Implementation of a SDH STM-N IC for B-ISDN Using VHDL Based Synthesis Tools

J.Carlos Calderón, José M. Tapia, Enric Corominas, Lluis París[1]

Centre Nacional de Microelectrònica, CNM-CSIC.
[1]Universitat Autònoma de Barcelona
Campus Universitat Autònoma de Barcelona 08193 Bellaterra, Barcelona, SPAIN

Abstract

A general VHDL description, allowing to synthesize a set of B-ISDN user network interface (UNI) ICs for the different SDH levels (155, 622,... Mbit/s), is presented. The synthesized UNI ICs allow a STM-1 on STM-N frame multiplexation or to directly perform the payload mapping of N 155 Mbit/s ATM cell streams on a STM-N frame in a compact and cost-effective manner.

This work focuses on three points: a comparison between both hardware methods, a discussion on the suitableness of using VHDL, and the choice of the appropriate design-validation methodology .

1: Introduction

The future broadband ISDN (B-ISDN) is conceived as an all-purpose digital network. It will provide an integrated access that will support a wide variety of services for its customers (such as video, data and voice) included emerging new services with yet unknown characteristics [1]. New worldwide standards (SDH, ATM) have been established to specify the transmission bit rates and frame formats at the two lowest protocol layers.

SDH (synchronous digital hierarchy) is a digital transmission standard that defines a new digital hierarchy for the transmission on physical networks and a frame structure for multiplexing digital traffic. In North America it is called the synchronous optical network (SONET) [2,3]. The synchronous transport module level 1 (STM-1) at 155.52 Mbit/s is the basic SDH signal. All other signals in the hierarchy are integer multiples of the STM-1, and are named STM-N. Two new signals are just defined in the hierarchy: STM-4 at 622.08 Mbit/s, and STM-16 at 2.48832 Gbit/s.

ATM (asynchronous transfer mode) is the standardized transport, multiplexing and switching technique proposed as the basis for B-ISDN. It employs fixed-size packets, or "cells", with a 5-byte header and a 48-byte information payload. The current recommendations include SDH as the physical layer transmission standard. It is defined the B-ISDN user network interface (UNI) SDH-based at 155.52 Mbit/s, but there is a straightforward way of creating a 622.08 Mbit/s (or 2.48832 Gbit/s) frame from four (or sixteen) STM-1s according to [4].

We have developed a general VHDL description that, by means of a VHDL synthesis tool, allows the designer to obtain a set of B-ISDN UNI ICs for the different SDH levels. The synthesized ICs perform the SDH overhead processing and provide payload mapping of ATM cells. Two main reasons have motivated this work: the implicit scalability of the SDH standard is a very suitable feature to be captured with the descriptive capability of VHDL. On the other hand, although at this time we need only the SDH basic rate (this work belongs to a project called UNICORN[1]), in the future this project will require the design of UNI ICs at greater SDH rates.

The paper is structured as follows. Section 2 is a brief description of the SDH frame format, section 3 compares two different hardware strategies, section 4 explains how VHDL solves the relevant design problems, section 5 explains the used design-validation methodology, section 6 defines a set of functions to be performed by a cost-effective framer, section 7 presents some experimental results, and finally, conclusions are shown in section 8.

2: STM-N frame structure and functions

At the physical level the B-ISDN UNI has a bit rate of 155.52 Mbit/s, or 622.08 Mbit/s [5], but only in the first case the effective bit rate (excluding the physical layer related information) is defined: its value is 149.76 Mbit/s and complies with SDH. A scheme of the SDH STM-N frame structure, and the payload mapping of ATM cells is shown in Fig. 1. The SDH frame (transmitted every 125µs) is byte-structured, and consists of 9 rows of 270*N columns. The first 9*N columns comprise the section overhead (SOH) and the administrative unit (AU) pointers.

[1]This work is supported by the CICYT project TIC- 1249/93.

The remaining 261*N columns are the synchronous payload envelope (SPE). In the SPE N columns are dedicated to the path overhead (POH) bytes, and 260*N columns (for STM-N carrying ATM only) constitute the payload (2340*N payload bytes), a mapping of ATM cells in the case of the B-ISDN SDH-based UNI.

