accf8e"
,x"3f72afa"     ,x"7e52cbe"     ,x"53dea3d"
,x"7f9df4e"     ,x"3ebd90a"     ,x"1331f89"
```

```
crcTable14  <=  ,x"6bf278f"     ,x"0cb5db3"     ,x"4644b34"     ,x"5a192a4"
(
                ,x"36466a6"     ,x"6c8cd4c"     ,x"1bf0a1d"     ,x"413a1f7"
 x"0000000"
                ,x"7534f64"     ,x"2ffe48e"     ,x"58823df"     ,x"0248835"
,x"43729c2"
                ,x"440c301"     ,x"724a5a7"     ,x"38bb320"     ,x"5ffc91c"
,x"1ec68eb"
                ,x"077eac3"     ,x"3138c65"     ,x"7bc9ae2"     ,x"1c8e0de"
,x"5db4129"
                ,x"5acabea"     ,x"103bd6d"     ,x"267dbcb"     ,x"7cb7021"
,x"3d8d1d6"
                ,x"19b8228"     ,x"53494af"     ,x"650f209"     ,x"3fc59e3"
,x"7eff814"
                ,x"79812d7"     ,x"0efd586"     ,x"5437e6c"     ,x"62718ca"
,x"234b93d"
                ,x"3af3b15"     ,x"4d8fc44"     ,x"17457ae"     ,x"2103108"
,x"60390ff"
                ,x"6747a3c"     ,x"2db6cbb"     ,x"4af1687"     ,x"2077ada"
,x"7b1a3ac"
                ,x"24353fe"     ,x"6ec4579"     ,x"0983f45"     ,x"6305318"
,x"3868a6e"
                ,x"3f160ad"     ,x"3370450"     ,x"69bafba"     ,x"3eb1231"
,x"65dcb47"
                ,x"7c6496f"     ,x"7002d92"     ,x"2ac8678"     ,x"7dc3bf3"
,x"26ae285"
                ,x"21d0846"     ,x"6b21ec1"     ,x"777c751"     ,x"1dfab0c"
,x"469727a"
                ,x"62a2184"     ,x"2853703"     ,x"340ee93"     ,x"5e882ce"
,x"05e5bb8"
                ,x"029b17b"     ,x"75e762a"     ,x"2f2ddc0"     ,x"033c3e7"
,x"5851a91"
                ,x"41e98b9"     ,x"3695fe8"     ,x"6c5f402"     ,x"404ea25"
,x"1b23353"
                ,x"1c5d990"     ,x"56acf17"     ,x"31eb52b"     ,x"5b6d976"
,x"6e17c37"
                ,x"5f2f052"     ,x"15de6d5"     ,x"7299ce9"     ,x"181f0b4"
,x"2d655f5"
                ,x"2a1bf36"     ,x"486a7fc"     ,x"12a0c16"     ,x"45ab19d"
,x"70d14dc"
                ,x"69696f4"     ,x"0b18e3e"     ,x"51d25d4"     ,x"06d985f"
,x"33a3d1e"
                ,x"34dd7dd"     ,x"7e2c15a"     ,x"0c664fd"     ,x"66e08a0"
,x"539ade1"
                ,x"77afe1f"     ,x"3d5e898"     ,x"4f14d3f"     ,x"2592162"
,x"10e8423"
                ,x"1796ee0"     ,x"60ea9b1"     ,x"3a2025b"     ,x"782604b"
,x"4d5c50a"
                ,x"54e4722"     ,x"2398073"     ,x"7952b99"     ,x"3b54989"
,x"0e2ecc8"
                ,x"095060b"     ,x"43a108c"     ,x"24e6ab0"     ,x"4e606ed"
,x"150df9b"
                ,x"4a22fc9"     ,x"00d394e"     ,x"6794372"     ,x"0d12f2f"
,x"567f659"
                ,x"5101c9a"     ,x"5d67867"     ,x"07ad38d"     ,x"50a6e06"
,x"0bcb770"
                ,x"1273558"     ,x"1e151a5"     ,x"44dfa4f"     ,x"13d47c4"
,x"48b9eb2"
                ,x"4fc7471"     ,x"05362f6"     ,x"196bb66"     ,x"73ed73b"
,x"2880e4d"
```

