



POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA,
INFORMAZIONE E BIOINGEGNERIA

Energy and power management in the computing continuum

Prof. William Fornaciari

Power Saving Knobs

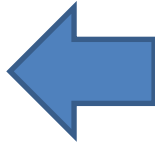
AA 2023 – 2024

William Fornaciari

william.fornaciari@polimi.it

Power Management at OS level

Outline

- Basics on power consumption
- **Power saving blocks and knobs** 
- Power states and ACPI
- Operating systems integration (the Linux case)
- Thermal issues
- Miscellanea
- Presentation of the assignments
- Conclusions

Basics of Power Management- SUMMARY

Power consumption components

$$P = \boxed{0.5 V_{DD}^2 f_{clock} C_L E_{sw}} + \boxed{t_{sc} V_{DD} I_{peak} f_{0 \rightarrow 1}} + \boxed{V_{DD} I_l}$$

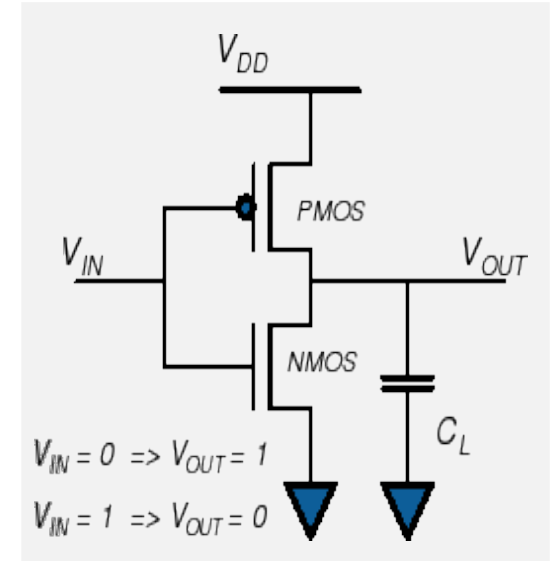
Dynamic power

- E_{sw} is the switching activity
- Data-dependent

Short-circuit power (Internal, dynamic)

Leakage power (static)

- 250nm technology scenario: marginal w.r.t. switching power
- Beyond 90nm: 35-50% of (logic) power budget



Power saving knobs

Dynamic power consumption

- The system is in **active** state
- Number of bits and Capacitance are fixed parameters
 - We can simplify the previous relationship

$$P_{dyn} \propto f V^2$$

- *Clock frequency* and *Voltage* instead can be dynamically controlled via **Phase-locked loop (PLL)** circuits and **voltage regulators**
- Alternatively, we could pause the activity of the circuit with techniques like **idle injection**
 - Computing system processors executing a sequence of NOP instructions

Power saving knobs

Static power consumption

- The system is in **idle** state (not processing)

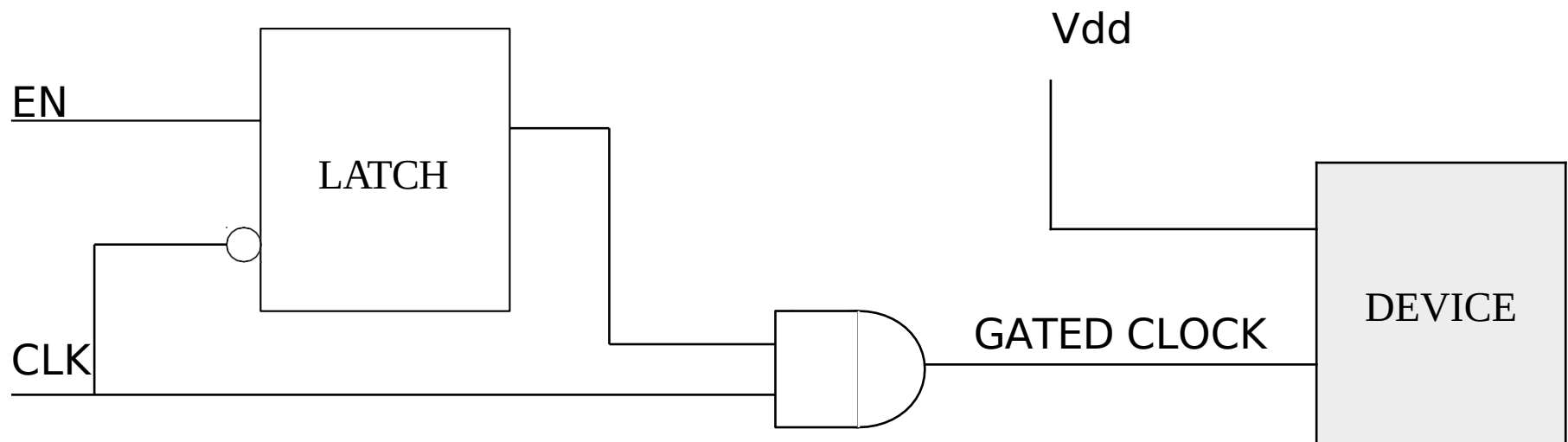
$$P_{sta} = I_{leak} V$$

- We can decrease the power consumption by reducing the *voltage* down to a minimal value needed to retain the status of the system
- We can break the *current* flow in specific system components / devices by powering them off or via **clock (or power) gating**

Power saving knobs

Clock gating

- The propagation of the clock signal towards the device can be controlled (enabled/disabled)
- While a device is clock-gated, it can still be powered on
 - No activity, but possibility of retention of the context (registers, memory content)

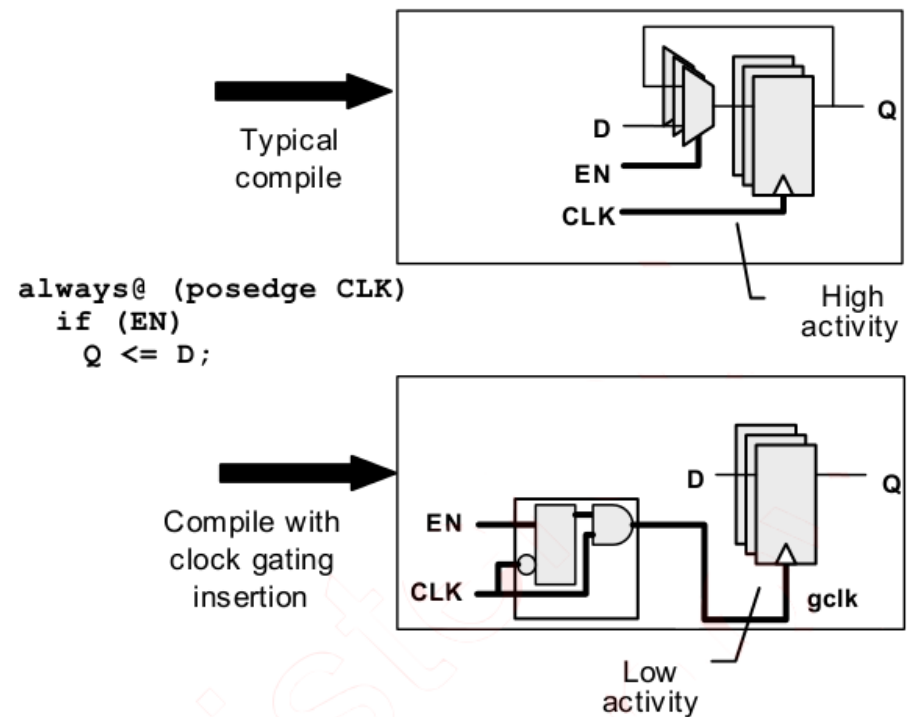


Clock gating

Up to 50% of the dynamic power is spent for clock distribution. Turn the clock off, when the logic block is idle, can reduce significantly dynamic power.

Automatic stuff: modern synthesis tools support automatic clock gating identifying the logic block where the clock gating can be applied without changing the logical function.

- Most libraries include specific clock gating cells that are recognized the synthesis tool
- Clock gating explicitly coding is too error prone, it may produce glitches, functional errors ...



