



**Politecnico
di Torino**

Report

Matteo Battilana Numero Matricola
Salvatore Gabriele La Greca s281589
Giovanni Pollo s290136

1 Explanation

The idea of the code is to start by inserting one functional unit per type, and the chosen one is the slowest and the smallest. After that we start the while loop in order to explore some possibilities and find the best one. The used logic is based on these steps:

1. Filling the area
2. Executing minimum latency area constrained algorithm
3. Computing latency
4. Saving the latency if it is the best

This while loop stops when we iterate with two constraints that are the exactly the same. In fact, when we reach that condition, it means that our algorithm has reached it's best state.

The trick we used is to perform a sharing inside the minimum latency area constrained algorithm. In fact, every time we check the maximum number of unit needed in order to obtain that scheduling. For example if we launch the scheduling with 30 MUL but we can only run 15 MUL in parallel, we are wasting area. By doing this, every loop of the while we are able to save some area that can be invested in changing a unit with a faster one (if needed) or in adding a unit of another kind.

An important decision was taken in order to satisfy the tradeoff precision-speed. When we execute the step of filling the area we used a multiplier. When we have a lot of free area (SCRIVERLO MEGLIO) we set the multiplier to 2. By doing this we increment the number of unit by 2. When we are reaching the area limit we start incrementing by 1. This allows to improve performance but not losing much in granularity of the algorithm. The value of the multiplier can be changed based on the precision-speed ratio we would like to obtain. This allows the algorithm to be reusable and suitable for different use cases.

The minimum latency area constrained algorithm is taken from the laboratory exercises with little adaptation for managing the presence of different speed and area units.

The algorithm to compute the latency uses the last scheduled node time.

The score is then computed and printed (the minimum latency is computed using the ASAP algorithm of the laboratory).