Kernel Features for Reducing Power Consumption on Embedded Devices

Krzysztof Kozlowski

k.kozlowski@samsung.com

Samsung R&D Institute Poland

SAMSUNG

Agenda

- Overview and goals
- CPUfreq (with clock down)
- Run-time PM and power domains
- SoC low power states
- CPU idle drivers
- Devfreq
- Summary

SAMSUNG

Overview and goals

- Limit the consumption of energy by mobile device
- Do not hurt performance (at least to some extend)
- Target devices: smartphones, tablets, wearables
  - Mobile devices are somehow different
    - Energy consumption is an important factor (How often do you have to charge your phone?)
    - Mobile device is mostly idle (more opportunities to sleep)
- The speech will focus on ARM architecture and Samsung's Exynos System-on-Chip family, although ideas are not limited to Exynos

SAMSUNG

Overview and goals (2)

- Mainline Linux kernel is changing very fast
  - Some details about specific kernel drivers may become obsolete soon
  - Details as for current mainline kernel: 3.16 and linux-next (from August)

Disclaimer

- All measurements were done on development devices:
  - Custom kernels
  - Custom operating systems
- They are not representative for market/end products
- Sometimes measurements on these devices may not be even close to market products
- Many measurements were done in specific custom configuration, very different from market product, e.g.:
  - No CPU idle driver, no CPUfreq driver
  - Booted to init=/bin/sh

SAMSUNG

Environment for tests

- Measured on:
  - Trats2, smartphone (Exynos 4412, 4 cores, freq 200-1400 MHz)
  - Gear1-like wearable (Exynos 4212, 2 cores, 200-1400 MHz)
  - Exynos 3250 development board (2 cores, 100-1000 MHz)
- Kernel used:
  - Exynos 4212 and 4412 : linux-next (next-20140804)
  - Exynos 3250: internal Linux kernel tree 3.10
    - Lack of full support in mainline

SAMSUNG

CPUfreq

- Ondemand governor adjusts the frequency and voltage to current load
- Specific conditions to embedded world – one frequency and voltage for whole cluster
  - On most SoCs: one clock frequency/voltage for all CPUs
  - Except big.LITTLE (e.g. 2 clusters on quad-core Exynos Octa)
- Dual cluster SoCs have big.LITTLE CPUfreq driver

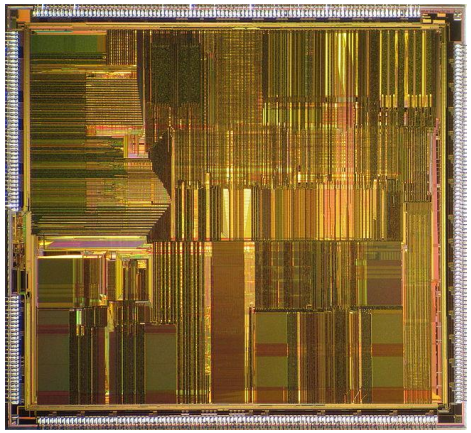


Photo by Pauli Rautakorpi

SAMSUNG

CPUfreq on SoCs with one cluster

- Separate CPUfreq drivers for each SoC
- Moving toward one generic cpufreq-cpu0 driver
- cpufreq-cpu0 requires:
  - Clock to operate on (provided by clock driver)
  - Optionally voltage regulator (provided by regulator driver)
  - Table of Operating Performance Points from Device Tree
    - OPP is a tuple of frequency and voltage
- Clock/regulator/OPP frameworks add necessary abstraction layer and make cpufreq-cpu0 a generic solution

ARMCLK clock down feature

- Reduces the frequency upon entering WFI or WFE instruction (Wait for Interrupt/Event)
  - All cores must be idle
- Behaves like a hardware ondemand CPUfreq governor
  - But only for clock frequency (voltages remain untouched)
- Supported by most of Exynos SoCs
  - As part of clock driver for Exynos 3250, Exynos 4 and Exynos 5250 [1]

