Title:

Design and Prototyping of an SDH-E1 Mapper Soft-core

Authors:

CÉSAR A. M. MARCON	marcon@inf.ufrgs.br
JOSÉ C. S. PALMA	jcspalma@inf.ufrgs.br
NEY L. V. CALAZANS	calazans@inf.pucrs.br
FERNANDO G. MORAES	moraes@inf.pucrs.br

Authors Affiliation:
Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS/FACIN
Av. Ipiranga, 6681 - Prédio 30 / Bloco 4 - 90619-900 - Porto Alegre - RS - BRAZIL

Abstract. This paper describes the design and prototyping of EMS, a telecommunication intellectual property soft-core developed in the scope of industry-academia cooperation. EMS performs insertion (mapping) and extraction (demapping) of E1 channels into/from Synchronous Digital Hierarchy (SDH) frames. The basic SDH frame is transmitted in 155.52 Mbps rate, allowing to pack up to sixty-three 2.048 Mbps E1 channels. E1 channels belong to the Plesiochronous Digital Hierarchy (PDH). The paper addresses the solution of several synchronization problems implied by the E1 channels mapping/demapping process. EMS was fully described in RTL VHDL. It was functionally validated by simulation and prototyped in FPGA platforms. Together with the exploration of the techniques involved in embedding PDH into SDH frames, another contribution of the work is the availability of a reusable and parameterizable telecom core with high performance, low latency, and small size.

Keywords: SDH, E1, SDH-E1 mapping/demapping, soft IP core.

Design and Prototyping of an SDH-E1 Mapper Soft-core

Abstract. This paper describes the design and prototyping of EMS, a telecommunication intellectual property soft-core developed in the scope of industry-academia cooperation. EMS performs insertion (mapping) and extraction (demapping) of E1 channels into/from Synchronous Digital Hierarchy (SDH) frames. The basic SDH frame is transmitted in 155.52 Mbps rate, allowing to pack up to sixty-three 2.048 Mbps E1 channels. E1 channels belong to the Plesiochronous Digital Hierarchy (PDH). The paper addresses the solution of several synchronization problems implied by the E1 channels mapping/demapping process. EMS was fully described in RTL VHDL. It was functionally validated by simulation and prototyped in FPGA platforms. Together with the exploration of the techniques involved in embedding PDH into SDH frames, another contribution of the work is the availability of a reusable and parameterizable telecom core with high performance, low latency, and small size.

Keywords: SDH, E1, SDH-E1 mapping/demapping, soft IP core.

1. Introduction

Over the last three decades, communication networks are evolving from analog to digital [1], which results in better transmission quality and larger bandwidth. In many areas, this technological change significantly boosted telecommunication systems research. With the advent of the World Wide Web comes an unprecedented increase in data traffic. The telecommunication systems complexity is growing faster, due to: (i) media and protocol diversity; (ii) applications nature variety; (iii) increased data communication speed. To cope with these fast changing scenarios, new telecommunication technologies are necessary. The Synchronous Digital Hierarchy is one example technology, due to its high data communication speed and high capacity to pack telecom protocols with different natures, like E1, ATM and others.

This work introduces EMS, an **E**1 channel **M**apper/Demapper into/from **S**DH frame. EMS is an intellectual property (IP) soft-core developed in the scope of industry-academia research and development cooperation. EMS is a successor of an E1 IP soft-core design [2] developed within the scope of the same cooperation. An IP core is a pre-designed and pre-verified hardware module used in combination to compose larger circuits, typically custom VLSI integrated circuits or large programmable devices, such as multimillion-gate FPGAs. According to availability, IP cores can be classified into soft, firm or hard cores [3]. Soft-cores are open-source codes, enabling high flexibility. Soft core design may not guarantee some functionality, such as exact timing in the final implementation. IP cores are increasingly important resources for complex digital systems design in general and particularly for telecom systems. According to ITRS report [4], by 2012 90% of complex silicon chips surface will be composed by cores. In addition, approximately 50% of the programmable devices sold today are used in telecom applications [5].

The goal of this paper is to present the design of the EMS core. The main contributions are: (*i*) the design of an IP soft-core compliant with specific ITU-T standards [6][7][8]; (*ii*) the prototyping of the system in hardware to guarantee timing constraints using Xilinx Virtex FPGAs; (*iii*) present results of accurate buffering analysis to perform E1 mapping and demapping into/from SDH frames; (*iv*) make available a design with small footprint, which enables straightforward scaling and IP reuse in larger circuits.

The rest of this paper is organized as follows. Section 2 summarizes some important aspects of the plesiochronous digital hierarchy, which defines the characteristic of E1 carriers. In Section 3, some basic concepts of SDH are explained. Section 4 presents reviews previous works related to SDH mappers. Section 5 introduces the EMS architecture. Section 6 describes the EMS validation and prototyping, while Section 7 presents some conclusions.

2. Plesiochronous Digital Hierarchy

The term plesiochronous describes communication system where transmitted signals have the same nominal digital rate but are synchronized with different clocks. The Plesiochronous Digital Hierarchy (PDH) is a hierarchy of data and voice transmission systems that communicate using plesiochronous synchronization. PDH is a conventional multiplexing technology for network transmission systems.

European/South-American and North-American/Japanese versions of the PDH system differ slightly, but the operating principles are the same. North-American/Japanese basic data transfer rate is a stream transmitted at 1.544 Mbps, called a T1 channel. The European/South-American basic data transfer rate is a data and voice stream of 2.048 Mbps, namely E1 channel. For voice transmission, an E1 channel is broken down into 30 data and voice channels of 64 Kbps plus 2 64 Kbps channels used for synchronization and signaling. Each channel in a frame contains 8 bits and is called a time slot. Therefore, a frame contains 256 bits. Each time slot corresponds to a 64 Kbps channel carrying 8 bits of either data or an 8 KHz digitalized voice sample. Time slots are combined using Timing Division Multiplexing (TDM) at 2.048 MHz. Consequently, a frame is transmitted each 125 μs. Multiple frames are grouped to transport alignment, error detection and service information. Eight consecutive frames constitute an E1 submultiframe (SMF) structure. Two consecutive E1 SMFs form an E1 multiframe (MF). E1 carrier equipment transmits and/or receives an MF each 2 ms. The E1 rate is allowed to vary ± 50 ppm, which is equivalent to ± 102.4 bps. This means that different E1 data streams can be running at slightly different rates.

PDH can combine multiple E1 channels generating other data and voice streams, such as 8 Mbps (E2), 34 Mbps (E3), 140 Mbps (E4) and 565 Mbps (E5) which are used in several kinds of communication systems. As an instance, 565 Mbps is typically used to transmit data over an optic fiber system for long distance communication.

