Reprint

Bridge-on-a-chip: Internetworking ATM with the IEEE802.11 Wireless LAN

M. Iliopoulos, A. Maniatopoulos and T. Antonakopoulos

The European Conference on Networks and Optical Communications – NOC'98

MANCHESTER, UK, JUNE 1998

Bridge-on-a-Chip: Inter-networking ATM with the IEEE802.11 Wireless LAN

M. Iliopoulos[1], A. Maniatopoulos[1] and T. Antonakopoulos[2]

[1] *DCT Hellas, Navmahias Ellis 1, 26441 Patras, Greece*

[2] *University of Patras, Department. of Electrical Engineering, 26500 Rio - Patras, Greece*

Abstract: Inter-networking ATM with legacy LANs requires dynamic and transparent connectivity, flexibility on resources utilization, quality of service support and effective network management. Inter-networking devices must be characterized by low operational costs and the best possible price-to-performance ratio. This paper presents a "Bridge-on-a-Chip" solution for inter-networking ATM networks with IEEE802.11 wireless LANs.

The bridge architecture is based on the ARM/Thumb RISC processor and uses a multi-master configuration for supporting the two network interfaces, the bridge relay functions and the network management functions. The bridge microcode implements LAN Emulation (LANE) and Multiprotocol over ATM (MPOA), for supporting bridging functions over legacy protocols in ATM, as well as logical grouping of users independent of their physical location, as presented in the virtual workgroup concept.

1. Introduction

Wireless LANs are becoming very popular because of their inherent flexibility and ease of use. The recently approved IEEE 802.11 Standard [1] for Wireless Network Medium Access Control (MAC) and Physical Layer Specification (PHY), allows vendors to develop products as interoperable as today's wired LAN products, based on accepted industry standards. This will expand the use of wireless networks in a variety of commercial and industrial applications. Wireless LAN products that support IEEE 802.11 standard have a tremendous market potential, as they support high data rates, work in the world-wide license free 2.4 ISM band, and have a high performance in terms of range and power consumption.

On the other hand the ATM network, which is considered the emerging technology for Broadband-ISDN, exploits the vast base of existing LAN application software by LAN Emulation (LANE) [2], that emulates services of existing LAN's across an ATM network. The inter-networking of Wireless LAN and ATM networks using LAN emulation, allows dynamic connectivity, flexibility and quality of service support to user groups independent of their physical location. This work presents a "Bridge-on-a-Chip" solution for inter-networking ATM networks with IEEE802.11 wireless LANs and describes in details the architecture of such a device.

This work has been partially supported by the ESPRIT/OMI "VNET: Design, Implementation and Demonstration of an Adapter Card for Wireless Local Area Networks - Project No 23631" and "JEDI-FIRE - Project No 25530" projects of the European Union.

2. The Networking Environment

Wireless to ATM Bridge (WAAB) is the inter-networking device for interconnecting a Wireless LAN with other Wireless LANs, legacy LANs and ATM nodes over the ATM infrastructure. As it is shown in Fig. 1, there are three different scenarios for such communications. The first scenario supports communication between nodes attached to different WLANs, while the second scenario is used for exchanging information between mobile users and ATM native nodes. The last scenario is used for providing transparent communication between nodes attached to legacy LANs and the WLAN.

The WAAB, which acts as an Access Point (AP) to the WLAN, receives the packets that are destined outside the WLAN and forwards them to the proper ATM destinations (VCs) by using either LAN Emulation functions (LANE) or IP over ATM. In case the WLAN user is mobile, roaming functions are also supported at the WAAB bridges.

