

GLSDK DRA7xx PM DVFS User Guide

Contents

About this manual

Acronyms & definitions

Introduction

Overview

- Frequency Scaling
 - CPUFreq
 - CpuFreq Architecture
 - Policy and Governor
 - Policy
 - Frequency Table
 - OPP List
 - Governor
 - Driver
 - Governors provided by CPUFreq
- Enabling DVFS/CPUFreq in Linux
 - Default Governor
 - Build Multiple governors
 - CPUFreq Platform Driver
 - CPUFreq Frequency transition notification
 - CPUFreq User Interface
- Disabling DVFS / CPUfreq
- Voltage Scaling
 - Voltage Layer
- Transitioning to a different Power state(OPPs)
 - Userspace governor
 - All other governors

References

About this manual

This document gives an overview and usage of Dynamic Voltage and Frequency scaling(DVFS) on DRA7xx SOC.

Acronyms & definitions

Acronyms	
Acronym	Definition
DVFS	Dynamic Frequency and Voltage Scaling
CPUFreq	CPU Frequency scaling framework
OPP	Operating Performance Point

Introduction

Dynamic Voltage and Frequency scaling is a framework to change the frequency and/or operating voltage of a processor(s) based on system performance requirements at the given point of time.

Overview

DVFS framework operation involves two tasks listed below:

1. [Frequency Scaling](#)
2. [Voltage Scaling](#).

These two are explained below in detail.

Frequency Scaling

Frequency scaling involves increasing or decreasing the cpu frequency based on a Governor's decision (user input, incase of UserSpace governor) and act upon the decision.

CPUFreq

CPUFreq is a linux kernel framework that monitors the performance requirements of a processor(s) and based on requirements takes decision either to increase to meet the performance requirements or decrease operating frequency in order to save power whenever possible.

CpuFreq Architecture

CPUFreq consists of two components

- *The Governor* - that makes decisions
- *The Driver* - acts based on the decisions made by the governor

Policy and Governor

Policy

Policy is set of rules the system is bound by such as min and max frequency for each cpu, availability of a frequency. Policy for a cpu is created during the CPUFreq framework initialization based on the frequency table.

Frequency Table

Frequency table consists of available frequencies for all cpus in the system. Frequency table is generated/populated based on the Operating Performance Point(OPP) list for each cpu.

OPP List

Operating performance Point(OPP) is a tuple consisting a frequency value and voltage required to run at the frequency. OPP table contains OPPs with a cpu/device name they are applicable to and an availability flag. OPP information of each device is added to OPP list.

Governor

Governor continuously monitors the system performance requirements and when the requirement to change the frequency arises it checks the current cpu policy for frequency limits and requests the driver to change the frequency.

Driver

Multiple drivers can exist in the kernel but there will be only one scaling driver which performs actions based on governor's decision. When the governor request the driver to change the frequency to a target value, driver checks the frequency availability in the OPP list. If its found it scales the device to new frequency.

Governors provided by CPUFreq

The following governors are available in CPUFreq frame work

- Performance - Sets the OPP/Frequency to the highest available
- Powersave - Sets the OPP/Frequency to the lowest available
- Ondemand - Sets the OPP/Frequency based on cpuload trend - starting with OPP_NOM and jumping to max when there's an increase in cpuload
- Userspace - Sets the OPP/Frequency to the opp specified by user
- Conservative - Sets the OPP/Frequency based on cpuload trend - starting with OPP_NOM and increasing/decreasing in steps based on cpuload

'NOTE: For more information each governor refer to Linux kernel documentation @ [<kernel>/Documentation/cpu-freq/governors.txt](#)

Enabling DVFS/CPUFreq in Linux

To enable please do the following:

1. Launch menuconfig "make ARCH=arm menuconfig" 2. Enable CpuFrequency scaling and select at least one governor (ondemand,performance etc):

```
CPU Power management options --->
[*] CPU Frequency scaling
    ....
    Default CPUFreq governor (ondemand) --->
    *- 'performance' governor
    <M> 'powersave' governor
    <M> 'userspace' governor for userspace frequency scaling
    *- 'ondemand' cpufreq policy governor
    <M> 'conservative' cpufreq governor
    <*> Generic CPU0 cpufreq driver
    ARM CPU frequency scaling drivers --->
```

2. Enable OPP Library

```
Power management options --->
    *- Power Management support
    ....
    *- Run-time PM core functionality
    [*] Power Management Debug Support
    [ ] Extra PM attributes in sysfs for low-level debugging/testing
    [ ] Test suspend/resume and wakealarm during bootup
    < > Advanced Power Management Emulation
    *- Operating Performance Point (OPP) Layer Library
    [ ] Enable workqueue power-efficient mode by default
```

Default Governor

Default governor is "ondemand". The default governor can be changed through 'menuconfig': **Note: It's recommended to use "ondemand" governor as it's known to increase device life/POH greatly and save power when higher performance is not required. Using "performance" governor will cause higher power consumption, leakage, heating and may reduce device Power On Hours**

```

Choose the governor you want to use as default governor by selecting it from the list at:
CPU Power management options --->
[*] CPU frequency scaling
  Default CPUFreq governor (ondemand) --->
    () performance
    () powersave
    () userspace
    (x) ondemand
    () conservative
  
