A new model of placement quality measurement for online task placement

Yi Lu, Thomas Marconi, Georgi Gaydadjiev and Koen Bertels
Computer Engineering Lab., TU Delft, The Netherlands
$\{yilu, thomas, georgi\}$@ce.et.tudelft.nl, k.l.m.bertels@tudelft.nl

Abstract— **With the arrival of partial reconfiguration technology, modern FPGAs support tasks that can be loaded in (removed from) the FPGA individually without interrupting other tasks already running on the same FPGA. Many online task placement algorithms designed for such partially reconfigurable systems have been proposed to provide efficient and fast task placement. In these algorithms, the resource wastage and task rejection rate are usually used to measure placement quality. However, these algorithms only calculate them individually. These considerations can not reflect the overall situation of placement quality during the application execution. In this paper, we propose a novel model for placement quality measurement, which consists of resource wastage from both placed task side and rejected task side as well as the information of task rejection rate and task life time.**

I. Introduction

The FPGAs are broadly used in partially reconfigurable systems. With the reconfigurability of the FPGAs, these systems can achieve more flexible adaptation to various applications by reconfiguring the FPGAs when required. Most of these systems use single configuration per FPGA which means the whole FPGA is reconfigured. The complete reconfiguration usually brings long reconfiguration time and increased power consumption.

In recent years FPGAs with partial reconfiguration support can address these problems by only reconfiguring the necessary part when required. However, the partial reconfiguration technology brings more complex 2D FGPA area partitioning which implies that efficient task placement algorithms are required. The offline and online approaches are normally used to solve this problem. In the offline task placement algorithms, the optimized location of each task is decided before the application is running based on profiling results. In the online task placement algorithms, no information is available about each task until the task arrives. The algorithms search suitable locations for arrival tasks at run time. This implies that highly efficient resource management is required. The existing online task placement algorithms use the resource wastage as a measurement of algorithm quality. The resource wastage is defined as the unusable FPGA partitioning fragmentation appearing during the task placement process.

In previous proposals, only resource wastage caused by placed tasks and task rejection rate are used separately to measure the placement quality. This can not reflect the efficiency of using FPGA during the application execution because of the lack of the rejected task details. In this paper, we propose a new model to measure the placement quality during the task placement process. By using our model, the placement quality can be clearly observed using a complex number plane. The main contributions of this paper are:

- a new model to measure the placement quality for online task placement on the FPGA;
- the mapping of the placement quality onto the complex plane

In section II we present related task placement approaches. Then, section III details our proposal for new placement quality measurement. Next, in section IV, we apply our model to measure the placement quality for existing online task placement algorithms. Finally, we discuss future work in section V.

II. Related Work

Bazargan et al. [1] considered the problem of task placement on the FPGA as the well-studied 2D bin-packing problem. In addition, they described efficient tree data structure for online FPGA resources management. Walder et al. [2] improved Bazargan's on-line algorithm and used a hash matrix to manage free FPGA resources which guarantees the constant location searching time. Tabero et al. [3] used vertex-lists to record the free spaces, where each vertex is a possible location for arrival tasks. A new arrival task is placed by selecting a suitable vertex from the list. Ahmadinia et al. [4] proposed a new way to manage the FPGA free resources, that only stored the information about used space. Also, the authors considered the connectivity among tasks when placing an arriving task. In [5], Steiger et al. described an enhanced version of Bazargan's placement algorithm considering both scheduling and placement for an on-line solution. Marconi et. al. [6] proposed an online task placement algorithm based on the assumption that the sizes of tasks obey the normal distribution, which means most tasks in an application have medium size. Therefor, they partitioned the FPGA surface into different size blocks in which most have medium size.

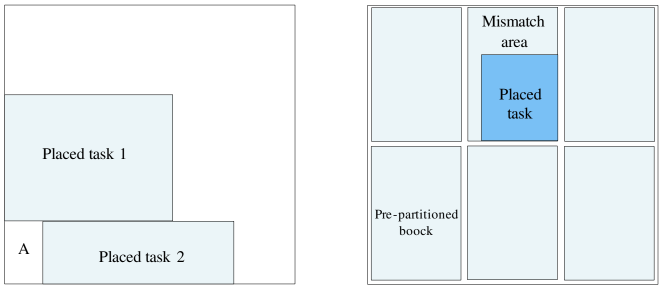
In these algorithms, rejection rate and resource wastage are the two factors used to measure the placement quality. The resource waste only provides the information about the unusable fragmentation on the FGPA caused by the placed tasks. The rejection rate only gives how many percents of tasks are rejected during the application. However, it is more useful to know how many FPGA resources should be used for these rejected tasks, together with the resource wastage brought by placed tasks during the entire application lifetime, which reflects the application efficiency using the FPGA. In this paper, we propose the new placement quality measurement model which takes into account the effect from both rejected tasks and placed tasks. The details about the model will be described in the next section.

