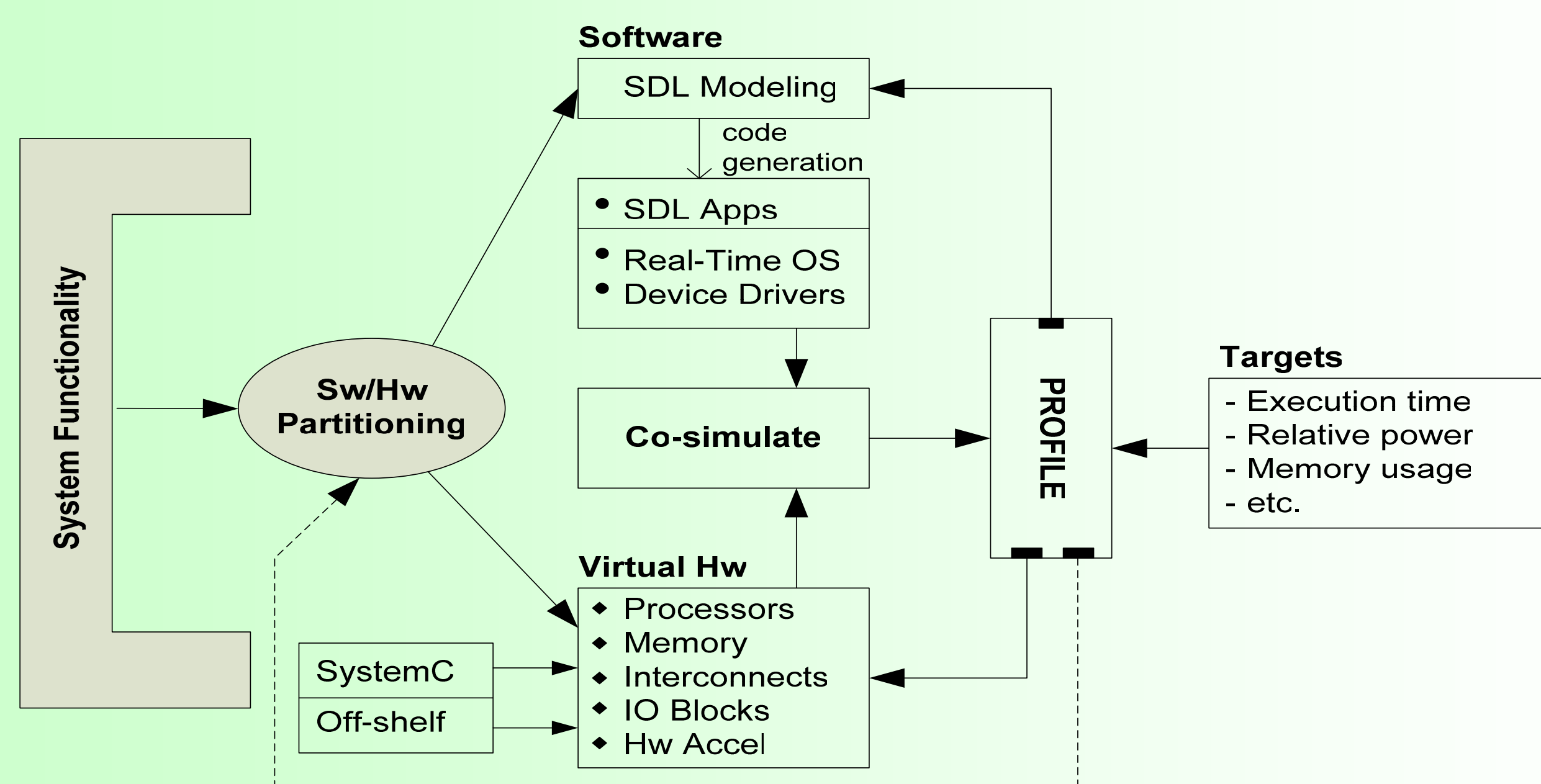


Introduction

Due to the rising complexity of embedded systems, design decisions of both hardware and software must be taken at early stage of system development. We propose a methodology that allows rapid investigation of several design options for hardware and software within a single design flow by using abstract and high-level modeling techniques. We apply this methodology to customize the processor and memory subsystem of a mobile phone platform, which runs a modem application on top of an L4-based Real-Time Operating System.

