Reprint

The C2 Communication Coprocessor

D. Pelekanos, T. Antonakopoulos and V. Makios

The 5th International Conference on Advances in Communication and Control – COMCON'5

CHANIA, JUNE 1995

THE C^2 COMMUNICATION COPROCESSOR

D. Pelekanos, T. Antonakopoulos and V. Makios

Telecommunications and Technology of Information Division
Department of Electrical Engineering and Computers Technology
University of Patras, 26500 Patras, Greece
Tel: +30 (61) 997 286, Fax: +30 (61) 997 342

Abstract

The C^2 Communication Coprocessor is the basic unit of a distributed communication system, incorporating wireless, power line and telephone lines transmissions. The C^2 Communication Coprocessor consists of a set of reconfigurable processing elements, capable to perform different functions in different configurations. The architecture of the C^2 Communication Coprocessor can be easily implemented in a single ASIC, by using multiple commercially available CPU cores, for satisfying various communication requirements.

I. INTRODUCTION

The problem of interconnecting equipment and devices belonging into different communication systems has been addressed extensively in the past years [1]. The 'global communication system' includes medium and high-speed local area networks, wide area networks, wireless networks etc. The interfacing of different communication systems is mainly implemented by interconnecting central switching facilities, although another, probably distributed, approach has to be used for satisfying the required flexibility. Wireless networks, like CDPD [2] and Mobitex [3], offer a wide range of services and the required user mobility. This is performed by connecting their base stations with nodes of public communication systems. The available solutions provide a wide range of functions and services, but new research directions with more flexible architecture and easier to integrate are under consideration [4]. Especially, the introduction of Personal Communication Services will expand the flexibility and functionality of notebooks, personal assistance devices etc., thus enlarging an already big market [5]. Such a system requires that a new communication infrastructure is installed and new operational procedures have to be determined. The cost and time required for such a development is not affordable for a number of users who require a limited set of such functions with the minimum installation cost.

These users already use telephone line modems for data transfer and a lot of their application software has been based on this capability. They use either common telephone

lines or dedicated leased lines and their protocol stack has been based on a connection oriented service at the lower layers. The draw-back of such a solution is that, it requires a telephone socket for accessing the network, thus minimizing its flexibility. This disadvantage can be overcome by extending the accessibility of a specific telephone socket, by using various transmission media, like power lines and RF. Such a network should combine existing public switching telephone networks with wireless networks and power lines system, providing to the user a unified communication system.

Section II presents the system architecture and describes the network functions. In section III, the basic unit of the proposed configuration, the C^2 Communication Coprocessor, is described in terms of hardware modules and flow of data. The interconnection unit of the C^2 Communication Coprocessor is described in section IV.

II. System Architecture

The proposed network architecture is shown in Figure 1. The network is composed of PSTNs, power line networks and wireless networks. There are three different types of communication devices, 'intelligent modems', which can be combined to provide various configurations. These communication devices are directly interconnected and no switching system is required in between. The PSTN modems are commercially available devices, while the other two types, the wireless modem and the power line modem, implement more sophisticated communication functions, thus allowing the establishment of user requested connections without using any central switching equipment. All intelligent modems, which have been developed using the C^2 Communication Coprocessor, have the same user interface with the PSTN interface, that is based on the AT commands de facto standard [2].