Architectural Blocks for PM

Clock domains

Hardware blocks fed with the same gated clock supporting clock gating

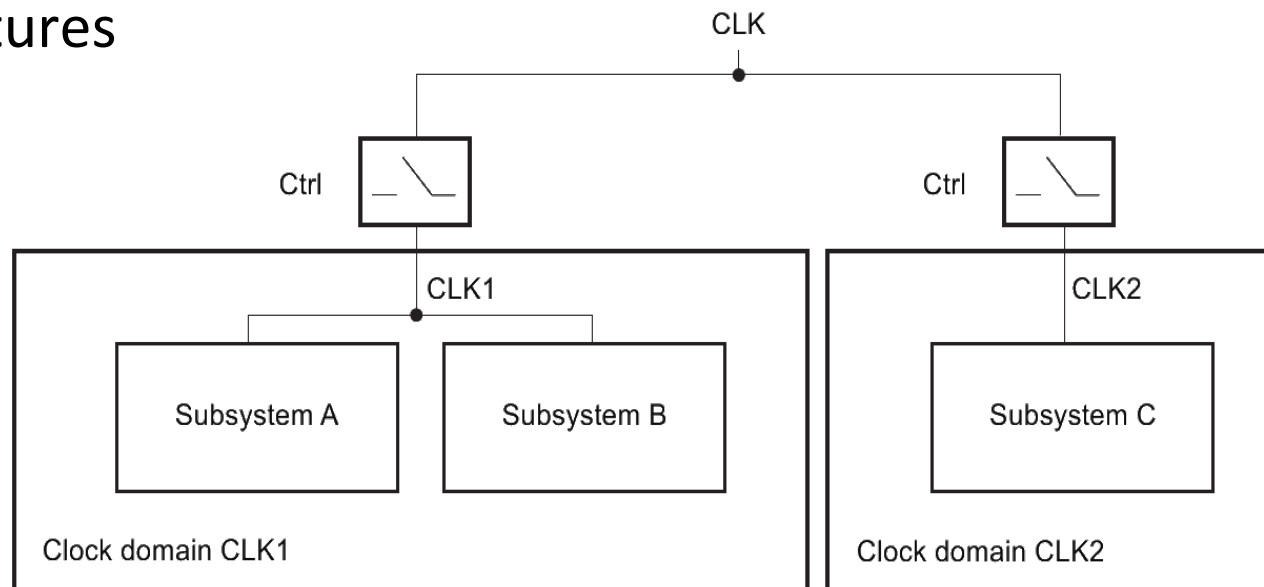
Support clock gating to control dynamic power

- Cut a clock to a group of inactive blocks
- Lower active-power consumption

Two possible states: active or inactive

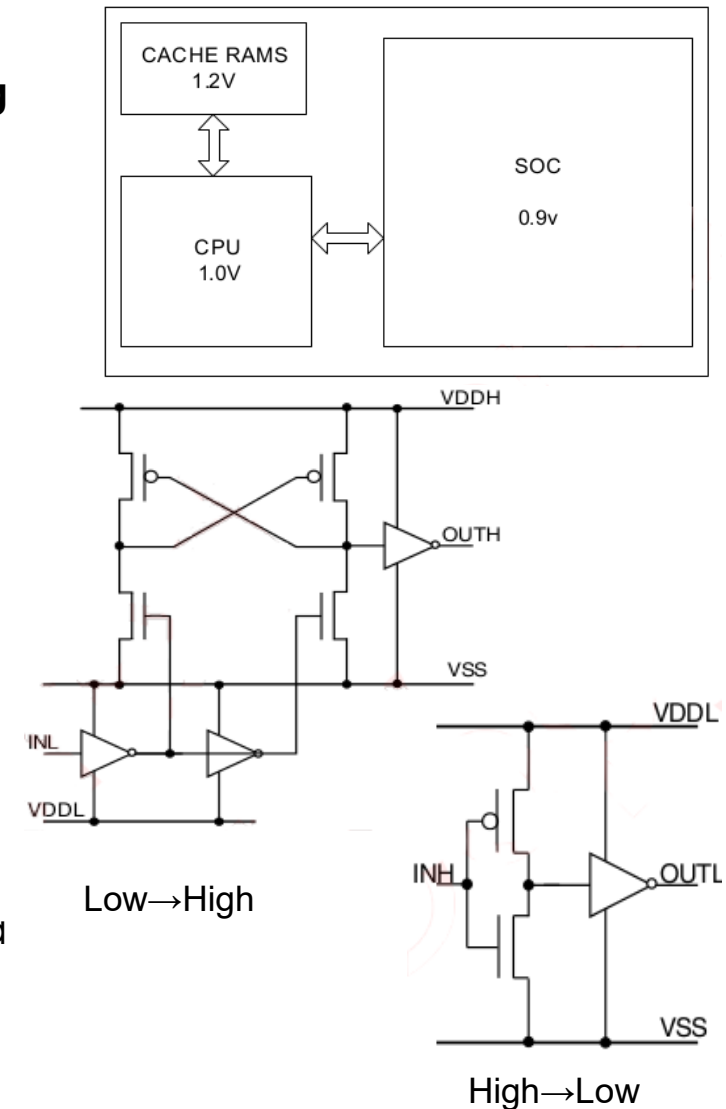
Each domain can have its own frequency

- Enabling Globally Asynchronous Locally Synchronous (GALS) architectures



Multi Vdd

- Since dynamic power is proportional to V_{dd}^2 , lowering V_{dd} on selected blocks helps reduce power significantly
- Apply different voltage to different logic blocks according to performance requirement
 - Account for lower performance
 - Additional complexity to the design
 - More complex power grid
- What about design issues?
 - Unidirectional level shifter design
 - **HighV** → **LowV** are easier to design as 2 not in series, a buffers
 - **LowV** → **HighV** are much complicated to design
 - Level shifter placement
 - **HighV** → **LowV** are placed in the lower domain, destination
 - **LowV** → **HighV** placed in the higher domain with a V_{ddL} line from the lower domain
 - Timing analysis done in parallel with Muti-VDD generation

