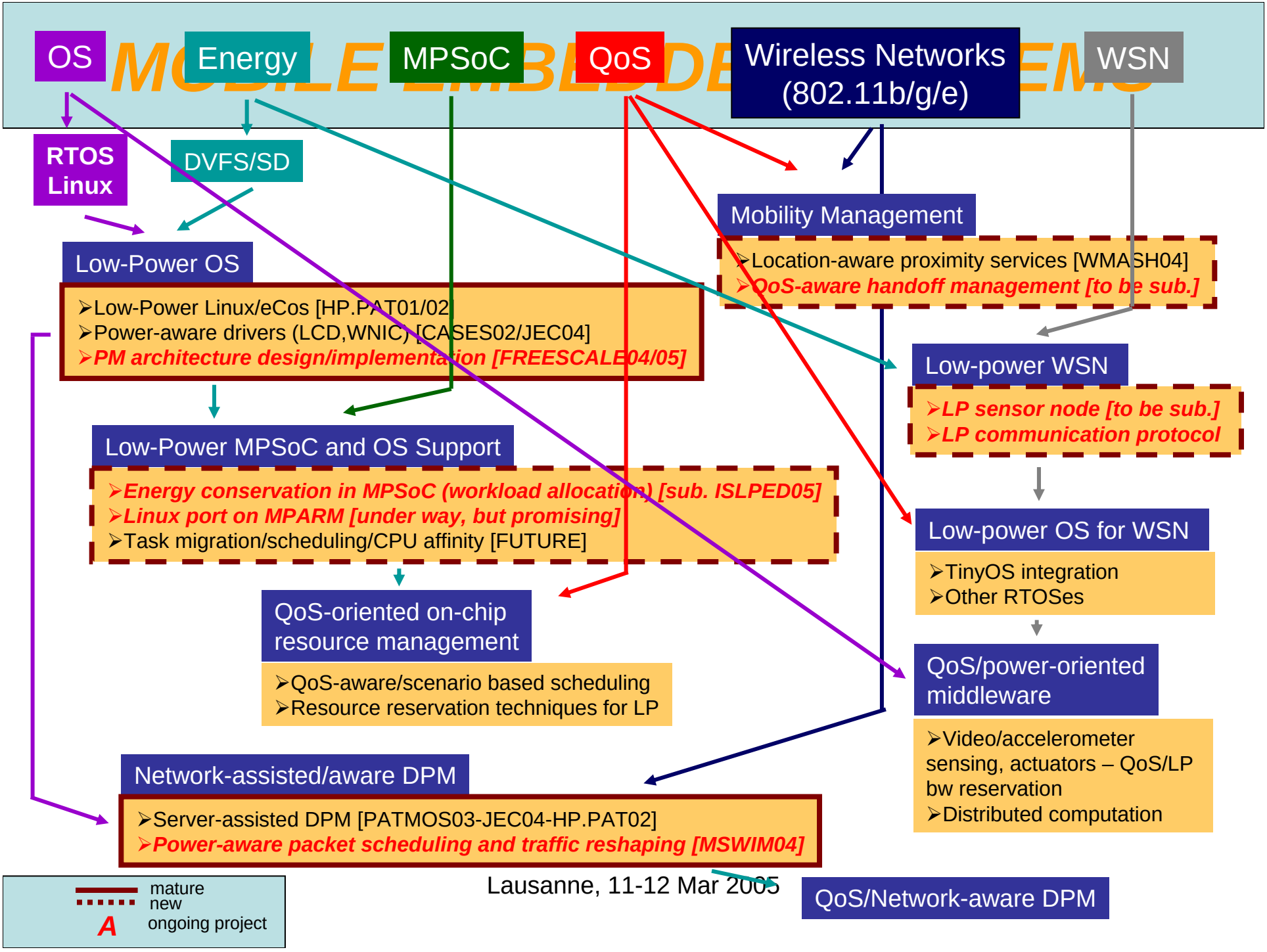
Energy & QoS in Mobile Platforms

Andrea Acquaviva

Lausanne, 11-12 Mar 2005

Research Picture

Lausanne, 11-12 Mar 2005


OS
Energy
MPSoC
QoS
Wireless Networks (802.11b/g/e)
WSN
RTOS Linux
DVFS/SD
Low-Power OS
➢Low-Power Linux/eCos [HP.PAT01/02]
➢Power-aware drivers (LCD,WNIC) [CASES02/JEC04]
➢PM architecture design/implementation [FREESCALE04/05]
Low-Power MPSoC and OS Support
➢Energy conservation in MPSoC (workload allocation) [sub. ISLPED05]
➢Linux port on MPARM [under way, but promising]
➢Task migration/scheduling/CPU affinity [FUTURE]
QoS-oriented on-chip resource management
➢QoS-aware/scenario based scheduling
➢Resource reservation techniques for LP
Network-assisted/aware DPM
➢Server-assisted DPM [PATMOS03-JEC04-HP.PAT02]
➢Power-aware packet scheduling and traffic reshaping [MSWIM04]
Mobility Management
➢Location-aware proximity services [WMASH04]
➢QoS-aware handoff management [to be sub.]
Low-power WSN
➢LP sensor node [to be sub.]
➢LP communication protocol
Low-power OS for WSN
➢TinyOS integration
➢Other RTOSes
QoS/power-oriented middleware
➢Video/accelerometer sensing, actuators – QoS/LP bw reservation
➢Distributed computation
QoS/Network-aware DPM
mature
new
A
ongoing project

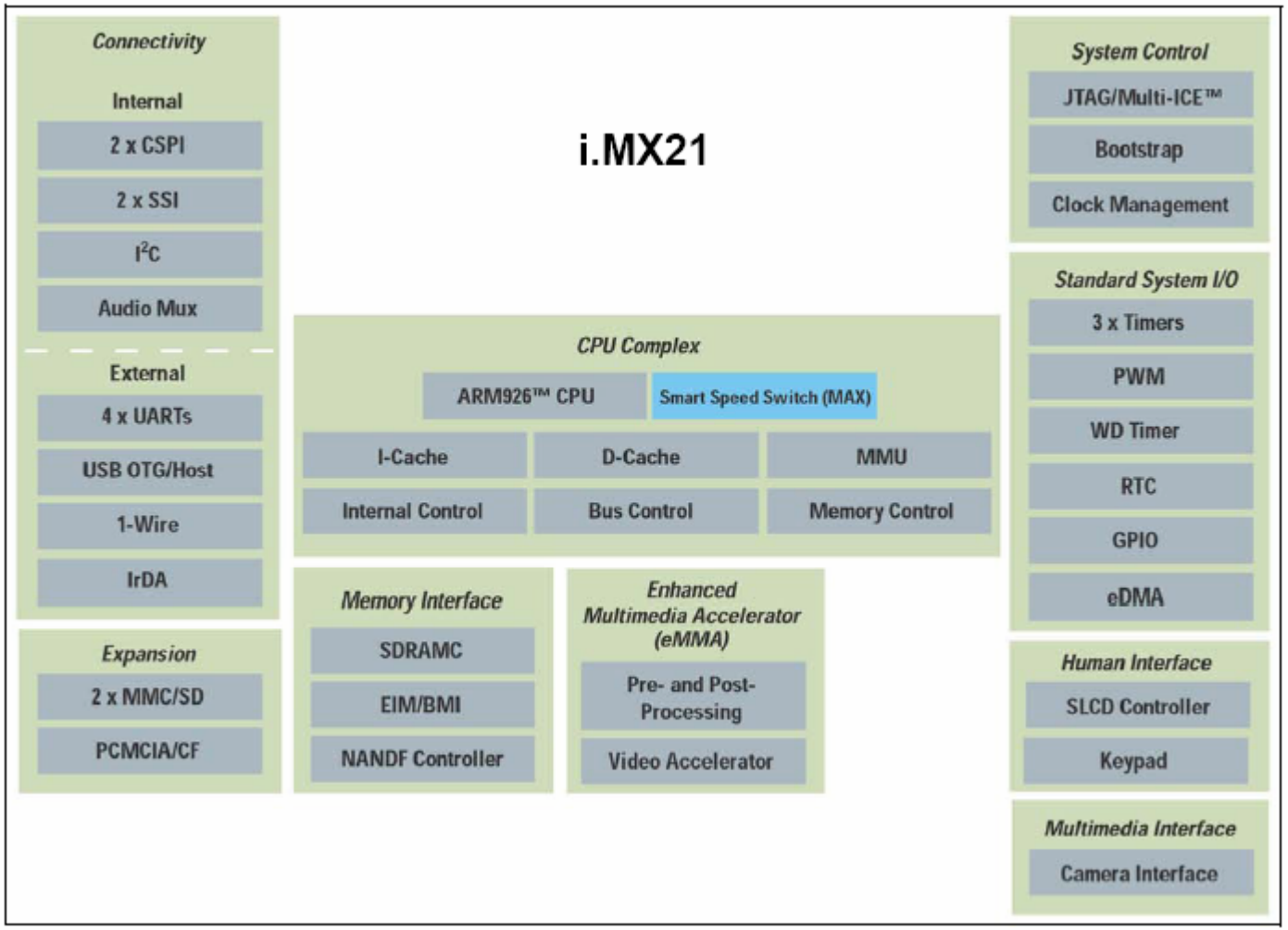
Ongoing Research Projects

Freescale XEC Project

- Description:
  - design and implementation of an OS-level power management architecture for supporting DVFS/shutdown of multimedia mobile platforms (ex. i.MX21 and next)
- Keywords:
  - **Energy** (DVFS/SD),**OS**, **QoS**
- Research focus:
  - Target base-band section
  - Design a flexible DPM infrastructure (portable to different applications and platforms)
  - Design a real-time power estimation model (fast architectural adaptation)
  - Handle compliancy with industrial standards and third-party software, platform and OS independency (Symbian, Embedded Linux, WinCE)
- Set-up:
  - high level modeling (simulink) and real-hardware implementation and evaluation (now on i.MX21 platform)
- Started:
  - July 2004
- Collaboration:
  - Freescale Semiconductor, BolognaU: Luca Benini, Martino Ruggiero