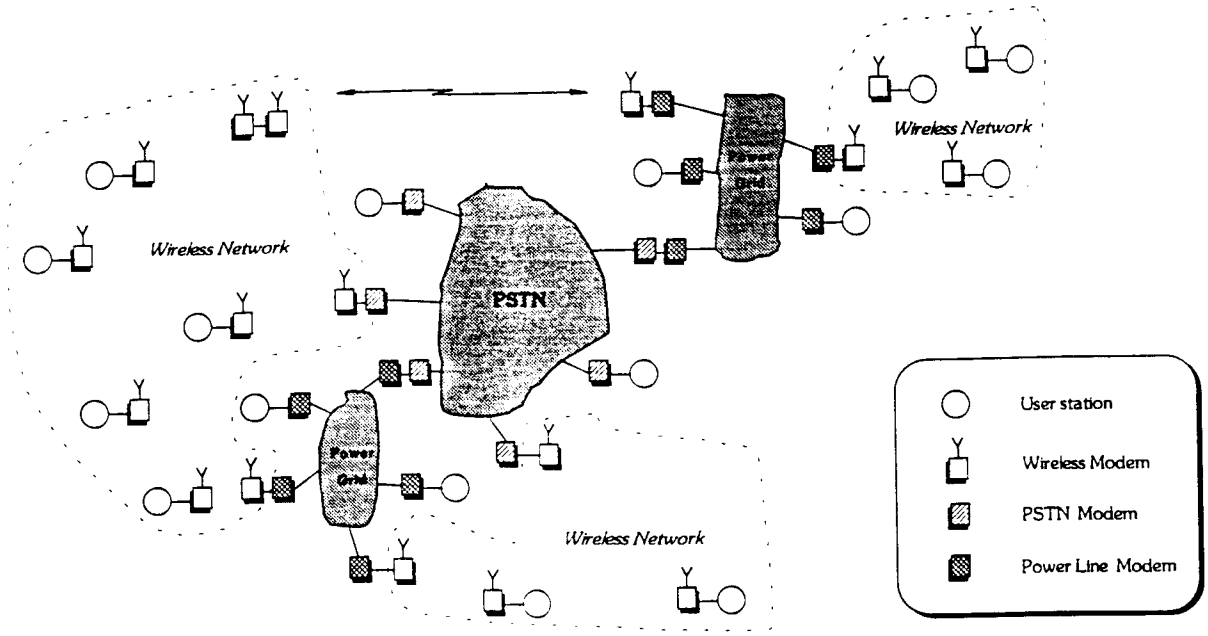

Fig. 1 The Network Architecture.

Each intelligent modem has a user configurable address which is unique in the respective subnetwork and becomes unique in the whole network architecture, when it is combined with the subnetwork unique address. The address of each subnetwork is defined by the

respective PSTN address (telephone socket), where the subnetwork is connected. If a subnetwork is not directly connected to the PSTN, then its address is defined by the combination of the address of the intelligent modem, that interconnects this subnetwork with a subnetwork that is connected to the PSTN, and the address of the attached subnetwork. In case there are two or more alternative routes for interconnecting two end-systems, the user originated the connection has to determine the complete path. If a part of a specified route has been allocated to another connection, the connection in-progress is interrupted and the user has to try again via the same or another route.

The network has been based on the ISO-RM communication protocol stack. At the physical layer, power line or RF transceivers are used. The subnetworks use a network protocol for single, group and broadcasting message delivery. The transport protocols ensure reliable message delivery with different message types and characteristics. The subnetwork protocols also provide management and diagnostic functions and message authentication. The system services provided are distinguished into three categories:

- Application services,
- Management services and
- Configuration services.

The application services use a set of protocols for supporting widely used applications. These services include commonly used services like connection establishment, file transfer, terminal access etc. and custom configured services, like collection of sensorial data. The management services include security services, medium access and transmission services, while the configuration services include set-up of protocol parameters, testing procedures etc.

The used protocol stack allows the implementation of services for custom applications. During connection set-up, protocol dependent parameters are negotiated and thus, reliable, transparent data transfer is achieved. Each intelligent modem supports a small set of routing functions and the number of concurrent connections via a single modem is determined by the modem processing power and the available memory.

Although the system is based on connection oriented data transfers, the wireless network and the power lines network use a multi-access protocol, based on the CSMA protocol, which is upgraded to the connection oriented service by the data link and the transport protocols. Whenever a connection is established between two adjacent nodes, a connection monitoring process is activated in both stations. In this case, if one of the nodes does not respond to periodically generated maintenance commands, a 'hang-up' signal is generated to the respective user.

