Engineering of ATM TDM Access Applications for Linux MPC8260 Integrated Access Devices

APOSTOLOS MELIONES
Ellemedia Technologies
223 Syngrou Ave., GR-17121, Athens
meliones@ellemedia.com

Abstract: - Network processors provide enormous potential for differentiated high-end integrated networking and communications systems. The PPC PowerQUICC architecture has made this processor family a basic element for networking and communications systems. Its impressive market share has made this processor family the undisputed leader in the communications processor market. On the other hand, Linux is particularly popular in network embedded systems because of its proven networking capabilities and well suiting for networking applications and services that require high reliability and high availability, and the flexibility it offers to specifically tailor it for a special hardware configuration. Furthermore, ATM is very popular in WAN backbone and access networks, especially because of the large investments made by Telcos and ISPs and the inherent support of scalable bandwidths and guaranteed QoS, which facilitates new classes of multimedia services. This paper demystifies for the first time the engineering of Linux serial ATM on the MPC8260 communications processor and derivative CPUs, including the programming of an E1/T1 access application. This development is important for the implementation of ATM over TDM access applications for integrated access devices and multi-service access platforms, including DSL applications and inverse multiplexing over multiple TDM lines. This work is important to the embedded networking community for the development of ATM access devices for high-end communication and networking systems.

Keywords: - Linux ATM, serial ATM, ATM device driver, MPC8260, MPC8264, MPC8266, TDM, T1/E1, IMA, DSL, ATM access device, integrated access device

1 Introduction

Today, the use of Linux in embedded systems is no laughing matter. Linux has become a preferred operating system for embedded systems, mainly because of its open source, maturity and robustness, the multitude of open source and free software projects around it, the community developing it, and the flexibility it offers to specifically tailor it for a special hardware configuration or for support of a certain type of application [1]. Industry is increasingly adopting Linux because of its proven networking capabilities and well suiting for networking applications and services that require high reliability and high availability, and the flexibility it offers to specifically tailor it for customized hardware configurations. ATM dominates WAN backbone and access network technology, especially because of large investments made by Telcos and ISPs and the inherent support of scalable bandwidths and guaranteed QoS in terms of throughput and jitter, which facilitates new classes of multimedia services. However, the real world is dominated by connectionless IP networks. Running IP over ATM is mainly an encapsulation issue, defined in RFC1483/1577. A comprehensive overview of ATM is presented in [2].

According to a report by Gartner Dataquest [3], the PowerQUICC family's market share reached 82.7 percent in 2002, making the processor family the undisputed leader in the communications processor market. To date, 150M units have been shipped. MPC8260 is a PowerQUICC-II architecture single-chip integrated microprocessor and peripheral combination that can be used in a broad range of controller applications, particularly in communications and networking products. MPC8260 is the next generation MPC860, delivering higher levels of system performance, peripheral integration and programmability to the communications market. The integration of a PowerPC core, a RISC-based communications processor module (CPM) and a circuit board's worth of system interface and control functions in a dual-processor architecture provides enormous potential in developing differentiated high-end networking and communications systems, such as SOHO routers, integrated access devices, DSLAMs, central office switches, wireless infrastructure base stations, enterprise and VPN routers, media gateways for packet telephony etc., while significantly reducing time-to-market development stages. The CPM off-loads peripheral tasks from the embedded processing core and supports multiple

high bit rate communications protocols. The degree of system integration could significantly reduce a system's component count, shorten customers time-to-market and slash component cost.

The engineering of Embedded Linux ATM for the MPC8260 family of communications processors on Linux kernel 2.4 is described for the first time in [4]. There we sketch the reference system and driver architecture and discuss items of the SAR mechanism and UTOPIA mode programming, while resolving important issues related to the Linux kernel, such as the packet forwarding performance and support of new devices in earlier kernels. We further describe the driver environment, which includes kernel and user space Linux ATM support. Details of the required ATM device driver interface elements across Linux kernel 2.6 are given in [5]. At the time of writing, Linux ATM for the MPC8260 supports UBR, CBR, VBR-rt and VBR-nrt traffic types, AAL0 and AAL5 cell formats, as well as UTOPIA and serial interfaces. A pre-alpha release of the device driver including the UTOPIA interface and UBR/CBR services was released to the Linux community to appease the ever-increasing demand for the release of a Linux ATM driver for the MPC8260 [6]. The importance of this development to the embedded networking community is evident by several thousand visits to the released driver web page.