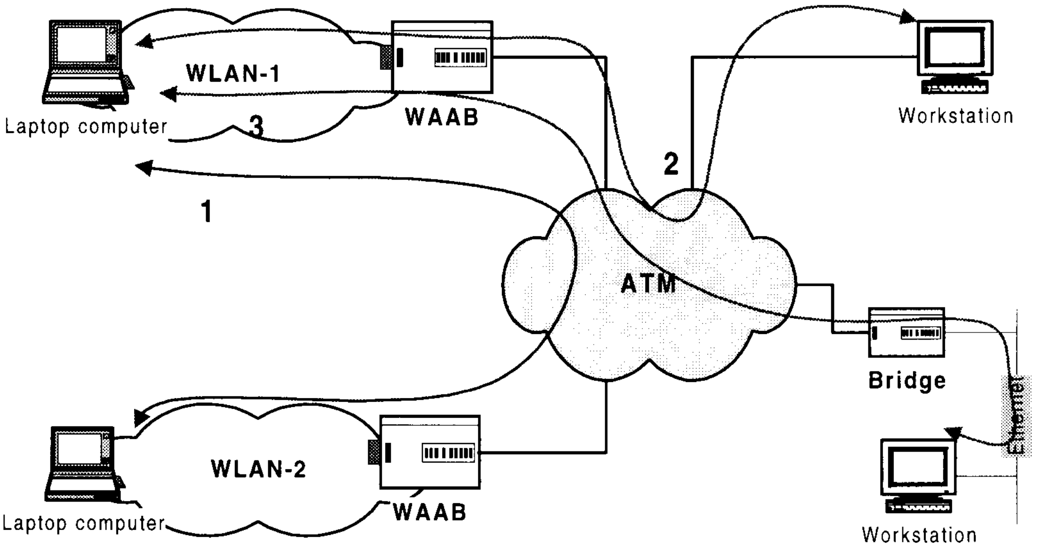

Figure 1: The WAAB inter-networking environment.

The data transactions over this unified environment are categorized according to the type of the end-to-end devices:

1. ***End stations transactions***: When two end stations communicate (irrespective to the type of network they belong), the inter-networking between the different networks should be transparent,
2. ***Inter-networking devices transactions***: The WAAB bridges exchange data for management and maintenance purposes (roaming, authentication etc.) and supports the IEEE802.1d protocol for being compatible with transparent bridges.

The WAAB bridge acts as an Access Point, implements all necessary communication protocols for supporting inter-networking functions such as LAN Emulation (LANE) and Multi-protocol over ATM (MPOA), implements logical grouping of users independent of their physical location and provides secure links by implementing encryption algorithms. Such a bridge can be implemented using either a set of commercially available network adapters and embedded systems architectures (like PC104*plus*), or using a custom architecture for minimizing its size and number of parts. The final bridge architecture was decided based on a number of system requirements.

The block diagram of the system that performs the inter-networking between ATM and IEEE802.11 wireless LAN is shown in Fig. 2. WAAB receives data from both networks, stores them locally for further processing, installs and maintains permanent and temporary VCCs and transmits the packets/cells to the proper destination. In order to implement such a device, a high processing power CPU is required along with the two network interfaces. At the WLAN side, there are two different physical layers (PHYs) described at the IEEE802.11 standard and there are some other PHYs under development for supporting higher speeds in a different frequency band. There is no standard interface for describing the WLAN MAC to physical layer interface, so a generic interface must be designed. At the ATM side, there is a large number of physical layers at a large number of transmission speeds, but all of them use the UTOPIA [3] interface for exchanging data with the ATM layer. All other functions at the two network interfaces are independent from their physical layers and they can be integrated into a single chip along with the main bridge functions. As it is shown in Fig. 2, the proposed bridge architecture is based on a bridge-on-a-chip module which performs all the inter-networking and bridging functions and three more modules, the ATM PHY, the Wireless PHY and the RAM modules, for allowing compact system implementation, flexibility for supporting almost all possible physical interfaces and thus reducing the system cost.

The objective of this "Bridge-on-a-Chip" solution is to create a compact wireless LAN to ATM bridge and to demonstrate its functionality and the potential for volume production of such a device. This is a hard problem to solve since these two networks are different in characteristics, while their applications have contradictory requirements. Expansion of the wireless network area coverage is possible by means of the so called (according to the 802.11 std) Access Points (AP). Multiple access points link the WLAN to the ATM and allow users to efficiently share network resources. The APs not only provide communication with the ATM network but also mediate wireless network traffic in their immediate neighborhood. Multiple access points/bridges can provide wireless coverage for an entire building or campus. Bridge functions and management is a challenging task due to the non-deterministic behavior of the traffic generally used in multimedia data communications with the mix of isochronous services.