III. The new placement quality model

The model proposed in this paper is named "placement quality measurement in complex number domain" (PQM-IC). The PQM-IC model takes the effects from both placed tasks and rejected tasks into account. In this section, we first define some concepts used in the following discussion. Thereafter, our model is detailed. Finally, the complex plane representation of our model is described.

A. Definitions

Life time: When we mention life time of an area, it means the time period that this area exists on the FPGA. E.g. the life time of placed task i means the total execution time of the task i. The life time of an application is the time from its starting to the end.
Waste rectangle: In the online task placement algorithms with non pre-partitioned FPGA area model [7], the waste rectangle is defined as the rectangle which is smaller than the minimum task size shown in the application, e.g. in figure 1 (a), the area A is smaller than the size of any task in the application, so this area A is the waste rectangle which can not fit any task.
Real resource wastage and real waste product: For the online task placement algorithm with pre-partitioned FPGA surface model, the real resource wastage is defined as the area unoccupied by the placed task in its assigned pre-partitioned block as shown in figure 1 (b), which is referred to as *mismatch area*; the real waste product is the product of a mismatch area and its life time on the FPGA. For the algorithms with non pre-partitioned FGPA area model, the real resource wastage is the sum of waste rectangle areas; the real waste product is the product of a waste rectangle with its life time on the FPGA. In previous task placement algorithms, only this real resource wastage is taken into account when measure the placement quality.
Imaginary resource wastage and imaginary waste product: The imaginary resource wastage is defined as the area of rejected tasks which were supposed to be implemented on the FPGA but could not fit. The imaginary waste product is the product of rejected task size and its life time running on the FPGA if it is placed on the FPGA. This imaginary is a novel concept proposed in this paper. By using this imaginary resource wastage, we can finally combined the real resource wastage and imaginary resource wastage together to provide a clear depiction of placement quality. The details are presented in the next section.



(a) (b)

Fig. 1. Resource waste by placed tasks

B. Our model

By adopting the complex number representations, our $PQM\text{-}IC$ model uses complex numbers. There are two different styles in our $PQM\text{-}IC$ model.

The first style:

$$A_{ac} = \frac{\sum_{i=1}^{n}((S_{block} - S_{aci}) \times T_{lifei})}{S_{all} \times T_{app}} \times \frac{n}{m+n} \qquad ......(1)$$

$$A_{ac} = \frac{\sum(S_{less} \times T_{period})}{S_{all} \times T_{appless}} \times \frac{n}{m+n} \qquad ......(1)'$$

$$B_{rj} = \frac{\sum_{j=1}^{m}(S_{rejectj} \times T_{lifej})}{S_{all} \times T_{reject}} \times \frac{m}{m+n}) \qquad ......(2)$$

$$r_w = A_{ac} + iB_{rj} \qquad ......(3)$$

$$R_w = \sqrt{A_{ac}^2 + B_{rj}^2} \qquad ......(4)$$

$$\tan a = \frac{B_{rj}}{A_{ac}} \qquad ......(5)$$

In equation (1), S_{block} represents the size of the block assigned to the accepted task i; S_{aci} is the size of accepted task i; T_{lifei} is the life time of task i; T_{app} is the application life time. In equation (1)$'$, S_{less} is the size of a waste rectangle; T_{period} is the life time of the waste rectangle; $T_{appless}$ is the total application execution time when there are waste rectangles on the FPGA. In equation (2), $S_{rejectj}$ stands for the size of the rejected task j and T_{reject} for is the total life time of rejected tasks if they are mapped on the FPGA. For these three equations, S_{all} is the size of the whole FPGA; m and n stand for total number of accepted tasks and rejected tasks respectively.

As shown in equation (3), the placement quality measurement defined in this paper follows complex number format. The real part A_{ac} and the imaginary part B_{rj} correspond to the resource wastage brought by placed tasks and rejected tasks respectively.

In equation (1) where the A_{ac} is defined for pre-partitioned FPGA area model (e.g. 1D, 2D and IF in this paper), the numerator represents the sum of the product of real resource wastage and its existing period; the denominator is given as the product of the complete FPGA area and the application execution time; the quotient of them reflects that how many percents of the FPGA resource are wasted by placed tasks during the application execution. Then by multiplying the rate of the placed task taking from the number of total input tasks, we average the resource wastage caused by placed tasks. In equation (1)$'$, the A_{ac} is defined for flexibly partitioned FPGA area model (e.g. BBF and BFF). The equation (2) define the average resource wastage from rejected tasks. The equation (1)$'$ and (2) holds Similar explanation as equation (1). The equation (4) gives the absolute value for placement quality which is used for our comparison.