```
,x"309fef9"     ,x"57d84c5"     ,x"1d29242"     ,x"6f637e5"
,x"6d2bfd0"     ,x"37e143a"     ,x"409d36b"     ,x"1a57881"
,x"2e59612"     ,x"7493df8"     ,x"03efaa9"     ,x"5925143"
,x"357a541"     ,x"2927cd1"     ,x"63d6a56"     ,x"049106a"
,x"7608c83"     ,x"6a55513"     ,x"20a4394"     ,x"47e39a8"
,x"2bbcdaa"     ,x"7176640"     ,x"7d102bd"     ,x"27da957"
,x"68ce468"     ,x"3204f82"     ,x"3e62b7f"     ,x"64a8095"
,x"08f7497"     ,x"6fb0eab"     ,x"254182c"     ,x"391c1bc"
,x"4b85d55"     ,x"2cc2769"     ,x"66331ee"     ,x"7a6e87e"
,x"1631c7c"     ,x"4cfb796"     ,x"3b870c7"     ,x"614db2d"
,x"55435be"     ,x"0f89e54"     ,x"78f5905"     ,x"223f2ef"
,x"647b9db"     ,x"523df7d"     ,x"18cc9fa"     ,x"7f8b3c6"
,x"2709019"     ,x"114f6bf"     ,x"5bbe038"     ,x"3cf9a04"
,x"7abd130"     ,x"304c7b7"     ,x"060a111"     ,x"5cc0afb"
,x"39cf8f2"     ,x"733ee75"     ,x"45788d3"     ,x"1fb2339"
,x"59f680d"     ,x"2e8af5c"     ,x"74404b6"     ,x"4206210"
,x"1a841cf"     ,x"6df869e"     ,x"3732d74"     ,x"0174bd2"
,x"47300e6"     ,x"0dc1661"     ,x"6a86c5d"     );
,x"0442924"     ,x"4eb3fa3"     ,x"29f459f"
,x"1f61a77"     ,x"1307e8a"     ,x"49cd560"
,x"5c133b5"     ,x"5075748"     ,x"0abfca2"
,x"01a729c"     ,x"4b5641b"     ,x"570bd8b"
,x"42d5b5e"     ,x"0824dd9"     ,x"1479449"
,x"22ecba1"     ,x"5590cf0"     ,x"0f5a71a"
,x"619e263"     ,x"16e2532"     ,x"4c28ed8"
,x"3c2a34a"     ,x"76db5cd"     ,x"119cff1"
,x"7f58a88"     ,x"35a9c0f"     ,x"52ee633"
,x"0a6c5ec"     ,x"681dd26"     ,x"32d76cc"
,x"491ec2e"     ,x"2b6f4e4"     ,x"71a5f0e"
,x"14aad07"     ,x"5e5bb80"     ,x"2c11e27"
```

