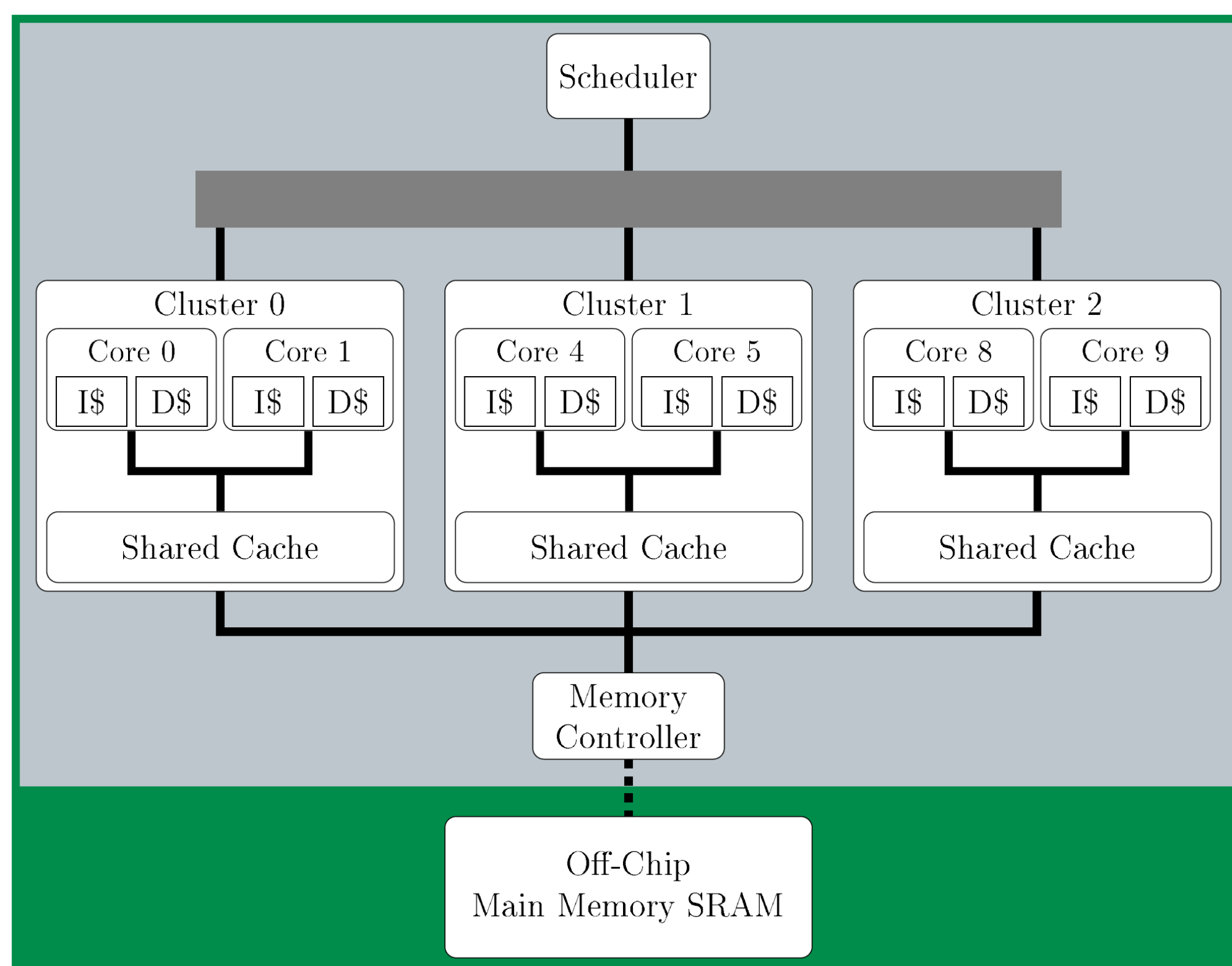


Real-Time & Mixed-Criticality Systems

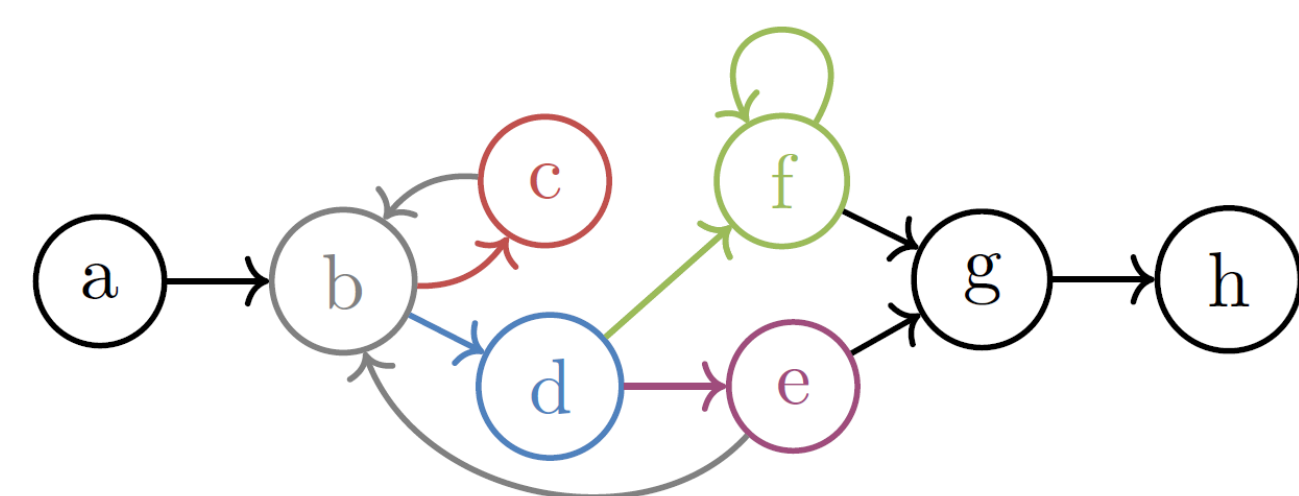
Emad Jacob Maroun, PostDoc, ejama@dtu.dk

Real-Time & Mixed-Criticality Systems: Many systems require guarantees on the execution time of programs. Examples include vehicle control system (avionics, self-driving cars), health-care devices, or various heavy-industry control application. Missing program execution deadline may cause great loss of value or serious body injury. Therefore, special systems must be designed with custom software to ensure guarantees on execution times. **Worst-case execution time (WCET)** analysis is used to give upper bounds on program/function execution times. These projects revolve around writing software and tools to enable the development of safety-critical systems that are easily analyzable.

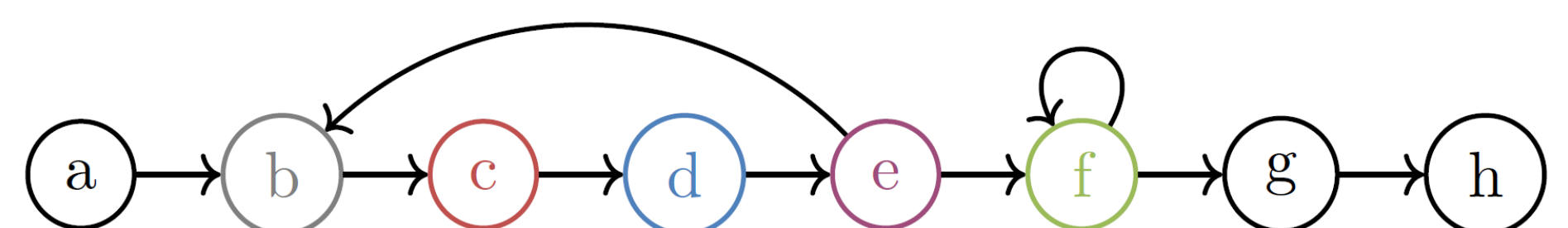


Cache & Analysis Projects: WCET analysis tracks each function's possible runtime execution with the possible state of the hardware to find an upper bound on the function's execution time. The state of the caches at any point in the program greatly affects the WCET result. Therefore, proper cache implementation and analysis is key to low WCET bounds. Project include **implementing data caches or cache analysis for our custom WCET analyzer 'Platin'**, implementing different cache architectures in hardware or simulation, or adding support for schedulability analysis in Platin.

Compiler & Algorithm Projects: Custom compilation techniques can be used to generate code that is more amenable to WCET analysis. Additionally, special algorithm can be used that have lower WCETs instead of low average-case execution times. Projects include implement **if-conversion for traditional code or investigating graph algorithms with better WCET performance**



Traditional



Single-Path

Single-Path Code Generation: A technique for generating code with constant execution time. It converts each function into a control flow graph with only one path of execution. WCET analysis is unnecessary: running the function on arbitrary inputs will give the WCET. Projects on this topic include **optimizing the code generation, automatic loop-bound propagation, supporting more flow fact types, or implementing a constant execution time memory arbiter.**