

ARM mach-shmobile

CPU Core PM Overview

Magnus Damm

June 2012

Outline

Hardware

- Overview

- Clocks and Voltage Control

- Power Domains

Software

- CPU Hotplug

- CPUIdle

- Future Work

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIde

Future Work

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

Overview

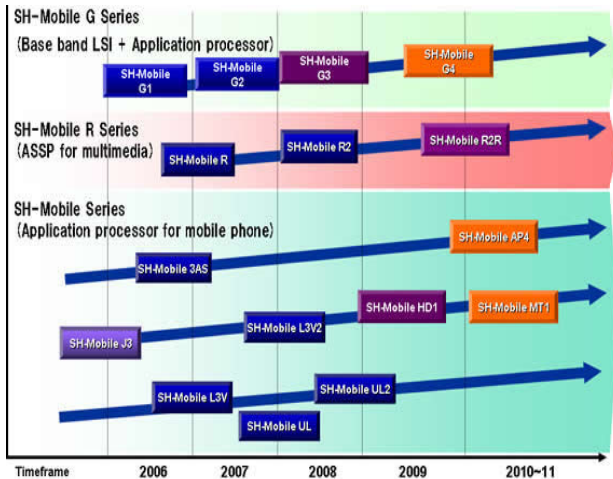
What is hiding under `arch/arm/mach-shmobile/`?

Product lines:

- ▶ R-Mobile - Portable devices
- ▶ R-Car - Automotive
- ▶ Emma Mobile - Former NEC Electronics SoCs

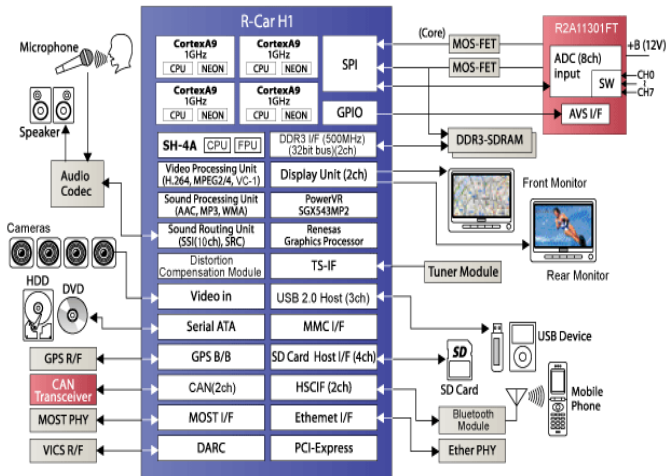
R-Mobile / SH-Mobile

sh7367, sh7377, sh73a0, sh7372, r8a7740
 [ARM1176JZF-S, Cortex-A8, Cortex-A9]



R-Car

r8a7779 [Quad Cortex-A9]



Emma Mobile

EMEV2 [Dual Cortex-A9]

Media Performance

▶ HD

▶ EMMA Mobile/EV2

- CPU ARM Cortex™-A9 w/NEON 533MHz
- EV2 Dual CPU
- AV Engine H.264, MPEG2, MPEG4, VC-1 MP3, AAC, WMA, AC-3 Dec
- Graphics 14.7Mpol/sec, 500Mpol/sec (EV2)
- Mobile DDR-400/DDR2-533
- Full HD(1980x1080) Video

▶ EMMA Mobile1-S

- CPU ARM11™ 500MHz
- DSP K7 500MHz
- H.264 Video Engine
- Mobile DDR x32bit 166MHz
- WVGA I/F
- D1 Video

▶ D1/VGA

▶ QVGA

▶ MP201

- CPU ARM9™ 250MHz
- DSP K6 250MHz
- Mobile SDR 125MHz
- WQVGA LCD I/F
- QVGA Video

~CY2008

2009

2010

2011

▶ EMMA Mobile-Y

Under Planning

Full HD (1980x1080) Video

H.264 (BP/MP/HP) Dec

MPEG2 (MP) Dec

MPEG4 (SP/ASP)

VC-1 (SP/MP/AP) Dec

Full HD @30fps

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

Clock Control - What makes the SoC tick?