Methodology



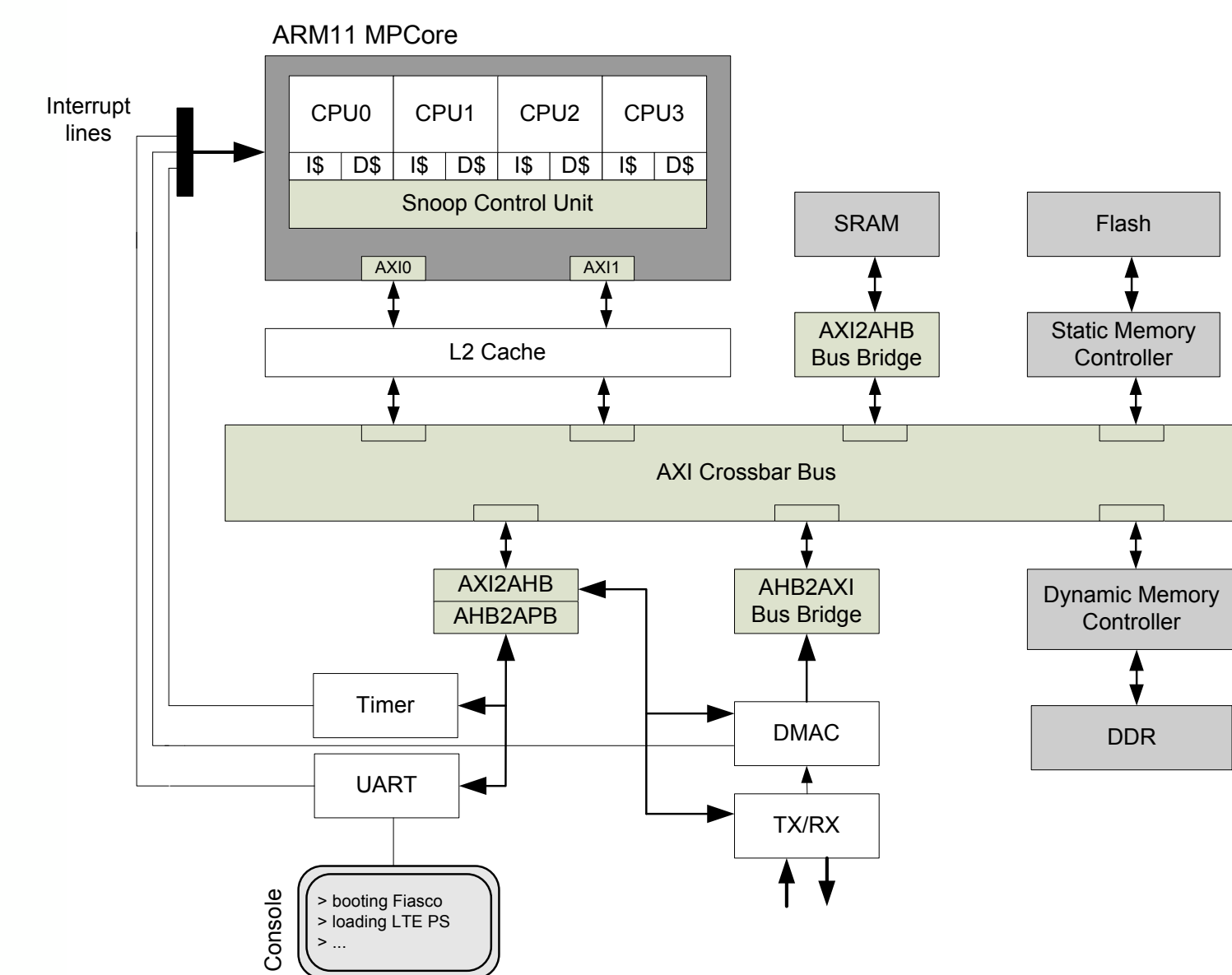
- Allows for early identification of System bottlenecks with respect to predefined constraints, such as execution time and power
- High-level software modeling of Hardware architecture using SystemC/C++
- Abstract modeling and easy modification of software architecture using Specification and Description Language (SDL)
- Provides means for gradual software and hardware refinement and optimization
- Bridges the Gap between non-compatible software and hardware design flows
- Profiling, analysis, and evaluation of several hardware and software design combinations aids the process of hardware/software partitioning and software verification