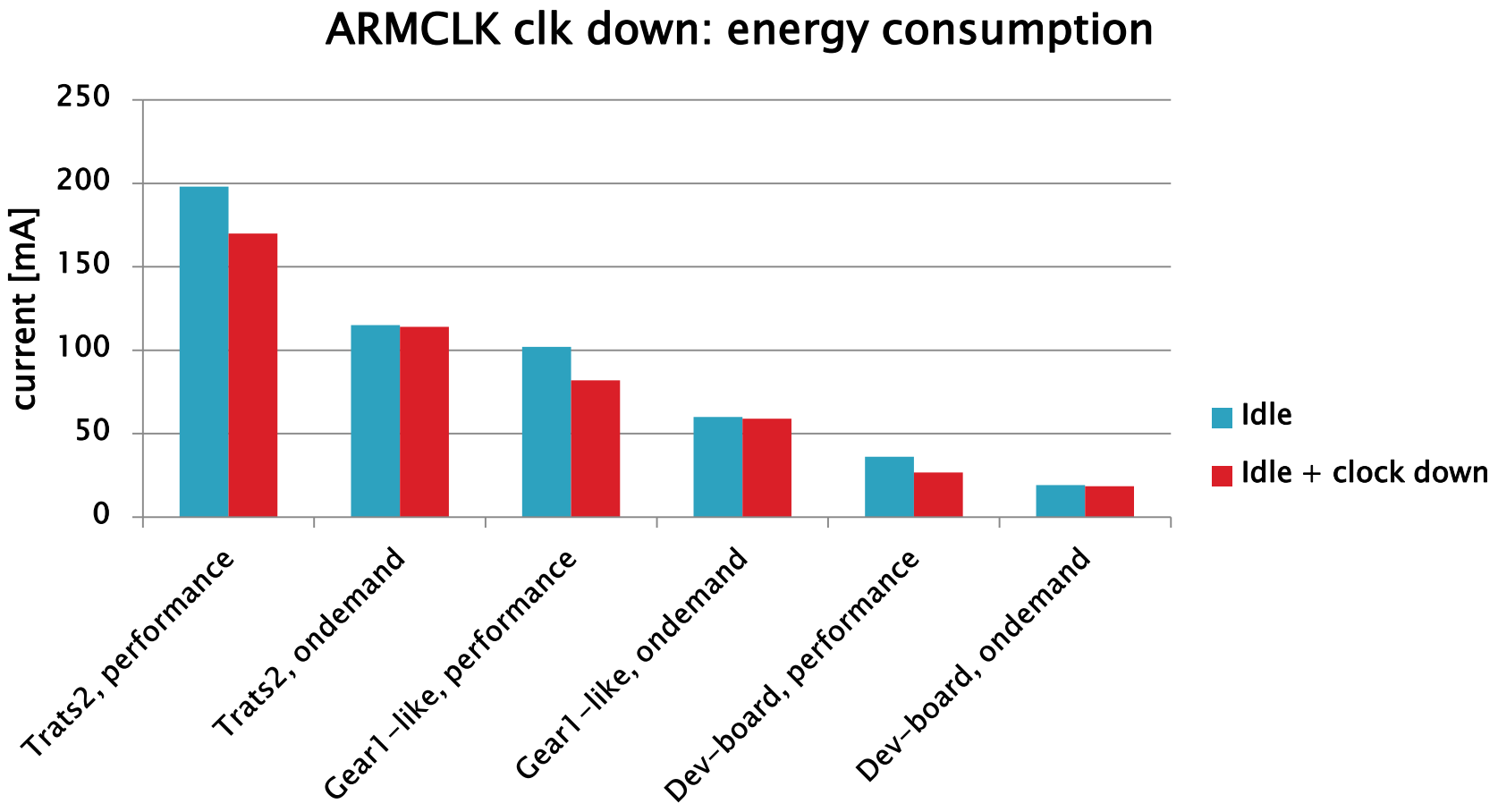
ARMCLK clk down: energy consumption

Board	SoC	Frequency [MHz]	Idle [mA]	Idle + clock down [mA]
Trats2	Exynos 4412	1400	198	170
		200	115	114
Gear1-like	Exynos 4212	1400	102	82
		200	60	59
Dev-board	Exynos 3250	1000	36.2	26.7
		100	19.2	18.5

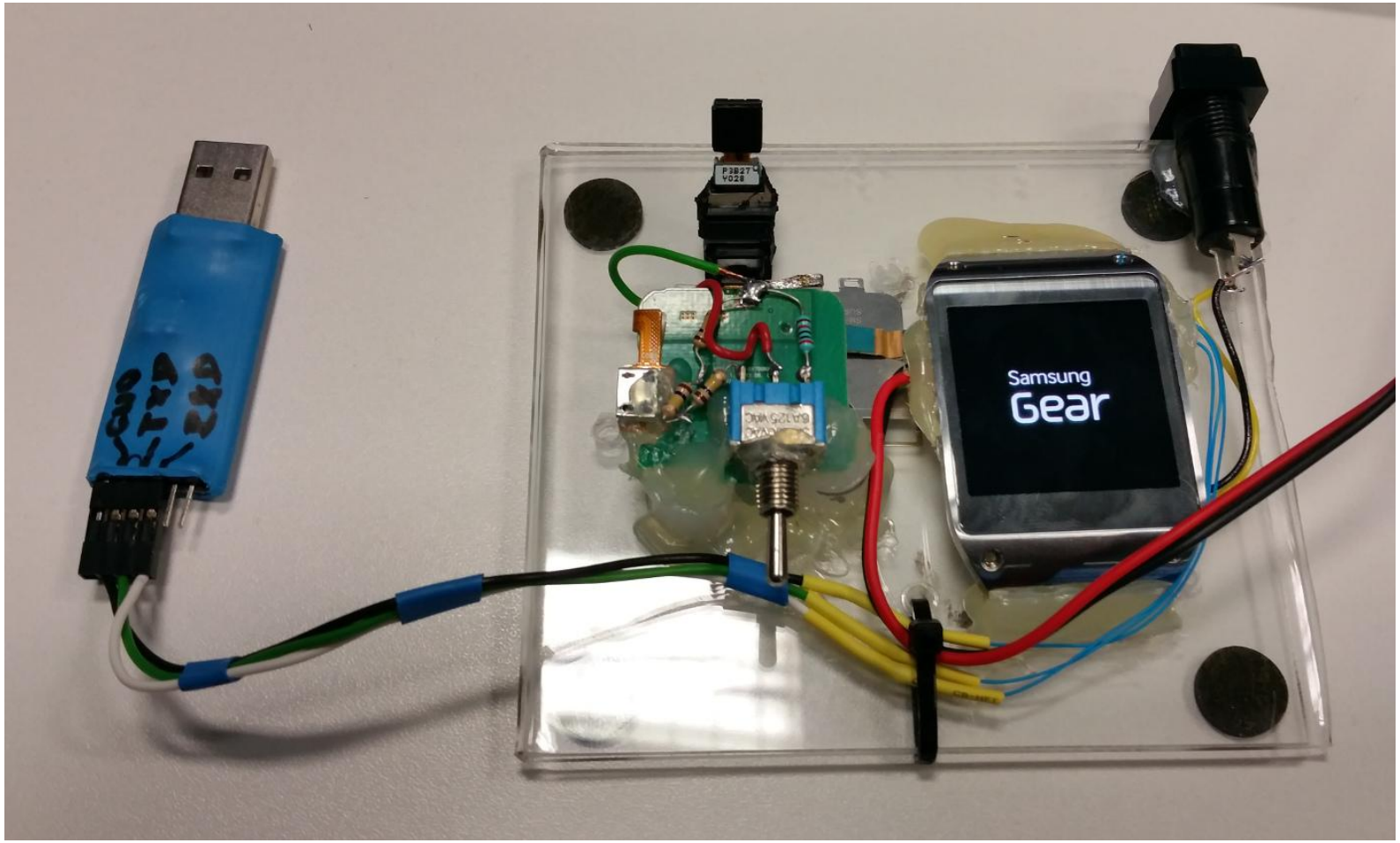
- Measurements in idle mode (basic CPU idle – WFI), no load
- No benefits if CPUfreq governor is ondemand which is quite obvious
  - ARMCLK cannot get below minimal frequency used in CPUfreq driver
- Ondemand CPUfreq reduces also ARM voltage

SAMSUNG

ARMCLK clk down: energy consump. (2)



Wearable... for developers!