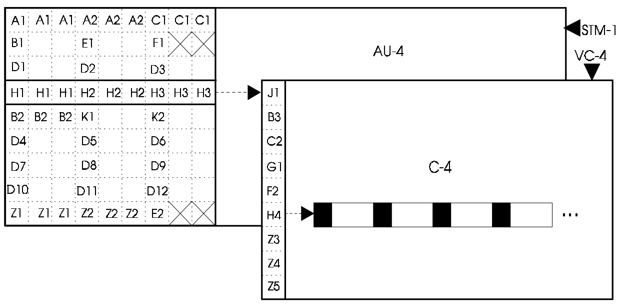

Fig.1. STM-1 frame structure

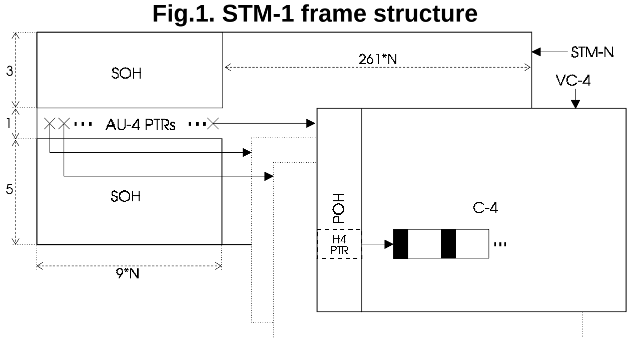

Fig.2. STM-N frame structure

Overhead bytes provide functions such as framing and operation and maintenance. Next, the function of some of them is commented (as needed by section 4). The framing pattern, in the first SOH row, consists of 3*N A1 bytes and 3*N A2 bytes (A1=F6h, A2=28h). In transmission-reception, the STM-N frame is scrambled-descrambled. The only bytes left unscrambled are the identification A1, A2 & C1 bytes. Byte B1 in SOH, 3*N B2 bytes in SOH, and byte B3 in POH, are BIP-8 (bit interleaved parity 8 code) error monitoring bytes. As defined in [6], the AU-4 pointer (one pointer for each transported VC-4) value has a valid range from 0 to 782, and gives the offset, in 3-byte increments from the start of the SPE. Interpretation of the pointer word allows the receiver to decide different actions: to replace the current pointer value, to increment or decrement the current pointer value, etc.

3: STM-N framing. Hardware solutions

There are two basic approaches to perform the payload mapping of N 155 Mbit/s ATM cell streams onto a STM-N frame. In the first approach (the higher level one) the transmitter connects N STM-1-framer ICs to a high-speed frame multiplexer (MUX). The MUX combines the N parallel frames by synchronizing the N STM-1 framers and byte-interleaving their outputs [8]. In addition, the MUX must perform the STM-N scrambling process and the frame scrambling must be disabled in the STM-1 framers. In the receiver side, a demultiplexer (DMUX) performs frame detection on its incoming STM-N signal, and synchronizes the STM-1 receivers. Fig. 3 is an example of an STM-1 to STM-4 frame multiplexation, showing also the ATM ICs. A great disadvantage of this method is the need of using five ICs, increasing the cost, the PCB-area required and the system design complexity.

The second approach is the development of a single UNI IC that directly supports the payload mapping of N 155 Mbit/s ATM cell streams on a STM-N frame (see Fig. 4). This solution strongly reduces the PCB-area required and the system design complexity, and it also reduces the need for passing critical timing information on board, but it is necessary to know how expensive the STM-N framer can result. In order to maximize the system performance, the framer must be kept to the simplest form. The reason is that a portion of its circuitry works at the line rate, forcing the use of an expensive high-speed technology [9,10]. Obviously, an undesirable result is to obtain a cost (in terms of silicon area) comparable to N times the equivalent STM-1 framer cost. Besides, an excessive cost will lead to an unfeasible IC. Therefore, it is necessary to carefully define what functions will be performed by the UNI IC.

The developed VHDL description allows the user to synthesize framer ICs supporting both multiplexation strategies. For example, a STM-4 framer (622.08 Mbit/s) can be synthesized, and then can be connected as shown in Fig.3, obtaining a STM-16 framer (2.488 Mbit/s).