Ongoing Research Projects

Freescale XEC Project

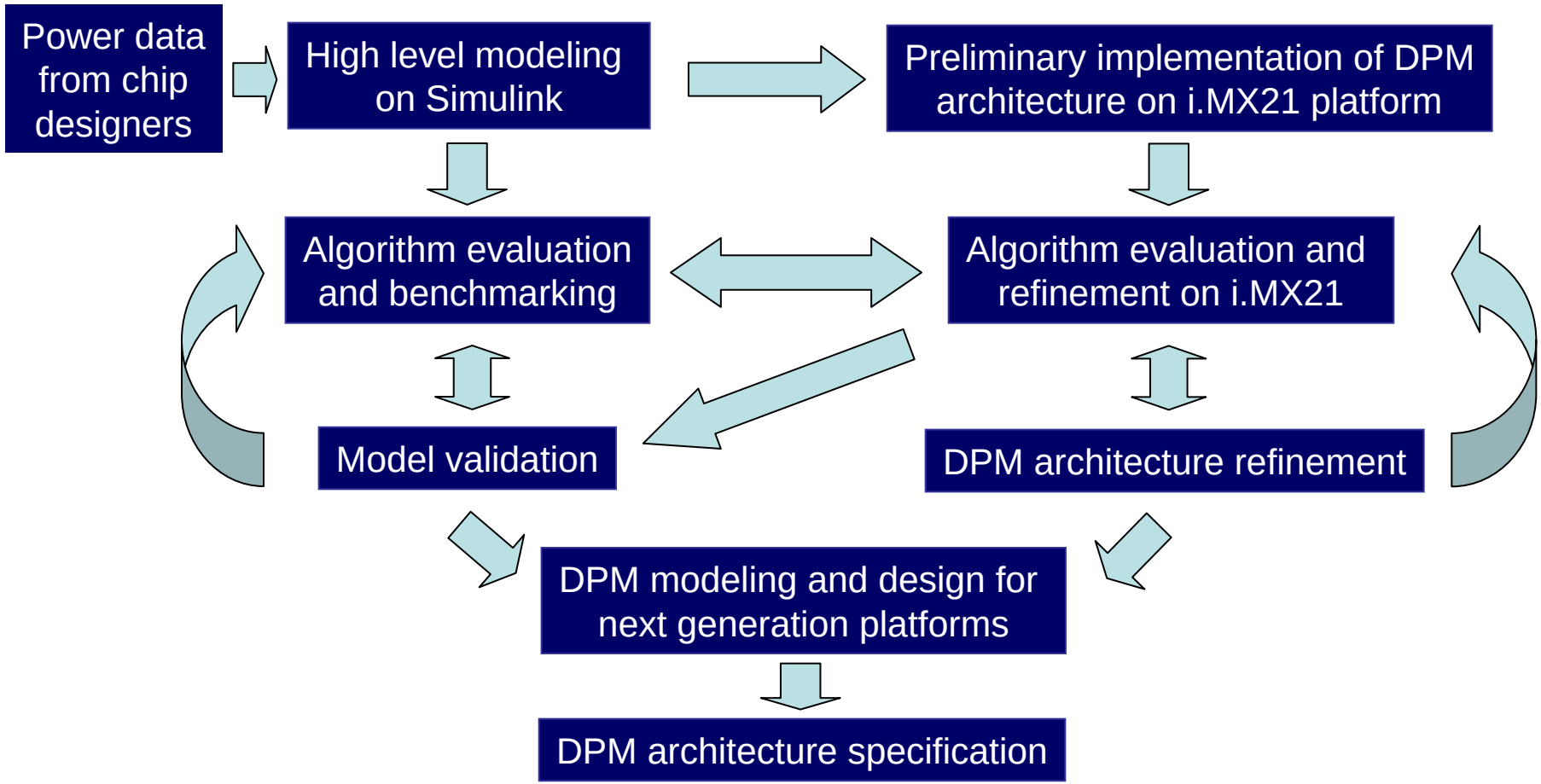


i.MX21 functional block diagram

XEC Project

Recent Results

Project Flow

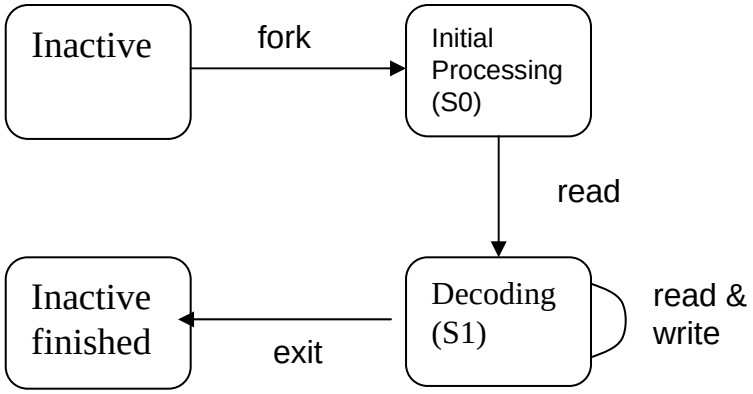


XEC Project

Recent Results

Modeling

- High level event-driven system model for DPM policy evaluation and benchmarking
  - OS-accurate (system call level)
  - Not functionally accurate
  - Provide real-time simulation and flexibility
  - Power estimation based on Power State Machine (PSM) [Benini00]
- Workload is specified through its OS interactions (system calls)



XEC Project

Recent Results

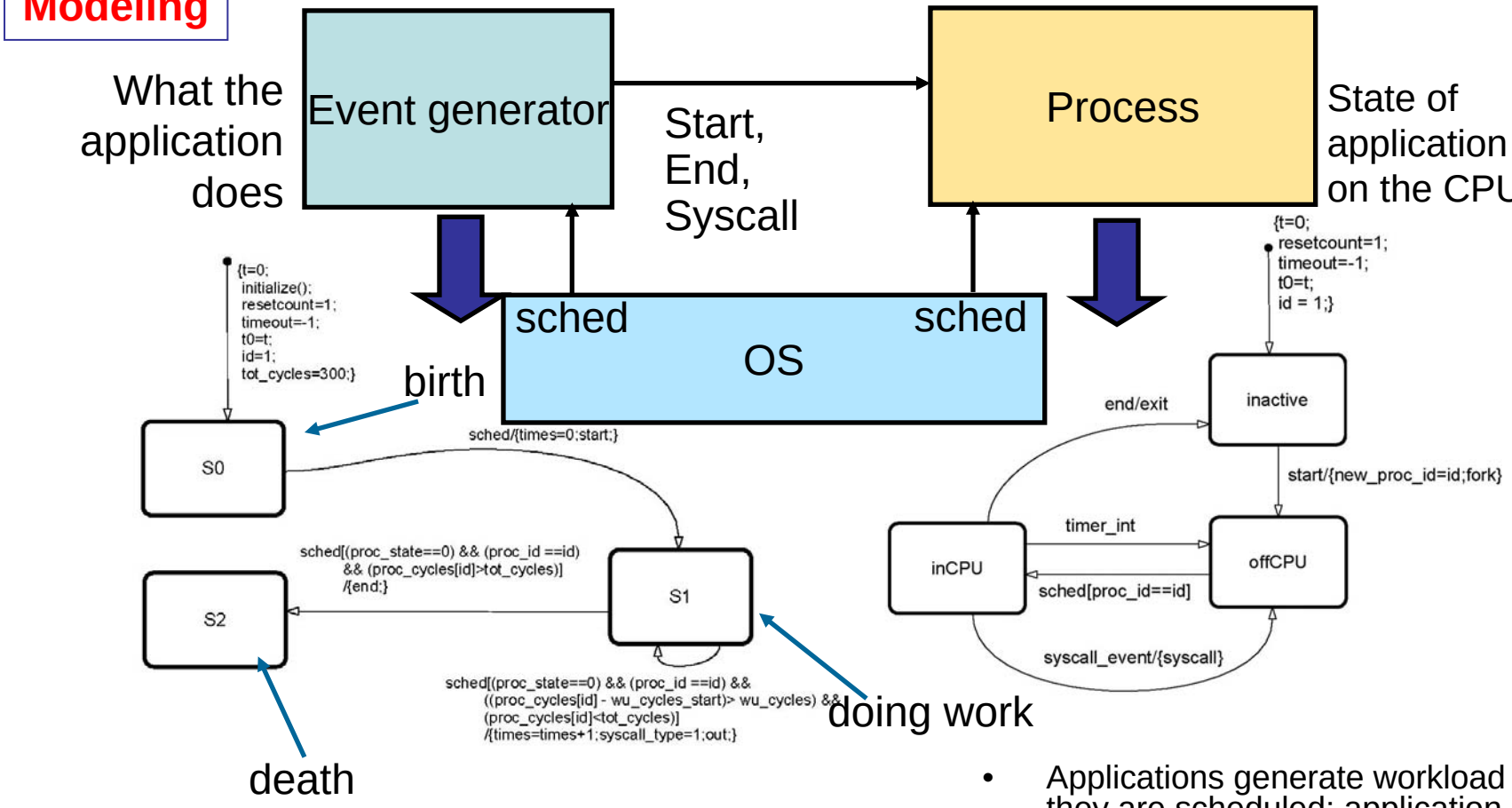
Modeling

Activity	*Syscall type*	*Device or sema ID*	*Event Type*	*CPU cycles*	*MEM cycles*	*State*	*Trigger*
Creation	fork	-	-	-	XXX	-	Latency=10sec
Initial Processing	-	-	-	Y	Z	S0	-
Read from network	read	Network interface		-		-	-
Decoding	-	-	-			S1	-
Write to LCD	write	LCD		-		-	-
Read from network	read	Network interface		-		-	-
Decoding	-	-	-			S1	-
Write to LCD	write	LCD		-		-	-
…							
Destruction	exit	-	-			-	Decoding = 100

