

Mapping Periodic Realtime Tasks to Reconfigurable Hardware

Klaus Danne and Marco Platzner

Computer Engineering Group & Paderborn Center for Parallel Computing
University of Paderborn

`marco.platzner@computer.org`

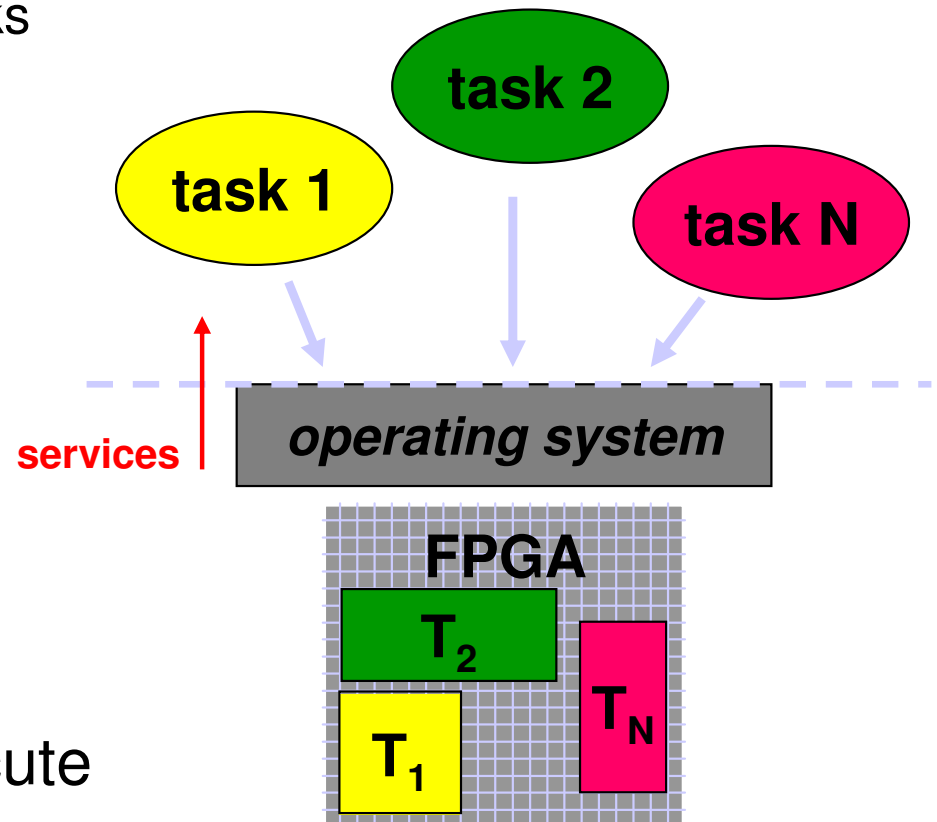


Overview

- motivation, models and metrics
- three scheduling approaches
 - ◆ global EDF
 - ◆ partitioned EDF
 - ◆ MSDL
- comparison and realization
- outlook

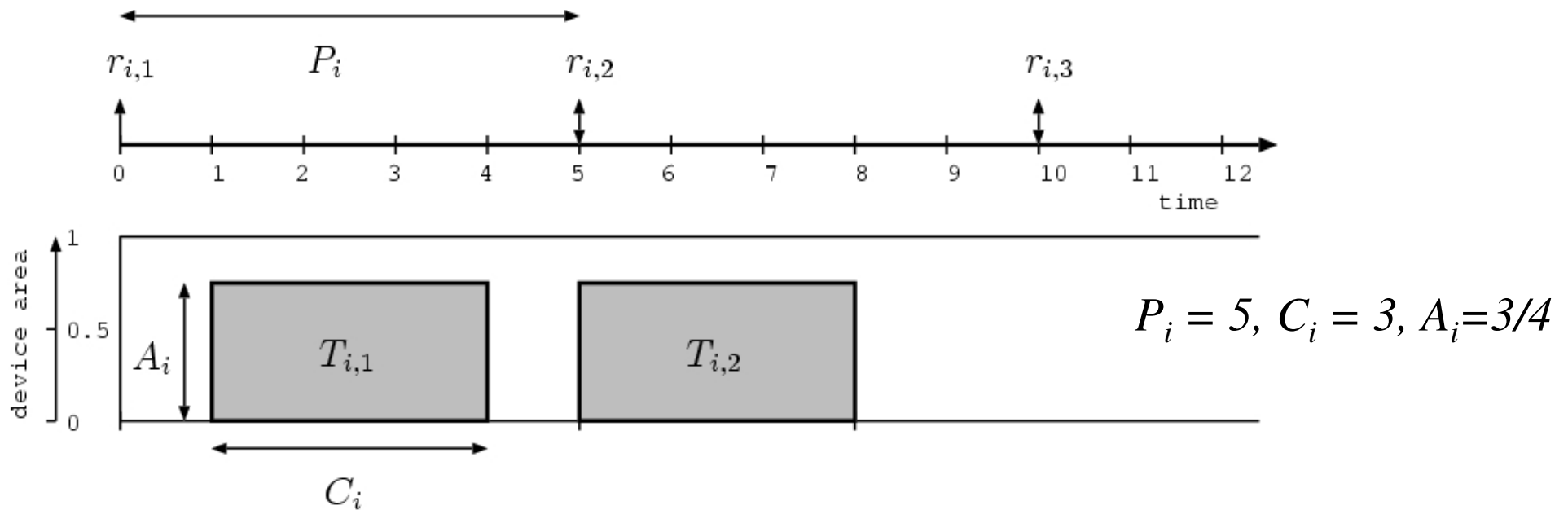
Motivation

- multitasking in hardware
 - ◆ circuits turned into hardware tasks
 - ◆ runtime environment schedules, places and executes these tasks