3. Synchronous Digital Hierarchy

The Synchronous Digital Hierarchy (SDH) and the Synchronous Optical NETwork (SONET) are hierarchies used in Europe/South-America and North-America/Japan, respectively. Both systems employ synchronous time division multiplexing techniques to transmit different tributaries (E1, Ethernet, ATM, etc) through the same physical channel. A primary goal in the development of the SDH/SONET formats is to define a synchronous optical hierarchy with sufficient flexibility to carry payloads of different types. SONET and SDH are based on transmission at rates that are integer multiples of 51.840 Mbps. SONET basic frame structure is called synchronous transport signal level one (STS-1). SDH basic modular signal is called synchronous transport module level one (STM-1). The STM-1 rate is an extension of the basic STS-1 (for this reason also called STM-0) and operates at 155.52 Mbps, carrying three interleaved STS-1 frames.

Synchronous hierarchies differ from PDH in the exactness of data transport rate. SDH systems are tightly synchronized to network base clocks, making the entire network operate synchronously. The structure of SDH synchronization network (SSN) is founded on master-slave mode hierarchy of clock. The SSN highest hierarchy level is performed with a high precision clock defined as primary reference clock (PRC) [9] and is generally implemented by atomic frequency oscillators. The remaining of the hierarchy is organized as a tree. The reference timing-signal generated by PRC is distributed to the clocks of lower hierarchy levels, which are named slave clocks (SCs). SC tracks the reference-timing signal by means of phase-locked loop (PLL) systems.

SDH is a multiplexed structure. Different containers (C-11, C-12, C-2, C-3 and C-4) with different rates are mapped to virtual containers (VC-11, VC-12, VC-2, VC-3 and VC-4). Pointers implement virtual container alignment, generating tributary units (TU-11, TU-12, TU-2 and TU-3) or administrative units (AU-3 and AU-4). Tributary units are multiplexed in tributary unit groups (TUG-2 and TUG-3) according to container rate. TUG-2 can be multiplexed in VC-3 or TUG-3, and TUG-3 is multiplexed in VC-4. Administrative units are grouped in administrative unit group (AUG). Finally, AUG is multiplexed in one or more STM-1s.

This work focuses only in the path highlighted in Figure 1. The E1 channel is packed into C-12, implying the insertion of a control and stuffing byte. C-12 is mapped into VC-12. A pointer implements virtual container alignment, generating TU-12. Three TU-12 are multiplexed into TUG-2, seven TUG-2 are multiplexed into TUG-3, and three TUG-3 are multiplexed into VC-4. Pointers implement virtual container alignment, generating the AU-4. The AU-4 is grouped into the AUG, and the AUG is multiplexed into one or more STM-1s.

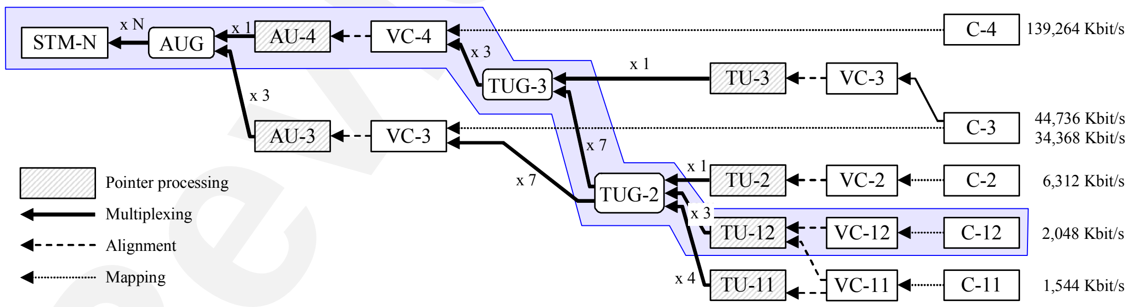

Figure 1 – SDH Multiplexing structure [6]. The marked path underlines the focus of this work.

Figure 2 exemplifies one possible composition of the STM-1 frame structure. This composition reflects how EMS operates. Each STM-1 frame has 9 rows and 270 columns. Each column has a width of 1 byte. The first nine columns are transport overhead, combining a pointer (AU-4 PTR) and section overhead (SOH). SOH contains framing error monitoring and management. AU-4 PTR identifies the VC-4 start point.

STM-1 payload, or AU-4, uses the remaining 261 columns. The first three VC-4 columns are VC-4 path overhead (POH) and two stuffing columns. Three interleaved TUG-3 are mapped in the remaining 258 columns of VC-4 (VC-4 payload). The VC-4 payload is composed by 6 stuffing columns and 63 interleaved TU-12s. Each TU-12 is distributed along four columns, summing up a total of 36 bytes (9 bytes per column). The VC-12 virtual container preceded by a POH forms a TU-12. As a consequence, each VC-12 is composed by an E1 carrier plus two stuffing/control bytes.

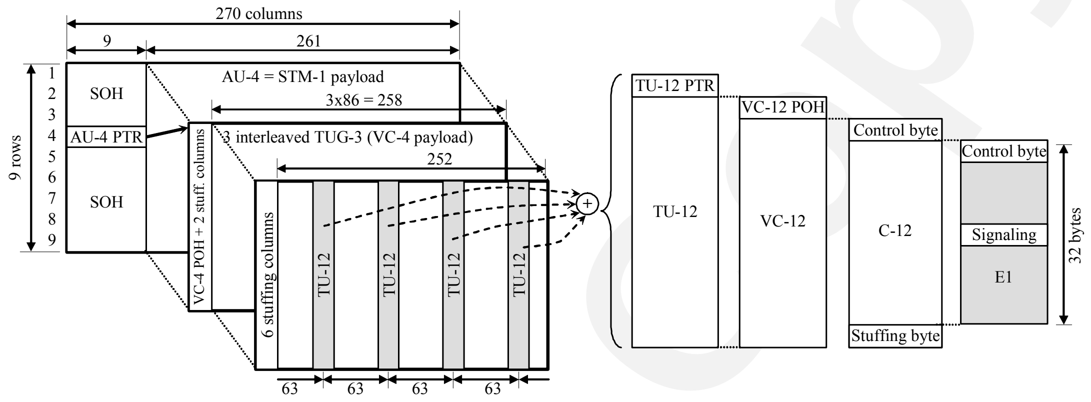

Figure 2 – SDH STM-1 frame structure.

Each frame transports 19,440 bits (270 x 9 bytes) at a 155.52 Mbps rate, implying a frame period of 125 μs. Four frames compose a super-frame (SF), the highest structural level of SDH STM-1. As depicted in Figure 3, the STM-1 payload pointed by the J0 byte may float inside the STM-1 frame. J1 marks the start point of each VC-4 payload inside the STM-1 payload.