XEC Project

Recent Results

Modeling

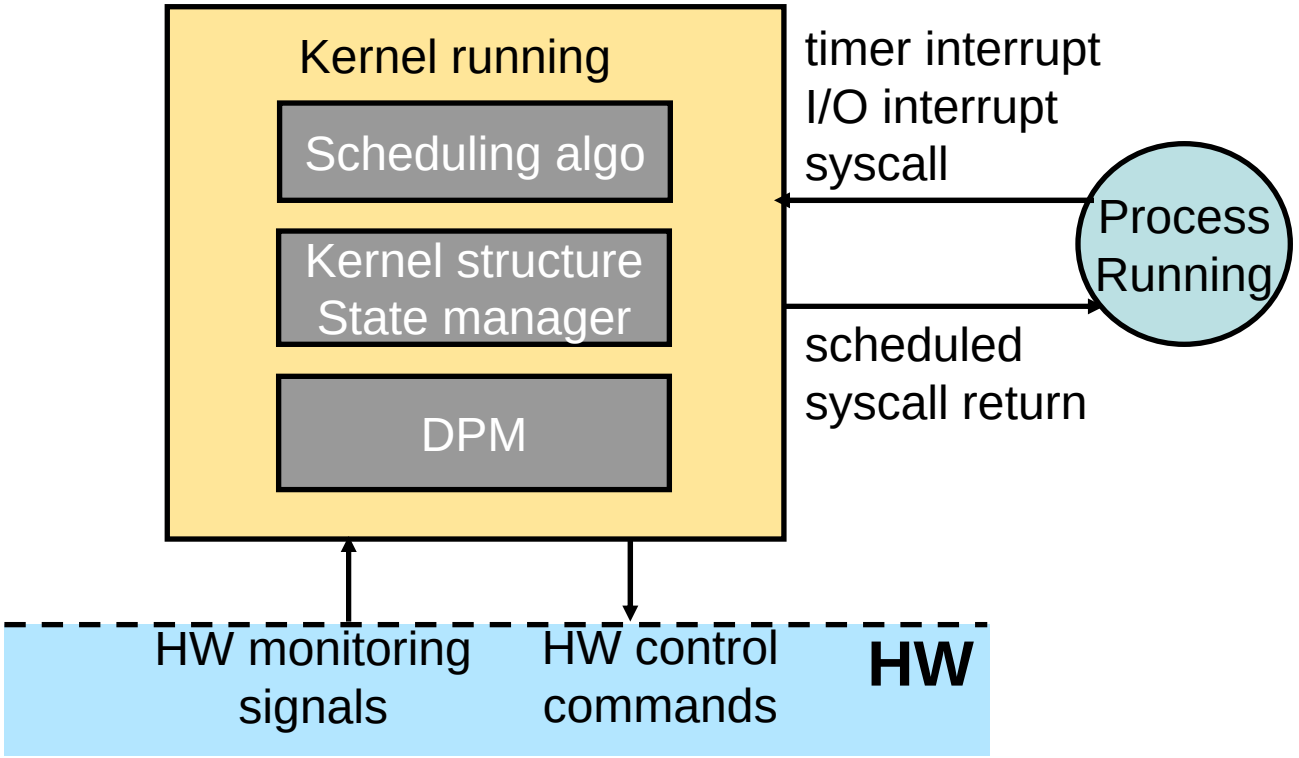


- Applications generate workload when they are scheduled: application workload is modulated by the OS

XEC Project

Recent Results

Modeling

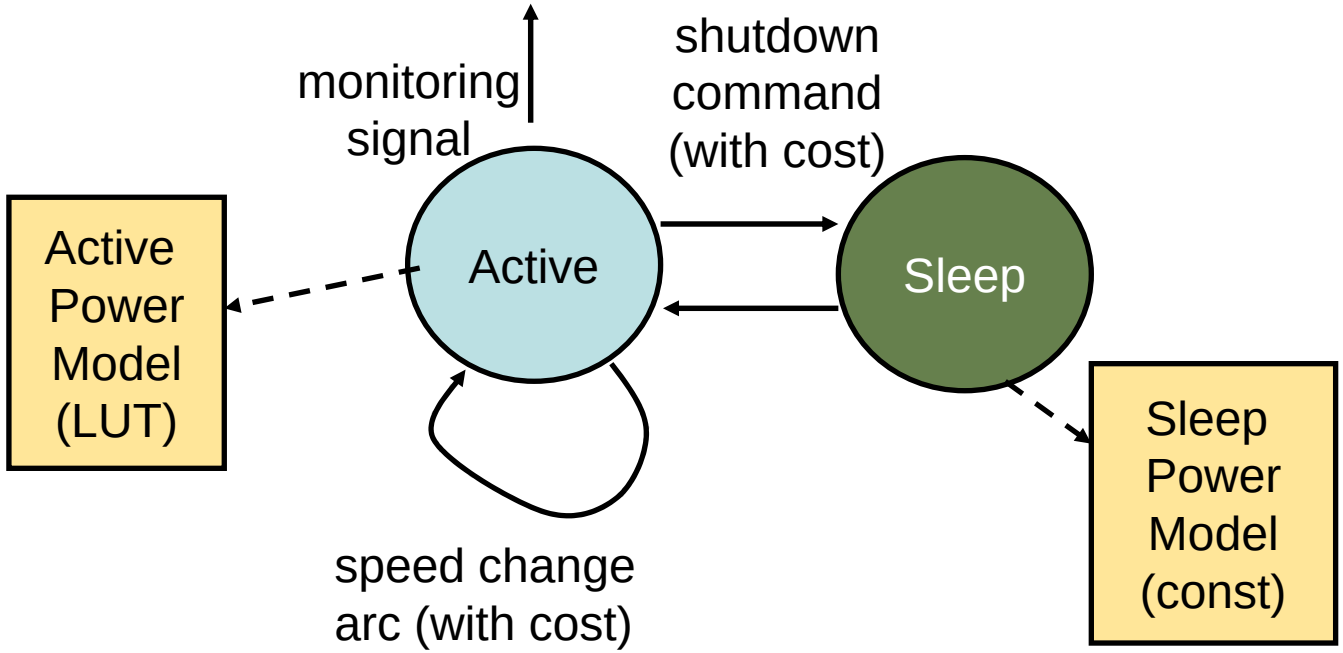


- Manages scheduler and all operations performed in kernel state

XEC Project

Recent Results

Modeling

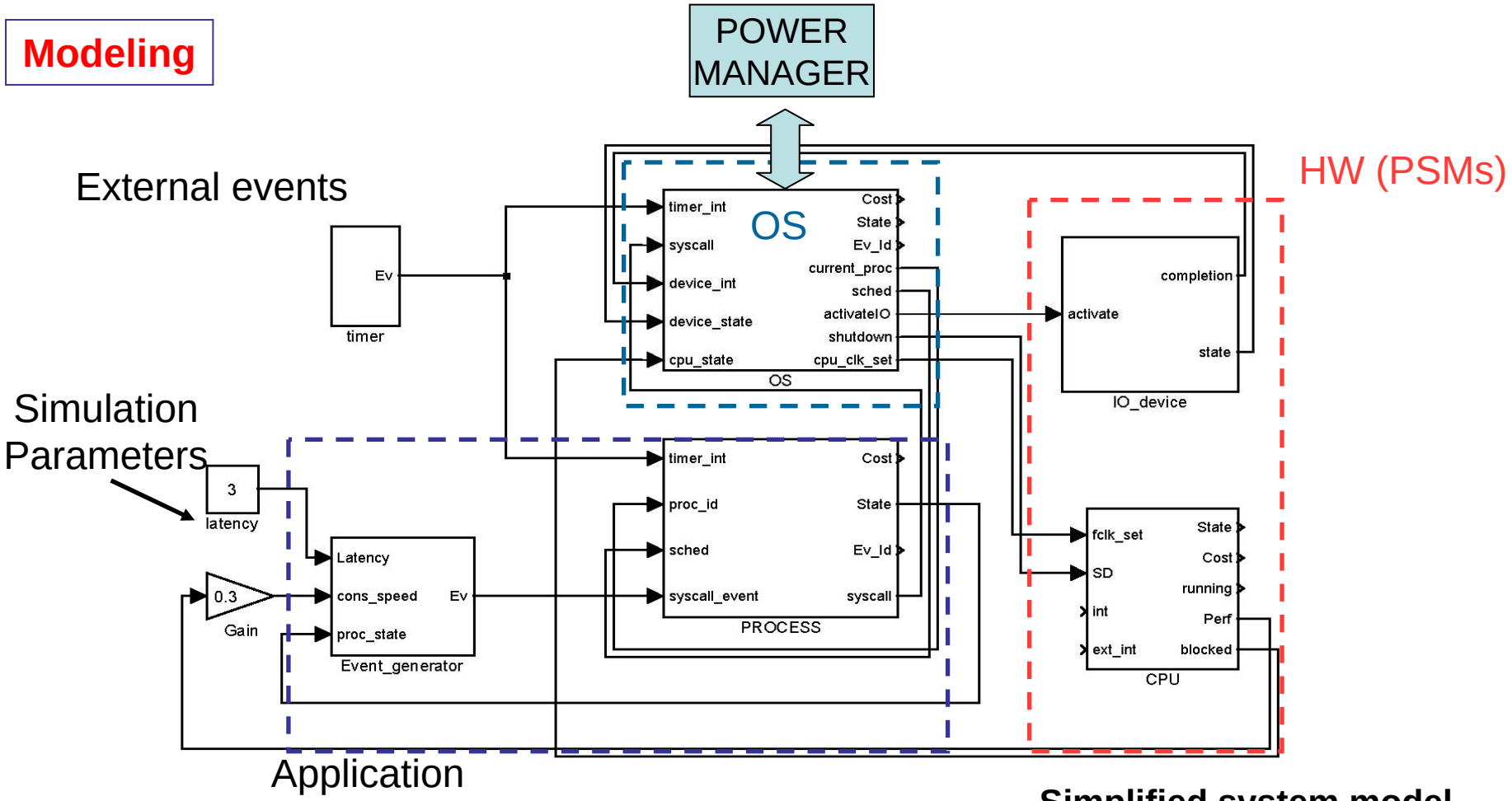


- *Power state machine*: provides monitoring information, includes power model
- CPU PSM and a PSM for every peripheral we are interested in modeling

