Computers and Electrical Engineering 59 (2017) 15–26

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Towards real-time crops surveillance for disease classification: exploiting parallelism in computer vision☆

Tallha Akram, Syed Rameez Naqvi*, Sajjad Ali Haider, Muhammad Kamran

Department of Electrical Engineering, COMSATS Institute of Information Technology, G.T. Road, Wah Cantonment, Pakistan

ARTICLE INFO

Article history:
Received 27 July 2016
Revised 21 February 2017
Accepted 21 February 2017

Keywords:
Plant diseases
Object of interest detection
Features extraction
Parallel processing
Asynchronous
Systems-on-chip
Networks-on-chip

ABSTRACT

Considering the incessantly increasing economic losses due to plant diseases in the agricultural sector, we have designed a real-time system capable of classifying plant diseases. In this context, we have proposed an image processing algorithm that transforms the image into three colorspaces, which are processed simultaneously. The algorithm executes in a series of intermediate steps, including contrast stretching, feature vector construction, and identification of salient regions. To enable effective execution, we have also proposed the underlying On-Chip communication architecture that allows efficient interconnection between the three digital signal processing cores, each processing its own colorspace. The architecture has been synthesized for 90 nm process, as well as on an FPGA, achieving a post-layout operational frequency of 644 MHz, and an area of 1208.9 μm^2 on the die. We demonstrate that our system outperforms few existing works in literature in terms of accuracy and computation time.

1. Introduction

In the agricultural industry, plant diseases are the major cause of economic losses worldwide . Plants display a range of symptoms from their early stage of attack to the final stage of damage, which can be monitored by utilizing efficient machine vision technology in surveillance systems. The conventionally built surveillance systems would normally require several image samples to be acquired, and then analyzed offline for the extraction of the infected regions. This analysis is further followed by manual pesticide application. The whole procedure becomes tedious if the fields span over several acres – demands large human resource and still remains a time-taking process [1].

An autonomous system, such as a surveillance drone or an unmanned rover, having real-time image acquisition and feature extraction capability, will not only be able to sweep the entire area and capture several samples, but will be able to determine the type of the infection. Subsequently, the system shall be able to determine the appropriate pesticide with minimal human intervention. In this quest , a necessary milestone will be to develop a fast image processing algorithm with an equally fast underlying hardware. Several image processing algorithms have been proposed that exploit parallelism to achieve higher performance. Parallelism, however, is only as good as the number of computing resources available on the hardware the algorithm is running on. The modern trend in digital design called system-on-chip (SoC) designs allows the designer to incorporate several computing resources on a single chip . These computing resources may include processing

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Huimin Lu.

* Corresponding author.
E-mail addresses: rameeznaqvi@ciitwah.edu.pk, rameeznaqvi@yahoo.com (S.R. Naqvi).

http://dx.doi.org/10.1016/j.compeleceng.2017.02.020
0045-7906/

elements (PE), intellectual property (IP) blocks, digital signal processing (DSP) cores, and to name a few. An efficient image processing algorithm would divide an entire image into several blocks, and then let the central processing unit (master) of the SoC distribute a block to each of the DSP cores (slaves) available, thereby allowing all the blocks to be processed simultaneously . The distribution of blocks to the cores, however, is once again limited by the intercommunication infrastructure. Since the last decade it has been established that networks-on-chip (NoC) is the only viable communication architecture for SoCs [2], which unlike the shared bus allows several message transfers in parallel.

To this end, we have devised an algorithm that transforms an image into three colorspaces, each of which may be processed at the same time. Naturally, this algorithm requires three processing cores on the SoC, and a suitable NoC architecture. In this paper we present details of our proposed algorithm, and the NoC. Considering the stringent power budget of the modern embedded systems, we have adopted an asynchronous implementation of the NoC (further abbreviated as ANoC), thereby relieving the system from the issues of global clocking and synchronization [3].

The rest of the paper is organized as follows: Section 2 reviews basics of NoC, and few of the existing parallel image processing architectures. The proposed algorithm is presented in Section 3. Descriptions of the ANoC architecture and system design are detailed in Section 4. Section 5 presents simulation results of the NoC router and the algorithm, before we conclude the paper in Section 6.

2. Background and related work

2.1. Basics of NoCs

An NoC is a programmable grid-like network of routers, allowing interconnection of several subsystems, called PEs or *computing resources*. In this network, each router has its distinct address in the form of (x,y) coordinates. Unlike the shared bus architecture, an NoC allows multiple PEs to communicate with each other simultaneously, therefore scales well with the increasing number of PEs. The primary objective of a router is to guide the incoming traffic from a computing resource to its desired output port connecting to another computing resource. Data messages from one computing resource to another travel in the form of packets, which are further comprised of flow control digits (flits) [4]. Usually there are two or more flits that make up a packet. The first flit is termed as the header flit, and the following flits that form the payload may be divided into two categories: body and tail. Since the two communicating entities may be significantly apart in the network, each flit may need to go through several other routers connected to various PEs. The path followed by the flits leads to two categories of routing: deterministic and nondeterministic (adaptive). In the former, which is the simplest and most widely adopted routing mechanism, each flit always follows the same path between two specific entities. This path is encoded within the header flit. The simplest form of the deterministic routing is the XY routing. As the name suggests, each packet (and so all the flits) must traverse the network along the X - axis first until it has reached the x - coordinate of the destination node, followed by traversing the network along the Y - axis before reaching the destination. It has been formally proven that this algorithm would not lead to deadlocks[1] by forbidding a few turns.

2.2. NoC based image processing architectures

Dave et al. [2] have recently proposed a torus NoC based image processing platform. Their algorithm is composed of six functions: gray scale and black & white processing, image inversion and rotation, histogram equalization, VGA display interface, computer interface, and shared memory. The authors have distributed each of these tasks to a single PE, thereby requiring an NoC comprising six nodes in total. They have implemented their design on a field programmable gate array (FPGA), which consumes 27% resources of the device.

Joshi et al. [5] have also made use of a torus topology, and presented the design of a reconfigurable router suitable for image processing applications. They have implemented their design on a Virtex - II FPGA. Joshi et al. in another work [6] have made use of a 2D-mesh topology with four nodes in total. The four function units include a wavelet engine and 2D convolver, a wireless interface, a display unit, and a memory module. The algorithm makes use of real time wavelet based denoising. The design has been implemented on Virtex - 2, where the NoC consumes 655 slices.

Yang [7] has made use of through silicon via (TSV) to implement a 3D NoC for 3D imaging VLSI system. The design makes use of four PEs and a RISC processor. The system has been synthesized for 130nm technology, with 208 pins, and a total gate count of 980,000. The achieved maximum operational frequency is 25MHz. The reconfigurable system allows several MPEG/JPEG algorithms to be implemented, including DCT/IDCT, pipeline image operation, multi-layer stacking method, and reconfigurable self-repairable memory.

The primary difference between all these existing works and our solution is twofold. We have opted for asynchronous design style that will give us an edge in terms of energy efficiency, and more robustness. Secondly, we propose a novel image processing method that distributes the images into three colorspaces, and exploits the available parallelism better than the approaches discussed above.

[1] A deadlock in the network occurs when several flits wait on one another to release a resource, forming a cyclic dependency.