Linux Power Management Optimization on the Nvidia Jetson Platform

Merlin Friesen
Linux Power Management Optimization on Nvidia Jetson

About You – Target Audience

- The presentation is introductory

- It is intended for any one interested in:
  - Embedded systems
  - System on Chip (SoC) Architecture
  - Linux / ARM power management on the Nvidia Jetson platform
Linux Power Management Optimization on Nvidia Jetson

About Me

- Merlin Friesen

- I have worked for a number of semiconductor companies
  - All developing chips for the cellular / tablet space

- I have lead teams in:
  - Chip validation
    - Pre and Post Silicon
  - System software development

- I currently work with the Nvidia Tegra in a robotics application
Linux Power Management Optimization on Nvidia Jetson

Outline

Overview of the Jetson Platform

Overview of the Tegra K1 System on Chip (SoC)

SoC Power Management
- Power Management Unit (PMU)
- Power islands
- Dynamic Voltage and Frequency Scaling (DVFS)
- Auto clock gating

System Software
- ARM cores
  - cpufreq
  - cpuidle
- Device drivers
  - Power management interfaces

Data Driven Power Management Techniques
Linux Power Management Optimization on Nvidia Jetson

Jetson Platform

SD/MMC

Tegra K1

DDR3 2GB

12V Power

PMU

HDMI

Ethernet

Flash 16Gb
Linux Power Management Optimization on Nvidia Jetson

Jetson Platform

Jetson ships with Ubuntu pre-installed
  - Compilation tools are pre-installed
    - No cross compilation environment is required
  - With the ethernet port enabled it is very easy to customize Ubuntu
    - sudo apt-get install <package>
  - Or you can use your preferred ARM based Linux kernel
Linux Power Management Optimization on Nvidia Jetson

Jetson Platform

Jetson is a very high end embedded platform
  - Compare to other popular embedded platforms
    - Raspberry Pi
      - ARM 1176 single core at 700Mhz
    - Pi 2
      - Cortex A7 * 4 at 900Mhz
    - Beaglebone Black
      - ARM Cortex A8 single core at 1Ghz
    - Arduino
Linux Power Management Optimization on Nvidia Jetson

Tegra K1 System on Chip (SoC)

- The Jetson platform is built around the Tegra K1 chip
  - 28nm process
  - 15mm^2
  - ARM A15 x 4 with a 5th A15 in a Low Power CPU complex
    - Maximum frequency 2.3 GHz
- GPU
  - 192 CUDA cores
  - OpenGL 4.4
    - OpenGL ES 3.1
- 4K HDMI
Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

Tegra K1 System on Chip (SoC)

- Highly integrated cores like this are driving the mobile phone and tablet markets

- The K1 is in a similar class of mobile devices from:
  - Broadcom
  - MediaTek
  - Qualcomm
  - Samsung

- Given their use in mobile handsets and tablets these devices have state of the art semiconductor power management
Linux Power Management Optimization on Nvidia Jetson

Jetson Platform

It is finding use in high end applications
- Drones
- Vision
- Robotics
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Overview: Description of key SoC power Management hardware features

Power Management Unit (PMU)
- The PMU is a discrete Integrated Circuit
- It supplies all the power rails to the SoC
- Jetson uses the AMS 3722
  - Tegra K1 communicates with it via I2C bus
    - System software sends commands to it to change settings on the various power rails
- The device offers us no debug information
  - There are no registers telling us current draw etc.
Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Power Islands
- The chip is divided into Power Islands or Domains
- All cores in a Power Islands use the same power rail
- Examples of Power Islands
  - CPU
    - Each CPU (1-5) is in a separate power domain
    - All handled by the Flow Controller
  - Video (VE)
    - Includes Camera (CSI), Image Sensor Processor (ISP)
  - Video Decode Engine (VDE)
  - SAX
  - SATA
Power Islands
- Examples of Power Islands (cont'd)
  - SOR
    - HDMI, Display (DSI)
  - XUSB
    - USB Device
  - XUSB
    - USB Host
- Always on Domain (AOD)
  - Includes the ARM7 COP processor that handles PM
  - Real Time Clock
    - Interrupt Controller
- To turn a domain off all the cores in the domain must be idle
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Dynamic Voltage and Frequency Scaling (DVFS)
- Lower frequency implies lower power requirements
- Dynamically changing frequency based on the load allows for fine grained power control
- The Tegra K1 has predefined Frequency / Voltage pairs
  - For example, the ARM processor complex can be set to the following values:

```bash
# cat/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
51000 102000 204000 312000 564000 696000 828000 960000 1092000 1122000 1224000 1326000 1428000
1530000 1632000 1734000 1836000 1938000 2014500 2116500 2218500 2320500
```