- investigate techniques to execute periodic realtime tasks
 - ◆ models different from single/multi-processor
 - ◆ important class of applications

Models and Metrics



- set of periodic real-time tasks $\Gamma = \{T_1, \dots, T_n\}$
 - ◆ relative deadline equals period P_i
 - ◆ worst-case computation time C_i
 - ◆ area A_i

- system model
 - ◆ preemptive multitasking
 - ◆ task set R can be executed in parallel, iff $\sum_{T_i \in R} A_i \leq 1$

Utilization Metrics

- time utilization $U^T(\Gamma) = \sum_{T_i \in \Gamma} \frac{C_i}{P_i}$
 - ◆ captures the fraction of time a task set Γ occupies the device when the tasks are executed sequentially

- system utilization $U^S(\Gamma) = \sum_{T_i \in \Gamma} \frac{C_i}{P_i} A_i$
 - ◆ captures the average system load generated by task set Γ

T_i	P_i	C_i	A_i	$U^T(T_i)$	$U^S(T_i)$
T_1	4	2	1/2	1/2	1/4
T_2	6	5	1/4	5/6	5/24
T_3	12	3	3/4	1/4	3/16
T_4	12	2	1/4	1/6	1/24
				1.75	0.69

Overview

- ✓ motivation, models and metrics

- three scheduling approaches
 - ◆ global EDF
 - ◆ partitioned EDF
 - ◆ MSDL

- comparison and realization

- outlook

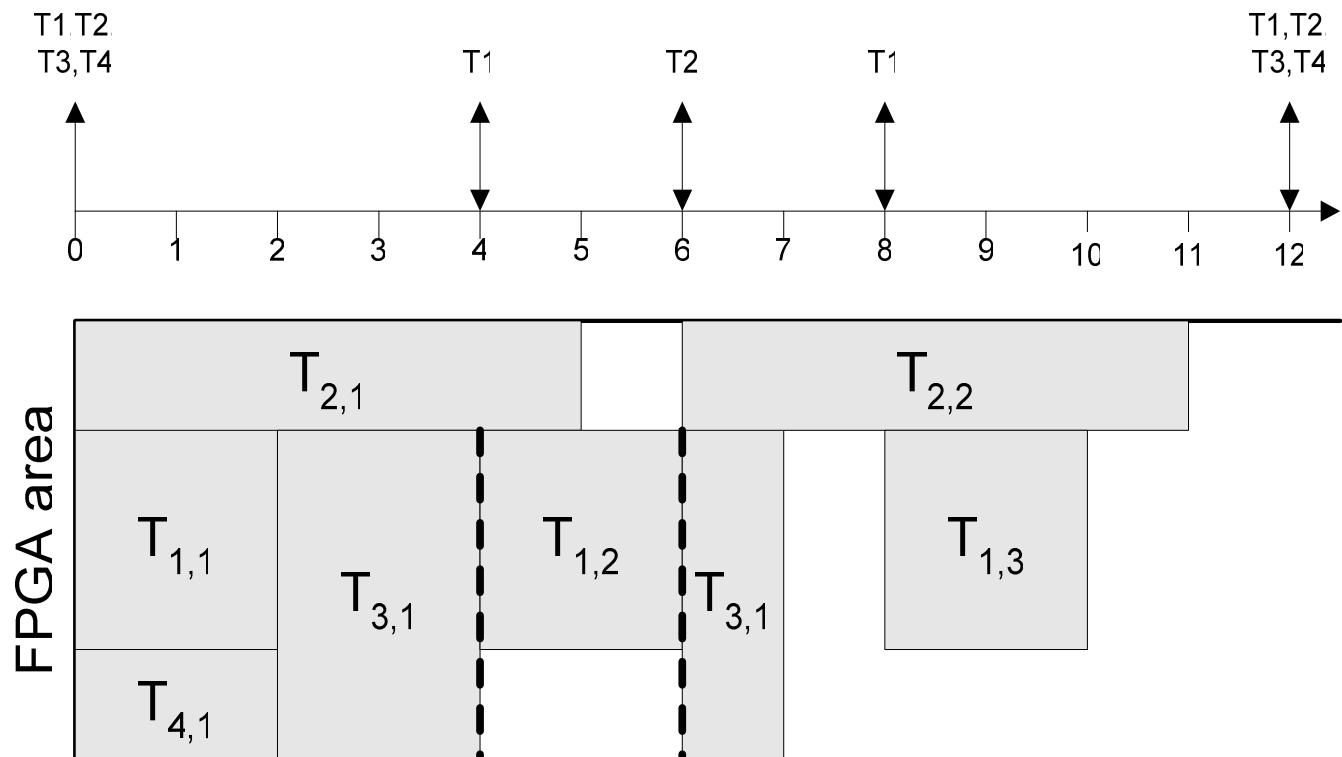
Global EDF Scheduling

EDF – Next Fit (EDF-NF)

