

Московский Физико-Технический Институт
Факультет Радиотехники и Кибернетики
Кафедра Информатики и Вычислительной Техники

Разработка программных средств управления частотой микропроцессора E1C+

Студент: Аблакотов М.А.
Научный руководитель: Кравцунов Е.М.

ЗАО МЦСТ, отделение ОС Эльбрус

Москва, 2016

Введение

- **Динамическое масштабирование частоты** - автоматическая подстройка частоты «на лету» в целях экономии потребляемой микропроцессором энергии либо уменьшения количества тепла, генерируемого чипом
- В процессорах E1C+ для управления частотой внедрён **модуль РМС** (Power Management Controller), поддержка которого в ядре Linux отсутствовала
- Одним из наиболее эффективных путей для уменьшения потребляемой микропроцессором мощности и количества выделяемого тепла в ядре Linux является **система управления частотой CPUFreq**

Задачи

Разработка программных средств динамического управления частотой процессора E1C+:

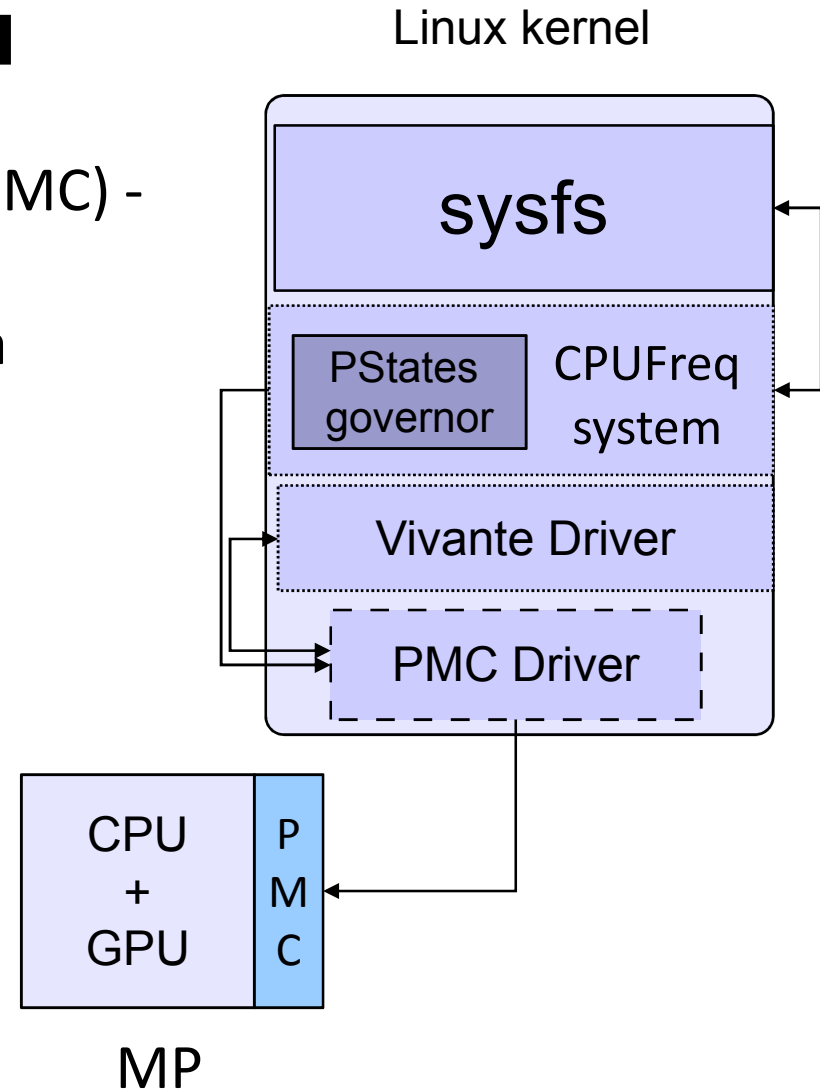
- **драйвер РМС** для контроля частоты процессора
- **governor** в составе системы CPUFreq для автономного динамического управления частотой процессора в зависимости от температуры

Общая схема взаимодействия систем управления

➤ Power Management Controller (PMC) - контроллер управления энергопотреблением процессора