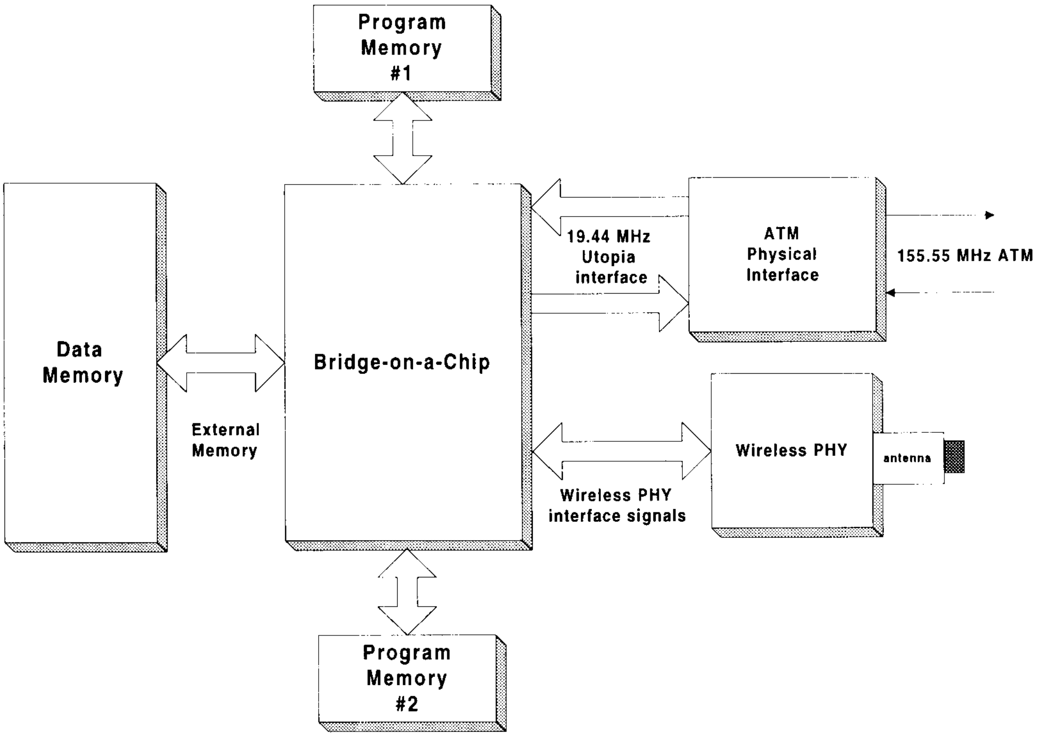

Figure 2: Wireless LAN to ATM Bridge

3. The “bridge-on-a-chip” architecture

Figure 3 shows the architecture of the ‘bridge-on-a-chip’. The architecture is based on the ARM processor [4] and its internal bus (AMBA) [5] functionality. It uses shared memory dual processor configuration with independent internal buses. The major architectural blocks of that bridge are the following:

- **The ATM-SAR Processing Unit (ASPU):** This block performs all necessary functions for interfacing with ATM PHY, such as VPI/VCI header processing, queue based Segmentation and Reassembly, Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Available Bit Rate (ABR) support through timers and rate matching units, and various other functions for ATM adaptation layer [6] support, e.g. CRC-32 checking for AAL5 support, CRC-10 checking for AAL3/4 support etc. As illustrated in this figure, the ASPU consists of two independent blocks, the Reassembly Processor (RP) and the Segmentation Processor (SP), that perform all the cell and data transfers to/from the memory using dedicated paths. Internal FIFOs offer temporal storage of cells until they are transferred to the common memory. All the configuration and control information is passed to the configuration registers (contained in the RP and SP modules), through inter-networking ASB (IWASB).
- **WLAN MAC Unit:** This block performs all the necessary functions for interfacing with Wireless PHYs, such as transparent Baseband Programming, TSF counter and CRC-32 support. It also performs all IEEE 802.11 MAC functions such as, Frame formatting , DCF, Fragmentation, Reassembly, RTS/ACK/CTS etc.
  The WLAN MAC Unit, is based on an independent ARM processor (WLANARM), which is supported by peripherals (e.g. timers, interrupt and reset controller etc.), and a Physical Attachment Interface unit (PAI) for communicating with the Wireless PHY device. This device uses an independent memory bus for program execution and exchanges information with other modules through common memory.
- **Inter-networking Unit**: This module performs all the bridge relay functions, network management operations and other high level inter-networking functions. The inter-networking unit is also based on an independent ARM processor (IWARM) with local execution RAM, which uses shared memory for exchanging information with other modules. The inter-networking unit configures and controls the ATM and WLAN units through ASB configuration registers.
- **Common Memory Interface Controller (CMIC):** This module offers arbitration between all system modules (SP, RP, IWARM and WLANARM) for accessing the common RAM. The memory controller communicates with the ARM cores through two independent ASB buses, and with the ASPU through reassembly and segmentation paths.

This ‘bridge-on-a-chip’ architecture has the following advantages:

- The ARM processors have independent internal buses, and dedicated program memory buses that allow the cores to execute instructions at full speed, whereas a shared internal bus or external memory bus would limit the performance (due to bus conflicts).
- The shared memory scheme has the advantage of single transfers from one network type (WLAN / ATM) to the other (ATM / WLAN), whereas a distributed memory scheme would cause duplication of data and hence wasting of memory space and bus time.
- The dedicated hardware for ATM SAR processing, which also accesses the shared memory, reduces the load of the inter-networking unit.
- Control and management is passed through inter-networking ASB making interoperability easier.

- The WLAN and inter-networking functions can be changed and updated to new requirements since they are implemented in micro-code which is executed by the ARM processors, whereas the ATM hardware uses a more custom architecture.

