An Introduction to Asymmetric Multiprocessing: when this architecture can be a game changer and how to survive it.

Nicola La Gloria, Laura Nao

Kynetics LLC
Santa Clara, California
Hybrid Architecture: warpx.io

The Hybrid Design Architecture (HDA) combines the power of an application processor with the ease-of-use of micro-controllers.
Software on warpx HDA
SMP vs AMP

SMP on homogeneous architectures:

- Single OS controlling two or more identical cores sharing system resources
- Dynamic scheduling and load balancing
SMP vs AMP

**AMP** on heterogeneous architectures:

- Different OS on each core --> full-featured OS alongside a real-time kernel
- Inter processor communication protocol
- Efficient when the application can be statically partitioned across cores - high performance is achieved locally
Supervised vs Not Supervised

- Strong isolation
- Hides non-trivial AMP details (e.g. resource assignment, inter-core communication)
- Security and robustness
- Overhead of a software layer

- Achieve best performances by running natively
- Boot sequence complexity
- Harder to debug
Interprocessor Communication

- RPMsg Lite, OpenAMP RPMsg, ...
- VirtIO, Virtqueue, Vring
- Shmem, MU, Mailbox

Transport Layer
VirtIO / Virtqueue
MAC Layer
Shared Memory
Inter-core Interrupts
Physical Layer
NXP i.MX7 overview

- Cortex-A7 core + Cortex-M4 core
- **Master - Slave** architecture
  - A7 is the master
  - M4 is the slave
- Inter processor communication
  - MU - Messaging Unit
  - RPMsg component (OpenAMP framework)
- Safe sharing of resources
  - RDC - Resource Domain Controller
NXP i.MX7 - RDC
NXP i.MX7 IPC - MU
MAC (VirtIO)
The OpenAMP framework - RPMsg
RPMsg on Linux
Hybrid Linux/FreeRTOS Demo

Demo Goal:

- IMU sensor (I2C) read by MCU task
- Calculate objective function (module of acc, mag, gyro vectors)
- Log/plot sensor samples on MPU
- Safely recover from a kernel panic

Hardware Setup

- Boundary Devices Nitrogen 7, Toradex Colibri i.MX7 SOM
  - NXP i.MX7D processor - ARM dual Cortex-A7 + ARM Cortex-M4
  - Segger J-Link Probe
Cortex M4 Bring Up (1)

Environment setup:

- Download FreeRTOS sources
  [https://github.com/boundarydevices/freertos-boundary.git](https://github.com/boundarydevices/freertos-boundary.git)
- Download GNU ARM Embedded Toolchain
- Example applications for Cortex-M4 are located in the
  examples/imx7d_nitrogen7_m4/ folder
- Scripts for building both debug and release binaries are available in the
  armgcc subfolder
M4 Binary application can be loaded on the Cortex-M4 in different ways:

- **U-Boot** - ums gadget + m4update
- using **remoteproc** framework (linux userspace)
- using **imx-m4fwloader** from NXP (linux userspace): [https://github.com/codeauroraforum/imx-m4fwloader](https://github.com/codeauroraforum/imx-m4fwloader)

M4 code can be linked and loaded to one of the following:

- **TCM** - 32KB (preferred)
- **OCRAM** - 32KB
- **DDR** - up to 1MB
- **QSPI Flash** - 128KB
IDE Setup

- Eclipse for C/C+
  - GNU MCU Eclipse: plugins and tools for embedded ARM development -
    [https://marketplace.eclipse.org/content/gnu-mcu-eclipse](https://marketplace.eclipse.org/content/gnu-mcu-eclipse)
- GDB
- J-Link scripts for iMX7D for debugging both Cortex-A7 cores and Cortex-M4 -
  [https://wiki.segger.com/IMX7D](https://wiki.segger.com/IMX7D)
- FreeRTOS Kernel Awareness plugin from NXP
- ARM DS-5 (not free)
- Sourcery Codebench (not free)
Workbench

- Breakpoints (MCU)
- FreeRTOS kernel awareness
- MPU console
- MCU console

Embedded Linux Conference

OpenIoTSummit
Demo Parameters

Remote core:

- Sample IMUs every 10ms
- Calculate the objective function on MCU (module of vectors)
- Buffer of 300 elements = 3Kb (stored TCM Memory only 32 Kb)
- Items (12 byte each) are dequeued and sent to master 10 at a time every 100 ms

Master core:

- Master reads incoming samples by polling the character device
Architecture Overview

**start_cmd, stop_cmd, heartbeat**

**Linux 4.9**

**FreeRTOS 1.0.1**

---

**IMU Polling Task**
- Continuously poll IMU sensors
- Enqueue samples in shared buffer

**Data Sender Task**
- Init rmsg channel
- Send data to master

---

**rmsg_char_client**

**User space**

```
./dev/rmsg_ctrl
./dev/rmsg0
```

**Kernel space**

**rmsg_char**

---

**rmsg channel**

**OpenIoT Summit**
The OpenAMP framework - RPMsg
Control Flow (2 cores)

- S0 RPMg channel is down
- S1 RPMg channel is up, /dev/rpmsg0 is created
- S2 RPMg channel is up, endpoint created, data is dumped into a log file

open /dev/rpmsg0

register rpmsg char driver + probe

rpmsg_char_client

close /dev/rpmsg0

read /dev/rpmsg0

Data sender task

Channel created

Master is dead

start_cmd

Master heartbeat

stop_cmd

- S0 RPMg channel is down
- S1 RPMg channel is up (sampling IMU sensor, buffering data)
- S2 RPMg channel is up, sending data to master core, (sampling IMU sensor, buffering data),

Register rpmsg char driver + probe

open /dev/rpmsg0

close /dev/rpmsg0

read /dev/rpmsg0

rpmsg_char_client

Data sender task

Channel created

Master is dead

start_cmd

Master heartbeat

stop_cmd

- S0 RPMg channel is down
- S1 RPMg channel is up (sampling IMU sensor, buffering data)
- S2 RPMg channel is up, sending data to master core, (sampling IMU sensor, buffering data),
What if Linux Kernel Panics

- **Kexec**: system call to load and boot into another kernel from the currently running kernel (4.9.74).
  - crashkernel=128M [normal kernel cmdline]
  - irqpoll, nosmp, resetDevices [crash kernel cmdline]
  - --load-panic option
- **Kdump**: Linux mechanism to dump machine memory content on kernel panic.
- Kexec/Kdump support on ARM platforms is still experimental
Video of the Demo
Pitfalls

- Before announcing the remote service, MCU checks whether master is up. If notification arrives too early (virtqueue kick function call) when booting crash kernel the system might hang.
- Sometimes kexec still hangs and fails to soft-reboot - more frequent when streaming continuously instead of sending data bursts (but we don’t know why).
References

- OpenAMP project page: [https://github.com/OpenAMP/](https://github.com/OpenAMP/)
- Maintainers:
  - Open-amp:
    - Wendy Liang
  - RPMsg (Linux):
    - Ohad Ben-Cohen
    - Bjorn Andersson
  - Kexec (Linux):
    - Eric Biederman
  - Kdump (Linux):
    - Dave Young
    - Baoquan He
Q/A