SELECTIVE FITTING STRATEGY BASED REAL TIME PLACEMENT ALGORITHM FOR DYNAMICALLY RECONFIGURABLE FPGAs

B.Premalatha[1], S.Umamaheswari[2]

[1] *Assistant Professor,Electronics and Communication Engineering, Sri Krishna College of technology, Coimbatore, TamilNadu,India.*

[2] *Associate Professor, Electronics and Communication Engineering,Coimbatore Institute of Technology, Coimbatore, TamilNadu,India.*

Abstract

Engineers in the field of Advanced computing paradigm struggles to satisfy the demand of high performance applications in terms of speed such as image processing, embedded computing ,video stream processing etc.,. Providing high speed in terms of truly multitasking of reconfigurable computing devices such as FPGAs acts as suitable computing platform for such applications. Two main problems in FPGA to fulfil the requirement are scheduling and placement of incoming hardware tasks. Scheduling and placement are the two process that depends on each other. Improper scheduling affects the placement performance. To the above two things, one more factor called fragmentation related closely that affects performance of FPGAs during placement. The performance metric used in this paper is task rejection ratio. Effective placement algorithm results minimum task rejection ratio. In this paper, we address the problem of real time scheduling and placement of hardware tasks by considering the factor called fragmentation in the objective to minimize the task rejection ratio. We developed an Selective fitting strategy based algorithm and simulation is carried out. Results are compared with basic placement fitting strategy such as first fit, best fit and worst fit. Our algorithm shows better performance in term of task rejection ratio.

Key Words*: Dynamically Reconfigurable FPGAs, On-line Scheduling and Placement, Resource management, Task Rejection ratio.*

---***---

1. INTRODUCTION

Reconfigurable Hardware devices such as Field Programmable Gate arrays (FPGAs) are becoming the dominant technology for high performance computing applications. Dedicated programmable electronic device is used as key mechanism in the such applications named robot, toys, cameras, mobile telecommunication etc.,Today the high speed based design applications appear endless(i.e., those application s require trillions of operations to perform per second). But the design engineers facing the challenges in design with constraints of power, flexibility, performance in terms of cost, speed etc.,

Recent Semiconductor technology of high speed and high density digital design pushes out these constraints. Reconfigurable FPGAs consists of limited resources for execution. So the operating system should properly manage limited resources for more number of tasks to execute by process named Scheduling and Placement. Incoming hardware tasks has to be properly scheduled and placed inside the FPGA, to minimize the task rejection rate. Truly multitasking in Reconfigurable Hardware devices such as FPGAs speed up the execution of application. In this paper, we address real time scheduling and fragmentation aware placement of hardware tasks in targeting the minimum task rejection ratio. Paper is organized as follows, About Reconfigurable FPGAs, its Classification ,issues and the performance metrics are discussed in Section 2. Section 3 consists of Literature Survey about the scheduling and placement algorithms. Fragmentation and fitting strategies are discussed in Section 4. Proposed algorithm along with calculations involved is discussed in Section 5. In Section 6 we discussed about the simulation results. Conclusion and future work is discussed in Section 7.

2. RECONFIGURABLE FPGAs

The related work pertinent to our method includes scheduling and placement algorithms to minimize task rejection ratio in dynamically reconfigurable FPGAs. Static Random access Memory (SRAM) based Programmable Logic devices are named as Field Programmable Gate Arrays (FPGAs) which are also called as reconfigurable hardware devices. These devices play a major role in high speed design due to is various advantages. FPGAs are first introduced in mid 1980s by Xilinx, but the recently introduced devices are with advanced features. Dynamic run time reconfigurability of SRAM based FPGA wets the thirsty requirement of such high speed applications. Also it brought out the new possibilities in applications in terms of flexibility, power etc., Most of the high performance applications are dynamic in nature and can be satisfied by using recently developed FPGAs such as Virtex II, Virtex II pro, Virtex 4, Virtex 5 ,Virtex 6, and Virtex 7 from Xilinx family. The various Xilinx family Virtex FPGAs varies based on the technology and it satisfies the demanding requirements of dynamic applications such as software defined radio, aerospace, railways etc in addition to above mentioned applications. Block Diagram of FPGA Architecture (Homogeneous type) is shown in Figure.1.

