EMBEDDED SYSTEM FOR VIDEO AND SIGNAL PROCESSING

Slavy Georgiev Mihov[1], Dimitar Stoykov Dimitrov[2], Krasimir Angelov Stoyanov[3], Doycho Dimitrov Doychev[4]

[1, 4] Faculty of Electronic Engineering and Technologies, Technical University – Sofia, 1797, Sofia, Bulgaria, phone: +359 2 965 32 81,e-mail: smihov@abv.bg, dddoychev@tu-sofia.bg

[2, 3] E-mail: dimitarstoikov@yahoo.com, krasi_stoyanov@abv.bg

Embedded microprocessor systems are becoming more and more widespread in techniques and machinery. In this paper, is made a description of a custom built system based on Analog Devices Blackfin processor family (ADSP-BF531/2/3). The system operates under Linux, which gives it flexibility and universality as an embedded solution for several of tasks. Some typical applications of such a device are: processing of audio and video signals, Ethernet connectivity (routers, switches gateways, access points and embedded servers), specialized systems for monitoring and control and so on. The involved hardware provides essential computation power at reasonable cost, which makes the system applicable in scenarios with high requirements. It's up to the software what the exact functions of the embedded device will be. The use of an operating system is a great benefit for it, allowing high level control of the available resources.

Keywords: embedded system, Blackfin processor, Linux Operating System

1. INTRODUCTION

Embedded microprocessor systems are indispensable for tasks of control, data processing, services and many others. This project aims to build from scratch a fully functional computer system provided with suitable software, capable of performing wide variety of tasks. For many reasons is decided, the system to be build upon an Analog Devices Blackfin DSP processor and especially ADSP-BF531/2/3. Some of them are: its high performance, power efficiency, widespread adoption, relatively low price, and last but not lest – its availability in an easy to maintain package. These three processors (ADSP-BF531, ADSP-BF532, and ADSP-BF533) are all completely pin-to-pin compatible - differing solely with respect to their performance and on-chip memory. This offers the ability to scale up or down the system, depending upon the end application needs.

The microprocessor system is provided with an operation system, capable of utilizing the full potentiality of the hardware. For the purpose is used a special Linux distribution called uClinux. The original uClinux was a derivative of Linux 2.0 kernel intended for microcontrollers without Memory Management Units (MMUs). However, the Linux/Microcontroller Project has grown both in brand recognition and coverage of processor architectures. Today's uClinux as an operating system includes Linux kernel releases for 2.0, 2.4 and 2.6 as well as a collection of user applications, libraries and tool chains. The benefits of using an operating system in an embedded system, though disputative, are a major consideration in the decision to harness Linux in this project.

Similar systems of Linux-driven embedded hardware are spread in the commercial world in the form of various network devices (switches, routers), multimedia solutions (cameras, media players) and other specialized systems.

2. SYSTEM OVERVIEW

The present project is based on the idea of building a microprocessor system for universal usage, capable of running Linux and small user applications. There are various similar solutions that happen to be quite successful. For clearing out what configuration should the system have, it is made an investigation of the existing solutions based on BF531/2/3.