III. THE C^2 ARCHITECTURE

The C^2 Communication Coprocessor is the basic processing unit of the previously described communication system. It consists of a number of microcontrollers which are connected via bus bridge devices and multiple-port RAMs. The C^2 Communication Coprocessor has been designed so that different processing elements are implemented on different CPUs and the firmware partitioning has been performed using the protocol complexity. In the architecture presented in Figure 2, there are three CPUs which exchange data via a Four-Port RAM and control information via three Bus Bridges. The CPU that includes the debugging port is the system scheduler, which co-ordinates the flow of information between the user interface and the network interface. The debugging port is used mainly during the system development life-cycle and provides a unified way to download programs to the different CPUs, to co-ordinate their program execution and to perform simple debugging functions, like display of registers, set of break points etc.

During normal system operation, the CPU attached to the debugging port is used for executing the network protocols and for that reason this is a 16-bit CPU. One of the CPUs that have an expansion port implements the user interface. Although the user interface is very simple, an RS-232C interface, this CPU implements the extended AT-commands set and fully emulates the performance of a telephone line modem. The other CPU is used for implementing the medium access protocol and its expansion port is used for interfacing dedicated hardware, like front-end processors prototyped using FPGAs, and for controlling other devices like synthesized RF transceivers.

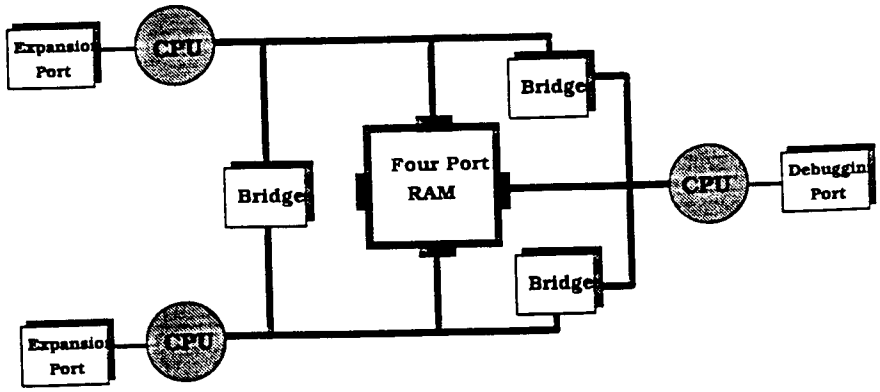

Fig. 2 The C^2 Communication Coprocessor

The C^2 Communication Coprocessor is flexible enough, so different functional devices can be easily implemented. For example, the C^2 Communication Coprocessor can be used for implementing intelligent bridges between homogeneous or heterogeneous networks, by attaching the interface of each network to one of the two available expansion ports. In this case the two 8-bit CPUs implement the two access protocols, while the intermediate CPU implements the relay functions.

When more processing functions have to be performed and more power is required, different system architectures can be implemented using multiple-port RAMs and bus bridge devices. Figure 3 shows an example of such a system which contains five CPUs. The medium access control and the application dependent sections are single CPU modules, while the protocol processing is performed by three CPUs. The central CPU organizes the protocol processing, while the other two CPUs implement the protocol processing (encoding/decoding). The central CPU is informed on the processing status of each 'protocol' CPU and shares the processing load accordingly. The dual-port RAMs are used for storing the data of the different processing stages and decouple the data flow of these stages.

In the proposed network, due to the transmission conditions, the length of transmitted packets is small compared to other LANs, thus the processing time of each packet becomes comparable to the packet transfer inside the C^2 Communication Coprocessor. So the multiple data transfers do not decrease the system performance, while they allow easier implementation and system reconfigurability.

The C^2 Communication Coprocessor has been implemented using different types of MCUs. The CPUs attached to the expansion ports are 8-bit MC68HC11F1 MCUs, while the processor for protocol is the 16-bit MC68HC16Z1 MCU. The MC68HC11F1 [6] is a low-power, high-speed MCU, having a non-multiplexed bus and operates up to a nominal bus speed of 4 MHz. It contains A/D converters, multiple I/O ports supporting timing related functions, timer subsystem for protocol related timing functions, serial I/O, EEPROM etc. The MC68HC16Z1 [7] is the upgrade of the MC68HC11 family in the 16-bit area, which

operates at 16 MHz and provides multiple system integration functions with enhanced communication facilities.

