Gestion hybride de la consommation par DPM et DVFS de MPSoC temps réel

Michel Auguin

Co-auteurs :

Muhammad Khurram Bhatti
Cécile Belleudy

Laboratoire LEAT
Université Nice Sophia-Antipolis, CNRS

ASR 17/01/2012

Presentation Plan

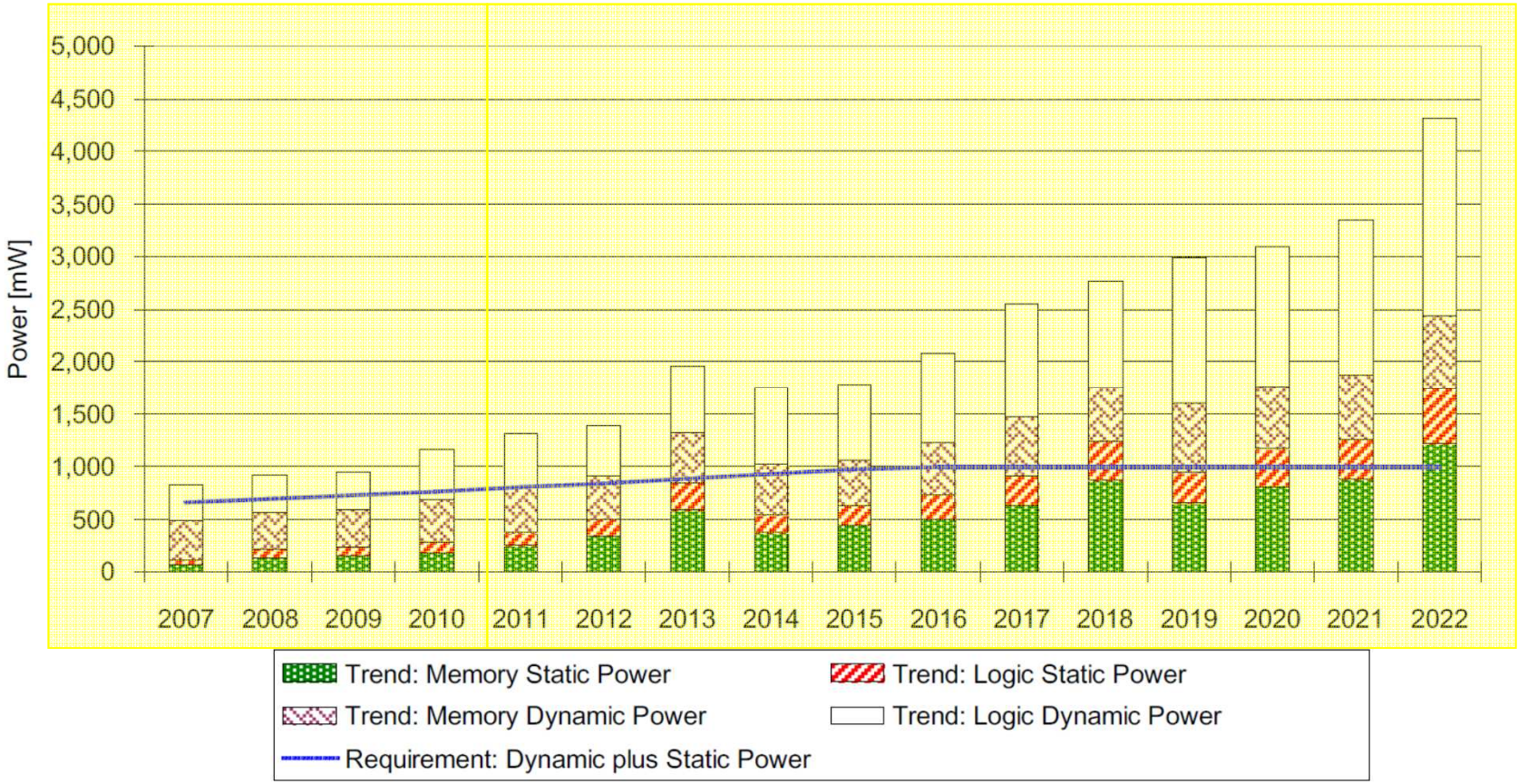
- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

Problem Statement

- An increasing number of battery-powered embedded system exploit multi-core technology, both for:
  - Performance requirements
  - Energy consideration
- Example of domains where embedded multi-core systems are considered:
  - Mobile devices, Multimedia-based equipments
  - Electronic devices for automotive, avionics…
  - i.e. systems concerned by real time behavior and energy budget constraints.

Problem Statement (cont'd)

- Trend (ITRS 2008 update)
  - Power consumption in portable equipment



Problem Statement (cont'd)

- Hardware mechanisms are available at micro-architecture level for controlling power:
  - Power switches, Vdd hopping technique
  - Level shifters, isolation cells….
- A fine control of the power architecture can be achieved, including at runtime.
- But what power control strategies could be applied, mainly at runtime, to exploit capabilities of the power-controlled HW architecture?

Problem Statement (cont'd)

- Real-time and Multiprocessor perspective
  - Temporal correctness
    - Scheduling technique
    - Schedulability analysis
- Power effective perspective
  - Reduce energy/power consumption
  - Extend battery life, reduce thermal effects
  - But impact temporal characteristics!

Problem Statement (cont'd)

Energy-efficiency and Scheduling correctness of Real-time Multiprocessor systems are closely related problems, which should be tackled together for best results

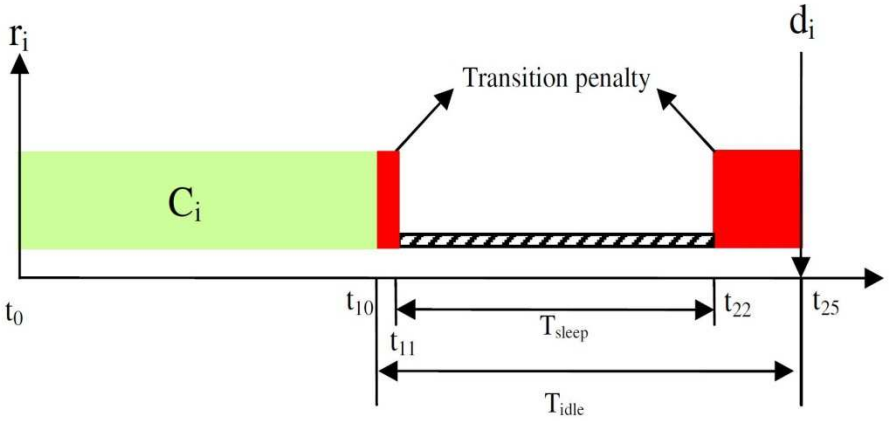
Problem Statement (cont'd)

- Working context:
  - SMP multiprocessor architecture
    - Identical processors
  - Task model:
    - Periodic, independent tasks
  - Global scheduling
    - Full task migration scheduling
  - Simulation tool for experiments
    - STORM v3.2 (developed at IRCCyN)
      - http://storm.rts-software.org

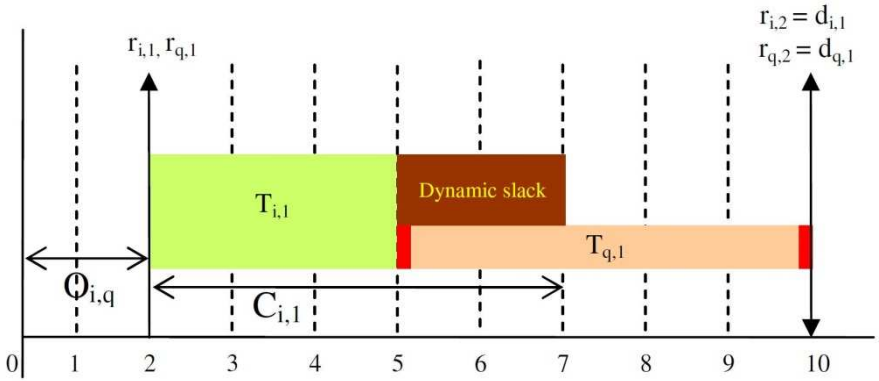
Problem Statement (cont'd)

