SMT Report

p.soleimanybaraijany@studio.unibo.it[1],
maryam.salehi@studio.unibo.it[2], and
shadi.farzankia@studio.unibo.it[3]

[1]University of Bologna

September 2022

1 Introduction

Very large-scale integration(VLSI) is the process of embedding numerous transistors onto a silicon semiconductor microchip.

VLSI technology emerged in the late 1970s when microcomputer chips were developing. The last few decades have witnessed a remarkable growth in the electronics industry thanks to the advancements in the VLSI technology. Presently, cellular communication and smartphones afford unprecedented processing capabilities and portability due to technological improvements and it is forecasted that this trend will continue as there is an ever-increasing demand for state-of-the-art devices. In VLSI design, maintaining a small area and high performance have always been two conflicting constraints considered by integrated circuit designers.

In this project[1], we try to propose a solution to the size problem of VLSI circuits. A certain number of circuits are embedded in a plate with a fixed width and the ultimate goal is to minimize the height of the plate.

Four different approaches including constraint programming (CP), Propositional Satisfiability (SAT), Satisfiability Modulo Theories (SMT), and linear programming (LP) have been utilized to solve the problem and the results of all approaches have been compared and analyzed to find the best solution.
This report explains how Satisfiability Modulo Theories (SMT), was used to solve the VLSI problem.

2 Instance Format and Solution Format

In this section, the format of input VLSI instances as well as the output solutions are presented.

2.1 Instance Format

Input instances are defined as follows:

w
n
$x_0 \quad y_0$
$x_1 \quad y_1$
...

Where:

w: width of the plate
n: number of circuits to be embedded
x_i: the horizontal dimension of the ith circuit
y_i: the vertical dimension of the ith circuit

For example, a file with the following lines:

9
5
3 3
2 4
2 8
3 9
4 12

describes an instance in which the silicon plate's width is 9, and we need to place 5 circuits, with the dimensions 3×3, 2×4, 2×8, 3×9, and 4×12. Figure 1 shows the graphical representation of the instance.

2.2 Output format

After solving the problem using the input instances, the output can be represented as:
$w \quad max\_height$
n
$x_0 \quad y_0 \quad pos_{x_0} \quad pos_{y_0}$
$x_1 \quad y_1 \quad pos_{x_1} \quad pos_{y_1}$
...

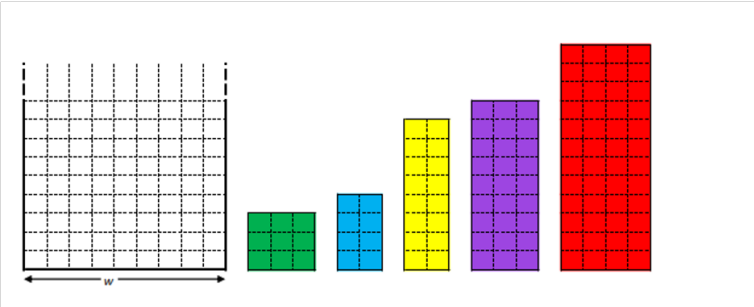

Figure 1: Graphical representation of an instance

where:

w, n, xi, yi : same as input instance
max_height: The optimal height of the plate
pos_{x_i}= horizontal coordinate of the bottom left point of the ith circuit
pos_{y_i}= vertical coordinate of the bottom left point of the ith circuit
For example, a possible solution for the instance presented in figure 1 can be as follows:

```
9    12
5
3    3    4    0
2    4    7    0
2    8    7    4
3    9    4    3
4    12    0    0
```

Which states that the maximum height is 12 and the left bottom corner of the 3×3 circuit is at (4,0). The output has been depicted in figure 2.

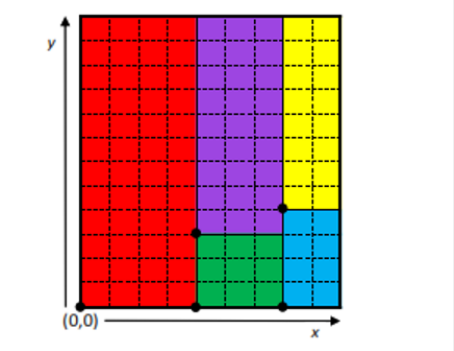

Figure 2: Graphical representation of one solution

3 Preliminary Concepts and Modeling

Satisfiability Modulo Theories (SMT) generalizes Boolean Satisfiability (SAT). It is a decision problem for logical first order formulas with respect to combinations of background theories such as: arithmetic, bit-vectors, arrays, and uninterpreted functions. An SMT solver is a tool for deciding the satisfiability (or dually the validity) of formulas. SMT solvers enable applications such as extended static checking, predicate abstraction, test case generation, and bounded model checking over infinite domains.

