Introducton

- The reconfigurable hardware devices are commonly used in real-time systems; these devices are featured by a high density of heterogeneous resources, by the multitasking and supply a reasonable flexibility to applications need.
- Efficient management of hardware tasks and hardware resources is strongly required. But the scheduling and the placement methods suffer from the issues of fragmentation, tasks rejection, overheads, high power consumption.
- Requiring a novel method of placement which aims at the optimized use of the resources and targets all above mentioned issues.
- Placement sub-functions:
  - Partitioning: Handling free space.
  - Fitting: Selection of a feasible placement solution.

**Terminology**

- Device level: The device is a set of heterogeneous resources \( n_i \).
- Application Level: Tasks (Tj) are featured by:
  - Physical model: \( T_{j,PHY} = (a_{ij}, t_{j}) \).
  - Functional model: Period (Pj), WCET (Cj), Deadline, Priority.
- Conceptual level: There are four elements:
  - Static Region (SR) = \( \{ b_{ij}, t_{ij} \} \) (Static components).
  - Reconfigurable Region (RRj). The free space of heterogeneous resources dedicated for placing tasks.
  - \( R_{j,PHY} = (a_{ij}, t_{j}) \) (N.RZi) ⊂ RRj.
  - Reconfigurable Zone (RZj): Virtual block of heterogeneous resources specialized for a class of hardware tasks.
- Reconfigurable Bloc (RBj): The smallest abstraction unit matching the reconfiguration granularity (DIVj) in the device.

**Flow of Classification of Hardware Tasks**

**STEP 1: RZs Types search**

Step 1: RZs types search or hardware tasks classes search

- RZ reference = 0 // reference of RZ types
- RZ: list of all RZ types
- For all tasks Ti ∈ \( T \):
  - \( T_{i,PHY} = (a_{ij}, t_{j}) \)
  - \( T_{j,PHY} = (a_{ij}, t_{j}) \)
  - \( R_{j,PHY} = (a_{ij}, t_{j}) \) in \( N.RZ \)
  - \( R_{j,PHY} = (a_{ij}, t_{j}) \) in \( R.Z \)
  - \( R_{j,PHY} = (a_{ij}, t_{j}) \) in \( R.Z \)

**STEP 2: Hardware Tasks Classification**

- Computing of Occupation rate of each RZj:
  \[ \text{Occupation rate of } RZ_j = \frac{\sum a_{ij}}{P_i} \]
- Computing of costs D between tasks and their non-optimal RZj:
  \[ d_{ijk} = X_{i,k} - Z_{j,k} \]

**STEP 3: RBs/RZs Increasing**

Step 3 is performed when an overload within RZj is detected

- **Action 1:** Increasing the number of reloaded RZ types (RZj) till their overload is covered.
  \[ \text{Number overloaded } RZ_j = \frac{\sum a_{ij}}{P_i} \]

- **Action 2:** Adding RBs to a non-overloaded RZ, and accepting tasks having optimal overload RZ and giving Und distance with them. RZj increases the RBs as required by these tasks.

**Off-line Placement of RZs and on-line Mapping of Hardware Tasks**

- As *Generic Placement*, Placement/Mapping consists of:
  - Partitioning: searching RPBs for each RZ
  - Two-level fitting:
    - Placing RZs on selected RPBs.
    - Mapping Tasks on placed RZs with specified occupation rates (\( \Omega_i \)).
- Place/Mapping is a problem of optimization under constraints (K,D,F):
  - \( X = (X_1, X_2) \):
    - \( X_1 \): coordinates of RPBs.
    - \( X_2 \): coordinates of RZs.
  - \( D = (D_1, D_2) \): set of domains of values for \( X \).
  - C: set of constraints
    - Objective function evaluating:
      - Resources efficiency.
      - Optimal mapping.
      - Communication cost.

- **Proposal of an Exhaustive Placement/Mapping**
  - exhaustive_placement(input A, input X1,D1,C), output solution)
  - Begin
    - If all variables of X1 are assigned to a value in A then
      - A is a complete assignment for all variables X1 are placed?
      - Exhaustive_placement(RZ(D2,C2), solution)
        - Begin
        - End
      - Else
        - A is a partial assignment: some tasks are not all placed?
        - Choose a variable Z from X1 which is not assigned to a value in A
        - For all value \( V \) in D1 Do
          - \( A = A \cup (V, Z) \)
          - Exhaustive_placement(A, X1,D1,C), solution)
    - End
    - End

- **Conclusion**:
  - The proposed exhaustive Placement/Mapping increases exponentially with the number of RZs and with the number of tasks.
  - For example, to place 6 RZs and to map 8 tasks in Virtex V SX50 device the research space for this problem has a size of: \( (33386248) \times (8)^{26} \)
  - Placement/Mapping problem is NP-complete problem. It could be resolved by means of:
    - Complete methods as: Branch and Bound, dynamic programmation, etc.
    - Methaheuristics as: evolutionary methods, proximity methods and hybrid methods.