Design Optimization of Mixed-Criticality Real-Time Embedded Systems

In this article, we are interested in implementing mixed-criticality real-time embedded applications on a given heterogeneous distributed architecture. Applications have different criticality levels, captured by their Safety-Integrity Level (SIL), and are scheduled using static-cyclic scheduling. According to certification standards, mixed-criticality tasks can be integrated onto the same architecture only if there is enough spatial and temporal separation among them. We consider that the separation is provided by partitioning, such that applications run in separate partitions, and each partition is allocated several time slots on a processor. Tasks of different SILs can share a partition only if they are all elevated to the highest SIL among them. Such elevation leads to increased development costs, which increase dramatically with each SIL. Tasks of higher SILs can be decomposed into redundant structures of lower SIL tasks. We are interested to determine (i) the mapping of tasks to processors, (ii) the assignment of tasks to partitions, (iii) the decomposition of tasks into redundant lower SIL tasks, (iv) the sequence and size of the partition time slots on each processor, and (v) the schedule tables, such that all the applications are schedulable and the development costs are minimized. We have proposed a Tabu Search-based approach to solve this optimization problem. The proposed algorithm has been evaluated using several synthetic and real-life benchmarks.