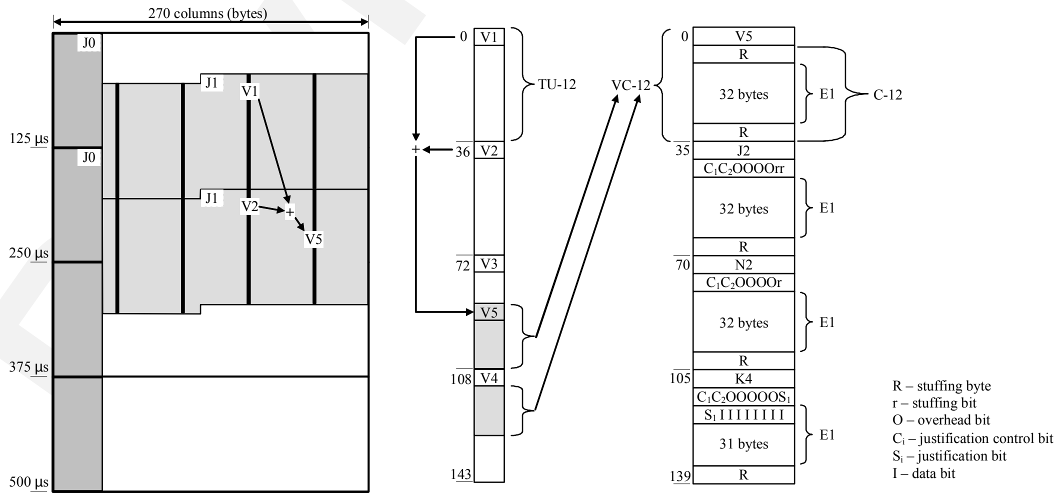

Figure 3 – SDH-STM1 super-frame structure with VC-12 mapping.

In each SF, there are four different kinds of TU-12 PTR (V1, V2, V3, and V4).

V1 and V2 pointers are joined to compose the V5 address, which marks the beginning of C-12. When PDH and SDH clock frequencies are synchronized, there are 1024 bits of useful data transmitted in 4 E1 channels. Otherwise, it is possible to add or subtract one bit from each SF. The C-12 structure of SF merges two or three extra bytes with 32 or 31 E1 bytes. The first C-12 encloses two stuffing bytes and an E1 channel. The second and the third C-12 of each SF contain one control byte, one stuffing byte and one E1 channel. The last C-12 of each SF encloses one control byte, seven data bits and a justification opportunity bit, one stuffing byte and 31 bytes of E1 channel. The majority of C_1s and C_2s bits implies positive or negative frequency justifying. This means that when two or more C_i bits are 1, the S_i bits are valid data. This, in turn, means that 1025 bits of data are available for SF. On the other hand, if two or more C_i bits are 0, the S_i bits are stuffing bits, meaning that 1023 bits of data are available for SF. The number of bytes implies slight changes of clock frequency. For instance, 1025 bits implies 2,050 KHz and 1023 bits implies 2,046 KHz. This frequency change allows mapping/demapping a PDH frame into/from an SDH frame without data loss.

4. Related Work

The demand for telecommunication services leads to a significant offer of SDH systems in the market [10][11][12]. Each distinct equipment presents its own distinguishing features, different prices, and use cases. In addition, the high complexity of SDH systems like asynchronous to synchronous mapping, design for testability, abstract modeling and others, drives the research in this field. Lin et al. (1994) propose a flexible architecture for implementing an SDH STM-1 Add-Drop multiplexer [13]. They implemented an internal Telecom Bus-like architecture, to provide E1 and E3 communication services between internal and external circuits (adapters/converters).

Yongming et al. (1996) considered three ways of mapping asynchronous 2.048 Mbit/s tributaries into SDH VC-12: asynchronous, bit synchronous and byte synchronous [14]. The authors focus on the asynchronous mapping, discussing positive/zero/negative justification to improve the capacity of elastic buffer store. Fuqiang et al. (1996) provide a similar analysis, quantifying the best elastic buffer sizes for PDH to SDH and SDH to PDH conversions [15].

Xiaoru and Lieguang (1996) prototyped and verified an SDH STM-1 in 8 3190/3090 Xilinx FPGAs [16]. The final target implementation was a 0.5μm three-layer

CMOS gate-array. Thalmann et al. (1999) report an architecture of an Add-Drop/Terminal-Multiplexer for SDH, allowing to integrate all digital functions into one ASIC [17]. The idea is based in two approaches: buffer usage optimization and embedded processor, which substitutes various large hardware blocks.

Clauberg et al. (1999) introduced a scalable modular architecture for SDH/SONET technology, by exploiting the regular multiplexing principle inherent to this hierarchy [18]. They demonstrated the feasibility of their architecture with a framer chip, able to handle 4 STM-1 and variations of STM-4. Herkersdorf et al. (2000) complemented Clauberg works by covering the mapping of ATM, IP and T1/T3 traffic streams into SDH/SONET ranging from OC-1 to OC-48/STM-16 [19].

Röwer et al. (2000) implemented an SDH/SONET input block using a new paradigm called *programmable intellectual property* [20]. In it, modules can be reconfigured by downloading new software versions into IP embedded processors.

Peng, Depeng and Lieguang (2000) developed THXC, an SDH cross-connected ASIC with an embedded BIST circuit [21]. THXC is programmable and monitored by an external computer, designed to allow various switching rates and cascadability.

Silveira and Van Noije (2000) presented the modeling of an E1 mapper for SDH Systems, pointing out the difficulties to implement them, due to the synchronization mechanisms and the nature of the PDH information carried in SDH frames [22].

Baechtold et al. (2001) implemented a single-chip for 4xOC-3c, OC-12c SDH/SONET framing [23]. The chip features: low power consumption, integrated clock recovery that fulfils the ITU-T, Bellcore and ANSI jitter requirements, and functions to enable low-cost digital cross-connect and add/drop multiplexing systems.

The multiplexing section overhead (MSOH) is a central part of SDH circuits since it treats many frame errors. Torres et al. (2003) presented an MSOH processor for STM-0/STS-1 to STM-4/STS-12 [24]. Their work purpose was to show the requirements specification, architecture and verification of such systems.

The present work introduces the EMS architecture, which is similar to Peng et al. (2000) system with the scalability introduced by Clauberg et al. (1999). In addition, the proposed approach is distinct from that of Yongming et al. (1996) and Fuqiang et al. (1996), achieving better elastic buffers sizing for PDH to SDH mapping and vice-versa.

5. The EMS Architecture

EMS is a scalable architecture, allowing from 1 to 63 E1 channels mapping/demapping into/from an SDH frame. The smallest functional EMS operates with one E1 channel and is called *Basic EMS*, or simply BEMS. When EMS operates with full capacity (63 E1 channels), it is capable of dealing with an entire SDH STM-1 being called in this case *STM-1 EMS*, or simply SEMS.

The BEMS external interface is comprised by three signals sets as depicted in Figure 4 and described in Table 1. The first set, composed by `RST_N`, `CK32_768` and `CK65_536` signals, is used to perform system control and synchronization. The second set is composed by `DTBYCK`, `DTBPAY`, `DTBJ0J1`, `DTBDATA` and `DTBDATAOUT` signals. These signals implement a partial Telecom Bus interface. Telecom Bus is a byte-wide parallel format that serializes and deserializes data and identifies the J0 and J1 pointers in the SDH frame, reducing the core operating frequency from 155.52 MHz to 19.44 MHz [11]. The third set is composed by `E1IN`, `CKE1IN`, `E1OUT`, `CKE1OUT` and `CHANNEL` signals. These signals implement the E1 interface.