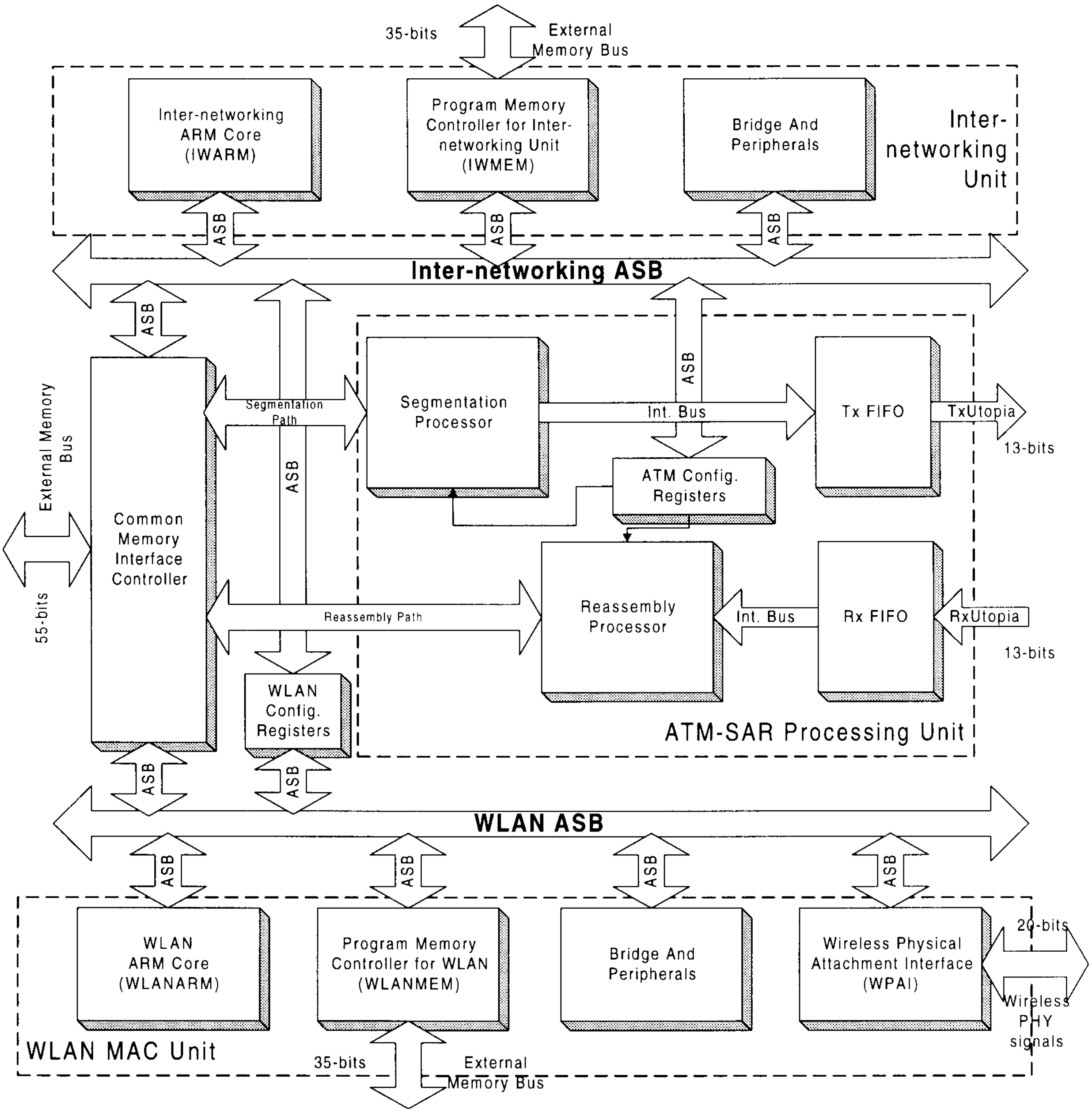

Figure 3: Bridge-on-a-chip Architecture

4. Bridge Internal Data Flow

This section presents the data flow inside the chip and describes how the various modules are used for supporting it. Figure 4 shows a detailed description of the data flow at the ATM to WLAN direction, while Figure 5 shows the data flow at the WLAN to ATM direction.

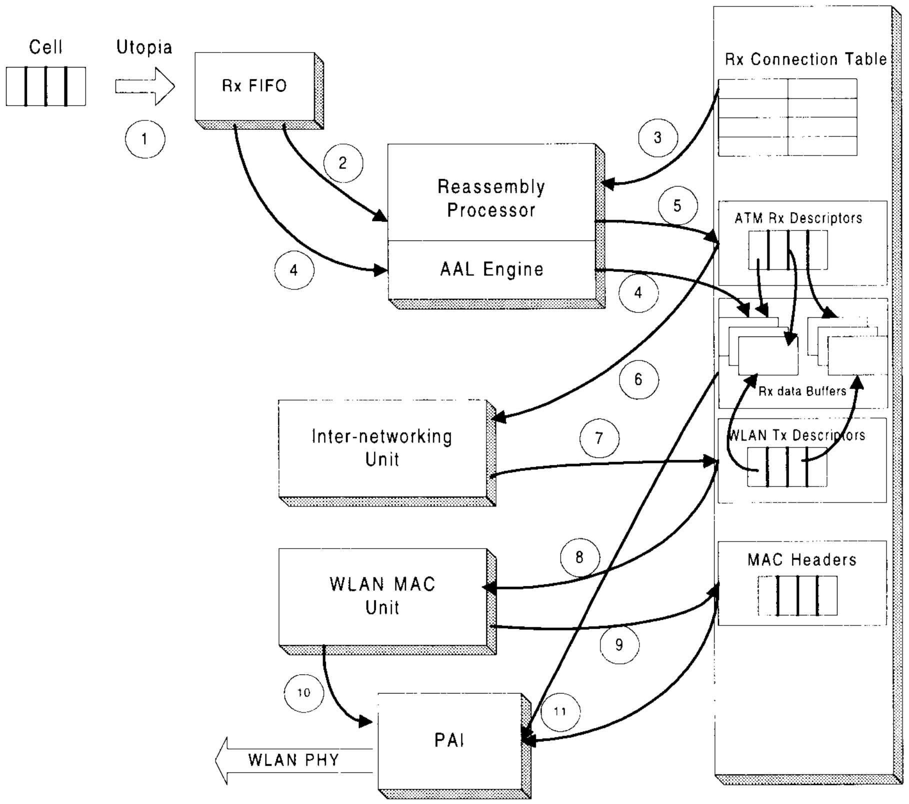

Figure 4: ATM Receive - WLAN Transmit Data flow

When a cell is received at the ATM side, the following steps are executed:

The cell is stored at the local Rx FIFO (step 1), and the Reassembly Processor (RP) reads the cell header and forms the VPI/VCI address (step 2). The RP reads the Rx Connection Table according to VPI/VCI address and checks for a valid connection. If the connection is valid, the RP reads the rest of the information needed for packet reassembly (step 3), and the AAL engine performs the appropriate AAL functions (CRC checking, cell sequence number checking etc.), and transfers the cell in the appropriate Rx data buffer (step 4). In step 5, the RP constructs the Rx Buffer descriptors in memory and informs the inter-networking unit that a packet has been successfully reassembled.

The inter-networking unit reads the Rx buffer descriptors, performs the appropriate bridging functions such as LAN emulation or Multiprotocol over ATM (step 6), and constructs the Tx buffer descriptors for WLAN transmit. It informs WLAN MAC Unit for packet transmission and passes a pointer to the Tx buffer descriptors (step 7).

The WLAN MAC Unit reads the Tx buffer descriptors (step 8), constructs the MAC headers and performs all the necessary IEEE 802.11 functions (step 9). Then it programs the DMA engine of PAI (step 10). At the final step (step 11), the PAI reads the data and temporarily stores them in internal FIFO for transmission.

When a packet is received at the WLAN side, the following steps are executed: the packet is stored at the internal FIFO of PAI (step 1), the PAI informs the WLAN MAC unit for packet reception, and for any other packet information (header type, CRC correct etc.) (step 2). The WLAN MAC Unit programs the PAI to perform DMA transfers to the appropriate buffers in common memory (step 3), and constructs Rx buffer descriptors which reside also in the common memory. Then it performs the appropriate MAC functions such as error correction, reassembly, packet filtering etc. (step 4). Then it informs the inter-networking unit for the completion of the reception of a packet.

The inter-networking unit, after reading the Rx buffer descriptors and performing the appropriate bridging functions (step 5), constructs the Tx buffer descriptors for ATM transmission. Then it informs the ATM Unit (step 6), and constructs/updates the Transmit Schedule Tables and Segmentation Queue Descriptors (step 7).

The Segmentation Processor reads the Transmit Schedule Table, the Segmentation Queue Descriptors (step 8), and the Tx buffer descriptors (step 9) before constructing the appropriate cell header (step 10). At the final step (step 11), the AAL engine performs the appropriate AAL functions (CRC generation, cell sequence number generation etc.), and transfers the cell from the Tx data buffer (pointed by the buffer descriptor) to the Tx FIFO. The cell is then passed to ATM Physical device through the Utopia interface.