XEC Project

Recent Results

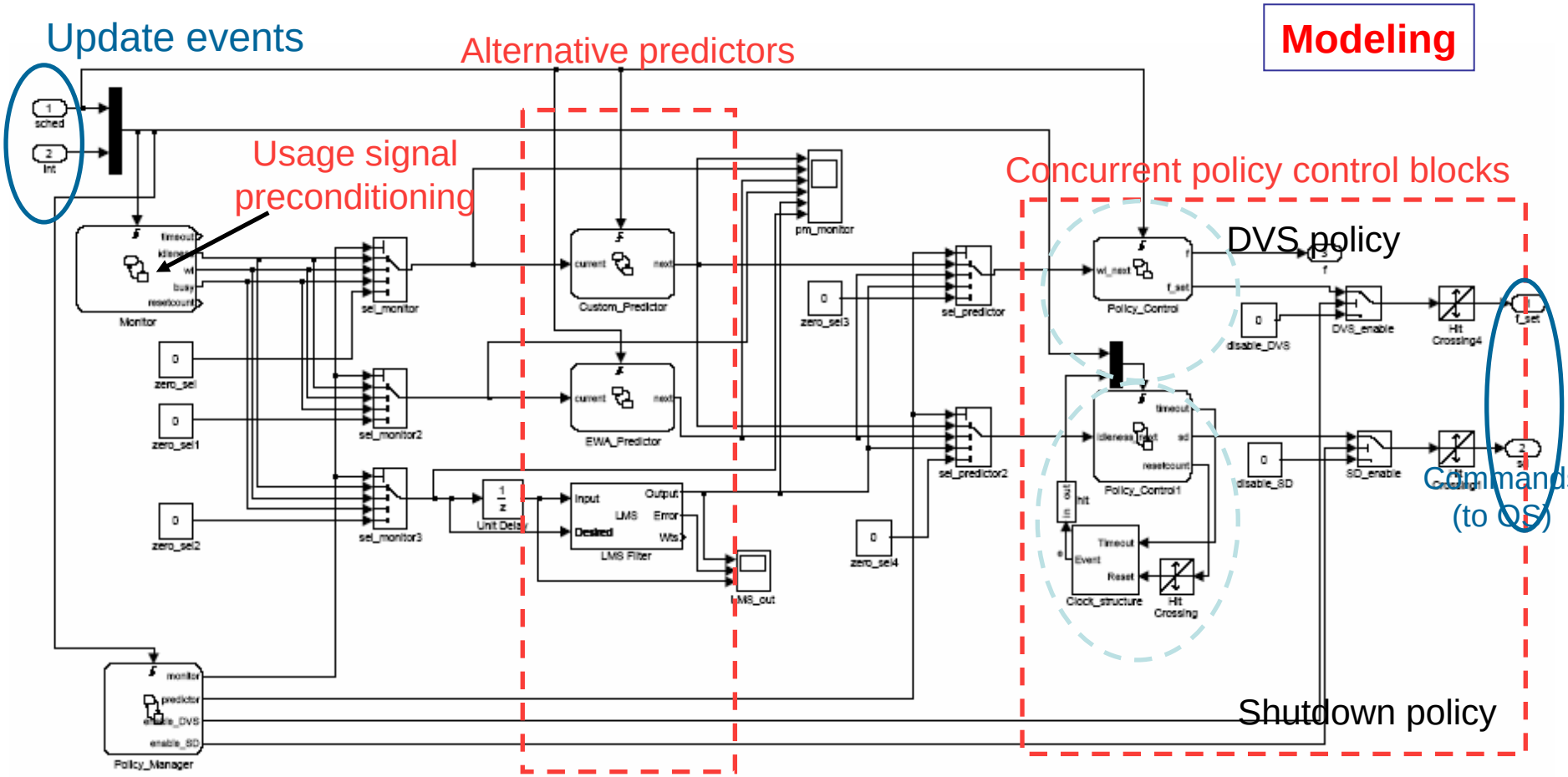
Modeling



Simplified system model

XEC Project

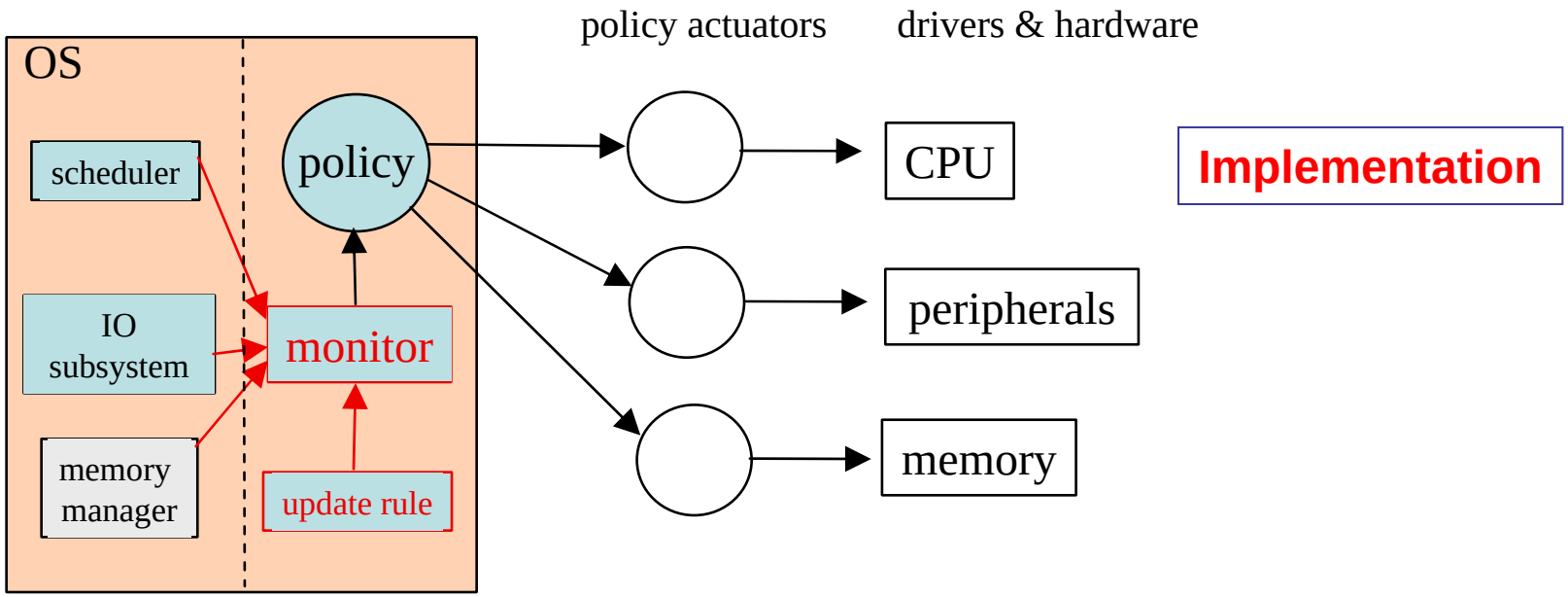
Recent Results



Lausanne, 11-12 Mar 2005

XEC Project

PARASITIC DPM

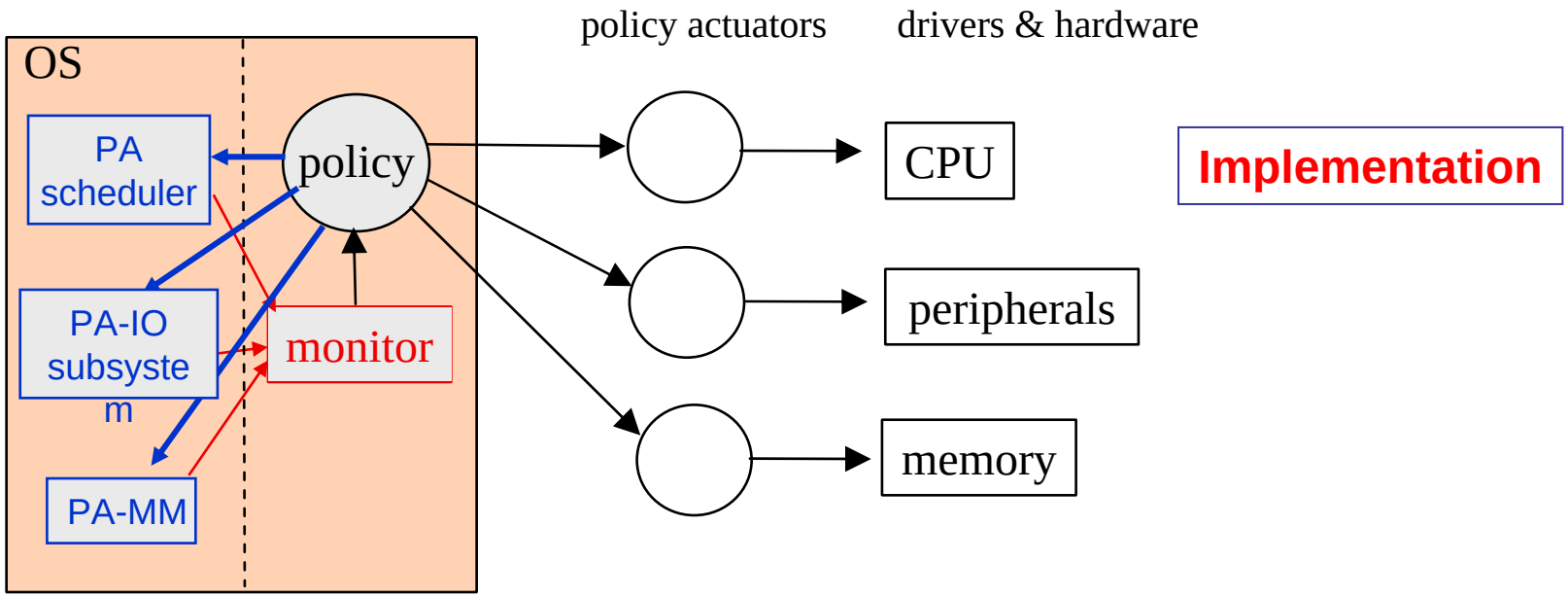


- Monitor receives info from OS components
- Update rule establish decision points
- Both can be at kernel level or user level