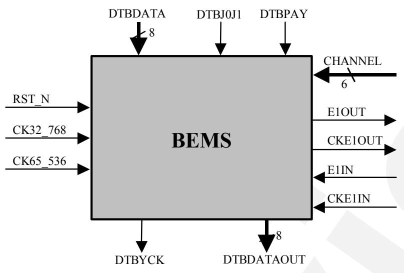

Figure 4 – BEMS external interface.

Table 1 – Signals description of BEMS interface.

Pin	Description	Signal set
RST_N	System reset	Control and synchronism signals
CK32_768	32.768 MHz clock	
CK65_536	65.536 MHz clock	
DTBYCK	19.44 MHz clock	Telecom Bus signals
DTBPAY	Marks the payload area	
DTBJ0J1	Marks the J0 and J1 bytes occurrence	
DTBDATA	Data from Telecom Bus	
DTBDATAOUT	Data to Telecom Bus	
CHANNEL	Number of add/drop channels	E1 signals
E1IN	E1 input frame	
CKE1IN	E1 input frame clock	
E1OUT	E1 output frame	
CKE1OUT	E1 output frame clock	

To implement E1 channel mapping into SDH frames, the EMS core adds data from an E1 channel (`E1IN` signal) to the Telecom Bus (`DTBDATAOUT` signal). The E1 demapping from SDH is implemented by receiving data from the Telecom Bus (`DTBDATA` signal) and mapping these to an E1 channel (`E1OUT` signal). The selected E1 channel is addressed by the `CHANNEL` signal, which allows specifying one out of 63 channels.

Five main modules and an auxiliary logic circuit (channel multiplexing, buffering, some control, and glue logic) compose the BEMS system (see Figure 5):

(i) *ColumnAddress* – identifies the TU-12 start pointer in the Telecom Bus;

(ii) *Delay* – signals the moment to insert data in a valid VC-4 column;

(iii) *V5Enable* – has three basic functions: *(a)* storing the SDH data when its address corresponds to the channel to be changed; *(b)* indicating when a column is valid; *(c)* indicating the super-frame start;

(iv) *VC12Drop* – extracts data from the Telecom Bus and sends them to the corresponding E1 channel;

(v) *VC12Add* – receives data from an E1 channel and inserts them into VC-12, with frequency justification, if necessary.

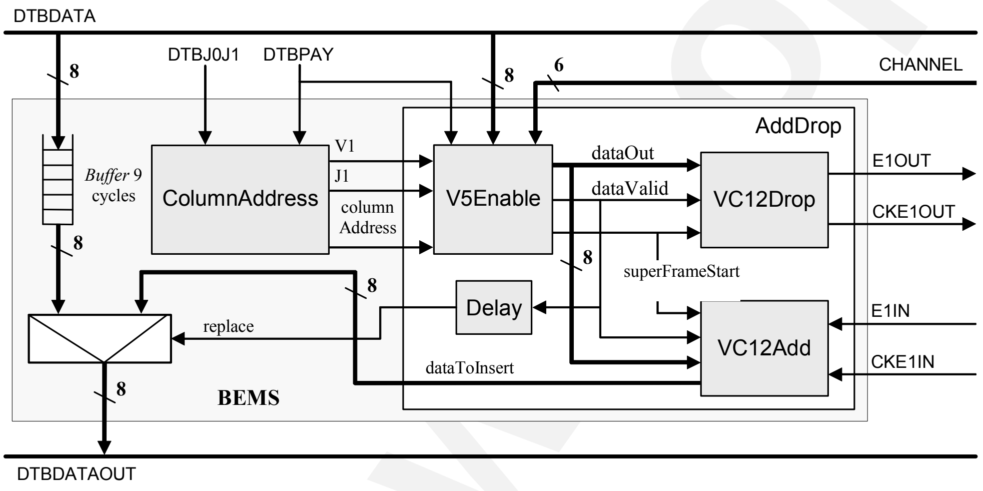

Figure 5 – BEMS internal modules organization.

Delay, V5Enable, *VC12Add* and *VC12Drop* modules are encapsulated in a module called *AddDrop*. This hierarchical structure provides scalability to EMS. To increase the number of E1 channels to *n* it suffices to instantiate the *AddDrop* module *n* times together with adding an extra structure to control DTBDATAOUT multiplexing. Figure 6 depicts SEMS, composed by 63 *AddDrop* modules.

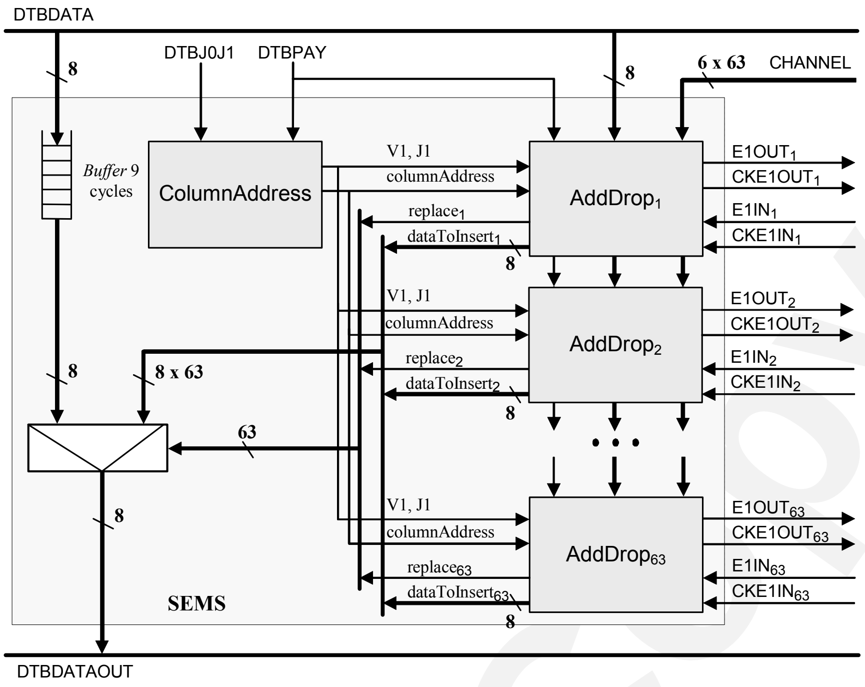

Figure 6 –SEMS internal module structure.

5.1 ColumnAddress Module

The *ColumnAddress* module receives as inputs the Telecom Bus control signals and generates the J1 pointer (payload start), V1 pointer (TU-12 start) and the number of each column present in the payload (through the `columnAddress` signal). The column number allows the system to locate specific data of a channel in the Telecom Bus. The J1 pointer corresponds to the first column number. The V1 pointer allows the system to locate the VC-12 internal pointers (e.g. V5 pointer).

