EXPERIENCE IN USING CREDIT CARD SIZE BOARDS BASED ON COLDFIRE MICROPROCESSORS AND RUNNING UNDER µCLINUX

Giorgio Bassato, INFN, Laboratori Nazionali di Legnaro, Padova, Italy
Weidong Zou, China Institute of Atomic Energy, Beijing, China

Abstract

The availability of microcontrollers with a high density of integrated peripherals and dissipating an extremely low power has made possible the realization of micro-computer boards with a size significantly smaller than the consolidated standards like the PC104 or PrPMC. One of emerging product lines of such generation of ICs is the family of Coldfire microprocessors (by Freescale)[1].
We tested a commercial board produced by SenTec Elektronik[2] (Germany) based on the MCF5329 processor and running under µClinux to evaluate if it could be used in the design of a new resonator controller for our superconducting Linac. This paper describes the main characteristics of this device and reports some considerations concerning the test of a prototype board.

THE MCF5329 ARCHITECTURE

The MCF5329 belongs to the mid-range class (V3) of Coldfire processors. At 240 MHz of core clock, it features about 210 MIPS (Dhrystone 2.1) dissipating just 0.25W. The processor includes an arithmetic unit with hardware support for integer Divide and Multiply-Accumulate (MAC) operations and has an instruction pipeline with branch prediction capability. The chip has an internal cache memory of 16 KB and a 32 KB static RAM. As most of V3 family devices, the processor includes a Fast Ethernet Controller (FEC) and a high number of communication peripherals: three UARTS, one USB, one CANbus, one QSPI (Queued Serial Protocol Interface) and one I²C bus controller. Moreover, MCF5329 has the provision (quite unique for this class of components) of driving a low resolution LCD, that makes it extremely suitable for the design of portable instruments. Specifically designed for the embedded equipment market, the philosophy underlying the Coldfire processors architecture is the reduction to the bare minimum of requirements for external hardware. In fact, for most applications, the only components that need to be added are a clock generator, a SDRAM (the controller is included in the chip) and a FLASH memory to load the software. Of course, since the electrical levels at the I/O pins are LVT compatible (3.3 V), the system designer will have to add the physical interface drivers for the communication ports being used. To simplify the integration with application specific hardware, the MCF5329 has a dedicated, 32 bit wide, data bus (FlexBus) to exchange data with external I/O devices.

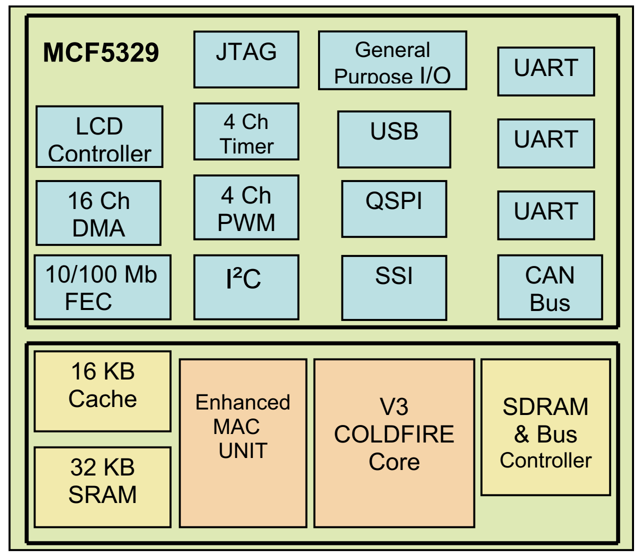

Figure 1: The MCF5329 block diagram.

THE SENTEC COBRA5329 BOARD

We chose for our evaluation the COBRA5329 because of its extremely compact size (just 48×68 mm.), the low power dissipation and the good performance over cost ratio. The module is easily pluggable on a daughter board thanks to the stackable, high density, HIROSE SMD connectors. The unit includes the CPU, running at 180 or 240 MHz, 16 MB of SDRAM, 16 MB of bank-selectable FLASH memory, the Ethernet physical interface (National DP83848), the clock generator together with the initial reset circuitry and a temperature sensor that is connected to the CPU through the I²C interface. A miniature DIP selector allows the user to choose the bank of FLASH to boot from; this is used to switch between the µClinux OS and the debugger software (the debugger is necessary to write an updated µClinux image into the FLASH). The evaluation hardware from Sentec includes a carrier board containing the power regulators, the Ethernet line transformers with the RJ45 connector, the RS232 drivers with two DB connectors and some utility LEDs to monitor the board activity. An MMC card interface is also available but we didn't use it in our application.

