The ReconOS Project: Real-Time Multitasking on Reconfigurable Architectures

Klaus Danne, Enno Lübbers and Marco Platzner
{danne, enno.luebbers, platzner}@upb.de
Computer Engineering Group
University of Paderborn

Colloquium of DFG SPP 1148 (Rekonfigurierbare Rechensysteme)
June 2006, Darmstadt
Motivation

**trends**
- embedded systems based on reconfigurable hardware (e.g. system on chip)
- device densities increase, partial reconfigurability improves
- many applications have dynamic task sets

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

**investigate techniques to execute periodic real-time tasks**
- models different from single- / multiprocessor
- rarely studied for reconfigurable hardware
Overview

✓ motivation

■ models, metrics, goals

■ three scheduling approaches
  • global EDF
  • partitioned EDF
  • server based

■ comparison and realization

■ reconfigurable hardware operating system (ReconOS)
  • concept
  • current work and next steps
Models and Metrics

- set of periodic real-time tasks $\Gamma = \{T_1, \ldots, T_n\}$
  - relative deadline equals period $P_i$
  - 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 \( \Gamma \) 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 \( \Gamma \)

<table>
<thead>
<tr>
<th>( T_i )</th>
<th>( P_i )</th>
<th>( C_i )</th>
<th>( A_i )</th>
<th>( U^T(T_i) )</th>
<th>( U^S(T_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_1 )</td>
<td>4</td>
<td>2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>( T_2 )</td>
<td>6</td>
<td>5</td>
<td>1/4</td>
<td>5/6</td>
<td>5/24</td>
</tr>
<tr>
<td>( T_3 )</td>
<td>12</td>
<td>3</td>
<td>3/4</td>
<td>1/4</td>
<td>3/16</td>
</tr>
<tr>
<td>( T_4 )</td>
<td>12</td>
<td>2</td>
<td>1/4</td>
<td>1/6</td>
<td>1/24</td>
</tr>
</tbody>
</table>

\[ \begin{array}{c|c}
T & U^T(T) & U^S(T) \\
--- & --- & --- \\
 & 1.75 & 0.69 \\
\end{array} \]
Scheduler Goals:

- high scheduling performance
  - meet all task deadlines under high device utilization

- efficient schedulability test
  - guarantee at design time, that no deadline will be missed

- practicability
  - reasonable assumptions
  - few realization issues
  - low overhead
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

<table>
<thead>
<tr>
<th>$T_i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$A_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>4</td>
<td>2</td>
<td>1/2</td>
</tr>
<tr>
<td>$T_2$</td>
<td>6</td>
<td>5</td>
<td>1/4</td>
</tr>
<tr>
<td>$T_3$</td>
<td>12</td>
<td>3</td>
<td>3/4</td>
</tr>
<tr>
<td>$T_4$</td>
<td>12</td>
<td>2</td>
<td>1/4</td>
</tr>
</tbody>
</table>
Global EDF Schedulability

- "simulate" through the hyper period of the task set
  - which is unrealistic for all but the smallest task sets

- schedulability test based on task parameters

\[
\forall T_k \in \Gamma : \\
U^S(\Gamma) \leq (A(H) - A_{max}) \cdot (1 - U^T(T_k)) + U^S(T_k)
\]

- can be evaluated in linear time
- sufficient, but not necessary
- uses resource augmentation approach and is based on the multi-processor scheduling test of [Goossens, Funk & Baruah, 2003]

[Danne, K. & Platzner, M. An EDF Schedulability Test for Periodic Tasks on Reconfigurable Hardware Devices, (LCTES 2006)]
Partitioned EDF Scheduling

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

- solved by
  - integer linear programming
  - heuristic Next-Fit Decreasing-Area

example:

\[ G_1 = \{T_1, T_3, T_4\} \]
\[ G_2 = \{T_2\} \]

<table>
<thead>
<tr>
<th>$T_i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$A_i$</th>
<th>$U^T(T_i)$</th>
<th>$U^S(T_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>4</td>
<td>2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$T_2$</td>
<td>6</td>
<td>5</td>
<td>1/4</td>
<td>5/6</td>
<td>5/24</td>
</tr>
<tr>
<td>$T_3$</td>
<td>12</td>
<td>3</td>
<td>3/4</td>
<td>1/4</td>
<td>3/16</td>
</tr>
<tr>
<td>$T_4$</td>
<td>12</td>
<td>2</td>
<td>1/4</td>
<td>1/6</td>
<td>1/24</td>
</tr>
</tbody>
</table>

1.75 0.69

[Danne, K. & Plattner, M. Partitioned Scheduling of Periodic Real-Time Tasks onto Reconfigurable Hardware (RAW 2006)]
Server-based Scheduling

■ approach
  • reduce the number of different configurations by grouping tasks together

■ server task $S_i$
  • artificial periodic task that reserves area and execution time for other tasks
    \[ S_i = (R_i, P_i, C_i, A_i) \]
    \[ R_i = \{T_a, T_b, \ldots \} \subseteq \Gamma \]
  • when the server $S_i$ executes, all tasks $R_i$ execute
  • the server area $A_i$ is the cumulative area over all tasks $R_i$

■ find a set of servers $\Omega$ that
  • can be sequentially executed by EDF, ie. $U^T(\Omega) \leq 1$
  • reserve sufficient time and area, such that all tasks meet their deadlines
### MSDL Scheduling