- ◆ ready tasks are queued according to non-decreasing deadlines
- ◆ scan through the ready queue on every task release and termination
 - if a task fits onto the device, add it to the set of running tasks and execute it
 - otherwise, postpone the task and proceed with the next task in the queue

example

T_i	P_i	C_i	A_i
T_1	4	2	1/2
T_2	6	5	1/4
T_3	12	3	3/4
T_4	12	2	1/4



Global EDF Schedulability Test

$$\forall T_k \in \Gamma :$$

$$U^S(\Gamma) \leq (A(H) - A_{max}) \cdot (1 - U^T(T_k)) + U^S(T_k)$$

A_{max} maximum task area

$A(H)$... area of the reconfigurable device

- ◆ can be evaluated in linear time
- ◆ test is sufficient, but not necessary
- ◆ proof uses a resource augmentation approach

[K.Danne and M.Platzner, An EDF Schedulability Test for Periodic Tasks on Reconfigurable Hardware Devices. *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, 2006]

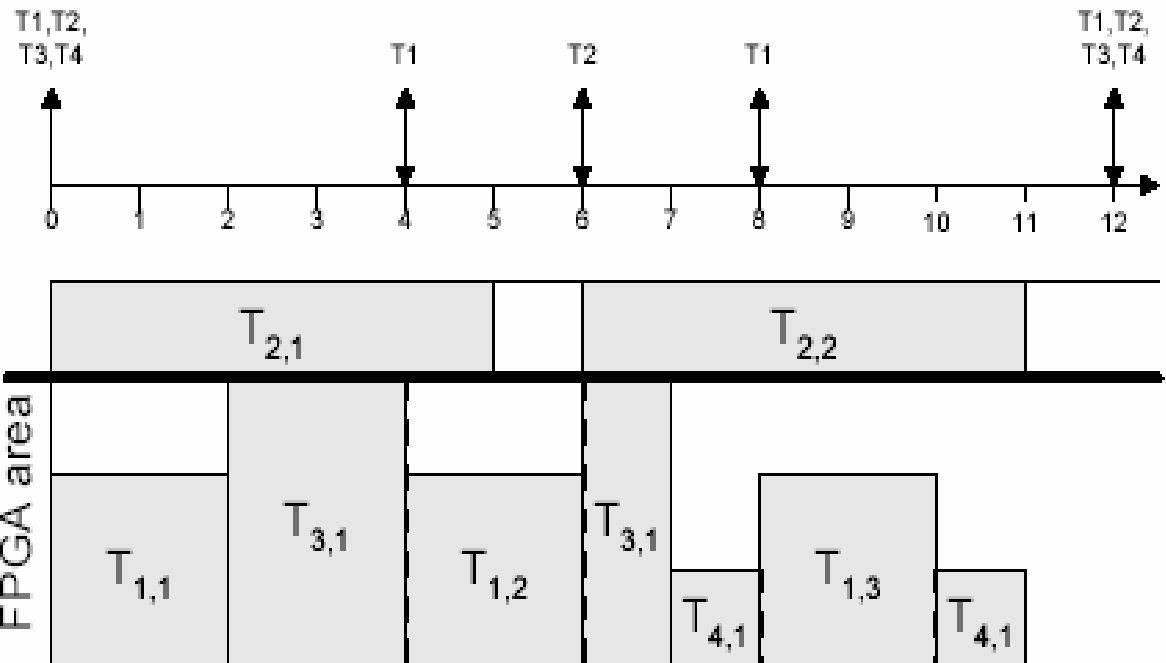
Partitioned EDF Scheduling

partitioned EDF

- the task set Γ is partitioned into subsets G_1, \dots, G_m
- each subset G_i is scheduled separately by sequential EDF

T_i	P_i	C_i	A_i	$U^T(T_i)$	$U^S(T_i)$
T_1	4	2	1/2	1/2	1/4
T_2	6	5	1/4	5/6	5/24
T_3	12	3	3/4	1/4	3/16
T_4	12	2	1/4	1/6	1/24
				1.75	0.69

example

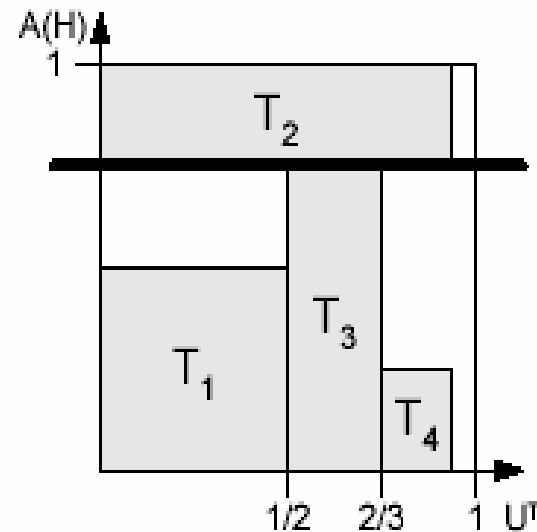


$$G_1 = \{T_1, T_3, T_4\}$$

$$G_2 = \{T_2\}$$

Partitioning Problem

- given a task set Γ , find a partitioning G_1, \dots, G_m such that
 - ◆ each partition can be scheduled by sequential EDF, ie. $U^T(G_i) \leq 1$
 - ◆ the slots required for G_1, \dots, G_m fit together onto the device