- Hooks may be required for monitoring and call back

XEC Project

INTEGRATED DPM



kernel modifications are required for

- Power aware scheduling
  - Task order is affected by policy
- The policy is inside the scheduler

- Insert kernel hooks for call-back functions
- Call-back functions trigger
  - kernel modules to override scheduling policy
  - Perform performance monitoring

XEC Project

Recent Results

- Preliminary XEC architecture implementation on i.MX21
- First implementation of **ARM IEM** (Vertigo) performance prediction algorithm
  - Per-task (not overall) processor utilization
  - Idleness prediction over task natural period (bounded by yield system calls)
- Both model and HW implementation highlight Vertigo shortcomings
  - Task slack time overestimation in presence of blocking system calls
  - Frequency oscillation in presence of multiple tasks
  - Non stationary workload generated by monolithic applications (overshooting does not work) or videogame-like benchmarks
  - Applications using on-chip video codec (ex: eMMA) (handle effects of non-independent clock domains)
- Possible solution: switch between DVFS/shutdown (on-off) policies depending on average system idleness level

XEC Project

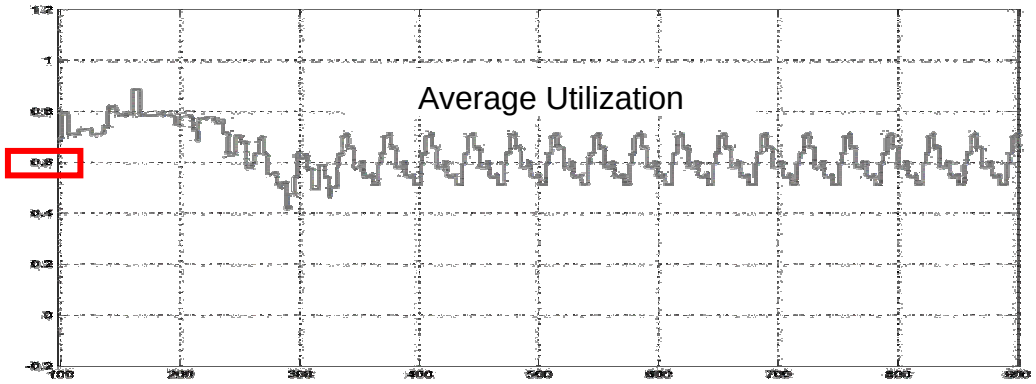
Recent Results

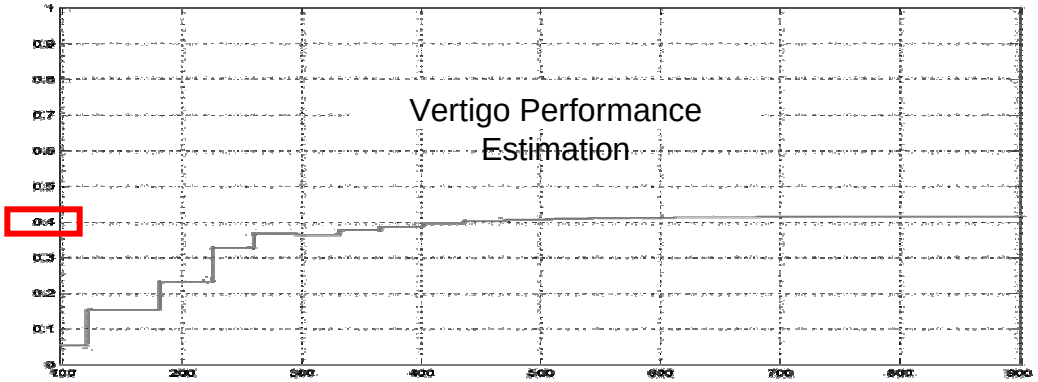
Modeling

Effects of Blocking system calls

Considering blocking idle periods in deadline computation lead to overestimation of slack time and underestimation of required performance. As a consequence, the speed of the processor will be set to a value lower than the optimal one

A decoding task is accessing a video device using non-blocking system calls and another output device (say and audio codec) using blocking system calls. A large number of deadline misses (87%) is obtained without using any guarding bound





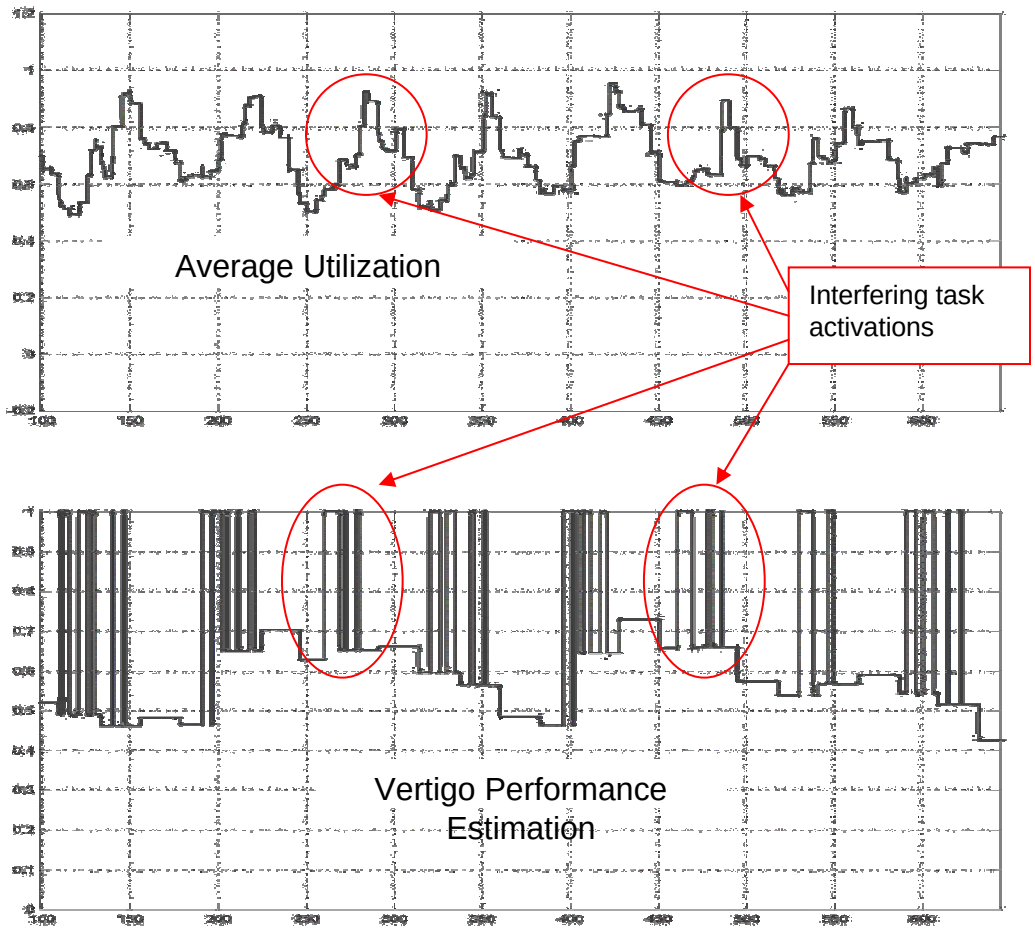
XEC Project

Recent Results

Modeling

Effects of multiple tasks

Another problem is that when multiple processes are in the ready queue, the performance value must be updated each scheduling interval. This is because processor speed is computed task by task. If processes have strongly different performance requirements, this leads the frequency to oscillate with large steps.

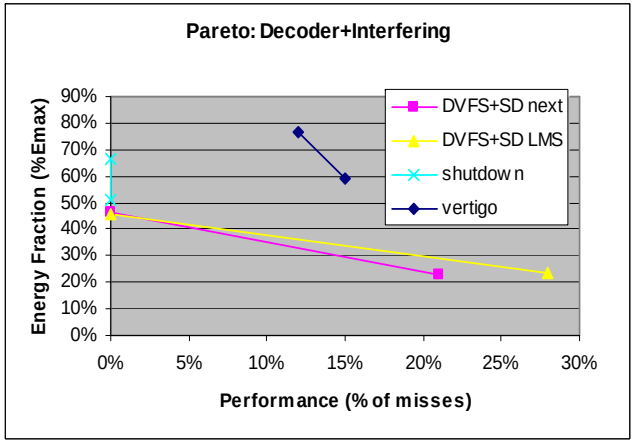


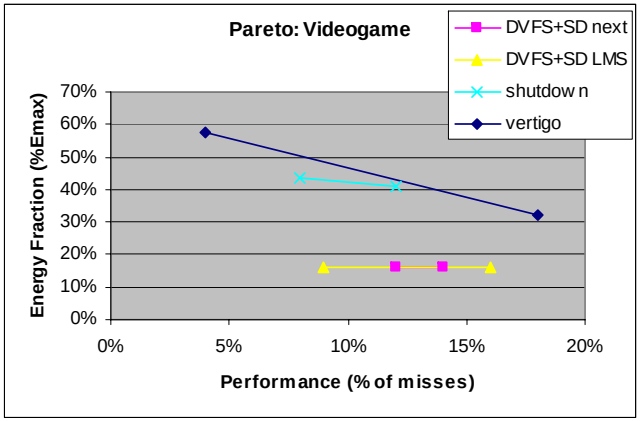
XEC Project

Recent Results

Modeling

Our test policy is based on a simple next or LMS prediction and a more complex frequency setting policy. Since our prediction is not task based, the arrival of new tasks causes deadline misses, that we can avoid by adding an overshoot factor. We believe that by combining the two approaches we could obtain an efficient prediction and frequency setting policy.





