Realtime capabilities of low-end PowerPC and ARM boards for embedded systems

Alexander Bauer
PHYTEC Messtechnik Gmbh
Robert-Koch-Str.39, 55129 Mainz, Germany
a.bauer@phytec.de

Abstract

With the stepwise integration of the Realtime Preemption Patches (RT-Preempt) into the Mainline Linux kernel and their support for architectures other than Intel and AMD, there are now a number of choices which board to use for a particular embedded realtime project running Mainline Linux. In order to select the appropriate processor and clock frequency, it would be desirable to have some generally applicable ranges of worst-case latencies that can be obtained using the various processor types and conditions.

We, therefore, determined the internal worst-case latency of PowerPC and ARM boards running Linux 2.6.20 and above patched with RT-Preempt. The PowerPC-board (Phytec phyCORE-MPC5200B) was running at 266 and 400 MHz, the ARM board (Phytec phyCORE-PXA270) was running at 266 and 520 MHz.

This article will provide the details of the various measurement set-ups, present the results and discuss them with respect to the design differences between PowerPC and ARM.

1 Introduction

In the embedded market there is a wide range of processors to choose from. A processor is typically selected for a customer design because of it features, e.g. video interface and peripherals, and the clock frequency. With the growing importance of Linux and especially realtime Linux for customer designs in the embedded market, it is also essential to choose the right processor that will cope with the realtime requirements of the customer's application.

Phytec offers different microcontroller boards for the embedded market. Currently, we support realtime Linux, based on a vanilla kernel with the RT-Preempt patches, on an ARM (PXA270) and PowerPC (MPC5200) platform.

To help our customers with the decision which processor fits best to their realtime application, we performed several latency tests on the two platforms with different test scenarios.

This paper presents the results of the latency tests and discusses the results with respect of the different processor designs.

2 Latency Tests

For the latency tests based on MPC5200 we used the PHYTEC phyCORE MPC5200 board with 400 MHz as a reference platform. As an example of the effect of the processor speed we reduced the core clock frequency by modifying the core PLL configuration pins on the development board to 266 MHz.

The reference platform for the PXA270 was the phyCORE-PXA270 board running at 520 MHz. We disabled the turbo mode of the processor by clearing the turbo bit in the U-Boot to reduce the core frequency to 260 MHz.

The two systems were running the 2.6.20 kernel with the RT-Preempt patches maintained by Ingo Molnar and Thomas Gleixner.

The latency of the system was measured by the cyclictest tool from Thomas Gleixner. This tool measures the task switch latencies of a system using high resolution timers.

We performed several tests that were lasting up to 72 hours on idle systems and systems being stressed using flood ping request.

Another test setup was used where an external interrupt source was connected to the system. The external interrupt triggers an interrupt handler that was setting a digital output pin.

The latency between the trigger of the interrupt and the occurence of the level change of the output pin was measured with an oscilloscope.

2.1 Cyclictest on MPC5200

On the MPC5200, the cyclictest shows a worst-case latency of around 60 microseconds on an idle system running at 266 MHz and 45 microseconds at 400 MHz.

On a system being flooded with ping requests, the maximum latency increased to 120 microseconds on the 266 MHz system and 95 microseconds on the 400 MHz system.

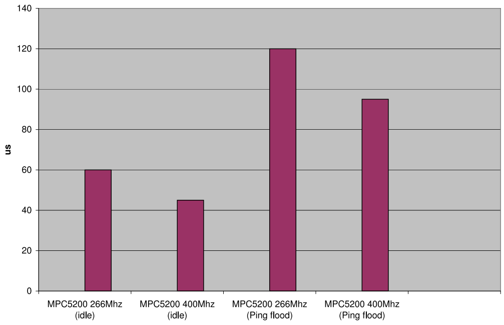

FIGURE 1: *Cyclictest results of MPC5200*

As shown in Figure 1, the latency of the system is increasing when the processor speed is reduced, but the latency is not linear to the processor speed.

2.2 Cyclictest on PXA270

The cyclictest on the PXA270 shows a latency of 450 microseconds on an idle system running at 520 MHz and 550 microseconds with 260 MHz.

The latency increases on the system under stress with ping floods to 550 microseconds at 520 MHz and 600 microseconds at 260 MHz.