The Bus Bridge, which is extensively described in the next section, has been implemented in an FPGA and provides an effective interface for interconnecting MCUs. Its architecture is based on two FIFOs and on a set of control and status registers with interrupt capabilities.

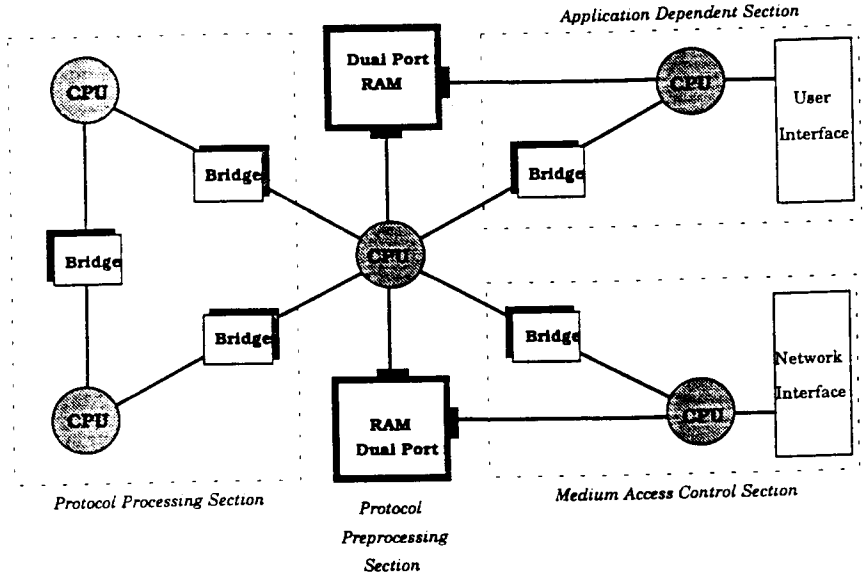

Fig. 3 A high processing power architecture.

The C2 Communication Coprocessor uses the RTXC real-time kernel as its operating system [8]. The RTXC is a multitasking kernel which manages tasks and timing events, transfers blocks of data between tasks, manages the use of system RAM, defines exclusive access to system resources and allows the inclusion of specific device drivers for system peripherals. A device driver has been developed for accessing the bus bridge registers and for exchanging information using the internal FIFOs and special device drivers have also been used for controlling the user interface and the expansion ports. All these drivers are interrupt driven tasks.

IV. The Bus Bridge and External Interfaces

The Bus Bridge provides a simple but effective interface for interconnecting microcontrollers designed for single processor applications. The Bus Bridge supports slave type accesses for either Intel type or Motorola type busses and its architecture is based on two FIFOs and on a set of control and status registers with interrupt capabilities.

The Bus Bridge block diagram is shown in Figure 4. Each microcontroller sets its interface preferences (like sources of interrupt) by setting the respective bits of the control registers and recognizes the cause of an interrupt by reading the status registers. The Bus Bridge allows either the implementation of a byte-level exchange mechanism or transfer of blocks of data, depending on the FIFOs length. In this case, a command header precedes the transfer of any data block. The Bus Bridge architecture allows full duplex data transfer in both directions.

A very important feature of the Bus Bridge architecture is that it can support, in co-operation with firmware routines, debugging functions of a multiprocessor environment via a single interface. Debugging is generally implemented in software or using CPU microcode and its basic features include viewing and altering of registers and memory positions, and perform of special test features. During debugging, normal Bus Bridge handling is suspended and special microcode has to be used for performing the debugging functions. During debugging mode, no interrupt associated with the normal Bus Bridge function is allowed to apply.