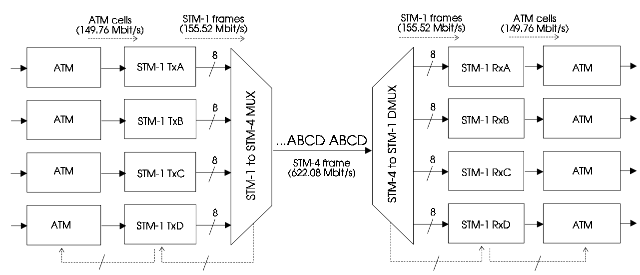

Fig.3. Multiplexation using a synchr-multiplexer IC

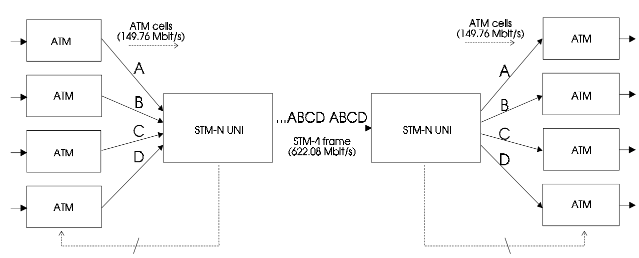

Fig.4. Multiplexation using a STM-N IC

4: VHDL suitability. Development and portability aspects

Widely found benefits of HDLs have been exposed by different authors [11,12]. Here we just present the relevant problems in the development of the general STM-N design (allowing to obtain different UNI ICs, with different functional requirements and working at different rates) and show how VHDL solves these difficulties.

The first problem is the high complexity of some UNI functions. The VHDL control-flow modeling capability allows the designer to describe complex functions in a very simple and suitable way, because of its structured control properties. An example is the frame scrambling and descrambling operations. The reference-model standarizes a synchronous scrambler [6, 13], with the polynomial x^7+x^6+1. This is a serial-defined method, but at the UNI IC it is preferable to perform this operation in a parallel manner (the traditional serial algorithm is too slow). A low-level solution is to calculate, with a program (which has to be written first) the parallel scrambler-descrambler equations, to simplify these, and then to introduce this result in the EDA-tool. Transforming these equations to their gate level equivalent is an error prone task. By using VHDL, a sparse description generates the same result in a direct way [14]. The following VHDL description is the synthesizable frame-scrambler/descrambler model:

```
entity scrambler is
    port (  clear, ck, reset, no_scramb : in Std_Logic;
            data_in : in Std_Logic_Vector(7 downto 0);
            data_out : out Std_Logic_Vector(7 downto 0) );
end scrambler;
architecture scrambler of scrambler is
signal r, r_new : Std_Logic_Vector(6 downto 0);
begin
    FF_scramb: process(ck, clear)       -- seq. circuitry
    begin
        if (clear = '0') then r <= "1111111";
        elsif (ck'event) and (ck = '1') then
            if (reset = '1') then r <= "1111111";
            else r <= r_new; end if;
        end if;
    end process;
    XOR_scramb: process(data_in, r, reset, no_scramb)
    variable t : integer range 0 to 7;
    variable r1, r2 : Std_Logic_Vector(6 downto 0);  begin
        r1 := r;
        for t in 0 to 7 loop        -- parallelism=8 bits
            -- Here the output is calculated
            if (reset = '0' and no_scramb = '0') then
                data_out(7-t) <= data_in(7-t) xor r1(6);
            else data_out(7-t) <= data_in(7-t); end if;
            -- Here the new scrambler-state is calculated
            r2(0) := r1(5) xor r1(6);
            r2(6 downto 1) := r1(5 downto 0);
            r1 := r2;
        end loop;
        r_new <= r1;
    end process;
end scrambler;
```

Modified versions of the previous model has been used to design several UNI-IC functions: the frame synchronous scrambler/descrambler, the ATM-cell information-field selfsynchronized scrambler/descrambler and the ATM-cell HEC (header error control) generation/verification. Another example is the AU-4 pointer processor (AU-4 PP). In order to carry out the standardized AU-4 PP algorithms [6], a set of actions must be performed when receiving the fourth SOH-row bytes (H1 and H2 octets). Table I shows the meaning of the H1 and H2 octets (the pointer word format). Interpretation of the pointer word follows the rules given in [6]:

octet	H1								H2							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
value	NDF				--		AU-PTR									
type	N	N	N	N	-	-	I	D	I	D	I	D	I	D	I	D

Table I

(1) If the NDF is "1001", the incoming AU-PTR value shall replace the current one.
(2) If the majority of the I bits of the 10-bit value are inverted, a positive stuff is indicated and the current pointer value is to be incremented in one.
(3) If the majority of the D bits of the 10-bit value are inverted, a negative stuff is indicated and the current pointer value is to be decremented in one.
(4) Any other variation of the pointer value is ignored unless a consistent new value is received in three consecutive frames.