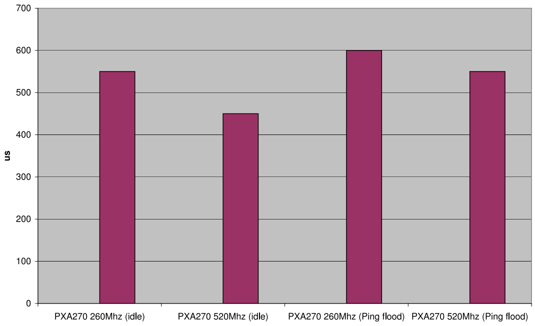

FIGURE 2: *Cyclictest results of PXA270*

As shown by cyclictest, the worst-case latencies of the PXA270 are around ten times higher than on the MPC5200 running with a slower clock speed.

2.3 Interrupt latency MPC5200

The interrupt latency of the MPC5200 was determined to be around eight microseconds on the 400-MHz system in idle mode and around ten microseconds with ping floods.

The 266-MHz system shows nearly identical results with nine microseconds in idle mode and eleven microseconds under stress.

2.4 Interrupt latency PXA270

The PXA270 has, similarly to the other test scenario, a higher latency as the MPC5200.

The PXA270 running at 520 MHz needs around 70 microseconds from the detection of interrupt to branch to the Interrupt Service Routine and set the level at the output pin.

Under stress, the latency was measured at around 90 microseconds with some outliers up to 150 microseconds. The 260-MHz system latency was determined at around 85 microseconds in idle mode and 120 under stress. There are also outliers on this system going up to 260 microseconds.

3 Processor Design

The results show that the processor speed has an influence on the latency but not in the same amount as the speed ratio.

On the other hand, the PXA270 has in both test environments worst case latencies that were about a factor 10 higher than the MPC5200, despite of the fact that the PXA270 is running at a higher clock frequency.
Differences in the processor design between the PowerPC and the ARM architecture might have an impact on the latency.

3.1 Cache architecture

An import difference between the two processors is the cache architecture.
The PXA270 is based on an ARM core that uses a virtual cache architecture. In this architecture, the cache is located between the CPU and the MMU. This design has the advantage that on a cache hit the address does not need to be converted to a physical address. But the disadvantage of this architecture is that when a thread switch occurs the cache needs to be flushed which imposes an additionally latency.
In contrast, the MPC5200 is equipped with an e300 core, where the cache is placed between the MMU and the memory (physical cache). In this design, the address translation has to take place before the cache access. This leads to a slower access time on the cache because of the address translation but the cache contents is consistent in different threads, in such a way that the cache does not need to be flushed.
In addition, the PXA270 is equipped with 32k data and instruction cache while the MPC5200 only has 16k caches. In general, a bigger cache is an advantage because it increases the probability of cache hits. But when it comes to task switching, a bigger cache is a drawback because flushing the cache takes more time.

3.2 RAM interface

The PXA270 memory controller only supports SDRAM that runs with 104 MHz.
The MPC5200 also supports SDRAM and in addition DDR SDRAM running at 133 MHz. The MPC5200 was equipped with DDR SDRAM for these tests. The access times on DDR SDRAM are faster than on SDRAM, this gives the MPC5200 a performance advantage against the PXA270.

4 Conclusion

The measurements of this article reveal the following results:
1. The clock speed is not the most important factor for a realtime application.
A better realtime behavior cannot be achieved by simply choosing a processor with a higher clock frequency.
The internal processor architecture obviously has a more important impact on a processor's realtime capability. It is important how the internal cache is designed and connected to the Memory Management Unit. Also the speed of the RAM interface will have an impact on the realtime quality.
2. The two processors show different results in the latency tests although they were running at nearly the same clock speed.
The PXA270 is not specially designed for realtime applications. Its focus is on the multimedia market and it is widely adopted in the handheld and mobile phone market.
The MPC5200 is designed for high performance applications and thus more suitable for demanding realtime applications.
Neverless, the PXA270 is a good choice for applications with lesser realtime requirements and more focus on multimedia applications
Further researches had to be done on the influence of components, e.g. the RAM interface. Therefore a test with an MPC5200 with SDRAM will be done to see how much the RAM interface effects the realtime performance.
There are more tests planned for the future with the i.MX31 processor from Freescale, which also has an ARM core but a DDR SDRAM interface.
The latency measures in this article were done with basic methods. No special efforts have been undertaken to identify single latency sources and to optimize the code. It is well conceivable that shorter worst-case latency values can be obtained when a specific processor is selected for a given project and the settings can be optimized for the individual project requirements.

References

[1] *http://people.redhat.com/mingo/realtime-preempt*

[2] *http://git.kernel.org/?p=linux/kernel/git/tglx/rt-tests.git;a=summary*

[3] *http://www.freescale.com*

[4] *http://www.marvell.com*