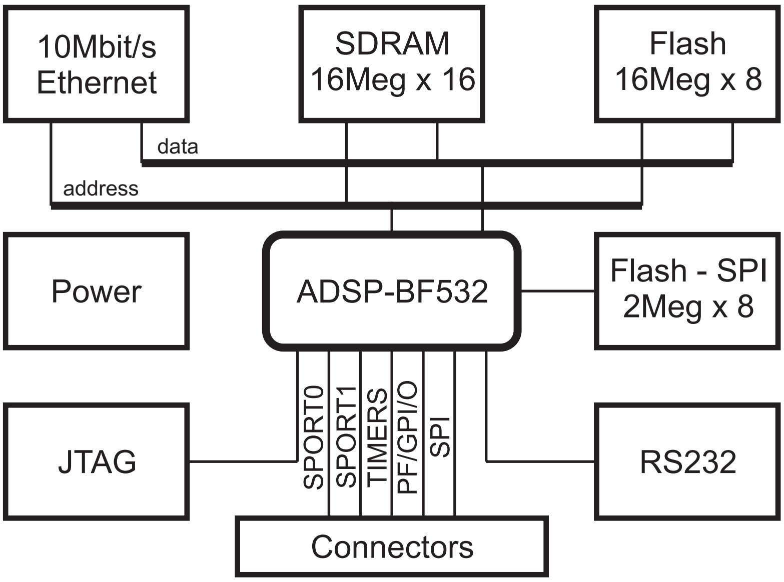

Figure 1 – Structure of the digital module.

The system is designed in two modules: an analog block for processing analog signals such as audio and video, and a digital block including the processor and all the peripheral devices essential for its operation (fig. 1). The digital module is designed mainly according the requirements of the Linux distribution. The system is provided with 32 MB SDRAM (16Mx16) and 16 MB NAND FLASH. The FLASH memory is used for storing the image of the Linux distribution, the file system and all the applications intended for execution. These 8 MB FLASH are quite sufficient for the needs of 4MB of the operation system. The amount of RAM memory also exceeds the needs of the Linux kernel and allows support for running very "greedy" user applications. In addition, the system is provided with a serial (SPI) EEPROM for storing the bootloader, which loads the OS on system reset.

The digital module of the main processor is supplied with interfaces for communication with the outer world and especially, with a personal computer. BF531/2/3 has an embedded UART module that is used for communication via RS232. This interface is used primarily in the stage of development of the board. As another opportunity for transferring data is provided 10 Mb/s Ethernet connection. Ethernet, with its speed and bandwidth, gives immense possibilities for communication, which are beyond comparison with RS232. It is intended for use mainly in the stage of operation, for transferring either large amounts of data or extremely intensive dataflow.

The unreleased prototype design of the system was provided with two Ethernet modules. One of them is a parallel Ethernet controller and the other is a serial SPI controller. The system was intended to use either the first or the second one. The presence of two Ethernet modules makes the system applicable in specific tasks, where both of them can be utilized. Such tasks, where embedded solutions can be involved include network specific tasks like routing, traffic monitoring, firewall and gateway applications, and many more. For some time it is very popular to use embedded devices in the role of web servers, providing some basic services accessible through World Wide Web. Actually the two-Ethernet variant of the initial designed has not been released till now, which narrows the range of possible application for the hardware.

The analog part of the board is aimed to process audio and video signals. It is formed as a separate module with autonomous power supply, which assembles with the digital module in one piece. The analog module includes three 12-bit ADCs for processing video signal consisting of three channels (red, green and blue). It also includes a video encoder, for processing composite video signal (fig. 2). The analog part of the board manipulates analog signals, and transmits digital data to the processor board it is connected to. The data transfer is done via the Parallel Peripheral Interface (PPI) of BF531/2/3. Both the analog and digital circuits are separated, which avoids interferences in operation. In case of need, each module can be individually modified, as the change won't affect the other part of the device. Actually, the device is not limited only to these two modules. Its module structure allows the system to be scaled up with ease later.

