ATM Traffic Shaper: ATS

Juan Carlos Diaz, Pierre Plaza, Jesús Crespo,
Telefónica Investigación y Desarrollo, Madrid (SPAIN)

Abstract

The design and Implementation of an ATM Traffic Shaper (ATS) is here described. This IC was realised on a 0.35µ CMOS technology. The main function of the ATS is the collection of low bit rate traffics to fill a higher bit rate pipe in order to reduce the cost of ATM based services, nowadays mainly influenced by transmission cost. The circuit fits in several ATM system configurations but mainly will be used at the User-Network Interfaces or Network-Network interfaces. The IC was designed with a Top-Down methodology using as HDL, Verilog.

The Chip is pad limited and is encapsulated on a 208 PQFP Package. The circuit complexity is 38 Kgates and its working frequency is 32Mhz. A circuit prototype was build with FPGAs in order to validate the RTL description.

1: Introduction

The ATS ASIC has been designed in the frame of a European JESSI project (AE-103: PNAP). The objective of the latter is to design an ATM [1] Public Network Access Point, which might implement the functions [2]:

- ATM layer functions [3], such as cell label identification and translation.
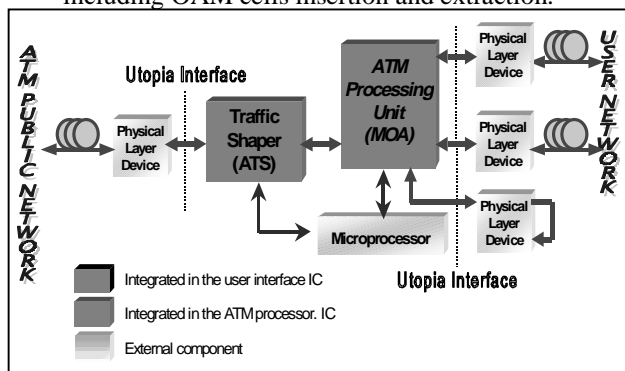- OAM (Operation and Maintenance) processing [4], including OAM cells insertion and extraction.

Figure 1: PNAP Architecture

- Traffic contract verification [5], policing (the incoming flow is managed in accordance to the pre-programmed parameters), and discarding or tagging the non-conforming cells.
- Traffic Shaping, avoiding cell losses in the public network nodes. It will also allow, for non-delay sensitive traffic, to obtain a better utilisation of the network resources.

As a result of the above function specifications it was decided that two ASICs were needed: The MOA (ATM Layer Processor) [6] and the ATS devices.

The Architecture of the PNAP system is shown in Figure 1. Basically the PNAP will interconnect several low bandwidth users to a common high bandwidth channel connected to the Public Network. The PNAP circuit manages two layer of the ATM transport network: ATM layer and Physical layer.

2: Traffic shaping

The traffic shaping function is very important for the new broadband services that are being deployed in order to avoid information loss, to provide to the end user several traffic contract options in terms of bandwidth and to insure an optimal use of the communication channels.

Figure 2: Traffic Shaping
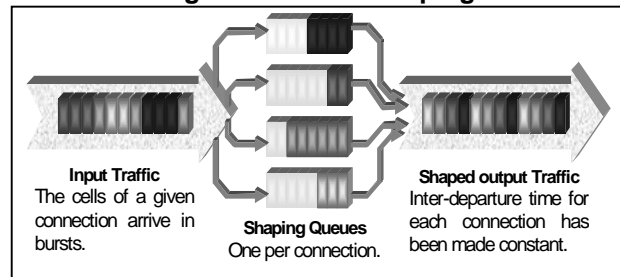
Traffic shaping is a mechanism that alters the traffic characteristics of a stream of cells on a connection to achieve better network efficiency whilst meeting the Quality of Service objectives, or to ensure conformance at a subsequent interface. Traffic shaping must maintain cell sequence integrity on a connection [7]. The ATS circuit is