Next, it appears a part of the AU-4 pointer processor VHDL model: a combinational circuit detecting the occurrence of these conditions.

```
signal ndf : boolean;     -- TRUE if "New Data Flag" detected
signal I : boolean;       -- TRUE if "positive stuff"
signal D : boolean;       -- TRUE if "negative stuff" detected
signal NP : boolean;          -- TRUE if "new pointer" detected
signal NPconst : boolean;   -- TRUE if constant new pointer
signal RxPerr : boolean;    -- TRUE if "out of range"
signal NDFin : Std_Logic_Vector(3 downto 0);
signal AUPTRin : Std_Logic_Vector(9 downto 0);
signal AUPTRnew : Std_Logic_Vector(9 downto 0);
signal AUPTR : integer range 0 to 782;      -- pointer value
```

```
Detect_NDF: process( NDFin )       -- "new-data-flag" detection
variable NDFactive : Std_Logic_Vector(3 downto 0);
variable i, Neq : integer;
begin
    Neq := 0; NDFactive := "1001";
    for i in 3 downto 0 loop
        if NDFin(i) = NDFactive(i) then Neq := Neq + 1; end if;
    end loop;
    ndf <= (Neq >= 3);             -- "new-data-flag" detection
end process;
detect_remaining: process(AUPTRin, AUPTRnew, AUPTR, ndf)
variable AUPTRin_int : integer range 0 to 1023;
variable AUPTRbits : Std_Logic_Vector(9 downto 0);
variable ND, NI, j : integer;
begin
    AUPTRin_int :=  bvToI(To_BitVector(AUPTRin));
    AUPTRbits := To_X01(iToBv(AUPTR, 10));
    ND := 0; NI := 0;
    for j in 0 to 4 loop
        if AUPTRin(2*j) /= AUPTRbits(2*j) then
        ND := ND + 1; end if;
        if AUPTRin(2*j+1) /= AUPTRbits(2*j+1) then
        NI := NI + 1;          end if;
    end loop;
    RxPerr <= AUPTRin_int > 782;           -- "out-of-range"
    D <= FALSE; I <= FALSE; NP <= FALSE;
    if not(ndf) then       -- HERE WE TAKE DECISIONS
        if (ND>=3) and (NI<=2) then        -- "negative stuff"
            D <= TRUE; RxPerr <= FALSE;
        elsif (NI>= 3) and (ND<= 2) then  -- "positive stuff"
            I <= TRUE; RxPerr <= FALSE;
        elsif (NI > 0) or (ND > 0) then       -- "new pointer"
            NP <= TRUE; end if;
    end if;
    NPconst <= AUPTRin = AUPTRnew;
end process;
```

The second problem takes root on the main project objective, to obtain the STM-N UNI generalized description. As shown on section 2, the STM-N frame can be represented in a 9 x (270·N) matrix-format. In a very simplified exposition of the problem, the difference among the STM-N (N=1, 4, ..) UNI ICs is just the number of columns to process (270, 1080, ..). Next, the principal STM-1 to STM-N design changes are briefly discussed.

Both the transmitter and the receiver will follow an extended column-count sequence (270*N). The framing-pattern at the first SOH-row is extended to a general synchronism-pattern of 3*N A1-bytes and 3*N A2-bytes. The BIP-24 code, located at the three SOH B2 bytes, is extended to a general BIP-24*N code. The AU-4 pointer processor is modified to work over a set of N AU-4 pointers. These processes have been implemented as simple FSMs working over N sets of AU-4 pointers, B2 bytes, etc. These data sets are structured in a ring, so that the required circuitry is minimized. In this case, the high data-modeling capability of VHDL has allowed us to obtain a general STM-N hardware description in a direct way. Multi-dimensional array definitions have been widely used in this work.

