An Open-Source Design and Validation Platform for Reconfigurable Systems

Alessandra Bonetto, Andrea Cazzaniga, Gianluca Durelli, Christian Pilato, Donatella Sciuto, Marco D. Santambrogio

Politecnico di Milano - Dipartimento di Elettronica ed Informazione Milano (Italy)
{bonetto, acazzaniga, pilato, sciuto, santambr}@elet.polimi.it, gianluca.durelli@mail.polimi.it

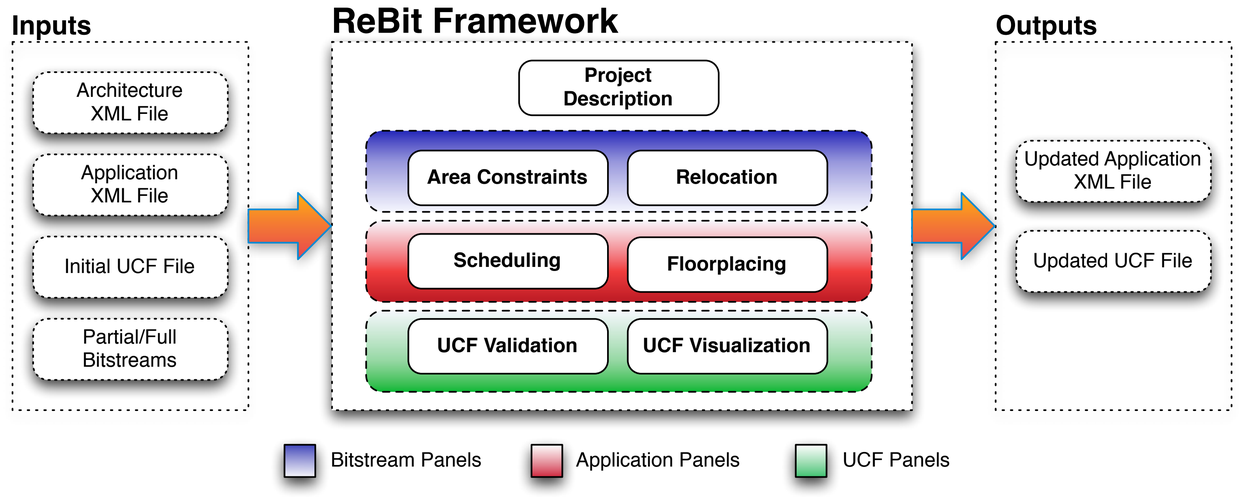
Abstract

Reconfigurable computing is a hot topic for research, as the possibilities and the technology offered by the reconfigurable devices improve year after year both in terms of available configurable logic resources and the possibilities offered to exploit them. This has led CAD tools to grow both in complexity and effectiveness. The expertise required to develop and test a complete system-on-chip using vendors tools has subsequently increased, forcing some designers to create their own tools as support to official development flows. Within this field quite few works have been developed, with respect to the huge effort that has been spent in the exploitation of architectural designs.
ReBit is an open-source tool able to help the designer in exploring different placement solutions in the architecture refinement process and in testing the correct execution of an application on a real device.

Framework Interfaces and Panels

ReBit is composed by several different panels, that make possible for the user to manage bitstreams and UCF and to schedule and floor place an application, starting from its task graph.

- **Project description:** panel to manage input files.
- **Area Constraints:** panel to check if any combination of partial bitstreams generates an area conflict.
- **Relocation:** used to compute and visualize all the possible placements to relocate a partial bitstream.
- **UCF validation:** this panel checks an UCF file for common errors.
- **UCF visualization:** graphical visualization of UCF areas constraints.
- **Scheduling:** schedules an partitioned application TG using a specific algorithm.
- **Floorplacing:** floorplaces a scheduled application to generate a compliant UCF.



Applicative Scenarios

Validation of a manually generated system
Already generated bitstreams and/or UCF files are checked for possible errors.

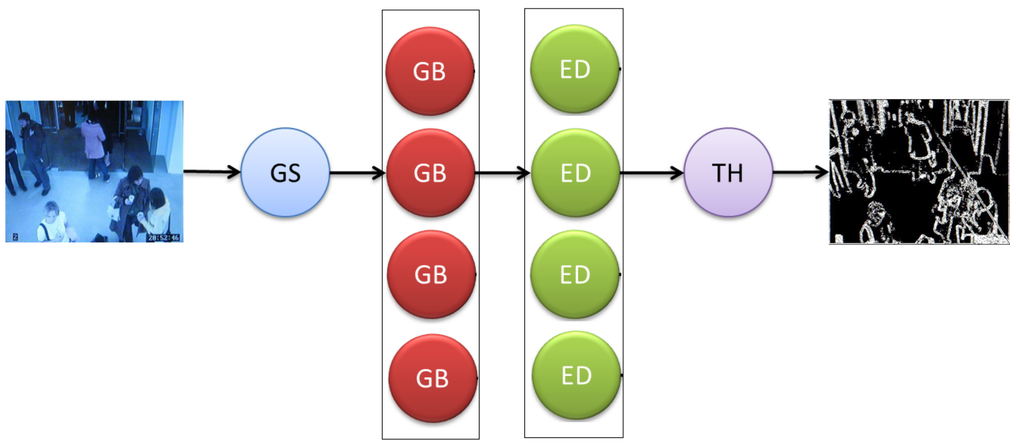
Validation of an application
An hardware application is tested to verify whether the code execution is correct on a real device.