Architectural Exploration

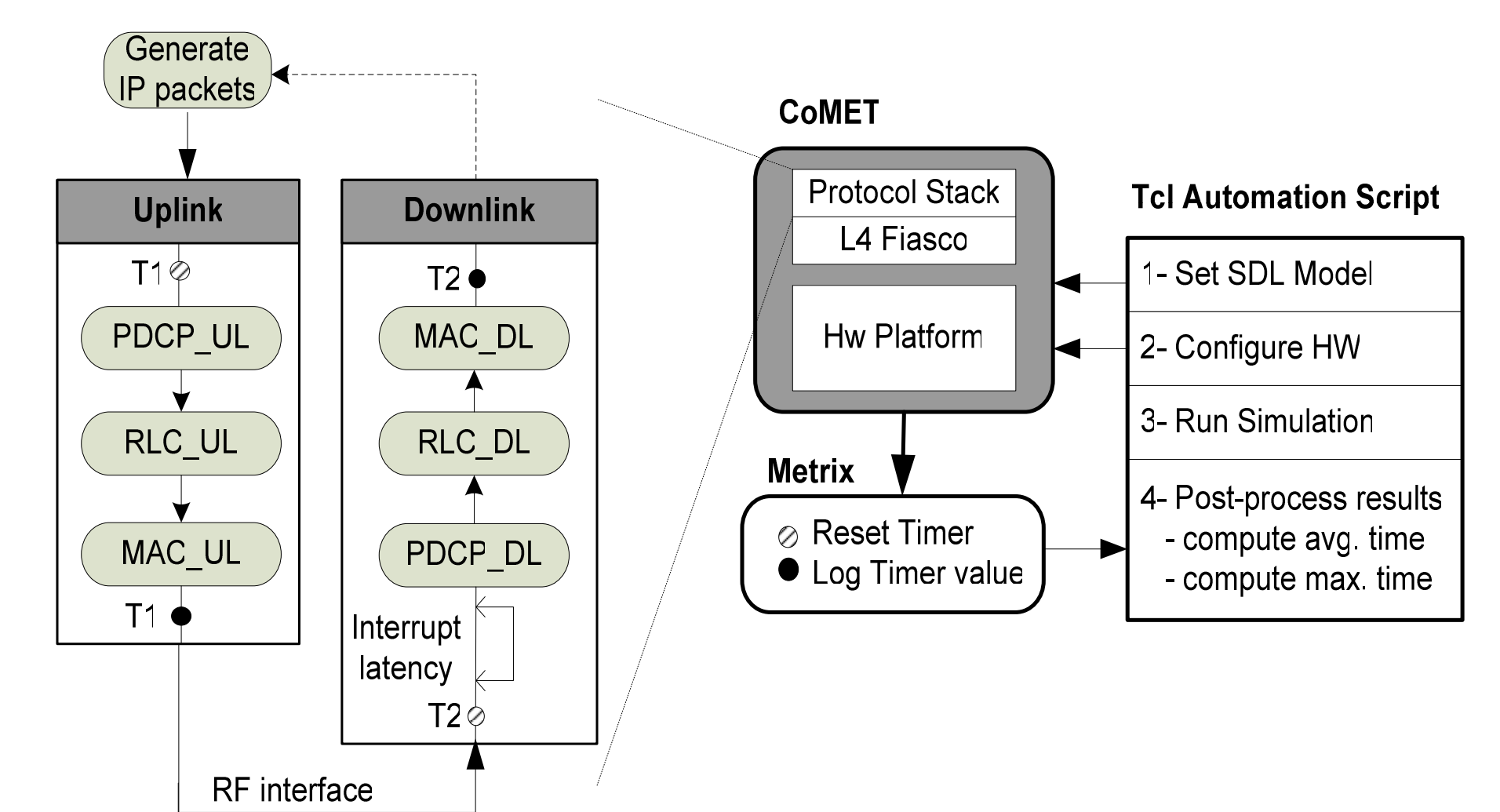
- Simulations based on **single core** execution
- First evaluation step applied on 12 design points representing different combination of hardware parameters, such as cache size, memory latency, and core frequency.
- By increasing the frequency to 350 MHz, i.e. the second bar in Fig. 1 and 2, the execution times is shortened by 10% and 35 % at cache sizes 8KB and 64KB, respectively.
- Shrinking the memory latency to one fourth, i.e. the third bar in Fig. 1 and 2, leads to 40% processing time reduction at 8KB and only 10% reduction at 16KB since the latter has much lower number of cache misses.
- Setting the timing requirements for MAC and RLC processing to 80 us and taking into consideration maximum processing time in Fig. 3, we infer that design points based on 32KB cache size are most suitable for further evaluation.
- Variation of the number of running threads in Fig. 4 highlights the thread management overhead. Note that no parallelism takes place since the software is executed on a single core.
- Finally, the cache size of 32KB, core frequency of 350 MHz, and full memory latency, i.e. design point d7, with 1 execution thread is recognized as the best design configuration the satisfies the timing requirements and balances area and power consumption.