➤ Используемые подсистемы ядра Linux

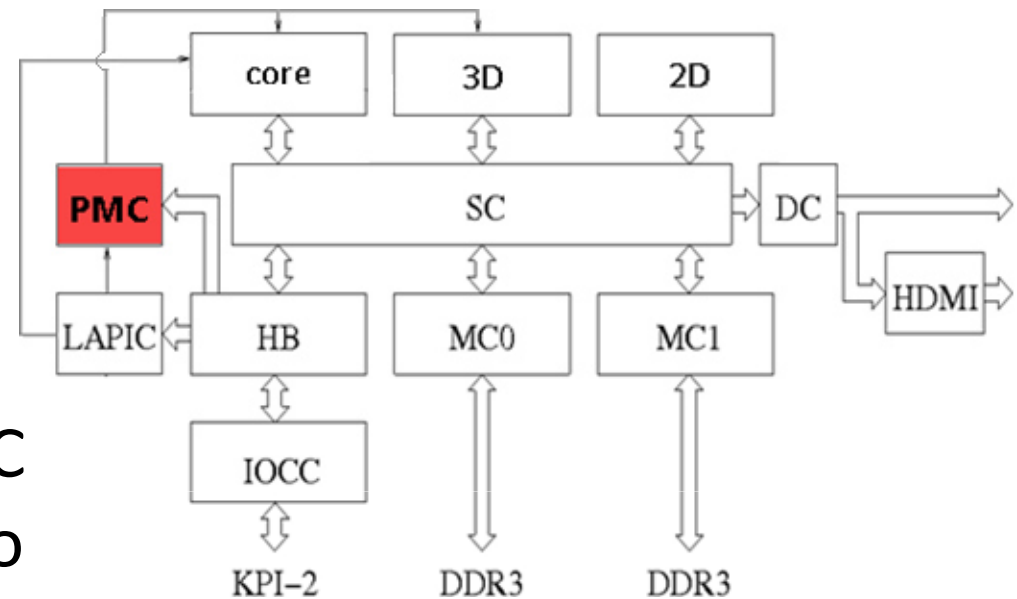
- Sysfs интерфейс
- Драйвер Vivante GPU
- Драйвер PMC
- *GovernorPStates* - программное средство динамического управления частотой процессора в составе системы CPUFreq



Power Manager Controller процессора E1C+

Модуль PMC
организует режимы
энергосбережения MP

Функционал модуля PMC
модуля вычислительного
ядра и модуля 3D графики:



Блок-схема MP E1C+

- Изменение **частоты** работы вычислительного ядра и модуля 3D графики
- Снятие показаний **температуры** вычислительного ядра и северного моста

P-состояния модуля вычислительного ядра и модуля 3D графики

P-состояние (Power State) – набор коэффициентов, задающих рабочую частоту модуля

Модуль	Коэффициент	Описание
Core	FID	Коэффициент частоты
3D	clk1x	Коэффициент делителя основной частоты
	clkSh	Коэффициент делителя шейдерных блоков

Состояние	FID	F (МГц)	clk1x	F(МГц)	clkSh	F(МГц)
P0	0xE0	984	0xE0	984	0xE0	984
P1	0x00	400	0x00	400	0x00	400
P2	0x20	200	0x20	200	0x20	200
P3	0x60	143	0x60	143	0x60	143

Регистры управления переходами между P-состояниями

4 регистра *P_State_(3D_)value_X* - управление соответствующими P-состояниями CPU – установка частоты и напряжения.

Регистр *P_State_(3D_)Ctrl* – выбор рабочего P-состояния CPU.