- Basic principles of DPM & DVFS

Dynamic Power Management

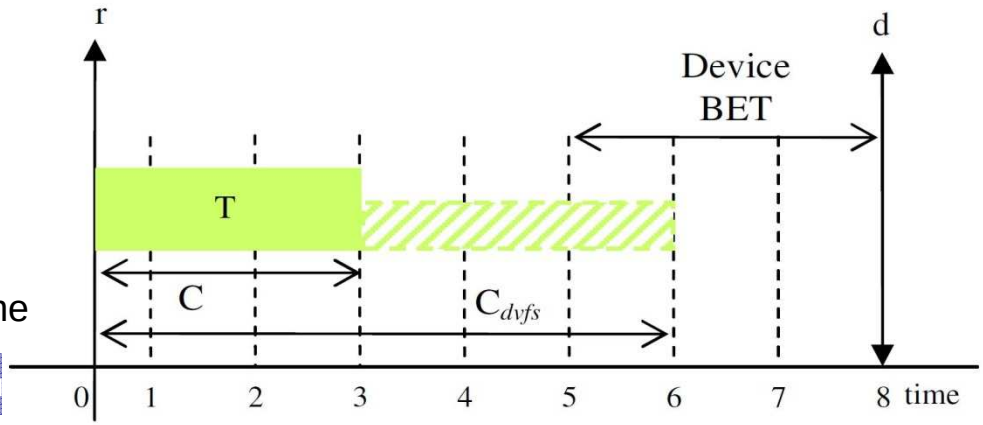


Dynamic Voltage and Frequency Scaling



What technique should be used:
DPM ? DVFS ? Both ?

Interplay of DPM and DVFS



BET : Break Even Time

Presentation Plan

- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

Presentation Plan

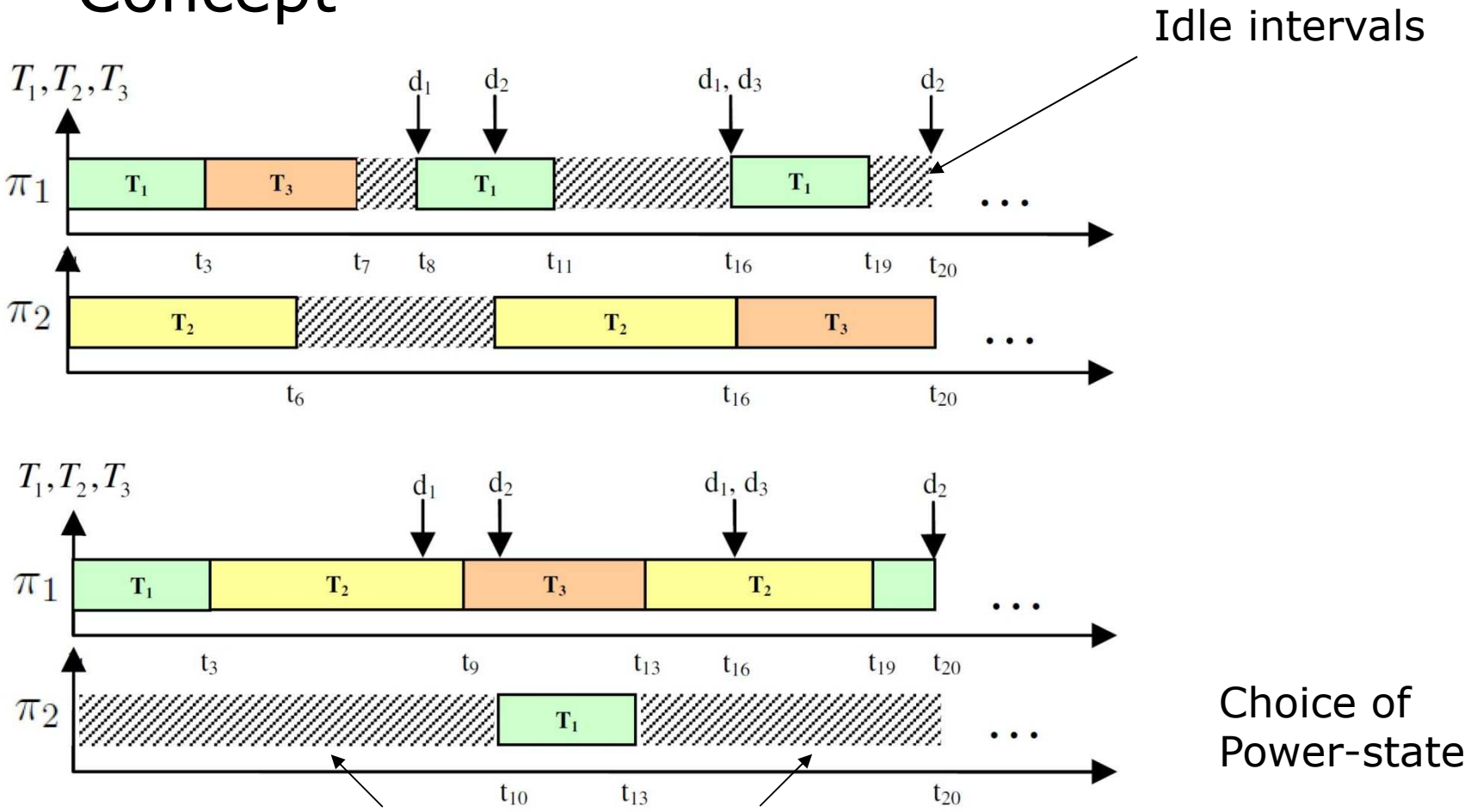
- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

Assertive DPM

- DPM technique for only Multiprocessor systems
- Rationale
  - Non-predictive execution time of tasks
    - Avoid speculation for balancing efficiency vs safety
  - Aggressive extraction of idle intervals
    - Cumulated idle time suites state-transitions
  - AsDPM : Admission control for ready tasks
    - Need-based resource utilization

Assertive DPM (cont'd)

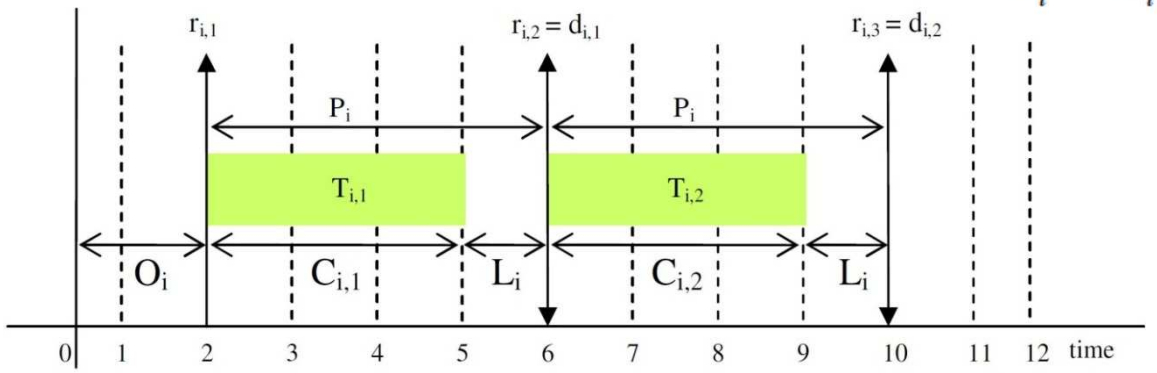
- Concept



Assertive DPM (cont'd)

- Working principle
  - Laxity (L_i): Measure of urgency to execute (w.r.t. deadline) on a particular proc.