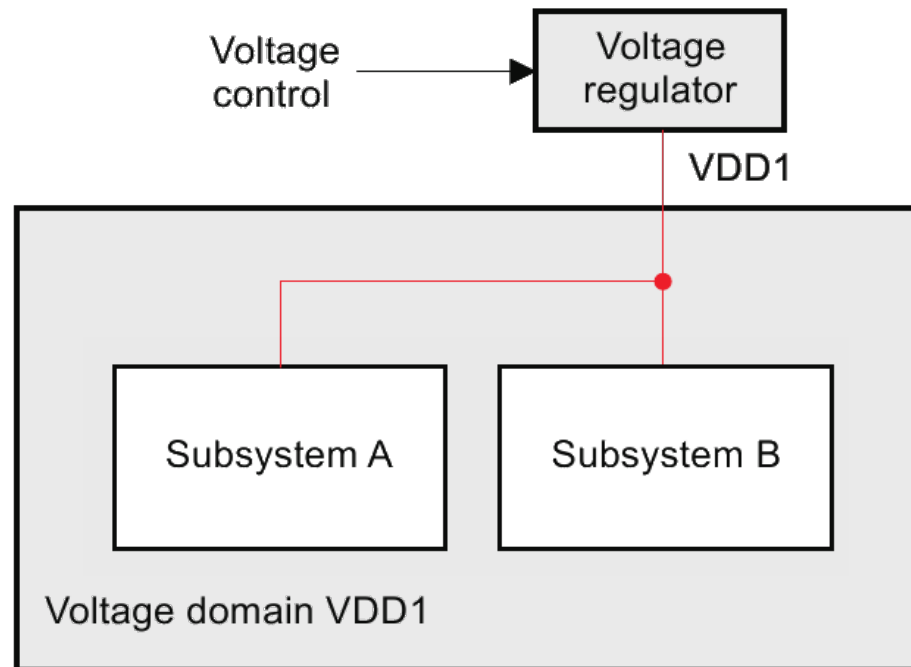


Architectural Blocks for PM

Voltage domains

Groups of HW blocks supplied by same voltage regulator

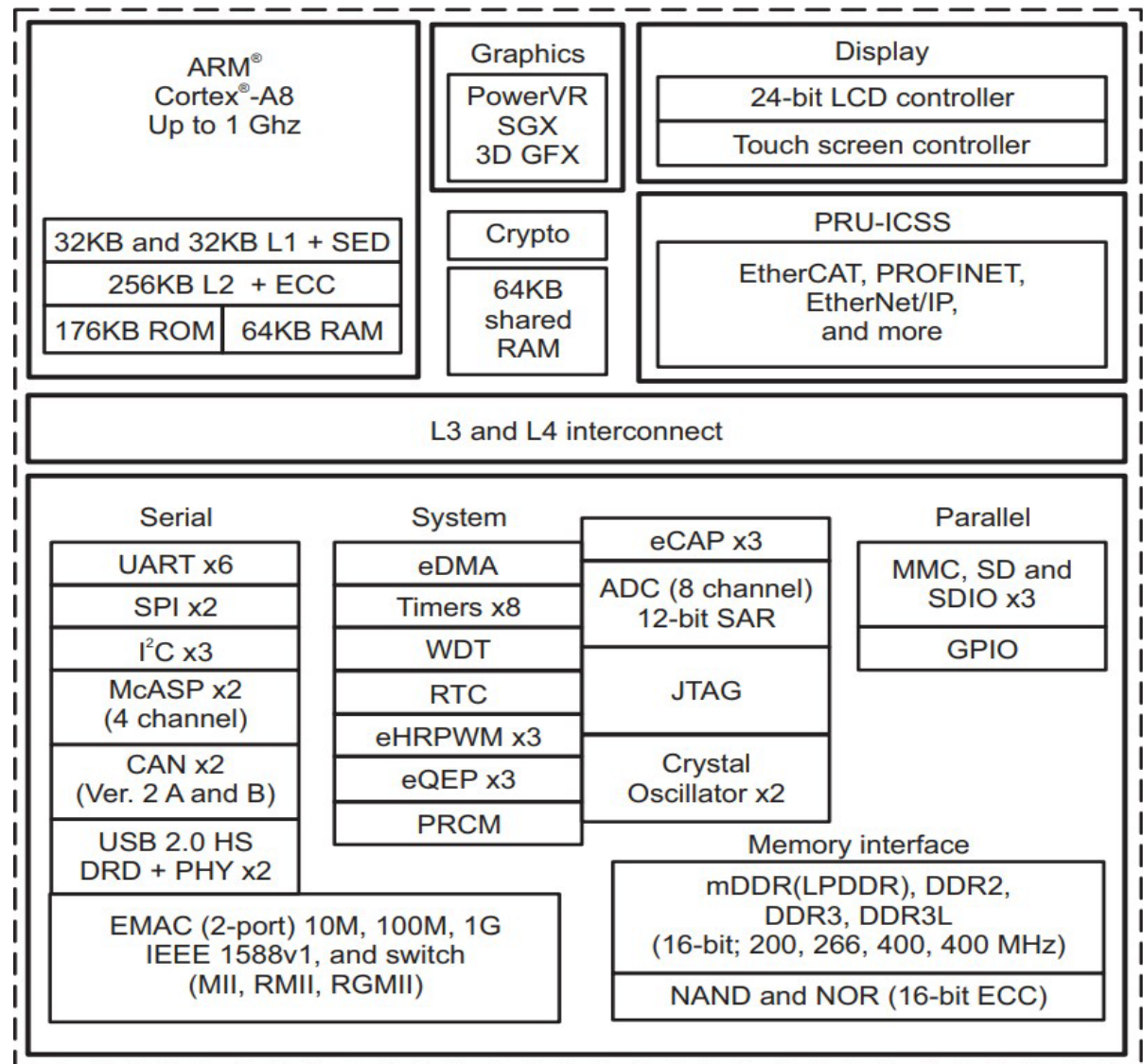
- Power consumption can be controlled by regulating voltages independently
- Assign different operating V to different HW blocks
- Voltage scaling of device subsections
- Voltage scaling allows to reduce power consumption



Power saving knobs

Power domains

- Modern System-on-chip are extremely complex, featuring a multitude of subsystem and functional units



AM335x Functional Block Diagram

Multi-Threshold Logic

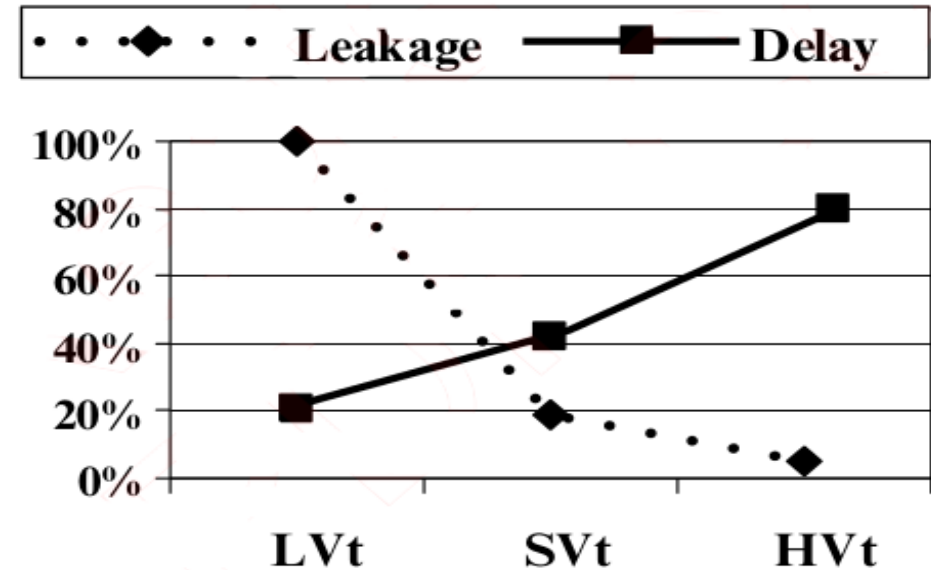
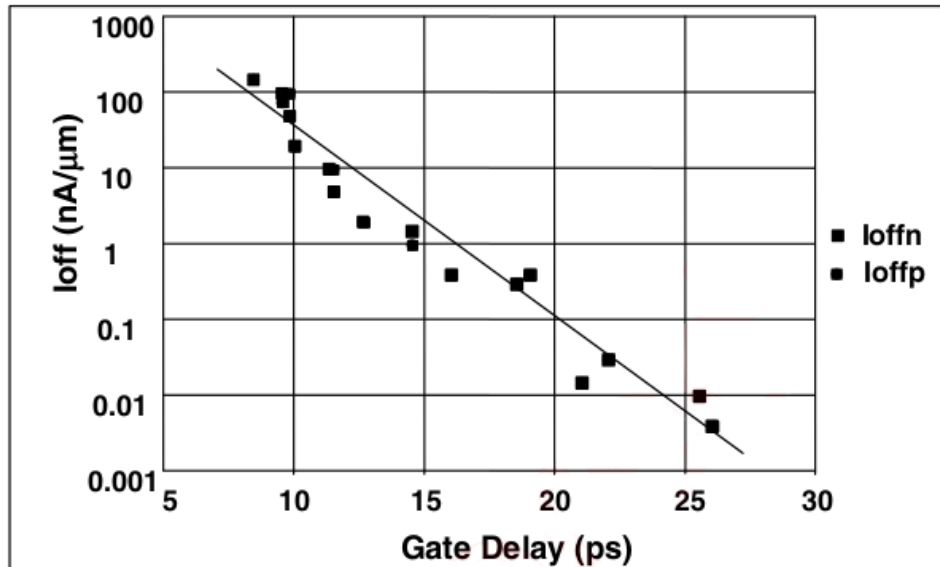
As geometries have shrunk to 90nm and below, using libraries with multiple VT has become a common way of reducing leakage current

- **Many libraries today offer two or three versions of their cells:**

- Low VT, Standard VT, High VT

The synthesis tools can take advantage of these libraries to optimize timing and power simultaneously

- **Dual VT flows are quite common.** The goal of this approach is to minimize the total number of fast transistor by deploying them only when required to meet timing.



