SITARA™ ARM® PROCESSORS

Boot camp

TEXAS INSTRUMENTS

Linux Power Management Overview and Hands-on

In this session you will learn how to improve product power performance by minimizing power consumption and guaranteeing system performance. In addition, power management techniques enabled via the Linux SDK will be discussed.

LAB: http://processors.wiki.ti.com/index.php/Sitara_Linux_Training

July 2012

Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0)

Agenda

- What You Will Learn
- Motivation for PM
- Overview of Power Management Techniques
- Understanding PM Features in the AMSDK
- Hands-on Labs Throughout to Reinforce Learning
- All of the above focused on AM335x.

Pre-work Check List

- ❑ Installed and configured VMWare Player v4 or later

- ❑ Installed Ubuntu 10.04

- ❑ Installed the latest Sitara Linux SDK and CCSv5

- ❑ Within the Sitara Linux SDK, ran the setup.sh (to install required host packages)

- ❑ Using a Sitara EVM, followed the QSG to connect ethernet, serial cables, SD card and 5V power

- ❑ Booted the EVM and noticed the Matrix GUI application launcher on the LCD

- ❑ Pulled the ipaddr of your EVM and ran remote Matrix using a web browser

- ❑ Brought the USB to Serial cable you confirmed on your setup (preferable)

What you will learn

- GOAL: Improve product power performance
  - Minimize power consumption
  - Guarantee system performance
- Motivation for power management techniques
- Understand power management techniques of AM335x system
  - DVFS: Dynamic Voltage and Frequency Scaling
  - SmartReflex (aka AVS: Adaptive Voltage Scaling)
  - Dynamic Power Switching (DPS)
    - Implemented by Linux's Runtime PM Framework
  - Static Leakage Management (SLM)
    - Device Idle and Standby modes
  - Basic PM hardware architecture:
    - Voltage domains, Power domains, Clock domains
- (Hands on) Gain experience with AMSDK PM Features and Power Measurements
  - Linux kernel configuration
  - Use of PMDC for Power Measurements
  - PM Software Interface (Command line and Matrix tools)
    - Frequency selection (CPUFreq)
    - Suspend/Resume
    - CPU Idle

Why power management?

TI Customers

- ✓ Competitive differentiator
- ✓ Less heat dissipation
- ✓ Smaller, sleeker products with smaller battery
- ✓ Longer battery life
- ✓ Better Features + Increased Usage = Increased Revenue
- ✓ Quicker time-to-marked with advanced features

End Users

- ✓ Smaller, sleeker products
- ✓ Cooler device
- ✓ More exciting, power hungry features like multimedia, streaming video, etc.
- ✓ Improved experience (longer battery life)
- ✓ Lower cost

Power Management Principles

1) Power consumed is proportional to frequency and square of voltage.
    - Small reductions in voltage can be very significant
    - Applicable PM Techniques:
        - Dynamic Voltage and Frequency Scaling (DVFS)
        - SmartReflex (aka Adaptive Voltage Scaling or AVS)

2) Keeping devices powered on consumes a lot of power. Cut or reduce power to entire device or idle device portions.
    - Applicable PM Techniques:
        - Dynamic Power Switching (DPS)
            - Turn off what you're not using!
            - At a granular, per-module level
            - Called "Runtime PM" in Linux
        - Static Leakage Management (SLM)
            - System level idle and/or suspend

Power Management Principles (cont.)

3) For proper operation, you should run at one of the recommended operating voltage and frequency combinations (OPPs)

- Too low voltages can result in data propagation errors and system failure
- Too high voltages result in excessive power consumption
- Hardware timing closure results in a few PROVEN operating performance points (OPPs) which guarantee the system to work properly.

Power Management Techniques: Basics

PM Techniques can be categorized as “ ” or “Idle”:

Adaptive Voltage Scaling

Dynamic Power Switching

➢ Active PM Technique

Performed while device is on and in use. Involves all systems components: HW modules, device drivers, OS & apps. Techniques: AVS (SmartReflex), DPS & DVFS.

