Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores

Proyecto de Diseño en Ingeniería en Computadores

Design Document v1

Daniel Moya Sánchez

March 2, 2018

Table 1: Revision History

Date	Version	Description	Author
02 March 2018	1.0	Initial version of: Design Document for Design of ASIPs) for Approximate Computing	Daniel Moya

1 Introduction

1.1 Purpose

The primary purpose of this document is to present a description of the design elements of an ASIP and its use for implementing error-tolerant applications.

1.2 Scope

This research concerns the selection of error-tolerant applications and their implementations in optimized hardware. For this, ASIP configurations using specific approximate instructions for the selected applications are to be delivered. Furthermore, each ASIP configuration will be described by a set of parameters that the final system will possess, such as energy efficiency, area, execution time, and output error. This project is expected to help approximate computing to be a more widespread tendency and generate a strong base knowledge for future projects where there is freedom to choose the parameters of hardware running a certain type of application in terms of resource consumption and accepted error.

1.3 Context

Since approximated computing is still in its infancy, a lot of research and testing is still needed, so the users of the developed ASIPs are the same research groups behind this project.

Figure 1 shows the general scheme for an approximate possible application. As depicted, the approximate application is expected to have a pipeline structure, where one of its stages or sections (in that case the N-th stage) can be approximate. The purpose of the ASIP configuration is to provide the user flexibility between the original version of an application and the optimized one, allowing for a customized balance between resource consumption and output error. The approximate section needs to perform the original action but with less resource consumption, and still allowing for a acceptable output error, considering that a quality change in a specific section can have a big impact on other sections that depend of it.

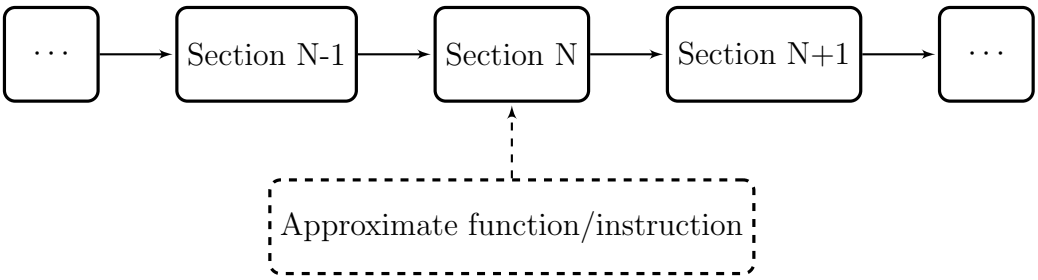

Figure 1: General context of the application.

1.4 Summary

The ASIP configurations that will be developed are expected to impact on the Instruction Set Architecture (ISA) and the processor components. The modifications made are specific to an application and must keep the original functionality intact. Since this research concerns about a low-level abstraction solution to the approximate computing paradigm, no high-level design is presented.

References

2 Glossary

Table 2: Definitions

Term	Definition
ASIP	Application Specific Instruction Set Processor. This means that, although the processor can execute a wide range of applications, it is optimized for a specific one, in which it can execute with improved performance (for instance, energy consumption or execution time would be lower) compared to a General Purpose Proccesor (GPP).
ISA	Instruction Set Architecture. It refers to the assembly instructions (low-level) that can execute in a certain processor.

3 Composition

The ASIP configurations that will be developed are expected to impact on the Instruction Set Architecture (ISA) and the processor components. For example, an assembly instruction may be created to be equivalent of two other assembly instructions, with added hardware