5.2 Delay Module

The *Delay* module generates the `replace` control signal, responsible for defining the exact moment to insert data in a valid VC-4 column.

5.3 V5Enable Module

The *V5Enable* module searches for valid data (`dataValid` signal) in the Telecom Bus. It also indicates the super-frame start (`superFrameStart` signal). In addition, the *V5Enable* module stores data from Telecom Bus and forwards them to *VC12Drop* and *VC12Add* modules. The module also detects valid columns through a table mapped into a ROM-like structure. The table contains the first TU-12 channel number inside VC-4. For example, 9 (“0001001”) is the channel 1 address (second position of the partial VHDL code in Figure 7). The first word of the ROM corresponds to the channel 0 and is not used. The remaining 63 words correspond to each one of the 63 channels inside of VC-4 payload.

```
constant channel_position: rom :=
  ("1111111", -- don't used
   "0001001", "0011110", "0110011", "0001100", "0100001", "0110110", "0001111", "0100100",
   "0111001", "0010010", "0100111", "0111100", "0010101", "0101010", "0111111", "0011000",
   "0101101", "1000010", "0011011", "0110000", "1000101", "0001010", "0011111", "0110100",
   "0001101", "0100010", "0110111", "0010000", "0100101", "0111010", "0010011", "0101000",
   "0111101", "0010110", "0101011", "1000000", "0011001", "0101110", "1000011", "0011100",
   "0110001", "1000110", "0001011", "0100000", "0110101", "0001110", "0100011", "0111000",
   "0010001", "0100110", "0111011", "0010100", "0101001", "0111110", "0010111", "0101100",
   "1000001", "0011010", "0101111", "1000100", "0011101", "0110010", "1000111");
```

Figure 7 – ROM-like structure containing TU-12 addresses.

Based on the first occurrence of each channel, the *V5Enable* module computes the others three TU-12 occurrences into the VC-4 payload, by adding 63, 126 and 189 to the first column value, as is illustrated in the VHDL code of Figure 8. The `c1` signal corresponds to the value obtained from the ROM structure. The `c2`, `c3` and `c4` signals correspond to the other three addresses.

```
c1 <= ("00" & channelPosition(channel));
c2 <= ("00" & channelPosition(channel)) + "000111111";
c3 <= ("00" & channelPosition(channel)) + "001111110";
c4 <= ("00" & channelPosition(channel)) + "010111101";

validColumn <= '1' when(columnAddress=c1 or columnAddress=c2 or
                        columnAddress=c3 or columnAddress=c4)
                        and dtbPay = '1'
                else '0';
```

Figure 8 – VHDL code for computation of the channel address.

To extract VC-12 from TU-12 it is necessary to remove the bytes corresponding to V1, V2, V3 and V4 pointers that are represented by TU-12 PTR in Figure 2. These 4-byte positions are also provided by the `columnAddress` signal. It is possible to observe in the partial VHDL code of Figure 9 that data is valid (`dataValid` = 1) only if its address is different from the pointer addresses (1, 36, 72 and 108, respectively). In this case, a valid data is provided through the `dataOut` signal.

```
if(validColumn = '1') then
  if(v1tmp = '1' OR counter144 = 36 OR counter144 = 72 OR counter144 = 108) then
    dataValid <= '0';
  else
    dataValid <= '1';
    dataOut <= dtbDATA;
  end if;
else
  dataValid <= '0';
end if;
```

Figure 9 – Extraction of valid data.

The V5 pointer address (represented by TU-12 POH in Figure 2) is computed joining the two least significant bits of the V1 pointer and all eight bits of the V2 pointer. This results in an offset, shown in Figure 3. The VC-12 address is obtained starting at the V2 pointer. For example, if the address of V5 pointer is 0, it means that this pointer is located in the first byte after the V2 pointer. The pointers V1, V2, V3 and V4 must be skipped. When the offset is between 0 and 34, it means that V5 starts after V2 and before V3 pointers, being necessary to add 37 to the offset, to compose the V5 pointer address. When the offset is between 35 and 69, it is necessary to add 38 to the offset, skipping the bytes above the V3 pointer. The last case is when the offset is between 105 and 139. In this case, the offset must be decreased by 104. The resulting address will be between V1 and V2 pointers. To search the 144 bytes of TU-12, this module uses a counter called `counter144` (see Figure 9). The `counter144` counter is set to 0 when the V1 pointer is detected. When the value of this pointer is equal to the V5 pointer address, the beginning of the super-frame is signaled through the `superFrameStart` signal.

5.4 VC12Drop Module

The *VC12Drop* module is responsible for extracting data from the Telecom Bus and sending these to the E1 output channel. This module has an internal 64-bit circular FIFO buffer. The write pointer starts pointing at position 0. To avoid data loss, the FIFO is read only when half of it is written. In other words, when the write pointer reaches position 32, the system starts the drop operation. After this, the FIFO reading operation is continuously active, and the reading clock is adjusted according to the difference between the write and read pointers, expressed by the DELTA signal. This operation performs zero/negative/positive frequency justification. If DELTA is greater than the FIFO length, there is data loss. If DELTA is too small, the system can drop wrong data. The FIFO is dimensioned to avoid these problems.

A hysteresis mechanism was implemented to control the variation of the DELTA signal. It allows keeping minimal and maximal distances between read and write pointers before executing positive or negative justifications. Figure 10 shows this behavior. When the DELTA signal is between the minimal and maximal values, the system operates at the nominal frequency (2.048 MHz). To obtain a 2.048 MHz clock, the 65.536 MHz reference clock is divided by 32. When the DELTA signal reaches the maximal hysteresis value (48), the reading frequency (CKE1OUT signal) must be increased, and when the DELTA signal reaches the minimal hysteresis value (16), the reading frequency must be decreased.

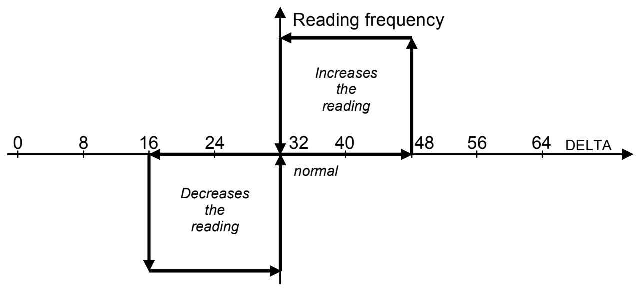

Figure 10 – Hysteresis behavior for VC12Drop.