- approaches
 - ◆ integer linear programming
 - ◆ heuristics, eg. Next-Fit Decreasing Area

[K.Danne and M.Platzner. Partitioned Scheduling of Periodic Real-Time Tasks onto Reconfigurable Hardware, *Reconfigurable Architectures Workshop (RAW)*, 2006]

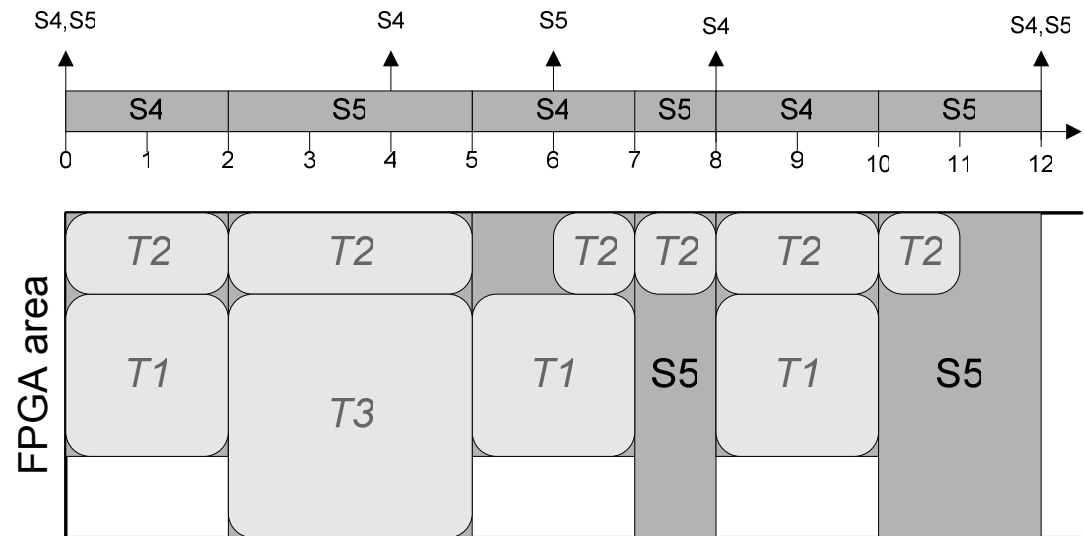
MSDL (merge servers & distribute load) Algorithm

■ example

S_i	R_i	P_i	C_i	A_i	U_i^T	U_i^S
S_1	T_1	4	2	1/2	1/2	1/4
S_2	T_2	6	5	1/4	5/6	5/24
S_3	T_3	12	3	3/4	1/4	3/16
					1.58	0.65

S_1	T_1	4	0	1/2	0	0
S_2	T_2	6	3	1/4	1/2	1/8
S_3	T_3	12	3	3/4	1/4	3/16
S_4	T_1, T_2	4	2	3/4	1/2	3/8
					1.25	0.69

S_2	T_2	6	0	1/4	0	0
S_3	T_3	12	0	3/4	0	0
S_4	T_1, T_2	4	2	3/4	1/2	3/8
S_5	T_2, T_3	6	3	1	1/2	1/2
					1	0.88



trading area for time

[K.Danne and M.Platzner. A Heuristic Approach to Schedule Periodic Real-time Tasks on Reconfigurable Hardware. *International Conference on Field Programmable Logic and Applications (FPL)*, 2005]

