Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

***Abstract*—**
***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Floorplanning is one of the major challenges in the field of dynamic parital reconfiguration. The placement of the static and reconfigurable slots on the FPGA fabric must satisfy the application requirements set by the application designer while also respecting the technological constraints set by the manufacturer. The conventional approach for an automated generation of FPGA floorplans usually involves two steps. First for each slot, all the possible rectangular slots that satisfy the resources requirement of the slot are enumerated. This is done by starting a scan on the fpga fabric from the bottom left corner and lisitng all the rectangles that contain all the necessary resources for the respective slots. Then some sort of heuristics/optimization is applied to choose the optimal ones from the set of possible slots. This approach has many problems {*to be listed later*}

Our approach instead focuses on applying the optimization process on a lower level of abstraction of the fpga fabric i.e., rather than applying optimization to select the most optimal one from a set of pre-scanned slots, we modeled the different types of resources on the fpga and their distribution along with forbidden regions as a set of constraints and added these constraints to the predefined constraints related to dpr.

Let us consider a floorplanning example where we have to make a floorplan for two slots S_1 and S_2 on the FPGA fabric. Each slot has resource requirements denoted as $\{D_1, B_1, C_1\}$ and $\{D_2, B_2, C_2\}$ where D, B and C represent DSP, BRAM and CLB respectively.
Our proposed system takes as an input in the resource requirement of each slot and a description of the resource distribution of the FPGA fabric and it returns the placement coordinates of the slots on the fpga fabric.
A slot is represented using 4 parameters i.e. the two bottom left coordinates and the width and the height of the slot. In our considered example the slots S_1 and S_2 are represented as (x_1, y_1, w_1, h_1) and (x_2, y_2, w_2, h_2). A forbidden region is also represented as a slot hence a forbidden region F_i can be represented as fx_i, fy_k, fw_i, fy_i

Identify applicable funding agency here. If none, delete this.

II. FPGA resource representation

A. transformation to binary

The FPGA is divided into multiple grids (columns) of different resources with heterogenous distribution. Heterogeneous distribution implies the layout of different resources differs from clock region to clock region. The resources on the two dimensional region are represented with x and y coordinate system. the X axis runs from 0 to W - 1 and the Y axis runs from 0 to H-1.

To use linear optimization the x or y axis can also be represented with a set of binary variables which denote each row or column respectively. Choosing which axis to change in to binary is an important design decision. In all the FPGA families that were chosen to be studied for this project, the Y axis was observed to contain less number of resources than the x axis. For example in a kintex xc7z045fbv676 contains 100 columns on the x axis and 70 rows on the y axis. Perhaps representing the y axis with binar of variables was a better option as it results in a less number of binary variables. The reduction on the number of binary variables to represent each row on the y axis was made possible by assigning a row per bram instead of rows per clb or per dsp on each column as a bram spans at least 5 or more clb high and 2 or more dsp high. This is actually a neccessary abstraction as brams are indivisble and a finegrained assignment of rows (that is based on clbs or dsps) ends up generating a floorplan that violates rules in the implemetnation tool.

B. FPGA resource finger-printing

The FPGA is now abstracted using binary variables on the y axis and integers on the x axis. The fpga is also divided into clock regions. A clock regions spans r rows high and H which

is the total number of rows is the summation of each row in each clock region i.e.,

$$H = \sum_{j=0}^{clk\_reg} r \tag{1}$$

The distribution of each resource on the x axis of the FPGA in a single row can be represented using a piece-wise linear function. For example in zynq xc7z015 the number of clbs on the x axis between the bottom left corner i.e., (0, 0) and a point x, on the x axis is represented using F(x) as

$$F(x) = \begin{cases} x, & \mathbf{0 \leq x < 4}, \\ (x-1), & \mathbf{4 \leq x < 7}, \\ (x-2), & \mathbf{7 \leq x < 10}, \\ (x-3), & \mathbf{10 \leq x < 15}, \\ (x-4), & \mathbf{15 \leq x < 18}, \\ (x-5), & \mathbf{18 \leq x < 22}, \\ (x-6), & \mathbf{22 \leq x < 25}, \\ (x-7), & \mathbf{25 \leq x < W}, \end{cases} \tag{2}$$

