Notes for the FloorPlanning Paper

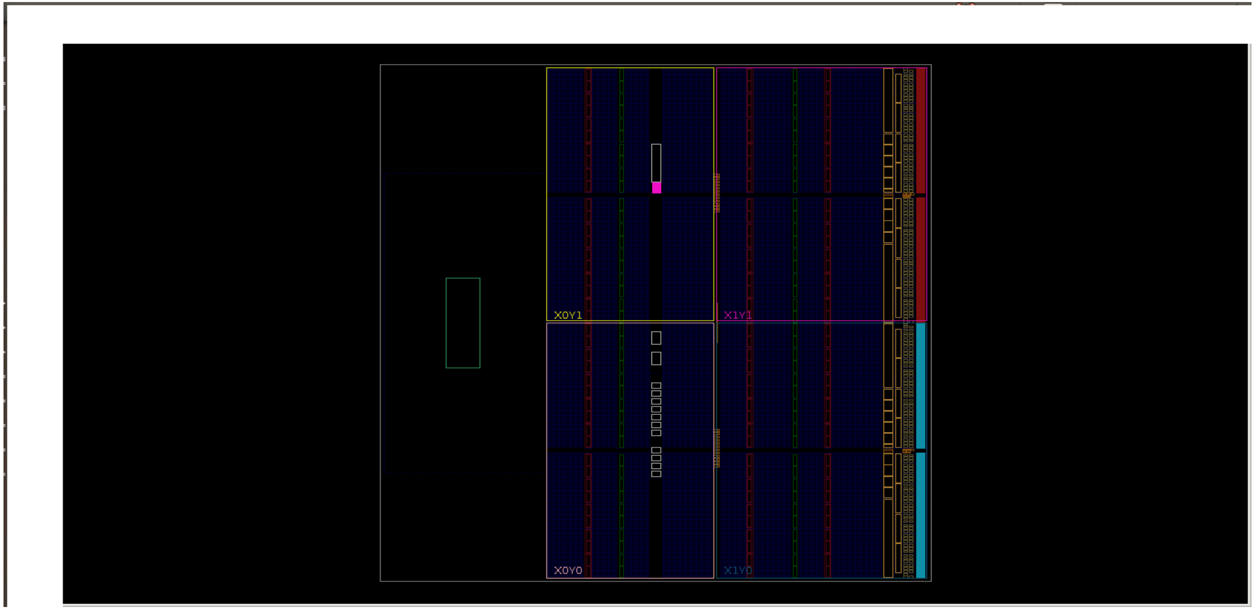
1 Introduction

The conventional approach for generating FPGA floorplans usually involves two steps. First, all the possible slots are enumerated. This is done by starting a scan on the fpga fabric from the bottom left corner and lisitng all the possible rectangles which contain all the necessary resources for the respective slots. Then some sort of heuristics/optimization is applied to choose the best ones from the set of possible slots.
Our approach instead focuses on direct ILP/MILP modeling of the FPGA floorplanning problem and trying to find the appropriate slots based on the defined constraints.

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 as an input takes 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).

2 FPGA resource mapping

Consider the resource layout in one of the quadrants of xc7z015 FPGA in the picture below



The number of BRAMs on the x axis of the FPGA fabric is expressed as

$$A(x) = \begin{cases} 0, & \mathbf{0{<}x{<}4}, \\ 1, & \mathbf{4{\leq}x{<}18}, \\ 2, & \mathbf{18{\leq}x{<}25}, \\ 3, & \mathbf{25{\leq}x{<}W}, \end{cases} \tag{1}$$

meanwhile the number of BRAMs on the y axis equals

$$B(y) = \begin{cases} y\%5, & \mathbf{0{\leq}y{<}H}, \end{cases} \tag{2}$$

The total number of BRAMs inside a slot S$_i$ equals to

$$BRAM(x_i, y_i, w_i, h_i) = (A(x_i + w_i) - A(x_i)) \cdot (B(y_i + h_i) - B(y_i)) \tag{3}$$

The number of DSPs on the x axis of the FPGA fabric is expressed as

$$C(x) = \begin{cases} 0, & \mathbf{0{<}x{<}7}, \\ 1, & \mathbf{7{\leq}x{<}22}, \\ 2, & \mathbf{22{\leq}x{<}W}, \end{cases} \tag{4}$$

meanwhile the number of DSPs on the y axis equals

$$D(y) = \begin{cases} y\%5, & \mathbf{0{\leq}y{<}H}, \end{cases} \tag{5}$$

The total number of DSPs inside a slot S$_i$ equals to

$$DSP(x_i, y_i, w_i, h_i) = (C(x_i + w_i) - C(x_i)) \cdot (D(y_i + h_i) - D(y_i)) \tag{6}$$

The number of CLB on the x axis is expressed

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

The number of CLB on the y axis is expressed as

$$G(y) = \Big\{ y, \quad \mathbf{0{\leq}y{<}H} \tag{8}$$

The total number of CLBs inside a slot S$_i$ equals to

$$CLB(x_i, y_i, w_i, h_i) = (F(x_i + w_i) - F(x_i)) \cdot (G(y_i + h_i) - G(y_i)) \tag{9}$$

where

$$y_i \in \{0, H - 1\} \quad and \quad x_i \in \{0, W - 1\}$$

3 Design

3.1 optimization variables

To encode the ILP/MILP formulation the following binary and real variables are defined.
For each slot S$_i$