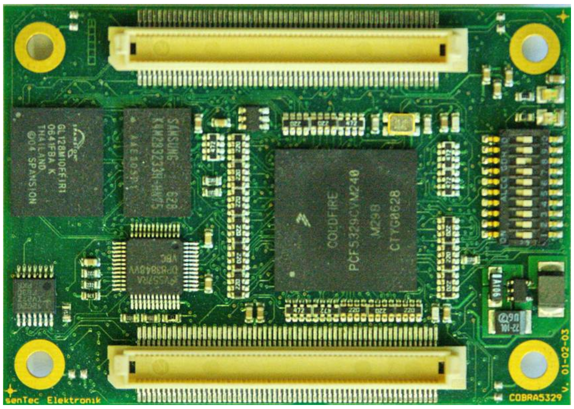

Figure 2: Picture of COBRA5329 board.

The couple of 120 pin SMD connectors carry out the signals associated to the internal peripherals and all the processor signals required to interface the module to an user designed hardware. In particular, other than the address and data bus lines (and related control signals), it is worthy to mention a number of configurable FlexBus Chip Select (FB_CSx) signals that make really straightforward to map an external register over the processor I/O space .

One key feature of the COBRA5329 is the software support. The module comes equipped with a small footprint Linux version, particularly suitable for embedded application: the μCLinux OS, developed by Emlix [3]. The next paragraph will describe the main characteristics of this operating system.

THE μCLINUX OS

Up to a few years ago, Linux couldn't be taken into serious consideration for embedded control applications. The reason is quite obvious: a fully featured system needs a large memory and a hard disk where a file system must be created. Moreover, supporting a multi-user environment requires a quite complex mechanism of memory management to avoid conflicts on resources usage. Today the scenario of embedded software market has completely changed and some companies offer a version of Linux extremely reduced in size but still maintaining the features (i.e the basic network protocols) that make it ideal to support the design of a networked equipment. The μCLinux by Emlix is one of the leading products specifically realized for the market of embedded devices.

The key characteristics of μCLinux can be summarized as follows:

- the OS can be stored in a memory space as little as 2 MB
- there is no memory management
- a unique user is configured (with root privileges)
- a Unix compliant file system is created on RAM
- the I/O system is compatible with standard Unix ioctl() functions.

The NFS client protocol is implemented in the standard OS configuration: this allows to mount, on a local directory, the file system of a remote server that can be used for the application development.

This is an essential feature of μCLinux: in fact, due to the limited amount of available memory, the development utilities (like GNU cross-compilers) must be installed on a remote host, typically another Linux PC with adequate RAM and disk space.

A few hints should be given on the development cycle. It is worthy to note that the developer cannot install his new application in the FLASH by just overwriting it: adding an application requires rebuilding the μCLinux image and this process can only be carried out through a chain of "makefile" scripts provided in the development environment. In practice, on the host side, the user can work on a file system (called "romfs") that is the exact copy of what will be deployed to the target. In this copy, the user will have to install the source code of his application and modify the proper "makefile" to reflect this change. Once a new OS image has been created, the on-board debugger can be used to download and write it to the FLASH.

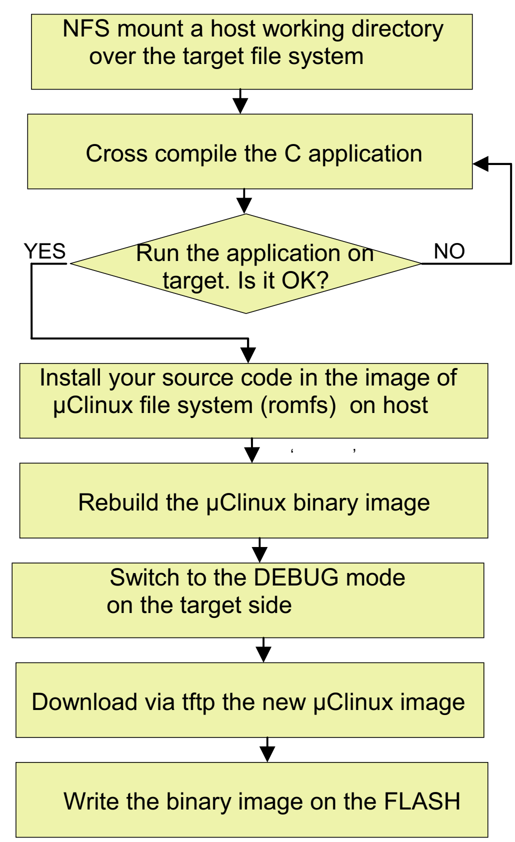

Figure 3: The software development flow.