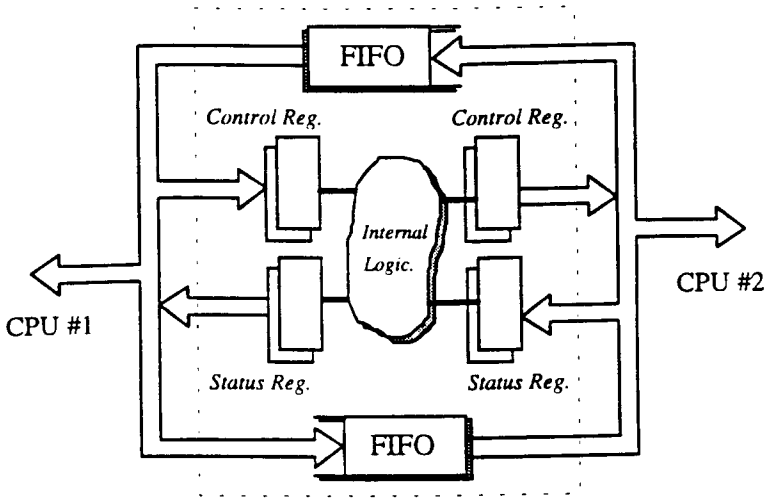

Fig. 4. The bus bridge architecture

The Power Line Interface implements the node physical layer and has been based on commercially available power line transceivers with the appropriate coupling circuit. The power line transceivers use a direct sequence spread spectrum (DS-SS) technique for interfacing the error-prone communication channel [9]. The generated spectrum is between 100kHz and 450kHz and a 10Kbps transmission rate is achieved. Figure 5 shows the power line interface architecture. The line filter and coupling network adapts the AC line to the transceiver characteristics and allows the signalling on the power line. The Power Line Framer is a processing unit, implemented using an FPGA, which adapts the power line transceiver serial interface to the Expansion Port interface. This unit adds the requested preamble and synchronization bytes, implements low-level functions of the access protocol and supports frame extraction at the receiving side.

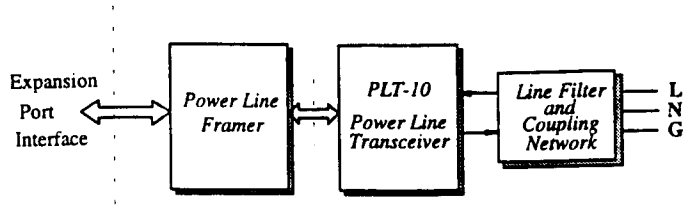

Figure 5. The power line interface

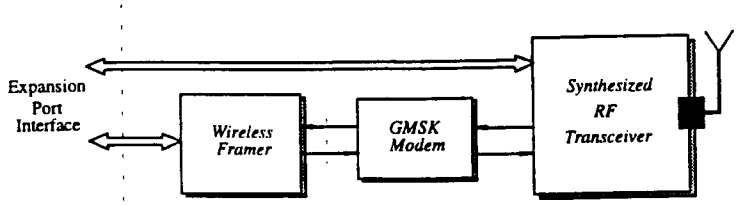

Figure 6. The wireless interface

V. CONCLUSIONS

A communication system which combines PSTNs with wireless networks and power line systems, providing a standard interface to the end user has been presented. The network uses the C^2 Communication Coprocessor as its basic unit, which consists of a set of reconfigurable processing elements, for performing the required functions. The C^2 Communication Coprocessor uses also a common interface for accessing different communication media and it can be easily implemented in a single integrated circuit, by combining commercially available CPU cores with the proposed bus bridge interface module.

REFERENCES

[1] B. N. Jain and A.K. Agrawala: "Open Systems Interconnection: Its Architecture and Protocols", Elsevier Science Publishers, The Netherlands, 1990.

[2] CDPD Industries: "Cellular Digital Packet Data System Specification", Release 1.0, July 19, 1993

[3] RAM Mobile Inc.: "Mobitex Interface Specification", 1992

[4] Advanced Communications Technologies and Services - ACTS: "Call for Proposals", European Commission, DGXIII, August 1994.

[5] Ron Schneiderman, "Wireless Personal Communications", IEEE Press, New York, 1994.

[6] Motorola Inc.: "MC68HC11F1 Technical Data", 1989.

[7] Motorola Inc.: "MC68HC16Z1 Technical Data", 1992.

[8] Embedded Systems Inc.: "RTXC Real-time kernel", 1994.

[9] Echelon Corp.: "PLT-10 Power line transceiver", April 1993.

[10] Bud Simciak and Sam Rizk: "GMSK for High Speed Wireless Modems", MX-COM Inc., 1994.