XEC Project

Ongoing Work

- Preliminary model validation
  - Average core power consumption is within 10% from real HW (i.MX21) using mp3 audio and mp2 video benchmarks
  - Measure averaged over a 1 minute interval
- The model is built to simulate at least 1 hour
  - What is the error on energy estimation?
- Known issues:
  - HW: not modeled on-chip components, core functional details
  - SW: platform dependent OS implementation (syscall/ISR overhead)
- The model is under refinement

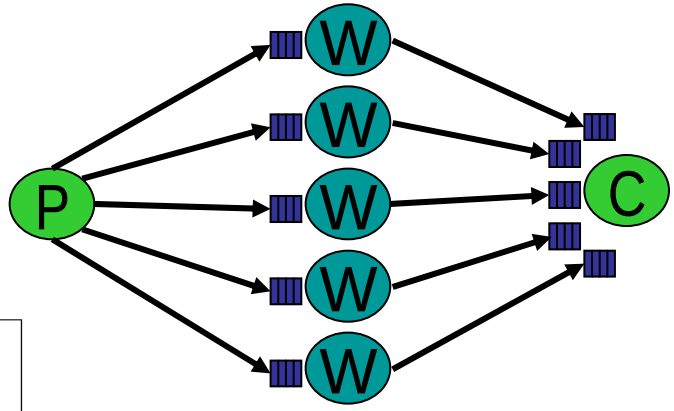
Energy Conservation in MPSoC

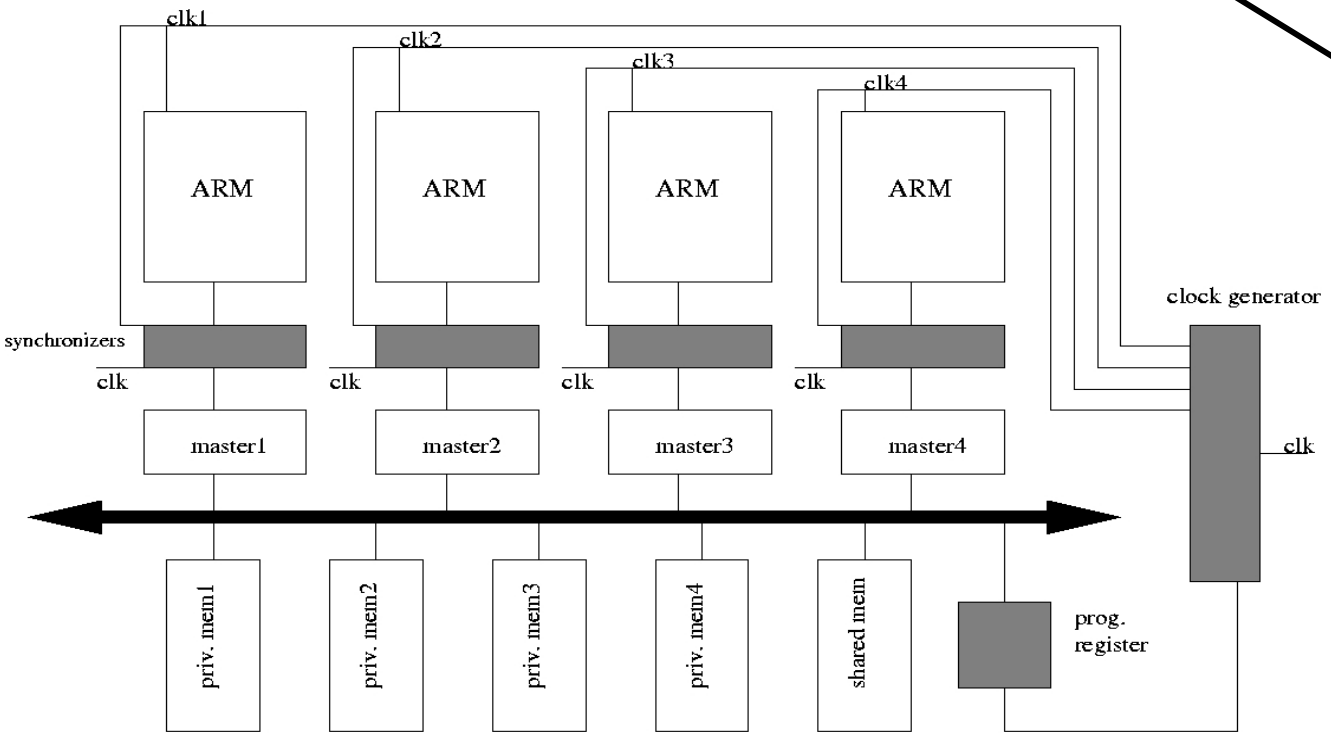
- Description:
  - Application specific power aware workload allocation for voltage scalable MPSoC platforms.
- Keywords:
  - **Energy conservation** (DVFS), workload allocation, **MPSoC**, **QoS**
- Research focus:
  - DES encryption algorithm (streaming application with uncorrelated data frames)
  - Static smart exploration of energy/throughput design space for different traffic conditions
  - Not all system resources are allocated to DES. QoS-based approach to workload allocation and voltage/frequency selection depending on traffic parameters.
- Set-up:
  - Timing-accurate simulation engine with variable frequency support (MPARM)
- Started:
  - Jan 2005
- Collaboration:
  - BolognaU: Martino Ruggiero, Davide Bertozzi, Luca Benini

Energy Conservation in MPSoC

Application & System Models

- Support for variable frequency cores in MPARM
- Mapping DES on MPSoC
  - A task for each core
  - 1 producer, 1 consumer task
  - N worker tasks

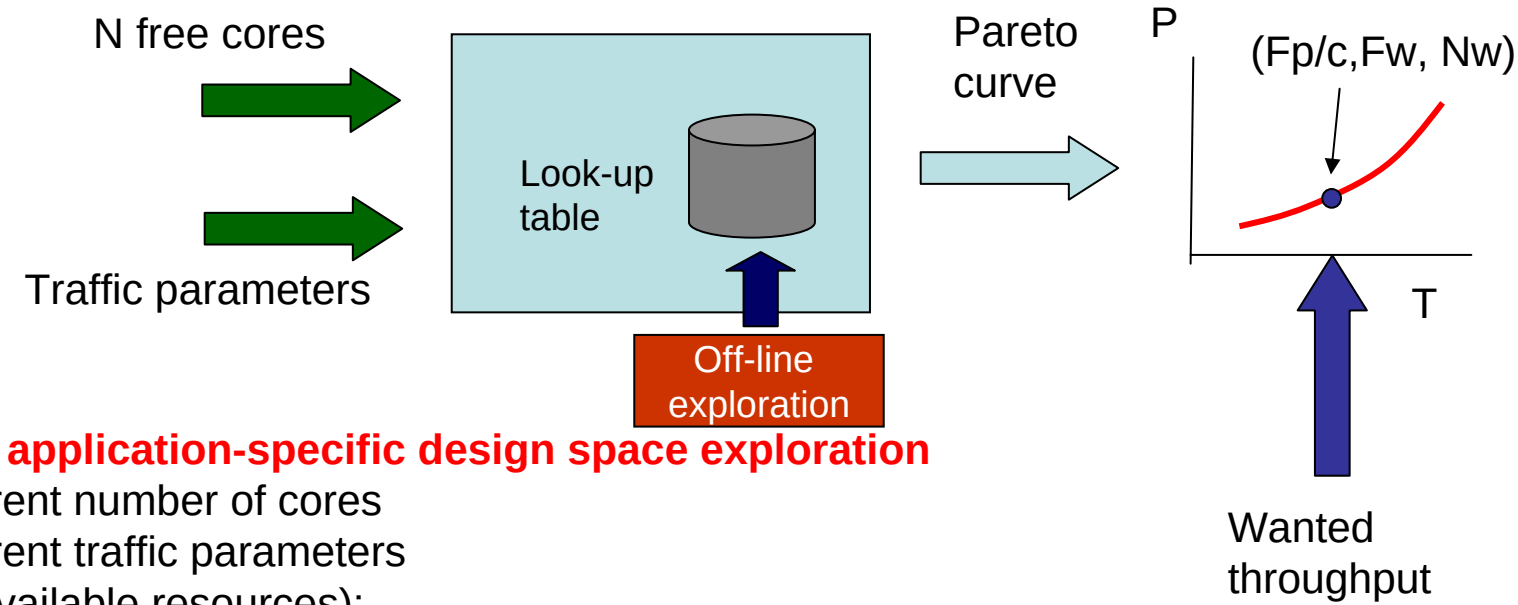




- Shared memory communication
- DES workload is self-balanced (uncorrelated input data frames)
- Fp = Fc
- Fbus = MAX