- x$_i$, y$_i$ w$_i$, h$_i$ represent the bottom left coordinates, the width and the height of the rectangle respectively

For two slots S$_i$ and S$_k$

- γ_{ik} ∈ [0,1] is a binary variable used to identify whether S$_i$ is found on the left or the right of S$_k$
  γ_{ik} = 1 if x$_i$ ≤ x$_k$ [i.e. S$_i$ is on the left of S$_k$]
- θ_{ik} ∈ [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$_i$ ≤ y$_k$ [i.e. S$_i$ is found below S$_k$]
- Δ_{ik} ∈ [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]
- D$_w$, B$_w$ and C$_w$ represent wasted DSPs, BRAMs and CLBs in S$_i$ respectively
- α_i is a real variable that is used to express the bound on the amount of wasted resources in a slot S$_i$
- ρ and ν are vectors which contain the x and y coordinates of all the forbidden columns and rows on the FPGA fabric respectively

- η_{ik} is a real variable that expresses the bound on the wirelength between S$_i$ and S$_k$

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

constraint 1: Each slot must incorporate enough resources.
For slot S$_i$

$$\begin{aligned} CLB(x_i, y_i, w_i, h_i) &\geq C_i \\ DSP(x_i, y_i, w_i, h_i) &\geq D_i \\ BRAM(x_i, y_i, w_i, h_i) &\geq B_i \end{aligned} \tag{10}$$

Constraint 2: 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. Two slots S$_1$ {x$_1$, y$_1$, w$_1$, h$_1$} and S$_2${ x$_2$, y$_2$, w$_2$, h$_2$ } do not interfere i.e. do not have at least one common tile between them, under the following conditions

```
if $x_1 \leq x_2$ and $y_1 \leq y_2$ then
  $x_1 + w_1 < x_2$ or $y_1 + h_1 < y_2$
else if $x_1 \geq x_2$ and $y_1 \geq y_2$ then
  $x_2 + w_2 < x_1$ or $y_2 + h_2 < y_1$
else if $x_1 < x_2$ and $y_1 > y_2$ then
  $x_1 + w_1 < x_2$ or $y_2 + h_2 < y_1$
else
  $x_2 + w_2 < x_2$ or $y_1 + h_1 < y_2$
end if
```

Hence using the conditions above Δ_{ik} which is used to test interference between slots S$_i$ and S$_k$ is expressed as

$$\begin{aligned} \Delta_{ik} = \gamma_{ik}(1 - (x_k - x_i - w_i)) \cdot \theta(1 - (y_k - y_i - h_i)) \quad &+ \\ (1 - \gamma_{ik})(1 - (x_i - x_k - w_k)) \cdot (1 - \theta_{ik})(1 - (y_i - y_k - h_k)) \quad &+ \\ \gamma_{ik}(1 - (x_k - x_i - w_i)) \cdot (1 - \theta_{ik})(1 - (y_i - y_k - h_k)) \quad &+ \\ (1 - \gamma_{ik})(1 - (x_i - x_k - h_k)) \cdot \theta(1 - (y_k - y_i - h_i)) & \end{aligned} \tag{11}$$

Therefore for N slots on an FPGA it can be said that slot S$_i$ faces no interferance from the N-1 slots if

$$\sum_{k=1}^{N-1} \Delta_{ik} = 0 \tag{12}$$

constraint 3: Global resources, clock resources (central clock column, BUFG, BUFR, MMCM etc...), static components (BSCAN, ICAP, XADC etc...) must not be included in the rectangles. The ρ and ν vectors which contain the x and y coordinates of these resources on the FPGA fabric. These resources are not included in a slot S$_i$ if

$$\begin{aligned} (x_i, x_i + w_i) \notin \rho \\ (y_i, y_i + h_i) \notin \nu \end{aligned} \tag{13}$$

constraint 4: The left and right edges of the rectangles must be placed between CLB-CLB, CLB-BRAM or CLB-DSP and not between two interconnect columns (INT-INT).

constraint 5: The number of wasted resources in a slot S$_i$ must be bounded. Resources are considered wasted if they are included in a slot while not being used. The wasted resources D$_w$, B$_w$ and C$_w$ in slot S$_i$ can be calculated as

$$\begin{aligned} D_w &= D_i - DSP(x_i, y_i, w_i, h_i) \\ B_w &= B_i - BRAM(x_i, y_i, w_i, h_i) \\ C_w &= C_i - CLB(x_i, y_i, w_i, h_i) \end{aligned} \tag{14}$$

Hence the constraint on wasted resources in S$_i$ can be expressed as

$$\alpha_i \leq (\phi_d \cdot D_w) + (\phi_b \cdot B_w) + (\phi_c \cdot C_w) \tag{15}$$

where ϕ_d, ϕ_b and ϕ_c are the weights which signify the importance of resources. All FPGA resources are not equally valuable hence wasting DSPs is worse than BRAMs which in turn is worse than CLBs. Accordingly

$$\phi_d > \phi_b > \phi_c \tag{16}$$

constraint 6: Other constraints such as minimal wire length between slots (to reduce routing delay and power consumption) can be used as constraints to choose the optimal slots