Three types of clocks:

- ▶ Low Frequency: 32 KHz (Mandatory)
- ▶ Medium Frequency: 10–30 MHz
- ▶ High Frequency: 40–80 MHz

Voltage Control

PMIC interfaces:

- ▶ R-Mobile: I2C + GPIO
- ▶ R-Car: SPI + GPIO
- ▶ Emma Mobile: SPI + GPIO

Actual SoC and PMIC varies with board design.

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

Power Domains - Overview

SoC Power Domains:

- ▶ R-Mobile: Fine grained, parent-child
- ▶ R-Car: Coarse grained, flat
- ▶ Emma Mobile: Fine grained, flat

Power Domains - R-Mobile

sh7372 CPU Core Power Domains:

- ▶ Core Standby: CPU Core + L1 cache
- ▶ A3SM: Above + L2 cache
- ▶ A4S: Above + Power Domains with I/O Devices

Power Domains - R-Car

R-Car CPU Power Domains from `smp-r8a7779.c`:

```
static struct r8a7779_pm_ch r8a7779_ch_cpu1 = {  
    .chan_offs = 0x40, /* PWRSR0 .. PWRER0 */  
    .chan_bit = 1, /* ARM1 */  
    .isr_bit = 1, /* ARM1 */  
};
```

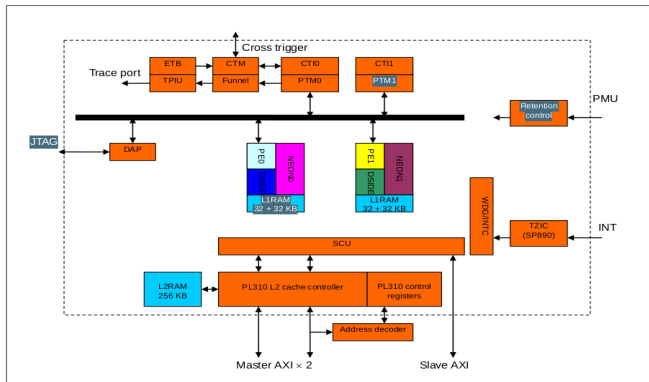
```
static struct r8a7779_pm_ch r8a7779_ch_cpu2 = {  
    .isr_bit = 2, /* ARM2 */
```

```
static struct r8a7779_pm_ch r8a7779_ch_cpu3 = {  
    .isr_bit = 3, /* ARM3 */
```

CPU0 always on, CPU1 -> CPU3 in individual power domains

Power Domains - Emma Mobile

EMEV2 CPU Core Power Domains:



Power domain	PD_TOP	PD_HM	PD_DS0	PD_PE0	PD_NE0	PD_DS1	PD_PE1	PD_NE1
Modules	L1RAM L2RAM Retention F/F	SCU/PL310 CoreSight IP	DSIDE0	PE0	NEON0	DSIDE1	PE1	NEON1

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

CPU Hotplug - Overview

In-kernel interface:

- ▶ For bring up / shutdown of secondary CPU Cores
- ▶ Used by kernel code such as Suspend-to-RAM and Kexec

Also tied into a standard /sys user space interface

CPU Hotplug - User Space Interface

Turn off CPU3:

```
# echo 0 >/sys/devices/system/cpu/cpu3/online
```

Turn on CPU3:

```
# echo 1 >/sys/devices/system/cpu/cpu3/online
```

Check current state of CPU3:

```
# cat /sys/devices/system/cpu/cpu3/online
```

CPU Hotplug - ARM CPU Shutdown Interface

Trigger shutdown from user space:

```
# echo 0 >/sys/devices/system/cpu/cpu3/online
```

ARM kernel executes this function on the offline target CPU:

```
void platform_cpu_die(unsigned int cpu);
```

This is followed by this function call on a different CPU:

```
int platform_cpu_kill(unsigned int cpu);
```

`platform_cpu_kill()` returns 1 on success.

