Open Source Embedded Asterisk-based PBX

David Rowe, President Rowetel - October 04, 2006

Editor's note: the Open Source Asterisk PBX software was the subject of three articles on Audio DesignLine:

Build a PBX using Asterisk
Linux PBX Part 2
Linux PBX Pt 3

Asterisk is a popular Open Source IP-PBX which until now has been available mainly on x86 platforms. Typically an Asterisk system consists of the Asterisk software plus some PCI line interface hardware.

In this two part series we describe the design an embedded Asterisk PBX using the Analog Devices Blackfin processor and Silicon Labs line interface hardware. This approach has many advantages over the PC & PCI card approach including small form factor, low power consumption, and very low cost. However it also has some challenges, for example small memory and CPU resources, and a relatively complex development environment.

In this article we examine the hardware design of traditional x86 based IP-PBX systems and introduce the Blackfin processor. We then explore the design of an Embedded IP-PBX using the Blackfin, and examine its pros and cons.

IP-PBX Architecture 101
To understand the challenges in building an embedded IP-PBX we first need to take a look at a typical PC-based x86 Asterisk PBX.

The fact that general purpose Linux applications and DSP-intensive operations are happening on the same processor at the same time requires careful design and partitioning, for example allocating precious internal memory to DSP state variables and buffers to speed processing.
Benefits of running Linux on a DSP

Unlike many other DSPs the GNU toolchain has been ported to the Blackfin. This provides Engineers with familiar tools for developing and testing code, for example gcc, GNU make and the GNU-debugger gdb. The cross-compiler tools run on either Linux or Windows (via Cygwin). The GNU toolchain is mature and not subject to the support and upgrade issues of proprietary tools offered by many other DSP vendors. It is also free - making the entry cost to Blackfin development very low!

The ability to run uClinux on the Blackfin means a wealth of applications (for example embedded web servers, flash tools, TCP/IP networking) run "out of the box" and are immediately available.

The Blackfin-uClinux combination is well supported by Analog Devices in a similar fashion to other "Open Source" projects - there are public forums, CVS source repositories, a general spirit of cooperation, and a vibrant community. This is a fresh approach for DSP chips which have generally had proprietary tool chains with expensive "per seat" license models and paid-for support. Hats off to Analog Devices for supporting this initiative.

There are many "open source" reference hardware designs for the Blackfin (for example processor boards, ADC and USB daughter cards) and the cost of entry is just $226 for a Blackfin "STAMP" development card.

Interfaces between processors are an unwieldy part of many system designs. Combining the DSP and MCU operations onto a single processor removes hardware (e.g. register or memory mapped) and software (device driver/API) interfaces - traditionally a source of pain in multi-processor designs. A classic problem is synchronization issues, and the need for "two JTAG pods" to debug a multi-processor system. In the Embedded IP-PBX design presented here, the DSP part of the IP-PBX is now a "process" (in the Unix sense) on the uClinux-Asterisk system rather than a separate processor.

Conclusion

In this part we looked at traditional PC-based IP-PBX architecture and introduced an approach to building an embedded IP-PBX using the Blackfin processor.

In Part 2 we will explore the schematic-level design of line interface hardware for the IP-PBX. Then we will explore the device driver software that "glues" the line interface hardware to Asterisk running on the Blackfin.