In this paper we describe the engineering of the serial ATM interface, which is integrated with the UTOPIA and SAR mechanisms of MPC8260 Linux ATM. This development is important for the implementation of ATM over TDM access applications for integrated access devices and multi-service access platforms, including DSL applications and inverse multiplexing over multiple TDM access interfaces. Serial ATM implementation on Linux MPC826x systems for integrated ATM TDM access applications, including the programming of an E1/T1 access application, is detailed and demystified here for the first time. The reference system architecture is sketched in section 2. Section 3 is devoted to the serial ATM interface and the programming of the serial ATM mode, while section 4 details the programming of an E1/T1 access application. Embedded HTTP and SNMP system management interfaces for configuration and monitoring of the ATM access interface are described in [4].

2 System Architecture

Fig.1 depicts the MPC8260 Linux ATM reference architecture. The reference system is powered by MPC8264, and includes a UTOPIA ATM interface, an ATM T1/E1 WAN interface through a transmission convergence sub-layer and a multi-port fast Ethernet LAN, connected to CPM peripheral ports FCC1, FCC2 and FCC3 respectively. The implementation of the serial ATM interface and the T1/E1 application are incremental tasks that require the existence of the ATM SAR and UTOPIA subsystems developed in [4]. With minor additions, the same architecture can also support xDSL applications and inverse multiplexing for ATM over a multi-port T1/E1 WAN interface. Fig.2 sketches the architecture of the MPC8260 Linux ATM driver. New elements with respect to [4] are the building blocks for the serial ATM interface and the T1/E1 application, which are the topics detailed in the rest of the paper.

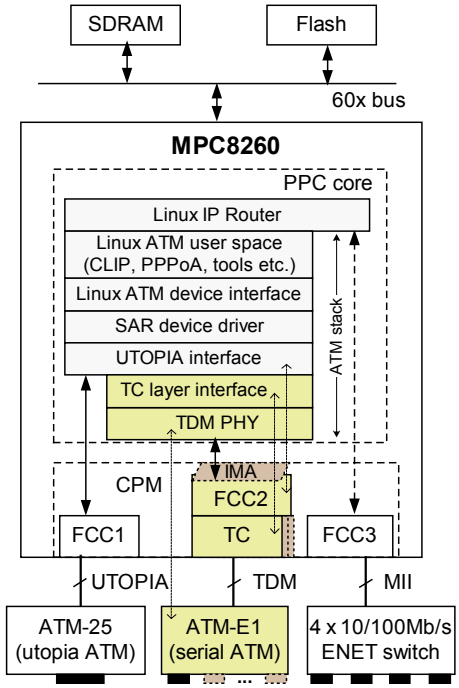

Fig.1: Reference system architecture.

3 Serial ATM Interface

MPC8264, MPC8266 (HiP4 version), and MPC8280 (HiP7 version) are pin compatible enhanced derivatives of the MPC8260 which can support applications which receive ATM traffic over the standard serial protocols like E1, T1, and xDSL via their serial interface ports (SIx TDMx). In all these devices, besides offering significant performance increases and power savings over the established devices, with high speeds in the core, CPM and system bus, the ATM transmission convergence functionality is implemented internally. This allows the use of standard low-cost PHY devices in system applications instead