Регистры *P_State_value_X*

Номер бита	Тип бита	Название	Описание
7:0	RW	FID	Коэффициент FID

Регистры *P_State_3D_value_X*

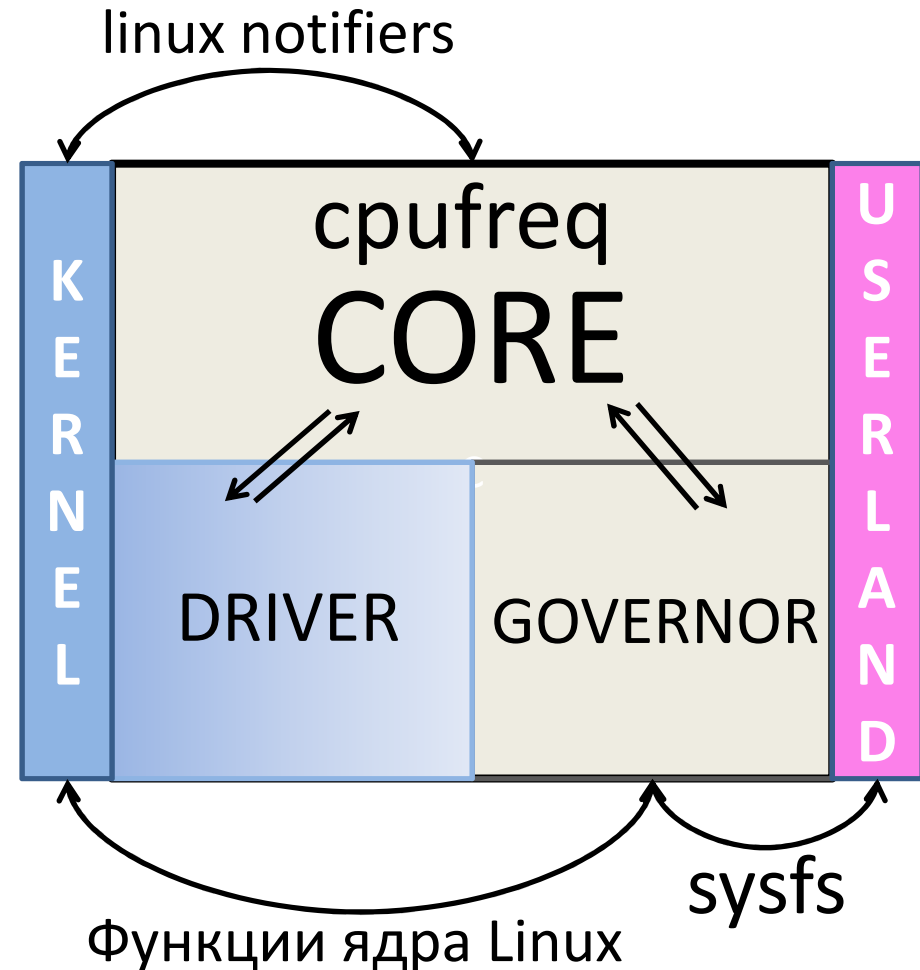
Номер бита	Тип бита	Название	Описание
15:8	RW	FID1	Коэффициент clk1x
7:0	RW	FID2	Коэффициент clkSh

Регистр *P_State_(3D_)Ctrl*

Номер бита	Тип бита	Название	Описание
1:0	RW	PNUM	Состояние: P0, P1, P2, P3

Система управления частотой процессора CPUFreq

- Core
 - координация работы остальных частей и уведомление ядра ОС об изменении частоты
- Driver
 - непосредственное изменение частоты
- Governor
 - правила установки частоты и её **выбор** согласно этим правилам



Реализованный драйвер PMS

- **Расчёт** таблицы аппаратно доступных частот CPU/GPU
- **Установка** требуемой частоты CPU/GPU:
 - Поиск ближайшей доступной частоты
 - Проверка наличия P-состояния с этой частотой
 - Добавление нового P-состояния при его отсутствии
 - Установка нового P-состояния
- **Установка** требуемого **состояния** (scale) GPU (для драйвера Vivante)
- Чтение текущего **состояния** (scale) GPU
- Чтение текущей **температуры** вычислительного ядра

Интерфейс реализованного драйвера РМС

- Userspace - sysfs

Доступные параметры:

- temp_cur0

чтение текущей **температуры** кристалла

- load_threshold

установка **границы увеличения частоты GPU**

- dynamic_gpu_frequency_limit_low

установка **нижней границы** частоты GPU

- dynamic_gpu_frequency_limit_high

установка **верхней границы** частоты GPU

Интерфейс реализованного драйвера РМС

- Kernel

- int pmc_get_temp_cur0()
 чтение текущей **температуры** кристалла
- int pmc_l_gpufreq_set_scale(unsigned char scale)
 установка требуемого состояния (scale) GPU
- int pmc_l_gpufreq_get_scale()
 чтение текущего состояние (scale) GPU
- int pmc_l_gpufreq_get_frequency()
 чтение текущей **частоты** GPU
- static struct cpufreq_driver pmc_l_cpufreq_driver = {
 .verify – указатель на верифицирующую функцию
 .target – указатель на функцию установки запрашиваемой частоты
 указанного ядра
 .init – указатель на функцию инициализации драйвера
 .exit - указатель на функцию извлечения драйвера
 .get – указатель на функцию, возвращающую текущую частоту
 указанного ядра
 .name – имя драйвера
}

Анализ потребления мощности процессором

Формула расчета потребляемой процессором мощности

$$P_{tot} = (aU^2 F_{clock}) + (bUF_{clock}) + AUT^2 e^{\frac{-B}{T}}$$

Таблица рассчитанных Р-состояний модуля вычислительного ядра при заданных температуре и ограничению по мощности

$P_{tot} \setminus T^{\circ}C$	50	60	70	80	90	100	110	120	130
5.1 Вт	p0	p0	p0	p0	p0	p0	p0	p5	p10
4.85 Вт	p0	p0	p0	p0	p0	p0	p2	p9	p15
4.6 Вт	p0	p0	p0	p0	p0	p0	p7	p13	p18
4.35 Вт	p0	p0	p0	p0	p0	p6	p11	p17	p21
4.1 Вт	p0	p0	p0	p0	p4	p10	p17	p21	halt
3.85 Вт	p0	p0	p0	p3	p9	p16	p21	halt	halt
3.6 Вт	p0	p0	p1	p8	p15	p20	halt	halt	halt
3.35 Вт	p0	p0	p7	p14	p19	halt	halt	halt	halt
3.10 Вт	p0	p6	p13	p19	halt	halt	halt	halt	halt
2.85 Вт	p6	p12	p19	halt	halt	halt	halt	halt	halt
2.6 Вт	p13	p19	halt	halt	halt	halt	halt	halt	halt
2.35 Вт	p20	halt	halt	halt	halt	halt	halt	halt	halt
2.1 Вт	halt	halt	halt	halt	halt	halt	halt	halt	halt

Таблица доступных Р-состояний модуля вычислительного ядра

Состояние	F (МГц)	U (мВ)
P0	984	999
P1	914	999
P2	800	999
P3	711	999
P4	512	999
P5	400	999
P6	200	999

Реализованный PStates governor

- Задачи
 - Повысить **стабильность** работы процессора при **высоких температурах**
 - **Понизить энергопотребление** процессора
- Функционал
 - Работает **автономно** от пользователя
 - Позволяет устанавливать **ограничение по потребляемой мощности**
 - Детектирует **переключение** с источника питания на батарею и обратно
 - **Выбирает** максимальную доступную **частоту** в зависимости от лимита мощности и температуры

Реализованный PStates governor.

Интерфейс

- `sysfs (/sys/devices/system/cpu/cpufreq/pstates/)`

Доступные параметры:

- `sampling_rate` – **период** обновления частоты CPU
 - `sampling_rate_min` – **минимальный период** обновления частоты CPU
 - `power_consumption_limit` – максимально потребляемая CPU **мощность** в мВт
 - `temperature` – **температура** кристалла
- `boot options`
 - `init_cpu_pwr_limit` - максимально потребляемая CPU **мощность от источника** в мВт
 - `battery_pwr` - максимально потребляемая CPU **мощность от батареи** в мВт

Результаты работы

- **разработан и внедрен драйвер PMS** для поддержки управления частотой CPU и GPU для процессоров E1C+ в ядре Linux
- **разработан и внедрен governor Pstates** - система динамического управления частотой CPU в составе системы CPUFreq в ядре Linux

Дальнейшее развитие

- Дополнить текущую систему (Pstates governor) возможностью динамического перевода ядер процессоров в состояние сна в мультипроцессорных системах

Спасибо за внимание!