Next, the unavoidable technology-mapping problem appears. The different STM-N UNI ICs will be implemented using different technologies, because of the very different line-rates. In the STM-1 case (155.52 Mbit/s) a 1μm [8,15] or sub-μm [16] CMOS technology may be used. In the STM-4 case (622.08 Mbit/s) a nMOS [17], bipolar or BiCMOS technology is needed. In the STM-16 case (2.48832 Gbit/s) a GaAs [9] technology must be used. Therefore, to take advantage of a general STM-N framer model, an essential requisite is the technology independence. It is necessary to use a design-tool allowing to easily retarget the design to different vendor libraries. Once again it is attained with VHDL characteristics: capability to make technology-independent architecture selections is a well known VHDL feature.

Portability is a problem not yet solved. VHDL systems available today are not compatible enough: the synthesizable VHDL subsets are most times incompatible due to that designs can't be transferred from one environment to another. For example, predefined attributes are missing in some environments. Nevertheless, simulators are more compatible: it is possible to write VHDL stimulus and monitor models, allowing to verify models regardless of the simulation environment. In order to verify the designed UNI-IC the following stimulus and monitor VHDL models have been created in this project:

- A μC model. It is a simplified model of a single-chip 8-bit microcontroller. It allows sequences of read-write cycles, stored in text files (in a program-like manner).
- An ATM-IC model. It performs the transmission and reception of the ATM cells. The ATM-IC model transmit and receive text-files (simplifying the verification process).
- A mux-demux model.
- A stimulus-responses translator. This module generates a stimulus-responses text file in the logic simulator format (see section 5), in order to simulate the synthesized netlist with a non-VHDL simulator.

5: Design and validation methodology

Fig 5. shows the design and validation methodology. In the process shown in Fig.5 we have used two different design tools: the Compass Design Tools and the Vantage VHDL-simulator. We will explain how these tools have been used in the main steps (see the arrow-tags in Fig.5):

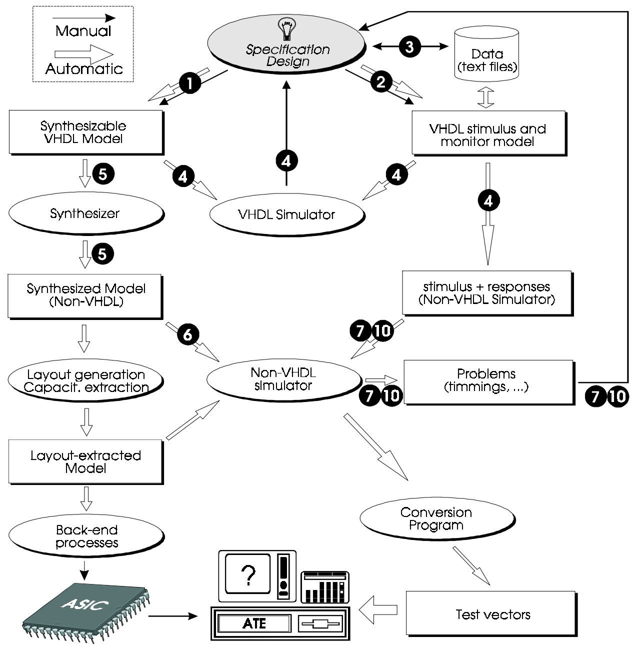

Fig.5. Design and validation methodology

- Steps 1 and 2. In order to simplify the design process a mixed textual-graphical description has been used: the structural description has been made graphically, drawing and interconnecting boxes. Compass automatically generates the structural description (VHDL files).
- Step 3. Additionally, the VHDL stimulus and monitor model can read and write data text-files.
- Step 4. The Vantage VHDL-simulator has been used in the functional validation process. A stimulus and responses text file in the Compass simulator format, QSIM, is generated at this stage.
- Step 5. Once the functional validation process ends, the VHDL model is synthesized, using the Compass Tools.
- Steps 6 and 7. In order to correct the possible temporal and electrical problems, the Compass logic-simulator is used. Compass allows to automatically generate a VHDL model of the synthesized circuit (a VHDL-netlist, with accurate timing models), but we do not use this alternative: the logic-simulator is much more CPU time-efficient.
- Step 8. The layout is generated (using Compass), and a layout-extracted model is also generated.
- Steps 9 and 10. Equivalent to steps 6 and 7.