Overview

- ✓ motivation, models and metrics

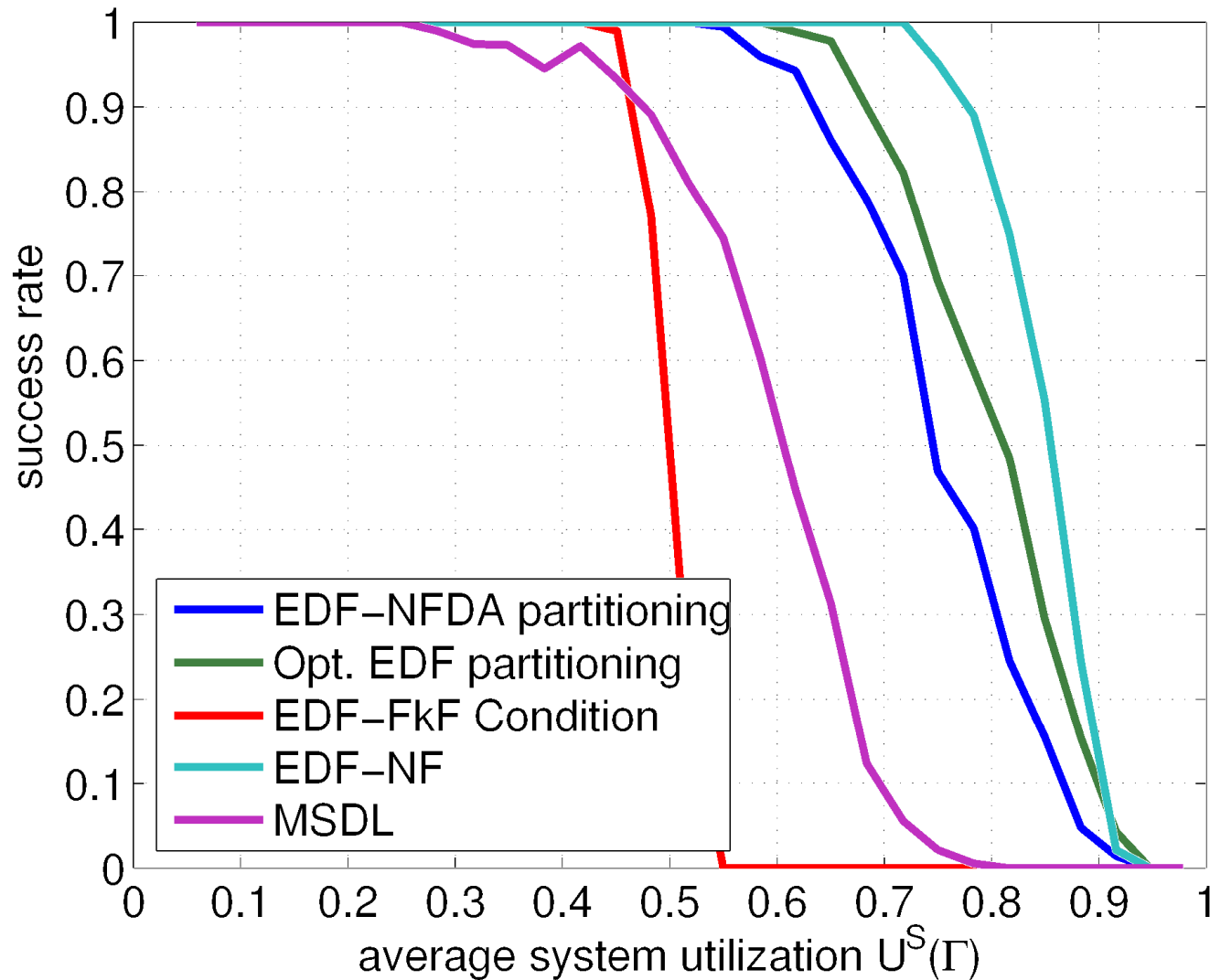
- ✓ three scheduling approaches
 - ◆ global EDF
 - ◆ partitioned EDF
 - ◆ MSDL

- comparison and realization

- outlook

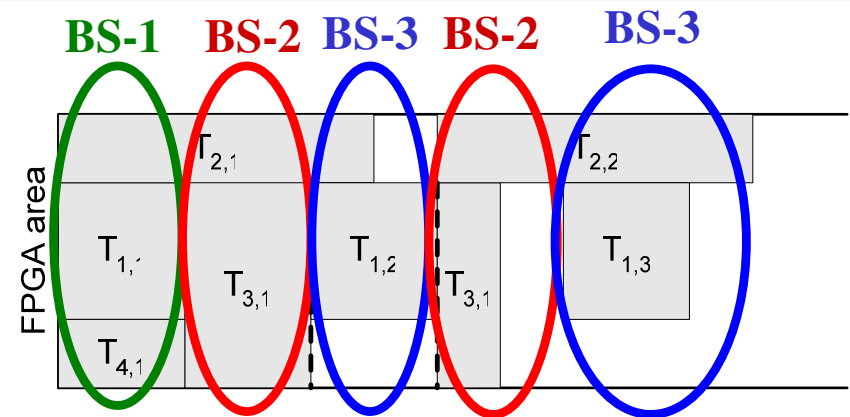
Scheduling Performance – Simulation

- randomly generated task sets
 - ◆ parameters: $C_i = \{1,2,\dots,30\}$, $U^T(T_i) = [0.2,0.4]$, $A_i = [0.2,0.4]$