Summary on standard low power techniques

Tech-nique	Power Benefit	Timing Penalty	Area Penalty	Impact: Architec-ture	Impact: Design	Impact: Verifica-tion	Impact: Place & Route
Multi Vt	Medium	Little	Little	Low	Low	None	Low
Clock Gating	Medium	Little	Little	Low	Low	None	Low
Multi Voltage	Large	Some	Little	High	Medium	Low	Medium

Power Gating

Allows to reduce both leakage and dynamic power at the same time, while it is first designed to face leakage power

- **More invasive than clock gating:**

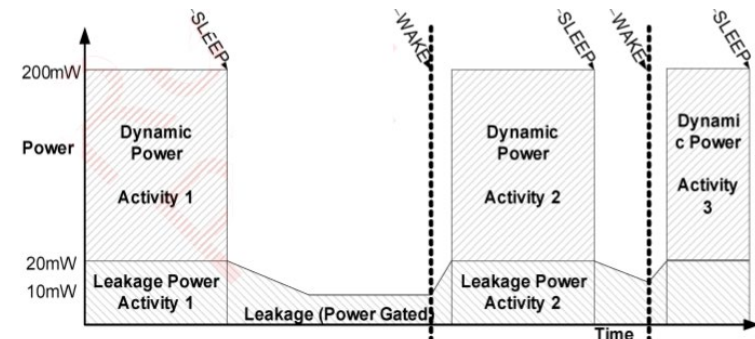
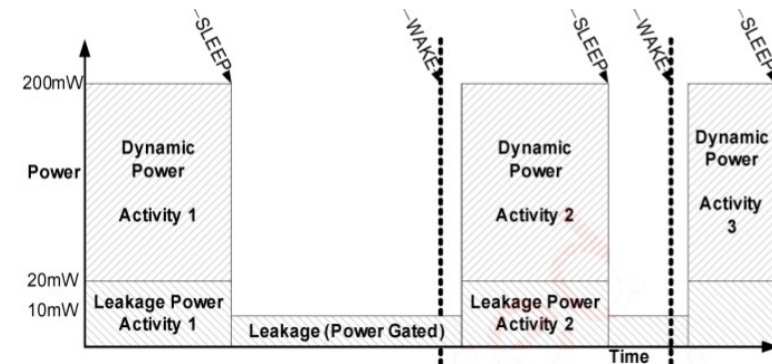
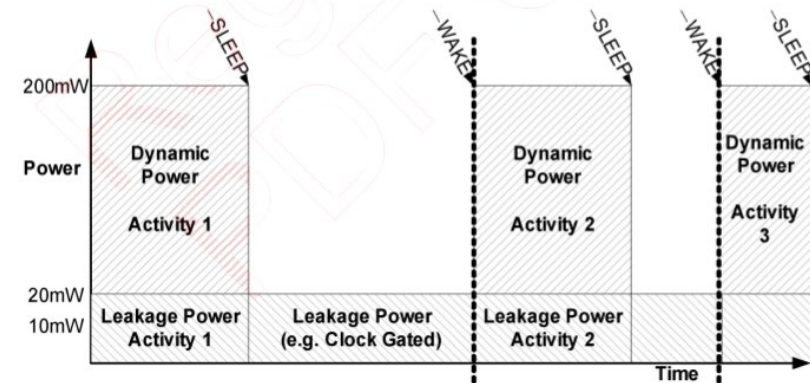
- Affects inter-block communication
- Adds significant time delay
- Require retention capabilities to save the state

- **Activated both at software and hardware level**

- Block shutting down can be scheduled by control software as part of device driver
- Initiated via hardware by timers or power management controllers

- **Any event, there is an architectural trade-off:**

- Amount of possible leakage power saving
- Entry and exit penalties
- Entry and exit dissipated energy



Power Gating impact on different sub-systems

• **Cached CPU subsystem**

- Power gating the entire CPU provides very good leakage power reduction
- Wake-up time response to an interrupt has significant system level design implications
- Cache content are lost every time CPU is powered down, then I will spend power to restore cache blocks
- The net energy savings depend on the sleep/wake activity profile

• **Peripheral subsystem**

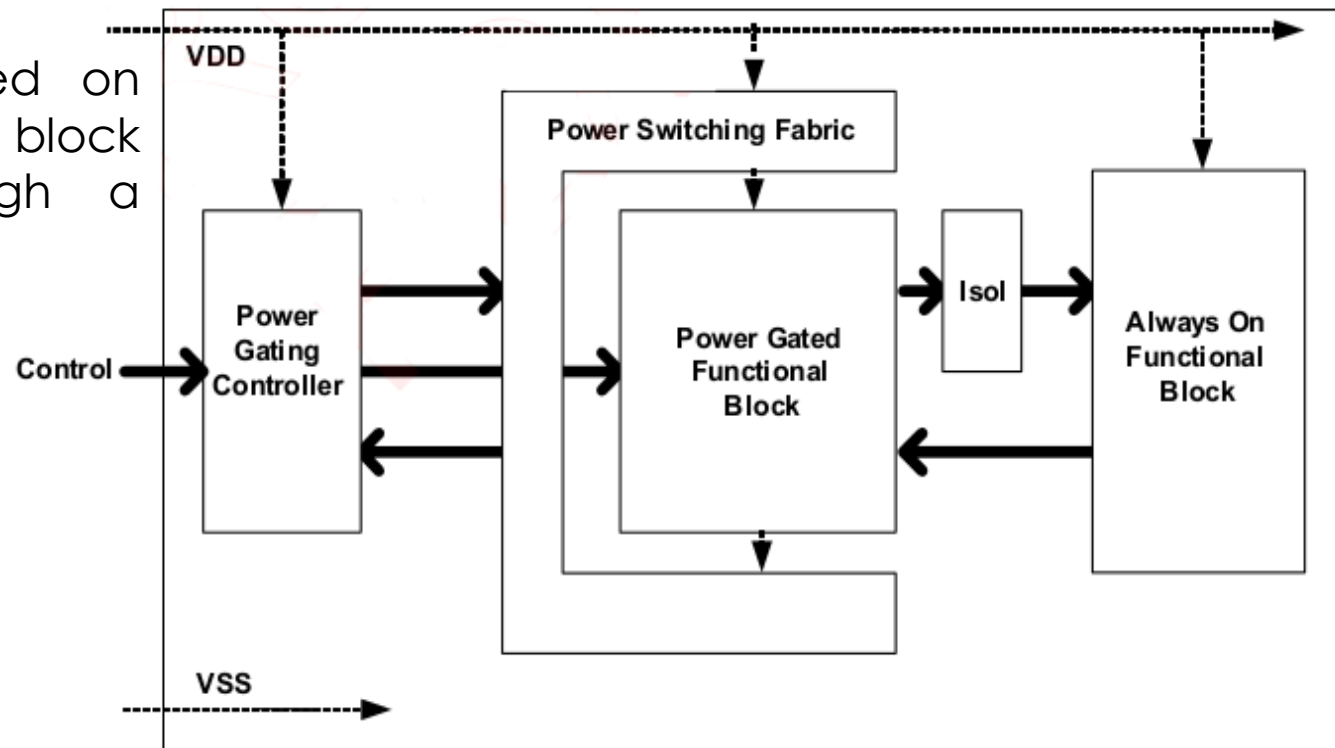
- The device driver requires explicitly load/restore key state
- A better approach for peripheral is the internal state retention, even partial

• **Multi-core CPU**

- Power gating individual CPUs provides very good leakage power reduction
- Power gated CPUs have completed the task, thus local cache state is useless
- More flexible and allows for dynamic power saving algorithms

Power Gating Basics

- Selectively powering down certain blocks in the chip while keeping other blocks powered on
- Unlike an always powered on block the power gated block receives its power through a power-switching network
- We will briefly detail:
 - Power switching fabric
 - Isolation cells
 - Power gating controller
 - State retention
 - Area and timing impact



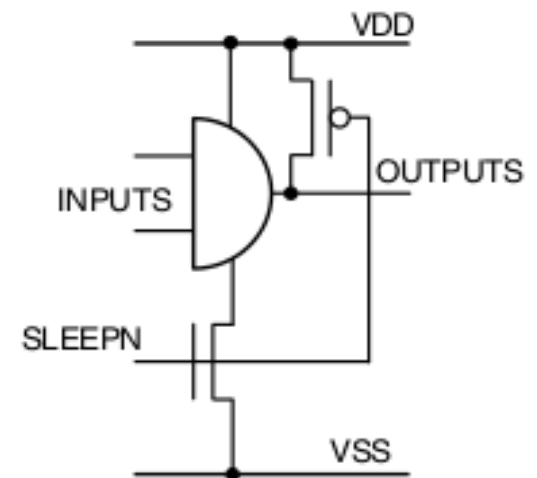
Power Gating: Coarse VS Fine Grain

- **Fine grain**

- The switch is placed locally inside each standard cell in the library.
- The switch is quite large since it has to provide the maximum current possible required by the cell
- 2x-4x area overhead
- The behavior of the entire cell can be easily characterized, allowing to use traditional synthesis flows