- cpufreq uses this capability to reduce frequency (power)
SoC Power Management

**Auto Clock Gating**
- Cores are designed to turn off automatically when there is no work
- When the core clock is shut off power consumption is greatly reduced*
- How does this happen?
  - Chip level RTL design tools look at enable signals
    - When the enable is not present the clock driving a block is automatically turned off
- eg I2C transfers

**Thermal Sensing**
- Chips now include thermal sensing and cores will be freq reduced or shut down if temperatures get too high
  - This is done to protect the chip
Linux Power Management Optimization on Nvidia Jetson

System Software

Software Controlling ARM Power Management

cpufreq
- Controls frequency / power to the ARM CPU complex
- Voltage / Frequency pairs are defined by the chip manufacturer
  - They can be found in the Device Tree
- cpufreq has pluggable governors
  - Ondemand
    - Widely used
  - Userspace
  - Performance
  - Powersave

```
root@tegra-ubuntu:~# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
ondemand
root@tegra-ubuntu:~#
```
Linux Power Management Optimization on Nvidia Jetson

System Software

cpuidle
- controls what happens when a CPU has no work to perform
- Two governors are available
  - ladder
  - menu
    - main governor in use

WFI
- ARM assembly instruction
- It is used to put the core to sleep
- To sleep the last instruction executed is WFI

asm
  ...
  # Ensure interrupts are enabled for wakeup
  wfi  # Wait For Interrupt
  ...
  # Code executed when core wakes up
Linux Power Management Optimization on Nvidia Jetson

System Software

Tickless idle
- The kernel can be configured to run without the usual scheduler timer tick
- This reduces power consumption as CPUs are not woken up 'x' times / second
- CONFIG_NO_HZ_IDLE=y is used widely by embedded ARM implementations
- The Nvidia Tegra kernel uses it as well:
System Software

Device Drivers

Static Power Management Interfaces
- These are the legacy interfaces called when specific devices are suspended or resumed

- Standard struct used by all device drivers:

```c
struct dev_pm_ops {
    ..
    suspend()                 # entry points called by the kernel
    resume()                  # on power up and down
    ..
}
```
Linux Power Management Optimization on Nvidia Jetson

System Software

Dynamic Power Management
  Runtime PM
    - Controls idle for devices (as opposed to just the CPU)
    - pm_runtime_get
      - tell the Power Manager that you want to use the core
    - pm_runtime_put
      - tell the Power Manager that you do not need the core

- These interfaces use 'use counts' to decide when to shut down a core
- When the use count goes to 0 the core can be shut down
Linux Power Management Optimization on Nvidia Jetson

Data Driven Power Optimization Techniques

Overview
With the hardware and system software ground work laid out we can look at ways to improve power consumption

- Tools to help us view performance and power
- sysfs interfaces to control or monitor performance and power
- Improving our monitoring capability
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- ARM Streamline and gator
  - ARM Streamline is a graphical tool developed by ARM
  - It is designed to help view ARM performance
  - It collects and displays data, near real time, on a wide variety of system parameters
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- ARM Streamline and gator
  - The gator driver and the gator daemon run on the target
  - Gator collects data near real time & sends this to Streamline
  - Streamline connects to gator via the ethernet port

```
ubunto@tegra-ubuntu:/gator$ sudo insmod ./gator.ko
[sudo] password for ubuntu:
```

```
[       64.142847] gator: version magic '3.10.24 SMP preempt mod_unload ARMv7 p2v8' should be '3.10.24-g6a2d13a'
```

```
ubunto@tegra-ubuntu:/gator$
```

```
ubunto@tegra-ubuntu:/gator$ sudo ./gatord &
[1] 1411
```

```
ubunto@tegra-ubuntu:/gator$
```

```
ubunto@tegra-ubuntu:/gator$ ps -elf | grep gator
```

```
4 S root  1411 1376  2  80 0 - 1856 poll_s 05:07 ttyS0 00:00:00 sudo ./gatord
4 S root  1414 1411  0 61 -19 846 ep_pol 05:07 ? 00:00:00 ./gatord
0 S ubuntu 1416 1376  0 80 0 - 945 pipe_w 05:07 ttyS0 00:00:00 grep --color=auto gator
```

```
ubunto@tegra-ubuntu:/gator$
```

- gator is open source and available on github
Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- WattsUp Power Meter
  - In line power monitoring device
  - Connects between the input 120V AC line and the Jetson power supply
  - Maximum sampling rate is 1 sample / second
  - USB text output stream that can be captured in term window
    - Watts
    - Amps
    - Volts
- Source code is available for the capture software
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power
Linux Power Management Optimization on Nvidia Jetson

Suspend

```
#cd /sys/power
#echo lp1 > suspend/mode
#  
#echo mem > state
```

The term window will now lock up – the K1 is in Suspend state.

The power drops to 16ma.
I then pulled the fan power and it dropped to 0ma.