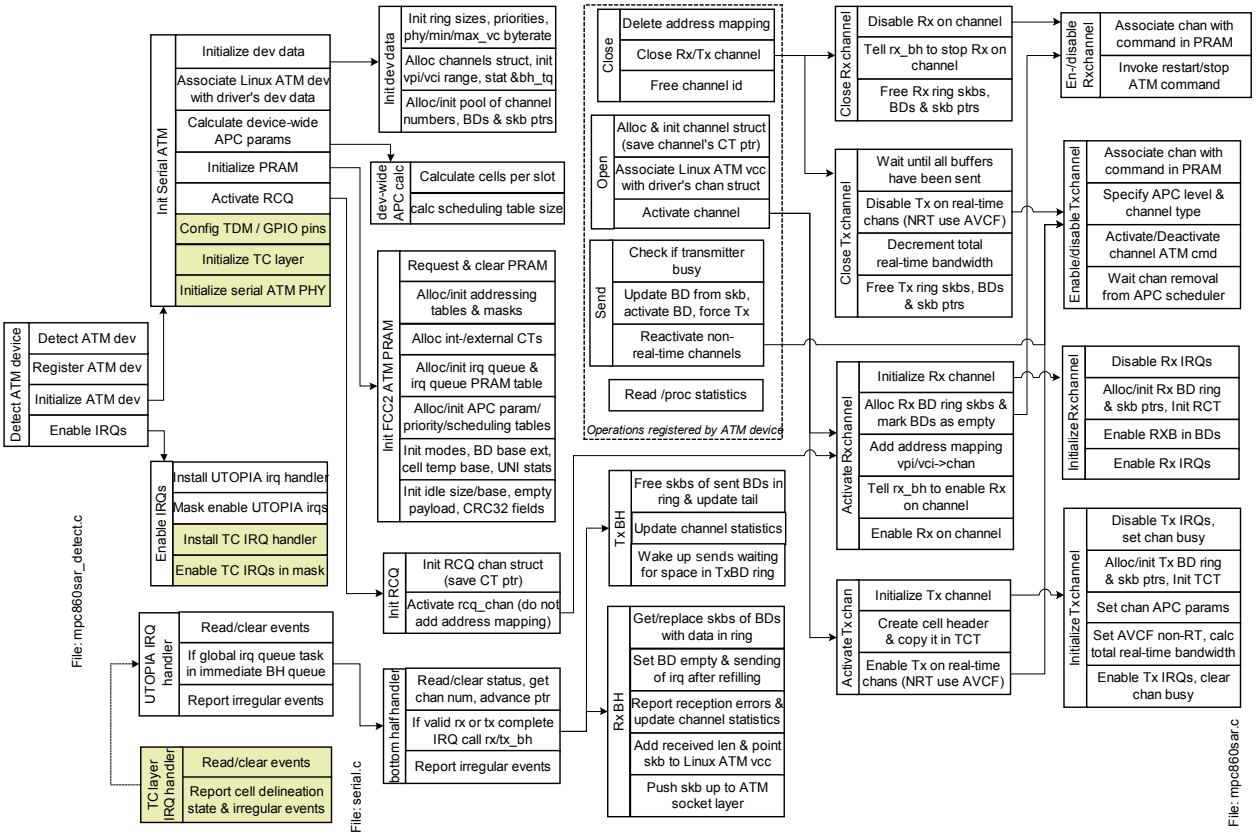

Fig.2: Linux MPC8260 serial ATM device driver architecture.

of PHYs that support UTOPIA bus devices. In the rest of this paper MPC8260 will refer to the TC layer enhanced derivatives. A typical TC layer application requires the use of one SI TDM channel per TC block. All TC blocks are internally connected to FCC2 via a UTOPIA 8-bit MPHY bus.

3.1 TC Layer Memory Map

The TC layer memory map is located in the reserved space that follows FCC3 memory block within the internal memory space of MPC8260. Corresponding memory map definitions are not included in the current Linux kernel distributions. The TC layer memory block is mapped on 260 bytes resolution at offset 0x11400 to the internal memory map base address. A TC block memory is represented with a struct of unsigned shorts representing the TC block registers and counters along with a reserved space. An overall struct groups the array of the individual TC blocks along with the general status and event registers. A global pointer variable to the TC layer memory block is assigned the address of the TC layer internal memory map.

3.2 Pin Configuration

Pin configuration involves the selection of dedicated TDM signals and general-purpose I/O pins, as illustrated in Fig.3, among the peripheral functions multiplexed onto the parallel I/O ports, through proper programming of the port registers. This assumes that the correct pins corresponding to receive and transmit data, clock and frame synchronization signals, required for operating the TC layer, and pins that compose the microprocessor bus interface are connected to the external line interface device. General-purpose pins include an 8-bit multiplexed address/data bus, device reset, chip select, address latch, and read/write operation signals. FCC2 UTOPIA signals are internally connected to the TC layer hardware. Therefore, all port pins pertinent to FCC2 UTOPIA functions are available for other purposes. The single TC block in our reference system is operated via the SI1 TDM A1 port. Any other TDM port can be alternatively used depending on the availability of pins. Multiple PHYs are handled along the same lines. In our reference system the microprocessor bus interface is shared via chip-select between the PHY devices for UTOPIA and serial ATM. The required pin configuration is summarized in Table 1. Pin configuration corresponds to the *Config TDM/ GPIO pins* architectural block in Fig.2.

3.3 Interrupt Handling

TC blocks are internally connected to FCC2 via a

UTOPIA MPHY bus. Therefore, TC layer activity involves internal FCC2 UTOPIA activity and related interrupts. Consequently, independent TC layer and UTOPIA interrupt handlers need to be installed in the Linux kernel in order to handle serial ATM interrupts, corresponding to the hardware interrupt vectors for the TC layer (0x2C) and FCC2 (0x21).
The TC layer interrupt handler reads/clears the general event register to identify the events from the TC blocks. It then reads/clears the corresponding TC block event register and prints warning messages for recorded events. For cell delineation in particular it notifies the status of the state machine. The FCC2 UTOPIA interrupt handler is automatically called following the insertion of an entry to the interrupt queue for the FCC2 UTOPIA port by the CPM. This is similar to the FCC1 UTOPIA interrupt handler. The serial ATM interrupts are masked in the TC layer and FCC2 mask registers throughout the registration and initialisation of the serial ATM device.