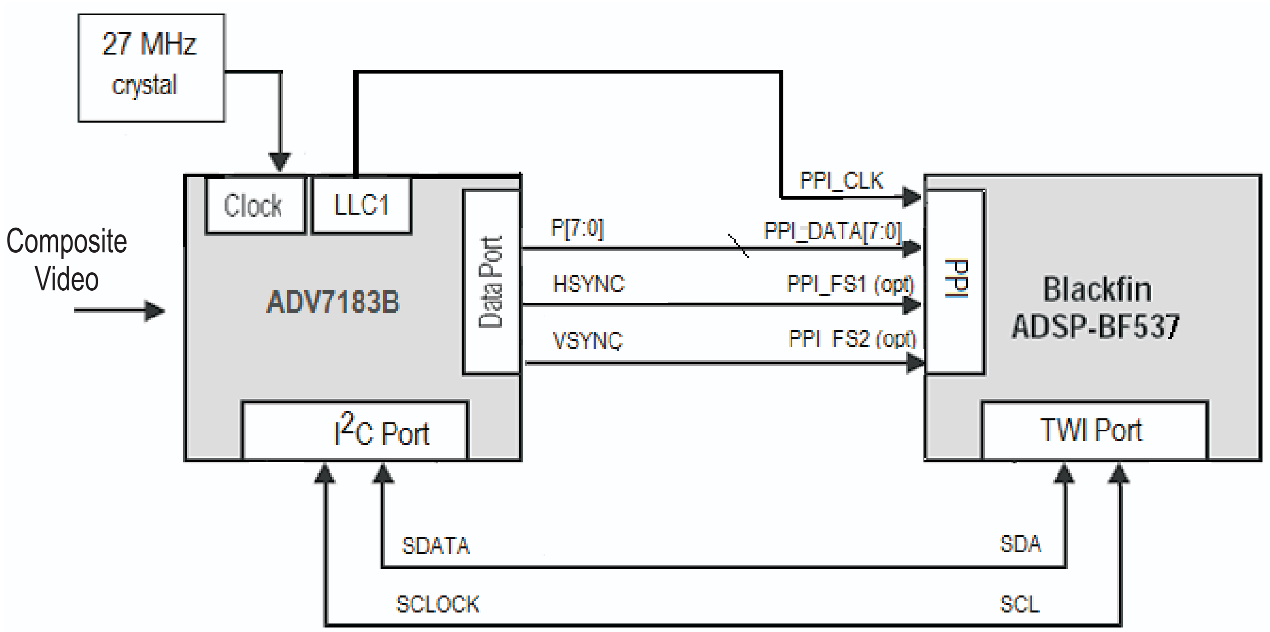

Figure 2 – Composite video signal processing.

3. OPERATION SYSTEM

For the purpose of performing its functions, the system should be provided with proper software. The software consists of various user applications for all the intended tasks and a suitable operation system to support them (fig. 3). For an operation system is chosen a special Linux distribution for microcontrollers – uClinux. uClinux is ported to many microcontroller families, among which is Analog Devices Blackfin DSP. This fact had a significant influence on the decision to choose BF531/2/3 for developing this system. Among the evident benefits of using an operation system in an embedded system, such as: fast application development, usage of control structures free of bugs, usage of pre-built sources, hardware abstraction, module division of the applications, multitasking, thread priorities and resource allocation, there are also some drawbacks:

- **memory consumption** – the minimum memory requirements for uClinux are 4-8MB SDRAM and 2MB Flash.
- **boot delay** – the kernel is fast, but sometimes not fast enough – expect to have a 2-3 second boot time.
- **interrupt latency** – on occasions, a uClinux device driver, or even the kernel will disable interrupts. Some critical kernel operations can not be interrupted, and it is unfortunate, but interrupts must be turned off for a bit. Care has been taken to keep critical regions as short as possible as they cause increased and variable interrupt latency.
- **robustness** – although a kernel has gone through lots of testing, and many people are using it - it is always possible that there are some undiscovered issues.