- **Coarse grain**

- A block of gates has its power switched by a collection of switch cells.
- More difficult to characterize
- Less area penalty
- Easier to introduce outside an already implemented cell

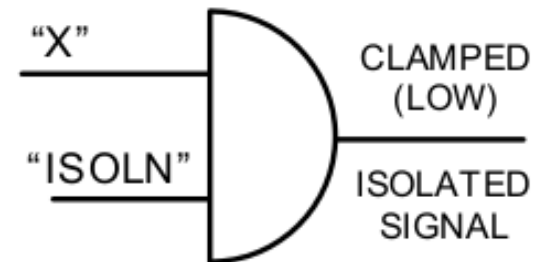
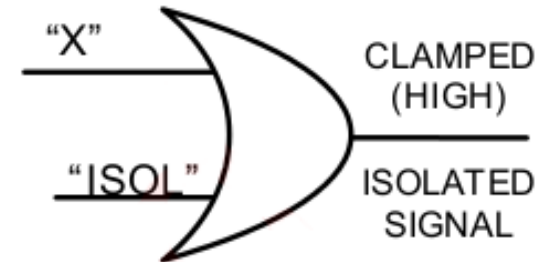


NOTE: strong convergence to coarse grain in last years

Power Gating: Isolation cells

Outputs: We want to be sure than none of the floating outputs of the power down block will result in spurious behavior in the power-up blocks:

- Logic isolation cells:
 - Clamp output value to '0' or '1'
 - High area penalty (area issues)
 - Full port delay (timing issues)
 - Drive the “ISOLx” line
 - **Recommended**
- Transistor based isolation:
 - Pull-up or pull-down transistors
 - Low timing and area penalties
 - Electro migration issues
 - Accurate timing to drive the output when block is completely off



Inputs: inputs of a powered down block are usually not an issue, just drive them to valid logic values by powered up blocks

Power Gating: Isolation Recommendations

Isolate the output of power gated blocks:

- Otherwise multiple isolation cells are required if signal is split

Use isolation cells instead of pull-up pull-down transistors

- Otherwise timing complication
- More complex power gating protocol

Ensure stuck-at-0 and stuck-at-1 faults can be detected during test on the isolation logic

Make sure the isolation cells really are always powered on

- They must stay outside the clock gated area

Avoid clock generated in a power gated block

- And used externally to the block
- It considerably complicates the clock tree synthesis

Architectural Blocks for PM

Power domains

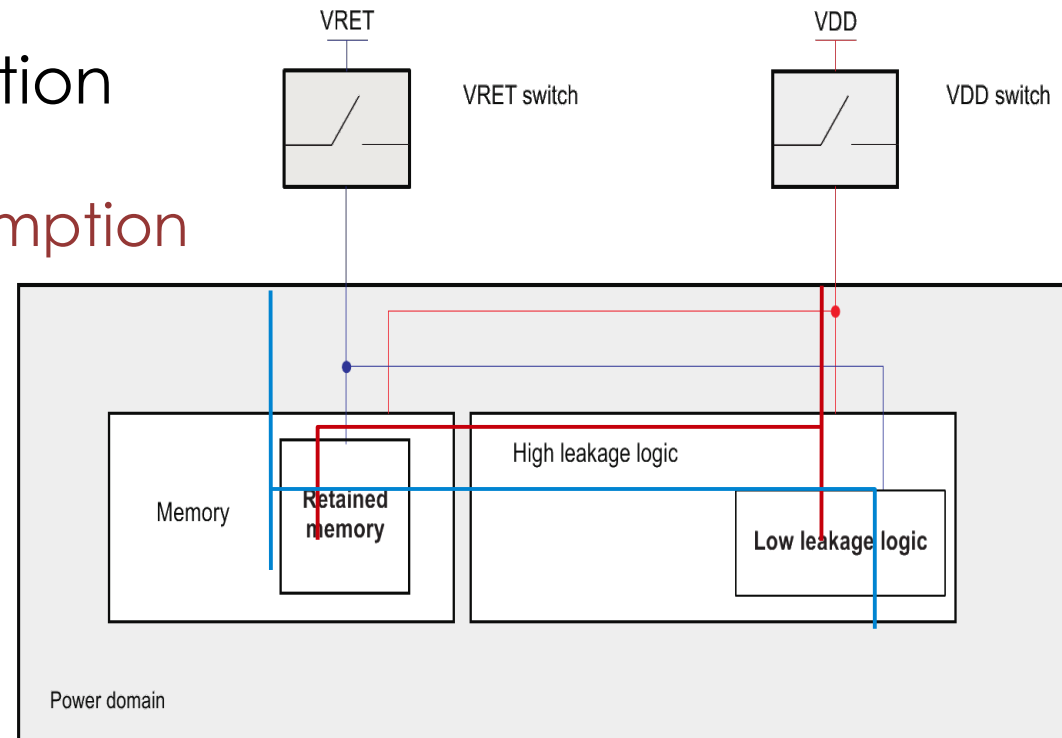
HW blocks supplied by dedicated power rails

- VDD for normal operation
- VRET for low-power operation
lower than active voltage
=> **reduced power consumption**
logic and memory are **not operational**,
but in a *retention state*