The fan draws about 16ma
Linux Power Management Optimization on Nvidia Jetson

Low Power

LP1
- Low Power 1
- VDD_CPU is off
- DRAM memory controller is off
- The DRAM state is maintained using self refresh mode

LP0
- Low Power 0
- VDD_CPU is off
- VDD_CORE is off
  - separate power rail supplied by the PMU
- DRAM memory controller is off
- The DRAM state is maintained using self refresh mode
Linux Power Management Optimization on Nvidia Jetson

Deep Sleep

# cd /sys/power
# echo lp0 > suspend/mode
#
# echo mem > state

Term will now lock up

- To Resume
  - generate an interrupt
  - eg insert SD/MMC card. This will wake CPU up.

- Alternately start a timer
  Which will generate an interrupt
Linux Power Management Optimization on Nvidia Jetson

- memtester queued up
- 4 processes

# cd /sys/kernel
# cat cluster/active
LP or G

LP {51Mhz .. 1092Mhz}
G {204Mhz .. 2.32Ghz}
Linux Power Management Optimization on Nvidia Jetson

- memtester queued up
- 4 processes

```bash
# cd /sys/devices/system/cpu/cpu0/cpufreq
# echo "userspace" > scaling_governor

and then write one of the frequencies:

# echo 51000 > scaling_setspeed
```
Linux Power Management Optimization on Nvidia Jetson

Power Optimization Techniques

- Simple Techniques for Power Optimization

  - Turning CPU cores OFF and ON manually

    echo 0 > /sys/devices/system/cpu/cpu/cpuquiet/tegra_cpuquiet/enable
    echo 0 > /sys/devices/system/cpu/cpu1/online
    echo 0 > /sys/devices/system/cpu/cpu2/online
    echo 0 > /sys/devices/system/cpu/cpu3/online
    echo LP > /sys/kernel/cluster/active
Linux Power Management Optimization on Nvidia Jetson

Controlling GPU performance

- List the available GPU frequency rates:

  ```
  # cat /sys/kernel/debug/clock/gbus/possible_rates
  72000 108000 180000 252000 324000 396000 468000 540000 612000
  648000 684000 708000 756000 804000 852000 (kHz)
  ```

- Then set a rate (in Hz):

  ```
  # echo 852000000 > /sys/kernel/debug/clock/override.gbus/rate
  # echo 1 > /sys/kernel/debug/clock/override.gbus/state
  ```

- Finally verify the rate is set correctly:

  ```
  # cat /sys/kernel/debug/clock/gbus/rate
  # 852000
  ```
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

- Limitations of WattsUp

- It is monitoring the platform power
  - 120V AC primary voltage
- The sampling frequency is low
  - Maximum frequency is 1 sample / sec
  - We are not seeing “micro changes” in current draw
- Current draw is in ma.
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

- ARM Energy Probe

- 3 probe points
  - Allows us to attach to 3 separate power rails
  - This can be connected to different PMU power rails
- USB interface to capture data stream
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

- ARM Energy Probe

- The device works based on Resistive Current Measurement – RCM
  - In line shunt resistors are required
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

ARM Energy Probe

The tool interfaces via USB to ARM Streamline

Various energy parameters Will be displayed in a Capture window in the tool

10Khz sampling rate
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

ARM Energy Probe

- A customized board is required to use the probe

- The ARM Coretile Express Platform
  - Example implementation

- If you are using Jetson to do early prototyping and planning on building your own custom board this is something to consider
  Especially if power is key to the success of your product
Linux Power Management Optimization on Nvidia Jetson

Improving Our Monitoring Capability

- RCM monitors on all the power rails
  - For greater visibility the ARM concept could be extended to all power rails

- On Chip Current Monitoring
  - Current monitoring built into the SoC
  - This allows real time monitoring of chip power rails
Linux Power Management Optimization on Nvidia Jetson

Recap

- We have reviewed the Jetson platform
  - Tegra K1 capabilities
  - Tegra K1 power management features

- Linux on Tegra
  - Kernel and device drivers

- We looked at some tools and techniques to monitor and improve power consumption
Linux Power Management Optimization on Nvidia Jetson

Questions?
Linux Power Management Optimization on Nvidia Jetson

Thank you!

Contact: merlin@gg-research.com
Linux Power Management Optimization on Nvidia Jetson

Backup Slides
Linux Power Management Optimization on Nvidia Jetson

Data Driven Power Optimization Techniques

Turning off the HDMI port if a display isn't required

If no HDMI monitor is plugged in during bootup, then the Jetson TK1 board will use less power since various display related features won't be enabled. Also, turning off the HDMI port can slightly reduce the power usage of the board. Run this as root (see the section below on how to get root privileges):

```
    echo -1 > /sys/kernel/debug/tegra_hdmi/hotplug
    echo 4 > /sys/class/graphics/fb0/blank
```

Setting I/O lines to tristate