A SIMPLE CONTROL APPLICATION

We evaluated the COBRA5329 board to verify if it could be used as processing unit in the design of a resonator controller to be installed in the RF system of our superconducting Linac. The operation of this controller is essentially based on an analogue feedback and the main purpose of the microcomputer is to receive from the operator console the parameters required to setup phase and amplitude in a resonant cavity. Many of the RF components mounted on the main board are digitally controlled or are set through a DAC; therefore, setting up the controller operation just means to transfer some binary patterns to the input of RF devices and read back the values of some variables.

The communication between the controller and the supervisor computer has to be accomplished through a RS232 interface for compatibility reasons with the previous controller version, but should be replaceable by an Ethernet link in the next Linac upgrade project. The controller model can be described in terms of a dozen of memory mapped registers whose width can vary from 8 to 16 bits (for simplicity we assume that all registers are 16 bit wide). From the software point of view, the application is really straightforward: all registers can be accessed from a C language application by just defining a pointer to them; no pseudo device drivers (i.e. /dev/mem) need to be used because µCLinux doesn't support memory protection mechanisms.

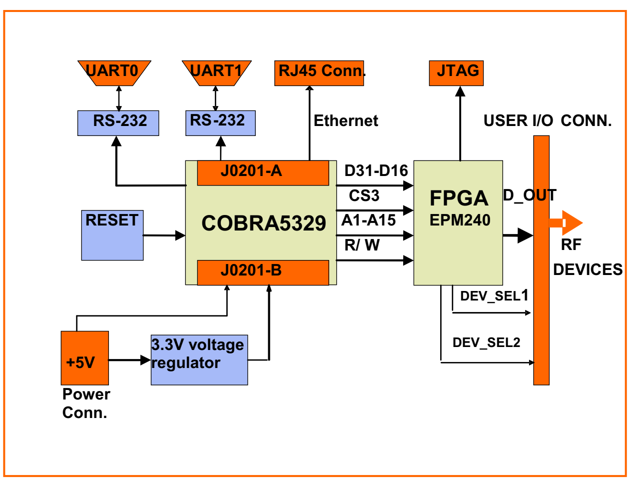

Figure 4: Basic controller layout

INTERFACING THE USER HARDWARE

The most simple way to extend the I/O capability of the board is to use a 3.3V compliant FPGA. An Altera MAXII (EPM240T100) was used to deploy the registers necessary to setup a group of RF devices. We decided to assign to these registers a block of addresses associated to one of the programmable FlexBus "chip select" signals (FB_CS3). The FPGA was designed in such a manner that its internal structure can be replicated to implement as many registers as the application demands. No other components were added, a part the Ethernet and the RS232 drivers and connectors.

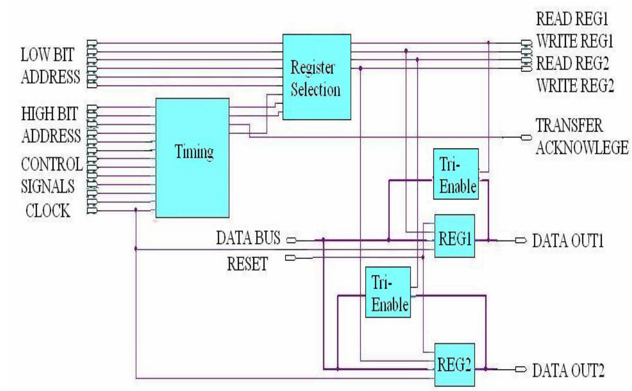

Figure 5: FPGA configuration.

CONCLUSIONS

µCLinux can represent, for the applications where the time response is not a critical issue, a good alternative to blazoned operating systems like Vxworks. Maintaining a file system structure on a memory based system is not a real advantage when dealing with tasks not requiring a local storage, but can be of help for developers familiar with a disk based Linux. The software tools provided with the system work well and, thanks to the NFS support, allow to develop and deploy a simple application in a relatively short time. We didn't try to develop a driver for a specific device: we can guess that, in this case, some support from Emlix would be necessary. For sake of completeness, the amount and quality of development tools provided by Wind River with Vxworks is clearly superior to those available with µCLinux, but also the cost is not comparable. The COBRA5329 is a powerful and cost effective board that finds an ideal application where the size and the dissipated power are a real constraint. In combination with µCLinux, it is an excellent platform for the design of an embedded controller.

REFERENCES

[1] http://www.freescale.com/coldfire
[2] http://www.sentec-elektronik.de
[3] http://www.emlix.com