SAMSUNG

Run-time PM

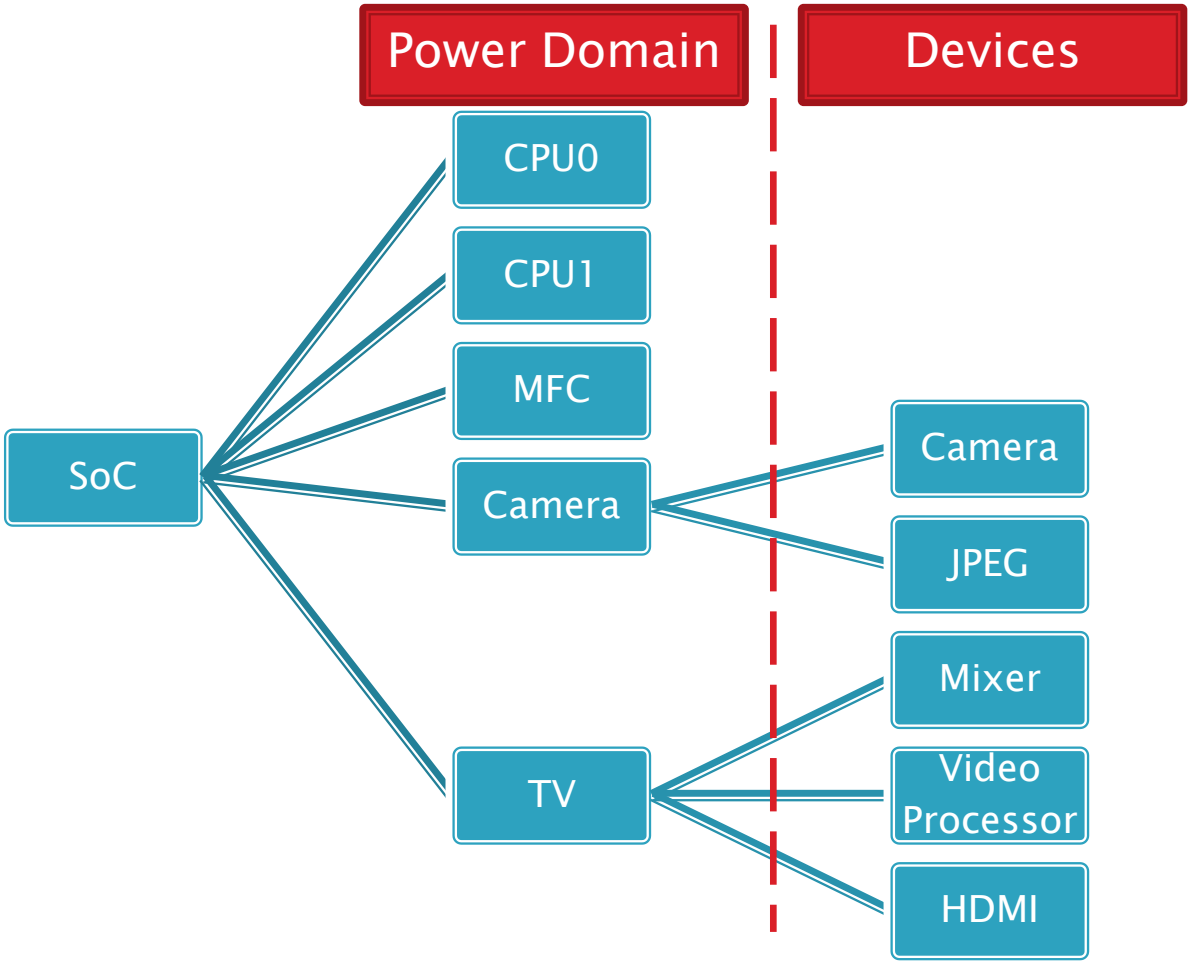
- Putting devices into low power states when not used
  - Optionally: automatically delayed suspends
- Needs support in device drivers
  - Driver specifies when it is working and idle
    - Usage counter (pm_runtime_get()/put())
    - Time of last activities (pm_runtime_mark_last_busy())
  - Driver implements runtime suspend and resume callbacks

Power domains

- Local power control, powered independently
- Example power domains:
  - CPU-s
  - Multi-Format Codec (MFC)
  - G3D (e.g. Mali)
  - LCD
  - Image Signal Processor
  - Camera
- Linux offers a generic power domain framework [2]
  - Used by SH-Mobile and Exynos
- Other vendors implement this on their own

Generic power domains

- Multiple Linux devices can be attached to a domain



SAMSUNG

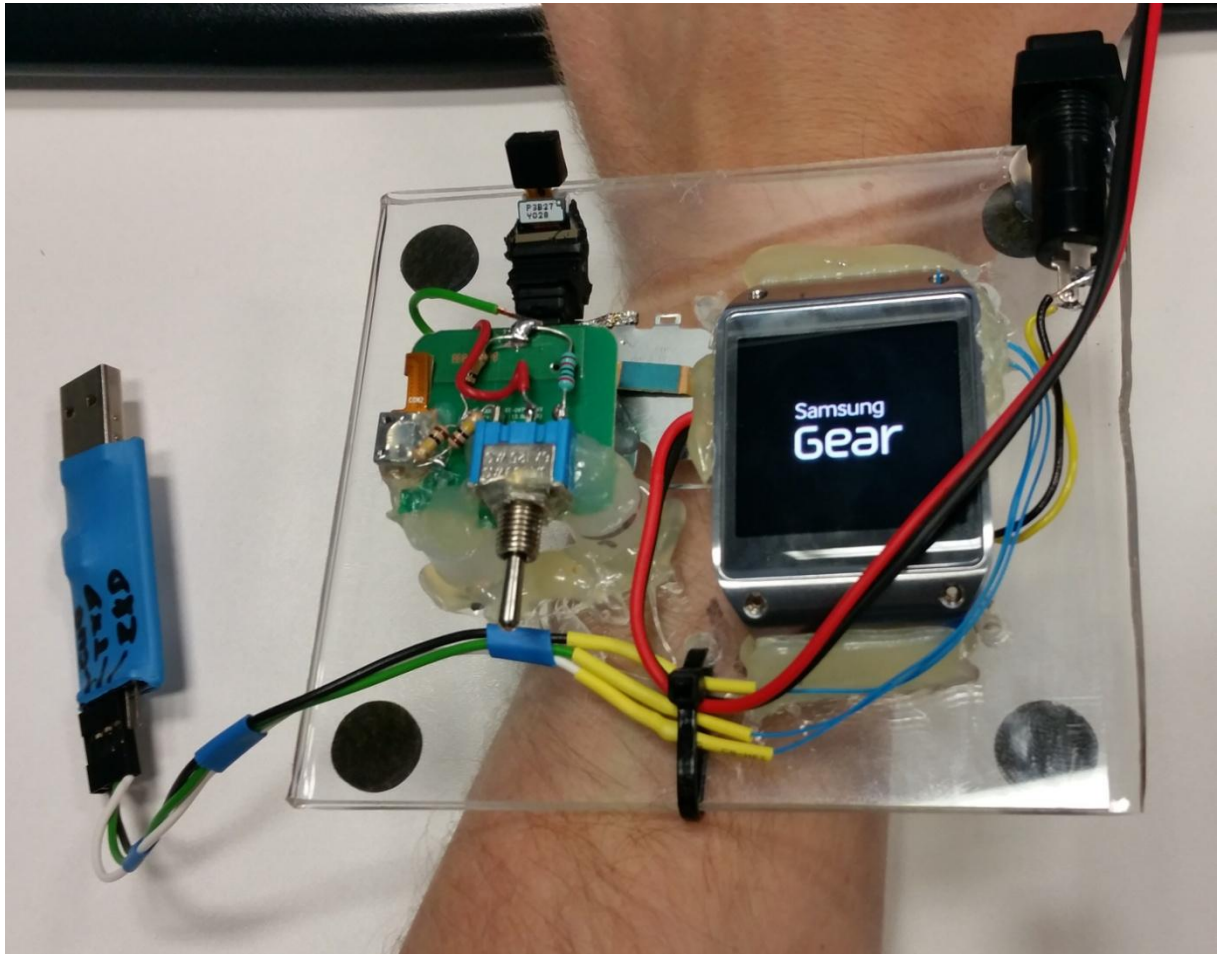
Generic power domains (2)

- Integrates with runtime PM
  - If all attached devices have been suspended, power down whole domain
- Differences in energy consumption:

Board	SoC	All domains enabled permanently [mA]	Power domain runtime PM [mA]	Diff [%]
Trats2	Exynos 4412	124	114	-8%
Dev-board	Exynos 3250	24.7	18.5	-25%

- Measurements in idle mode (basic CPU idle - WFI), no load

It is truly a wearable!