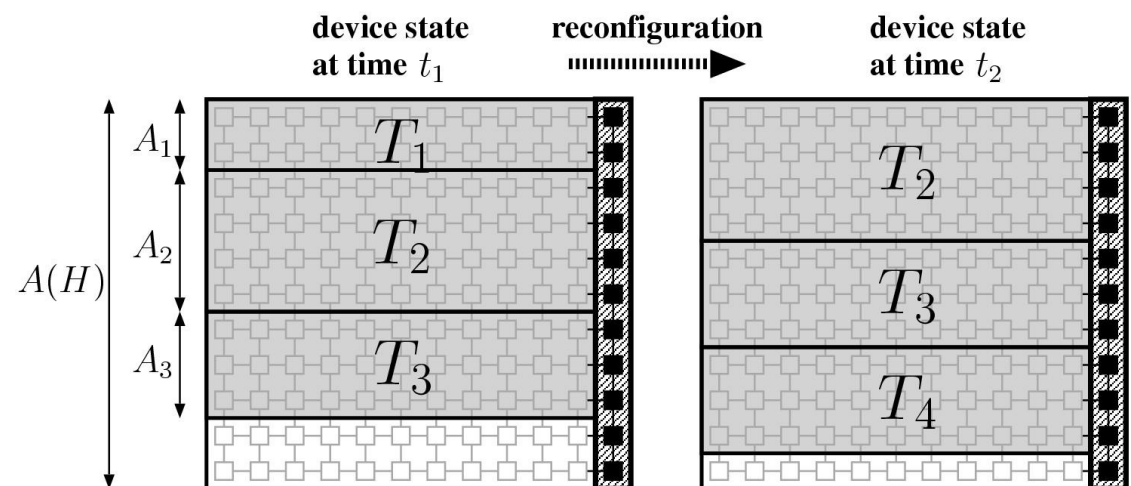


Realization – Global EDF

- full reconfiguration
 - ◆ compile bitstreams offline for all sets of running tasks
 - ◆ number of required bitstreams grows too rapidly

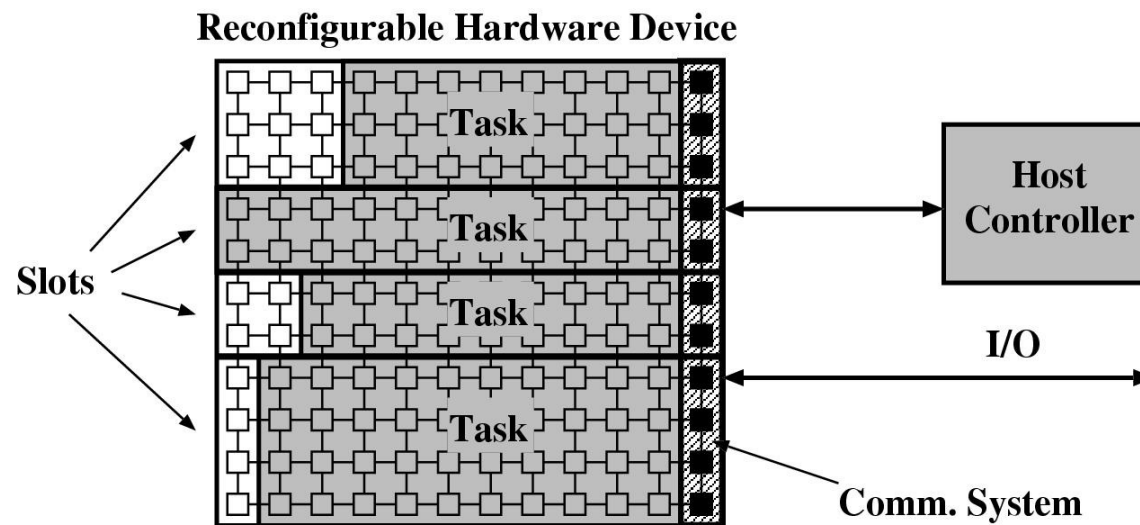


- partial reconfiguration: 1D variable area model
 - ◆ de-fragment area on every task start → task relocatability required
 - ◆ partial bitstream(s) for each task
 - ◆ suited to online scheduling



Realization – Partitioned EDF

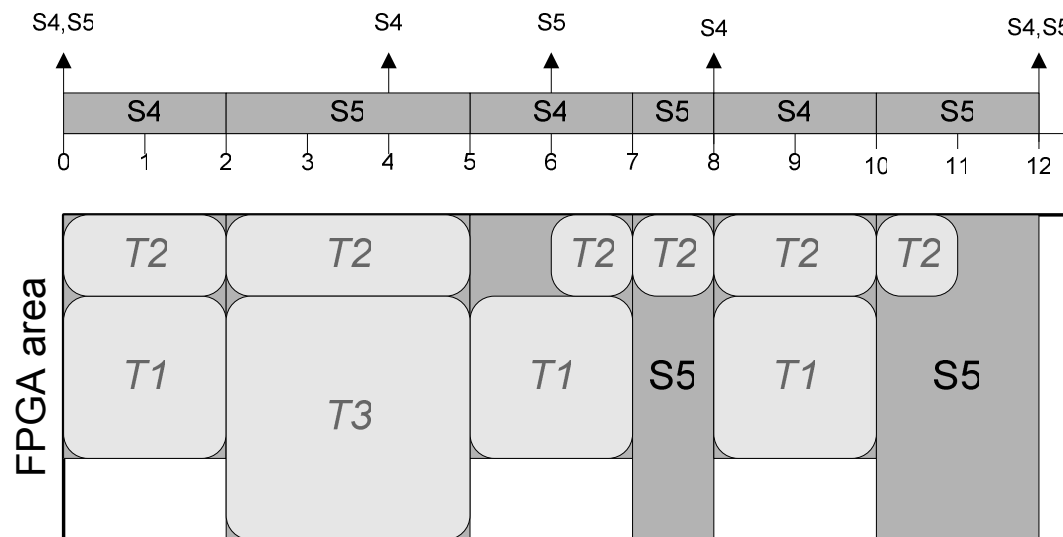
- matches perfectly the 1D slotted area model
 - ◆ in each slot, a subset of the tasks is executed sequentially
 - ◆ the slot sizes are determined offline
 - ◆ partial reconfiguration, but no task relocation required
 - ◆ well-suited for Xilinx-Virtex based prototyping platforms



- ◆ online scheduling?

