QoS driven dynamic partial reconfiguration: A tracking case study

Julien Mazuet[1,2], Dominique Heller[2], Catherine Dezan[2], Michel Narozny[1], Jean-Philippe Diguet[2]

[1]*Thales LAS-France*, Élancourt, France
[2]*Lab-STICC, CNRS, Université de Bretagne Sud / Université de Bretagne Occidentale*, Lorient / Brest, France

1. Context

- Radar applications (such as radar tracking) require increasingly computing resources due to the large amounts of data to process.
- The number of antenna elements (n) increases the hardware required for every channel processing (Figure 1).
- The array processing is even more complex (e.g. STAP matrix size is n^2).
- Different versions of each algorithm exist.
- BUT not all the radar algorithms have to run during the whole mission.

→ **Idea: Online hardware adaptation can save resources while giving flexibility. For radar tracking, the reconfiguration should rely on missions indicators.**

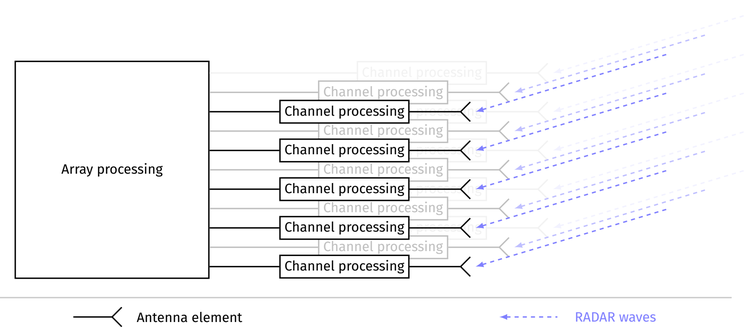

Figure 1: Schematic representation of the processing system for an AESA antenna

2. Methodology

Proposal: Make the reconfiguration decision relying on the quality of service (QoS) indicators collected during the mission.
Challenge: A clear separation of concerns (depicted in Figure 2) to make the signal processing experts (QoS indicator choice) and the SoC expert (HW/SW design) working efficiently together without unnecessary knowledge acquisition.

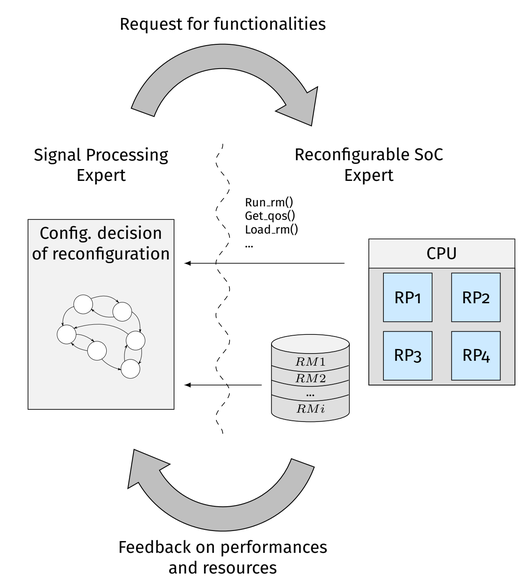

Figure 2: Schematic representation of the separation of concerns in the QoS driven dynamic partial reconfiguration (DPR) methodology

The signal processing expert:

- Defines the different algorithms to implement, and their different versions
- Designs the reconfiguration decision process, regarding QoS indicators

The system on a chip (SoC) expert:

- Designs the reconfigurable system
- Gives the signal processing expert feedback about performances and resources usage.
- Creates an API for the signal processing expert to access the QoS and trigger the reconfiguration

The target architecture is a SoC FPGA that includes (Layout on Figure 3):

- Static parts (Communications)
- Reconfigurable partitions (RP, purple areas in Figure 3)

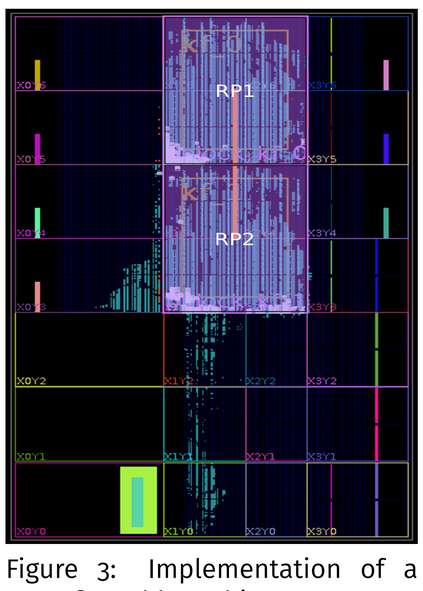

Figure 3: Implementation of a reconfigurable architecture onto a SoC FPGA (Zynq Ultrascale+)