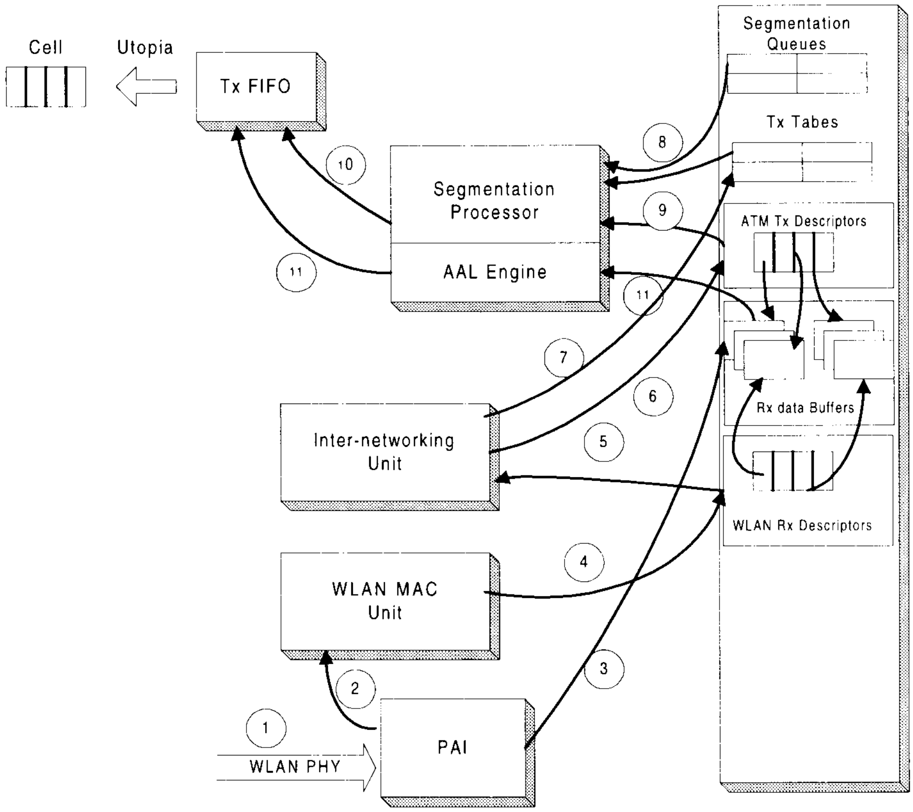

Figure 5: WLAN Receive - ATM Transmit Data flow

5. Bridge Modeling

The “bridge-on-a-chip” architecture described above was modeled and simulated in synthesizable Verilog code in order to prove its functionality. The hardware description language Verilog, offers design flexibility and implementation transportability in different technologies. Synthesizing the above architecture (except of the ARM cores) using Xilinx FPGA technology, we got the following results:

Module	*Estimation of gates*
Segmentation Processor	20.000 gates
Rx FIFO	5.000 gates
Reassembly Processor	20.000 gates
Tx FIFO	5.000 gates
WLAN glue logic and Memory controller	5.000 gates
Decoder/Arbiter/Bridge	2.000 gates
PAI	15.000 gates
Peripherals (interrupt contr. 2, 32-bit timers)	10.000 gates
IWARM glue logic and Memory controller	5.000 gates
Decoder/Arbiter/Bridge	2.000 gates
Peripherals (interrupt contr. 32-bit timer)	5.000 gates
CMIC	5.000 gates
Total	about 100.000 gates

Since the ARM core uses about 20.000 gates, we have a total of 140.000 gates. The external-pin requirements, as illustrated in figure 3, are about 180-pins which can be easily supported along with test and power pins by using a 240-pins QFP package.

6. Conclusions

The 'bridge-on-a-chip' architecture presented in this paper offers compact WLAN to ATM inter-networking, showing the potential for high volume production of such a device, and therefore reducing system cost. Moreover, the use of dual ARM processor configuration with independent internal buses, shared memory and dedicated hardware for ATM SAR processing, leads to a high performance system since the ARM cores are able to execute micro-code at full speed through the dedicated program memory buses. Finally, the data exchanging through common memory, minimizes bus transfers and improves system performance.

References

[1] IEEE Standards Department, "P802.11 Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification", 1997.

[2] The ATM Forum, "LAN Emulation over ATM", Version 1.0, (af-lane-0021.000), January 1995.

[3] The ATM Forum, "UTOPIA, An ATM -PHY Interface Specification", Level 1, v2.01, 21 March 1994.

[4] Advanced RISC Machines Ltd, "ARM Architecture Reference Manual", Document No: ARM DDI 0100B, Prentice Hall, 1996.

[5] Advanced RISC Machines Ltd., "Introduction to AMBA", Document No: ARM DVI 0010A, 1996

[6] ITU-T Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) specification", March 93.

[7] Brian Dorling, Daniel Freedman, Chris Metz and Jaap Burger, "Internetworking over ATM", Prentice Hall, NJ, 1996.