$$L_i = d_i - \left(t_c + C_i^{rem}\right)$$

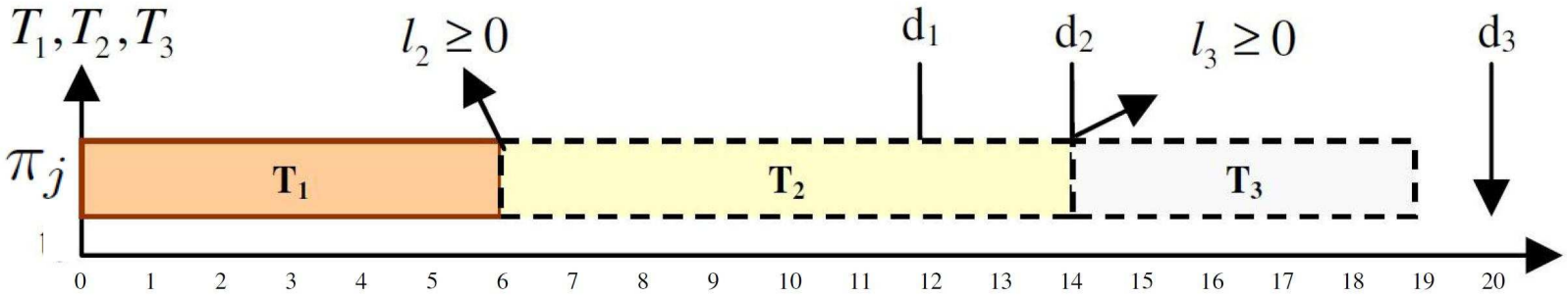


  - **Anticipative laxity** (l_i):
    - Laxity in the presence of higher-priority released tasks on particular processor

$$l_i^c = d_i - \left(t_c + C_c^{rem} + C_i^{rem} + \sum_{\tau_k \in \tau^d, \tau_k[\pi_j]} C_k^{rem} \right)$$

Assertive DPM (cont'd)

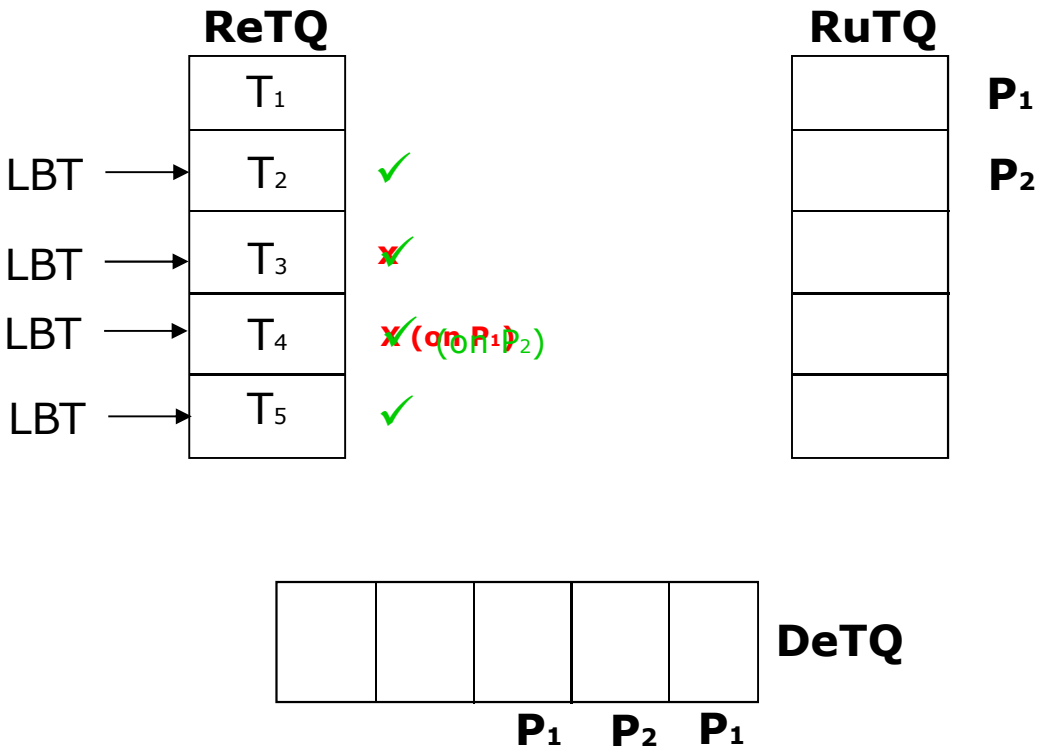
- Working principle
  - Anticipative laxity (l_i):

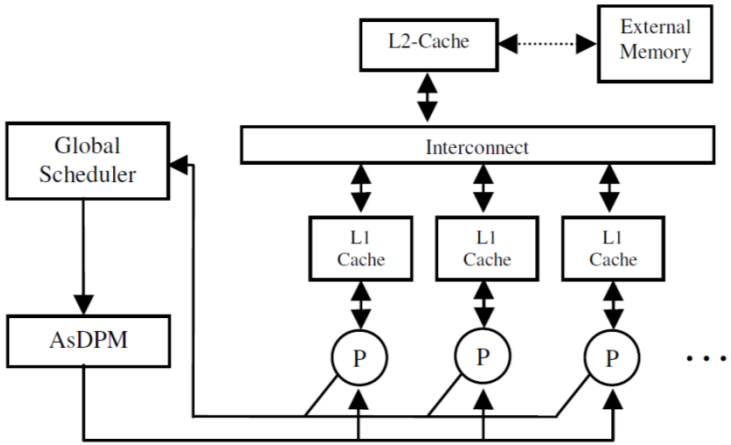


  - Laxity Bottom Test (LBT)
    - While l_i>0 starting execution of T_i could be delayed
    - Need to be reevaluated each time a task is released

Assertive DPM (cont'd)

- Working principle
  - Laxity Bottom Test (LBT)





– At most 2 processors required until next sched. event!

– LBT on every sched. event

ReTQ: Ready Task Queue
RuTQ: Running Task Queue
DeTQ: Deferred Task Queue

Assertive DPM - Results

- Example Configuration
  - Task Set:
    - Six tasks (n=6)
    - Hyper-period = 1200ms
  - Architecture:
    - Multiprocessor SMP architecture (PXA270)
  - Global Scheduling Policies:
    - Earliest Deadline First (EDF)
    - Least Laxity First (LLF)

Task characteristics:

Task	r_i	C_i	d_i	T_i
T_1	0	6	16	16
T_2	0	8	20	20
T_3	10	8	24	24
T_4	10	8	30	30
T_5	16	16	40	40
T_6	20	20	50	50