The number of clb, in a height of a single row, between x_1 and x_1 can then be represented as clb(x_1, x_2) such that

$$clb(x_1, x_2) = F(x_2) - F(x_1) \tag{3}$$

if β_{ijk} represents row k in clock region j for slot i on the fpga then C(x_i, y_i, w_i, h_i) which is the total number of clbs in a slot S_i can be calculated as

$$C(x_i, y_i, w_i, h_i) = \sum_{j=0}^{clk\_reg} \sum_{k=0}^{r-1} \beta_{ijk} \cdot (F(x_i + w_i) - F(x_i)) \tag{4}$$

where h_i which is the height of S_i and can be expressed as

$$h_i = \sum_{j=0}^{clk_reg} \sum_{k=0}^{r-1} \beta_{ijk} \tag{5}$$

The same resource finger-printing using piecewise linear functions can be done to the bram and dsp on the fpga this can then be used to determine the number of the resource within a slot.

III. Design

A. definition of optimization variables

To encode the MILP formulation the following binary and real variables are defined.

Variables related to the size, location and number of resources in slot S_i For each slot S_i

- N: set of reconfigurable regions
- S_i: slot i
- W: total width of the fpga fabric
- H: total height of the fpga fabric
- r: number of rows inside a single clock region
- F: set of forbidden regions
- F_k: forbidden region k $\in$ F
- x_i, y_i w_i, h_i $\in$ [Z] represent the bottom left coordinates, the width and the height of the rectangle respectively
- clb_i, $bram_i$ and dsp_i $\in$ [Z] represent the number of clb, bram and dsp between x_i and x_i + w_i in a single row respectively.
- clb_req_i, $bram\_req_i$ and dsp_req_i $\in$ [Z] represent the required number of clb, bram and dsp in S_i.
- β_{ijk} $\in$ [0,1] represents row k in clock region j for slot S_i.

Variables denoting the relationship between two slots
For two slots S_i and S_k

- γ_{ik} $\in$ [0,1] is a binary variable used to identify whether S_i is found on the left or on the right of S_k
  γ_{ik} = 1 if $x_i \leq x_k$ [i.e. S_i is on the left of S_k]
- θ_{ik} $\in$ [0,1] is a binary variable used to identify whether S_i is found on the top or bottom of of S_k
  θ_{ik} = 1 if $y_i \leq y_k$ [i.e. S_i is found below S_k]
- Γ_{ik} $\in$ [0,1] is used to denote if bottom right x coordinate of S_i is found to the right of the bottom left coordinate of S_k
  Γ = 1 if $x_i + w_i \geq x_k$
- η_{ik} $\in$ [0,1] is used to denote if bottom right x coordinate of S_k is found to the right of the bottom left coordinate of S_i
  η = 1 if $x_k + w_k \geq x_i$
- Ω_{ik} $\in$ [0,1] is used to denote if the top y coordinate of S_i is found above the lower y coordinate of S_k
  Ω = 1 if $y_i + h_i \geq y_k$
- Ψ_{ik} $\in$ [0,1] is used to denote if top y coordinate of S_k is found above the lower y coordinate of S_i
  Ψ = 1 if $y_k + h_k \geq y_i$
- Δ_{ik} $\in$ [0,1] is a binary variable which indicates interfernce between slots S_i and S_k.
  Δ_{ik} = 0 if there is no interference between the slots [i.e. not a single tile is shared between slots]

Variables denoting the relationship between S_i and forbidden region F_k.

- μ_{ik} $\in$ [0,1] is a binary variable used to identify whether S_i is found on the left or on the right of F_k
  μ_{ik} = 1 if $x_i \leq fx_k$ [i.e. S_i is on the left of F_k]
- ν_{ik} $\in$ [0,1] is a binary variable used to identify whether S_i is found on the top or bottom of of F_k
  ν_{ik} = 1 if $y_i \leq fy_k$ [i.e. S_i is found below F_k]
- $fbdn_1$ $\in$ [0,1] is used to denote if bottom right x coordinate of S_i is found to the right of the bottom left coordinate of F_k
  $fbdn_1$ = 1 if $x_i + w_i \geq fx_k$
- $fbdn_2$ $\in$ [0,1] is used to denote if bottom right x coordinate of F_k is found to the right of the bottom left coordinate of s_i
  $fbdn_2$ = 1 if $fx_k + fw_k \geq x_i$