3.1 Variables and Domains

3.1.1 Maximum and Minimum Height

The width of the plate is fixed and is equal to w, However, the height of the plate is a variable that should satisfy two criteria:

1. The height should be determined such that the area of plate is minimized.
2. We should be able to embed all the circuits on the plate without overlapping.

Considering the criteria above, the plate's minimum height is the maximum of the maximum width of the circuits embedded on the plate and the sum of the areas of all the circuits divided by width of the plate.

$$min\_height = max(y_{max}, \frac{\sum_{i=1}^{i} x_i * y_i}{w})$$

The height of the plate cannot be less than the maximum of the widths of the circuits embedded on the plate. This is because in the best case, all circuits can be placed in only one row. However, in most cases we need to have more than one row and a good estimation of the minimum height can be achieved by dividing the sum of the areas of all the circuits by width of the plate which is fixed.
In order to determine the maximum height, one needs to consider the worst case in which all of the circuits have the maximum width and only one circuit can be embedded in each row. As a result, the maximum height is equal to the sum of the maximum width of the circuits.

$$max\_height = \sum_{i=1}^{n} y_i$$

The variable height ranges from min_height to max_height. we seek to minimize the height.

$$min\_height \leq height \leq max\_height$$

3.1.2 Position of the Circuits

To determine where each circuit is located on the plate, only the coordinates of its left-bottom corner are kept. Variables pos_x and pos_y denote the horizontal and vertical coordinates of the left-bottom corner of the circuits respectively.
pos_x varies between 0 and width minus $min(x)$, where $min(x)$ is the minimum of the lengths of the circuits.
pos_y varies between zero and maximum height minus $min(y)$ where $min(y)$ is the minimum of the widths of the circuits.

$$0 \leq pos_x \leq w - x_{min}$$
$$0 \leq pos_y \leq max\_height - y_{min}$$

4 Constraints

Three constraints were used to solve the VLSI problem using SMT approach.

4.1 The Biggest Rectangle Constraint

The biggest rectangle constraint has been used to break the symmetry. By putting the circuit with the biggest area at coordinates $(0, 0)$, vertical, horizontal, and 180 degrees symmetry can be broken. In the beginning, circuits are ordered based on their areas and the one with the biggest will have the index 0. This circuit with index 0 will be put on $(0, 0)$.

$$pos_{x_0} = 0 \quad \wedge \quad pos_{y_0} = 0$$

4.2 Boundaries Constraints

It is important to make sure that all the circuits are embedded within the plate. To satisfy this criterion two constraints were introduced to limit the coordinates of the bottom left corner of the circuits:

$$\forall i \in 1..n : pos_{x_i} \geq 0 \wedge pos_{y_i} \geq 0 \wedge pos_{x_i} \leq w - min(x) \wedge pos_{y_i} \leq height - min(y)$$

constraint for checking that end points of each circuit do not exceed plate boundaries:

$$\forall i \in 1..n : pos_{x_i} + x_i \leq w \wedge pos_{y_i} + y_i \leq height$$

4.3 Non-Overlapping Constraints

Any pairs of rectangular circuits should be non-overlapping. If i and j refer to two different circuits, (xi,yi) and (xj,yj) denote the widths and heights of these circuits respectively, and (pos_{x_i}, pos_{y_i}) and (pos_{x_j}, pos_{y_j}) show the coordinates of the bottom left corner of the two circuits. If circuits i and j do not overlap, at least one of the following linear inequalities should hold:

$$\forall i, j \in 1..n \wedge i < j : pos_{x_i} + x_j \leq pos_{x_j} \vee pos_{x_i} - x_j \geq pos_{x_j} \vee pos_{y_i} + y_i \leq pos_{x_j}$$

$$\vee pos_{y_i} - y_j \geq pos_{y_j}$$

4.4 Cumulative Constraints

The main concept of modeling is that this problem can be considered a task scheduling problem and hence it is possible to use the cumulative constraint. the problem of putting rectangles (circuits) on a plate can be considered a task scheduling problem. Hence, it is possible to use the cumulative constraint. Two different views of the problem at hand can be assumed. If we consider circuits as activities in the task scheduling problem, from one point of view, duration is the vertical lengths of the circuits(y_i), the amount of resource each activity needs, is the horizontal lengths of the circuits(x_i), and the total number of resources is equal to the width of the plate (w).
From another point of view, duration is the horizontal length of the circuits(x_i), the amount of resource needed by each activity is the vertical length of the circuits(y_i), and the total number of resources is equal to the height of the plate.
Although adding the two cumulative constraints did not change the number of solved instances, it decreased the execution time.

