ABSTRACT

The idea of reconfigurable computing acts as a promising platform for executing both real and non-real time applications with the help of programmable devices such as Field Programmable Gate Arrays (FPGAs) in the Very Large Scale Integration (VLSI) market. The application design process started with vacuum tubes and bread-board based design styles. Due to more Time to Market (TTM), the above mentioned design styles are found to be not suitable for implementation of the recent high performance computing applications. In order to manage the design complexity of current applications, it is necessary to develop certain programmable devices. As a consequence, mask programmed and field programmable read only memory devices (PROMs) were designed. However, these design styles also contain certain disadvantages such as slow operation and high processing cost.

Following PROMs, a Masked Gate Array Application Specific Integrated Circuits (MGA-ASICs) were evolved with a high incurring cost. Therefore, in order to overcome the limitations of PROMs and MGA-ASICs based design styles, Programmable Logic Devices (PLDs) were introduced as a new dimension to the digital design. Subsequently, Complex PLDs (CPLDs) and FPGA devices also entered the computing field to bridge the gap between the PROMs and MGA-ASICs based designs.

In the province of reconfigurable computing, FPGA based design style has become a preferable method to implement both real and non-real time applications. Though the design styles based on FPGAs consumes more power, they have certain prominent features such as lesser design time, simple debugging, low cost, high parallelism, flexibility and

reconfigurability. Due to the advancements in VLSI technologies, programmable FPGAs are now available with very high density. Further the design process can be handled with the help of vendor-specific Computer Aided Design (CAD) tool. Design process in CAD flow of FPGA based design consists of several steps, in which the placement process plays a vital role in deciding the computing system performance. Current applications demand processing systems with a higher speed of execution, lesser power and area.

Field programmable gate arrays are well-suited for implementation in high performance applications due to the presence of high parallelism in its execution. Further, the partial-dynamic reconfigurable nature of FPGAs allows more than one task to be executed at a time which is known as multitasking for better resource utilization. Speed and area in FPGA based design style are decided in the placement process of design flow.To enhance the advantages of FPGAs based system design, efficient placement algorithms have to be developed to implement the real and non real time applications. In this research, various placement algorithms for executing real time independent and non-real time dependent hardware tasks on reconfigurable FPGAs have been discussed.

Placement algorithms with fast free space search and suitable fitting strategy are important to achieve the high speed of operation in real time applications execution. A new placement algorithm using Windowing OR Logic method with first fit strategy is proposed in this research. To execute the non real time applications execution using reconfigurable FPGAs, several optimization methods for placement have been discussed at the algorithmic and logic level. In the algorithmic level, the speed and area of non-real time application execution is decided by the wire length and the

selection of logic resources inside FPGAs. In this regard, various optimization algorithms such as particle swarm optimization, attractive-repulsive particle swarm optimization, gravitational search optimization and cuckoo search optimization algorithms have been applied to obtain the better placement with minimal wirelength and area. Further, a new method of logic level optimization has been introduced in the design entry of CAD process itself to achieve good placement of non real time applications execution on reconfigurable FPGAs.

Non real time hardware tasks consist of group of computing elements in which the adders acts as the basic computational element. In this research, the fast adder named carry select adder has been designed with the new concept of Multiplexer based AND-OR-INVERT (AOI) logic to achieve better placement. As a result, application design using the proposed adder is implemented on FPGAs with a minimal wirelength and area. Better placement in this approach is validated by Placement and Routing Architecture (PaR) obtained using Xilinx Simulation tool.

For experimentation purpose, the simulation parameters which model the real and non real time hardware tasks are randomly generated and used as an input to the placement algorithms. For real time independent hardware tasks placement algorithms, simulation has been carried out for the range of tasks width, height, arrival time, execution time, deadline and different FPGA size. Further, performance has been analyzed both for existing and proposed placement algorithms. From the simulation results, it can be observed that the proposed Windowing-OR-Logic method improves the average execution time which in turn increases the speed of execution especially for large scaled devices.

Performance improvement in execution time of the proposed Windowing OR Logic method was found to be 1.9 times better than maximal empty rectangle best fit overlap method, 6.2 times than least fit interference method, 8.99 times than Windowing CLook method and 12.8 times better than bottom left row column reduction method based placement algorithms.

In algorithmic approach, the non real time dependent hardware tasks are modelled using randomly generated directed data flow graphs (DDFGs) and the fitness function is formulated. To obtain the feasible placement of non real time hardware tasks, data flow graphs have been applied to optimization algorithms and the simulation out for several iterations with the objective of minimizing the value of fitness function considering both the single and multi-objective constraints. Cuckoo search optimization algorithm based placement of hardware tasks performs better with minimal wire length and area when compared to the others for all the input data sets generated.

Some discussions related to performance improvement of cuckoo search algorithm for wire length and area minimization are 38.6% in DDFG6 and 14.9 % in DDFG1 when compared to particle swarm optimization algorithm, 36.8% in DDFG5 and 13.9 % in DDFG1 when compared to attractive-repulsive particle swarm optimization and 33.5% in DDFG6 and 10.9% in DDFG8 than the gravitational search algorithm.

For multi-objective wirelength and area minimization, the performance improvement of cuckoo search algorithm is 62.4 % for the input DDFG2, which is better than the performance obtained using particle swarm optimization algorithm, 37.1% for the input DDFG3 which is better than attractive repulsive particle swarm optimization and 26% for the input

DDFG3 better when compared to the gravitational search algorithm. For experimentation in the logic level approach of non real time hardware tasks placement, carry select adder based on Multiplexer based AOI logic has been designed and the simulation has been carried out in Xilinx environment using Very High Speed Hardware Description Language (VHDL) coding. The Placement and Routing architecture, the number of LUTs used and the time taken to place and route are the parameters observed from the simulation environment to validate the feasible placement of proposed adder. Various high performance carry select adders have been considered for experimentation.

Further, the proposed carry select adder is applied to design Multiplier and Accumulator (MAC) unit with Wallace tree multipliers and mod-2 logic shift addition based multipliers. Targeting the Xilinx FPGAs, simulation results and PaR architecture of the proposed adder and its application on MAC units demonstrate better placement in terms of wirelength and area.