CPU Hotplug - ARM CPU Bring up Interface

Trigger boot from user space:

```
# echo 1 >/sys/devices/system/cpu/cpu3/online
```

ARM kernel executes this function on a different CPU:

```
int boot_secondary(unsigned int cpu, struct  
task_struct *idle)
```

CPU Hotplug - ARM CPU Bring up Interface

The ARM CPU Hotplug interface `boot_secondary()` is used for CPU Hotplug as well as SMP boot.

The SMP kernel can be instructed to boot with a single CPU.

- ▶ Set “maxcpus=1” on the kernel command line.
- ▶ Later use CPU Hotplug interface to boot secondary cores

Useful to track down SMP boot issues without early console.

CPU Hotplug - R-Car r8a7779

On r8a7779 power domains are controlled via CPU Hotplug:

- ▶ CPU0 is always on due to lack of power domain
- ▶ CPU1 -> CPU3 can be power managed via CPU Hotplug

Experimental PMIC integration code allows us to dynamically control the voltage based on power domains. As an example, lower voltage can be used when all CPU cores are off.

CPU Hotplug - Back porting to v3.0

CPU Hotplug can be used in older kernels:

- ▶ “maxcpus=1” is however broken in v3.0
- ▶ Back porting allowed us to solve the issue

Thanks to Simon Horman for his support with back porting.

CPU Hotplug - Back porting to v2.6.35

CPU Hotplug can also be used in ancient kernels:

- ▶ CPU Hotplug is broken in v2.6.35 (init section issue)
- ▶ Solved by local custom fix

Back porting v3.3-rc to v2.6.35 is non-trivial.

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIde

Future Work

CPUIdle - Overview

Architecture independent overview:

- ▶ Light: Low latency - Few dependencies - Basic Power Savings
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ Deep: High latency - Many dependencies - Good Power Savings

Theory: For best power savings, enter as deep mode as possible!

Both clocks and power domains are controlled via CPUIdle.

CPUIidle - Overview

CPUIidle implementation varies with hardware:

- ▶ R-Mobile: CPU Core and I/O Device Power Domains
- ▶ R-Car: CPU Core Power Domains, Coupled/Scheduler update?
- ▶ Emma Mobile: TBD

CPUIdle - R-Mobile

sh7272 CPUIdle / Suspend-to-RAM example:

- ▶ WFI: CPUIdle only
- ▶ Core Standby: CPUIdle and Suspend-to-RAM
- ▶ A3SM PLL ON: CPUIdle* only
- ▶ A3SM PLL OFF: CPUIdle* and Suspend-to-RAM
- ▶ A4S: Suspend-to-RAM only**

*A3SM CPUIdle patches about to be queued for next merge window

**A4S CPUIdle support is a work in progress

CPUIdle - R-Mobile SMP

Cortex-A9 SMP CPUIdle:

- ▶ WFI: Individual CPU control (clock only)
- ▶ Shutdown Mode: Individual CPU control (power domain)
- ▶ A2SL: Coupled including SCU etc (power domain)
- ▶ A3SM: Coupled include L2 cache (power domain)
- ▶ A4S: Power domain shared with I/O Devices (power domain)

WFI and Shutdown Mode trigger via WFI instruction

CPUIdle - R-Car SMP

Challenge #1:

- ▶ Individual power domains are not controlled via WFI

Solution:

- ▶ Use the coupled CPUIdle helper code

CPUIdle - R-Car SMP

Challenge #2:

- ▶ CPU0 always on by hardware design

Solution:

- ▶ Improve scheduler to prioritize tasks on CPU0

Outline

Hardware

Overview

Clocks and Voltage Control

Power Domains

Software

CPU Hotplug

CPUIdle

Future Work

Future Work

- ▶ CPU and I/O Power Domain control via deep CPUIdle states
- ▶ R-Car Coupled CPUIdle support and scheduler optimizations
- ▶ CPUIdle and CPU Hotplug sleep mode code sharing support