As depicted in Figure 3, the control justification bits (C_1s and C_2s) indicate if S_1 and/or S_2 are valid data bits. When the justification bits indicate valid data, bits contained in S_1 and/or S_2 must be written to the FIFO. At the nominal frequency, only one of them is valid data. When S_1 and S_2 are valid data, the amount of data to write in the FIFO is increased, and the DELTA signal is also increased, reaching the maximal hysteresis value. In this case, the reference clock is divided by 31, to obtain a 2.114 MHz frequency. This higher frequency reduces the DELTA signal to the normal range thus recovering the nominal frequency. When S_1 and S_2 are not valid data bits, they are not written to the FIFO, making the DELTA signal reach the minimal hysteresis value. In this case, the CKE1OUT frequency must be decreased, dividing the reference clock by 33, to obtain a 1.986 MHz frequency. This lower frequency increases the DELTA signal to the normal range and as a result, the nominal frequency is recovered.

The FIFO and hysteresis limits have to be dimensioned to avoid data loss, increased latency and memory usage. Short FIFOs can cause data loss, since bytes are written in burst using the 19.44 MHz clock frequency, while reading is continuously performed at a 2.048 MHz clock frequency.

Large FIFOs can increase memory cost and latency, and lead to improper operation. This is due to the time that the output frequency stays different from the nominal. Even with good FIFO dimensioning, the circuit may operate improperly due to the difference between minimum and maximum hysteresis limits. If the limits are too near or too far to/from each other, the output frequency will change too fast or too slow, violating the standard. This may, in turn, damage the signal recovery by an external E1 processing module.

Figure 11 shows the DELTA signal behavior for nominal clock operation, considering a FIFO with 64 bits, and maximum, medium and minimum hysteresis limits of 48, 32 and 16, respectively. Experimental data showed that the chosen hysteresis limits are adequate to respect ITU-T standards.

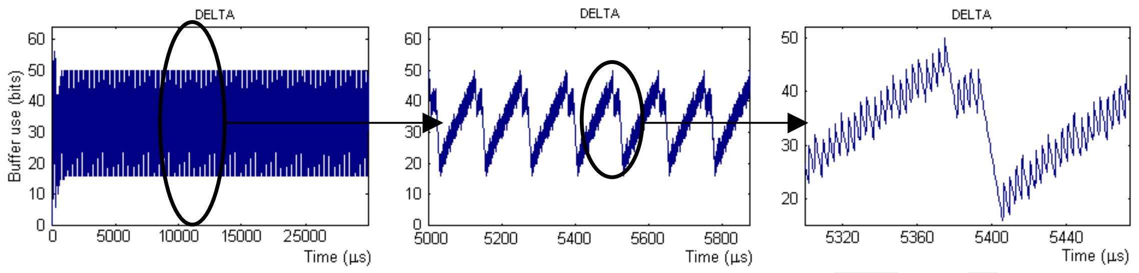

Figure 11 – DELTA signal behavior for VC12Drop nominal clock, for different time scales.

Figure 12 shows a partial simulation of the FIFO operation. The `readCount` signal (read pointer) is incremented at each `CKE1OUT` cycle, because the FIFO is always being read. The FIFO is written in bursts only when there is information to be dropped to the external E1 channel (i. e. when the `flagwrite` signal is equal to 1). Salient features in the simulation are numbered in Figure 12 and explained below.

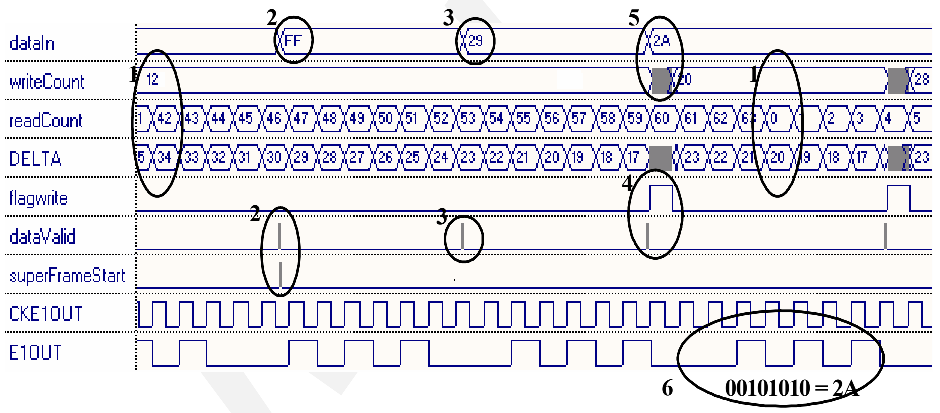

Figure 12 – Simulation showing FIFO writing by the VC12Drop module.

1. `DELTA` is the difference between `writeCont` and `readCount` with regard to the FIFO size. Since the buffer is implemented as a circular FIFO with 64 bits, there are two formulae to compute `DELTA`: (*i*) If `readCount` > `writeCont`, then `DELTA` ← FIFO size – `readCount` + `writeCont` (e.g. 64 – 42 + 12 = 34); (*ii*) Else `DELTA` ← `writeCont` – `readCount` (e.g. 20 – 0 = 20);
2. `superFrameStart` indicates the beginning of a super-frame, pointing to the V5 pointer, that is equal to FF (FF is a value used just for simulation purposes);
3. `dataValid` indicates the occurrence of valid data in the Telecom Bus. Since this is a control byte (VC-12 POH), the `flagwrite` is not active and this information is not written to the FIFO;
4. `dataValid` indicates the occurrence of valid data in Telecom Bus. In this moment, Telecom Bus contains payload data, activating `flagwrite`;
5. There are valid data in Telecom Bus and this hexadecimal value (2A) is E1 payload information, written in burst to the FIFO;

6. The data (2A) is extracted from the FIFO, and serially written into the E1 output channel. It is important to stress that this data is not the same data obtained in step 5, since there are other bytes in the FIFO with the same value.

Figure 13 shows another example simulation of the VC12Drop module, highlighting the frequency justification operation. The mark 1 shows the reference clock divided by 32 (`limitCounterClock` is 31, meaning that the range of the counter is from 0 to 31). Mark 2 shows the reference clock being divided by 33 (`limitCounterClock` is 32). This new value of `limitCounterClock` implies an increase of the clock period, meaning that fewer bits will be consumed by the E1 output channel.

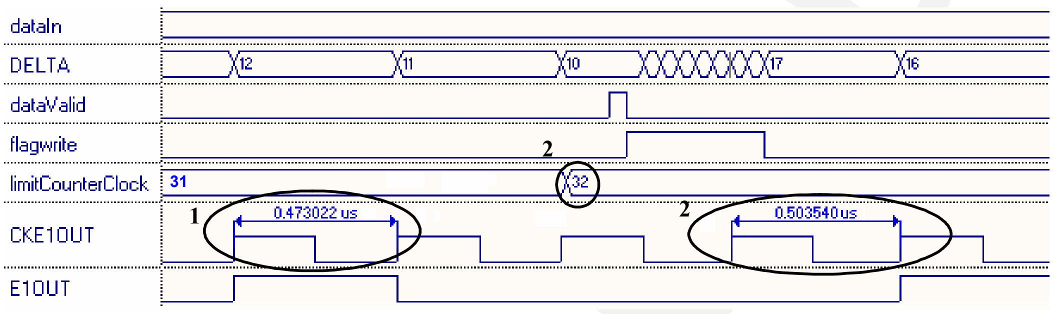

Figure 13 – Simulation of VC12Drop module during execution of a frequency justification operation.