```
crcTable15  <=  ,x"05c71cb"     ,x"0b8e396"     ,x"1955571"     ,x"171c72c"
(
                ,x"7fc9e78"     ,x"7180c25"     ,x"635bac2"     ,x"6d1289f"
 x"0000000"
                ,x"6277654"     ,x"67b079f"     ,x"3534205"     ,x"7b22325"
,x"7a0efb3"
                ,x"18799e7"     ,x"1dbe82c"     ,x"4f3adb6"     ,x"012cc96"
,x"6c3e409"
                ,x"0e4925d"     ,x"5ccd7c7"     ,x"590a60c"     ,x"5743451"
,x"1630bba"
                ,x"7447dee"     ,x"26c3874"     ,x"23049bf"     ,x"2d4dbe2"
,x"405f37d"
                ,x"2228529"     ,x"30f33ce"     ,x"3eba193"     ,x"3b7d058"
,x"3a51cce"
                ,x"5826a9a"     ,x"4afdc7d"     ,x"44b4e20"     ,x"4173feb"
,x"2c61774"
                ,x"4e16120"     ,x"1c924ba"     ,x"528459a"     ,x"21b94e1"
,x"566f8c7"
                ,x"3418e93"     ,x"669cb09"     ,x"288aa29"     ,x"5bb7b52"
,x"189dd95"
                ,x"7aeabc1"     ,x"70ac0b3"     ,x"7ee52ee"     ,x"4d870e8"
,x"6293226"
                ,x"00e4472"     ,x"0aa2f00"     ,x"04ebd5d"     ,x"3789f5b"
,x"74a399c"
                ,x"16d4fc8"     ,x"4450a52"     ,x"12db6e7"     ,x"61e679c"
,x"0ead62f"
                ,x"6cda07b"     ,x"3e5e5e1"     ,x"68d5954"     ,x"1be882f"
,x"58c2ee8"
                ,x"3ab58bc"     ,x"286ee5b"     ,x"2627c06"     ,x"0dd8395"
,x"22cc15b"
                ,x"40bb70f"     ,x"52601e8"     ,x"5c293b5"     ,x"77d6c26"
,x"34fcae1"
                ,x"568bcb5"     ,x"040f92f"     ,x"4a1980f"     ,x"3924974"
,x"4ef2552"
                ,x"2c85306"     ,x"7e0169c"     ,x"30177bc"     ,x"432a6c7"
,x"313bb2a"
                ,x"534cd7e"     ,x"6831d26"     ,x"6678f7b"     ,x"551ad7d"
,x"4b35499"
                ,x"29422cd"     ,x"123f295"     ,x"1c760c8"     ,x"2f142ce"
,x"5d05f23"
                ,x"3f72977"     ,x"6df6ced"     ,x"0a46b72"     ,x"797ba09"
,x"270b090"
                ,x"457c6c4"     ,x"17f835e"     ,x"70484c1"     ,x"03755ba"
,x"7164857"
                ,x"1313e03"     ,x"01c88e4"     ,x"0f81ab9"     ,x"1545e00"
,x"0b6a7e4"
                ,x"691d1b0"     ,x"7bc6757"     ,x"758f50a"     ,x"6f4b1b3"
,x"1d5ac5e"
                ,x"7f2da0a"     ,x"2da9f90"     ,x"63bfeb0"     ,x"1082fcb"
,x"67543ed"
                ,x"05235b9"     ,x"57a7023"     ,x"19b1103"     ,x"6a8c078"
,x"29a66bf"
                ,x"4bd10eb"     ,x"4197b99"     ,x"4fde9c4"     ,x"7cbcbc2"
,x"53a890c"
                ,x"31dff58"     ,x"3b9942a"     ,x"35d0677"     ,x"06b2471"
,x"45982b6"
                ,x"27ef4e2"     ,x"756b178"     ,x"23e0dcd"     ,x"50ddcb6"
,x"3f96d05"
                ,x"5de1b51"     ,x"0f65ecb"     ,x"59ee27e"     ,x"2ad3305"
,x"69f95c2"

,x"13f7a71"
```

```
,x"3ce38bf"     ,x"32aaae2"     ,x"2071c05"     ,x"2e38e58"
,x"46ed70c"     ,x"48a4551"     ,x"5a7f3b6"     ,x"54361eb"
,x"081f25e"     ,x"5e94eeb"     ,x"0c10b71"     ,x"4206a51"
,x"7211ded"     ,x"249a158"     ,x"761e4c2"     ,x"38085e2"
,x"6421657"     ,x"6a6840a"     ,x"602ef78"     ,x"6e67d25"
,x"1e2f9e4"     ,x"1066bb9"     ,x"1a200cb"     ,x"1469296"
,x"4840123"     ,x"0656003"     ,x"54d2599"     ,x"025992c"
,x"324ee90"     ,x"7c58fb0"     ,x"2edca2a"     ,x"785769f"
,x"247e52a"     ,x"2a37777"     ,x"38ec190"     ,x"36a53cd"
,x"5e70a99"     ,x"50398c4"     ,x"42e2e23"     ,x"4cabc7e"
,x"43ce2b5"     ,x"460937e"     ,x"148d6e4"     ,x"5a9b7c4"
,x"39c0d06"     ,x"3c07ccd"     ,x"6e83957"     ,x"2095877"
,x"2ff06bc"     ,x"7d74326"     ,x"78b32ed"     ,x"76fa0b0"
,x"55fe90f"     ,x"077ac95"     ,x"02bdd5e"     ,x"0cf4f03"
,x"03911c8"     ,x"114a72f"     ,x"1f03572"     ,x"1ac44b9"
,x"799fe7b"     ,x"6b4489c"     ,x"650dac1"     ,x"60cab0a"
,x"6faf5c1"     ,x"3d2b05b"     ,x"733d17b"     );
,x"15a1a72"     ,x"4725fe8"     ,x"0933ec8"
,x"5b53f20"     ,x"5115452"     ,x"5f5c60f"     end
                                                Behavioral;
,x"215d093"     ,x"2b1bbe1"     ,x"25529bc"
,x"376db29"     ,x"65e9eb3"     ,x"3362206"
,x"4d6349a"     ,x"1fe7100"     ,x"496cdb5"
,x"1b0cc5d"     ,x"09d7aba"     ,x"079e8e7"
,x"61023ee"     ,x"73d9509"     ,x"7d90754"
,x"7732854"     ,x"25b6dce"     ,x"6ba0cee"
,x"0d3c7e7"     ,x"5fb827d"     ,x"11ae35d"
,x"72f599f"     ,x"49889c7"     ,x"47c1b9a"
,x"08fb62c"     ,x"3386674"     ,x"3dcf429"
,x"1ecbd96"     ,x"4c4f80c"     ,x"2bfff93"
,x"64c5225"     ,x"36417bf"     ,x"51f1020"
```

APPENDIX – B: VHDL TEST BENCH (Simulation) CODE