Virtual Hardware Platform



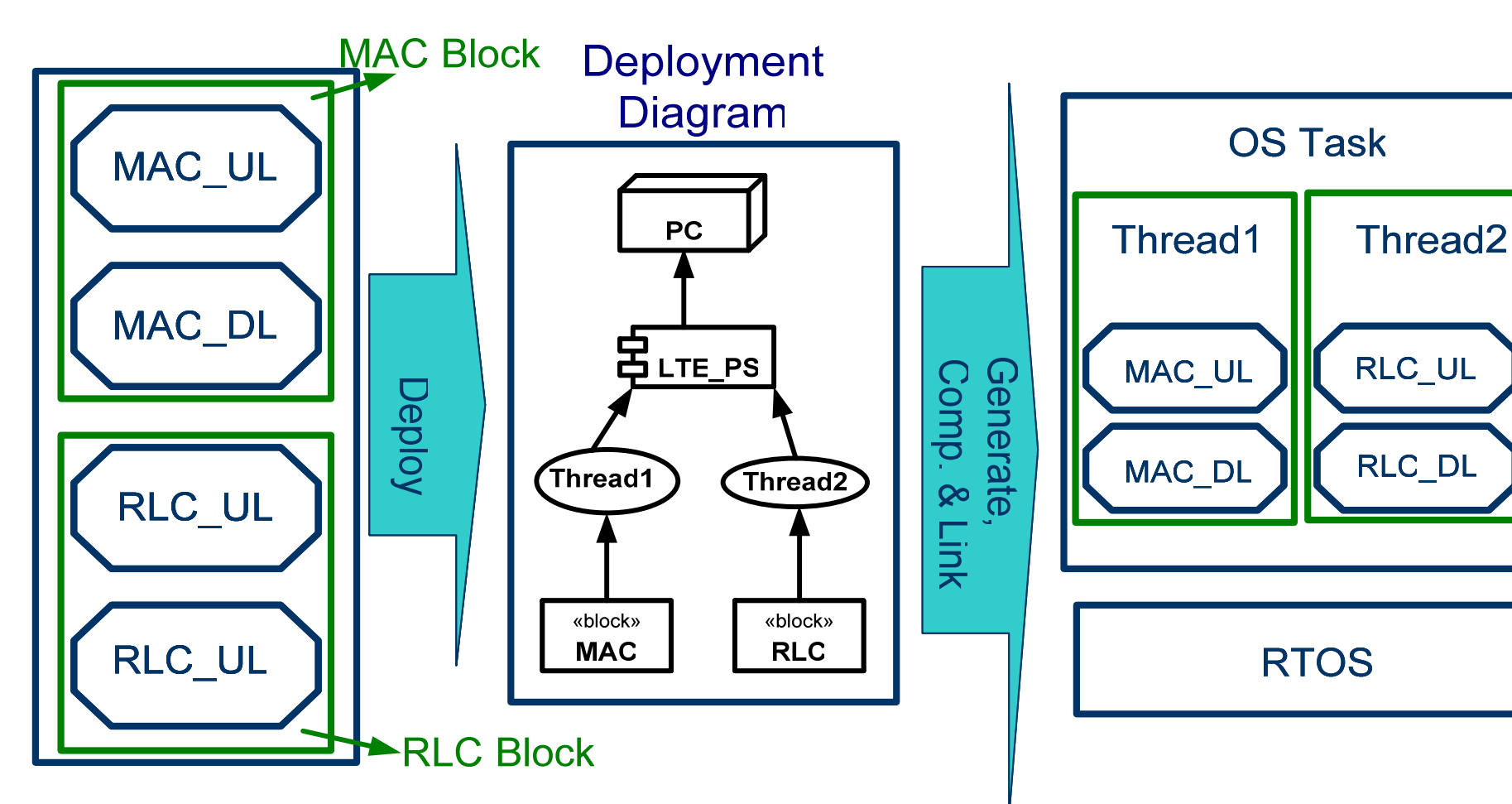
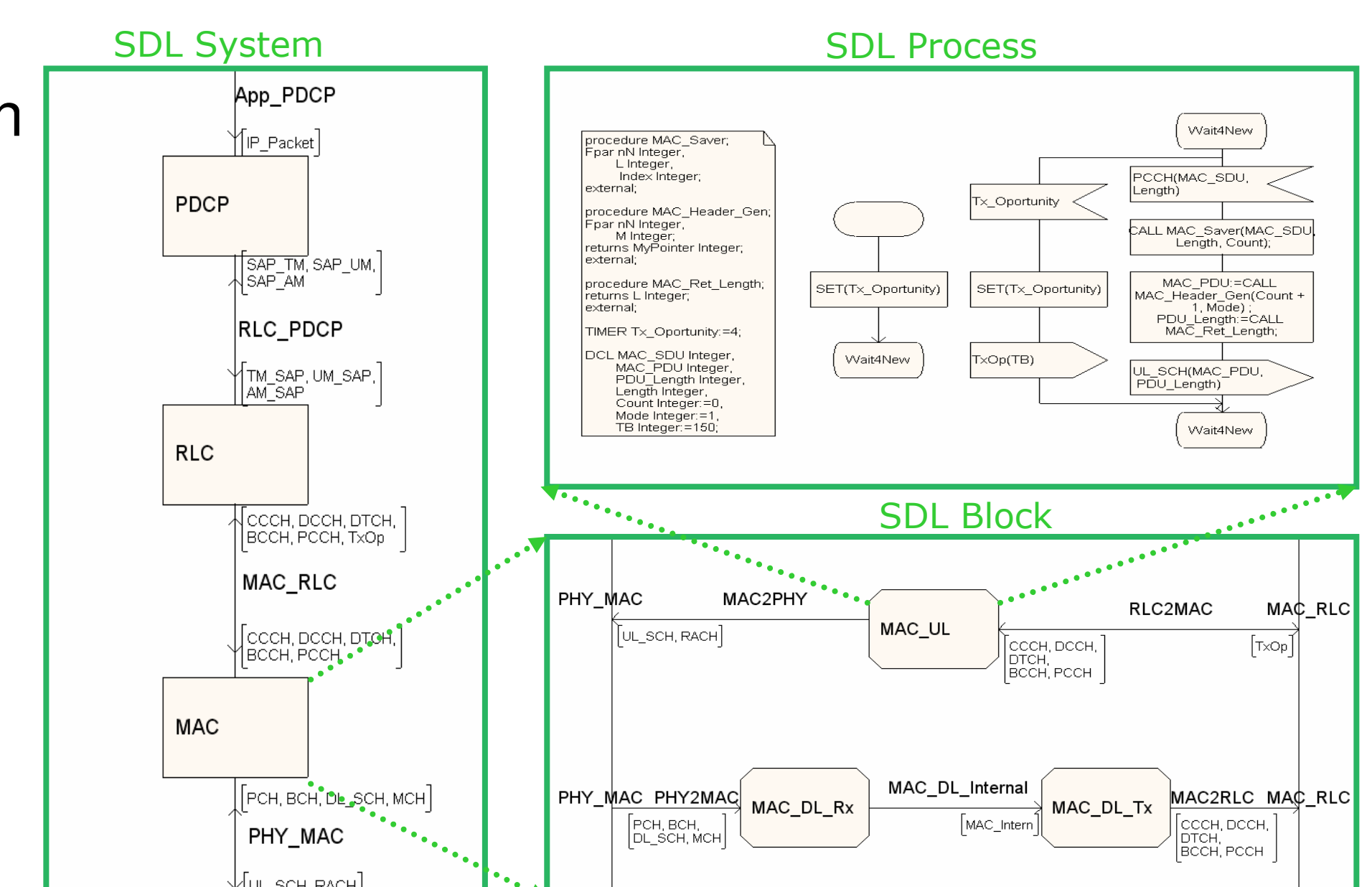
- ARM11 MPCore based and representative mobile phone platform
- Configurable System-on-Chip hardware architecture described at different levels of granularity
- Hardware generation of valid radio transport blocks to emulate the connection with transmitting base station