Idle PM Technique

Entire device is idled. Various idle states selected based on trade-off of power consumption and wakeup latency.

Techniques: SLM

Static Leakage Management

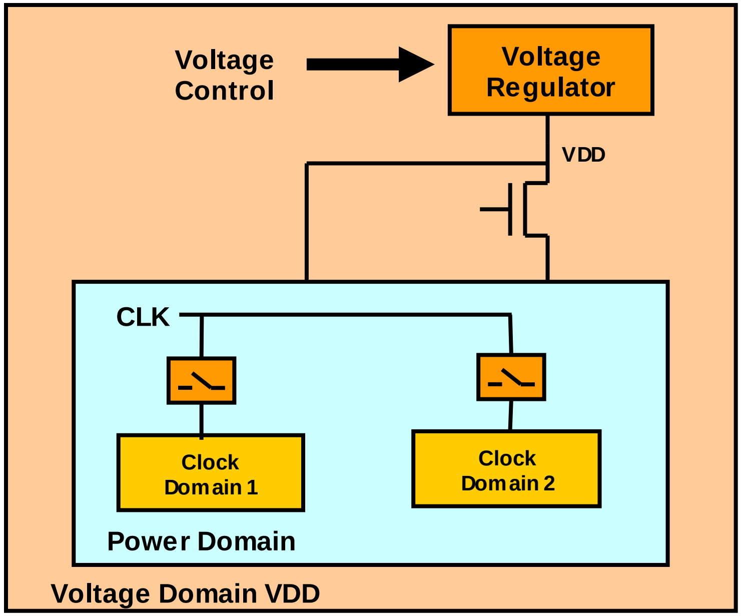
Dynamic Voltage & Frequency Scaling

➔ Power Management is centralized and is handled by the PRCM (Power, Reset and Clock Management) module.

PM Architecture Basics

Devices are partitioned into the following architectural blocks:

- Clock Domain:
  - One or more modules fed with the same clock
  - Clock has independent gating control
- Power Domain:
  - One or more modules fed with the same power rail
  - Power rail controlled by independent power switches
- Voltage Domain
  - Group of modules supplied by the same voltage regulator (embedded or external)
  - “VDDn” sometimes referred to in documentation as a “power rail”
- Domains hierarchy is generally:
  - Voltage domains contain one or more Power Domain
  - Power domains contain one or more Clock Domain
- Control of this infrastructure is done by the Power Reset and Clock Manager (PRCM), which is split in two entities:
  - PRM: Handles voltage, power, reset, wake-up management, system clock source control and clock generation
  - CM: Handles the clock generation, distribution and management



AM335x Power Domains Overview

Refer to Ch. 8 of the TRM for more information:

Power Supply	Power Domain	Modules
VDD_CORE	PD_WKUP	PRCM, Control Module, GPIO0, DMTIMER0, DMTIMER1, UART0, I2C0, TSC, WDT1, SmartReflex, L4_WKUP, DDR_PHY
VDD_CORE	PD_PER	EMIF4, EDMA, GPMC, OCMC, PRUSS, LCD controller, CPSW, USB, MMC0..2, DMTIMER2..7, Uart1..5, SPI0..1, I2C1..2, DCAN0..1, McASP0..1,ePWM0..2, eCAP0..2, eQeP0..1,GPIO1..3,ELM
VDD_CORE	PD_GFX	SGX530
VDD_MPU	PD_MPU	CPU, L1, L2 of MPU
VDD_RTC	PD_RTC	RTC

Lab Break 1

- Labs available at: http://processors.wiki.ti.com/index.php/Sitara_Linux_Training:_Power_Management
- Lab 1: Kernel Configuration
  - Learn the PM related kernel config options.
- Lab 2: Using the PMDC
  - Learn to make power measurements using the PMDC.