```
28 LIBRARY ieee;
29 USE ieee.std_logic_1164.ALL;
30
34
35 ENTITY crc32_slicing16_tb IS
36 END crc32_slicing16_tb;
37
38 ARCHITECTURE behavior OF crc32_slicing16_tb IS
39
40 -- Component Declaration for the Unit Under Test (UUT)
41
42 COMPONENT crc32_slicing16
43 PORT(
44 clk : IN std_logic;
45 reset : IN std_logic;
46 start : IN std_logic;
47 EOF : IN std_logic;
48 DataIn : IN std_logic_vector(127 downto 0);
49 crc_32 : OUT std_logic_vector(31 downto 0)
50 );
51 END COMPONENT;
52
53
54 --Inputs
55 signal clk : std_logic := '0';
56 signal reset : std_logic := '0';
57 signal start : std_logic := '0';
58 signal EOF : std_logic := '0';
59 signal DataIn : std_logic_vector(127 downto 0) := (others => '0');
60
61 --Outputs
62 signal crc_32 : std_logic_vector(31 downto 0);
63
64 -- Clock period definitions
65 constant clk_period : time := 1.25 ns;
66
67 BEGIN
68
69 -- Instantiate the Unit Under Test (UUT)
70 uut: crc32_slicing16 PORT MAP (
71 clk => clk,
72 reset => reset,
73 start => start,
74 EOF => EOF,
```

```
75 DataIn => DataIn,
76 crc_32 => crc_32
77 );
78
79 -- Clock process definitions
80 clk_process :process
81 begin
82 clk <= '0';
83 wait for clk_period/2;
84 clk <= '1';
85 wait for clk_period/2;
86 end process;
87
88
89 -- Stimulus process
90 stim_proc: process
91 begin
92 -- hold reset state for 100 ns.
93 wait for 100 ns;
94
95 wait for clk_period*10;
96
97 -- insert stimulus here
98
99
100 -- first 16 bytes
101 DataIn <= x"01020304050607080910111213141516";
102 start <= '1';
103 wait for clk_period;
104 start <= '0';
105
106
107 -- second 16 bytes
108 DataIn <= x"0A0B0C0D0E0F01020304050607080901";
109 start <= '1';
110 wait for clk_period;
111 start <= '0';
112
113 EOF <= '1';
114 wait for clk_period;
115 EOF <= '0';
116
117 wait for clk_period;
118 reset <= '1';
119
120 wait;
121 end process;
122
123 END;
124
```

APPENDIX-C: JAVA-CODE for Lookup table generator

```
public class CRC_slicingBy16 {


    static long[] crcTable0 = new long[256];
    static long[] crcTable1 = new long[256];
    static long[] crcTable2 = new long[256];
    static long[] crcTable3 = new long[256];
    static long[] crcTable4 = new long[256];
    static long[] crcTable5 = new long[256];
    static long[] crcTable6 = new long[256];
    static long[] crcTable7 = new long[256];
    static long[] crcTable8 = new long[256];
    static long[] crcTable9 = new long[256];
    static long[] crcTable10 = new long[256];
    static long[] crcTable11 = new long[256];
    static long[] crcTable12 = new long[256];
    static long[] crcTable13 = new long[256];
    static long[] crcTable14 = new long[256];
    static long[] crcTable15 = new long[256];