- Automation environment for evaluation of different hardware and software design parameters
- Simulation of hardware and software components using CoMET
- Profiling timers integrated in software allowing measurements with Metrix



SDL Model of LTE

- ITU-T Specification and Description Language (SDL) standard
- Structured approach using the hierarchical decomposition with system, block, process, and procedure
- Every sub-layer modeled as a separate SDL block
- LTE PS behavior modeled by Extended Finite-State Machines (EFSM)



- The deployment diagram maps SDL model to separate threads
- Every SDL block is treated as one thread
- SDL suite convert the graphical model to threaded application
- Pthread library used in order to manage the SDL threaded application

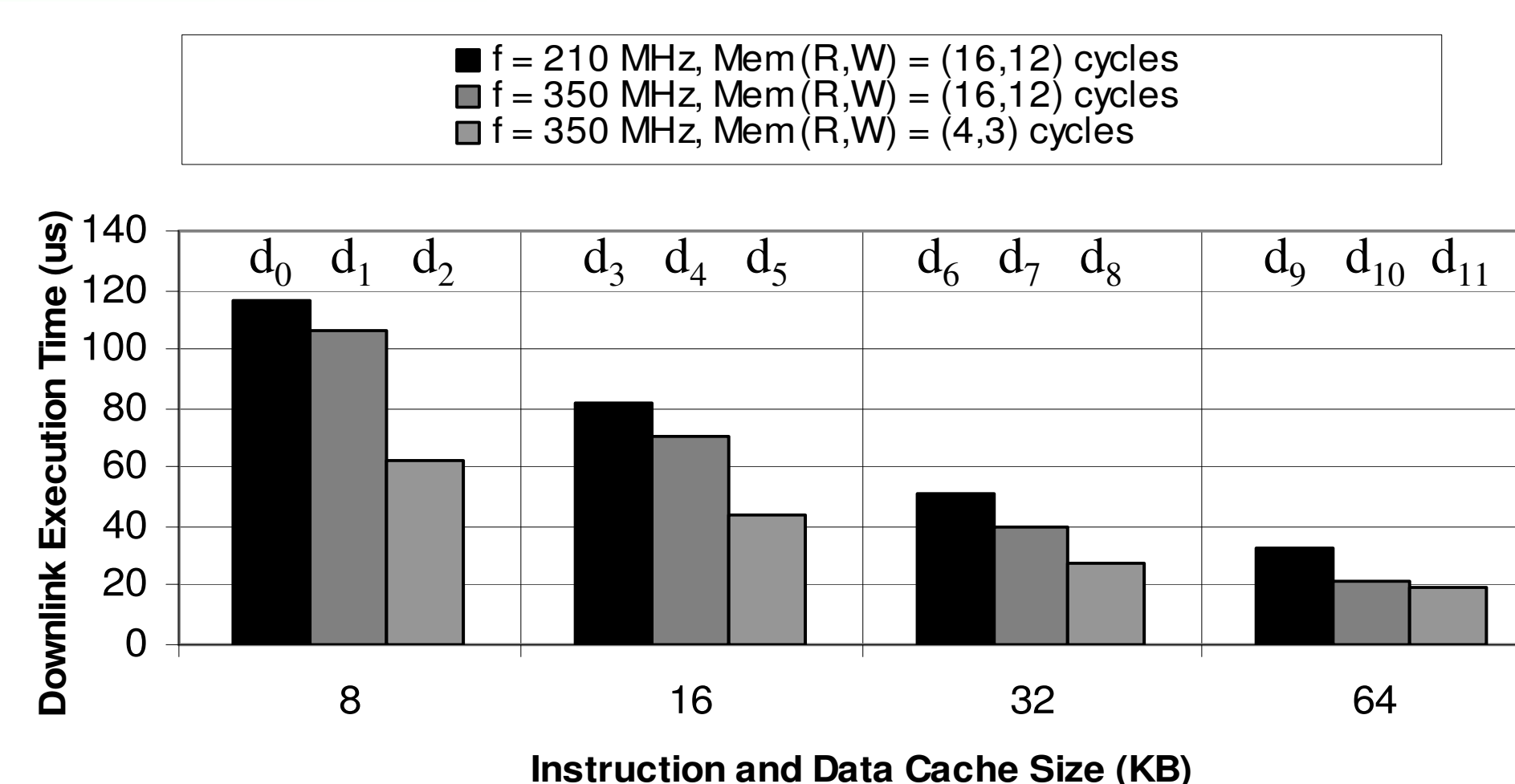