I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	CLB	CLB	CLB	CLB	CLB	CLB	CLB	CLB	I/O
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

Fig -1: FPGA Architecture (Homogeneous)

With increase in advance in VLSI technology, the SRAM based reconfigurable FPGAs are designed with millions and millions of transistors as resources to implement the logic. Field Programmable Gate Arrays consists of Configurable logic blocks arranged in matrix form surrounded by the interconnection switches and I/O pads.Configurable Logic blocks consists of group of logic cells (Each logic cell is named as frame) to implement the function defined by tasks.Configuration is defined as sending the bit streams that represents the functionality of tasks to FPGA through the reconfiguration controller. To execute an application, the multiple incoming hardware tasks can be placed and removed dynamically in FPGA once it completes the execution called as dynamic partially reconfigurable FPGAs.

2.1 FPGA Classification

Based on the Configurable Logic blocks arranged, the FPGA is classified into two types named i) Homogeneous FPGA ii) Heterogeneous FPGA. Basically these FPGAs are two-dimensional but depends on the mode of placement it is sub named as one-dimensional FPGAs. Homogeneous FPGAs **-** is one which consists of only Configurable Logic blocks inside the FPGA area to implement the function defined by hardware tasks. In 1D Homogeneous FPGA, the tasks can be placed for its execution only in vertical manner i.e., the height of FPGA is constant but the width can be varied. 2D Homogeneous FPGA of size(10x10) is shown in Figure.1. Once the tasks placed, if it occupies the full height of FPGA columns represents the 1 dimensional FPGA. The disadvantage of 1D Homogeneous FPGAs is more fragmentation and efficient defragmentation algorithm should be used. But in 2D Homogeneous FPGAs, the hardware tasks can be placed anywhere along the width and height of FPGA area. Heterogeneous FPGAs **-** is one which consists of Configurable logic blocks and special elements such as embedded memory, DSP blocks, high speed multipliers and special processors in between the CLBs.Heterogeneous FPGA is shown in Figure.2. Scheduling is very difficult in heteregeneous FPGAs but it offers many advantages and it is widely used now-a days. CLB's are arranged in vertical form, and so it is of 1D heterogeneous type. It also suffers from more fragmentation. (SB-Special Block- Memories, multipliers etc.,).

I/O	CLB	SB	CLB	CLB	SB	CLB	CLB	SB	I/O
I/O	CLB	SB	CLB	CLB	SB	CLB	CLB	SB	I/O
I/O	CLB	SB	CLB	CLB	SB	CLB	CLB	SB	I/O
I/O	CLB	SB	CLB	CLB	SB	CLB	CLB	SB	I/O
I/O	CLB	SB	CLB	CLB	SB	CLB	CLB	SB	I/O
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

Fig -2: FPGA Architecture (Heterogeneous)

2. 2 Challenging Issues in One-Dimensional FPGAs

In most of the Industrial applications, the placement concept will be in one-dimensional form i.e once the task gets placed inside FPGA,it occupies the full height in order to avoid the fragmentation. So in this paper, we discussed about the placement algorithm for one-dimensional FPGAs. Arriving time based online scheduling is used in this work. If the tasks not find place, it can be in waiting list, and then scheduled after it finds the place before its deadline meets.

Here we modelled the 1D -FPGA area as one-dimensional array of logic 1 and logic 0. The number of 1's and 0's in the array denotes the total number of configurable logic blocks(CLB) in FPGA.The logic 0 denotes the empty CLB and logic 1 represents the occupied CLB. In 1D FPGA, the height of the task occupies the full height of the vertical dimension of FPGA. Due to limited CLBs / resources in FPGA, the real time incoming tasks has to be placed efficiently and rejection rate has to be minimized. Placing more number of tasks in real time with in the limited resources is an challenging issue in reconfigurable computing platform. For eg., an 1D FPGA model of size 22X1 shown in Figure.3. In 1D FPGA, task occupied the full height, when it finds the placement. So the model for analysis has been taken as one-dimensional array of CLB's.