3. Illustration with a Kalman tracking algorithm

This example considers two Kalman filters implemented on two reconfigurable regions on a SoC FPGA. The system includes seven Kalman models of different complexities which can be more adapted to some parts of the test trajectory. The system is designed with the help of a signal processing expert, and **the likelihood of the innovation is selected as the QoS criteria.**

$$f_{QoS}(k) = i_k^{\mathrm{T}} \cdot \Sigma_i^{-1} \cdot i_k \quad \text{where:} \quad \begin{array}{r} k \text{ , time-step} \\ f_{QoS} \text{ , QoS function} \\ i \text{ , innovation vector} \\ \Sigma_i \text{ , innovation covariance} \end{array}$$

As SoC designers we choose to compute this QoS in the HW module since the Kalman filtering already needs the computation of Σ_i^{-1} for the optimal gain.

$$K_k = P_{k|k-1} \cdot H_k^{\mathrm{T}} \cdot \Sigma_i^{-1} \quad \text{where:} \quad \begin{array}{r} K \text{ , optimal Kalman gain} \\ P_{k|k-1} \text{ , predicted error covariance} \\ H \text{ , observation model} \end{array}$$

The QoS value is observed during a window and the worst Kalman filter is replaced periodically. If both Kalman QoS is under a given threshold, both Kalman are replaced.

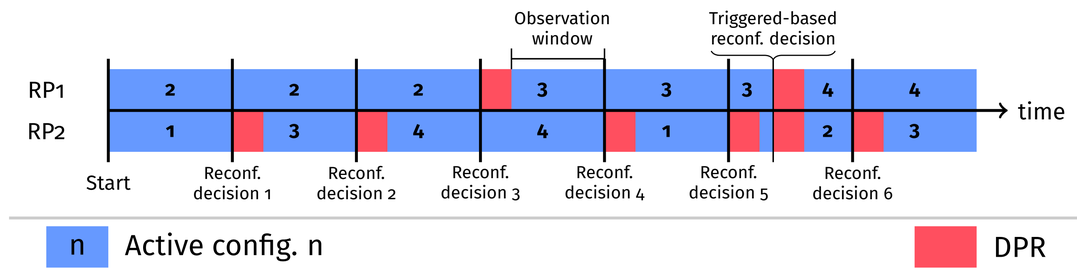

Figure 4: Example of QoS driven DPR control

4. Results on the tracking

The tracking highlights expected improvements. As shown in the left curve of the Figure 5, the system is able to choose the most adapted Kalman filter model.

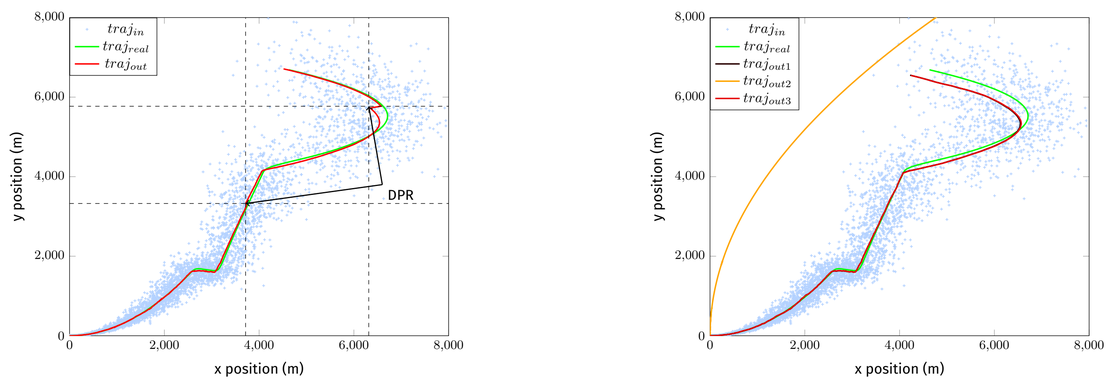

Figure 5: Tracking output of a the Kalman filter. On the left the QoS driven reconfiguration is used. On the right, the non-reconfigurable system was used

5. Conclusion and perspectives

The Kalman case study shows how QoS driven reconfiguration can improve the performances of a system. We will extend this concept to a full radar processing chain.

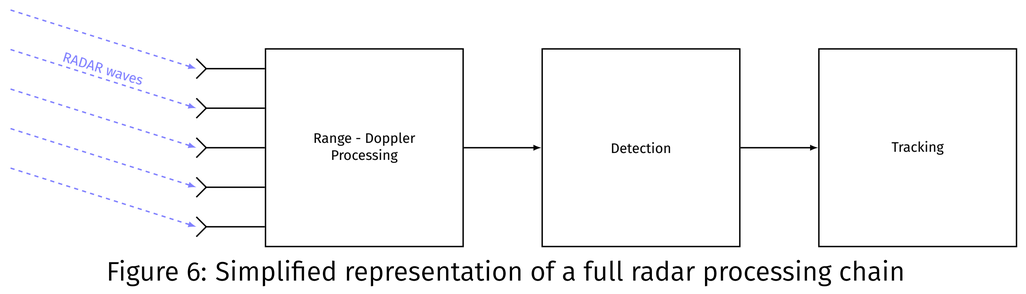

Figure 6: Simplified representation of a full radar processing chain

We also want to create a general method to catch expert knowledge and generate a QoS-aware reconfigurable architecture.