Energy Conservation in MPSoC

Problem Description



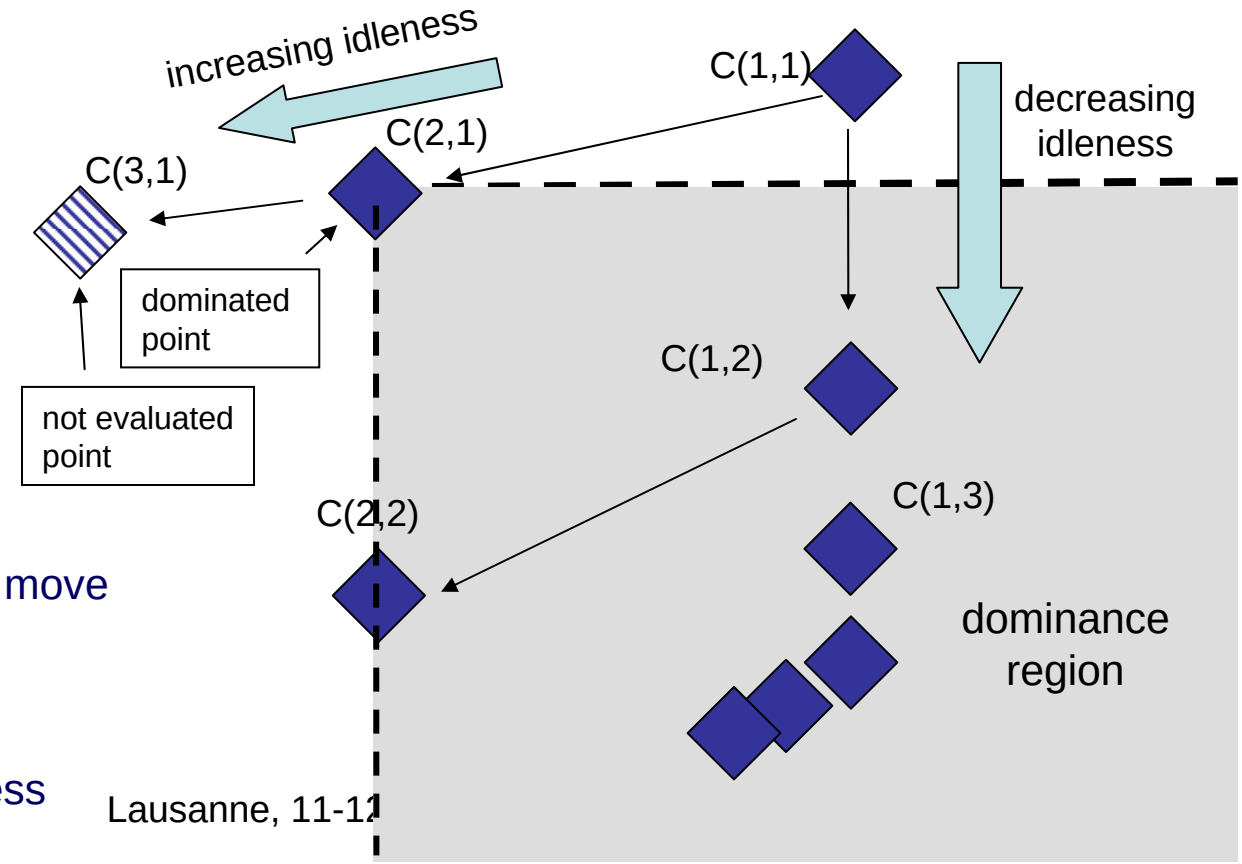
- **off-line application-specific design space exploration**
  - Different number of cores
  - Different traffic parameters
- input (available resources):
  - Number of free cores (available for DES algorithm)
  - Traffic conditions
- **on-line selection of optimal Pareto curve from a look-up table**
  - interpolation in case of missing entry
- output
  - Power/Throughput Pareto curve of system configurations

Energy Conservation in MPSoC

Off-line Design Space Exploration

Dominance condition: a configuration is said to be dominated if there is at least another configuration with higher or equal throughput and lower power

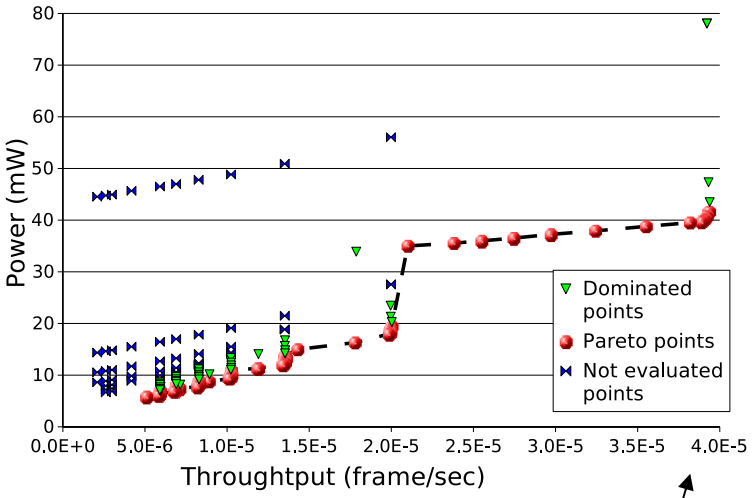
Search method: starting from Fp/c=Fmax, Fw=Fmax we find other configurations by scale down Fp/c or Fw. Other configurations are generated by further scaling down with the same rule.



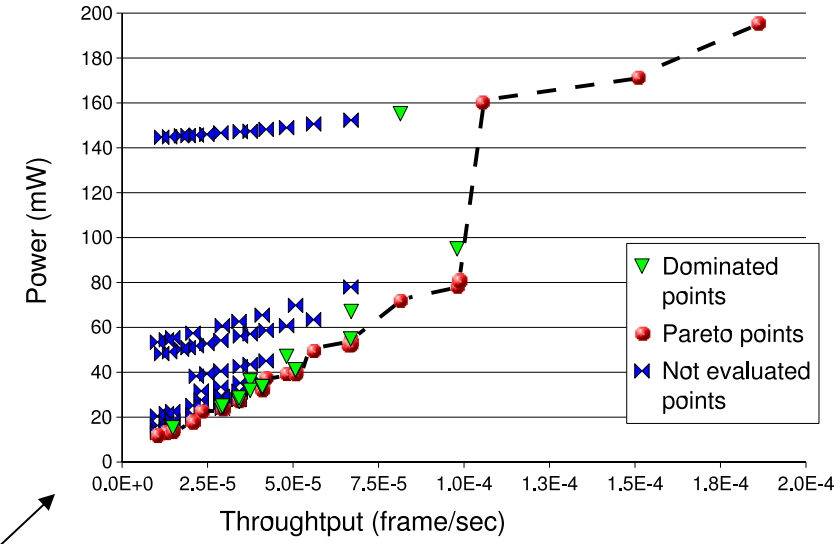
During exploration, we wish to move towards minimum Idleness configurations, so we discard configurations which lead to increasing idleness

Energy Conservation in MPSoC

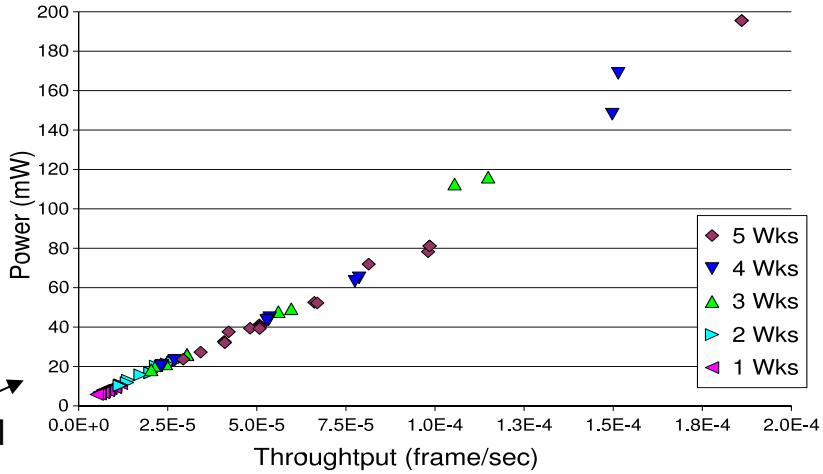
Recent Results



Pareto curve for 1 worker:
bottlenecks are workers. Scaling down Fp/c gives more efficient configurations



Pareto curve for 5 workers:
bottlenecks are either p/c or wks. To reach efficient configuration we need to scale both.

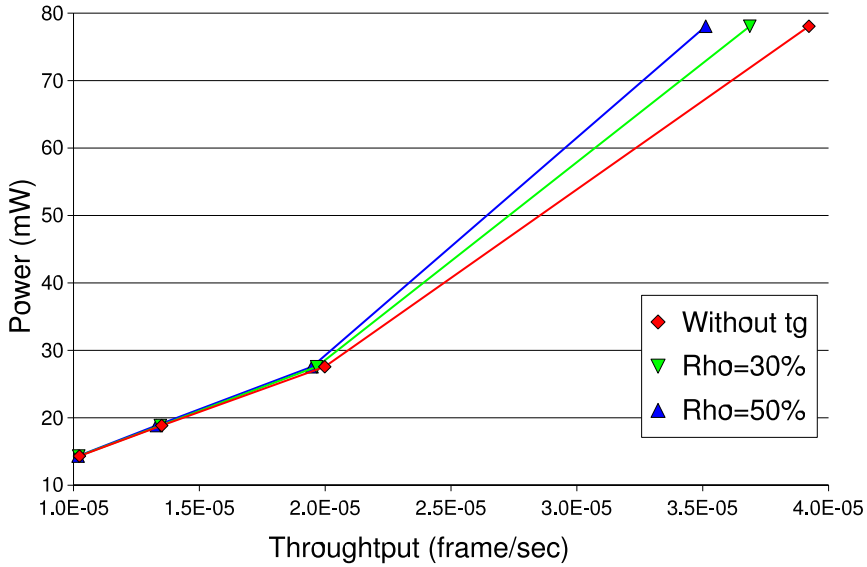


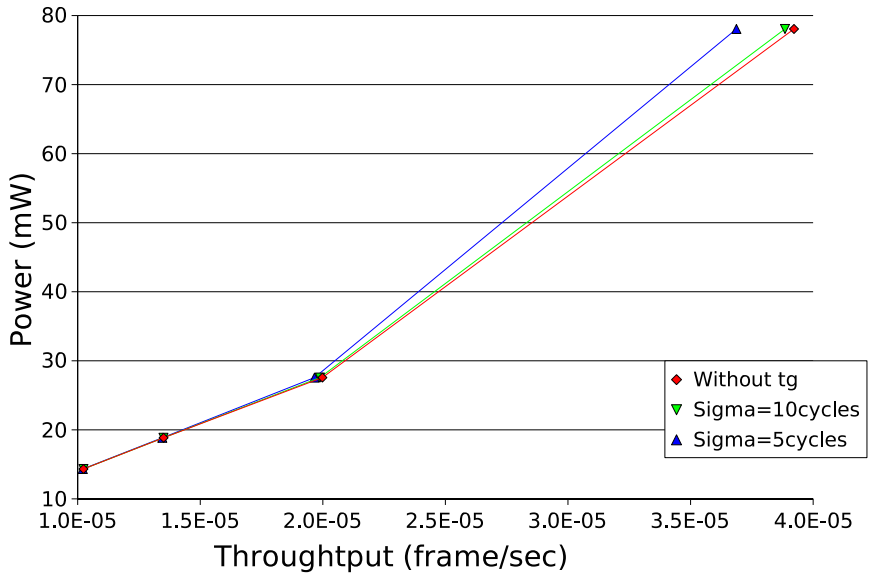
Overall Pareto curve obtained by comparing Pareto optimal curves for different number of workers