Realization – MSDL

- MSDL generates a small number of servers
 - ◆ bounded by the number of tasks
 - ◆ servers (bitstreams) are compiled offline
- makes sense only for full reconfiguration
- not suited to online scheduling



Overhead Analysis

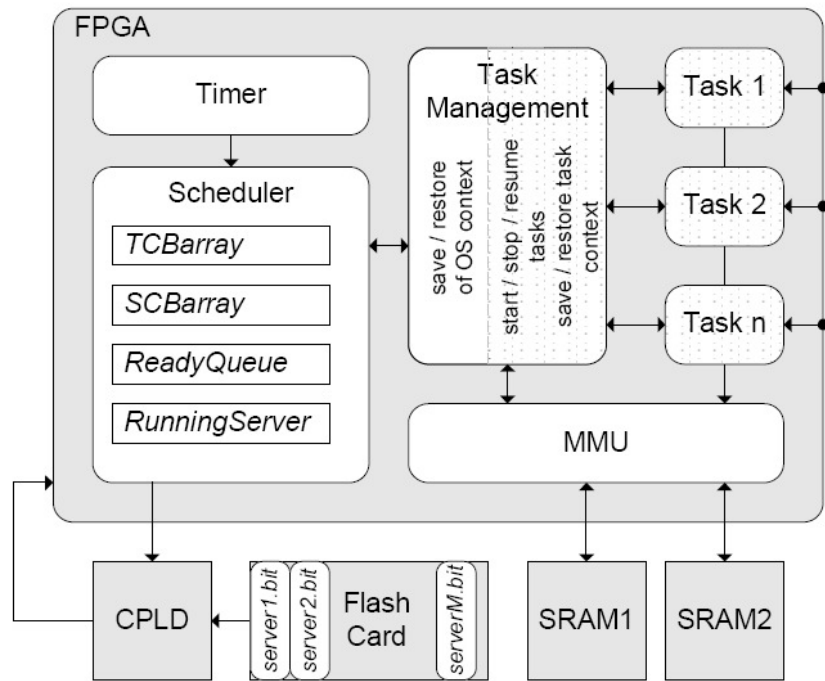
■ overheads due to

- ◆ FPGA reconfiguration time, task state save/restore time
- ◆ runtime system area overhead

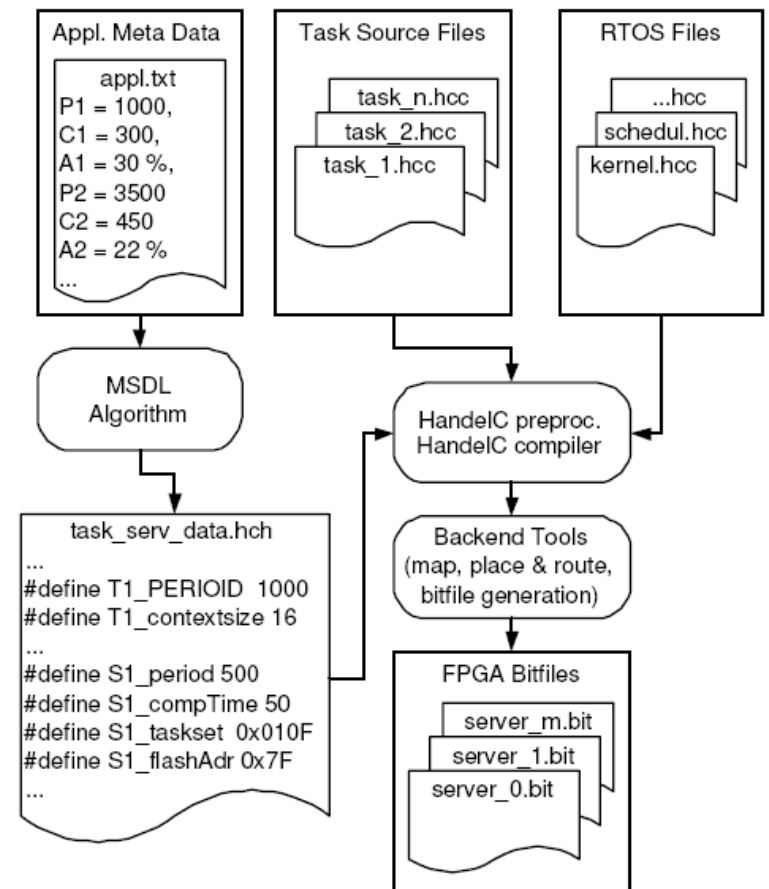
■ including reconfiguration time overheads in the analysis

- ◆ FPGA reconfiguration time τ_{RC}
- ◆ EDF: number of preemptions per request bounded by $N_i = \sum_{\Gamma/T_i} \left\lfloor \frac{P_i}{P_k} \right\rfloor$
- ◆ for MSDL
 - consider set of servers Ω , compute N_i
 - increase C_i by $(1+N_i) \cdot \tau_{RC}$ and check whether $U^T(\Omega) \leq 1$
- ◆ for partitioned EDF
 - consider each partition separately
 - increase C_i by $(1+N_i) \cdot \tau_{RC}$ and check whether $U^T(G_i) \leq 1$
- ◆ global EDF
 - tricky, as a preemption affects all running tasks