0	0	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1

Fig -3: One- Dimensional FPGA Area model

Concept of placement: For each incoming tasks, the free space is searched to place the tasks, from left side to right side of the area. While searching, If the number of continuous CLBs found are enough to place the task, select that space for placing based on any one of the fitting strategy. In this paper, the quality of placement is measured by the factor called task rejection rate(TRR) and in turn it depends on the fitting strategy. The task rejection is done only when the incoming tasks missed their deadlines. Most of the research work concentrates any one of the fitting strategy for the task placement and also the mixed fitting strategy to improve the performance. In this paper, all the three fitting strategies are intelligently selected based on tasks width and we minimize the task rejection ratio. The comparison between the first fit, best fit , worst fit and the proposed algorithm is also shown.***Performance metric:*** Metrics are used to measure performance of FPGA is the average task rejection ratio (TRR) in %. ***Reason for considering average performance metrics:*** The incoming hardware tasks are assumed to real (i.e., all the tasks parameters are unknown in advance.). Real time tasks are generated randomly and to decide the performance the various sets of tasks are generated. To finalize the performance the average performance is considered.

$$\text{Task Rejection Ratio (TRR)} = (N_I - N_P) / N_I \times 100 \qquad (1)$$

Where N_I =Total Number of Input tasks, N_P=Number of Tasks Placed.

$$\text{Average task rejection ratio} = \sum_i (TRR) \qquad (2)$$

Where i=1,2,3,…N. N=total number of tasks set.

Various different algorithms are developed for scheduling placement of tasks in FPGA and are literally surveyed to understand the concept.

3. LITERATURE SURVEY

Maissam Mansub Bassiri and Hadi Shahriar Shahhosieini [1], discussed about the hardware/software partitioning, scheduling and placement algorithms of FPGA targeting embedded system applications.[2] Guy Wassi-Leupi 2011, a thesis discussed about the Scheduling and placement algorithms and their performance are shown and motivates to find new algorithm for the same.

Thomas Marconi, Yi Liu, Koen Bertels and George Gaydadjidev, [3] proposed an intelligent stuffing method is used to improve the quality of placement and the performance results is shown as 89.75 reduction in total wasted area, 1.5% in scheduling time and 31.3 % shorter response time and is suitable for embedded applications. Tulika Mitra,[4],focused on developing the online hardware task scheduling and placement on 3D partially reconfigurable FPGAs, which is suitable for recent applications.

Thomas Marconi, Yi Liu, Koen Bertels and Georgi Gaydadjiev [5] proposed the various intelligent merging techniques for on -line placement in Partially reconfigurable computing systems and the performance results are shown.Bazargan’s algorithm, the placement quality is improved by 1.72 times. Thomas Marconi and

Basics of VLSI CAD automation and its algorithms are read from text book named as Sahib H.Gerez [6]. Jin Cui, Zonghua Gu, Weichen Liu & Qing xu Deng,2007 [7] proposed an efficient algorithm named Scan line algorithm SLA), for online management of PR-FPGA where the concept of finding the complete set of Maximum set of rectangles for on-line scheduling and placement of hardware tasks is used. The role of Reconfigurable hardware devices in embedded system and its application is known from Philip Gargia, Katherine Compton, Michael Schulte, Emly Blem and Wenyin Fu [8]. Operating system level of Reconfigurable Embedded system has been referred in [9]. Total literature survey about the scheduling and placement algorithm for reconfigurable devices has been done and prepared as paper [10].

4. FRAGMENTATION AND FITTING STRATEGIES

One of the reasons that effective resource utilization of FPGA gets affected by the factor called fragmentation. i.e., FPGA area gets splitted into various segments once the tasks are placed. Placement of hardware tasks inside FPGA area introduces fragmented area, which means the available space is not contiguous. Even though the number of CLBs are more than enough to place the incoming tasks, due to non-contiguous space the tasks may be either rejected or it has to wait for some time till the continuous space available. In general, placement of tasks is based on any one of the fitting strategy named as first fit, best fit and worst fit. Definitions of basic strategies are as follows:

First fit: First fit strategy selects the first free space for placing/accommodate the tasks.(search space time is less)

Best fit: Best fit strategy selects the smallest free space that is big enough to place the tasks.(search space time may be more/ less depends on the tasks width).

Worst fit: Worst-fit strategy selects the largest free space to place the tasks. (search space time may be more/ less but fragmentation is less).