```

Build Multiple governors

Multiple governors can be built and exist in the kernel by selecting them through "menuconfig".

```

Select the governors you want to build in to kernel.
CPU Power management options --->
[*] CPU Frequency scaling
  ....
  Default CPUFreq governor (ondemand) --->
  *- 'performance' governor
  <M> 'powersave' governor
  <M> 'userspace' governor for userspace frequency scaling
  *- 'ondemand' cpufreq policy governor
  <M> 'conservative' cpufreq governor
  <*> Generic CPU0 cpufreq driver
  ARM CPU frequency scaling drivers --->
  
```

Governors can be switched at any time through the 'sysfs' interface.

- To list all available governors:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
```

- To see current active governor

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- To switch to a different governor

```
echo -n "<governor_name>" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

e.g. to switch to 'userspace' governor

```
echo -n "userspace" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

CPUFreq Platform Driver

The default platform scaling driver is, cpufreq-cpu0, cpu generic driver (on eralier platforms this was omap-cpufreq)

```

CPU Power management options --->
[*] CPU Frequency scaling
  ....
  <*> Generic CPU0 cpufreq driver
  ARM CPU frequency scaling drivers --->
  
```

CPUFreq Frequency transition notification

CPUFreq driver supports notifications of frequency changes

CPUFreq User Interface

CPUFreq exports a lot of information to user through sysfs interface.

To see information pertaining to a cpu(let's say cpu0):

- current frequency of a cpu:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
```

- Available frequencies:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

- Available governors

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
```

- Current governor/ governor in use

```
cat /sys/devices/ system/cpu/cpu0/cpufreq/scaling_governor
```

Disabling DVFS / CPUfreq

1. Launch menuconfig "make ARCH=arm menuconfig" 2. De-select CPU Frequency scaling option through menu

```
CPU Power management options --->
[ ] CPU Frequency scaling
```

Voltage Scaling

Voltage scaling is achieved using voltage layer and regulator framework(driver). When the CPUFreq driver scales the device frequency, voltage corresponding to the frequency(target_voltage) is looked-up in the opp list. The device scale function requests the voltage layer to scale the device voltage to the target_voltage.

Voltage Layer

The voltage layer consists of the information of all voltage domains in the system and configures all VDDs during voltage layer initialization. When a VDD is configured a regulator supply handle is acquired and stored in the corresponding vdd structure.The regulators scale/set voltage function is plugged in to the VDD's voltage scale function pointer. Thus when a voltage change is requested forwarded to a VDD. The voltage layer requests the regulator framework to change the device voltage to the target voltage. Regulator driver verifies if the target voltage is in within the limits of the voltage domain and regulator supply constraints. If all the checks go through then the regulator changes voltage of the requested device to the target voltage.

Transitioning to a different Power state(OPPs)

Userspace governor

Using "userspace" governor one can change the current OPP:

- Select userspace governor

```
echo -n "userspace" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- When using "userspace" governor to change to a low/high power state/OPP, first check the available frequencies:

```
$cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

Select a frequency and to change current cpu frequency, execute the following:

```
$echo -n "<new_frequency>" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

when the frequency is changed, system voltage is also changed to meet the new requirements as part of scaling: This is done in two ways:

```
* when new frequency is higher (moving to high power state/opp)
* Voltage is increased first then the frequency,
* when new frequency is lower (moving to low power state/opp)
* Frequency is reduced first then the voltage.
```

All other governors

For all other governors user does not need to give input in order to move to a low power/high power state. The governor takes the decision to transit based on system requirements and does the scaling of frequency and voltage. Each governors provides some configuration options which can be set/modified through sysfs. For detailed configuration options available for each governor refer to kernel documentation on governors @ <kernel>/Documentation/cpu-freq/governors.txt

References

For more information on CPUFreq framework refer to the documentation available in Linux Kernel source @ <Kernel>/Documentation/cpu-freq.

{{		Keystone=	C2000=For	DaVinci=For	MSP430=For	OMAP35x=For	OMAPL1=For	MAVRK=For	For technical s
1. switchcategory:MultiCore=	▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum	▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum	technical support on the C2000 please post your questions on The C2000 Forum.	technical support on the DaVincoplease post your questions on The Forum. Please post only comments	technical support on MSP430 please post your questions on The MSP430 Forum. Please post	technical support on OMAP please post your questions on The OMAP Forum. Please post only comments	technical support on OMAP please post your questions on The OMAP Forum. Please post only	technical support on MAVRK please post your questions on The MAVRK Toolbox	please post you questions at http://e2e.ti.cor. Please post on comments abo article GLSDK DRA7xx PM D User Guide he

Forum. Please post only comments about the article **GLSDK DRA7xx PM DVFS User Guide** [here](#).