6: Cost-effective STM-N UNI IC

As discussed in section 3, in order to maximize the performance of the system, it is necessary to limit the STM-N UNI IC complexity. According to B-ISDN protocol reference model, the physical layer of the UNI is split into two sublayers: the physical medium sublayer (PMS) and the transmission convergence sublayer (TCS). The PMS functions, bit-timing and physical-medium, are achieved by using a commercially available electro-optical converter. The TCS functions are: cell rate decoupling, HEC generation-check, cell delineation, transmission frame adaptation and frame generation-recovery.

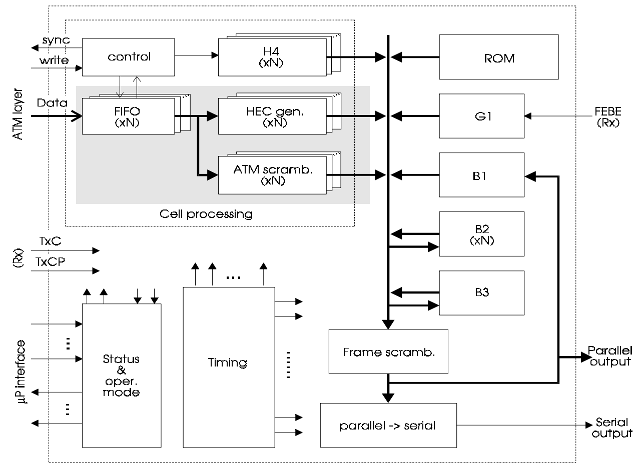

Fig.6. Transmitter

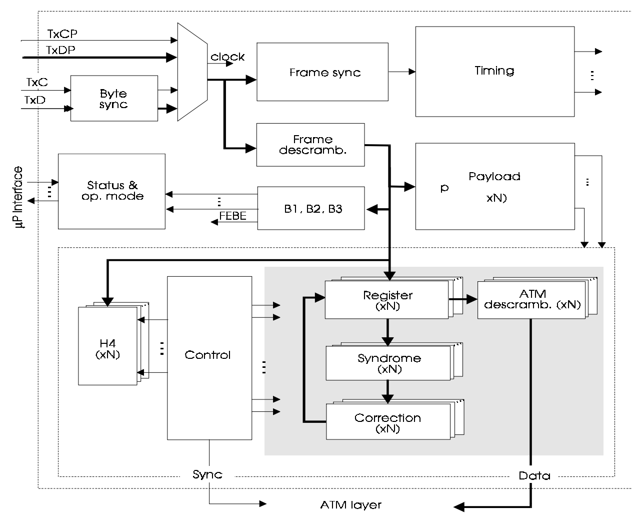

Fig.7. Receiver

Fig.6 and Fig.7 respectively show the transmitter and receiver UNI-architecture, supporting the whole TCS, as developed at UNICORN. The overlapped blocks at the figures must be repeated N-times in the circuit. In addition, this is an especially costly-hardware: a 67-byte FIFO (cell rate decoupling process) in the transmitter, a 13-byte FIFO in the receiver, parallel scrambling selfsynchronized, parallel HEC and syndrome generation, cell-header correction, etc. (these are very expensive blocks, as will be shown in section 7). The final conclusion is that an optimum/feasible STM-N framer must not include these blocks (the shaded region on Fig.6 and Fig.7). Therefore, the eliminated functions will be performed at the ATM IC, using a less expensive technology. In this way, only the general SDH functions and the POH-H4 pointer processing will be performed by the STM-N UNI IC.

Additional included functions in the designed STM-N UNI IC are the following:
- scrambled/descrambled operation. This mode can be used in parallel applications and also in the prototypes test.
- serial/parallel operation. The user can disable the serial-to-parallel and the parallel-to-serial converters. In this way, a set of STM-N ICs can be connected as depicted in Fig.3. Besides, a slower-ATE could be used.
- a µC interface, allowing an 8-bit microcontroller to read internal alarms and to select the modes of operation.
- the user can also insert the not normalized SOH bytes. This feature has been included only in the final designed IC (see below), and has been not included in the cost-effective UNI-IC (see section 7). The IC finally designed implements the whole TCS, because the UNICORN is an industrial project with different requirements: to obtain a powerful and flexible IC, to avoid the inclusion of TCS functions in the ATM device, etc.

