**Ve370 Introduction to Computer Organization Project 3 Report**

**Techniques and Future Development of Approximate Computing**

**I. Introduction and Background**

Recently, large-scale applications play a more and more important role in our daily life. But power and energy considerations is a limiting factor for digital hardware design. In the design, how to reduce energy consumption and the size of circuit becomes increasingly significant, especially for some mobile and embedded computing systems. Meanwhile, these applications also have an inherent tolerance to errors in computation, like applications that involve media processing, recognition and data mining. These applications share a common characteristic that it is not necessary to have a perfect accurate result. What they want is a result in acceptable range and small circuit size as well as low energy consumption. These applications are imprecision-tolerant. The sources of imprecision-tolerance are mainly divided into three parts: [1] perceptual limitations: these are determined by the ability of the human brain to fill in missing information and filter our high-frequency patterns; [2] redundant input data: this redundancy means that an algorithm is able to be lossy and still be adequate; [3] noisy inputs.

Approximate computing techniques have been proposed to solve this problem. The idea of approximate computing is that, use a smaller and less complex circuits to achieve the same goal of the large and complex circuit with an acceptable correct result. Approximate computing can be considered as a tradeoff between accuracy and energy. For some applications, the result need not be so accurate that we can use some approximate components to replace the accurate components in the circuit to make the size of circuit smaller and decrease the power with result in an acceptable range. We call this kind of applications error-tolerant application. To design an approximate circuit, we need some approximate units and some methods to know whether an approximate circuit works well.

In this paper, we will mainly talk about how approximate computing applies to FPGA and GPU and the future development of approximate computing. Before talking these parts, we will introduce some basic knowledge about approximate computing.

**2. Overview of Approximate Units and Metrics**

To build an approximate computing circuit, we need to build approximate arithmetic units first. Many people focus on the approximate arithmetic units. There are three main kinds of approximate computing units which are used most widely, approximate adder, approximate multiplier and approximate comparator.

There are many different approximate adders with different error distributions. In this part, we will only introduce several approximate adders.

*1) Approximate mirror adders:* It consists of five approximate mirror adders, which is an efficient adder. It removes some transistors to decrease power and simplify circuit.

2) *Accuracy configurable adder II*: It consists of several sub adders. The length of sub adders is smaller than the ACA-II. Sub adders are used to break the long carry chain. Except the first sub adder, other sub adders use several bits to predict the carry-in. Combine all the result of all sub adders and get the result. Because the long carry chain is broken, the size of circuit will be decreased and the speed of computation will increases.

The approximate multipliers which are widely used can be classified into following two types.

1): Multipliers with approximate partial products. This multiplier uses precise adders.

2): Multipliers with precise partial products. This multiplier uses approximate adders.

The main kind of approximate comparator just ignore several bits. It will ignore *k* LSBs and compare the rest *n-k* bits. When there is difference in the *k* LSBs, the result is wrong. The error rate of this comparator is

To know how an approximate unit and an approximate circuit work, we need some metrics for approximate computing

*Error rate* is the fraction of incorrect outputs out of a total number of inputs for an approximate circuit.

*Error significance* refers to the degree of error severity due to approximate operations.

*Error distance* is defined as the arithmetic distance between an inexact output and the correct output for a given input.

*Mean error distance* is the average effect of multiple inputs.

*Normalized error distance* is the normalization of MED for multiple-bit adders.

*Probability mass function* is used to see the error distribution for an approximate circuit. People always use PMF to check whether the approximate circuit gives an approximate result in acceptable range.