SAMSUNG

SoC low power states

- When in idle, enter low power state to save energy
- Wait for Interrupt (WFI), basic idle state
- Various SoCs support deeper low power states
  - Msm: retention, standalone power collapse, power collapse
  - Exynos: ARM Off TOP Running (AFTR), W-AFTR, Low Power Audio playback (LPA)
    - W-AFTR and LPA extend the AFTR low power mode
- Usually they have higher latency for enter and leave
  - Enter deeper state only if we won't be awaken right away

SoC low power states on Exynos

- System-level states
  - Whole system must be prepared
    1. CPU[1-n] must be powered off
    2. Then CPU0 triggers entering deep low power state
- ARM Off TOP Running (AFTR)
  - Cortex core is power gated (power supplied but internally gated)
  - Most of other modules are powered on (e.g. Audio, MMC, CoreSight, USB, I2C, UART etc.)

SoC low power states on Exynos (2)

- W-AFTR (on Exynos 3250)
  - AFTR + power gating everything that can be power gated
  - Except Dynamic Memory Controller, DDR, RTC
  - Fast wake-up
- Low Power Audio playback (LPA, on other Exynos SoCs)
  - AFTR + power gating everything that can be power gated
  - Contents of L2 cache and TOP modules are preserved (retention)
  - Audio related blocks are on

SoC low power states on Exynos (3)

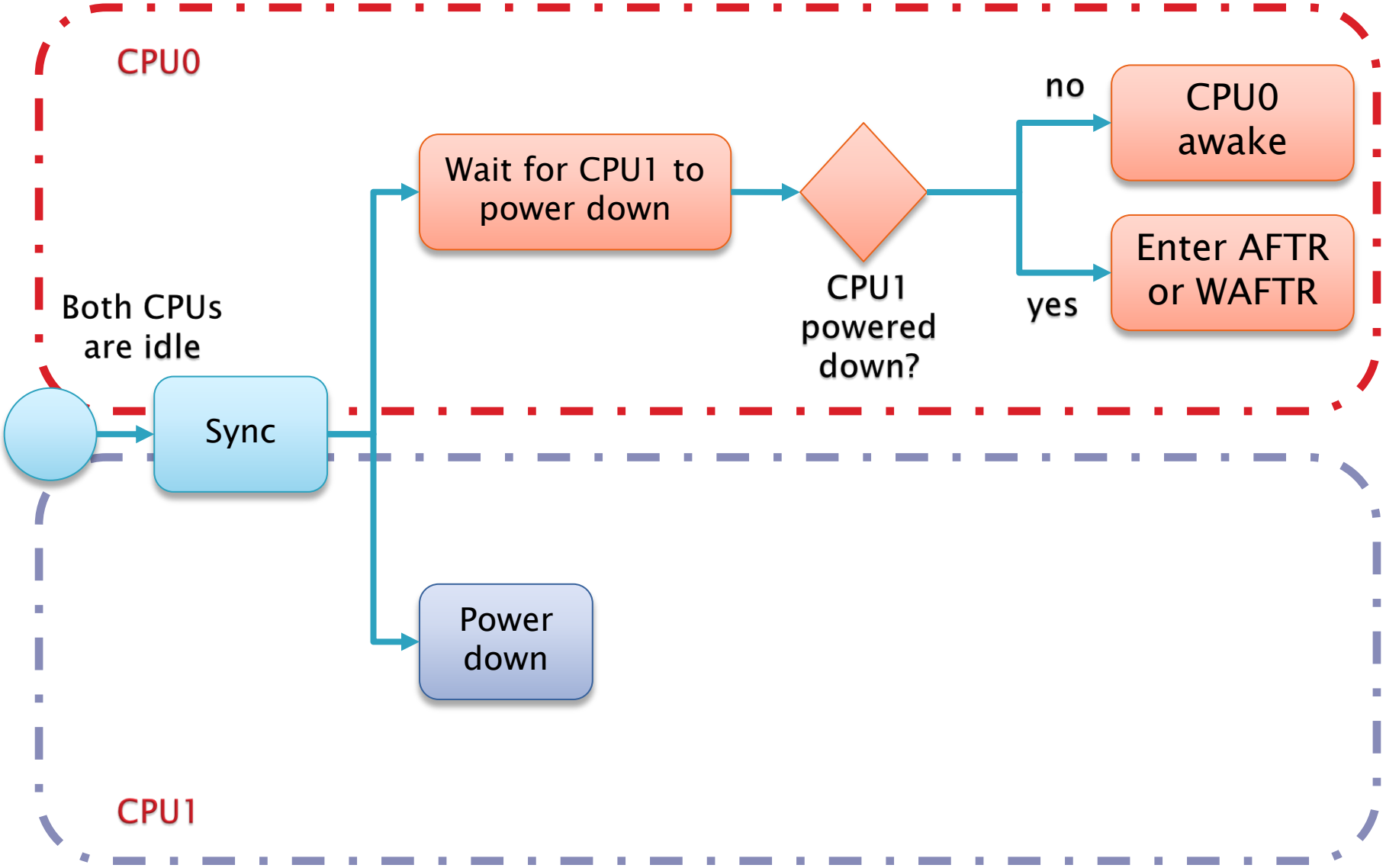
- Sleep
  - Power not supplied (regulators turned off)
  - Only ALIVE and RTC blocks are on
  - Entered with Suspend-to-RAM

Low power state	Subsystem	Requires additional drivers
WFI	Scheduler (idle loop)	No, built-in
AFTR, W-AFTR, LPA	CPU idle	CPU idle driver and support for board (arch/arm/mach-*)
Sleep	Suspend	support for board (arch/arm/mach-*)