$$cumulative(pos_y, y, x, w)$$

$$cumulative(pos_x, x, y, height)$$

5 Rotation

The second variant of the VLSI problem concerns the case in which rotation is admissible. Hence, each circuit can be rotated by 90 degrees and its width and height can be swapped. In order to take into take the rotation into consideration, we used three variables $circuit_x$, $circuit_y$, and rot. Variable rot is a Boolean array with a size equal to the number of circuits.
Variable $circuit_x$ and $circuit_y$ are the representation of x array and y array respectively, considering the rotation.

The solver decides whether each element of the rotation array should be filled with zero or one. If the solver chooses zero in the rotation array for one circuit, it means the height and width of the circuit will not be swapped and the $circuit_x$ and $circuit_y$ of that circuit are the same as x and y respectively.
If the solver chooses one in the rotation array for one circuit, it means the height and width of the circuit will be swapped and the $circuit_x$ and $circuit_y$ of that circuit are y and x respectively.

$$\forall i \in 1..n : circuit_x = (x[i] * (1 - rot[i]) + y[i] * rot[i]))\wedge$$
$$circuit_y = (y[i] * (1 - rot[i]) + x[i] * rot[i]))$$

When rotation is allowed, constraints introduced in section 4 should be revised as follows:

5.1 The Biggest Rectangle Constraint

The biggest rectangle constraint remains unchanged when the rotation of circuits is admissible. This is due to the fact this constraint is based on the areas of circuits that do not alter when we swap the widths and heights of the rectangular circuits. Two criteria determine the necessity and possibility of rotation:

$$pos_{x_0} = 0 \wedge pos_{y_0} = 0$$

1. If the width and height of a circuit are the same, rotation is not necessary.
2. If the height of a circuit is bigger than w (width of the plate), rotation is not allowed.

   [If (Or ($circuits_x$[i] == $circuits_y$[i], $circuits_y$[i] > w), rot[i] == 0, True) for i in range(n))]

5.2 Boundaries Constraints

It is important to make sure that all the circuits are embedded within the plate. To satisfy this criterion two constraints were introduced to limit the coordinates of the bottom left corner of the circuits:

$\forall$ i $\in$ 1..n: $pos_{x_i} \geq 0 \wedge pos_{y_i} \geq 0 \wedge pos_{x_i} \leq$ w - $min(circuit_x) \wedge pos_{y_i} \leq height - min(circuit_y)$

constraint for checking that end points of each circuit do not exceed plate boundaries:

$\forall i \in$ 1..n: $pos_{x_i} + circuit_{x_i} \leq$ w $\wedge$ $pos_{y_i} + circuit_{y_i} \leq$ height

5.3 cumulative Constraints

The main concept of modeling is that this problem can be considered a task scheduling problem and hence it is possible to use the cumulative constraint.

the problem of putting rectangles (circuits) on a plate can be considered a task scheduling problem. Hence, it is possible to use the cumulative constraint. Two different views of the problem at hand can be assumed. If we consider circuits as activities in the task scheduling problem, from one point of view, duration is the vertical lengths of the circuits(y_i), the amount of resource each activity needs, is the horizontal lengths of the circuits(x_i), and the total number of resources is equal to the width of the plate (w).
From another point of view, duration is the horizontal length of the circuits(x_i), the amount of resource needed by each activity is the vertical length of the circuits(y_i), and the total number of resources is equal to the height of the plate.
Although adding the two cumulative constraints did not change the number of solved instances, it decreased the execution time.

$$cumulative(pos_y, circuit_y, circuit_x, w)$$

$$cumulative(pos_x, circuit_x, circuit_y, height)$$

6 Results

Z3 is a new and efficient SMT Solver freely available from Microsoft Research used in various software verification and analysis applications. It has been used to solve the VLSI problem using SMT approach. The model was able to solve 31 out of 40 instances in without rotation version and 17 instances in rotation model.

References

[1] A. Ratna S. Banerjee and S. Roy. "Satisfiability modulo theory based methodology for floorplanning in VLSI circuits," in: *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, (2016), pp. 91–95. DOI: 10.1109/ISED.2016.7977061.