Model-Based Design Optimization using CDFG for Image Processing on FPGA

SURACHATE CHUMPOL †1 PANADDA SOLOD†2 KRERKCHAI THONGNOO†1
NATTHA JINDAPETCH†2

Abstract: Model-Based Design (MBD) is an effective design methodology and able to develop rapid prototyping by using MATLAB/Simulink. However, MBD method may be difficult to handle a complex model, e.g. image processing algorithms, containing multiple nested loops. In this paper, we use Control Data Flow Graph (CDFG) which is an intermediate representation for analyzing complex algorithms so that the complex image processing applications can be practically implemented on an FPGA.

Keywords: Model Based Design (MBD), Control Data Flow Graph (CDFG), Field Programmable Gate Array (FPGA)

1. Introduction

Recently, FPGA (Field Programmable Gate Array) applications are more complicated due to their higher degree of system integration. A traditional FPGA development method by hardware description language (HDL such as VHDL or Verilog) takes time to develop and validate, it also needs special skilled FPGA engineers. Fortunately, recent high level synthesis (HLS) tools offer easier and faster development solutions for the increasing needs in complex systems.

In literature [1], [2] and [3], the authors proposed alternative approaches by using a Model Based Design (MBD) that is an effective design methodology and able to develop rapid prototyping by using MATLAB/Simulink. In literature [4] and [5], the authors proposed optimization, such as fixed-point optimization to reduce the area consumption in FPGA, for a lower cost FPGA board with limited resources. Applications in literature [1], [2], [3], [4] and [5] employed only few conditions without nested loops. So these literatures cannot be good candidates for complex models. On the other hand, literature [6] proposed a new approach, high level synthesis from C programming with speed optimization by using control data flow graph (CDFG) which is an intermediate representation for analyzing complex algorithms. Recent researches are focusing on an algorithm design issue of high level synthesis methodology for complex FPGA applications.

In this research, we need to investigate the alternative methods to solve this issue. Model Based Design (MBD) is the one of alternative methods that needs to be investigated. We aim to investigate this method and find practical processes to develop FPGA applications by this method using high level synthesis tool, such as HDL coder provided by MathWorks. Due to the good characteristic of MBD that we can simulate and prove our design concept in advance, so developers can focus on algorithm design instead of FPGA implementation. For complex model synthesis such as image processing, we need to investigate and apply CDFG to achieve the optimized goals.

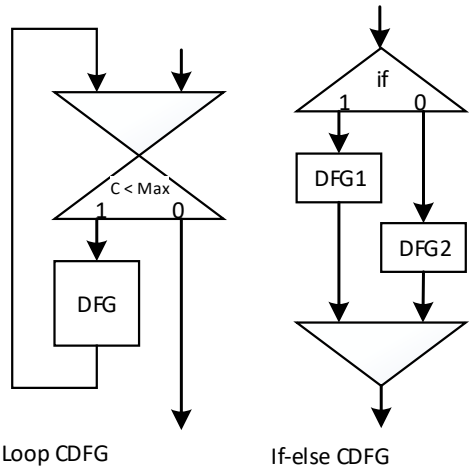


Fig. 1 CDGFs of the loop and the if-else statement.

2. CDFG Analysis Procedure

CDFG is an intermediate representation that combines control flow and data flow in the same graph. This raises the observability of data dependent among CDFGs so that scheduling and dataflow pipelining optimizations can be done.

Procedure: Analyze CDFG	
1:	**For each** loop or if statement **do**
2:	create $CDFG_{LX\_N}$ with node level assignment
3:	**For each** $CDFG_{LX\_i}$ where i=1, 2, …, N in the same level
4:	Determine data dependent
5:	**if** (data dependent)
6:	perform dataflow pipelining
7:	**else**
8:	perform parallel
9:	**For each** DFG in a loop-CDFG
10:	**if** (data dependent)
11:	perform pipelining
12:	**else**
13:	perform loop unrolling

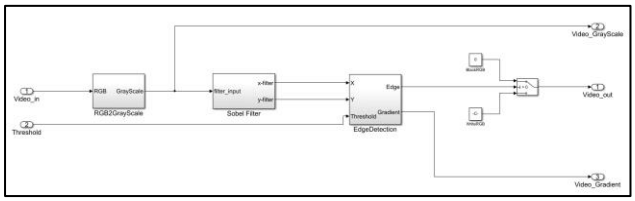
Fig. 2 CDFG analysis.

†1 Toyota Tsusho Nexty Electronics (Thailand) co , ltd , Bangkok, Thailand
†2 Department of Electrical Engineering, Prince of Songkla University, Thailand

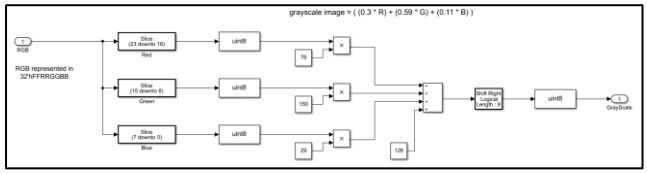
Fig.1 shows CDGFs of the loop (repeated from C=0 to C<Max) and the if-else statement. A CDFG has at least one Data Flow Graph (DFG). In nested loops or nested if-else statements, a CDFG contains inner CDFGs forming a hierarchical CDFG.