In the equation (5), the angle a is named *contribution factor*, the value of a reflects the contribution to the absolute value of placement quality from both real resource wastage and imaginary resource wastage. The large value of a [degree] means the average imaginary resource wastage (B_{rj}) is relative larger during the application execution compared to real resource wastage(A_{ac}). This corresponds to three situations during the application execution: 1) relatively large number of tasks are rejected, 2) few tasks with long computation time are rejected, 3) combination of 1) and 2). These cases imply that the task placement algorithm used in the reconfigurable system can not achieve a highly efficient FPGA usage when running the application, which is not expected when designing a task placement algorithm.

The second style:

In the second style, we only replace equation (5) with another two new equations (6) and (7), which define the angle b [degree]. We name the angle b the *reject factor*. The reject factor reflects the rejection rate during the application, which is the number of total input tasks divided by the number of rejected tasks.

$$\sin^2 b = \frac{m}{m+n} \qquad ......(6)$$

$$\cos^2 b = \frac{n}{m+n} \qquad ......(7)$$

In both model styles, the placement quality is depicted as a coupled vector as shown in figure 2.

C. Complex plane

With equations (1)-(5), the placement quality can be mapped onto the complex plane. As shown in figure 2, there are two placement quality vectors ($v1$ and $v2$). These two vectors have the same absolute value (V), but different contribution factors ($a1$ and $a2$). This means that although they have similar absolute placement quality values, it is obvious that there is more imaginary resource wastage in $v2$ because of the larger value of $a2$. By replacing equation (5) with equations (6) and (7), the angle b directly represents the rejection rate, e.g. between $v1$ and $v2$, although they have same absolute value, the larger $b2$ implies the high task rejection rate in the algorithm where the $v2$ is created. The analysis about these vectors is detailed in the next section.

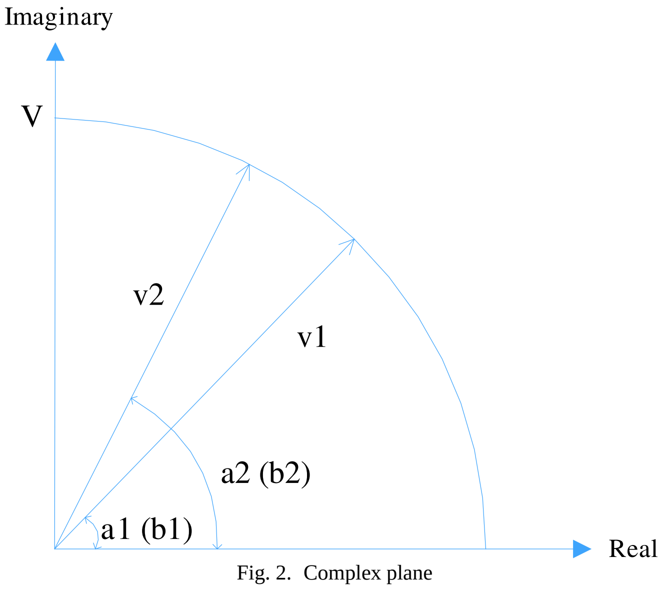


Fig. 2. Complex plane

IV. Experimental evaluations

We simulated four state of the art online task placement algorithms, BBF [1], fixed 1D [7], fixed 2D [7] and IF [8]. The first three approaches are previously proposed for online task placement. The IF is the new online task placement algorithm proposed by us recently. In the following context, we will briefly describe the IF algorithm [8].

A. The IF algorithm

The IF is characterized by fast allocation of available FPGA area and highly efficient usage of the FPGA resources. By initially partitioning the 2D FPGA surface into various size blocks based on application requirements, the IF implements merge, split and recover operations to these blocks. These operations guarantee the high resource usage by making the FPGA resource flexible redistribution according to run time requirement of applications. In addition, linked lists are used to store available free blocks and the available free block is always in the first node of linked list, which gives a fast searching time. There is an example in figure 3 to show how the IF works.