Dynamic Voltage and Frequency Scaling (DVFS)

Active PM Technique

- Technique used to scale operating frequency and voltage of hardware
- Well-characterized Operating Performance Points (OPPs) defined for each device
  - OPP specified as pair: (MPU FREQUENCY, VOLTAGE (for weak silicon))
    - Indicates minimum voltage at which ALL devices can meet that frequency requirement
  - For each OPP, software sends control signals to external regulators (DC/DC converters) to set the minimum voltage required.
  - The OS monitors the workload and can **dynamically** adapt voltage/frequency when using certain governor policies (“ondemand”).
- DVFS applicability:
  - AM335x: VDD_MPU and VDD_CORE
- Software Support:
  - OPP can be statically selected by the user via Matrix, or at the Linux command line.
    - Only if governor is “userspace”!
  - The CPUFreq driver in Linux supports multiple “governors”.
    - Governors implement different policies for dynamically managing OPP selection.
      - “powersave” governor: selects the lowest possible frequency
      - “performance” governor: selects the highest possible frequency.

Dynamic Voltage and Frequency Scaling (DVFS)

These charts show the defined OPP’s for current devices:

	OPP	ARM MHz	VDD_MPU
AM335x	SRTurbo	720	1.26
	120	600	1.2
	100	500	1.1
	50	275	0.95

OPP	L3/L4 MHz	VDD_CORE
100	200/100	1.1
50	100/50	0.95

	OPP	ARM MHz	Vdd1
AM37x	1G	1000	1.33
	130	800	1.27
	100	600	1.14
	50	300	0.97

OPP	L3 MHz	Vdd2
100	200	1.14
50	100	0.95

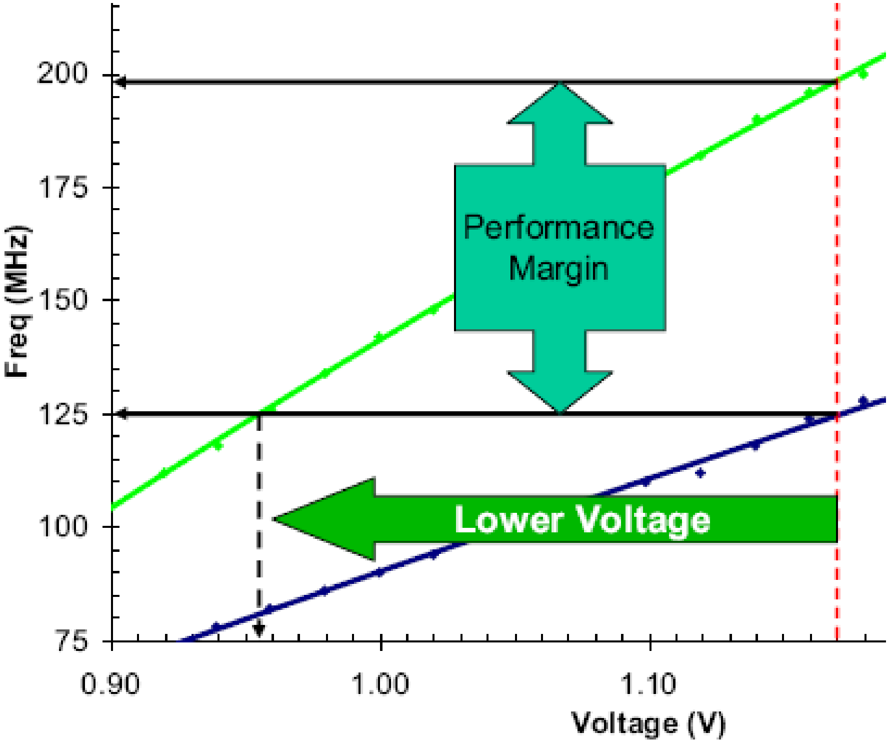
Active PM Technique

Adaptive Voltage Scaling

Active PM Technique

- **SmartReflex** (trademark of Texas Instruments)
  - Silicon manufacturing process yields a distribution of performance capability
  - For a given frequency requirement:
    - Hot/strong/fast devices can meet this at a lower voltage
    - Cold/weak/slow devices needs higher voltage
  - Simple system will set the higher voltage for operating all devices
  - **Smarter system will adapt operating voltage per device!**