3.4 Serial Mode Programming

This section describes the CPM configuration regarding the TC layer necessary to implement a serial ATM application using a single TC block. Serial mode programming corresponds to the *Initialize TC Layer* architectural block in Fig.2. Serial mode programming involves the configuration of the TC layer, the Serial Interface with the Time Slot Assigner, the TDM channel, the FCC2 operation mode, and the internal TC layer connection to FCC2. Using two or more TC blocks requires FCC2 to work in MPHY mode.
FCC2 is operated in ATM mode as a UTOPIA master in a single PHY environment. Idle cells are not discarded and receive parity check is disabled. The FCC2 receiver and transmitter remain disabled during the initialization. Setting the receiver in slave mode and enabling local loopback mode selects UTOPIA loopback for testing purposes.
FCC2 is operated in external rate mode. In this mode the external device determines the data rate and the CPM is responsible for keeping the FCC2 FIFO full by inserting ATM cells, or idle cells if ATM cells are not available. In general, the TC would never have its transmit FIFO empty. However, if the CPM is busy, and the TC is forced to generate an idle cell, an underrun condition occurs and an interrupt is sent to the host, if not masked. This mode loads the CPM more than the internal rate mode when the total transmission rate is less than the PHY rate, however the overhead to the CPM is negligible at the E1 rate.
Since the FCC2 UTOPIA bus is connected internally to the TC UTOPIA bus the parallel port pins are configured only for the active TDM and the microprocessor bus interface. The TC layer interrupt priority is increased over the SCCs. Passing of FCC2 and TC layer interrupt requests to the core is enabled in the SIU interrupt mask register. A command issued to the CPM initializes the transmit and receive parameters in the FCC2 PRAM involving the FCC2 (not the ATM) sub-block code. The FCC2 receiver and transmitter are enabled as the last step in the FCC2 initialization process. Using the CPM multiplexing logic FCC2 is connected to the TC layer, and CLK19 is connected to the TDM A1 receive clock for common Rx/Tx clock operation. In MPHY mode the user has further to program how many address lines are needed for FCC2. By default the master of FCC2 uses the multiplexed MPHY address pins between FCC1 and FCC2.
BRG7 is defined as the FCC2 UTOPIA clock. The input clock comes from BRGCLK (66MHz for a 133MHz CPM clock configuration) divided by 9. Dividing by an odd number ensures a 50% duty cycle. The operation of the TC layer requires a minimum frequency ratio of 1:2.5 between the serial clock and the UTOPIA clock. This allows supporting xDSL applications at bit rates up to 10 Mbps.
The TC1 layer block is configured using the TC layer mode register. The TC layer is enabled following the programming of the FCC2 and CPM multiplexing logic. The user selects which TC layer functions are enabled, such as scrambling/descrambling on the cell payload, coset and error check on the HEC, correction of single bit header errors, filtering of idle/unassigned cells, generation of underrun interrupts, byte align and internal loopback mode of operation. In our configuration TC1 is the only PHY on UTOPIA, error correction as well as inverse multiplexing is disabled, and reading a cell counter does not clear the counter. The default values are programmed for the alpha and delta parameters of the cell delineation state machine. These parameters are provided to help tune the system according to the line error characteristics of a specific application. Alpha determines the robustness against false misalignment due to bit errors. Delta determines the robustness against false delineation in the synchronization.
The SI is programmed to retrieve the data bits out of the E1 frame. The E1 transmission link consists of 32 x 64Kbps channels. Channels 0 and 16 are reserved for transmission management, while all other channels are used for payload. The SI RAM setting refers to the direct mapping of ATM cells onto E1 transmission frames according to ITU-T recommendation G.804 [11]. This specifies that ATM cells have to be carried in bits 9-128 and 137-256. PLCP

Table 1: MPC8260 serial ATM mode programming.

Serial Mode Programming Register	Value (hex)	Description
GFMR2	A	ATM mode, disable FCC2 Rx/Tx
FPSMR2	800400	Single PHY master mode, disable Rx parity check, do not discard idle cells
FTIRR2_PHY0	0	External rate mode
SCPRRL	80A07770	Increase TC layer interrupt priority over the SCCs
SIMRL	40080000	Allow interrupts from TC layer and FCC2
CPCR	16210280	Init Rx/Tx parameters in FCC2 PRAM
GFMR2	|=30	Enable FCC2 Rx/Tx
CMXFCR	|=920000	Connect FCC2 to the TC layer, FCC2 Rx/Tx clock is BRG7
CMXUAR	0	Default assignment of multiplexed MPHY address pins to FCC2 master
CMXSI1CR	|=80	TDMa Rx/Tx clock is CLK19
BRGC7	10010	FCC2 UTOPIA clock is 66MHz divided by 9
TCMODE1	F203	Enable TC layer Rx/Tx, no payload scrambling, coset enable, no header error correction, no cell filtering, TC1 is the only PHY on UTOPIA, do not clear counter
CDSMR	3980 (1040)	alpha=7, delta=6 (alpha=2, delta=1)
SI1RAM[0-2]	0002015E015A	SI Rx/Tx RAM settings to retrieve data out of the E1 frame
SI1RAM[3-5]	0002015E015B	
SI1AMR	68	Common Rx/Tx pins, frame sync active low and sampled with the rising edge of the channel clock, no bit delay between frame start bit and sync
SI1GMR	1	Enable TDMa
PODRA	&=CC3FFFFF	Port A register masks for dedicated pin assignments
PSORA	|=3C0[5], &=CF[7]	(CLK20, CLK19, L1RSYNC, L1TSYNC, L1RXD, L1TXD)
PDIRA	|=40[5], &=CC7F[5]	
PPARA	|=33C00000	
PODRD	|=3FC0, &=F13F[5]	Port D register masks for GP I/O pin assignments
PDIRD	|=EC03FC0	(~RES, ~CS, ~WR, ALE, ~RD, AD[0-7])
PPARD	&=F13FC03F	
PDATD	|=E40[5], &=FF7FC03F	