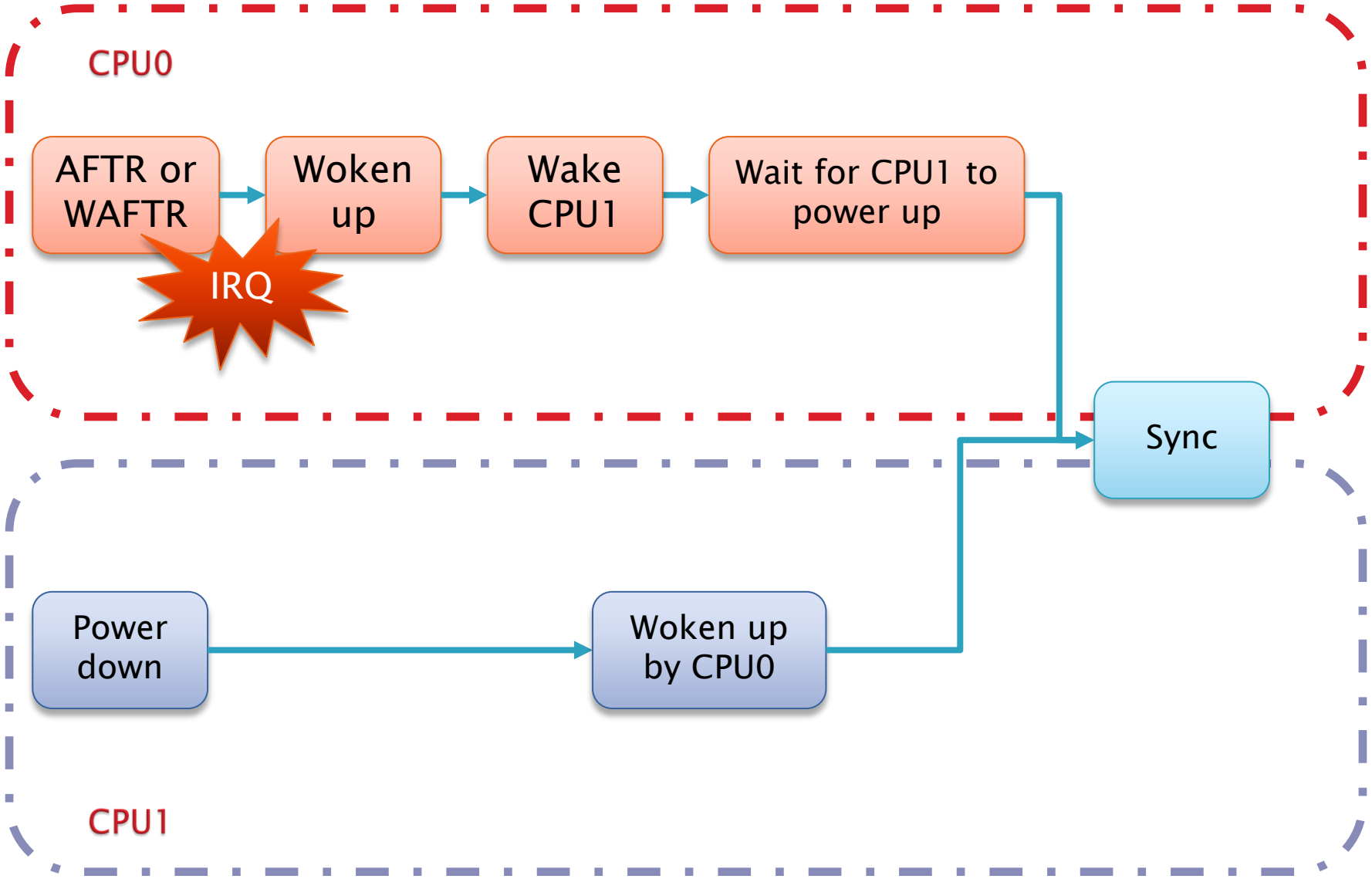
Entering deep low power states

- From CPU idle perspective: states are „coupled”
  - Entering AFTR/W-AFTR/LPA depends on powering down all other CPUs[1-n]
  - If only CPU0 is left on, then CPU idle triggers low power state
- User space may hot unplug idle CPUs[1-n]
  - echo 0 > /sys/bus/cpu/devices/cpu1/online
  - E.g. mpdecision daemon for msm
- Kernel may synchronize entering idle states
  - Coupled CPU idle drivers
- echo mem > /sys/power/state

Coupled CPU idle driver - entering



Coupled CPU idle driver - leaving



CPU idle drivers in Linux

- big.LITTLE
  - Suspend whole cluster
  - Supported Versatile TC2 (Linux 3.16) and Exynos 5420 (next)
- Exynos AFTR
  - Currently only for Exynos 4210 and Exynos 5250
  - Requires manual hot unplug of CPU1 to work
- Coupled CPU idle driver for Exynos (CPU1 off, CPU0 AFTR)
  - On going work, patch by Daniel Lezcano [3]

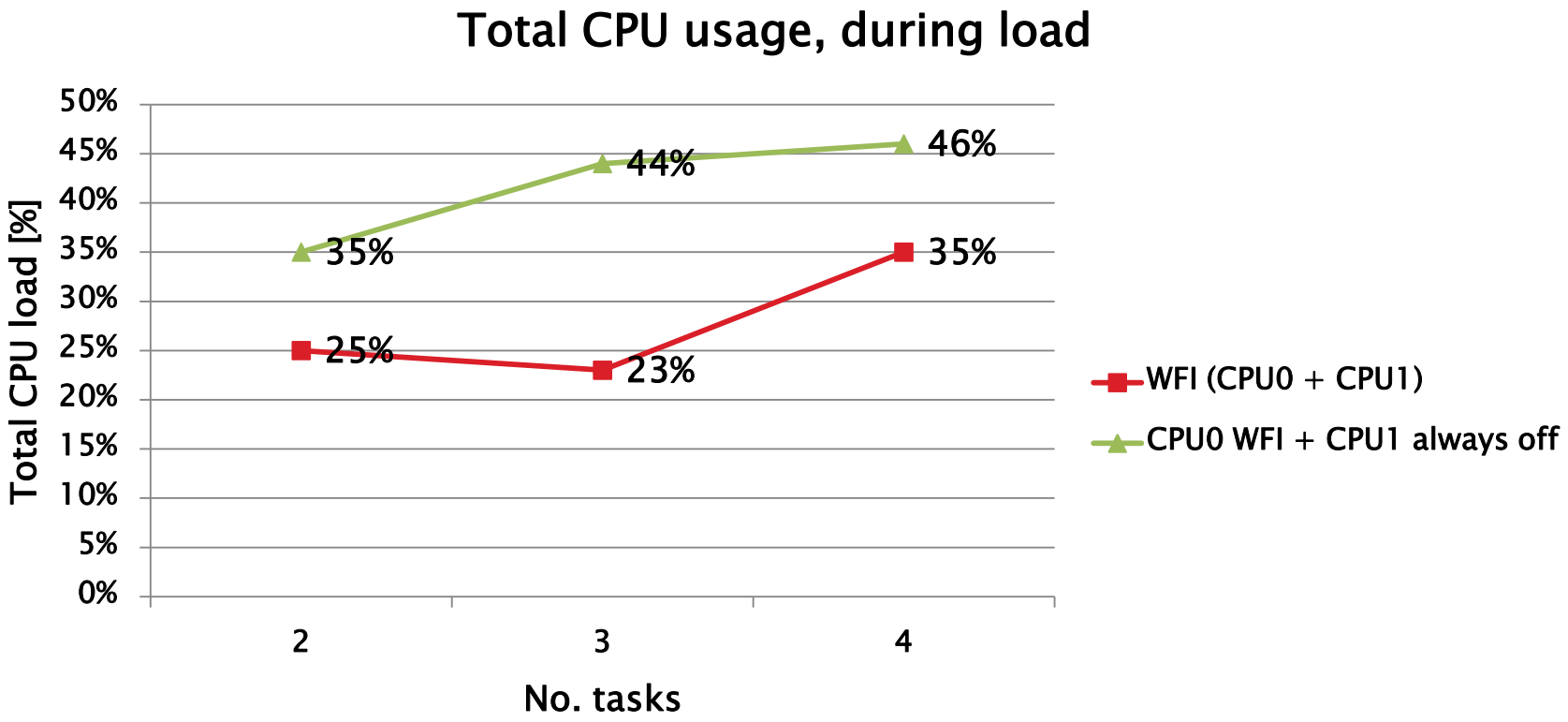
CPU idle measurements

- Due to lack of support in mainline all measurements were done on internal Linux kernel tree 3.10
- Development board with Exynos 3250 (2 cores, 100-1000 MHz)
- The goal of measurements is to show overall differences between SoC low power states
- Workload: simple computation tasks (work-sleep-work), mainly used to test scheduler