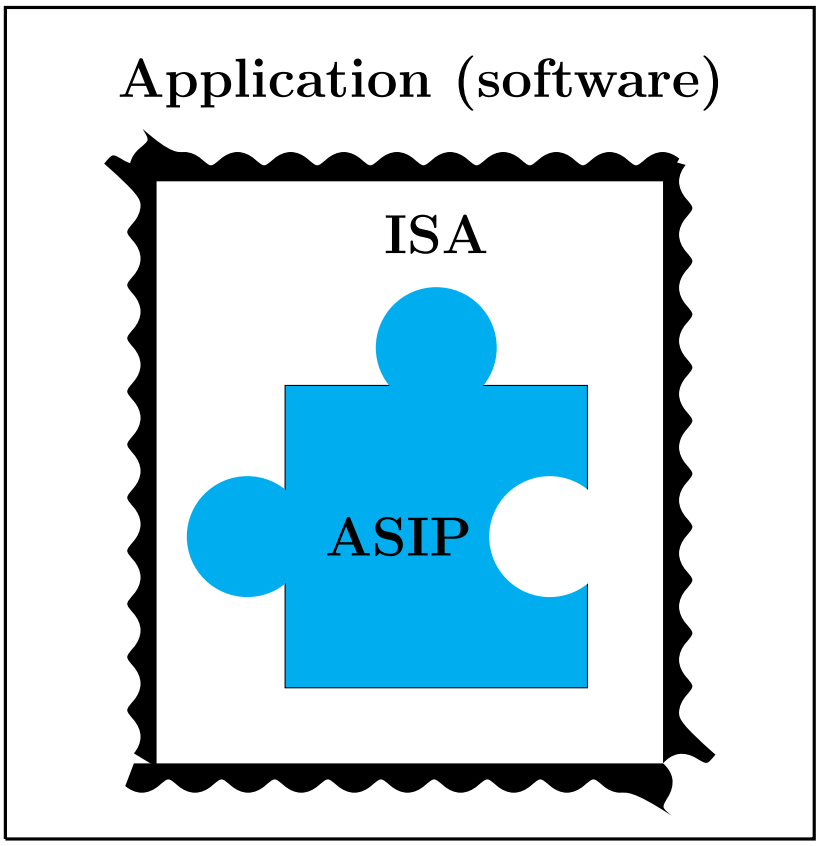

Figure 2: Composition level design

but less execution time, given that such operation is very frequent on the high level program. Figure 2 shows a general scheme for the interaction between these layers.

Figure 2 shows a level of abstraction from the inside ASIP jigsaw piece (low abstraction) to the outside softwate application (high abstraction). The ASIP is shown as a jigsaw piece due to processors and hardware components working together in several parts of an architecture, each performing an specific action simultaneously in a pipeline structure. Added modules have to fit in this jigsaw to keep everything else working, and removed hardware has to leave the application functionality intact. This means that, ideally, a perfect hardware jigsaw has to be made, where each specific piece its relevant and fits well with other pieces. The ISA is shown with fuzzy borders to give the sense that low level instructions can have a direct translation with the high level instructions; for example, a high level *add* instruction is pratically the same as its low-level counterpart, so no ISA-level optimization can be made, however, an *if* structure might be translated as several low-level instructions, so a specialized low-level instruction can be made in order to reduce the total number of assembly instructions that an *if* is translated to. Finally, the software application is shown as a box to highlight the fact that it is given as such and it is not modified, in other words, its constraints and specific details are kept.

4 Logical

Since the selected application is not going to be altered, this section does not apply.

5 Dependency

The approximate functions or instructions are expected to have to retain the original dependencies in a specific application.

6 Information

This section does not apply.

7 Patterns

This section does not apply.

8 Interfaces

This section does not apply.

8.1 User interface

Since the actual results of this research is knowledge, no user interface is made.

9 Structure

The hardware structure of the generated ASIPs will consist of the usual components for a pipeline processor (e.g ALU, registers) but with additional specialized components (according to the application) and reduced generic hardware (e.g ALU may only have needed operations).

10 Interaction

The interaction between the hardware components will follow a general pipeline scheme.

11 State dynamics

This section does not apply.

12 Algorithm

This section does not apply.

13 Resources

Since the resources consumption in an approximate application is one of this research's objetives topics, this section does not apply.