ARM versus x86 – insights gained from the OSADL QA Farm

Carsten Emde
Open Source Automation Development Lab
(OSADL) eG
ARM versus x86 – Disclaimer

• The Open Source Automation Development Lab (OSADL) primarily takes care of using Open Source software in the industry and in industrial products.

• One of our services is to run a **quality assessment farm**. We can use our farm data to compare ARM versus x86.

• However, since we only use Open Source software, our results are restricted to such software, and we – evidently – cannot make any conclusion on the behavior of any proprietary software running on the tested architectures.
What is the OSADL QA Farm? (1)

**OSADL Test Rack**

- Eight individual tablets
- Power supply 220 V, Ethernet, RS232
- 10/100/1000 Mb/s network switch with port mirroring
- Remote power switch with metering of every tablet
- Serial network adapter for every tablet
- KVM switch (optional) for every tablet
- One control server per rack
- Munin monitoring and Nagios event recording
- Additional scripts and tools developed by OSADL
- Designed and realized by kernel developer Thomas Gleixner
- Available at OSADL founding member Linutronix
OSADL QA Farm (2)

Individual systems are mounted on removable tablets
Cloud-based communication between test systems, data collectors and admin systems

Test center #1

Data collector

Management, evaluation

Test center #2

VPN Channels
OSADL Realtime QA Farm

- **Hardware**
  - CPUs
  - Benchmarks
  - Graphics
  - Benchmarks
  - Kernels
  - Boards/Distros
  - Latency monitoring
  - Latency plots
  - System data
  - Profiles
  - Compare
  - Awards

Overview of the processors under test and system locations (》 100 systems)
Processor families: ARM, MIPS, PowerPC, x86

- Processor family ARM (》 20 systems)
  - Manufacturers: Broadcom, Freescale, Marvell, NXP, TI, Xilinx

- Processor family x86 (》 60 systems)
  - Manufacturers: AMD, Intel, VIA
osadl.org/QA – system selector

Processor family ARM
Manufacturers: Broadcom Freescale Marvell NXP TI Xilinx

TI

• AM3354 @800 MHz: rack #8/slot #8
• AM3359 @720 MHz: rack #7/slot #5
• AM3359 @800 MHz: rack #a/slot #5
• AM3505 @500 MHz: rack #8/slot #6
• AM3517 @600 MHz: rack #2/slot #3
• OMAP3359 @1000 MHz: rack #7/slot #8
• OMAP335x @1000 MHz: rack #7/slot #6
• OMAP3525 @720 MHz: rack #2/slot #2
• OMAP4430 @1000 MHz: rack #9/slot #7 – rack #7/slot #7
• OMAP4460 @920 MHz: rack #4/slot #4
ARM versus x86 – criteria of comparison of suitability for embedded systems

1. Run-time stability (uptime without crashes)
2. Delay between market introduction and “reasonable” mainline Linux support
3. Amount of unavoidable closed-source software
4. Real-time support and real-time capabilities
5. Computational efficiency (computing power by watts)
6. Minimum power consumption
ARM versus x86 – criteria that will not be considered

1. Price and other purchase conditions
2. Long-term availability
3. Design support
4. Many other individual criteria
How to determine run-time stability?

Uptime - by day

Day

Uptime - by week

Week

Uptime - by month

Month

Uptime - by year

Year
How to determine real-time capabilities?

Signal path

External event, e.g. from a light barrier -> Gate latency (3) -> CPU IRQ (3) -> Interrupt service routine (9) -> Scheduling, context switch (15) -> Return from system call in user space

Total latency or preemption latency = 30

3
Gate latency

3
CPU IRQ

9
Interrupt service routine

15
Scheduling, context switch

6
IRQ latency

30
Total latency or preemption latency
How to determine real-time capabilities? Test setup

PowerPC 750FX@600MHz
64 MB SDRAM on SODIMM, 16 MB Flash-EPROM
10/100 Mb/s Network
2 serial channels RS232 and RS485
2 TTL Outputs, 4 TTL Inputs
4 Status LEDs
On-board FPGA

Interrupt service routine
Scheduling, context switch
### How to determine real-time capabilities? Data set