$$u_{sum}(\tau) \overset{def}{=} \sum_{\tau_i \in \tau} u_i = 2.20$$

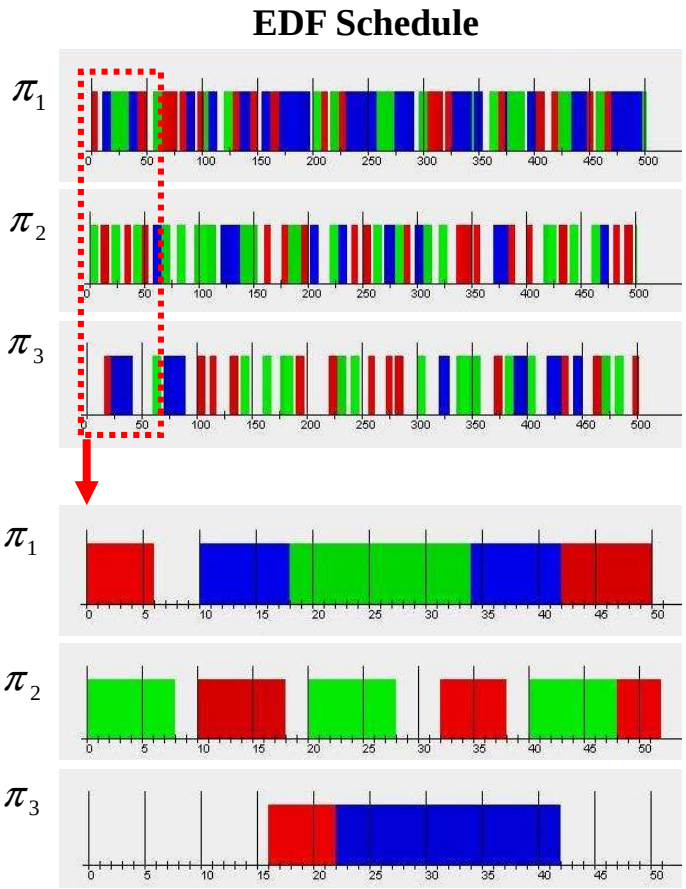
PXA270 energy states*

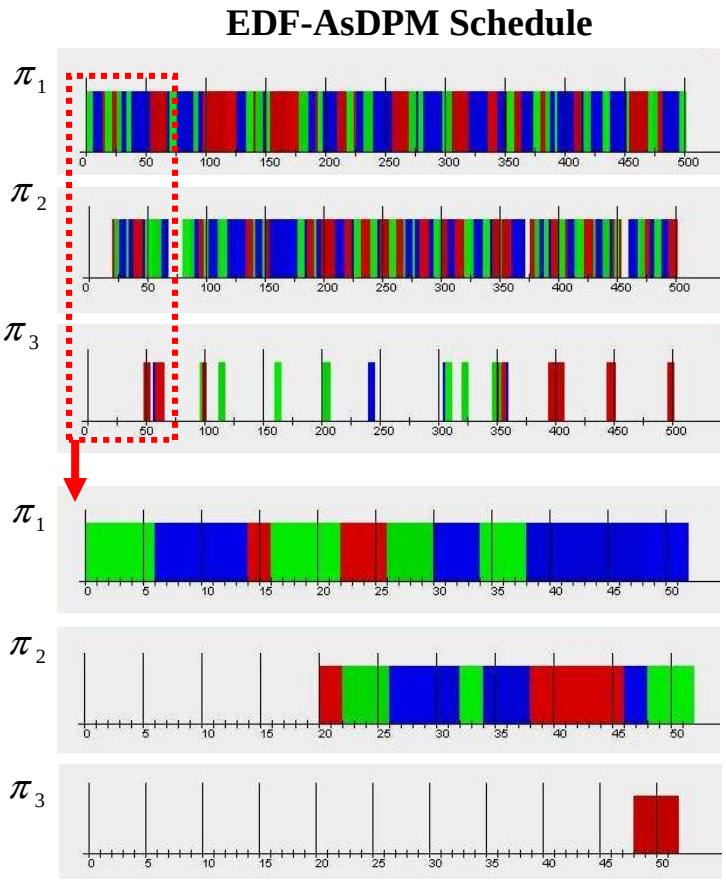
State	α (mw)	ψ (ms)
S_1 (Running)	925	
S_2 (Idle)	260	0.001
S_3 (Standby)	1.70	11.28
S_4 (Sleep)	0.16	136

* Values at F_{ref}=624 MHz and V_{dd}=1.55Volts

Assertive DPM – Results(cont'd)

- STORM's schedule: EDF algorithm





- @ WCET of tasks
- @ Max. frequency

Assertive DPM (cont'd)

- Simulation results
  - EDF Scheduling
    - Energy consumption:
      - 10.40% with WCET
      - 15.86% with AET
    - State transitions
      - 74.85% with WCET
      - 74.53% with AET

Performance with WCET of tasks

Performance	EDF	EDF-AsDPM
Average power(W)	2.224	1.993
Total Energy(J)	2.671	2.393
No. of state-transitions	167	42

Performance with AET of tasks

Performance	EDF	EDF-AsDPM
Average power(W)	1.906	1.605
Total Energy(J)	2.289	1.926
No. of state-transitions	216	55

Occupancy of processors

Processor occupancy (%)	EDF	EDF-AsDPM
P_1	86	99
P_2	73	94
P_3	56	20

Assertive DPM – Results(cont'd)

- Simulation results
  - Least Laxity First (LLF)
    - Energy consumption:
      - 9.70% with WCET
      - 17.58% with AET
    - State transitions
      - 59.76% with WCET
      - 66.35% with AET

Performance with WCET of tasks

Performance	LLF	LLF-AsDPM
Average power(W)	2.224	2.008
Total Energy(J)	2.671	2.412
No. of state-transitions	169	68

Performance with AET of tasks

Performance	LLF	LLF-AsDPM
Average power(W)	1.917	1.580
Total Energy(J)	2.303	1.898
No. of state-transitions	211	71

Occupancy of processors

Processor occupancy (%)	LLF	LLF-AsDPM
P_1	83	99
P_2	75	90
P_3	58	26

Assertive DPM (cont'd)

- Remarks

  - Admission control principle allows need-based resource utilization
  - Offline optimizations are possible
  - Transition to deeper power-states is possible

Presentation Plan

- Problem statement
- Contributions
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

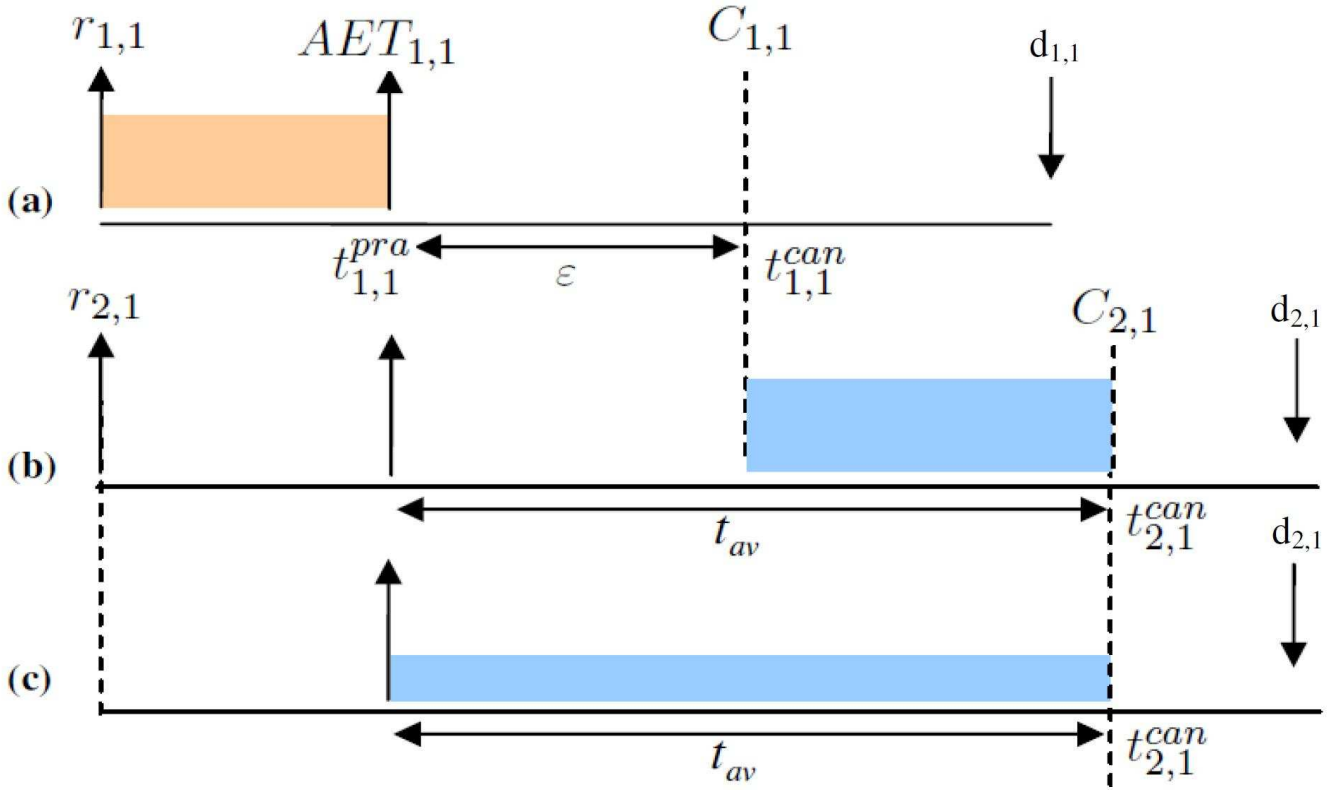
DSF DVFS

- Objective: Real time scheduling And energy optimization for multiprocessor

- Underlying principle:
  - Adjust frequency (power) so as to mimic a canonical schedule of tasks (real time behavior)
  - Canonical schedule: WCET of tasks, max frequency of processors and a global real time scheduling algorithm
    - *Canonical schedulability analysis* should holds in *practical* schedule
      - H. Aydin et al. (2004), V. Nélis et al. (2009)