- $\text{fbdn}_3 \in [0,1]$ is used to denote if the top y coordinate of S_i is found above the lower y coordinate of F_k
  $\text{fbdn}_3 = 1$ if $y_i + h_i \geq fy_k$
- $\text{fbdn}_4 \in [0,1]$ is used to denote if top y coordinate of F_k is found above the lower y coordinate of S_i
  fbdn_4 = 1 if $fy_k + fh_k \geq y_i$

B. Constraint definition

Slots for partial reconfiguration should fulfill the following constraints

- there must be enough resources within the slots
- A frame can not be shared between two reconfigurable partitions (no interference)
- static resources on the FPGA must not be included in the slots
- Left and right edges of slots must be placed in proper positions
- the amount of wasted resources should be minimized (Wasting DSPs is more expensive than BRAMs which in turn is more expensive than CLBs)
- Other optimizations such as lower wire length between slots or lower length to I/O etc... can be added as constraints

Semantics constraints

The following constraints ensure the soundness of some of the variables.

Constraint 1: : the x and y coordinates of S_i must be constrained not to exceed the boundaries of the fabric

$$\begin{aligned} x_i + w_i \leq W \\ y_i + h_i \leq H \end{aligned} \tag{6}$$

Constraint 2: The height of S_i must be the sum of chosen binary rows in each clock region

$$h_i = \sum_{j=0}^{clk\_reg} \sum_{k=0}^{r-1} \beta_{ijk} \tag{7}$$

Constraint 3: y_i must be constrained not to be greater than the lowest chosen row

$$y_i \leq \sum_{j=0}^{clk\_reg} \sum_{k=0}^{r} H - \beta_{ijk} \cdot (H - (k + (r-1) \cdot j)) \tag{8}$$

Constraint 4: rows of the same clock region in S_i must be contigious i.e., if β_{ij0} = 1 & β_{ij2} = 1 thenβ_{ij1} must also be equal to 1.

$$\beta_{ij(k+1)} \geq \beta_{ijk} + \beta_{ij(k+2)} - 1 \tag{9}$$

Resource constraints

These set of constraints ensure that each slot satisfies the resource requirements of the application. As an example, the number of clbs inside a slot is calculated using 4 where $F(x_i+w_i)$ - $F(x_i)$ is bounded below by 0 and above by U in such a way that

$0 \leq F(x_i+w_i) - F(x_i) \leq U$

1) linearization: In order to employ 4, which is a non linear function, as linear constraint it must be linearized. To linearize this function we define an intermediate real variable τ_{ijk}.

$\tau_{ijk} \in [R] \mid \tau_{ijk} = \beta_{ijk} \cdot (F(x_i+w_i) - F(x_i))$

$$C(x_i, y_i, w_i, h_i) = \sum_{j=0}^{clk\_reg} \sum_{k=0}^{r-1} \tau_{ijk} \tag{10}$$

$\tau_{ijk} \geq 0$

$\tau_{ijk} \leq \text{BIG\_M} \cdot \beta_{ijk}$

$\tau_{ijk} \leq F(x_i+w_i) - F(x_i)$

$\tau_{ijk} \geq F(x_i+w_i) - F(x_i) - (1 - \beta_{ijk})$

Constraint 5: Each slot must incorporate enough resources.

$$clb\_req_i = \sum_{j=0}^{clk\_reg} \sum_{k=0}^{r-1} \tau_{ijk} \tag{11}$$

Interference constraints

Interference can be between two slots *constriant 10*: A frame (tile) is the smallest reconfigurable physical region and it spans one clock region high and one resource type wide. A Reconfigurable Frame can not contain logic from more than one reconfigurable partition hence the boundaries of a slot S_i must be forced to fit in to clock region boundaries. This constraints is enforced by forcing all the rows in a clock region to also be included in the slot if atleast one is included i.e., if one row in a clock region j is part of a slot then the remaining rows within the clock region j must also be forced to be part of the same slot to satisfy this constraint.