- Retention state

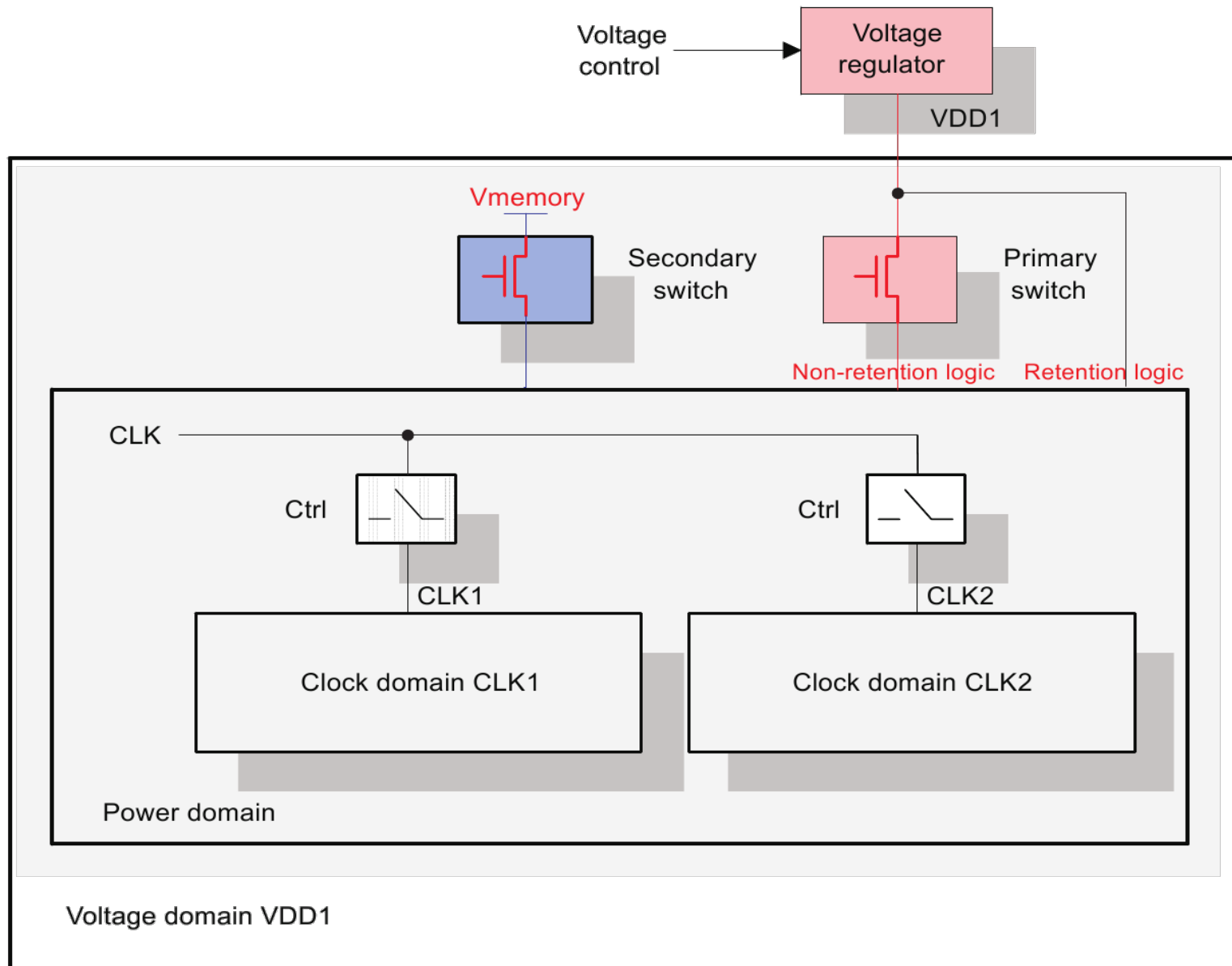
In addition to ON/OFF

Useful to switch to low-power idle mode without losing the context, and quickly switching back to active state when necessary



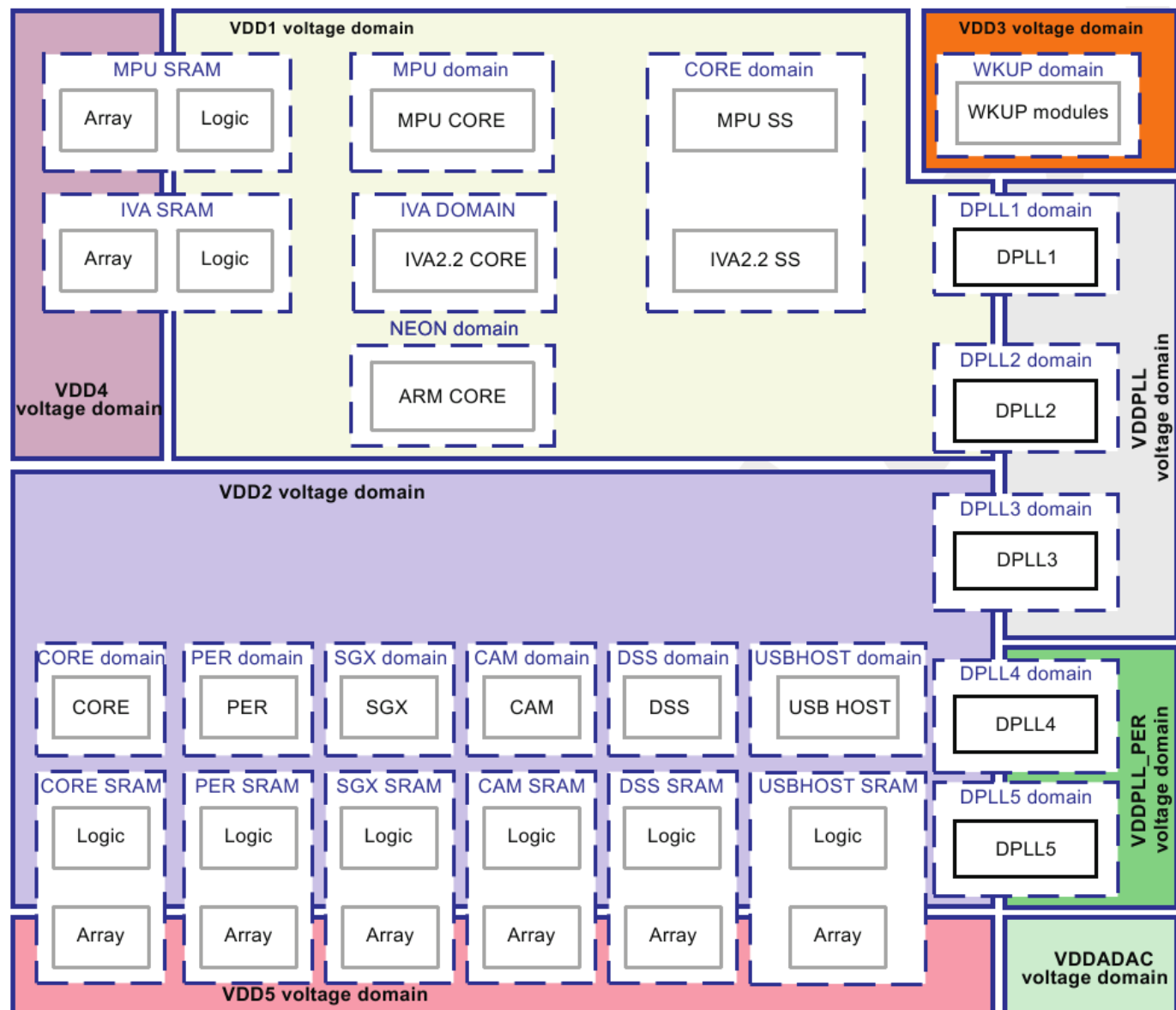
Architectural Blocks for PM

Clock, power and voltage domains hierarchical architecture



Architectural Blocks for PM

OMAP35xx voltage domains



DVFS - Dynamic Voltage and Frequency Scaling

Allocate a variable amount of energy to perform a task power consumption of a digital CMOS circuits

$$P = \alpha \cdot C_{eff} \cdot V^2 \cdot f$$

α switching factor
 C_{eff} effective capacitance
 V operating voltage
 f operating frequency

$$E = P \cdot T \propto V^2 \quad (\text{assuming } f \propto V, T \propto f^{-1})$$

ISSUES

- V contributes quadratically to P
- Frequency only reduction means linear power reduction
- There is a linear region between frequency and voltage
- Current and future sub-micron technologies cannot exploit V reduction too much

DVFS cond't

Different approaches with respect to your goals:

- **Maximize system idle time**
 - runs tasks at the highest OPP (operating performance point), complete the task quickly
 - automatic switch to a low-power mode when possible
 - **Minimize system idle time**
 - dynamic selection of optimal frequency and voltage
 - allow a task to be performed in the required amount of time
 - Operating Performance Points (OPP) voltage (V) and frequency (F) pair
 - The system always runs at the lowest OPP
 - **AVC - Adaptive Voltage Control**
 - Automatic control of the operating voltage
 - Silicon performances/power trade-off (deps on power consumed technology process, operating temperature variations)
 - Power-supply voltage is adapted to silicon performance based on:
 - Performance points (statically)
 - Temp, real-time device performance (dynamically)
-

Power Management Techniques

DVFS – Dynamic Voltage and Frequency Scaling

Basic principle

- Allocate a variable amount of energy resource to perform a task
- Energy required to run a task

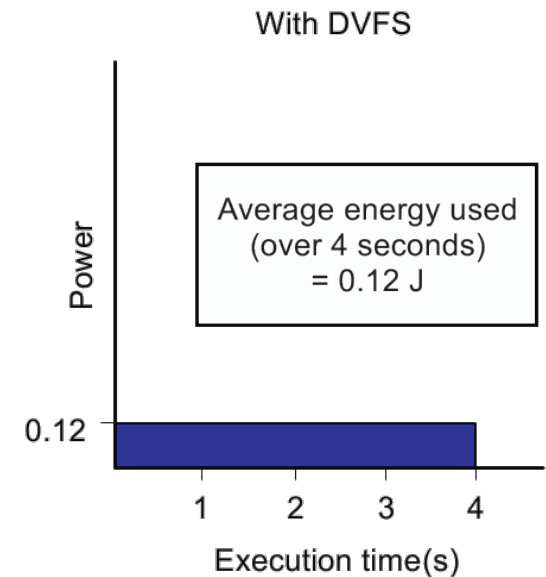
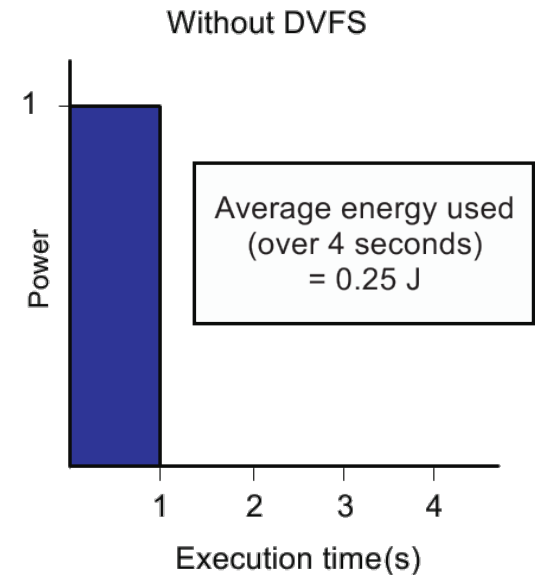
$$E \propto P \cdot T \propto V^2$$

