

6

Semi-Extended Tasks

Stack as a Shared Resource

We must either learn to live together as brothers or we are all going to perish together as fools.

A Christmas Sermon, 1967, Martin Luther King Jr.

In the first part of this thesis, I used the static interaction analysis to investigate on the properties of an eventually deployed RTCS. In this and the next chapter, I will constructively improve system properties, namely stack consumption and reschedule overhead, in a post-mapping optimization step that is only possible with the proposed whole-system view on the interaction analysis.

With the in-depth knowledge about the thread activation and preemption patterns, I propose the fine-grained sharing of statically-allocated stack space among different threads. Traditionally, we consider the stack space as *the* private resource of an RTOS thread. With the proposed fine-grained stack sharing, I break up this monolithic view and expose the potentials to reduce the overall memory requirement of the RTCS. Thereby, the proposed *semi-extended task (SET)* approach leaves the application logic in place and does not limit the flexibility of the system; it only exploits inefficiencies in the resource usage that originate from the segregation of the real-time application into multiple threads.