5. PROPOSED WORK AND CALCULATIONS INVOLVED

In most of the research work, researchers concentrate any one of the fitting strategy for the task placement and also the mixed fitting strategy to improve the performance sometimes. In this paper, the fitting strategies are selected in suitable form to achieve the goal of less fragmentation while placement i.e., depends on the available fragmented FPGA area and the incoming tasks width, the less fragmented fitting strategy is selected. The proposed modified fitting strategy based placement algorithm shown better performance than the basic fitting strategies. After simulating the code written in C, the comparison results between the first fit, best fit ,worst fit and proposed algorithm is shown..(in selecting the fitting strategy ,we avoid the least fragment of size 1,if any). Idea to phrase the formulae to calculate the fragmentation is referred in [11].

5.1 Calculations involved in proposed work

Let f1,f2,f3... be the fragmented segments in FPGA area after the tasks are placed in a time, then for the further incoming tasks, calculate the metric as follows :

(i) Consider the tasks width as w

(ii) Calculate the metric one by one as first fit, best fit, worst fit for fragmented space in FPGA: m1=f1-w; (due to first fit) ; M1= (1/m1)+ ∑ (1/f). f=f2,f3,.....(applicable only if m1~=0) else M1=∑ (1/f). f=f2,f3,.....

(iii) Similarly find m2: m2=f-w. (based on best fit). f will be any one of fragmented segmenf1,f2,f3..... then calculate M2=(1/m2)+ ∑ (1/f), where f=f1, f2, f3 excluding the selected fragment . ,(applicable only if m2~=0) else M2=∑ (1/f). f=f1,f2,f3,..... excluding the selected fragment.

(iv) Similarly find m3: m3=f-w .(based on worst fit). f will be any one of fragmented segmenf1,f2,f3..... then calculate M3=(1/m3)+ ∑ (1/f), where f=f1, f2, f3 excluding the selected fragment,(applicable only if m3~=0) else M3=∑ (1/f). f=f1,f2,f3,..... excluding the selected fragment

v) Find M= min{M1,M2,M3}.

If M=M1, first fit strategy is selected, M=M2, best fit strategy is selected,M=M3,worst fit strategy is selected. For various fragmented level inside FPGA area, based on the task width the value of this metric varies. Select the fitting strategy, which results less fragmentation metric .Over all performance is calculated in terms of task rejection ratio.

5.2 Model Calculation

Let us consider the number of CLBs inside the FPGA area is 22(value may be vary). Assume that the incoming task width of w=1, w=2 and w=3 at different instant. The fragmented area is shown below for our example,

Initially, the fragmented FPGA area is given as (f1=4;f2=2;f3=6)

0	0	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1

For example: w=2

As per first fit strategy,

1	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1

The metric is calculated as: m1=f1-w=2;
M1= (1/2)+(1/ 2)+(1/6)=1.16

As per best fit strategy,

0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1

The metric is calculated as : m2= f2-w=0; if m2=0; it is compactly packed. So do need to consider for metric calculation : M2= (1/4)+(1/ 6)=0.41.
As per worst fit strategy,

0	0	0	0	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1

The metric is calculated as : m3= f3-w=4 ; M3= (1/4)+(1/ 2)+(1/4)=1.
M=min{M1,M2,M3}=min{1.16, 0.41, 1}=0.41. Thus the fitting strategy is best fit strategy

Metric values for different tasks width is calculated and tabulated as in Table 1.

Table -1: Calculation of Metric for Various Tasks Width

S..no	Task width	Metric FF (M1)	Metric BF (M2)	Metric WF (M3)	M	
1.	W=1	1	1.41	0.95	0.95	M3
2.	W=2	1.16	0.41	1	0.41	M2
3.	W=3	1.66	1.66	1.08	1.08	M3
4.	W=4	0.66	0.66	1.25	0.66	M1
M1=FF-First Fit, M2=BF-Best Fit,M3= WF-Worst Fit						

6. SIMULATION RESULTS AND DISCUSSION

By considering the FPGA size of 50x1, the real time hardware tasks are generated in random of 20 sets of each 200 tasks. Coding is written in C & the simulation is carried out for first fit, best fit, worst fit strategies and also for proposed algorithm. The number of task rejected in each set and the response time is observed and the corresponding graph is plotted. Simulation is done for both fast arriving tasks and slow arriving tasks of width varies in random between ([2- 10], [5- 15], [10 -20]). The deadline for each task ranging from ([arr. time+ext ime+1], [arr. time+ex.time+k]), where k=5 or 10. The execution time lies between in random of [1 - 7]. Simulation results are tabulated in Table 2a, 2b,2c and Table 3a,3b,3c.