Select Operating Performance Point (OPP)

- A voltage (V) and frequency (f) pair

Minimize system idle time

- The system always runs at the lowest OPP that meets performance requirements
- Reduce dynamic *and* static power



Power Management Techniques

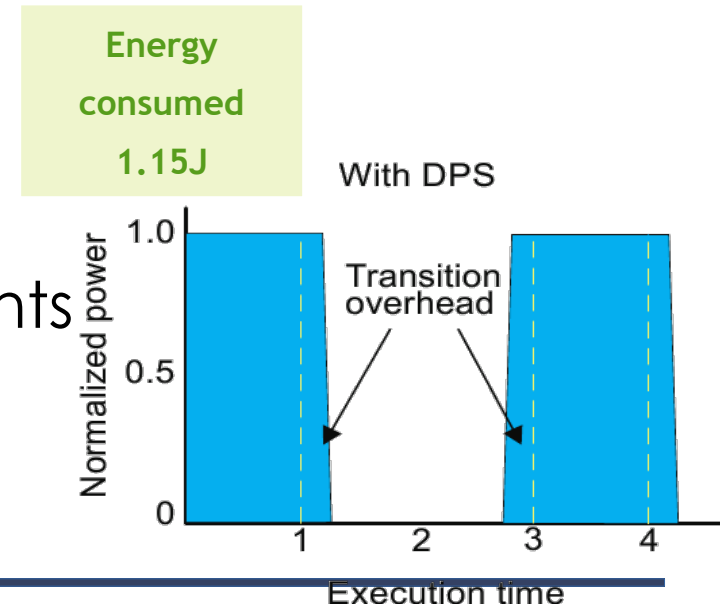
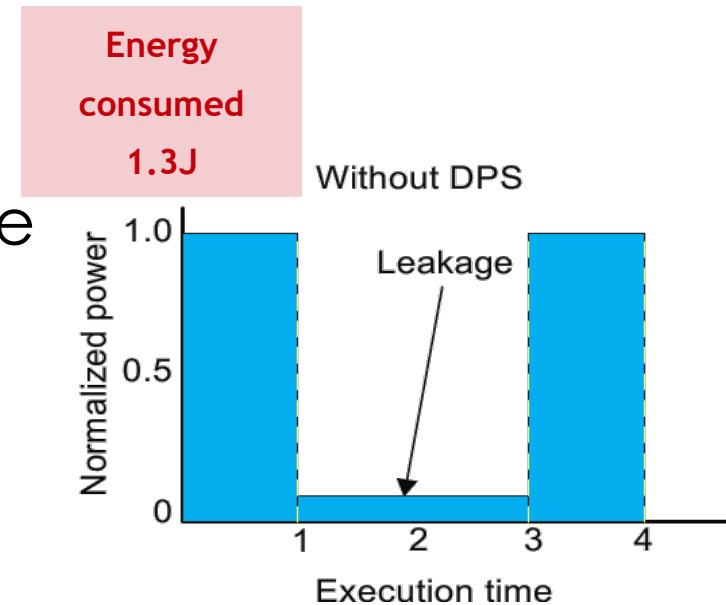
DPS – Dynamic Power Switching

Maximize system idle time

- Automatic switch to low-power mode
- Run tasks at high OPPs to complete tasks quickly
- Reduces leakage power

Main issues

- Transitions overhead
 - Recovery time and power
- Need to dynamically predict applications performance requirements



Power Management Techniques

SLM – Standby Leakage Management

Trades static power consumption for wake-up latency

- Switching the system between high- and low-power modes

Similar to DPS

- Different operating timescale

Latency allowed for mode transitions

DPS: compared to time constraints or deadlines of the application

SLM: compared to user sensitivity so that they do not degrade user experience

- Different context

Who define the transition constraints

DPS: tasks are running and we must grant application performances

SLM: applications not running and must grant system responsiveness

- Different wake-up events

Events used to exit the low-power mode

DPS: application-related, e.g. timer, DMA request, peripheral interrupt, ...

SLM: user-related, e.g. touch screen, key pressed, peripheral connections, ...

Power Management Techniques

AVS – Adaptive Voltage Scaling

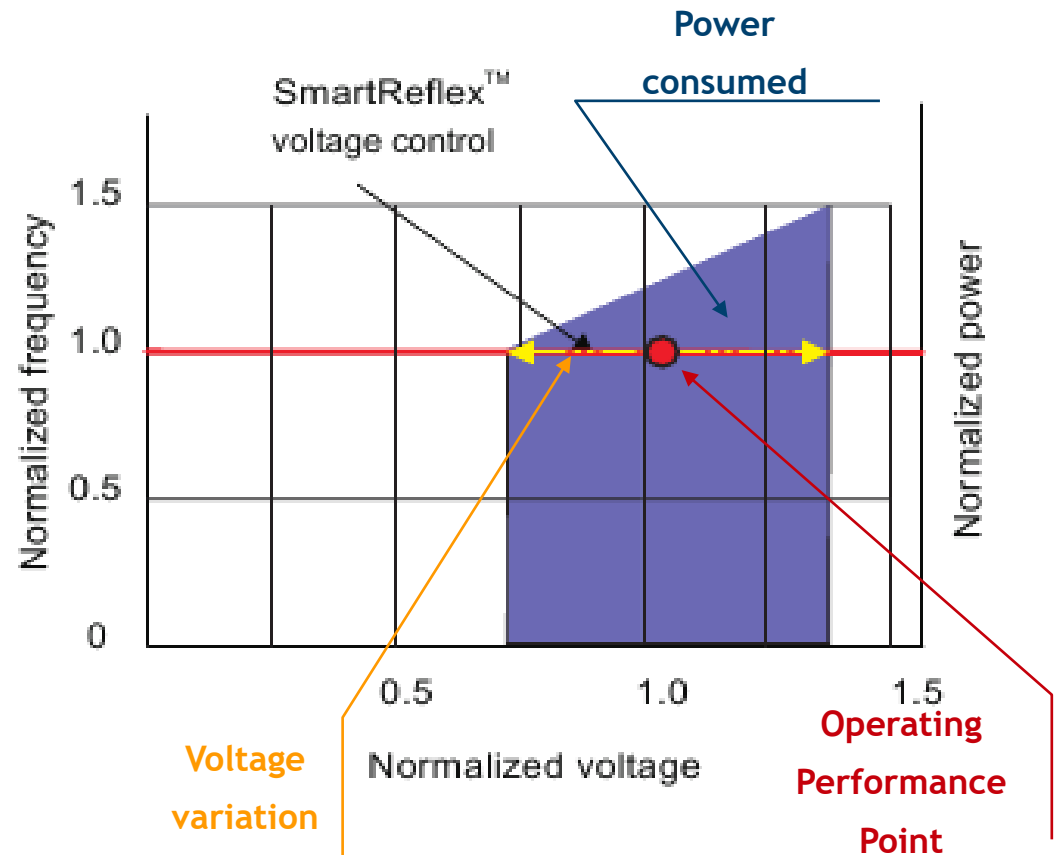
Silicon performance/power trade-off

- Power supply adapted to Silicon performance

Statically, based on performance points

Dynamically, based on temperature-induced real-time performance

- Optimal power/performance trade-off for all devices, across the technology process spectrum and temperature variations



Power Management Techniques

Combining PM techniques

Best active power savings through combination

- AVS

at boot-time to adapt to device process profiles
always to compensate temperature variations