cell mapping [12] can be supported as well. The programming of each entry in the Rx and Tx SI RAM determines the routing of the serial bits. Entries 0 and 3 ignore the bytes from channels 0 and 16, while entries 1-2 and 4-5 route bits 9-128 and 137-256, organised in byte groups, to FCC2. After the last entry the SI waits for the sync signal to start the next frame. Note that there must be only an even number of entries in a SI RAM frame. Therefore, an entry may need to be split into two entries.

The last step is to initialise the SI mode registers and enable TDMa on SI1. The same clock and frame sync signals are used for both receive and transmit sections of TDMa1. In order to interface correctly the E1 PHY timing (see §4) the frame sync pulse is active low and sampled with the rising edge of the channel clock, while frames are transmitted/received on the same clock as the sync. TDM can be operated in loopback testing with or without affecting the external serial lines. Enabling TDM is the last step in initialisation.

We also provided hooks for serial ATM device testing in external rate mode without the need of an access network and an operational PHY generating the data rate clock and frame sync pulse. For this it is necessary to hardwire $PC21_{BRG6}$ to $PC31_{CLK1}$ and to $PC29_{CLK3}$, and $PD8_{BRG5}$ to $PA6_{L1RSYNC}$, or any other viable combination, and to configure the port pins accordingly. We defined BRG6 as the external data rate 2MHz clock, which drives MPC8260 input clocks CLK1 and CLK3, CLK1 as the common Rx/Tx clock of SI1 TDMa, and BRG5 as the 8-KHz frame sync generated from CLK3. Then we were able to test the system by operating TDM in loopback mode. Note that HiP4 Rev. A.0 devices cannot be used in a real environment because they suffer from the CPM77 device error: The internal TC layer hardware is connected to the UTOPIA bus in the reverse order. This means the transmitter transmits the data LSB first and not MSB as required. The HEC is generated according to the reversed data. On the receive side data will be received in the right order, the HEC will be checked correctly but the data will be delivered to the UTOPIA Rx data bus in the reverse order. Such devices can be used only in loopback mode for development purposes.

4 E1/T1 Application Programming

In our reference system the TC layer is operated at E1 and T1 rates. The proposed reference design can also support xDSL applications at bit rates up to 10 Mbps, as well as IMA applications over multiple access interfaces. This section describes the integration of the external line interface component to the serial ATM interface of the MPC8260. We used the FALC56 line interface component to fulfil all required interfacing between an analog E1/T1 line and