CPU idle measurements (2)

- Tested configurations:
  - CPUs enter only WFI (no CPU idle driver)
  - Power down CPU1 from userspace permanently, CPU0 enters WFI (no CPU idle driver)
  - Coupled CPU idle: CPU1 power off, CPU0 AFTR
  - Coupled CPU idle: CPU1 power off, CPU0 AFTR or W-AFTR
    - W-AFTR is entered if certain conditions are met. If no, enter AFTR.
    - Conditions:
      - Camera, G3D, MFC power domains are off
      - Certain bus clocks are gated
      - MMC is idle

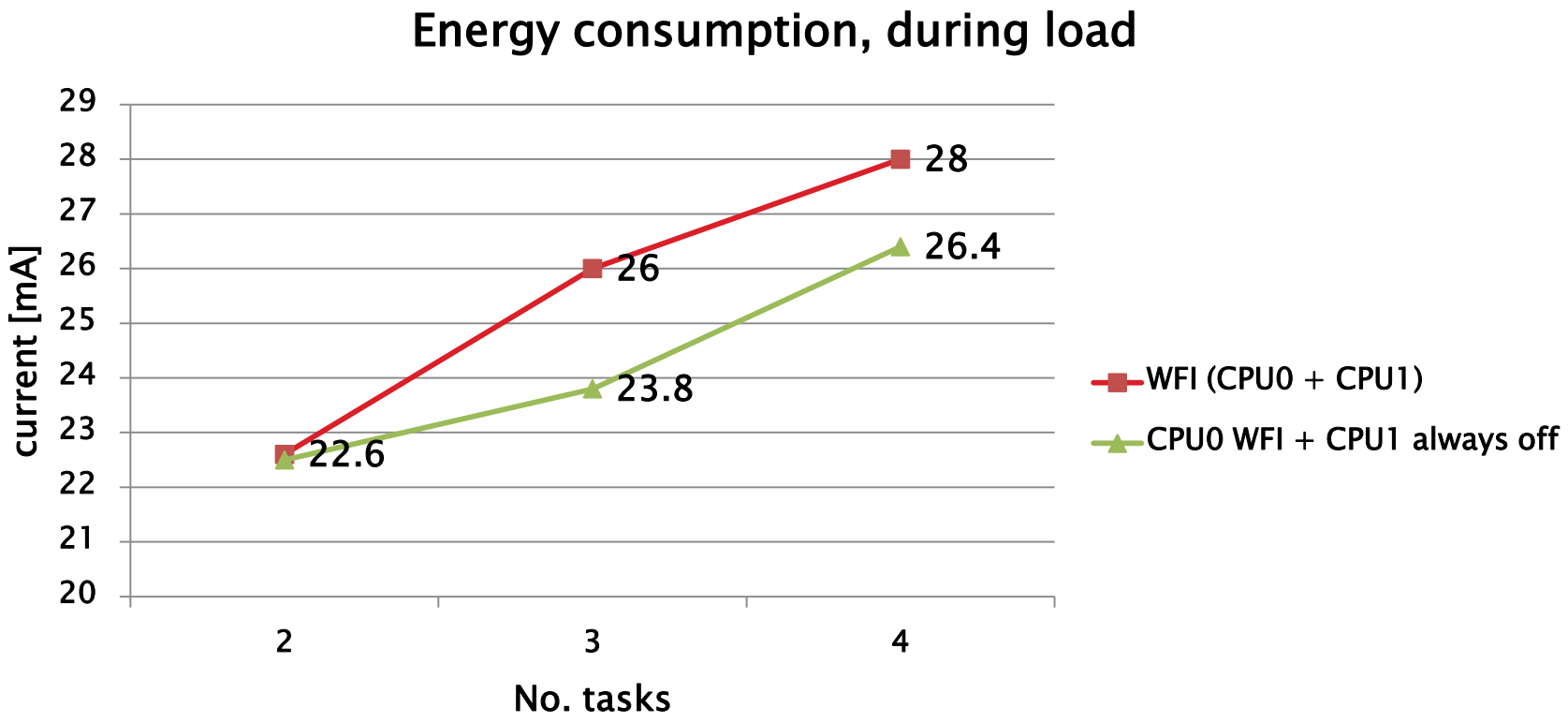
CPU idle: energy consumption, load



CPU frequency	2 tasks	3 tasks	4 tasks
WFI (CPU0 + CPU1)	200 MHz	200–600 MHz	200–600 MHz
CPU1 powered off	500 MHz	300–900 MHz	400–1000 MHz

SAMSUNG

CPU idle: energy consumption, load (2)



CPU frequency	2 tasks	3 tasks	4 tasks
WFI (CPU0 + CPU1)	200 MHz	200–600 MHz	200–600 MHz
CPU1 powered off	500 MHz	300–900 MHz	400–1000 MHz

SAMSUNG

CPU idle: energy consumption, load (3)

- What about coupled CPU idle drivers (CPU1 off, CPU0 AFTR or W-AFTR)?
  - No differences because the load prevented entering deeper idle states
- But a new driver which powers off CPU1 also when CPU0 is busy makes sense
  - Workload consolidation in scheduler could also help (increased idle time of CPU1)
  - Already discussed and some works are in progress [4]
    - Power-aware scheduling [5]
    - Sched packing [6]
    - Workload consolidation and CPU ConCurrency [7]