An All-hardware MSDL Implementation



Virtex-II3000

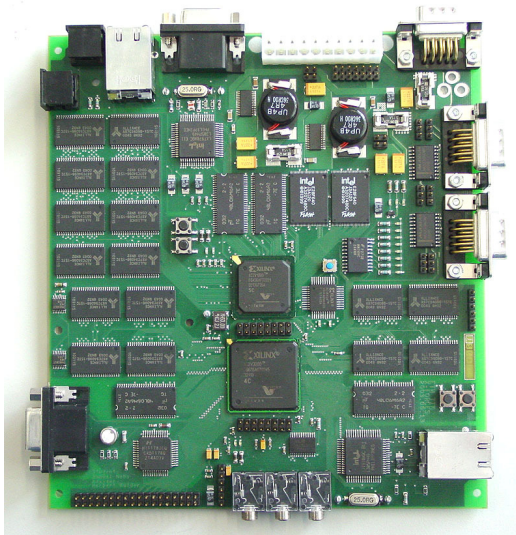


- ◆ area overhead (unoptimized, no MMU)
 - 16% LUTs, 9% FFs for 15 tasks and 9 servers
- ◆ time overhead
 - zero if no context switch required
 - a few hundred cycles + τ_{RC}

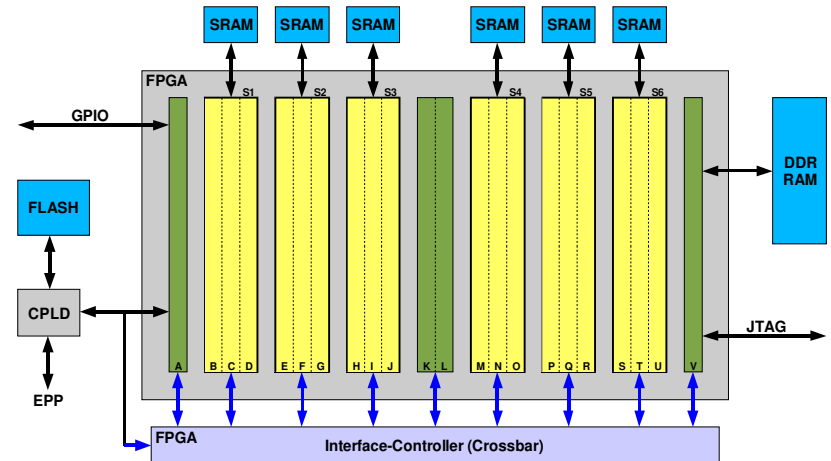
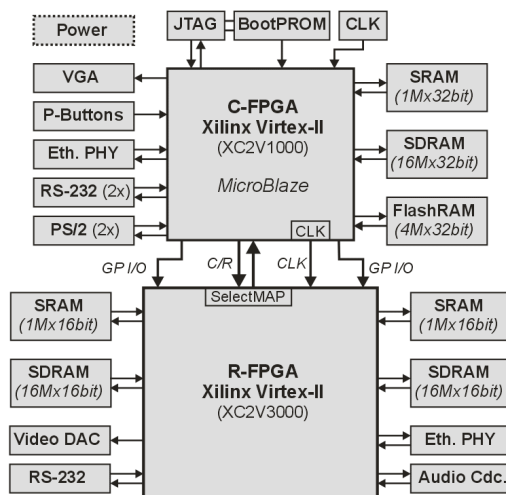
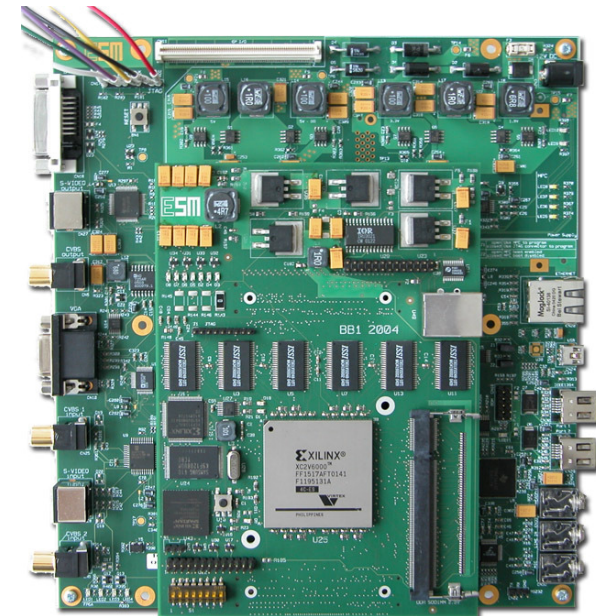
Outlook

- implementation using slot-based partial reconfiguration

XF-Board



Erlangen Slot Machine



Summary

- what did we achieve?
 - ◆ mapping techniques for periodic realtime tasks
 - ◆ efficient tests (sometimes pessimistic)
 - ◆ inclusion of overheads in the analysis (even more pessimistic)
 - ◆ ideas how to realize this with full/partial reconfiguration modes

- more detailed modelling needed!
 - ◆ application models
 - ◆ "microarchitectural" models to derive better, ie. less pessimistic, tests

- given current reconfiguration times, is this work useful?