DSF DVFS (cont'd)

- Concept (Dynamic Slack Reclamation)

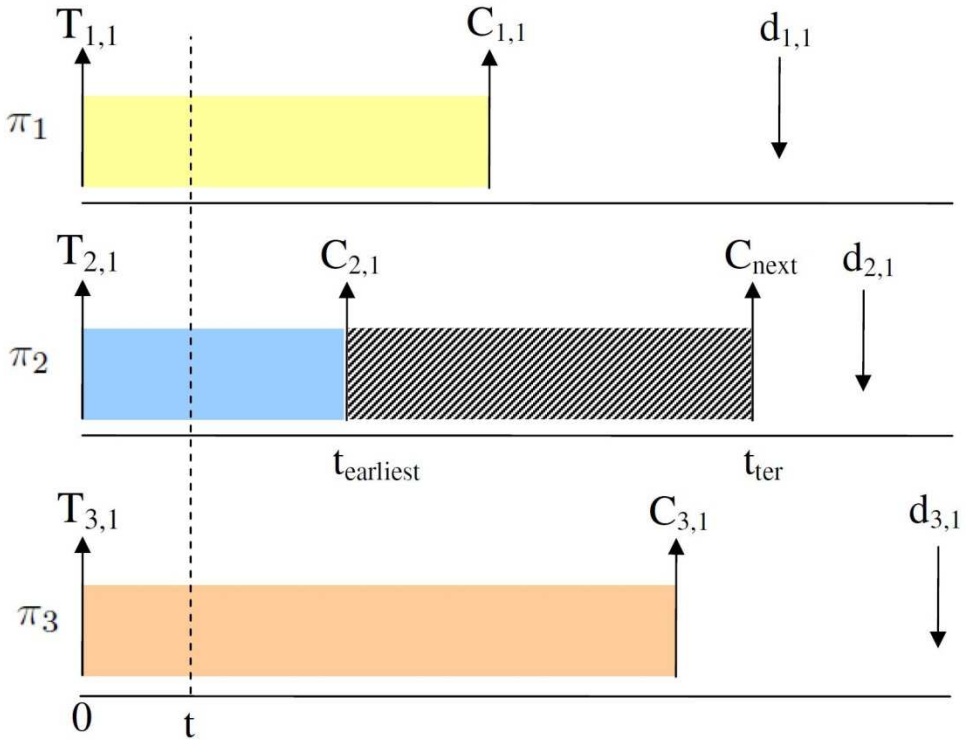


Dynamic slack is fully consumed by subsequent (single) task

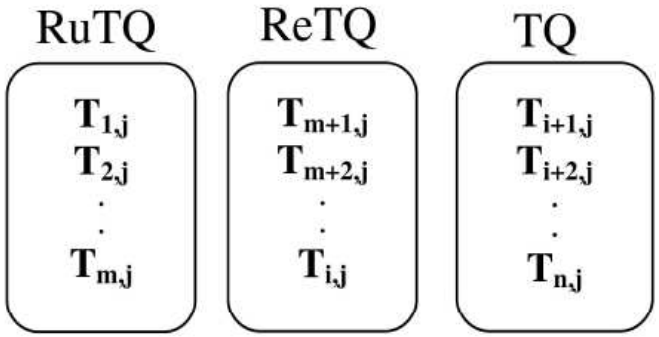
Slowdown factor does not allow a dispatched task to execute beyond its worst-case boundary defined in canonical schedule

DSF DVFS (cont'd)

- Concept (DSR algorithm)



Calculation of canonical schedule at run time



$$T_{next} = max\_priority\left[\left(T_{m+1,j}\right), \left(T_{i+1,j}\right)\right]$$

$$t_{earliest} = t + min_{i=1}^{m}\left[C_{i,j}\right]$$

$$t_{ter} = t_{earliest} + C_{next}$$

DSF DVFS (cont'd)

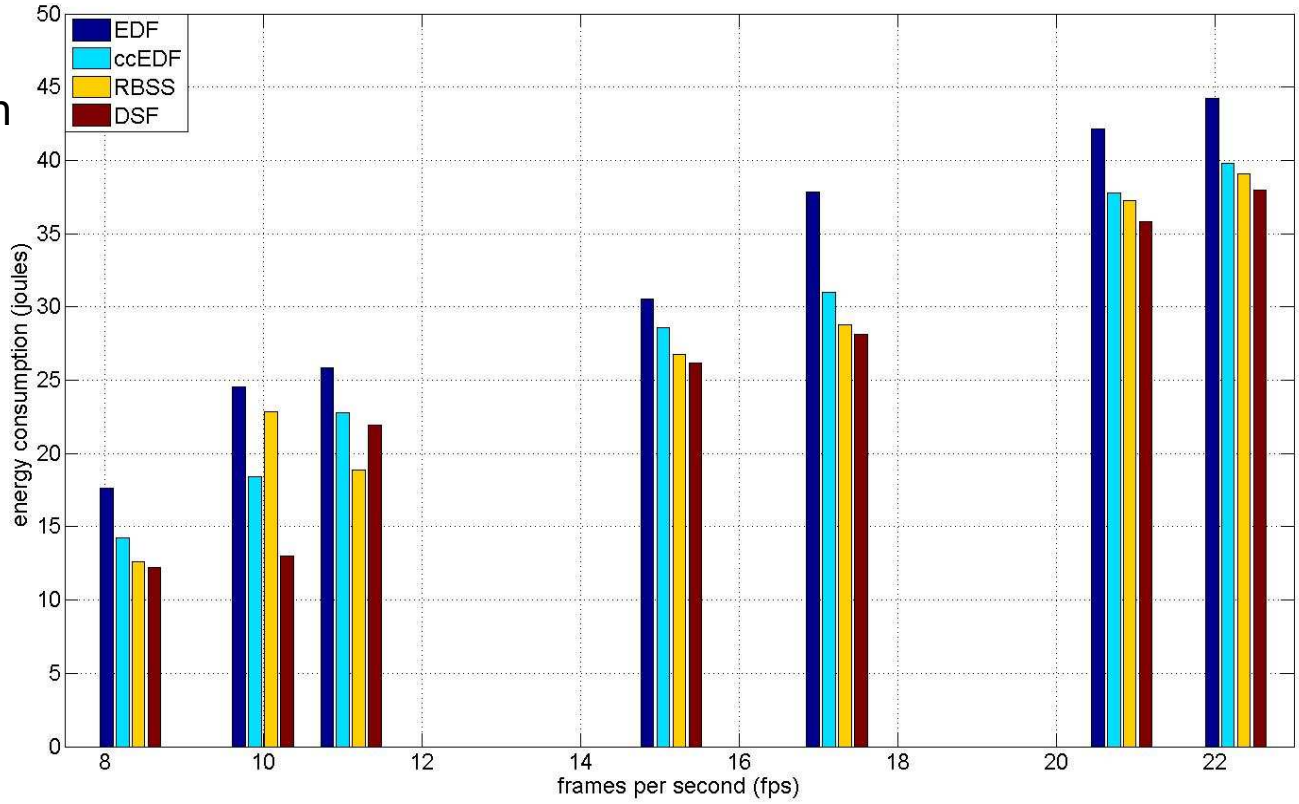
- Improving DSR:
  - OSM (Online Speculative Mechanism) for soft real-time tasks only
    - Tasks do not finish often with WCET
    - Speculates on probable execution time and adjust frequency on release time
  - *m*-TE (m-Task Extension) technique
    - Further slowdown when only *m*-tasks are left in ready tasks queue