SAMSUNG

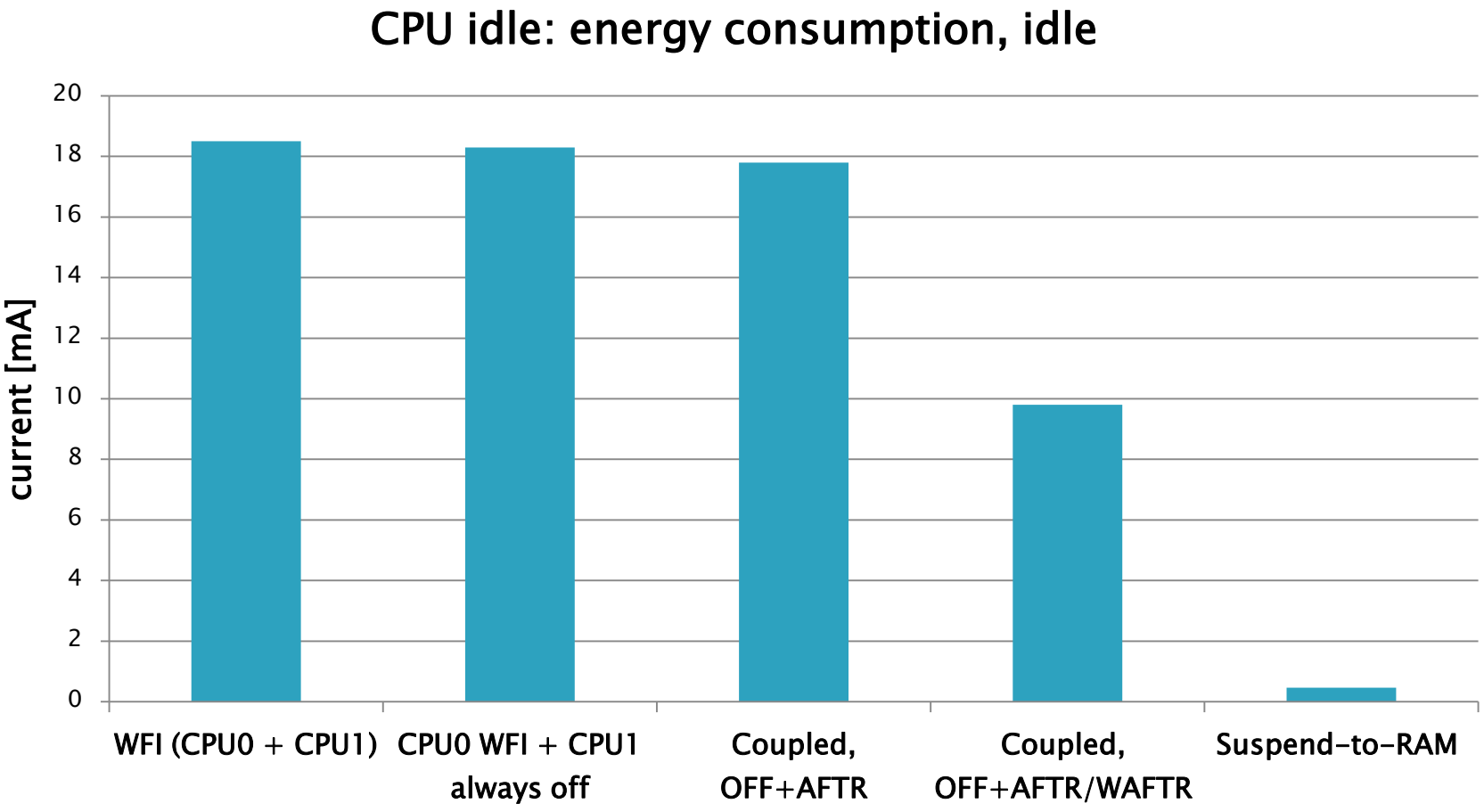
CPU idle: energy consumption, idle

CPU idle driver	Idle [mA]	Diff against WFI
WFI (CPU0 + CPU1)	18.5	
CPU0 WFI + CPU1 always off	18.3	-1.0%
CPU idle, coupled, OFF+AFTR	17.8	-3.8%
CPU idle, coupled, OFF+AFTR/WAFTR	9.8	-47.0%
Suspend-to-RAM	0.46	

- No load
- Exynos 3250 CPU @100 MHz

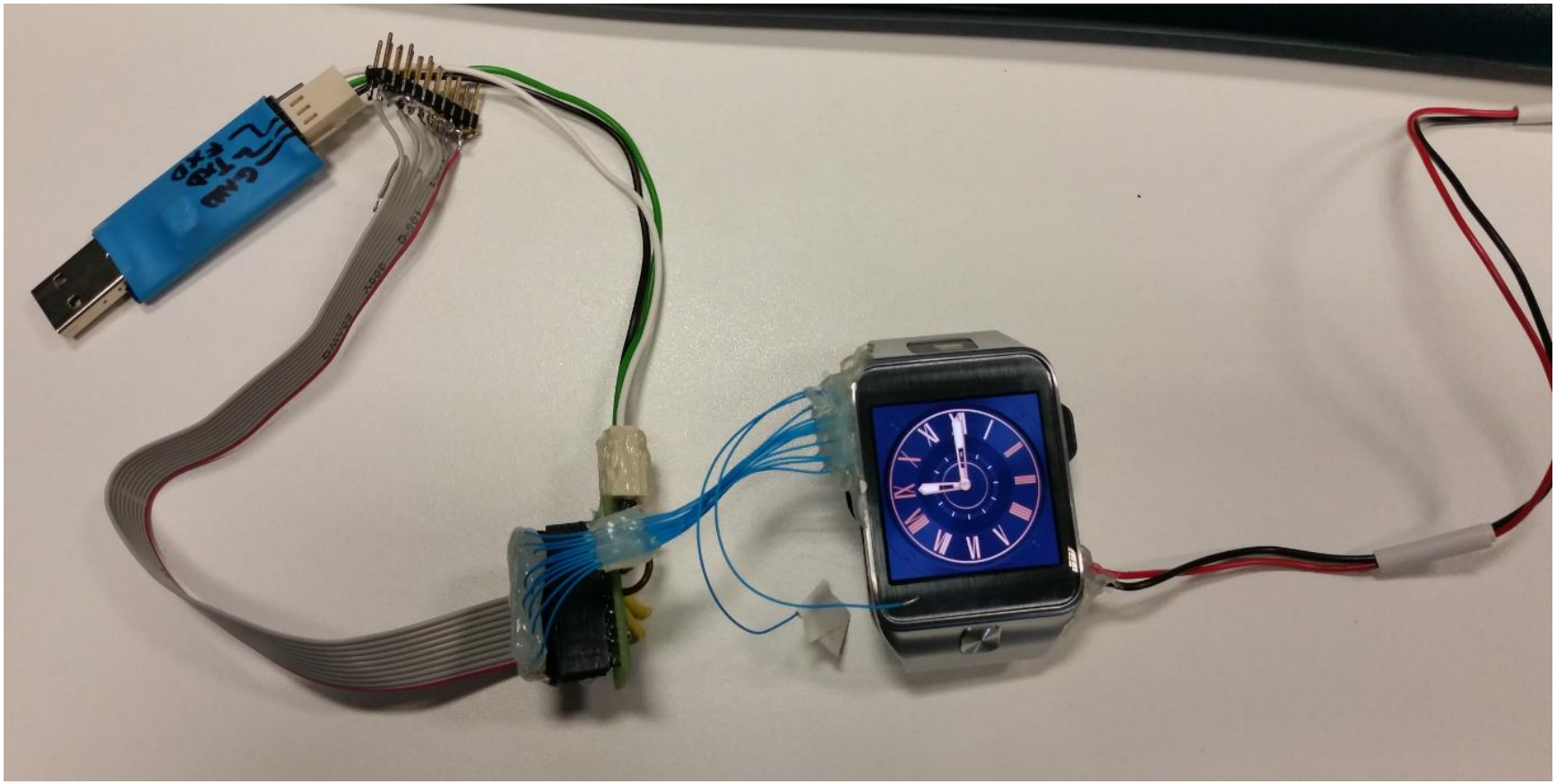
SAMSUNG

CPU idle: energy consumption, idle (2)



SAMSUNG

Improvements in development board...