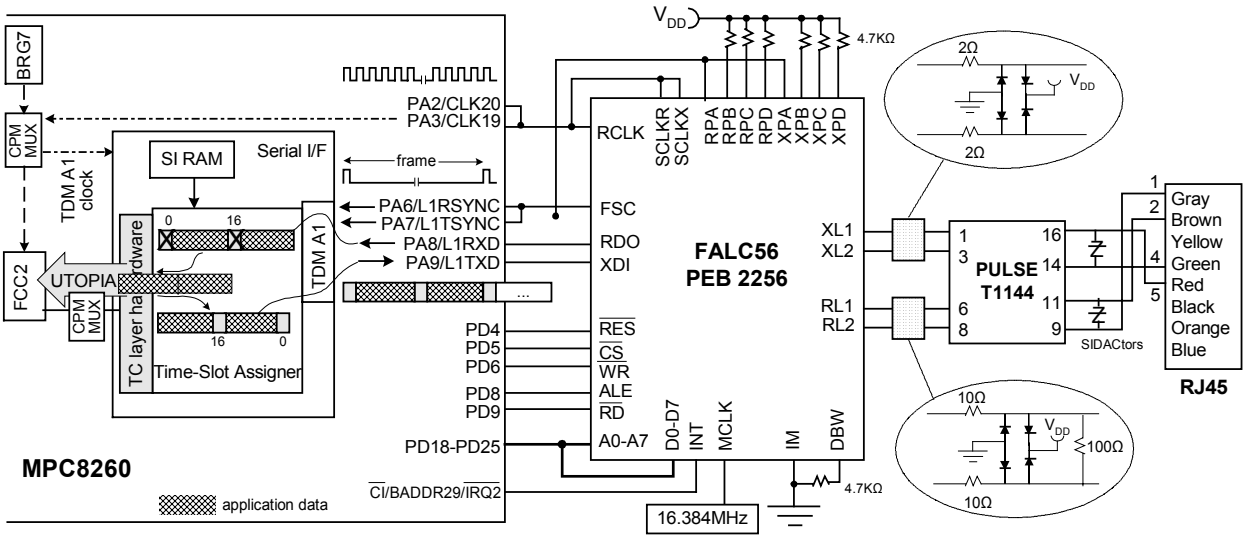

Fig.4: Block and schematic diagram of serial ATM application operated at the E1/T1 interface.

Table 2: Basic PHY device initialisation in E1 mode.

Basic hardware settings		
Pin	**Signal**	**Description**
DBW	Low	8-bit bus interface mode
IM	Low	Intel interface mode
RCLK		Connected to SCLKR and to SCLKX
FSC		Connected to RPA and to XPA
Basic device initialisation in E1 mode		
Register	**Value**	**Description**
GCM1-6	0x58D2C20710	16.384 MHz master clocking mode according to external MCLK
FMR1	Default	E1 mode select. Can be omitted
LIM0	(|= MAS)	Master mode. Analog input is the default. Local loop and receive equalizer long-haul modes can be selected. Resetting MAS selects slave mode
FMR0	|= XC|RC	HDB3 line interface code. Alternatively, AMI coding can be selected
FMR1	|= XFS	Transmit framing is CRC4 multiframe. Default is doubleframe
FMR2	|= RFS1	Receive framing is CRC4 multiframe. Default is doubleframe
FMR2	(|= RTM|DAIS)	Should be set to realize unframed data reception. Used only in unframed data mode
XSP	(|= TT0)	Set in unframed data mode. Timeslot 0 data on XDI is routed unchanged through PHY
PC5	|= CRP	Set RCLK output. SCLKR input (default). SCLKX is an input pin
CMR1	|= RS1|DCS	In master mode RCLK source is DCO-R. In slave mode RCLK is recovered from the line and DCS should be set to synch DCO-R on line RCLK. Frequency is 2.048 MHz
CMR2	Default	DCO-R frequency centered (SYNC pin NC). Internal system working clock and frame sync sourced by RCLK and FSC respectively. Can be omitted
SIC1, FMR1	Default	System clocking and data rate. Can be omitted
GPC1	|= CSFP1|CSFP0	FSC output, active low
SIC3	|= RESR	Data on XDI/RDO latched/clocked with the falling/rising edge of ~SCLKX/R
XC1, RC1	0x4	Adjusting the beginning of a frame on the system highway relative to ~SYPX/R (T=0)
XPM2	(= DAXLT)	Disable automatic tristating of XL1/2 upon detection of short on XL1/2 pins. Set when testing the system in external line loopback mode
CMDR	|= XRES|RRES	Software reset (check SIS.CEC to avoid loss of command)

the digital PCM system bus. The controller is operated in E1 mode. Fig.3 illustrates the block and schematic diagram of our reference serial ATM application operated at the E1 interface. Due to the particular resistor selection, the same schematic is valid for the T1 application as well.

Reset is applied for a minimum requested period of 10 µs. Bit-swapped read and write accesses to the PHY registers conform to the Intel non-multiplexed address and read/write cycle timing specified in the microprocessor interface electrical characteristics. A few hardware settings are necessary for the communication between MPC8260 and the PHY to enable byte access to all control and status registers and select the Intel interface mode. Furthermore, it is necessary to program at least the following basic E1 initialisations specific to the system and hardware environment after reset goes inactive. The PHY clocking unit is tuned to the selected reference frequency provided on pin MCLK. The appropriate register settings are calculated using a set of calculation formulas. E1 operation mode and analog line

interface is the default after reset. The PHY operates in master or slave mode and a receive line coding is selected for the data received from the ternary interface. RCLK and SCLKR/X pins are set to output and input respectively. RCLK, which defines the data rate clocking and the common clock for the TDM receiver and transmitter in our reference design, is set to a constant 2.048MHz clock. In master mode RCLK is generated by the internal DCO-R circuitry based on MCLK. In slave mode RCLK is recovered from the data stream received at the analog input line. The DCO-R circuitry is synchronized on the line recovered clock, otherwise the frame sync pulse is not generated due to inactive DCO-R. The working clocks for the receive and transmit system interfaces are sourced by SCLKR and SCLKX respectively. Combined with the respective hardware setting this defines RCLK as the internal working clock for the PHY. The working frame sync pulses for the receive and transmit system interfaces are sourced by RPA and XPA respectively. This defines FSC as the internal working frame sync pulse for the PHY. FSC is set to output and active low in order to produce an 8-KHz frame synchronization pulse that drives correctly the internal ~SYPR/X signals. Data on the receive/transmit system interface is latched/clocked with the opposite active edge of the channel clock used for MPC8260 TDM.