Fig. 1 Average Execution time for Downlink MAC/RLC packet processing

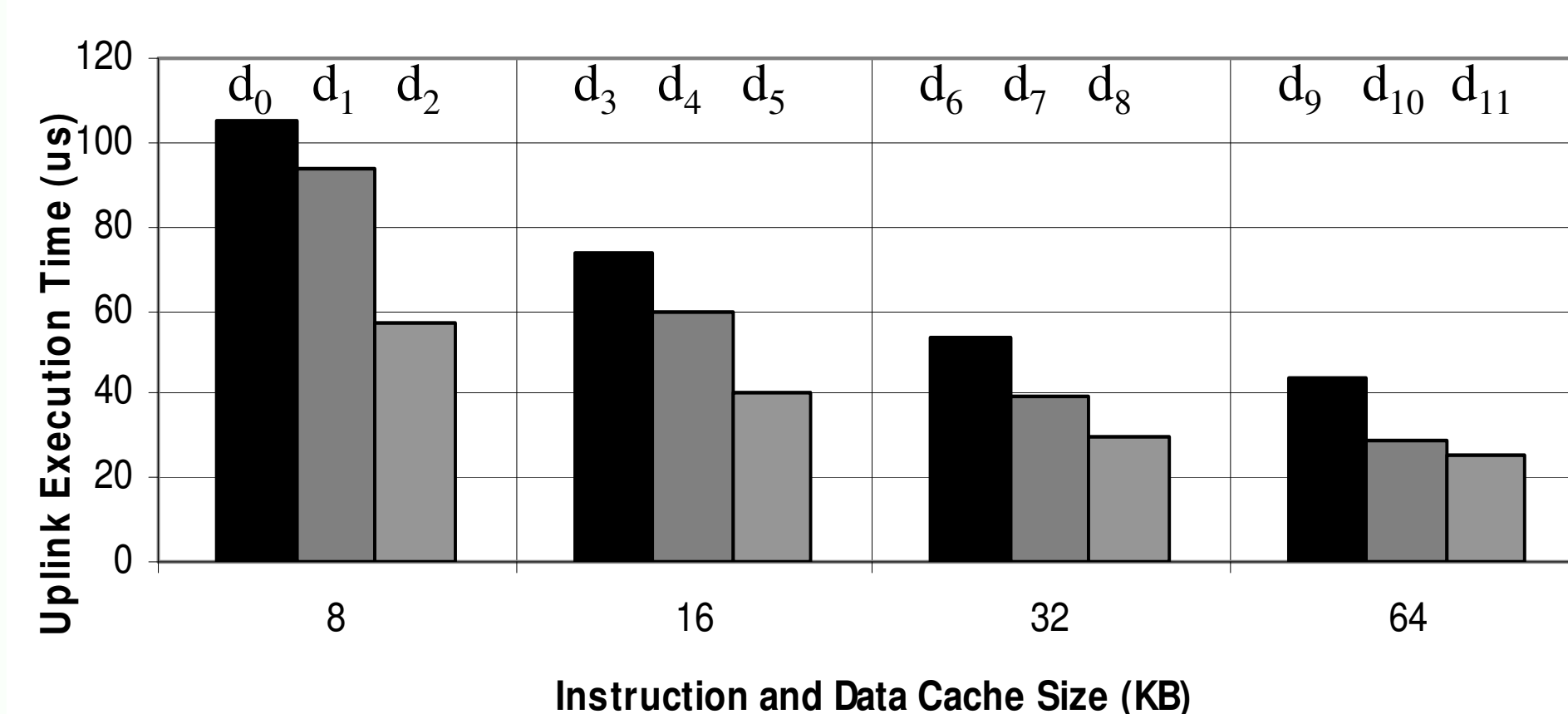


Fig. 2 Average Execution time for Uplink MAC/RLC packet processing

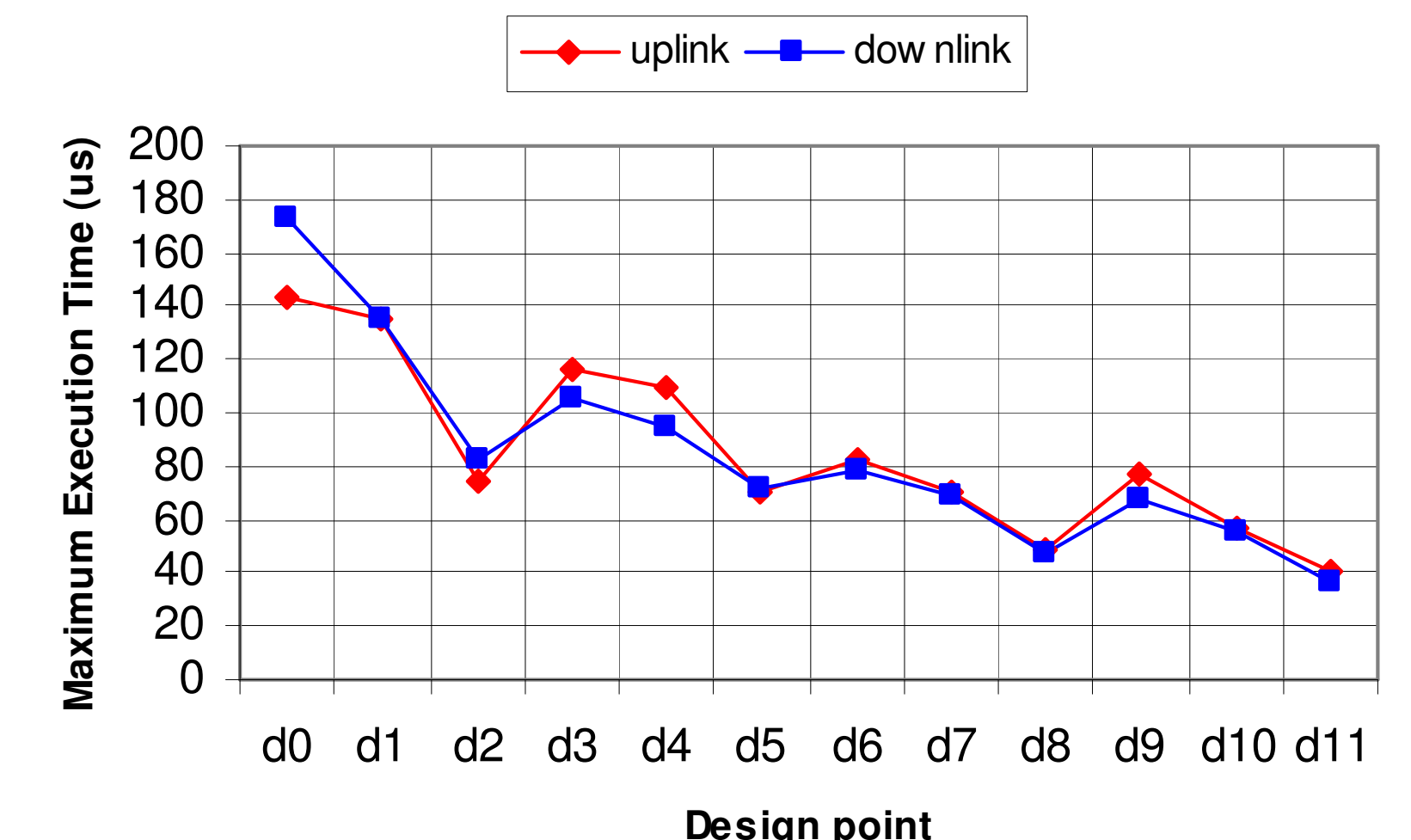


Fig. 3 Maximum execution time measured at different design configurations

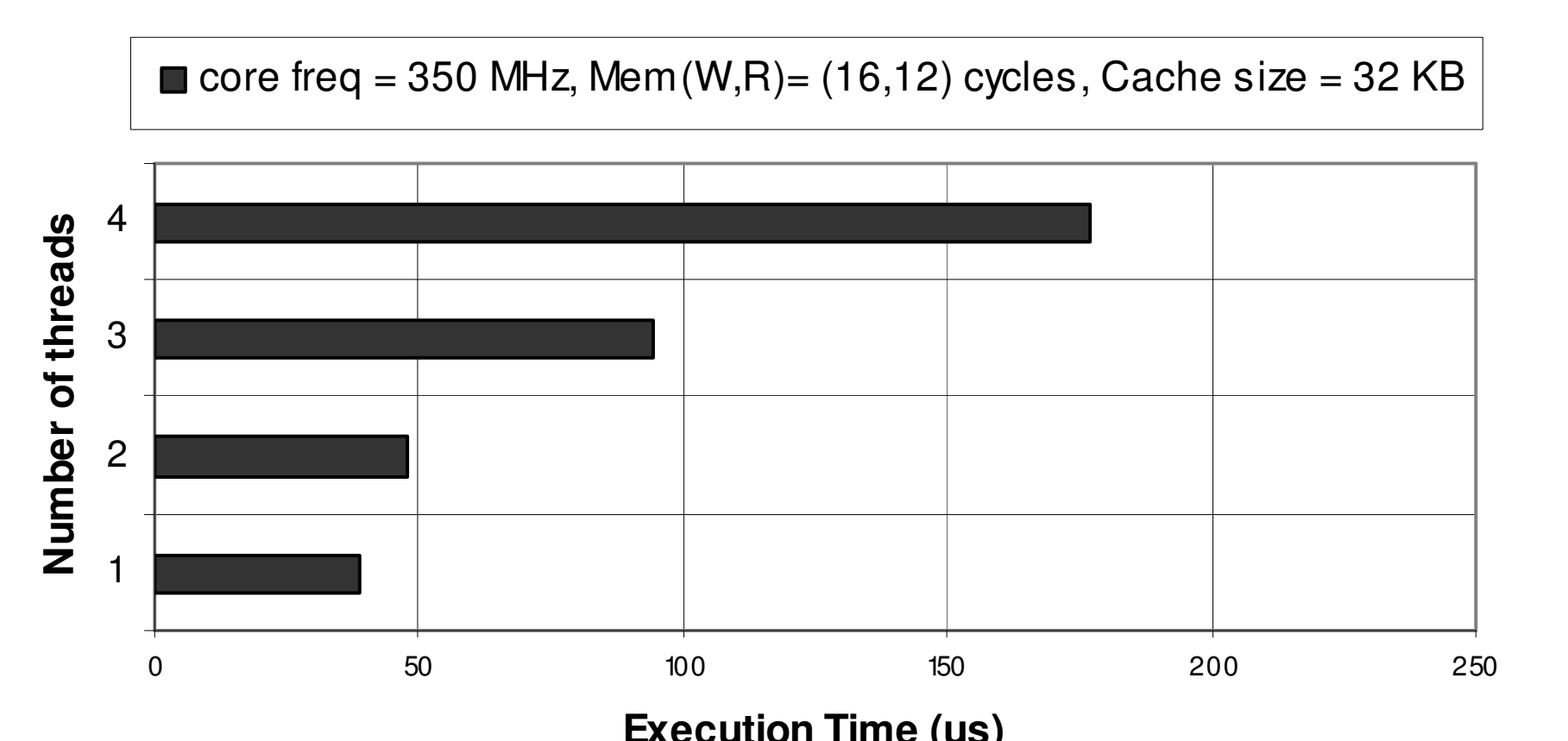


Fig. 4 Impact of thread management overhead on the execution time