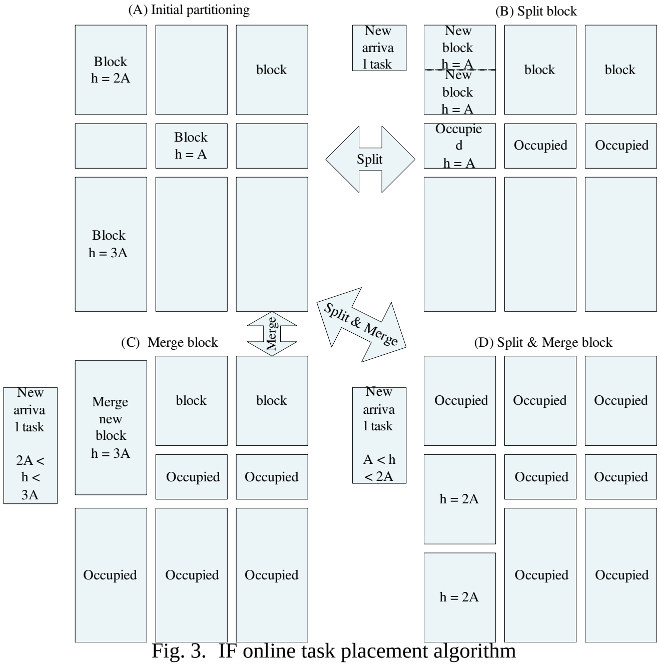


Fig. 3. IF online task placement algorithm

There are three types of split and merge processes: split only, merge only, and split-merge. They correspond to (B), (C), and (D) in Figure 3 respectively. The (A) is the FPGA initial partitioning which suitable for frequently used IP cores. The initial partitioning can be adjusted according to different application requirements. A split only process splits a large size block into smaller size blocks. The reverse is merge. For example, in Figure 3(B), all A-size blocks are occupied, when another A-size input task arrives. A 2A-size block is split into two A-size blocks which can be used by the new input task. The recovery process in the IF guarantees that these resources are able to be re-assembled into the original blocks which can be reused by all different size tasks. Overall, by making the initial partitioning on the FPGA and adding operations to the blocks, our IF shows better performance than the other three approaches in terms of algorithm execution time and task rejection rate. In the next section, we will use our proposed $PQM\text{-}IC$ to calculate resource wastage for all these algorithms and make comparison of their performance of resource wastage.

B. Simulation results

The simulation results of BFF, fixed 1D, fixed 2D and IF were collected to calculate resource wastage with our proposed

		T500	T1000	T1500	Tmix
1D	A_{ac}	0.1300	0.0780	0.0260	0.0770
	B_{rj}	0.0020	0.0050	0.0085	0.0050
	V	0.13	0.0780	0.0270	0.0772
	a	0.88	3.67	18.1	3.7
	b	15.21	14.77	14.3	14.54
2D	A_{ac}	0.0210	0.0166	0.0160	0.0250
	B_{rj}	0.0110	0.0340	0.0570	0.0055
	V	0.0270	0.0380	0.0590	0.0256
	a	33	64	74.3	12.4
	b	39.6	39.67	39.72	16
BBF	A_{ac}	0	0.0322	0.0393	0
	B_{rj}	0	0.0070	0.0286	0.0013
	V	0	0.0329	0.0486	0.0013
	a	0	12.26	36.04	90
	b	0	17.15	28.5	6.5
IF	A_{ac}	0.0200	0.0275	0.0260	0.0260
	B_{rj}	0.00009	0.0009	0.008	0.0025
	V	0.02	0.0275	0.0272	0.0260
	a	0.18	1.87	17.1	5.5
	b	3.27	6.33	14	8.77

TABLE I
SIMULATION RESULTS

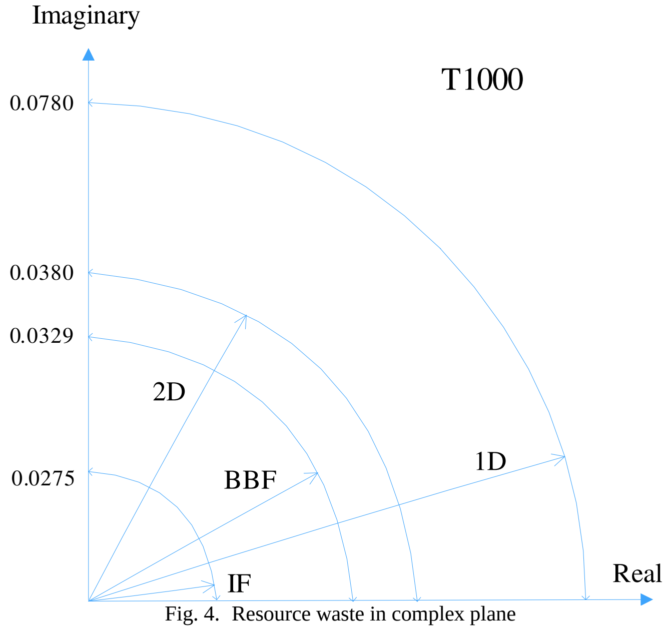


Fig. 4. Resource waste in complex plane

$PQM\text{-}IC$. All algorithms are programmed using C and simulated under Linux 2.6 with Intel Pentium(R) 4CPU 3.00GHz.