<table>
<thead>
<tr>
<th>Line #</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(No latency recording below 1 µs duration)</td>
</tr>
<tr>
<td>11</td>
<td>76</td>
<td>(A total of 76 latency values between 10 and 11 µs duration)</td>
</tr>
<tr>
<td></td>
<td>2238</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8800</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20027</td>
<td>(Most frequently observed latency values between 13 and 14 µs duration)</td>
</tr>
<tr>
<td></td>
<td>18433</td>
<td></td>
</tr>
<tr>
<td></td>
<td>430</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ . . ]</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>(No overflow)</td>
</tr>
</tbody>
</table>
How to determine real-time capabilities? Latency plot
How to determine real-time capabilities? Long-term (1)
How to determine real-time capabilities? Long-term (2)
How to determine computational efficiency?
Power consumption versus P states

CPU frequency scaling - by day

Electricity (power) at ros8 - by day

Embedded Systems Design Conference
July 3, 2014, Munich, Germany
ARM versus x86 – insights gained from the OSADL QA Farm
How to determine computational efficiency?
Power consumption versus P states

CPU C states - by day

Electricity (power) at r0s8 - by day

Embedded Systems Design Conference
July 3, 2014, Munich, Germany
ARM versus x86 – insights gained from the OSADL QA Farm
How to determine computational efficiency?

- Select performance governor (max. clock frequency)
- Disable any sleep states
- Create Dhrystone and Whetstone test programs with same compiler and comparable compiler options
- Run Dhrystone and Whetstone tests while measuring power consumption
- Calculate kilo Dhrystones/W
- Calculate Whetstones/W
How to determine computational efficiency?

- Select performance governor (max. clock frequency)
- Disable any sleep states
- Create Dhrystone and Whetstone test programs with same compiler and comparable compiler options
- Run Dhrystone and Whetstone tests while measuring power consumption
- Calculate kilo Dhrystones/W
- Calculate Whetstones/W

**WARNING!** Systems may not be comparable with respect to power supply and type and amount of peripheral devices.
## Expected results

<table>
<thead>
<tr>
<th>Criterion / Processor</th>
<th>ARM</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Suitability with respect to run-time stability</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>2. Suitability with respect to mainline Linux support delay</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>3. Suitability with respect to unavoidable closed-source software</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>4. Suitability with respect to real-time support and real-time capabilities</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>5. Suitability with respect to computational efficiency</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>6. Suitability with respect to minimum power consumption</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Results: Run-time stability examples

**x86** Example of a stable system

**ARM** Example of an unstable system
Results: Run-time stability

- **ARM**
  - All AM335x and i.MX6 systems are stable, no system crashes observed.
  - Other ARM systems (some of them very new, some of them very cheap) are not yet stable, we are working on this issue.

- **x86**
  - All x86 systems are stable, no systems crashes observed.
Results: Delay between market introduction and “reasonable” mainline Linux support for ARM

- ARM
  - Some older systems (introduced before 2011) still do not have mainline Linux support.
  - Some systems (e.g. introduced in 2012) are currently getting mainline Linux support.
  - Recently introduced systems got mainline Linux support within months.
Results: Delay between market introduction and “reasonable” mainline Linux support for x86

- x86
  - As a general rule, an x86 system has mainline Linux support at the date of its introduction thanks to Intel's and AMD's Linux support. The only exception to this rule may be lacking or insufficient support of on-board graphics.
Results: Amount of unavoidable closed-source software

<table>
<thead>
<tr>
<th>Which / Processor</th>
<th>ARM</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS</td>
<td>no</td>
<td>YES</td>
</tr>
<tr>
<td>On-board graphics controller</td>
<td>YES</td>
<td>YES/no</td>
</tr>
</tbody>
</table>
Results: Real-time support and real-time capabilities

- Rule of thumb:
  An industrial computer should be able to continue execution of a user-space program that is waiting for an unpredictable asynchronous event within $10^5$ times its clock interval.

