**Abstract:**

A Field-Programmable Gate Array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing. Once programmed they are similar to Application-Specific Integrated Circuits (ASICs). Their advantage as compared to traditional computer CPUs is that they use significantly less energy and produce less heat. If FPGAs are specifically programmed for data mining they could replace power hungry CPUs in

data centres and computer clusters in clouds and hence save significant energy for running and cooling the computer centres.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together". Logic blocks can be configured to perform complex combinational functions. This project focuses on implementing graph mining algorithms in FPGAs and uses these as canonical operations to execute complex data mining operations running on FPGAs or Graphical Processing

Units (GPU).

**(i) describe, with the past proposal as a guideline, the overall purpose/focus of the project**

This project focuses on taking advantage of data mining algorithms to speed up general Computer Aided Design (CAD) processes for FPGAs. With proper mining of the important information contained in the previous circuit designs, the time-consuming multi-iteration design cycle can be significantly reduced and thus a great amount of time and engineering effort can be saved.

More specifically, we have so far implemented a circuit similarity-based placement engine to accelerate the circuit design process. This proposed engine takes advantage of a novel and efficient circuit similarity detection methodology to quantitatively compute similarities among circuits. As such, the essential information in the previous circuit designs can be reused to guide the new design, and thus save the engineering effort and shorten the time to market.

So far, we have successfully applied the proposed circuit similarity-based placement to FPGA incremental designs and FPGA design space exploration. Our success on these two applications suggests a great potential to explore more applications. We are currently looking into FPGA routing process and FPGA verification process. On the other hand, we want to explore how to take advantage of FPGA to execute the time-consuming data mining algorithms in parallel.

**(ii) describe the motivation, related projects/applications, research questions, expected outcomes and timeline for the project**

**Motivation:**

Graph similarity is a general technique widely used in the data mining for various application domains including data warehousing, e-business, biochemical, software security and circuit design. However the current graph similarity algorithms have high complexity and cannot handle large network efficiently. This proposal project will develop scalable graph similarity algorithms to accelerate general data mining process, which can be applied to facilitate various time-consuming processes in the Computer Aided Design (CAD) processes for FPGAs.

**Context:**

Graph similarity is an extension of the graph kernel theory, which is used to quantitatively capture the intrinsic (and topological) information of graphs and use it compare multiple graphs. Given two graphs, graph similarity discovers the correspondence (similarity scores) between vertices and edges of them and therefore finds a matching between these two graphs. Matching elements of two data schemas or two data instances is one example application. Different from the graph isomorphism, graph similarity does not require the exactly same number of vertices or edges. One of the most popular measures of similarity between vertices in two graphs is iterative similarity score, which defines the similarity score of two vertices as the weighted sum of the scores of their neighbors. The time and space complexity of the iterative graph similarity is O(N\*M) for two graphs with N and M vertices, assuming the sparseness of both graphs. This complexity is unacceptable for large graphs such as social network (e.g., a medium site “eopinions.org” contains over one million vertices). Moreover, how to apply the graph similarity algorithm to other fields, such as circuit placement, routing or verification is a thriving yet un-solved problem.

**Proposal:**

The proposed project aims to accelerate the graph similarity algorithm, specifically, the iterative graph similarity algorithm, and apply it to the chip design domain to reduce various time-consuming CAD phases. We have adapted the iterative graph similarity algorithm to consider the unique circuit properties, such as fix I/O pads and directed connected wires, and used it to accelerate placement, one of the most time-consuming phase in CAD for FPGAs. We enhanced the performance of the adapted graph similarity algorithm by devising a support constraint and a level constraint to control the precision of the node matching. The experimental results showed that our circuit similarity-based algorithm can precisely detect the similarity between a new circuit and an optimized one, so that one can take advantage of the previous design efforts in the optimized circuit.

For our future work, we will continue improving the efficiency and effectiveness of the circuit similarity algorithm. We will also try to apply the circuit similarity-based placement to other time-consuming CAD phases, such as FPGA routing process and FPGA verification process. Moreover, we will to explore how to take advantage of FPGA to execute the time-consuming data mining algorithms in parallel.

**Results:**

We have successfully implemented the circuit similarity algorithm, and developed a circuit similarity-based toolset, including a circuit similarity placement tool and a circuit similarity visualization tool. We have experimentally demonstrated that the proposed circuit similarity-based placement is significantly faster than the state-of-the-art VPR placer with comparable placement quality.

We have published two conference papers and one journal paper is under review. One open-source software has been released and one U.S. patent proposal is being filed.