Custom algorithms verification
A custom scheduling or floor placing algorithm is easily integrated inside ReBit thanks to the presence of several API and data structures, e.g. to read an XML file or to update an UCF file.

Use Cases

Testing Application : Canny edge detection algorithm. The application execution is composed of four main steps: a gray scale conversion algorithm (GS), a Gaussian blur filter (GB), an edge detection filter using Laplace kernel (ED) and finally a threshold phase (TH).

Partitioning : We have decided to split the two filters (GB and ED) into four parallel nodes that operate separately on each quarter image. The application is thus partitioned into 10 different tasks, corresponding to 10 different task graph nodes.



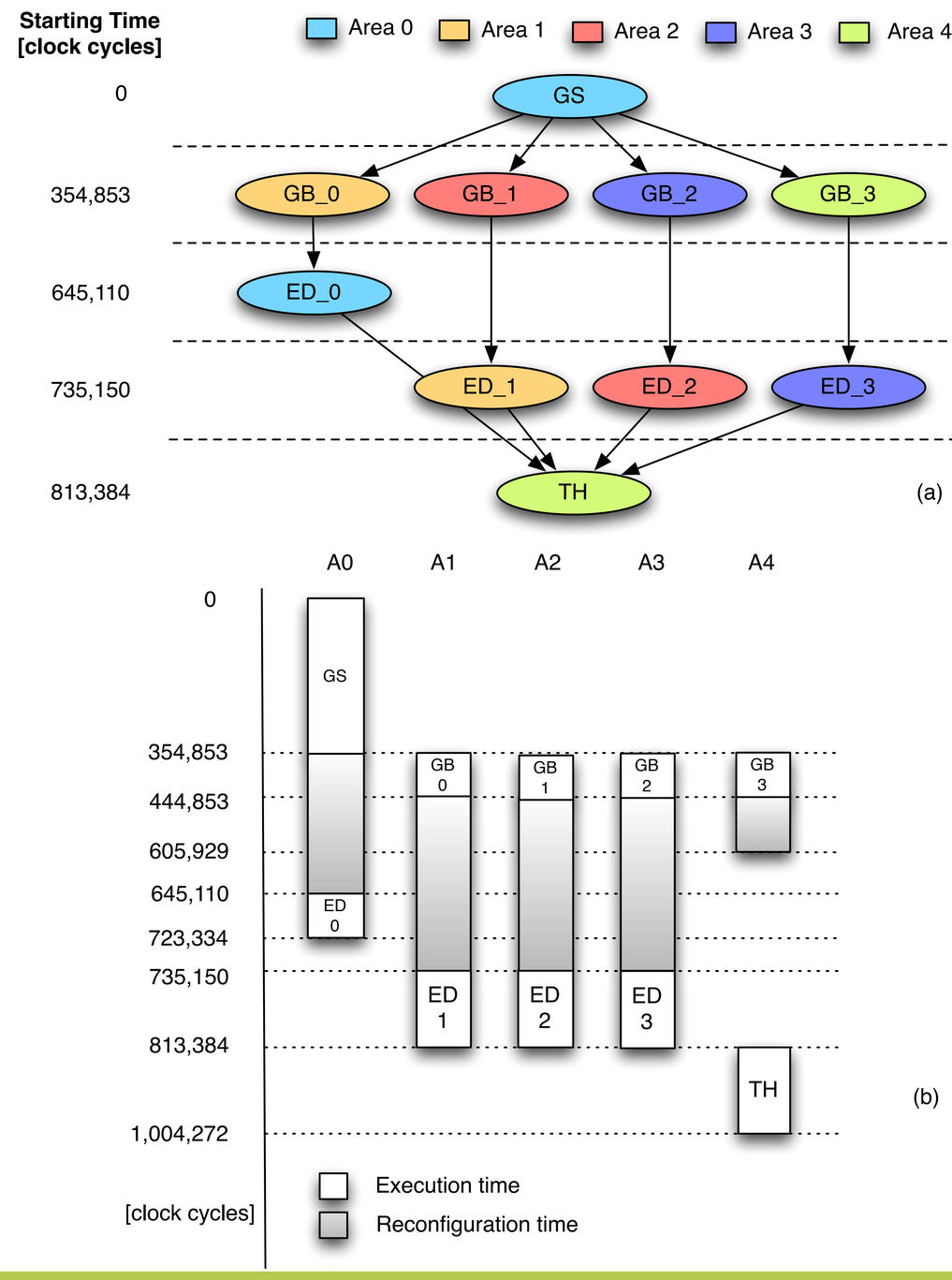
Input data : Estimated execution, reconfiguration times and area occupations.

Task	Execution Time [cycles]	Reconfig Time [cycles]	Slices
GS	354853	220376	5120
GB_i	90434	310745	3613
ED_i	78234	290297	2723
TH	190754	160176	3224

New Algorithm Pseudocode : Reconfigurable Aware Scheduling

1: **while** *there are still tasks to be scheduled* **do**
2: $area = select(AREAS)$
3: **if** $free(area)$ **then**
4: $time = ASAP(t, area)$
5: **else**
6: $reconfigure(area)$
7: $time = ASAP(t, area)$
8: **end if**
9: **end while**

Results : Starting time and tasks mapping of each task, using the Reconfigurable aware algorithm with the availability of 5 reconfigurable areas. Note that the number of available areas is an algorithm input.



ReBit is available at http://code.google.com/p/rebit/