
FPGA Implementation of a Priority Scheduler Module

A. Jason, A. David, etc.

Presented by Guangyi Cao



outline

- Ø Motivation
- Ø Principle and feature
- Ø Design methodology
- Ø Results of implementation

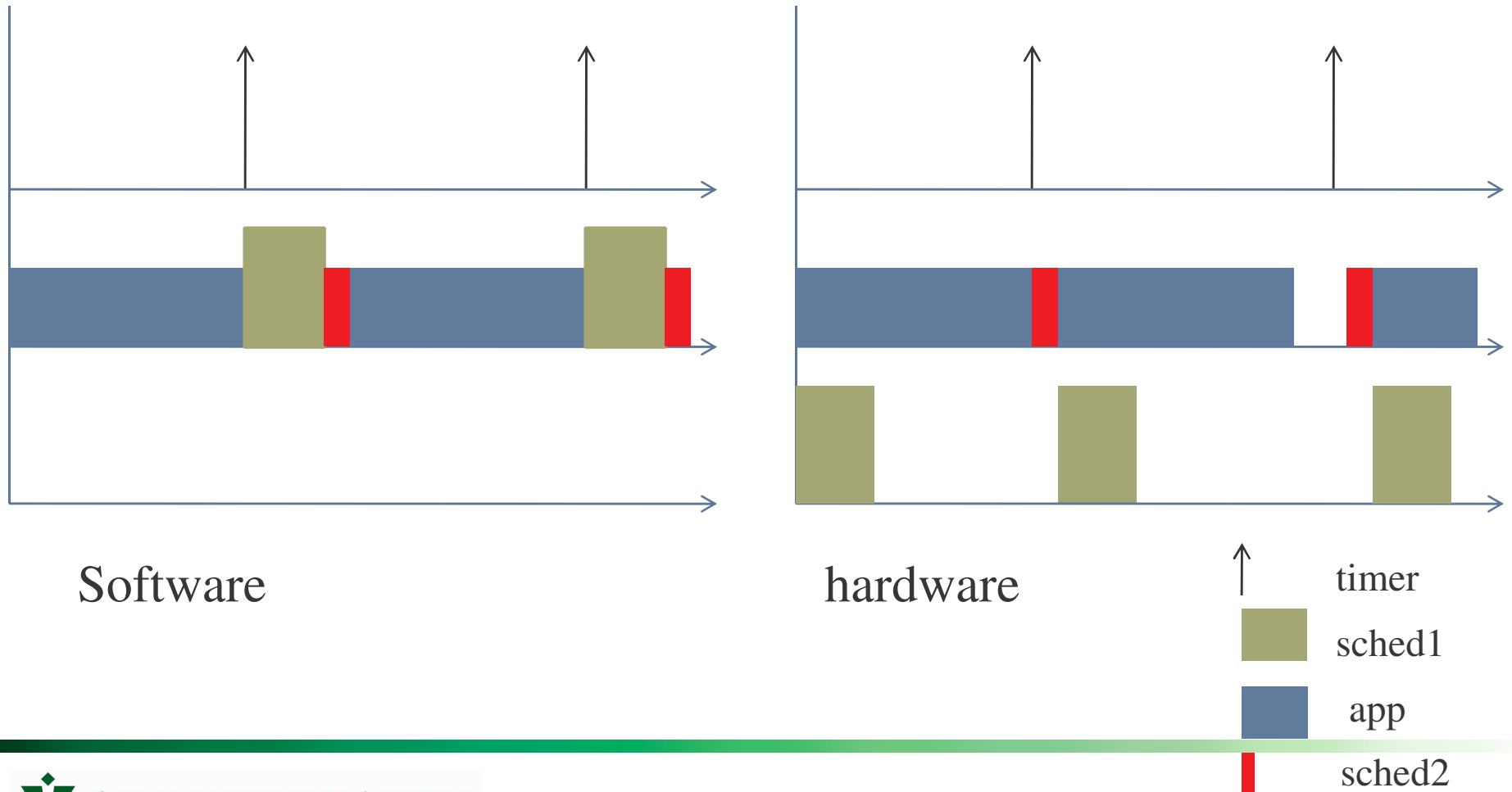


motivation

- Ø Migrate Scheduling design from RTOS into hardware module
- Ø Minimizing system overhead and jitter.
- Ø Provide more bandwidth to application programs
- Ø Adaptable and modifiable module