SAMSUNG

Devfreq

- Dynamic Voltage and Frequency Scaling for devices
  - e.g. Memory bus, MFC, G2D, ISP, SD/MMC
- Governors similar to CPUfreq (ondemand etc.)
  - Current load may be taken from device performance counters
    - Exynos: number of read and write operations in Dynamic Memory Controller
- Mainline: only drivers for Exynos 4 and Exynos 5250
  - ... but only for Exynos 5250 is working

Devfreq: energy consumption, idle

- Due to lack of support in mainline all measurements were done on internal Linux kernel tree 3.10
- Development board with Exynos 3250 (2 cores, 100-1000 MHz)
- The goal of measurements is to show overall difference that devfreq makes

Devfreq: energy consumption, idle (2)

- No devfreq means that highest voltages and clock rates are chosen for:
  - bus clocks
  - INT and MIF regulators supplying power to many SoC modules

Possible values for Exynos 3250 development board	Min	Max
INT voltage	0.80 V	1.00 V
MIF voltage	0.85 V	1.00 V
DMC clock	50 MHz	400 MHz
LEFT/RIGHT buses clock	50 MHz	133 MHz

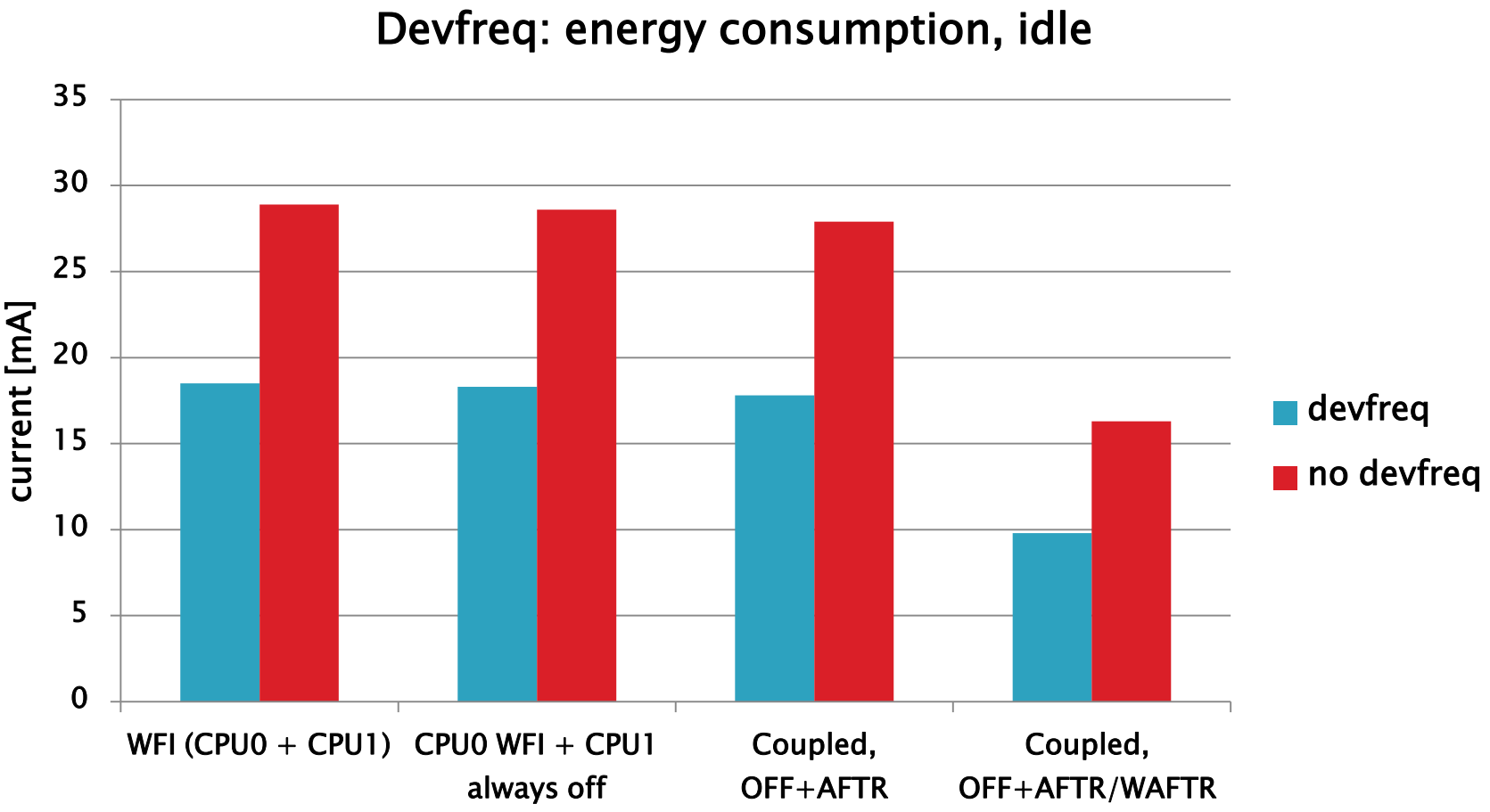
Devfreq: energy consumption, idle (3)

CPU idle driver	No devfreq [mA]	With devfreq [mA]	Diff [%]
WFI (CPU0 + CPU1)	28.9	18.5	-36%
CPU0 WFI + CPU1 always off	28.6	18.3	-36%
CPU idle, coupled, OFF+AFTR	27.9	17.8	-36%
CPU idle, coupled, OFF+AFTR/WAFTR	16.3	9.8	-40%

- No load
- Exynos 3250 CPU @100 MHz

SAMSUNG

Devfreq: energy consumption, idle (4)



Summary

- Linux kernel has a number of features for reducing energy usage
  - Unfortunately many drivers are not yet present in mainline
- Most benefits in reducing energy consumption come with:
  - Frequency and voltage scaling (CPUfreq, devfreq)
  - Entering low power system states (CPU idle)
  - Developing good drivers
    - For external resources use abstraction provided by the kernel (e.g. Common Clock Framework, regulators)
    - Runtime PM (if it is applicable)

Summary (2)

- Still a lot of work to do
  - Drivers: CPU idle, devfreq
  - Integration between scheduler, CPUfreq and CPU idle

Q&A

Thank you!

Any questions?

SAMSUNG

References

- [1] Common Clock Framework drivers for Exynos 4, 3250 and 5250
  drivers/clk/samsung/clk-exynos*.c
- [2] Generic Device Power Domains
  include/linux/om_domain.h
- [3] Coupled cpuidle driver for Exynos
  Exynos4: cpuidle: support dual CPUs with AFTR state
  http://thread.gmane.org/gmane.linux.power-management.general/44248
- [4] Toward a more power-efficient scheduler, Jonathan Corbet
  http://lwn.net/Articles/546664/

SAMSUNG

References (2)

- [5] Morten Rasmussen
  - Power-aware scheduling v2
    https://lkml.org/lkml/2013/10/11/547
  - sched: Energy cost model for energy-aware scheduling
    https://lkml.org/lkml/2014/5/23/621
- [6] Sched: packing tasks (and newer works), Vincent Guittot
  https://lkml.org/lkml/2013/10/18/121
- [7] A new CPU load metric for power-efficient scheduler: CPU ConCurrency, Yuyang Du
  https://lkml.org/lkml/2014/6/25/162