- Green line: “hot” device
- Blue line: “cold” device



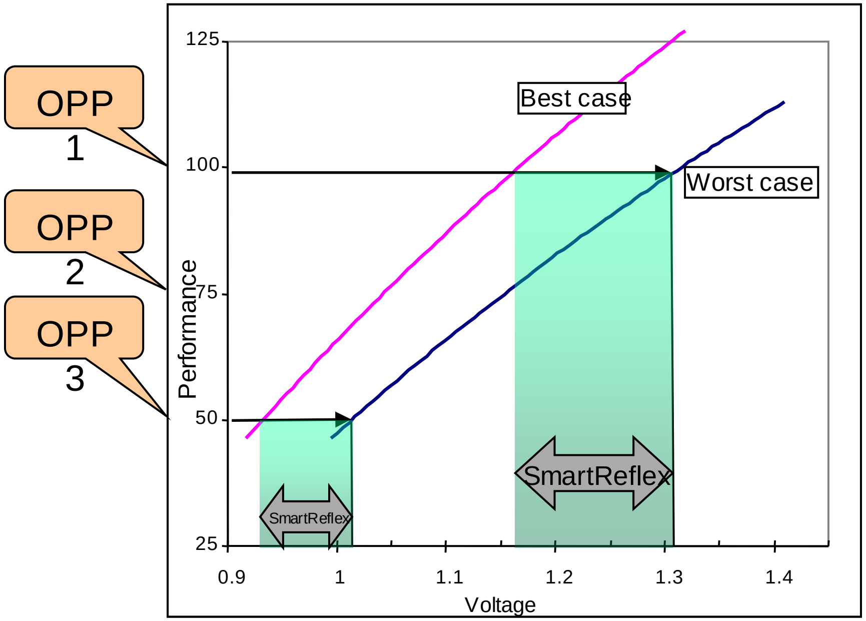
SmartReflex Classes

- **Class-0: Manufacturing Test Calibration**
  - At manufacturing test, the device-optimized operating point voltages are permanently fused into each die. A one time optimization to account for process variations.
- **Class-1: Boot-Time Software Calibration**
  - At boot-up time, device-optimized operating point voltages of the die are determined during calibration. Optimization also accounts for process variations.
- **Class-2: Continuous Software Calibration (AM335x uses Class-2B)**
  - SmartReflex sub-chip does real-time voltage optimization via software loop
  - Optimizes for process, temperature and silicon degradation effects
  - Variants:
    - **Class-2A**: Timer interrupt or other system event (e.g. frequency change) used to initiate interrogation of SmartReflex sub-chip
    - **Class-2B**: SmartReflex sub-chip generates a host CPU interrupt when frequency is outside acceptable range
  - **Key: Software intervention required**
- **Class-3 (AM37x): Continuous Hardware Calibration**
  - SmartReflex sub-chip has a dedicated hardware loop to dynamically optimize voltage for process, temperature and silicon degradation effects
  - MPU intervention not required – SmartReflex sub-chip communicates any required voltage change directly to the PMIC via a hardware interface (e.g. i2c)
  - Optimizes for process, temperature and silicon degradation effects
  - **Key: No Software intervention required**
- **Class-4: Fully Integrated Solution**
  - Class 3 hardware control loop plus voltage regulator integrated on single die

DVFS vs AVS

DVFS:

- Select OPP from available OPPs based on MIPS requirement in scenario
- Lower frequency requirement -> lower voltage
- All die treated the same for a particular application scenario



	OPP	ARM MHz	Vdd1
AM335x	OPPTurbo	720	1.26
	120	600	1.2
	100	500	1.1
	50	275	0.95

OPP	L3 MHz	Vdd2
100	200	1.1
50	100	0.95

AVS (SmartReflex)

- **After** selection of OPP, scale voltage based on device-specific properties (process capability, temperature)
- Faster (also stronger/warmer) process -> lower voltage
- Each die treated differently