Table-2 a Task Rejection Rate in % Versus Fitting Strategies for Slow Arriving Tasks

S..no	Fitting Strategies	Task Rejection Rate in %		
		Small width [2- 20]	Medium width [5 -15]	Large width [10- 20]
1.	First Fit	4.575	18.6	40.27
2.	Best Fit	3.9	18.875	35. 32
3.	Worst fit	3.05	18.55	39.05
4.	Proposed algorithm	2.7	18.05	33.7

Table-2 b Task Rejection rate in % Versus Fitting Strategies for Fast Arriving Tasks

S.no	Fitting Strategies	Task Rejection Rate in %		
		Small width [2- 20]	Medium width [5 -15]	Large width [10 -20]
1.	First Fit	1.55	16.25	40
2.	Best Fit	1.3	15	29.6
3.	Worst fit	0.725	16.3	37.55
4.	Proposed algorithm	0.325	12.525	33.6

Table-2 c Task Rejection rate in % Versus Fitting Strategies for Fast and Slow Arriving Tasks (1000 Tasks)

S.no	Fitting Strategies	Task Rejection Rate in %		
		Fast arriving tasks [5 -15]	Fast arriving tasks [2 - 20]	Slow arriving tasks [5 -15]
1.	First Fit	3.9	23.4	7.8
2.	Best Fit	4	14.8	7.1
3.	Worst fit	3.3	18.5	5.5
4.	Proposed algorithm	2.6	12.6	5

6.1 Graphical Representation of Results

Simulation results for random fast arriving tasks with width varies from [2- 10] ,[5 -15] and [10 -20] is shown below in Figure.4, Figure.5, Figure.6.

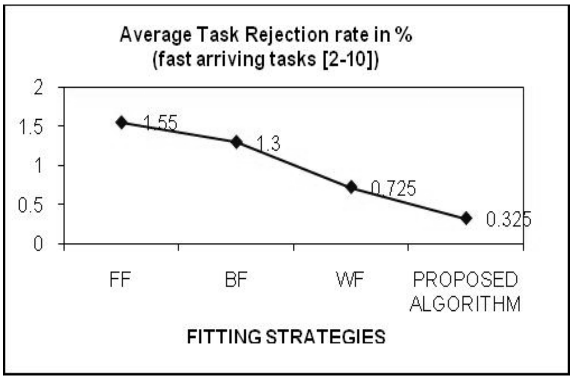

Fig-4: Task Rejection Rate in % versus Fitting Strategies for Fast Arriving Tasks of Width [2 -10]

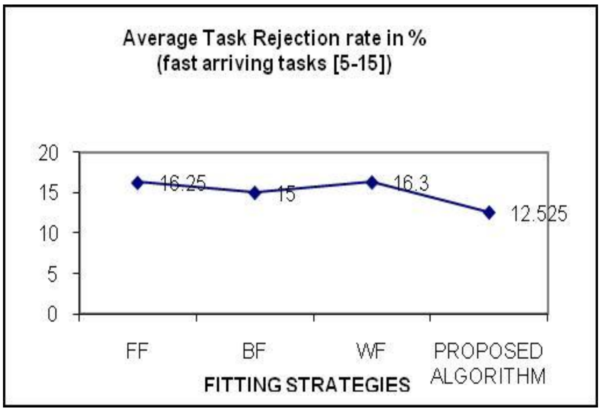

Fig-5: Task Rejection Rate in % versus Fitting Strategies for Fast Arriving Tasks of Width [5 -15]

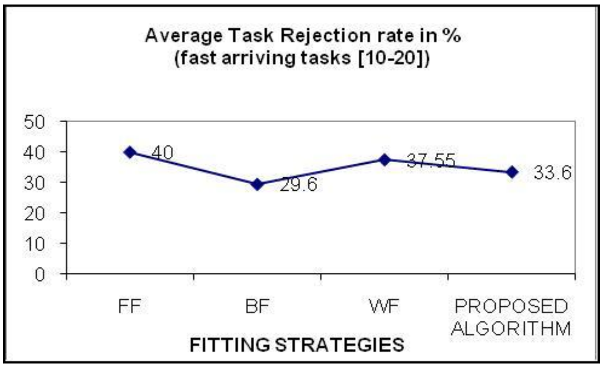

Fig-6:. Task Rejection Rate in % versus Fitting Strategies for Fast Arriving Tasks of Width [10-20]

For random slow arriving tasks with width varies from [2-10],[5- 15] and [10 -20] , the simulation results are shown in Figure.7, Figure.8and Figure.9.