DSF DVFS (cont'd)

- Results

H.264 video decoder application [Thales Com.]
- Slice parallelization

Best-case:
Up to 47% w.r.t. EDF
Up to 29% w.r.t. ccEDF
Up to 43% w.r.t. RBSS



MOTE 44% w.r.t EDF(k)

MORAOTE 32% w.r.t EDF

[Nélis08, Nélis09]

*RBSS: *Resource Based Slack Sharing Algorithm [CEA/LIST]*

DSF DVFS (cont'd)

- Remarks

  - Any correct canonical schedule could be used with DSF, provided it can be computed at runtime.
  - Canonical schedulability analysis can hold in Practical schedule with less computational overhead
  - OSM and m-TE can further ameliorate energy savings

Presentation Plan

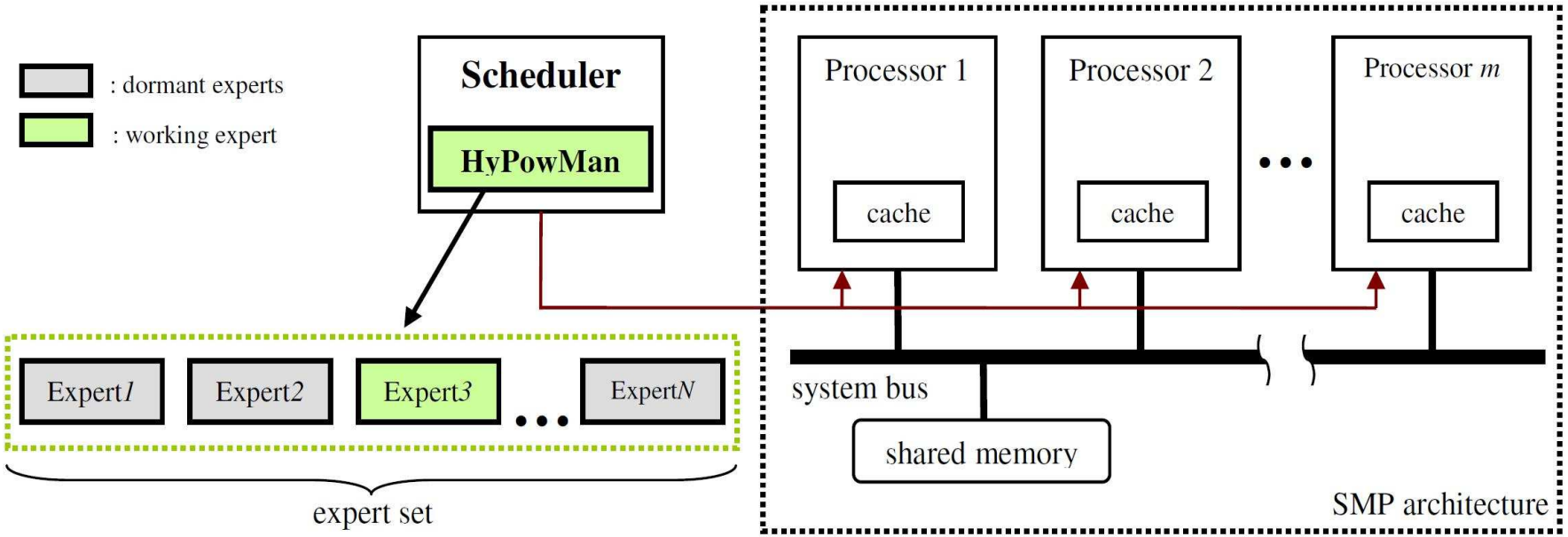
- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

HyPowMan

- Online and adaptive interplay of DPM and DVFS using machine-learning technique
- Rationale
  - Both policies can outperform each other
    - Target application and objective
    - Scheduling algorithm
    - Architecture configuration
  - No single policy fits perfectly most operating conditions
  - Generic scheme that can use existing policies

HyPowMan (cont'd)

- Concept
  - Converge towards the best-performing expert *within expert set only*



HyPowMan (cont'd)

- Concept
  - Probability vector

$$H^{input} = (h_1^{input}, h_2^{input}, ..., h_N^{input})$$

N = number of experts

$$H^{input} = W^{input} / \sum_{k=1}^{N} w_k^{input}$$

Values are normalized

  - Weight vector

$$W^{input} = (w_1^{input}, w_2^{input}, ..., w_N^{input})$$

$$\sum_{i=1}^{N} w_i^1 = 1$$

$$l_k^{input} = \alpha l_{ke}^{input} + (1-\alpha) l_{kp}^{input} \qquad (0 \le \alpha \le 1)$$

Loss factor: consists of loss in energy and loss in performance

$$w_k^{input+1} = w_k^{input} \beta^{l_k^{input}} \qquad \beta \in [0,1]$$

Update weight values when idle time or slack time occur

HyPowMan (cont'd)

- Concept
  - No intervention in decision-making process
  - Selection of experts
    - Statically selected single expert
    - Static profiling-based experts
    - Online selected experts
  - Computation time complexity
    - Scheduling event: Bounded by the time complexity of working expert
    - Active time: Sum of computation time of all experts

HyPowMan (cont'd)

- Results: bcet/wcet ratio

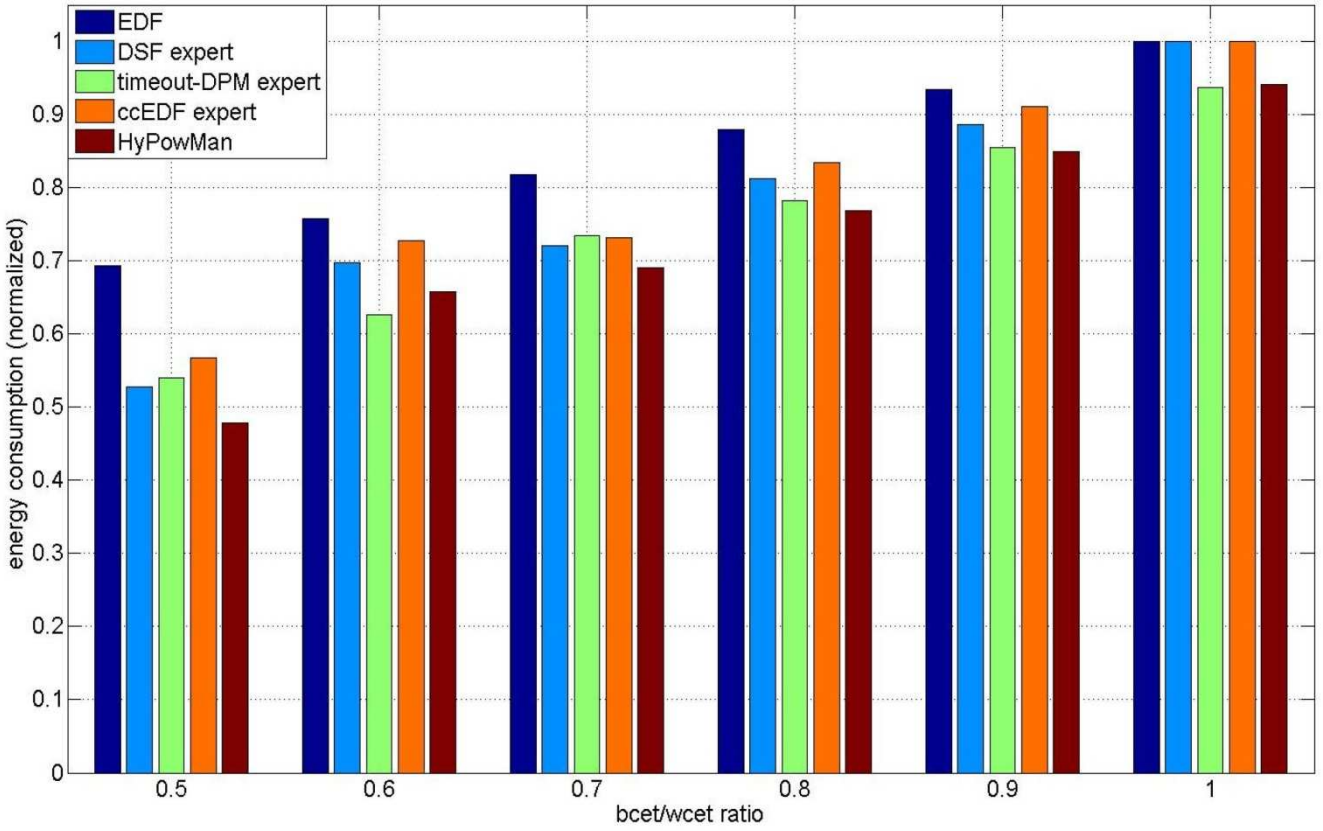
Synthetic task set
Application
(n=8, m=4)