To help us set this constraint we define an intermediate variable l_j which is the sum of β_{ijk} in clock region j

$$l_j = \sum_{k=0}^{r-1} \beta_{ijk} \tag{12}$$

then the following constraint will force the slot boundaries to be aligned with the clock region boundaries by forcing β_{ijk} to become part of the slot if at least one row with in the same clock region becomes part of slot S_i

$$\beta_{ijk} \geq \sum_{k=0}^{r-1} (l_j - beta_{ijk})/(r-1) \tag{13}$$

Constraint 6: : two regions S_i and S_k are said to be not interfering under the following conditions

if $x_i \leq x_k$ and $y_i \leq y_k$ **then**
  $x_i + w_i < x_k$ or $y_i + h_i < y_k$
else if $x_i \geq x_k$ and $y_i \geq y_k$ **then**
  $x_i + w_k < x_i$ or $y_k + h_k < y_i$

else if $x_1 < x_k$ and $y_i > y_k$ **then**
$x_i + w_i < x_k$ or $y_k + h_k < y_i$
else
$x_k + w_k < x_k$ or $y_i + h_i < y_k$
end if

This condition can be encoded into a set of MILP constraints as follows

$$\begin{aligned}
&\delta_{ik} \geq \gamma_{ik} + \theta_{ik} + \Gamma_{ik} + \Omega_{ik} - 3 \\
&\delta_{ik} \geq (1 - \gamma_{ik}) + \theta_{ik} + \eta_{ik} + \Omega_{ik} - 3 \\
&\delta_{ik} \geq \gamma_{ik} + (1 - \theta_{ik}) + \Gamma_{ik} + \Psi_{ik} - 3 \\
&\delta_{ik} \geq (1 - \gamma_{ik}) + (1 - \theta_{ik}) + \eta_{ik} + \Psi_{ik} - 3 \\
&\delta_{ik} = 0
\end{aligned} \tag{14}$$

Interference with Forbidden slots

Constraint 7: : As stated before forbidden regions are also modeled as a normal slots hence the constraint for non intereference between a slot S_i and a forbidden region F_k can be set in the same way as done in constraint 11

$$\begin{aligned}
&\delta_{ik} \geq \mu_{ik} + \nu_{ik} + fbdn_1 + fbdn_3 - 3 \\
&\delta_{ik} \geq (1 - \mu_{ik}) + \nu_{ik} + fbdn_2 + fbdn_3 - 3 \\
&\delta_{ik} \geq \mu_{ik} + (1 - \nu_{ik}) + fbdn_1 + fbdn_4 - 3 \\
&\delta_{ik} \geq (1 - \mu_{ik}) + (1 - \nu_{ik}) + fbdn_2 + fbdn_4 - 3 \\
&\delta_{ik} = 0
\end{aligned} \tag{15}$$

IV. Experiemtnal Results

A. Experimental Setup

How was the experiment implmented i.e, with what kind of prog. langaguge, on what optimization tool, on what platforms... What is the system being tested for ? What is the compositon of the synthethic task suite used for testing. What type of FPGAs are modeled ? What are the challenges when switiching between models.

The system is tested for Exec time Vs Num of Reconfigurable regions and Exec time Vs %of resources used by the system on both zynq and virtex

The system is also tested for % of wasted resources (coeficients used to set allowed percentage of resources to be wasted %)VS Exec time on both virtex and zynq. The average wasted resources are also reported when not imposing these constraints and what effect it has on exec time and feasibility in general. my guess is that upto a certain point (upto a certain % of utilization of resources) not imposing these constraints improves the exec time but after the utilization increases (i.e., more applications require more resources) not imposing these coeffecients leads to infeasible constrants. This has to be tested with a carefully designed test suit which will test the limist of the platform.

Finally a case study on a real application on Zynq.

this non linear function can be converted into a linear form according to [**?**] in the following form

$$BRAM(x_i, w_i) = \sum_{k=0}^{H-1} (\beta_k L + \tau) subject to \tau \geq m(x_i, w_i) - \beta_k L - U(1 - \beta_k \tag{16}$$

where L and U represent lower and upper bounds of $m(x_i, w_i)$ i.e. $L \leq m(x_i, w_i) \leq U$