5.5 VC12Add Module

The function of the *VC12Add* module is to insert data from an external E1 channel into the Telecom Bus. Data are received at CKE1IN rate, an operating frequency that may vary. According to this variation, the *VC12Add* module executes the frequency justification through S_1 and S_2 justification opportunity bits, and C_1 and C_2 justification control bits. The justification frequency is based on a hysteresis mechanism, analogous to the one used by the *VC12Drop* module. For data insertion, a 128-bit FIFO is needed, with minimum and maximum sizes of 32 and 96, respectively. The FIFO size of the *VC12Add* module is bigger than the corresponding FIFO of the *VC12Drop* module. This occurs as a result of the analysis of worst case synchronization conditions between PDH to SDH. These results pointed that during the Add operation the number of bits inserted may vary more widely than during the Drop operation.

When PDH and SDH clocks operate at their respective nominal frequencies, only one of S_1 or S_2 is used as valid data bit. When the CKE1IN frequency is higher than the nominal value, the amount of data written into the FIFO is increased and the DELTA signal reaches the maximum hysteresis value. This increases the reading of E1 data input and adds to the amount of data into Telecom Bus, avoiding data loss. These extra

Telecom Bus data must be inserted into S_1 and S_2 bits and the justification control bits must be set to 1. When the CKE1IN frequency is below the nominal value, the amount of data written to the FIFO is less than the amount of data read. As a result, the DELTA signal reaches the minimum hysteresis value. To avoid reading incorrect data, it is necessary to decrease the Telecom Bus reading speed. Thus, S_1 and S_2 bits are not filled with valid data and the justification control bits are set to 0.

Figure 14 illustrates the DELTA signal behavior for the *VC12Add* module, considering nominal (2.048 MHz), high (2.050 MHz) and low (2.046 MHz) E1 input clock values. With CKE1IN low frequency there are more Telecom Bus data processing than E1 input data generation. To avoid reading incorrect data, the *VC12Add* module reduces the number of valid data bits in each super-frame to 1023. Exactly the opposite happens when the E1 input clock (CKE1IN) has a frequency higher than the nominal value. To avoid data loss, the *VC12Add* module increases the number of valid data bits in each super-frame to 1025.

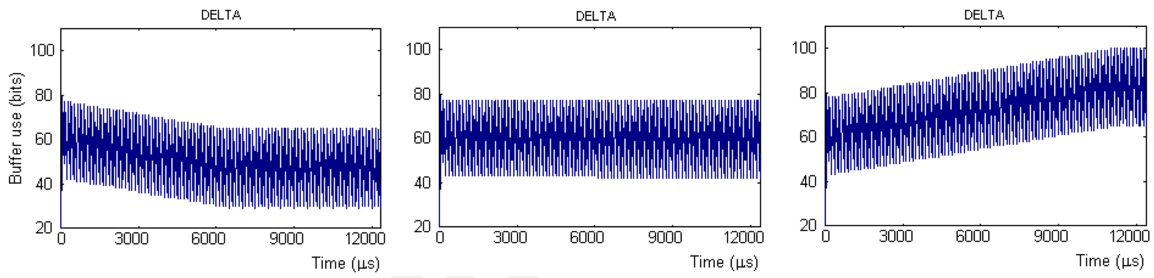

Figure 14 – DELTA signal behavior for VC12Add module, considering slow, nominal, and fast E1 input clocks, respectively.

6. Validation and Prototyping

One major difficulty with the EMS functional validation step is the number of simulation cycles needed to verify each design aspect, and the huge amount of output data produced and consumed during a simulation expected to provide even a moderate covering of the design characteristics. To alleviate the problem, external software was written. A parameterizable *test pattern generator* software creates test multiframes according to the parameters and the circuit under test. A *test pattern analyzer* software compares simulation results (output files) against the input files and input parameters.

The validation process was conducted in three scenarios of increasing complexity. The first scenario is a *loop test*, that considers the Telecom Bus and AddDrop circuits separately. The Telecom Bus circuit corresponds to the left side of

Figure 5, comprising the buffer, column address and multiplexer. The second scenario, *BEMS test*, evaluates the mapping/demapping of one E1 channel into/from an SDH frame. The last scenario, *global test*, evaluates the effects of mapping/demapping multiple E1 channels into/from an SDH frame.

In the *loop test*, depicted in Figure 15, the goal is to test the VC12 FIFO operations and the SDH bypass path. The testbench code in this scenario has also a set of VHDL assert conditions to detect exceptions and critical operations (e.g. DELTA errors).

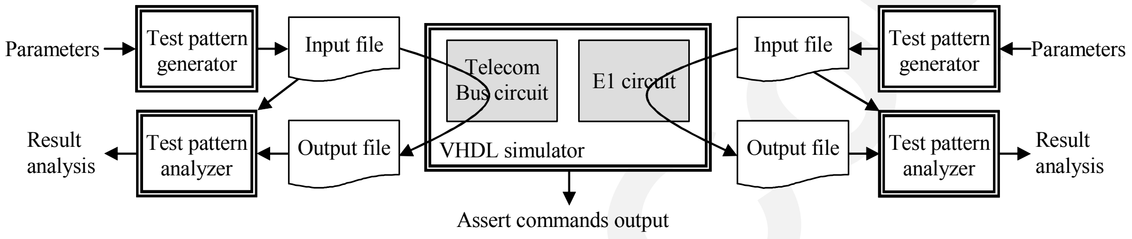

Figure 15 – Loop test for the functional validation process.

Next, the *BEMS test* is performed. There are two independent paths to exercise. Refer to Figure 16 for information about this and the *global test* scenarios. From the E1 input channel to the SDH frame, the generated data is packed into SDH as detailed in Figure 2. The test pattern analyzer software extracts the E1 information from the SDH output file and compares it to the E1 input file. From the SDH frame to the E1 output channel, the generated data is unpacked from the SDH frame to E1. The test pattern analyzer compares the unpacked data against the SDH input file. The *global test* is a generalization of the *BEMS test*, as can be perceived in Figure 16.