Energy savings:
23.12% for DSF

18% for ccEDF

47.94% for timeout-DPM

47.22% for HyPowMan



HyPowMan (cont'd)

- Results: number of tasks

Synthetic task set
Application
(n=8-24, m=4)

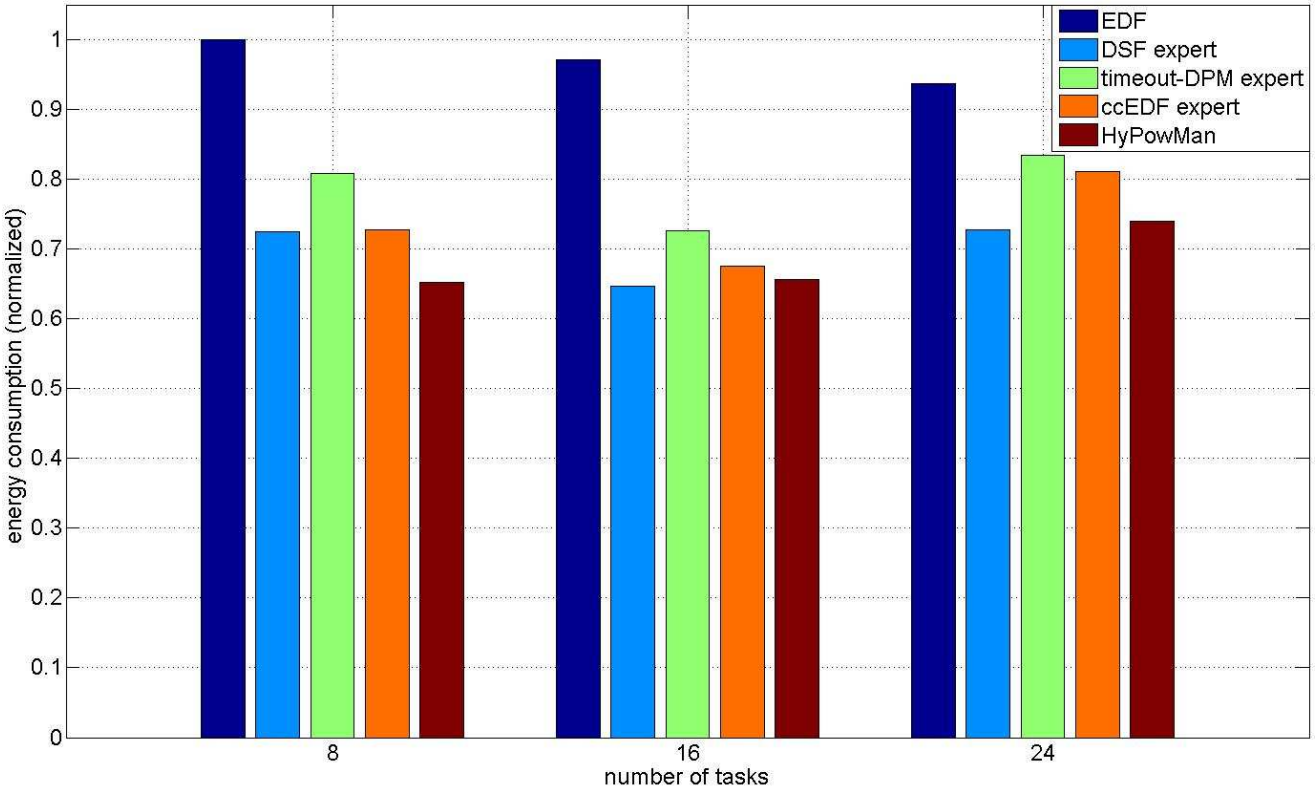
Energy savings:

18% for DSF

30.4% for ccEDF

10.2% for timeout-DPM

24.7% for HyPowMan

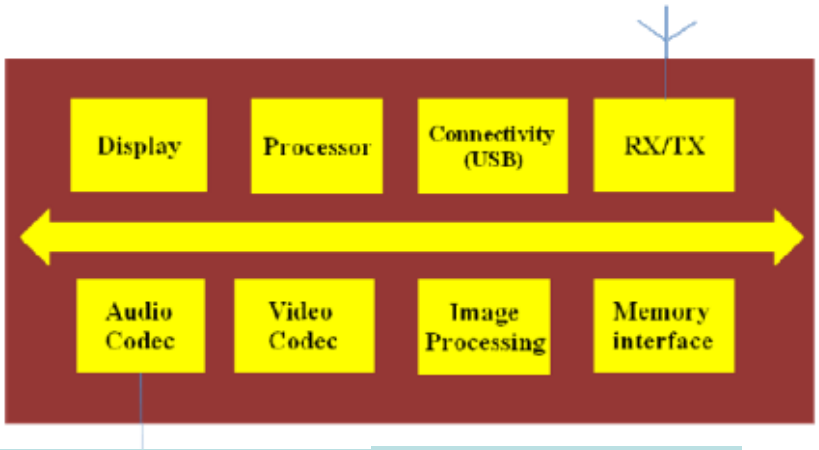


Presentation Plan

- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

- Design is decomposed in Power Domains:
- Complex power domains with nested and hierarchical power domains.

Retention registers
Isolation cells
Vdd hopping
Clock/reset domains

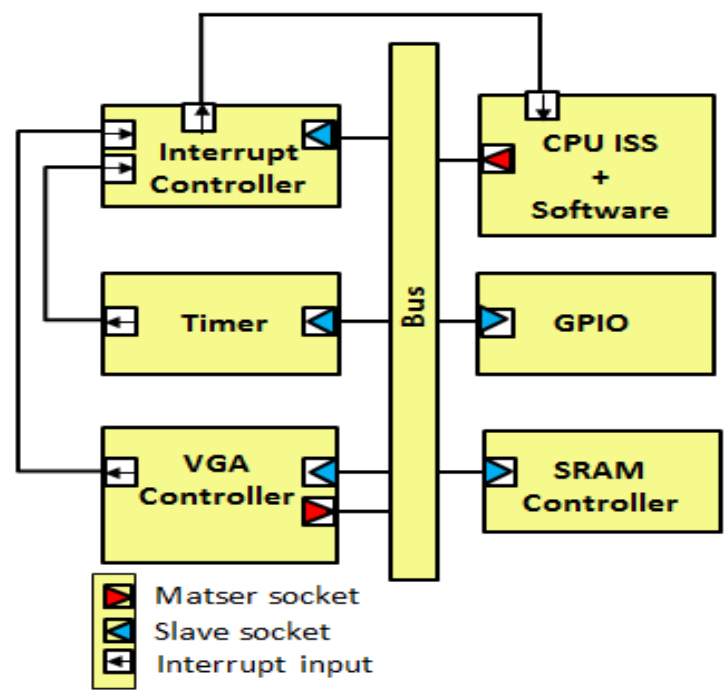


Full mode,
Text mode
Phone mode
Camera mode

Ply back mode,
Game mode
Off mode

	On	Txt	Phone	PIM_access	Camera	Play_back	Game	Standby	Off
PD_proc	1.2V	1.0V	1.0V	1.2V	1.0V	1.0V	1.2V	1.2V	OFF
PD_TX	1.2V	1.2V	1.2V	0.8V	OFF	OFF	OFF	1.0V	OFF
PD_Disp	1.2V	1.2V	0.8V	1.2V	1.2V	1.2V	1.2V	OFF	OFF
PD_Img	1.2V	1.0V	0.8V	OFF	1.2V	1.2V	OFF	OFF	OFF
PD_Rest	1.2V	1.0V	0.8V	OFF	0.8V	1.2V	1.2V	OFF	OFF