The simulation results for the four online task placement algorithms are shown in table I. In the figure 4, the simulation results with T1000 task set are depicted in complex plane. In order to make the figure clear, we do not follow exact scales, but keep the originally related positions of resource wastage vectors. According to the absolute value in figure 4, it is obvious that IF has better performance for placement quality compared to other algorithms. The 1D algorithm obtains worst performance, and the large real resource waste contributes to the worst placement quality, which is the mismatch of the pre-partitioned blocks and placed tasks. This implies that the pre-partitioning in this 1D algorithm can not bring optimized performance for this application. The 2D algorithm has the biggest contribution factor reflecting the high imaginary waste, which is normally brought by high task rejection rate. In addition, this means during the application execution, the FGPA can not be efficiently used by implementing the 2D task placement algorithm.

Difference between the new model and previous approaches: In previous approaches, the resource wastage and task rejection rate are used to measure placement quality. They are calculated individually as shown in table I. It is implicit to observe the placement quality from these scalar values. In our proposed new model presented here, we use the medium of complex numbers to combine these factors together in a single complex number. The placement quality represented by the complex number vector provides explicit depiction of the placement quality as shown in figure 4.

V. Conclusion and Future Work

In this paper, we proposed a new model ($PQM\text{-}IC$) to measure the placement quality of online task placement algorithms on the FPGA. By adopting complex number representation, the $PQM\text{-}IC$ depicts placement quality in the complex number vector. The overall FGPA resource usage during the application execution is directly represented in the depiction in the complex number plane. In the future, we will integrate this model into previously proposed online task placement algorithms with pre-partitioned FPGA model. In these algorithms, the $PQM\text{-}IC$ will be used to manage the dynamic FPGA partitioning to meet requirements of various applications.

VI. Acknowledgment

This work is sponsored by the hArtes project (IST-035143) supported by the Sixth Framework Programme of the European Community under the thematic area “Embedded Systems”.

References

[1] K. Bazargan, R. Kastner, and M. Sarrafzadeh, “Fast template placement for reconfigurable computing systems,” in *In IEEE Design and Test of Computers*, vol. 17, 2000, pp. 68–83.

[2] H. Walder, C. Steiger, and M. Platzner, “Fast online task placement on fpgas: Free space partitioning and 2d-hashing,” in *In Reconfigurable Architectures Workshop (RAW)*, Nice, France, April 2003.

[3] J. Tabero, J. Septien, H. Mecha, and D. Mozos, “Low fragmentation heuristic for task placement in 2d rtr hw management,” in *14th International Conference on Field Programmable Logic and Application, FPL 2004*, Antwerp, Belgium, Sept. 2004, pp. 241–250.

[4] A. Ahmadinia, C. Bobda, S. P. Fekete, J. Teich, and J. C. van der Veen, “Optimal free-space management and routing-conscious dynamic placement for reconfigurable devices,” in *IEEE Transactions on Computers*, vol. 56, May 2007, pp. 673–680.

[5] C. Steiger, H. Walder, M. Platzner, and L. Thiele, “Online scheduling and placement of real-time tasks to partially reconfigurable devices,” in *In Proceedings 24th IEEE International Real-Time Systems Symposium (RTSS)*, Cancun, Mexico, December 2003, pp. 224–235.

[6] T. Marconi, Y. Lu, G. Gaydadjiev, and K. Bertels, “An intelligent lookahead merge onine placement algorithm,” in *In Architecture and Compiler for Embedded Systems(ACES)*, Ter Elst, Edegem, Sept. 17-18, 2007.

[7] H. Walder and M. Platzner, “Online scheduling for block-partitioned reconfigurable devices,” in *In Design, Automation and Test in Europe, (DATE’03)*, Munich, Germany, March 2003, pp. 290–295.

[8] Y. Lu, T. Marconi, G. Gaydadjiev, and K. Bertels, “An on-line task placement algorithm for partially reconfigurable systems,” in *In Architecture and Compiler for Embedded Systems(ACES)*, Ter Elst, Edegem, Sept. 17-18, 2007.