#### Example

<table>
<thead>
<tr>
<th>$S_i$</th>
<th>$R_i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$A_i$</th>
<th>$U^T_i$</th>
<th>$U^S_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$T_1$</td>
<td>4</td>
<td>2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$T_2$</td>
<td>6</td>
<td>5</td>
<td>1/4</td>
<td>5/6</td>
<td>5/24</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$T_3$</td>
<td>12</td>
<td>3</td>
<td>3/4</td>
<td>1/4</td>
<td>3/16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.58</td>
<td>0.65</td>
</tr>
</tbody>
</table>

| $S'_1$ | $T_1$ | 4     | 0     | 1/2   | 1/2     | 1/4     |
| $S'_2$ | $T_2$ | 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   | 1/2     | 1/8     |
| $S'_3$ | $T_3$ | 12    | 0     | 3/4   | 1/4     | 3/16    |
| $S'_4$ | $T_1, T_2$ | 4 | 2 | 3/4 | 1/2 | 3/8 |
| $S'_5$ | $T_2, T_3$ | 6 | 3 | 1 | 1/2 | 1/2 |
|       |       |       |       |       | 1.12    | 0.88    |

FPGA area

[Smaller text]

[Smaller text]

trading area for time

[Danne, K. & Platzner, M. A Heuristic Approach to Schedule Periodic Real-Time Tasks on Reconfigurable Hardware (FPL 2005)]

[Danne, K. & Platzner, Periodic Real-Time Scheduling for FPGA Computers (WISES 2005)]
Prototype Implementation for MSDL

- pure FPGA approach (no CPU)
  - mini RTOS in hardware on FPGA
  - implemented in Handel-C

- full device reconfiguration

- automatic system-synthesis
  - planning by MSDL scheduler
  - Handel-C pre-processor
  - Handel-C compiler + Xilinx backend tools

[Danne, K. & Muehlenbernd, R. & Platzner, M. Executing hardware tasks on dynamically reconfigurable devices under real-time conditions (FPL 2006)]
Scheduling Performance

- analytic result: no dominance among the three approaches
- simulation results (on randomly generated task sets)

parameters:

\[ C_i = \{1,2,\ldots,30\} \]
\[ U^T(T_i) = [0.2,0.4] \]
\[ A_i = [0.2,0.4] \]
## Summary of Scheduling Approaches

<table>
<thead>
<tr>
<th></th>
<th>Global EDF</th>
<th>MSDL</th>
<th>Partitioned EDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedulability</td>
<td>&quot;Simulation&quot; / pessimistic test</td>
<td>Construct servers</td>
<td>ILP / NFDA heuristic</td>
</tr>
<tr>
<td>Performance</td>
<td>High / low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Reconfiguration, area model</td>
<td>Partial, 1D variable, needs re-locatable tasks</td>
<td>Full</td>
<td>Partial, 1D slotted</td>
</tr>
<tr>
<td>Amount of configuration data</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Online scheduling</td>
<td>Yes</td>
<td>No</td>
<td>Maybe</td>
</tr>
</tbody>
</table>
Model Extensions

- reconfiguration overhead
  - analyzed and bounded for all three models
  - reasonably small, if configuration time < 10% of computation time
  - global EDF suffers more than partitioned EDF and MSDL

- tasks with implementation alternatives
  - e.g. fast & large vs. slow & small
  - improves scheduling performance

- tasks share RAM banks
  - schedule access to RAM ports

[Danne, K. & Platzner, M. Memory-demanding Periodic Real-Time Applications on FPGA Computers (Work-in-Progress proc. ECRTS 2005)]
Overview

✓ motivation

✓ models, metrics, goals

✓ three scheduling approaches
  • global EDF
  • partitioned EDF
  • server based

✓ comparison and realization

■ reconfigurable hardware operating system (ReconOS)
  • concept
  • current work and next steps
ReconOS

- flexible operating system for reconfigurable hardware
- simplifies embedded SoC development by providing common OS services for hardware and software tasks

programming model:
- operating system objects
  - tasks
  - shared memory
  - semaphores
  - queues/FIFOs
  - timers
  - ...

- services
  - task management
  - memory management
  - synchronization
  - communication
Execution Model

- OS is custom-tailored to application requirements
  - OS services and objects can be selectively included or omitted
  - efficient usage of resources (logic area, memory)
  - task set must be known at design time
  - parts of the OS can be implemented either in software or in hardware

- OS interface (OSIF) provides operating system services for hardware tasks

- uses partial reconfigurability
  - If supported by hardware
Current Work

- development platform: Xilinx ML403 evaluation board
  - Virtex-4FX FPGA with embedded PowerPC405
  - supports partial reconfiguration

- real-time operating system
  - modified eCos for PPC, ported to ML403 platform
  - OSIF hardware interface handles service requests from hardware tasks

```c
reconos_sem_post(o_osif, i_osif, handle);
```
```c
sysbus
```
```c
cyg_sem_post(handle)
```
```c
eCos
```
Next Steps

- implementation and evaluation of additional OS services
  - shared memory
  - timers, signals, queues
  - preemptive multitasking (full and partial reconfigurability)

- applications
  - image processing (e.g. visual object tracking)
  - cryptography
  - networking

- port to other reconfigurable platforms
  - XF-Board
  - Erlangen Slot Machine (ESM)
Overview

✓ motivation

✓ models, metrics, goals

✓ three scheduling approaches
  • global EDF
  • partitioned EDF
  • server based

✓ comparison and realization

✓ reconfigurable hardware operating system (ReconOS)
  • concept
  • current work and next steps