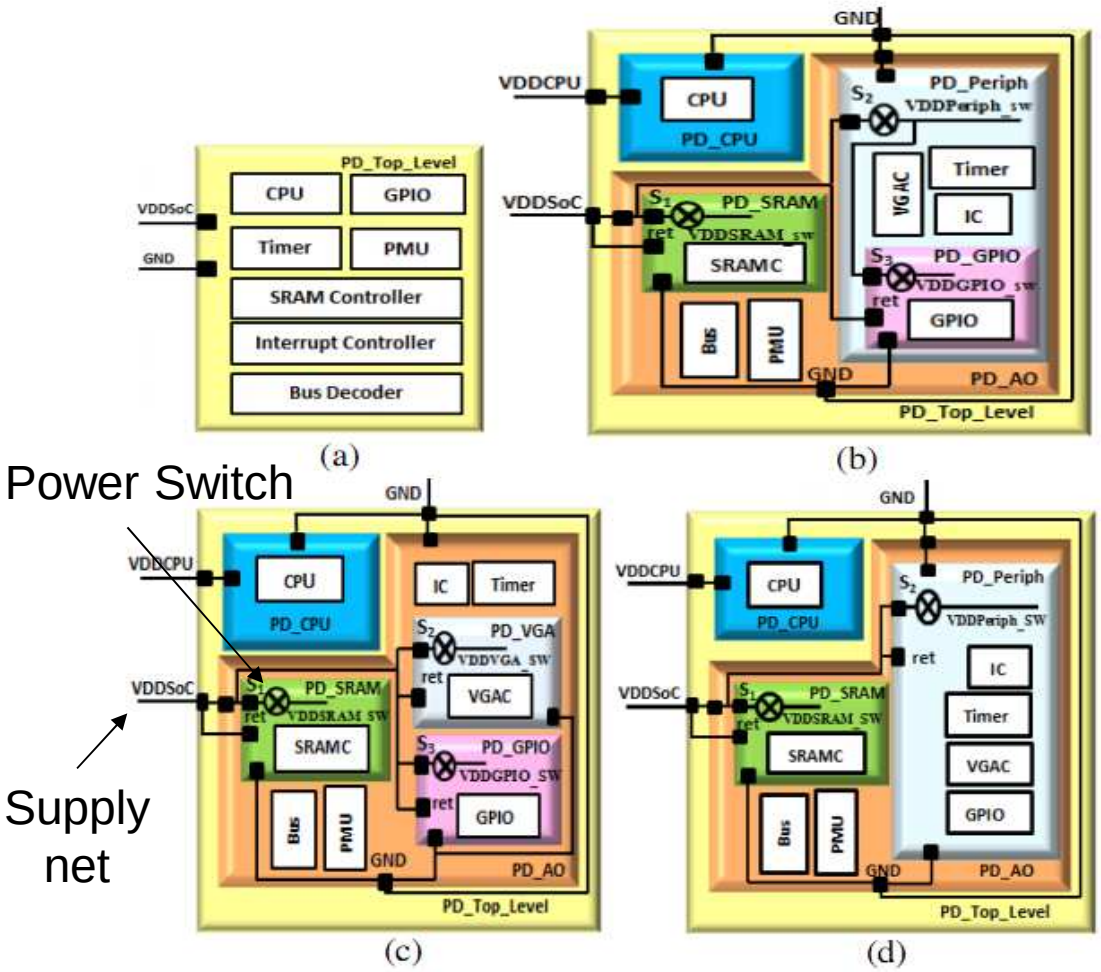
http://low-powerdesign.com/article_jasper_032310.htm



- Processing example

- The CPU computes an image
- The CPU initializes peripherals
- The VGA displays an image
- The CPU calculates a new image
- A button mapped to a GPIO is periodically checked

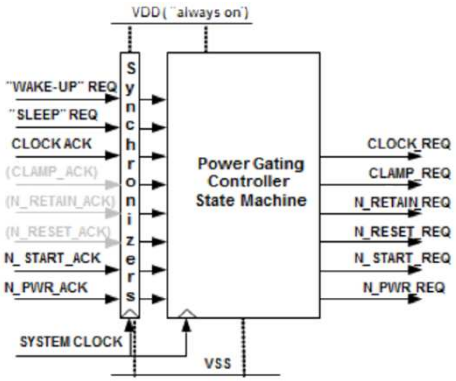
Periodic

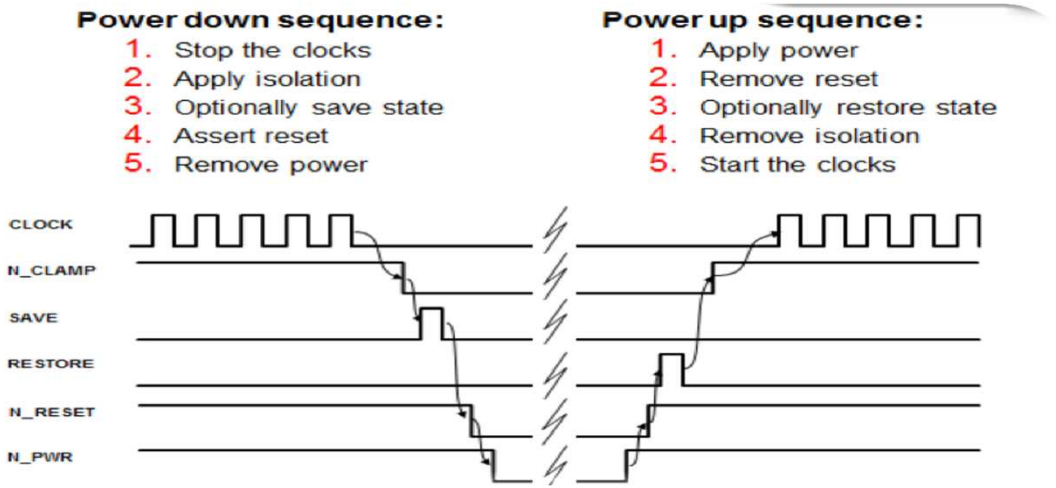


- Example of Power Domain partitioning :
  - These power architectures support the processing sequence

- Either PD partitioning results from processing modes,
- or application power strategy has to take PD partitioning into account

- But changing power states needs to respect precise timing sequence such that Hw is stable and coherent with processing requirements





Scheduling, schedulability have to consider these power state transitions?

Presentation Plan

- Problem statement
- Real Time Low Power Techniques:
  - The AsDPM technique
  - The DSF (DVFS) technique
  - The HyPowMan scheme
- Power-oriented hardware consideration
- Conclusion

Conclusions

- Contributions focus on scheduling-based solutions
  - Scheduling-based techniques
    - Assertive DPM technique
    - DSF DVFS technique
    - HyPowMan scheme
- Proposed solutions
  - Generic for a class of scheduling algorithms
  - Trade-off between complexity and efficiency
  - Low power techniques attempt to preserve real time characteristics

Conclusions (cont'd) - Perspectives

- API and implementation consideration
  - How V or F controls decided by scheduling algorithm for a core/task could be used for power architecture design? Are they compatible with a power architecture?
- Generalization of system model
  - Task models
  - Platform models
  - Scheduling algorithms
  - Formalization of proofs for *hard real-time constraints*
- Thermal aspects