7: Experimental results

In the analysis and verification process we used the VantageSpreadsheet tool , and the synthesis process was accomplished using the Compass ASIC-Synthesizer tool. In order to realize a STM-N ICs cost comparison (in terms of 4-transistor equivalent gates) the VHDL STM-N framer description was synthesized at three different levels: N = 1, 4 and 16. The ICs were mapped to the VLSI Technology, Inc. vsc350 (CMOS 1µm standard-cells) library. Thanks to a careful design of the serial blocks (working at the line rate) the synthesized STM-1 framer works up to 196 Mbit/s. Obviously, the STM-4 and STM-16 synthesis have just a cost-evaluation importance. In the table II we can see that the STM-N framer cost is not proportional to N (STM-1-cost x N), allowing a reasonable and feasible design. Finally, it is shown (see table III) the cost of the TCS blocks not included (as discussed in section 5). The transmitter and receiver FIFO memories do not appear in the table, because these are generated with a macro-cell compiler as a compact layout-block.

Synthesized framer IC	*# equiv-gates*
STM-1	6256
STM-4	14098
STM-16	36264

Table II

Block description	*# equiv-gates*
HEC generation	380
Syndrome verif. and header correction	3187
ATM scrambler	591
ATM descrambler	492

Table III

8: Conclusion

A VHDL description allowing to synthesize a set of different STM-N UNI ICs (N=1, 4,..) in a cost-effective manner has been developed. The project has two main goals: the development of an optimum and viable STM-N framer architecture selection, and the utilization of VHDL to the greatest advantage.

Two important VHDL features, the technology and process independence and its wide range of descriptive capability, make VHDL a very appropriate and sufficient stepping-stone to develop this project.

References

[1] R.Händel, M.N.Huber, "Integrated Broadband Networks". Engineering Communications. 1991 Addison-Wesley.
[2] R. Ballart, Y.C. Ching, "SONET: Now it's the Standard Optical Network" IEEE Commun. Mag. Mar.1989
[3] N.K.Cheung. "The Infrastructure for Gigabit Computers Networks".IEEE Commun. Mag. April 1992.
[4] CCITT Recommendation G.708. Ginebra 1991.
[5] CCITT Recommendation G.707. Ginebra 1991.
[6] CCITT Recommendation G.709. Ginebra 1991.
[7] CCITT Recommendation I.432.. Ginebra 1991.
[8] T.J.Robe, K.A.Walsh, "A SONET User Network Interface Integrated Circuit". IEEE Journal on Selected areas in Commun., Vol.9, No.5, Jun-1991
[9] D.T.Kong, "2.488 Gb/s SONET Multiplexer/ Demultiplexer with Frame Detection Capability". IEEE Journal on Selected Areas in Comm. Vol.9, No.5,Jun-1991
[10] T.Coe, R.Gates, W.Larkins, R.Deming, N.Hendrickson, I.Deyhimy, "1.5 Mb/s data rate communications gate array in GaAs", IEEE 1988 Custom Integrat.Cir.Conf., 1988
[11] R.Lipsett, C.Schaefer, C.Ussery, "VHDL: Hardware Description and Design", Kluwer Academic Publishers,1990.
[12] R.Airiau, J.M.Bergé, V.Olivé, J.Rouillard, "VHDL du langage à la Modélisation", Collection Informatique. Presses Polytechniques et Universitaries Romandes, 1990
[13] J.L.Massey, "Cryptography: Fundamentals and Applications", Adv. Technology Seminars, Zurich, 1990
[14] Olaf Geisler. Siemens AG,"VHDL based synthesis of a high speed full parallel 32 bit CRC generator/checker", VHDL-Forum for CAD in Europe. Spring'93 Meeting. March 14-17, 1993.
[15] VLSI Technology, Inc.,"VP15311 SLTX SONET line terminating transceiver designer's Guide", 1992
[16] F.Castaño,G.Salamanca, Alcatel-SESA, "TRX155 y MDVC3: dos circuitos integrados para SDH", VII Congreso de Diseño de Circuitos Integ., Toledo, Nov.1992
[17] H.T.Weston et al., "A submicrometer NMOS multiplexer-demultiplexer chip set for 622.08 Mbit/s SONET applications", IEEE Journal on Solid-State Circuits, Jul.1992.