Performance Investigation of Bit-Counting Algorithms with a Speedup to Lookup Table

Eyas El-Qawasmeh
Department of Computer Science and Information Systems
Jordan University of Science and Technology
Eyasa@usa.net
http://www.geocities.com/eyasjo/index.htm

Bit-counting refers to the operation of counting the number of “1”s in a given computer word or binary vector. Currently there are several algorithms to solve this problem. The simplest one is serial shifting. Many recent algorithms have evolved in the last few years to overcome the slowness of serial shifting. However, their performance behaviour was not studied deeply. In this paper, the performance behaviour of existing algorithms will be investigated with clarifying comments. Moreover, an enhanced lookup table algorithm that is faster than the existing algorithms is presented and evaluated.

Key words: Bit-Counting, Bit-Parallelism, Counters, Redundant Coding, Performance, Popcount.

1. INTRODUCTION
Given an array of binary vectors, it is often necessary to determine how many “1” bits the array contains. This bit-counting problem is used in many applications such as file processing, coding theory, and some relational databases which uses bitmaps to index data that satisfy given conditions (Berkovich, et al, 2000). For example, a comparison operation between two files, texts, in terms of Hamming distance, can use bit-counting. Other examples, include the common bitmaps technique with the utility of counting the bits in them (Gutman, 2000).

There are several bit-counting procedures currently known. They are: a) Serial shifting, b) Arithmetic logic counting, c) Hamming distance bit vertical counter, d) Emulated popcount, e) Macro popcount, f) Lookup table, and g) Frequency division (Berkovich, et al, 2000; Berkovich, et al, 1998; El-Qawasmeh, Hemidi, 2000; Gutman, 2000; Reingold, et al, 1977). However, their performance behaviour has not been studied collectively to determine the best procedure.

The main objective of this paper is to do a performance comparative study of the mentioned algorithms and to introduce enhancements to the lookup table algorithm. Performance analysis of each algorithm will be discussed. Results show that the proposed improvement is the best among all known algorithms, as we will see in the following sections.