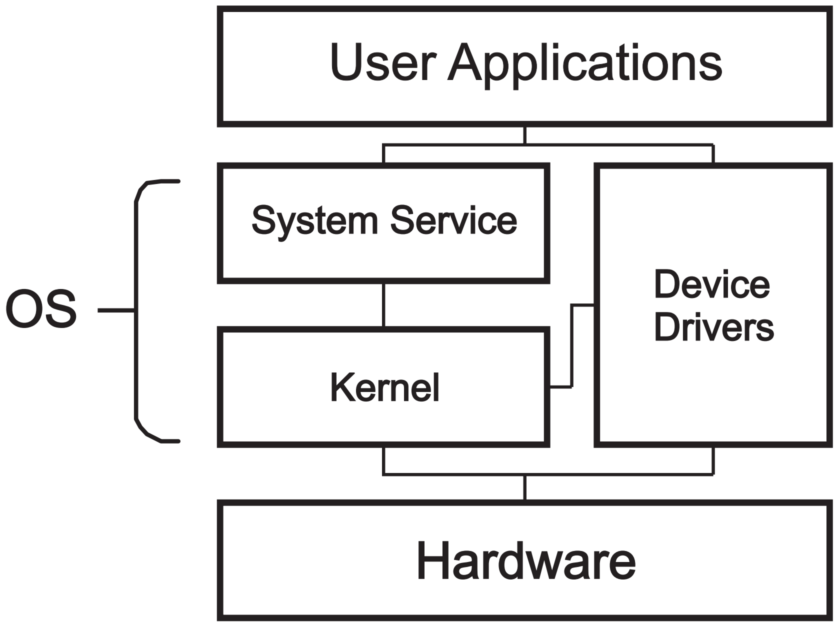

Figure 3 – Structure of an Embedded System with an Operating System.

To get started with the uClinux, one should obtain the full uClinux distribution sources and build them for the target processor. The sources come in a rather large file that provides 2.0, 2.4 and 2.6 kernels along with a good selection of applications, most of which have been ported to uClinux. There are two more ways to install uClinux to the board. The first one is to use precompiled binary files, built for the same processor and the other is to use precompiled binaries in RPM package manager. If the settings of the precompiled binaries are not satisfying and a customization is needed, one should rebuild the system from source for board's specific requirements. The entire build process consists in:

- installing a proper version of tool chain for the uClinux kernel
- configuring the kernel and the applications for the specific needs
- building the image
- downloading the image to the board
- booting

The configuration process is the most important portion of the kernel rebuild process. In this step one decides which features will be included in the final kernel and it can require lots of hardware knowledge. The current kernels have graphical configuration programs and provide help screens with tips for most of the configuration options. The results of the build process may vary, depending on the concrete settings made during configuration. At last, there should be image files of different types, containing the root file system, bootable objects and compressed bootable images. All these files are downloaded to the target board and stored in its FLASH memory.

4. BOOTLOADER

The "Das U-Boot" Universal Bootloader project provides firmware for many CPU architectures and boards, including many Blackfin platforms with full source code under GPL. U-Boot is a boot loader program that is stored in FLASH memory on the target system. U-Boot allows the target to load an ELF memory image from a serial connection, a network connection, or FLASH memory. When U-Boot is stored in FLASH memory on the target system it can be loaded on system reset, U-Boot can then obtain a uClinux memory image and boot uClinux.

This bootloader automatically boots the system on power up or reset of the board after the timer counts down. The countdown timer is setup with a default time of 30 seconds. If a key is pressed before this time the autoboot procedure halts and the user is brought to the U-Boot command line. A number of OS loading commands are available that can be executed from the U-Boot command line. U-Boot allows itself to be upgraded with a new version from within U-Boot. JTAG or ICE connection is not necessary in this case. U-Boot has support for the common network commands ping, tftp, and dhcp which makes it extremely suitable for loading images through the network. A memory image can also be downloaded through the serial port onto the target system. U-boot can be configured to boot either from NAND flash (fig. 4) or parallel NOR or serial (4-wire SPI-compatible interface) NOR flash devices.

Using U-Boot requires compiling it from source, configuring it for the specific requirements of the target board. There are also pre-build images that can be used only if the hardware resembles the features of the commercial kits.