Adjusting the beginning of a frame on the system highway relative to the frame sync signals through programming of the transmit and receive offset counters will stop the serial ATM controller from reporting receive CRC and length data errors, because the SI RAM mechanism routing framed data from the system interface to FCC2 will ignore correctly all and only non-data bits carried in time slots 0 and 16. When testing the system in external line loopback mode the TC layer cell delineation state machine of MPC8260 will not lock the synch state, unless automatic setting of the analog output into a high-impedance state upon detection of short on these pins is disabled. Software reset should follow the device initialisation. Table 2 gives the details of the most important parameters in terms of signals and control bits, which should be programmed for E1 operation compliant with our reference serial ATM architecture. A delay of 1 ms is applied immediately after setting MCLK for a correct start up of the primary access interface and power up of the system with cable on. The setup described in Tables 1 and 2 has been validated through interoperability tests performed with a Cisco 7200 router and a Lucent PacketStar PSAX 1250 multiservice media gateway both featuring multiport T1/E1 IMA adapters to provide LAN-to-WAN connectivity. Interoperability with PSAX required the only addition of setting the Tx byte aligned (TBA) mode in the TC layer to the above configuration. Fig.4 illustrates the realisation of the timing synchronization between the E1 transceiver and the serial ATM interface.

4.1 Master Mode Operation

Master mode operation invokes a periodic momentary loss of TC synchronisation due to the difference in clock frequency stability, usually termed phase jitter, between the user and network equipment master system clocks (output on pin RCLK at user side). Phase jitter piles up on sampling the received data stream and after a while results inevitably in incorrect sampling of the beginning of an ATM cell and loss of TC synchronisation, as depicted in Fig.5. The smaller the PPM value of the reference clock the larger the time interval between successive losses of TC synchronisation. In our reference system using a ±50-PPM MCLK reference clock master mode operation results in a dropped buffer rate of approximately 2.1‰ in TCP tests carried out with the TTCP testing utility, assuming buffer length is 8192 bytes. The time elapsed between a loss of TC synchronization and a subsequent lock of the synch state affects the user-space ATM signalling daemon and the ability to run Classical IP over ATM on the system. The signalling demon “No Response” timer will expire if locking the TC synch state takes longer than the timeout value. Consequently, signalling information cannot be exchanged blocking the flow of IP traffic between the user and network equipment. We solved this problem by either increasing the atmsigd timeout interval or shortening the transition time between the states of the TC cell delineation state machine by setting appropriate Alpha and Delta parameters. In our reference system the problem exists with default Alpha and Delta and is solved when setting Alpha=3 and Delta=2.

5 Conclusions

In this paper we presented the engineering of the serial ATM interface, which is integrated with our own Linux ATM for the MPC8260 communications processor and derivative CPUs. This development is important for the implementation of ATM over TDM WAN interfaces and access applications, including inverse multiplexing over multiple TDM access interfaces and DSL applications. The proposed system fits to a very popular powerful communications processor, which retains an impressive market share and is massively used in a broad range of high-end communication and networking systems. Serial ATM implementation on Linux MPC826x

