

The goal of this project is to implement an algorithm in TCL that takes in input an area constraint and a dataflow graph and allocates a number of resources from an RTL library in order to obtain the best latency within the received area constraints.

The implementation adopted consists of initially assigning one resource per operation needed, the smallest and latest. In case there are multiple functional unit associated to an available operation then the smallest one in terms of area is selected.

After this first analysis, the optimization procedure starts:

1. A function is called in order to produce a schedule for the loaded DFG and it returns the overall latency and a list, named 'da_incrementare', of required operations that aren't scheduled because the needed resources were already used in other nodes.
2. The optimization algorithm modifies 'lista_risorse', which is a list of resources used for the scheduling: the procedure modifies this list n times, where n is the number of elements in the list of pairs <operation, used> named 'resources_to_incr'. This list contains the operations needed by the DFG and the flag 'used' initially set to zero. This flag is set to zero if the operation is analyzed for the first time. Since in the first analysis the functional unit with less area have been allocated, is needed to evaluate for each operation if there is a version of the fu associated to that operation that gives a better overall latency. So when used='0' we have to evaluate which version of the fu gives best latency with less area, otherwise is used='1' and the best fu it's already known.
3. The procedure scans the list 'resources_to_incr': if the flag is zero it evaluates the best fu (min latency, area constrained), if flag is 1, it evaluates which fu added to the lista_risorse_test gives the best overall latency. In this procedure a list of pairs <best_fu, latency> is created.

This procedure is repeated a number of time equal to the number of resources in 'resources_to_incr'.

At the end of this process the list of pairs <best_fu, latency> is scanned in order to find the best latency. The corresponding fu associated to the best latency is added in 'lista_risorse'.

The DFG is scheduled with the updated 'lista_risorse': if the operation associated to the fu is no more present in list 'da_incrementare', the operation is removed from 'resources_to_incr' and if the total occupied area is lower than the constraint given the process is repeated.

The optimization procedure repeats until the area constraint is met or when there's no improvement in latency.

In the last case (no improvement in latency, but some available area) a local minimum can be encountered. A last analysis is computed to avoid this problem: the list, that contains operations that can be execute but whose resources are not available, is scanned. One resource per time is allocated until there's no more area available and it's evaluated if there is a latency improvement.

At the end of this optimization procedure 'lista_risorse' is returned, which contains:

1. list of pairs < node_id , start_time >
2. list of pairs < node_id , fu_id >
3. list of pairs < fu_id , #allocated >