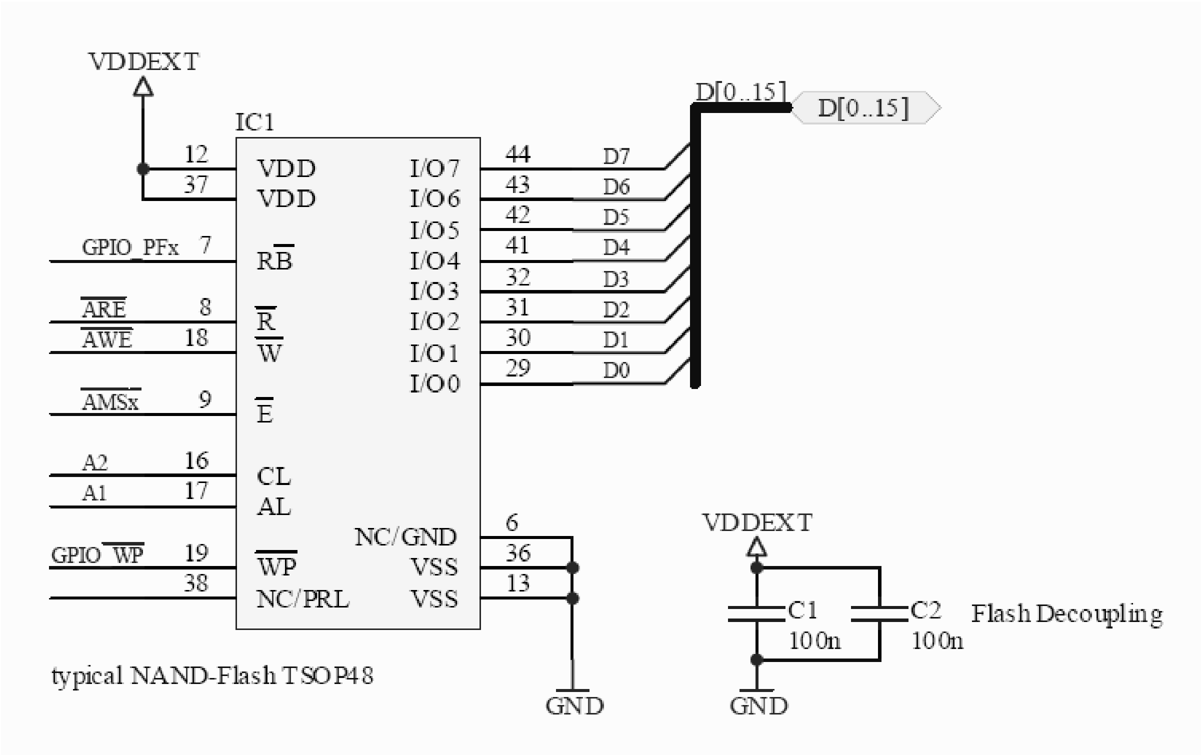

Figure 4 – Example Interfacing a NAND Flash to Blackfin's EBIU.

5. Conclusions

The system benefits the features of both the hardware (BF531/2/3) and the software (uClinux) embedded. These are, in particular: large variety of peripherals, high computation performance, reasonably low power consumption, low cost of system build-up, capability to support a full featured Linux distribution and all the advantages of using that embedded OS.

Along all the obvious benefits of using Linux as an operation system, there are also some drawbacks of using kernel in embedded systems: memory consumption, boot delay, interrupt latency, robustness. Despite these drawbacks, this combination of hardware and software gives this project universality and applicability in unlimited varieties of tasks, as an embedded solution.

6. References

[1] Oppenheim, A., A. Willsky, I. Young. *Signals and Systems.* Prince-Hall, Inc., 1983

[2] Gan, W., S. Kuo. *Embedded Signal Processing with the Micro Signal Architecture.* John Wiley & Sons, 2007.

[3] Katz, D., R. Gentile. *Embedded Media Processing.* Elsevier Ltd, 2007.

[4] Raghavan, P., A. Lad. *Embedded Linux System Design and Development.* CRC Press, 2005.

[5] Analog Devices – http://www.analog.com/

[6] Linux on the Blackfin Processor – http://blackfin.uclinux.org/gf/

[7] Linux on the Blackfin Processor Wiki – http://docs.blackfin.uclinux.org/doku.php

[8] Embedded Linux Microcontroller Project – http://www.uclinux.org/index.html

[9] Skyeye Open Source Simulator –http://www.skyeye.org/index.shtml