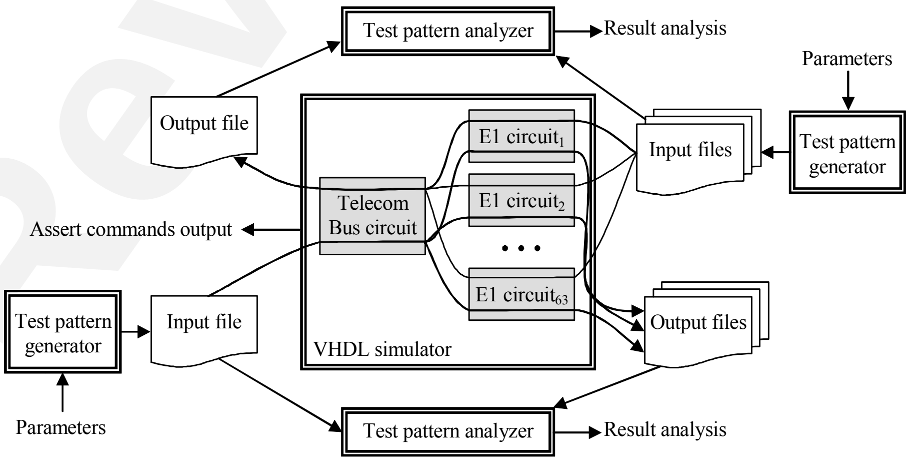

Figure 16 – EMS test and global test structure for the functional validation process.

BEMS has been described in 2150 lines of RTL VHDL. The description is portable, except for the FIFOs circuits, implemented using Xilinx FPGAs Block SelectRAM primitives. Once the design was validated at the functional level, the EMS was prototyped and validated in hardware. The VCC VW-300 prototyping platform was employed. This board contains a 300,000-gate Virtex FPGA. The BEMS design occupies 314 slices of 3,072, i.e. 10% of the FPGA device. The system is operational, fulfilling all design constraints of the original specification. Since the SDH is relatively small and has small latency (9 clock cycles), it allows cascading several instances of it in a single system.

7. Conclusions

The main contributions of this work are the development of: (*i*) The EMS soft-core, which performs mapping and demapping of E1 channels into/from SDH; (*ii*) A buffer technique for enhanced frequency justification control; (*iii*) A sophisticated validation technique to allow extensive tests of the circuit characteristics.

The developed core allows 9 cycles for frame propagation latency for any number of VC-12 implemented in STM-1, since all VC-12 circuits operate concurrently. Besides the low propagation latency, the EMS core has a small size, enabling its implementation in low-cost programmable devices.

References

[1] Ming-Chwan Chow. *Understanding SONET/SDH. Standards and Applications*. **Andan Publisher**, New Jersey, 1996.
[2] This reference has been omitted for the purpose of blind review.
[3] R. A. Bergamaschi, W. R. Lee. *Designing Systems-on-Chip Using Cores*. **37th Design Automation Conference (DAC 2000)**, pages: 420-425, 2000.
[4] International Technology Roadmap for Semiconductors, http://public.itrs.net, 2003.
[5] Xilinx, Inc. *Investor Fact sheet - Third Quarter Fiscal Year 2003*, http://www.xilinx.com, 2003.
[6] ITU-T, *Network Node for the SDH. Recommendation G. 707.* **Telecommunication Standardization Sector of ITU**, 1996.
[7] ITU-T, *Characteristics of SDH equipment functional blocks. Recommendation G. 783.* **Telecommunication Standardization Sector of ITU**, 1997.
[8] ITU-T, *Architecture of Transport Networks Based on the Synchronous Digital Hierarchy. Recommendation G. 803.* **Telecommunication Standardization Sector of ITU**, 1996.
[9] ITU-T, *Timing Characteristics of Primary Reference Clocks. Recommendation G.811.* **Telecommunication Standardization Sector of ITU**, 1997.
[10] PMC-Sierra Inc. PM5371 TUDX, Data Sheet, September 1998.
[11] Intel Corporation. IXF6151 28 T1/E1 Mapper, Data Sheet, January 2001.
[12] Transwitch Inc. Ethernet into STS-3/STM-1 SONET/SDH Mapper (TXC-04226B), Data Sheet, 5th ed, January 2004.

[13] Chia-Wen Lin, Wen-Hsien Hsu, Chang-Ching Wu, Shih-Chun Wang. *The Study of SDH STM-1 Add-Drop Multiplexer Architecture*. **Conference Singapore (ICCS'94)**, volume: 1, pages: 177-181, November 1994.

[14] Xu Yongming, Zhang Xiaopin, Ye Peida. *Asynchronous Mapping of 2.048 Mbit/s Tributary into SDH VC-12*. **Proceedings of International Conference on Communication Technology (ICCT'96)**, volume: 2, pages: 817-820, May 1996.

[15] Shi Fuqiang, Lin Xiaokang, Feng Chongxi. *Design of the Elastic Buffer Size for SDH Equipments*. **Proceedings of International Conference on Communication Technology (ICCT'96)**, volume: 2, pages: 800-804, May 1996.

[16] Zhang Xiaoru, Zeng Lieguang. *An SDH STM-1 Termination IC*. **2nd International Conference on ASIC**, pages: 179-182, October 1996.

[17] M. Thalmann, M. Stadler, T. Röwer, N. Felber, W. Fichtner. *A Single-Chip Solution for an ADM-1/TMX-1 SDH Telecommunication Node Element.* **Proceedings of the Twelfth Annual IEEE International ASIC/SOC Conference**, pages: 147-151, September 1999.

[18] R. Clauberg, A. Herkersdorf, W. Lemppenau, H. R. Schindler. *A Scalable Modular Architecture for SDH/SONET Technology.* **Proceedings of Eight International Conference on Computer Communications and Networks**, pages: 442-446, October 1999.

[19] A. Herkersdorf, P. Buchmann, R. Clauberg, W. Lemppenau, H. R. Schindler, D. Webb. *A Scalable SDH/SONET Framer Architecture for Datacom and Telco Applications*. **Proceedings of the International Zurich Seminar on Broadband Communications**, pages: 191-198, February 2000.

[20] T. Röwer, N. Stadler, M. Thalmann, H. Kaeslin, N. Felber, W. Fichtner. *A New Paradigm for Very Flexible SONET/SDH IP-Modules.* **Proceedings of the IEEE on Custom Integrated Circuits Conference (CICC 2000)**, pages: 533-536, May 2000.

[21] Rong Peng, Jin Depeng, Zeng Lieguang. *Design and Applications of SDH Cross-Connection ASIC*. In: Proceedings on International Conference on Communication Technology (WCC-ICCT 2000), volume: 2, pages: 1041-1045, August 2000.

[22] R Silveira e W A. M. Van Noije. *Modeling an E1/TU12 Mapper for SDH Systems*. **Proceedings of 13th Symposium on Integrated Circuits and Systems Design**, pages: 171-176, September 2000.

[23] P. H. Baechtold et al. *Single-Chip 622-Mb/s SDH/SONET Framer, Digital Cross-Connect and Add/Drop Multiplexer Solution.* **IEEE Journal of Solid-State Circuits**, volume: 36, issue: 1, pages: 74-80, January 2001.

[24] D. Torres, A. Redondo and M. E. Guzmán. *MSOH processor for STM-0/STS-1 to STM-4/STS-12: component of a SDH/SONET library*. **Microelectronics Reliability**, volume: 43, issue: 2, pages: 217-223, February 2003.