- Example:
  A clock frequency of 1 GHz is equivalent to a cycle duration of 1 ns. This duration times $10^5$ equals 100 µs. Thus, the worst-case latency of a 1-GHz processor should not exceed 100 µs.
Result example: Real-time capabilities of an ARM processor @996 MHz (100 million cycles)
Result example: Real-time capabilities of an ARM processor @996 MHz (22 billion cycles)
Result example: Real-time capabilities of an x86 processor @2900 MHz (100 million cycles)
Result example: Real-time capabilities of an x86 processor @2900 MHz (60 billion cycles)
Result example: Real-time capabilities of an x86 processor (100 million cycles), apparently good
Result example: Real-time capabilities of an x86 processor (150 billion cycles) SMI-related outliers
Results: Computational efficiency (integer arithmetic)

- **x86** Intel Core i7-E610 @2533 MHz
- **x86** AMD G Series T40EC0/E1 @1140 MHz
- **ARM** Freescale i.MX6 @996 MHz (with/without peripheral devices)
- **x86** Intel Atom D525 @1800 MHz
- **ARM** TI AM3359 @720 MHz
- **x86** AMD Geode @500 MHz
- **ARM** Broadcom BCM2708 @700 MHz
- **x86** Intel Core i3-4360 @3700 MHz (former “Haswell”)
- **ARM** Xilinx Zync @666 MHz
- **x86** Intel Celeron J190 @1990 MHz (former “Bay Trail”)
- **x86** Intel Pentium @133 MHz VMEbus
Results: Computational efficiency (floating point)

- x86 Intel Core i7-E610 @2533 MHz
- x86 AMD G Series T40EC0/E1 @1140 MHz
- ARM Freescale i.MX6 @996 MHz (with/without peripheral devices)
- x86 Intel Atom D525 @1800 MHz
- x86 AMD Geode @500 MHz
- ARM Broadcom BCM2708 @700 MHz
- ARM TI AM3359 @720 MHz
- x86 Intel Pentium @133 MHz VMEbus
- ARM Freescale i.MX6 @996 MHz (with/without peripheral devices)
- x86 Intel Celeron J1900 @1990 MHz (former “Bay Trail”)
- ARM Xilinx Zync @666 MHz
- x86 Intel Core i3-4360 @3700 MHz (former “Haswell”)

Graph showing computational efficiency (floating point) over time from 1995 to 2015, comparing x86 and ARM processors.
Result: Minimum power consumption

ARM: 1.7 watt

x86: 250 watt
# Table of the results

<table>
<thead>
<tr>
<th>Criterion / Processor</th>
<th>ARM</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Suitability with respect to run-time stability</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>2. Suitability with respect to mainline Linux support delay</td>
<td>70%</td>
<td>100%</td>
</tr>
<tr>
<td>3. Suitability with respect to unavoidable closed-source software</td>
<td>90%</td>
<td>80%</td>
</tr>
<tr>
<td>4. Suitability with respect to real-time support and real-time capabilities</td>
<td>90%</td>
<td>100%</td>
</tr>
<tr>
<td>5. Suitability with respect to computational efficiency</td>
<td>100%</td>
<td>90%</td>
</tr>
<tr>
<td>6. Suitability with respect to minimum power consumption</td>
<td>100%</td>
<td>70%</td>
</tr>
</tbody>
</table>
Estimated **ARM** and **x86** suitability for embedded systems over time

- **24 months ago**
  - ARM: 0.5
  - x86: 0.9
- **Now**
  - ARM: 0.8
  - x86: 0.9
- **In 24 months**
  - ARM: 0.9
  - x86: 0.9
What is on our wish list?

• **ARM**
  - Immediate mainline Linux support, no more vendor kernels
  - Linux device tree by default included in hardware delivery
  - Open Source graphics drivers

• **x86**
  - Open Source BIOS
  - Open Source graphics drivers
What else is on our wish list?

- Every provider of industrial computing systems who is claiming that at least one of the products runs on Linux should become OSADL member.

- Do not expect Linux-supporting companies such as Google to fix your Linux issues!
The OSADL QA Farm and all other OSADL activities would not be possible without the OSADL members.

The only OSADL member among the exhibitors of this Embedded Systems Design Conference!
Conclusion

Both **ARM** and **x86** processors are relatively well suitable to be used in industrial and other embedded systems that require stable, reliable and deterministic run-time behavior.

Important advantages of **ARM** processors are
- No BIOS and, therefore, no BIOS-related problems
- Power consumption in the range of about 2 to 20 watt

Important advantages of **x86** processors are
- Nearly perfect and early mainline Linux support
- Very high computing power and very short real-time response