    public static void main (String arg[])
    {

        for (int i = 0; i <= 255; i++)
        {
            long crc32 = (long) i;
            for (int j=0; j<8; j++)
                crc32 = (crc32>>1)^((crc32&1)*0x04C11DB7);
            crcTable0[i] = crc32;
        }


           for (int i=0; i<=255; i++)
           {

           crcTable1[i] = (crcTable0[i]>>8) ^
crcTable0[(int)(crcTable0[i] & 0xFF)];
           crcTable2[i] = (crcTable1[i]>>8) ^
crcTable0[(int)(crcTable1[i] & 0xFF)];
           crcTable3[i] = (crcTable2[i]>>8) ^
crcTable0[(int)(crcTable2[i] & 0xFF)];
           crcTable4[i] = (crcTable3[i]>>8) ^
crcTable0[(int)(crcTable3[i] & 0xFF)];
           crcTable5[i] = (crcTable4[i]>>8) ^
crcTable0[(int)(crcTable4[i] & 0xFF)];
           crcTable6[i] = (crcTable5[i]>>8) ^
crcTable0[(int)(crcTable5[i] & 0xFF)];
           crcTable7[i] = (crcTable6[i]>>8) ^
crcTable0[(int)(crcTable6[i] & 0xFF)];
```

```
            crcTable8[i] = (crcTable7[i]>>8) ^
crcTable0[(int)(crcTable7[i] & 0xFF)];
            crcTable9[i] = (crcTable8[i]>>8) ^
crcTable0[(int)(crcTable8[i] & 0xFF)];
            crcTable10[i] = (crcTable9[i]>>8) ^
crcTable0[(int)(crcTable9[i] & 0xFF)];
            crcTable11[i] = (crcTable10[i]>>8) ^
crcTable0[(int)(crcTable10[i] & 0xFF)];
            crcTable12[i] = (crcTable11[i]>>8) ^
crcTable0[(int)(crcTable11[i] & 0xFF)];
            crcTable13[i] = (crcTable12[i]>>8) ^
crcTable0[(int)(crcTable12[i] & 0xFF)];
            crcTable14[i] = (crcTable13[i]>>8) ^
crcTable0[(int)(crcTable13[i] & 0xFF)];
            crcTable15[i] = (crcTable14[i]>>8) ^
crcTable0[(int)(crcTable14[i] & 0xFF)];
            }


            int[][] dataIn = new int[2][4];


            dataIn[1][3] = 0x0A0B0C0D;
            dataIn[1][2] = 0x0E0F0102;
            dataIn[1][1] = 0x03040506;
            dataIn[1][0] = 0x07080901;


            dataIn[0][3] = 0x01020304;  //127 downto 96
            dataIn[0][2] = 0x05060708;  //95 downto 64
            dataIn[0][1] = 0x09101112;  // 63 downto 32
            dataIn[0][0] = 0x13141516;  //31 downto 0


            long crc = 0;

            for (int j = 0; j<=1 ; j++)
            {
            crc ^= dataIn[j][0];



                  crc = crcTable0[dataIn[j][3]>>24 & 0xFF] ^
                  crcTable1[dataIn[j][3] >> 16 & 0xFF] ^
                  crcTable2[dataIn[j][3] >> 8 & 0xFF] ^
                  crcTable3[dataIn[j][3] & 0xFF] ^

                  crcTable4[dataIn[j][2] >> 24 & 0xFF] ^
                  crcTable5[dataIn[j][2] >> 16 & 0xFF] ^
                  crcTable6[dataIn[j][2] >> 8 & 0xFF] ^
                  crcTable7[dataIn[j][2] & 0xFF] ^

                  crcTable8[dataIn[j][1] >> 24 & 0xFF] ^
                  crcTable9[dataIn[j][1] >> 16 & 0xFF] ^
                  crcTable10[dataIn[j][1] >> 8 & 0xFF] ^
                  crcTable11[dataIn[j][1] & 0xFF] ^
```

```
                    crcTable12[(int)(crc >> 24 & 0xFF)] ^
                    crcTable13[(int)(crc >> 16 & 0xFF)] ^
                    crcTable14[(int)(crc>> 8 & 0xFF)] ^
                    crcTable15[(int)(crc & 0xFF)];
                }
                  System.out.println(Long.toHexString(crc));

        }

    }
```