Principle and feature



Principle and feature

Ø Thread state information available to module

Ø Scheduling service directed toward module

- Enqueue or dequeue request from CPU
- Unblocking of semaphores
- timers

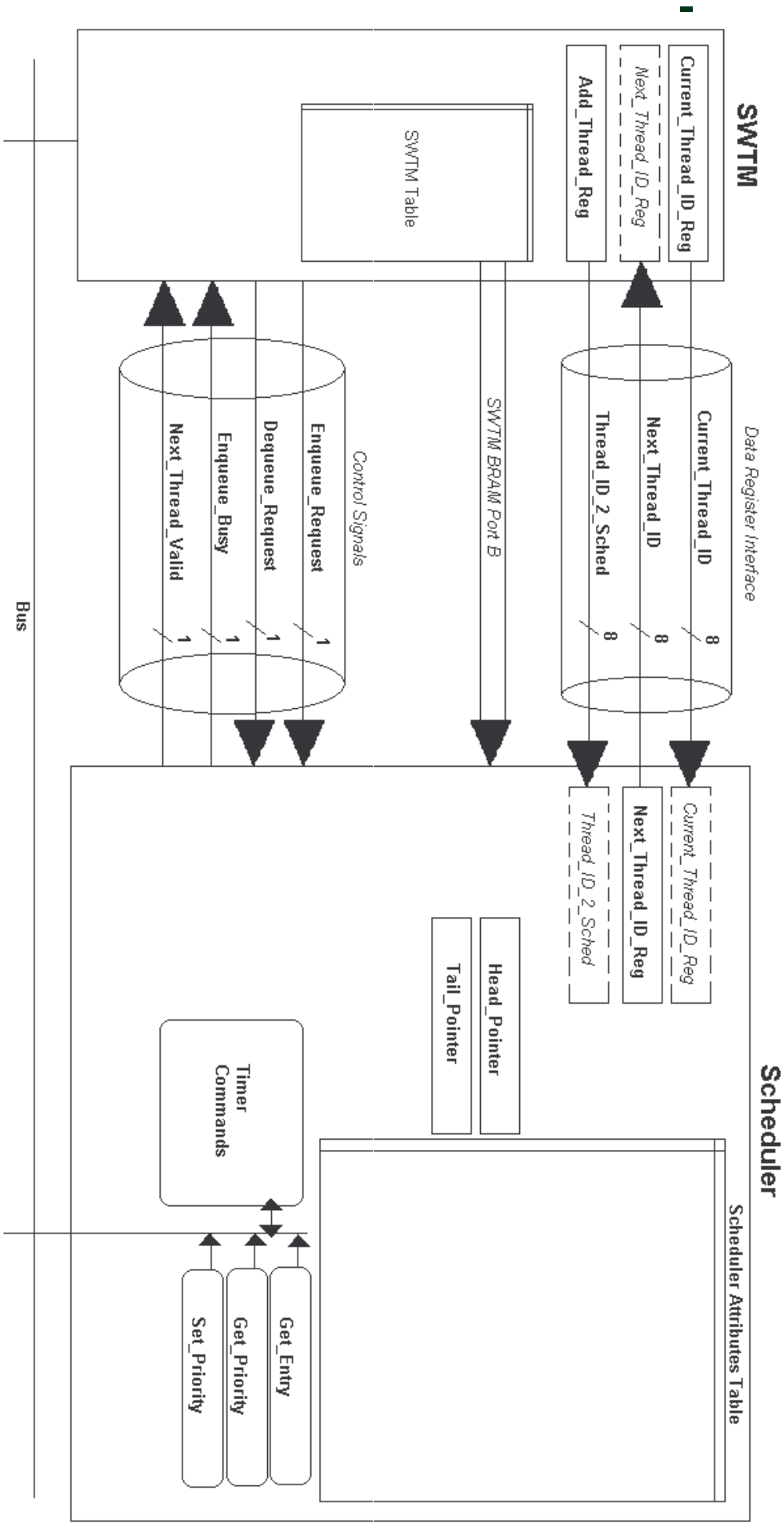


design

Ø Scheduling algorithm

- FIFO
- Round Robin
- Preemptive-priority scheduling





design

Ø Software thread manager and scheduler

- Enqueue dequeue control signal
- Current thread id and Which thread id to schedule

Ø scheduler attribute table

- Queue bit
- Next pointer
- Priority field

Ø Bus commands and SWTm commands



design

Ø Bus commands and SWTM commands

Command Type	Command Name	Command Actions
<i>SWTMcom</i>	Enqueue	Add a thread to the ready-to-run queue, update scheduling decision, and preempt if needed.
<i>SWTMcom</i>	Dequeue	Remove a thread from the ready-to-run queue, and calculate new scheduling decision (all in one pass through the queue).
<i>BUScom</i>	Get_Entry	Get the scheduler attribute entry for a thread.
<i>BUScom</i>	Get_Priority	Get the priority value for a thread.
<i>BUScom</i>	Set_Priority	Set the priority value for a thread, update scheduling decision, and preempt if needed.



design

Ø Scheduler attribute table

- FIFO structure
- Enqueue fast
- Dequeue slow

Ø Alternative: priority queue



simulation result

- Implemented in Memec 2vp7 development board
- Worst case for a dequeue operation take 10 us.
- Test in modelsim

# Threads	Time (ns)	Est. Per Thread Time(ns)
250	10060	40.24
128	5140	40.15625
64	2610	40.78125
32	1330	41.5625
16	690	43.125
2	130	65



Questions?