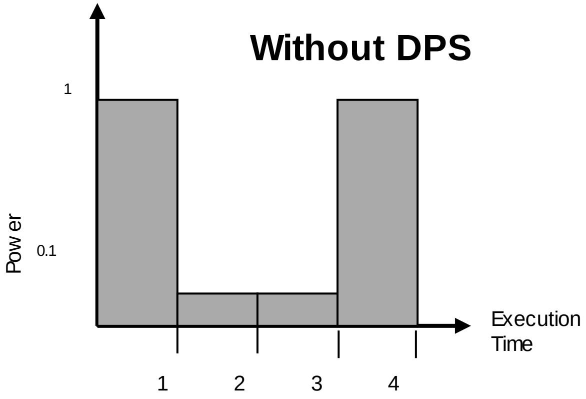
Dynamic Power Switching

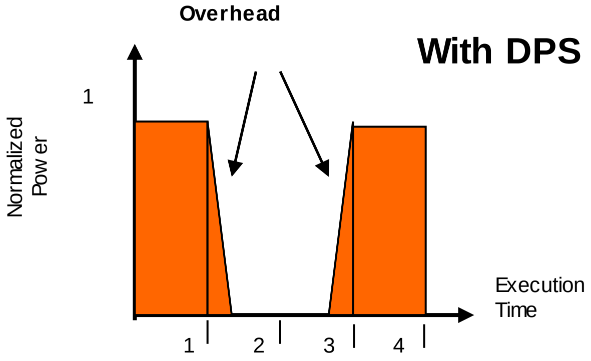
Active PM Technique

Allows certain modules to idle when not being used.

  – Example use cases: during MP3 playback, we could idle CPSW (among others)

- These modules operate in a high performance state to complete its tasks as fast as possible, then dynamically switch to a low-power state (“retention” or “off state”)
  - Context save/restore may be necessary if memory is lost → additional overhead
  - Acceptable wakeup latencies on the order of microseconds
- Software support: In Linux, handled by the “Runtime PM” framework. This is driver-local suspend/resume, managed entirely by individual device drivers operation independently. Drivers can relinquish clocks, resulting in portions of the device in a “clock stop” state.





Static Leakage Management

Idle PM Technique

- Device switches into low-power system modes automatically or under user requests when no application is running
  - Example: Media player shuts off display and enters standby after 10-seconds of on-time with no processing and no user input
  - Typically whole device is in some sort of standby mode
  - Acceptable latencies on the order of milliseconds. Deeper sleep state == longer wakeup latency.
- Devices have multiple sleep states (or C-states)
  - AM335x supports two C-states
    - C1 – MPU WFI
    - C2 – MPU WFI + DDR Self Refresh (lowest power)
- Software support:
    - CPUIdle algorithm automatically chooses from available sleep states based upon the inactivity period and current HW state. Useful for power savings during shorter periods of inactivity.
    - Suspend/Resume support available for long inactivity periods (Deep sleep state)

DPS vs. SLM Review

Active PM Technique	Idle PM Technique
Section of the device in low power mode	Entire device in low power mode (except WKUP domain)
Some parts of system stay active	Full system is inactive
Smaller transition latencies (us)	Larger transition latencies (ms)
Use case : Audio/video Playback - Some domains are going into an idle mode when not needed	Use case: OS idle: Drop into lower-power C-states Suspend-to-RAM: lowest power case

Lab Break 2

- Labs available at: http://processors.wiki.ti.com/index.php/Sitara_Linux_Training:_Power_Management
- Lab 3: Dynamic Voltage and Frequency Scaling (cpufreq)
  - Learn about frequency selection and cpufreq governors.
- Lab 4: Suspend/Resume
  - Learn how to manually suspend and resume the system.
- Lab 5: CPU Idle
  - Gain familiarity with the sysfs interface to CPU Idle information and the AM335x C-states

SITARA™ ARM® PROCESSORS

Boot camp

For more Sitara Boot Camp sessions visit:
www.ti.com/sitarabootcamp

THANK YOU!