Energy Conservation in MPSoC

Recent Results

Effect of different traffic conditions: interfering traffic **bandwidth**





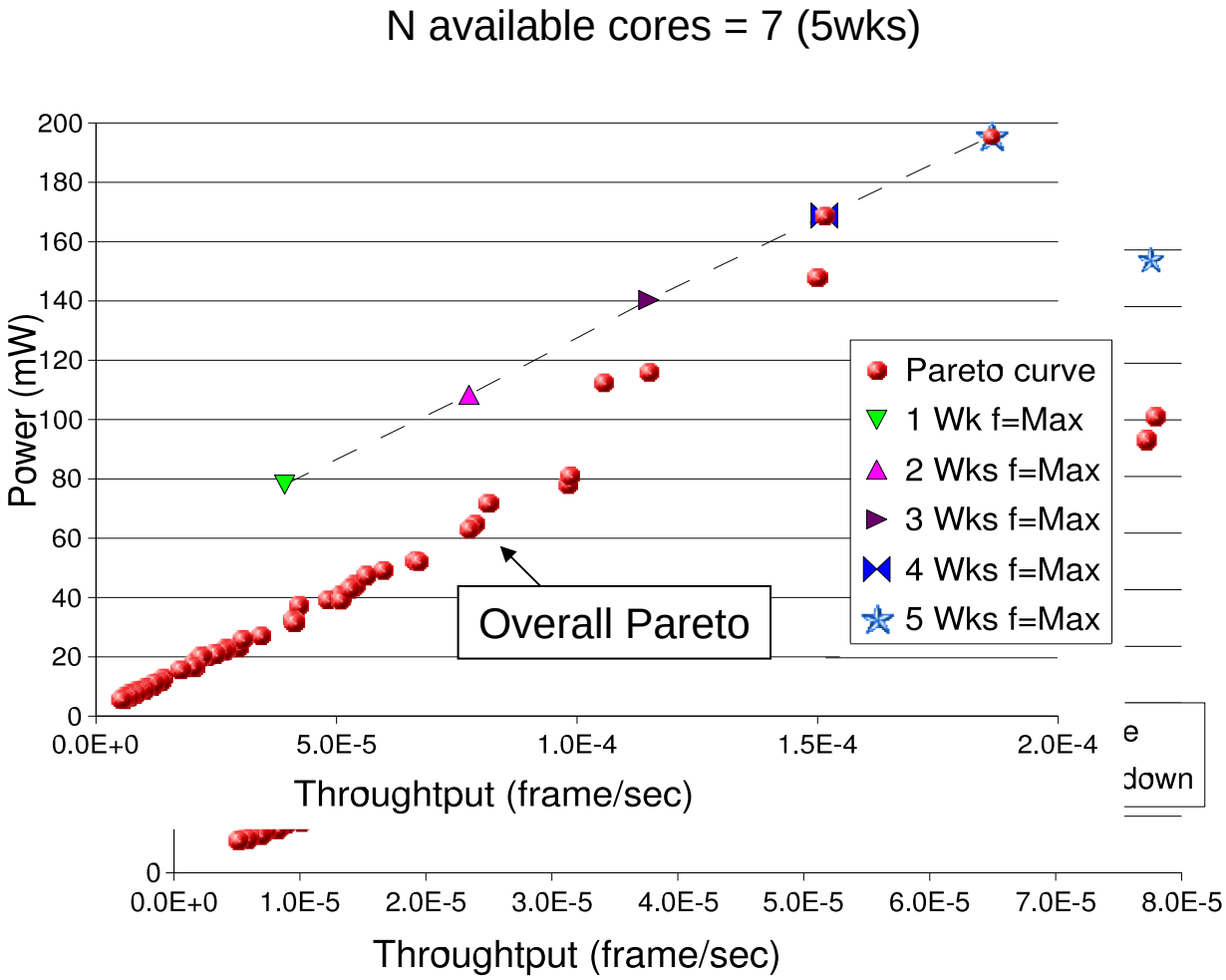
Effect of different traffic conditions: interfering traffic **granularity**

Energy Conservation in MPSoC

Recent Results

Comparison with
no frequency scaling and
tuning #wks approach:

50% savings for low
throughput

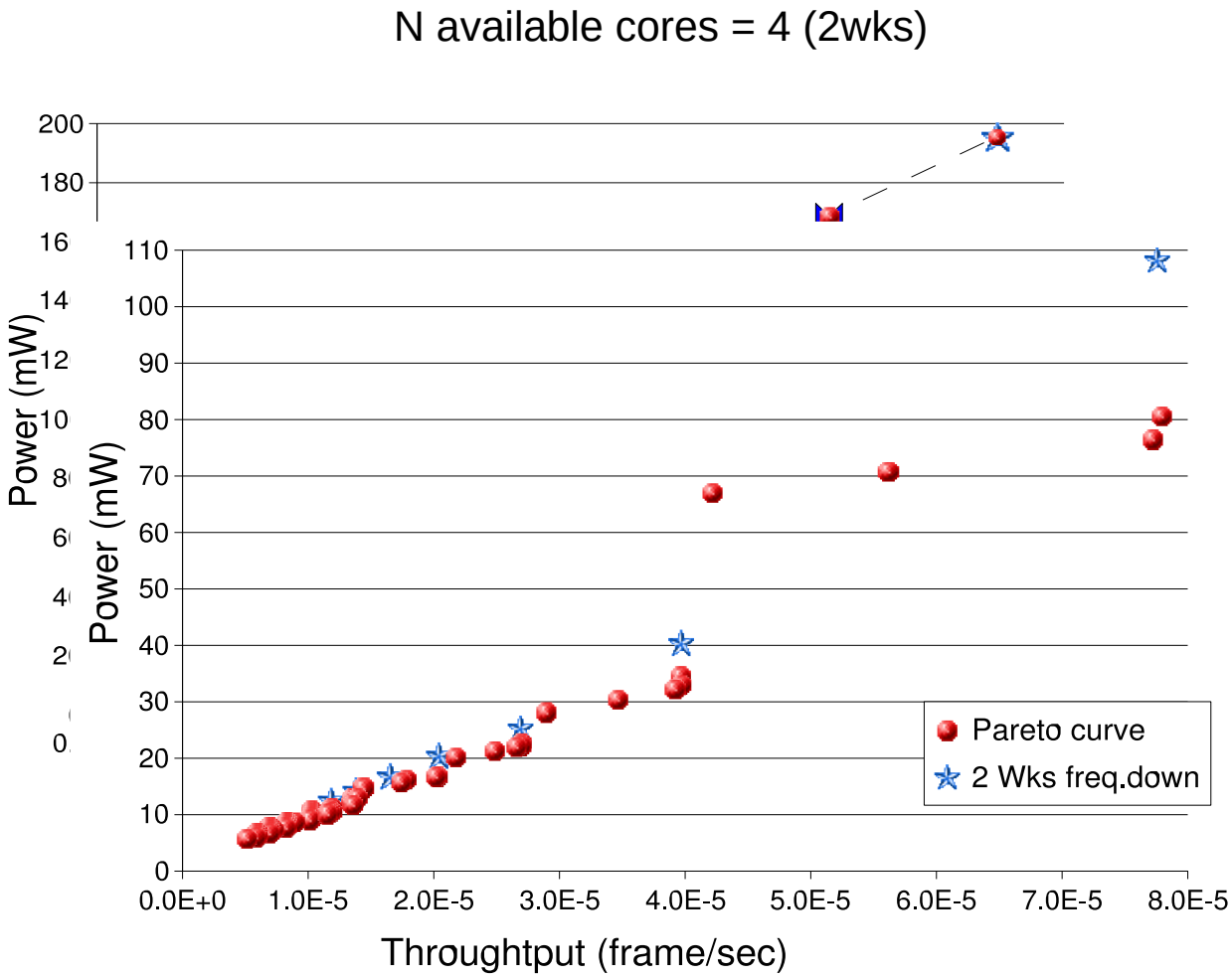


Energy Conservation in MPSoC

Recent Results

Comparison with frequency scaling and constant #wks approach:

30% savings for high throughput



Energy Conservation in MPSoC

Ongoing Work

- Evaluate look-up table size and overhead
- Application driven VS OS-driven approach?
- Handle dynamic workload conditions
  - Multimedia data-dependent workload (ex. H.263/mpeg2-4)
  - Need feed-back mechanism to determine instantaneous throughput
  - look at inter-processor message queues
  - Establish preemption points