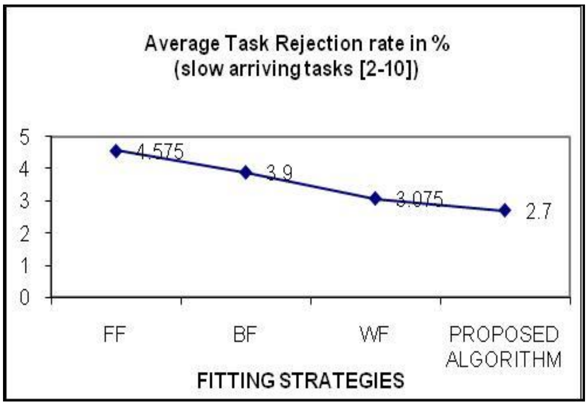

Fig-7: Task Rejection Rate in % versus Fitting Strategies for Slow Arriving Tasks of Width [2-10]

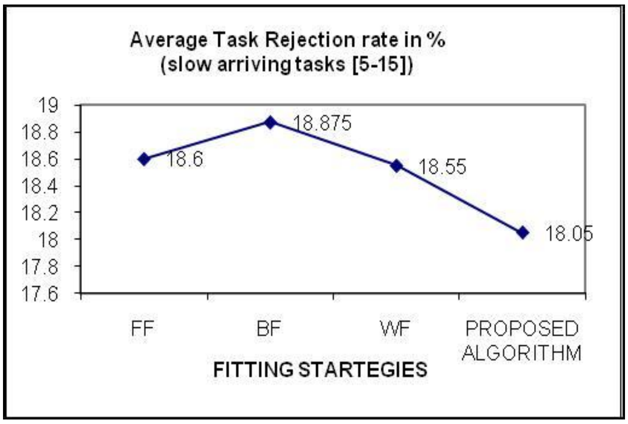

Fig-8: . Task Rejection Rate in % versus Fitting Strategies for Slow Arriving Tasks of Width [5-15]

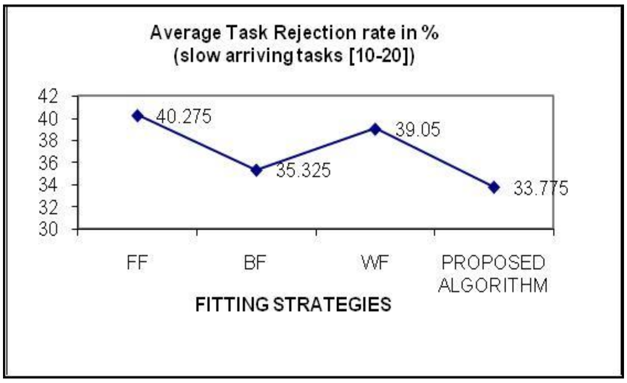

Fig-9: Task Rejection Rate in % versus Fitting Strategies for Slow Arriving Tasks of Width [10-20]

For randomly generating 1000 real time tasks and the algorithm is applied, the corresponding results are shown in Figure.10.

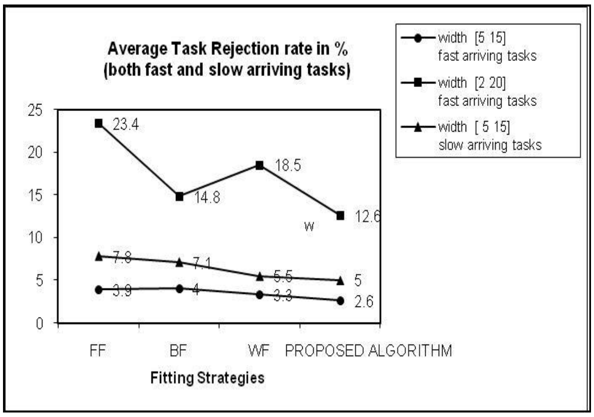

Fig-10: . Task Rejection Rate in % Versus Fitting Strategies for Fast and Slow Arriving Tasks.(1000 tasks)

From the simulation results we conclude that the proposed selective fitting strategy based algorithm produces the best placement of tasks in terms of minimum task rejection ratio as compared to first fit, best fit and worst fit strategies.