In this paper, a procedure to analyze a CDFG is shown in Fig.2. The data dependent among the CDFGs in the same level of the hierarchy is determined to perform either dataflow pipelining or parallel processing. Similarly, the data dependent among the operations in the same DFG is determined to perform either pipelining or loop unrolling that allows the parallelism.

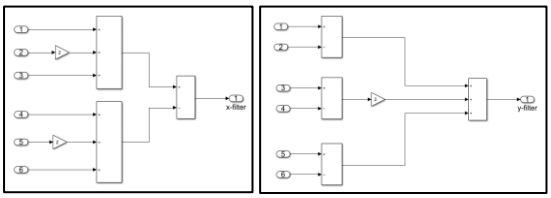
Main sub system in Simulink model



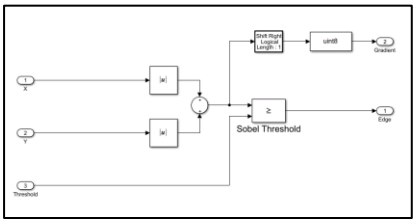
Gray scale sub system in Simulink model



Sobel filter sub system in Simulink model (x-axis and y-axis)



Edge detection sub system in Simulink model



Result image (Original, Gray scale, Gradient after Sobel filter and Edge detection)

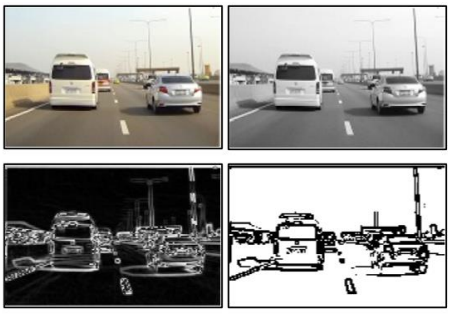


Fig.3 Edge detection algorithm in Simulink model.

3. CDFG Analysis Results

Fig.3 shows an example of edge detection algorithm in a Simulink model. There are 3 processes, Gray scale, Sobel filter, and Edge detection. As we generated HDL code by HDL coder, estimated processing time of each process is 3.703, 1.048 and 4.496 ns, respectively. For pixel streaming method, for example to process 3 pixels in these 3 processes, total processing time is 27.741 ns. Speed optimization by using control data flow graph (CDFG) [6] was employed by separating loop of each process and initiating as soon as the data are ready. The result of separated 3 processes, we can assume that processing time can be reduced to 18.239 ns. Observe that processing time of "Gray scale" with "Sobel filter" is just a little bit greater than "Edge detection" process, so we also can separate into only 2 processes, with almost same result as shown in Fig.4.

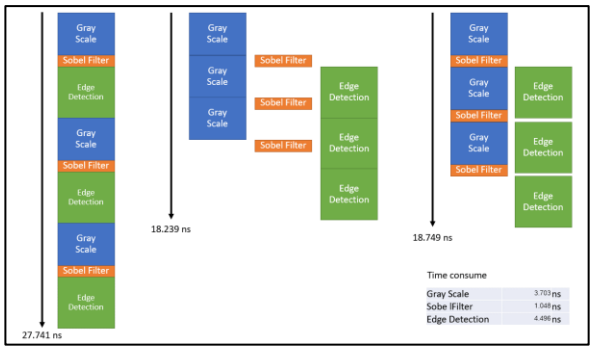


Fig.4 Processing time of dataflow architecture

4. Summary

This paper presents the application of dataflow architecture to optimize the speed by using CDFG analysis. For future work, we need to investigate how to implement this dataflow architecture concept in Simulink model.

Reference

[1] J.C. Ting Hai, O. Chee Pun, and T. Wooi Haw, "Accelerating Video and Image Processing Design for FPGA using HDL Coder and Simulink," in 2015 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET), pp. 28-32, 2015.

[2] S. Titri, "Rapid Prototyping of PVS into FPGA: From Model Based Design to FPGA/ASICs Implementation," in 2014 9th International Design and Test Symposium, pp.162-167, 2014.

[3] N. Othman, M. Hairol Jabbar, A. Kadir Mahamad, and F. Mahmud, "Luo Rudy Phase I Excitation Modeling towards HDL Coder Implementation for Real-time Simulation," Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia (UTHM), 2014.

[4] N. Othman, F. Mahmud, A. Kadir Mahamad, M. Hairol Jabbar and N. Atiqah Adon, "Cardiac Excitation Modeling: HDL Coder Optimization towards FPGA stand-alone Implementation," in 2014 IEEE International Conference on Control System, Computing and Engineering, pp.507-512, 2014.

[5] M. Besbes, S. Hadj Saıd, and F. M'Sahli, "FPGA implementation of high gain observer for induction machine using Simulink HDL Coder", 3rd International Conference on Control, Engineering & Information Technology (CEIT), 2015.

[6] S. Cheng, and J. Wawrzynek, "High Level Synthesis with a Dataflow Architectural Template" in 2nd International Workshop on Overlay Architectures for FPGAs (OLAF2016), pp.288-293, 2016.