Universidade de Aveiro

Mestrado Integrado em Engenharia de Computadores e Telemática

Arquitectura de Computadores Avançada

Assignment 1 – Cyclic Redundancy Check

Academic year 2017/2018

THEORETICAL BACKGROUND

Consider the *division algorithm* which is taught at primary school for the division of two non-negative integers, say, the division of 26378 by 745

$$\begin{array}{rl} 26378 & \vert 745 \\ \cline{2-2} 4028 & 35 \\ 303 & \end{array} \quad .$$

This algorithm is a direct consequence of a theorem in algebra, called the *division theorem*, which states that given two integers a and b, with $b \neq 0$, there is an unique pair of integers q and r such that

$$a = q \times b + r \quad \wedge \quad 0 \leq r < |b| \ .$$

As you may remember, the expression above was evaluated next to check whether the computations carried by hand for the application of the algorithm were correct or not – this last procedure is called the *real proof* of the division.

Any non-empty set where one can define the operations of addition, $+$, and multiplication, $\times$, between its elements, is called a *system of numbers*. When these operations have the same properties satisfied by the system of integers, we say it forms an *integral domain*.

An integral domain ID is, thus, characterized by

- the pair $(ID, +)$ to be an *abelian group*; that is, addition is associative, has a zero element, all the elements have a symmetric and is commutative
- the pair $(ID, \times)$ to be a *commutative semigroup* (multiplication is both associative and commutative) and, furthermore, for the existence of an identity element non-equal to zero and for the non-existence of zero dividers
- the validity of the distributive property of multiplication by addition.

It is worth noting that the non-existence of zero dividers plays a significant role for the division theorem to be valid, as it can be appreciated from the example below where the system of numbers is formed by the set $\{0,1,2,3,4,5\}$, together with the conventional operations of addition and multiplication carried out *mod* 6

$$\begin{aligned} 3 &= [(1 \times 2) \ mod \ 6 + 1] \ mod \ 6 \\ 3 &= [(4 \times 2) \ mod \ 6 + 1] \ mod \ 6 \ . \end{aligned}$$

An useful generalization of the concept comes from the consideration of sets of polynomials of degree n, $n \in \mathbb{N}$, with coefficients defined in some integral domain ID. The operations of addition and multiplication are the conventional operations of polynomial addition and multiplication. It can be shown that such *systems of numbers*, usually denoted by $ID[x]$, form an integral domain.

The *division theorem*, when the dividend polynomial is multiplied by x^p, becomes

$$a(x) \times x^p = q(x) \times b(x) + r(x) \quad \wedge \quad degree[r(x)] < degree[b(x)]$$

and can also be expressed as

$$a(x) \times x^p - r(x) = q(x) \times b(x) \quad (1)$$

where $p = degree[b(x)]$.

Suppose now that the coefficients of $a(x)$ represent the set of symbols of a message being transmitted, or of a word being stored in some medium, and that noise is present during transmission or storage so that some of the transmitted or stored symbols may be perceived as having a value other than the true one. The question that now arises is how confidant can one be that the received message, or the word read, is the still original one?

One way to proceed is to take advantage of the *division theorem* and transmit or store the left member of equation (1). At the receiving side, or when the word is read, one can perform polynomial division by $b(x)$ and check whether the remainder is still zero. If it is not, at least one wrong symbol is present; if it is, wrong symbols may, or may not, be present. However, one may try and find a polynomial $b(x)$ such that the probability of errors in the latter case be made small.

CYCLIC REDUNDANCY CHECK

In the digital world, the message or the word can be conceived as just a string of *zeros* and *ones*, which allows for the selection of GF(2) as the integral domain where polynomial coefficients are defined. GF(2) is the *Galois field* of two elements, 0 and 1. Addition and multiplication are the conventional operations of integer addition and multiplication, the first carried out *mod* 2. Notice that, in this domain, subtraction turns out to be equal to addition, $-0 = 0 \ \wedge \ -1 = 1$, being implemented as the *x-or* of the operands.

The *division algorithm*, thus, translates into

```
1000110101 |1001
0001110    |1001111
   01111
    01100
     01011
      0010
```

Let $t(x) = a(x) \times x^p - r(x)$ be the polynomial computed on the message to be transmitted, or the word to be stored, and $t'(x)$ be the received message, or the word read, then $e(x) = t'(x) - t(x)$ can be thought of as a string of *zeros* and *ones* where the *one* bits match the wrong symbol locations. As long as $b(x)$ does not divide $e(x)$, one immediately detects that $t'(x)$ is incorrect. If it does, however, there is still a chance that $t'(x)$ is incorrect. To minimize this probability, the polynomial $b(x)$ has to be carefully chosen.

Consider first the fact that if $b(x) \mid e(x) \wedge e(1) = 1$, there is an odd number of wrong symbols in the received message, or the word read, then $b(x) = 1$. Therefore, by choosing as $b(x)$ any polynomial with an even number of coefficients equal to *one*, one makes sure that any odd number of wrong symbols in $t'(x)$ are detected.

Consider next the fact that if there is a burst of wrong symbols in $t'(x)$ of length $k < p$, then one gets $e(x) = x^m \times (x^{k-1} + \cdots + 1)$ and $b(x)$ does not divide $e(x)$. Therefore, all bursts of wrong symbols of length lower than the degree of $b(x)$ are also detected.

Further analysis is complex and requires a lot of computation to find out the value of polynomials that minimize the probability of undetected errors on the received message, or the read word. The polynomial $b(x)$ is called the *generating polynomial* for a CRC scheme whose number of bits is defined by the degree of $b(x)$. A common choice for strings up to 64 bits is CRC-8, where the generating polynomial is

$$x^8 + x^7 + x^6 + x^4 + x^2 + 1 \ .$$

Design a digital circuit, called the *encoder*, which computes the *check string*, CRC-8 based, associated with a 16 bit data word. Design also a digital circuit, called the *checker*, which asserts the correction of the 24 bit CRC-8 coded word.

The assignment entails that some investigation should be made on finding the best possible algorithms for the implementation of the operations.

GRADING

- full specification of the *encoder* and proof of correctness of its design by VHDL simulation in Quartus – 14 valores
- full specification of the *checker* and proof of correctness of its design by VHDL simulation in Quartus – 17 valores.

DELIVARABLES

- an archive, named `CRC_T$G#.zip` (where $, equal to 1, … ,4, means the lab number and #, equal to 1, …, 10, means the group number), of the VHDL files of your solution
- a pdf file, up to 4 power point like pages, where the main ideas of the design are described.

DEADLINE

- November, 23, at midnight.