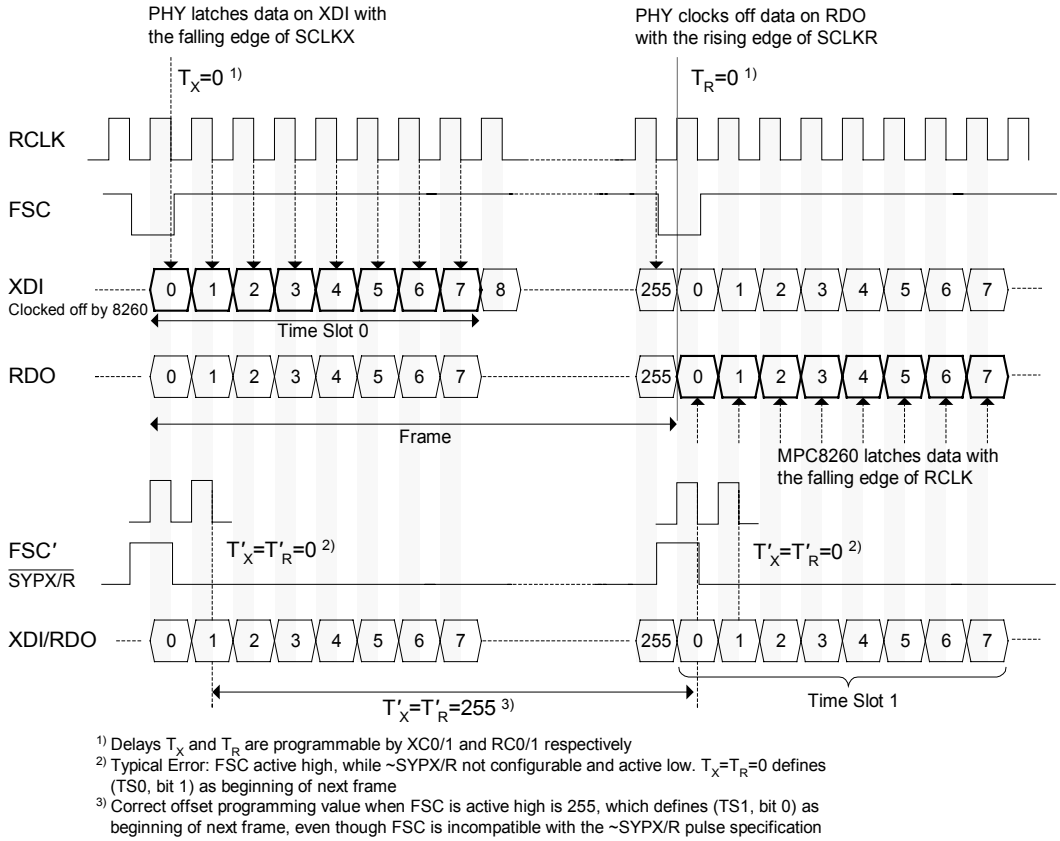

Fig.4: Realization of timing synchronization between E1 transceiver and MPC8260 serial ATM.

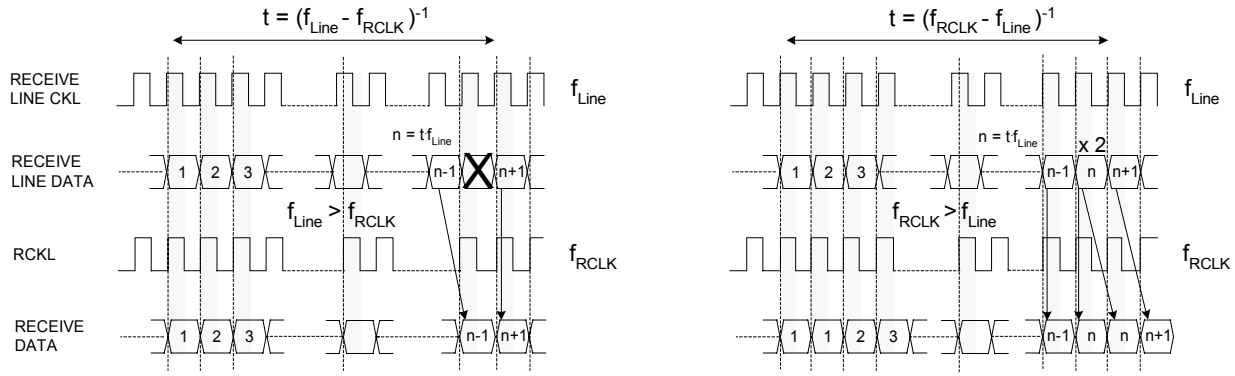

Fig.5: Loss of synchronization at ATM cell boundaries in master mode operation.

systems for integrated ATM TDM access applications, including the programming of an E1/T1 access application, is detailed and demystified here for the first time.

References

[1] K. Yaghmour, *Building Embedded Linux Systems*, O'Reilly, 2003, ISBN 0-596-00222-X.

[2] A. Alles, Internetworking with ATM, White Paper, *Cisco Systems*, May 1995.

[3] Communications Semiconductor and Optical Component Market Share, 2002 © 2003 Gartner Inc.

[4] A. Meliones, Engineering of Embedded Linux ATM for MPC8260 and Derivatives, 9th WSEAS CSCC Int. Conf.

[5] A. Meliones, S. Spanos, and N. Zervos, Linux ATM Interface for PowerPC-based Network Embedded Systems, WSEAS Transactions on Communications, Vol. 3, No. 2, April 2004, ISSN 1109-2742.

[6] SourceForge Linux ATM Driver for MPC8260, http://atm8260.sourceforge.net

[7] ITU-T G.804: Digital networks-ATM cell mapping into Plesiochronous Digital Hierarchy, 1993.

[8] ETSI recommendation ETS 300 213: Network Aspects; Metropolitan Area Network; Physical layer convergence procedure for 2,048 Mbps; 1992-12.