- DVFS

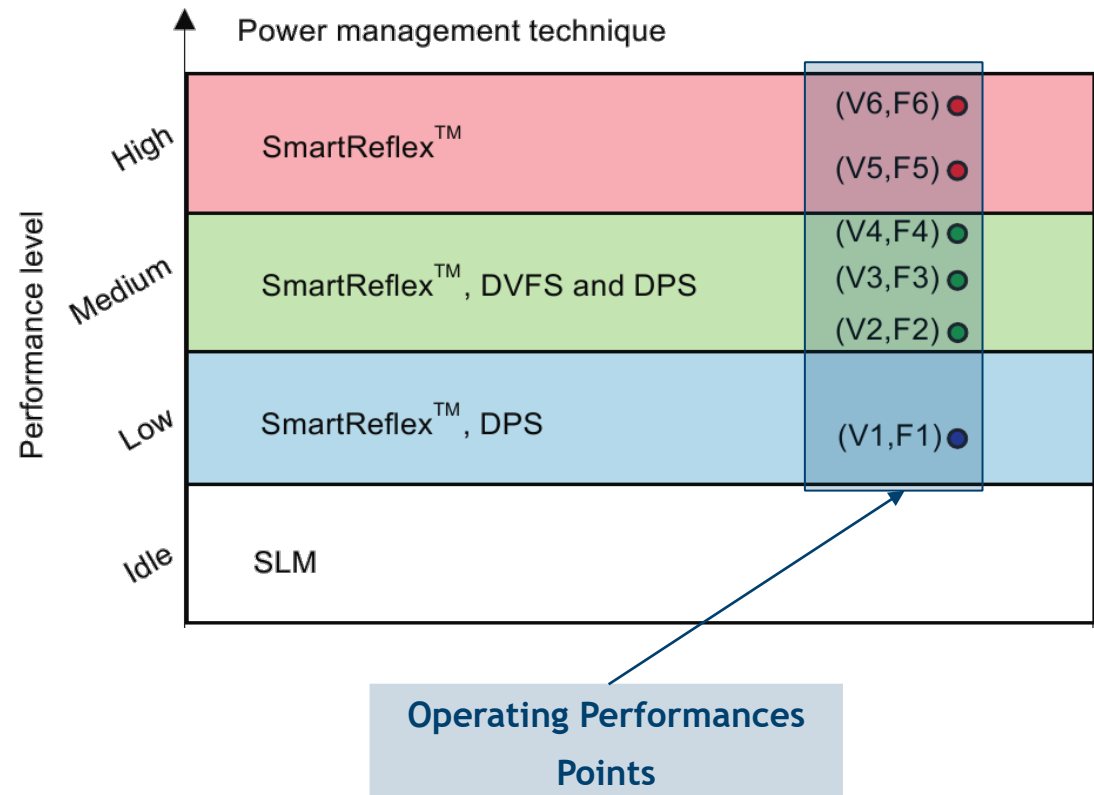
without DPS to scale f and
keep V constant

- DPS

with DVFS to set f to max
for given V

- SLM

no applications running



Power Management Techniques

Hardware mechanisms properties

	Power component optimization		Exploited tuning parameter		Required support	
	Static	Dynamic	V	f	HW	SW
Clock Gating		X		X	X	~
Voltage Scaling	X	X	X		X	X
Power Gating	X		X		X	X

Power Management Techniques

Modern approaches to Power Management

Key points for effective power management

- Exploit partial activity
 - Disable parts of the system when not needed
 - SW does part of the work, HW dependencies do the rest
 - Exploit existing system framework
 - Track dependencies
 - Track usage
 - System constraints assertion
 - Chains of notification
- Driver support required
- Support efficiently OFF modes
 - Use constraints to require operational restrictions
-

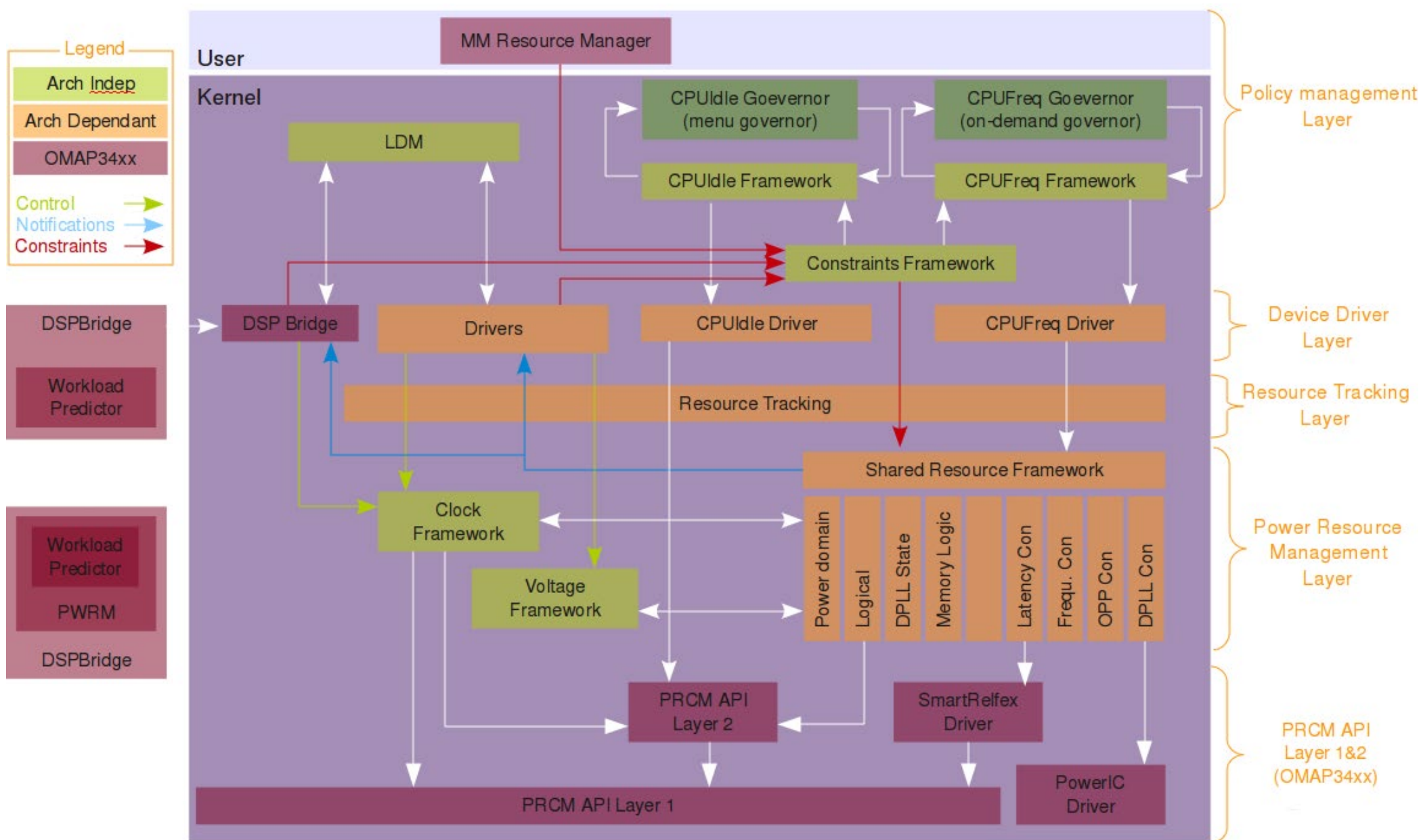
Power Management Techniques

Modern approaches to Power Management (cont'd)

- Control power sources
 - Clock and power domains
 - Voltage domains
 - Inactive state
 - Power saving in OS idle
 - Automatic choice among C-states (*idle states*)
 - System-wide sleep states
 - Active state
 - Dynamic power management
 - Automatic choice among P-states (*performance states*)
-

Power Management Techniques

What we already have (OMAP35xx Example, not new!)



Hardware and Software Co-Design

Which supports do we need?

Hardware support

- Observation points
- Control points
- Power management *mechanisms*

Software support

- HW logical view
- Control software
- Control *policies*

HW/SW co-design for true holistic power management

Is there any need to move part of the management at the hardware level?

Hardware and Software Co-Design

Outlining a possible approach

36