7. CONCLUSION AND FUTURE WORK

Minimum Task rejection ratio is obtained for the tasks execution in dynamically reconfigurable computing system by using the selective fitting strategy based placement algorithm. A selective fitting strategy based placement algorithm has been applied as an acceptance test of real time hardware tasks in real time computing environment. In this paper, we assumed 20 sets of 200 tasks in each set is (randomly generated) considered for algorithm verification In future, we extend this work to incorporate with real time heterogeneous computing systems. Applying the same to heterogeneous FPGAs which is widely used nowadays in industries and also for high speed computing system will be the future work.

REFERENCES

[1].Maissam Mansub Bassiri and HadiShahria Shahhosieini,"Future Baesd on-line Integrated HW/SW Partitioning and co- scheduling for Reconfigurable Co mputing systems",Journal of Information science and Engineering vol.27, 2011,pp. 1687-1711

[2]. Guy Wassi-Leupi,"On line scheduling for real time mu ltitasking on Reconfigurable Hardware Devices", thesis, July 2011.

[3]. Thomas Marconi, Yi Liu, Koen Bertels and George Gaydadjidev, ",On-line Scheduling and placement algorithm on partially reconfigurable devices, Springer, ARC 2008, LNCS 4943, pp 306-311,2008.

[4]. Thomas Marconi and Tulika Mitra,".A Novel Online Hardware task scheduling and placement Conference on Field-Programmable Technology (FPT'11), New Delhi, India, Dec 2011.ISBN: 978-1-4577-1740-6,2011.

[5]. Thomas Marconi, Yi Liu, Koen Bertels and Georgi Gaydadjiev,"Intelligent Merging on -line task placement algorithm for partial Reconfigurable system, DATE 2008,Proceedings of the conference on Design, automation and test in Europe, pp 1346-1351,ISBN-978-3-9810801-1-1,2008.

[6]. Sahib H.Gerez,"Algorithms for VLSI Design Automation. 2nd Edition, Wiley Student edition,New Delhi,2007.

[7]. Jin Cui, Zonghua Gu, Weichen Liu & Qing xu Deng,"An efficient algorithm for online soft real time task placement reconfigurable Hardware devices", Proceedings of tenth International symposium on object –oriented real time distributed computing, ISORC 2007,
May 2007, IEEE Computer Society,pp no. 321-328.

[8]. Philip Gargia, Katherine Compton, Michael Schulte, Emly Blem and Wenyin Fu," An overview of Reconfigurable Hardware in Embedded System", Eurasip Journal on Embedded system, ID 56320,2006,pp no, 1-19.

[9].Krishnamoorthy Baskaran and Thambipillai Srikanthan,"A Hardware Operating System based Approach for Run time Reconfigurable platform of Embedded Devices",2003.

[10]. Dr.S.Umamaheswari and Ms.B.Premalatha, **"**Survey of online Hardware Task Scheduling placement algorithms for partially reconfigurable computing systems", International Journal of Computing and Corporate Research, ISSN:2249-054X.

[11]. Ahmed Abou Elfarag,Hatem M .El-Boghdadi, Samir I.Shaheen,"Miss Ratio-Improvement for real time applications using Frag -mentation aware Placement",IEEE 2007.

BIOGRAPHIES

3..B.Premalatha received the B.Engg,,degree in Electronics and Communication Engineering from Kumaraguru College of Technology, Coimbatore, India in 2000 and the M.E degree in VLSI Design from College of Engineering, Anna University, Chennai,India in 2007. She is Doing Phd., in Reconfigurable Computing in Anna university Coimbatore as a part-time scholar since 2009. Currently she is an Assistant Professor, Department of Electronics and Communication Engineeing, Sri Krishna College of Technology, Coimbatore, India. Area of research interest include Reconfigruable computing, high level synthesis algorithms, VLSI.

Dr.S.Umamaheswari received the B.Engg degree in Electronics and communication Engineering from Government College of Technology, Coimbatore ,India in 1985 and the M.E Degree in Applied Electronics from Coimbatore Institute of Technology, Coimbatore, India in 1991 and Ph.D degree From Coimbatore Institute of Technology, Coimbatore, India in 2009.Currently she is an Associate Professor, Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, India. Area of research includes VLSI, Signal and Image processing.