a sort of queue administrator: it stores each cell in a different queue according to its connection identifier. The time interval between two consecutive cells from the same queue can be programmed independently for every active connection, and the resulting output traffic rate is constant and equal to the programmed one (see Figure. 2).
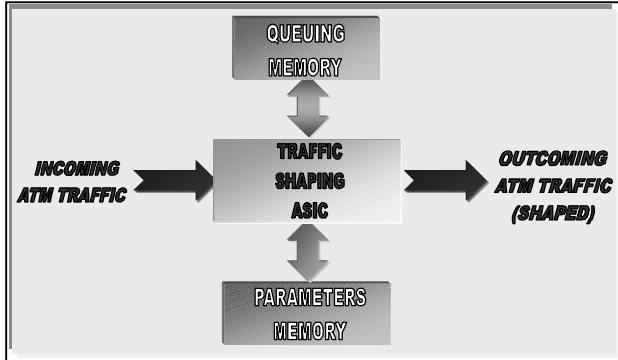

Figure 3: Traffic shaper functional structure.

The circuit can handle up to 256 connections simultaneously. Therefore a great amount of memory is required to build the associated queues. In fact, two external memories are needed, connected the way that is shown in Figure 3. In the queuing memory the ATM cells are stored corresponding to each possible connection. In the Parameters memory the queues configuration and associated parameters are stored.

3: Shaping algorithm description

The chosen shaping mechanism is based on a memory-less algorithm: the outcoming bit rate for each connection (queue) is calculated in terms of the speed that can be handled by the link (network connection).

$$V_c = V_l \cdot \frac{N}{2^k} \ \text{ with } N \in [0,2^k] \text{ and } V_c \in \left[V_l, \frac{V_l}{2^k}\right]$$

Where V_c in the connection speed, V_l the link speed, N is the parameter used to specify the connection speed.

To control the outcoming traffic, the Inter Departure Time (*IDT*) is fixed between consecutive cells:

$$Inter\ Departure\ Time\ (IDT) = \frac{2}{N \cdot V_l}$$

Once a cell of a given connection comes out, the algorithm calculates the next time its queue will be served again, $2^k/N$ cell slots after. Two data structures are required:

- A table containing the parameters of each connection (Parameters Memory); the main parameter is the outcoming bit rate (*N*).
- A schedule of pending connection deliveries, storing connection identifiers. The pointer that determines which connection identifier is processed

(the corresponding connection cell is delivered to the output) is a free running counter. Each counter value (queue position) represents a cell time slot. The maximum value of the counter is $2^k/V_l$.

The above solution has carried out the following limitations:

- The maximum *IDT* is fixed to the maximum value of the time counter. In other words, there is a minimum value for the outcoming bit rate. This limitation is negligible assuming the number of bits of the free-running counter is high.
- As the time units are finite, the outcoming bit rate can have only a finite number of values. This value granularity has been increased using two parameters: quotient and remainder of *IDT*.
- Two different connections may need the same cell slot; if this situation happens, the shaping algorithm controller looks for the next free time slot in the queue. This solution impacts the accuracy of the shaping algorithm when the link load is high.

A third module performing an Available Bit Rate control (ABR) [8] algorithm is planned for a future development. It will be placed next to the ATS to obtain a more flexible system: this circuit should extract and process the Resource Management cells (RM) [7], and modify the *IDT* values programmed, according to the network status.

4: System characterisation by simulation

This section presents a set of simulations performed to set the parameters of the shaper circuit:

- The number of traffic sources (connections) supported.
- The memory size, taking into account the system must keep the ATM quality standards (Cell Loss Probability, CLP, less than 10^9).

In order to fix the two parameters, a high level model of the system has been written using a hardware description language (Verilog), and some simulations have been performed using a realistic worst case for the data sources [9]:

- Traffic sources modelled as independent "on/off" generators, with a burstiness factor of 6 (the "off" mean time is five times greater than the "on" mean time). The "on" and "off" times fit in exponential distributions.
- 80% of load in the link, obtained as aggregate of traffic sources.

The outcoming Inter Departure Time (*IDT*) for each connection has been set in a range between 60% and 80% of the incoming *IDT*; this value is in fact the throughput of each individual queue.
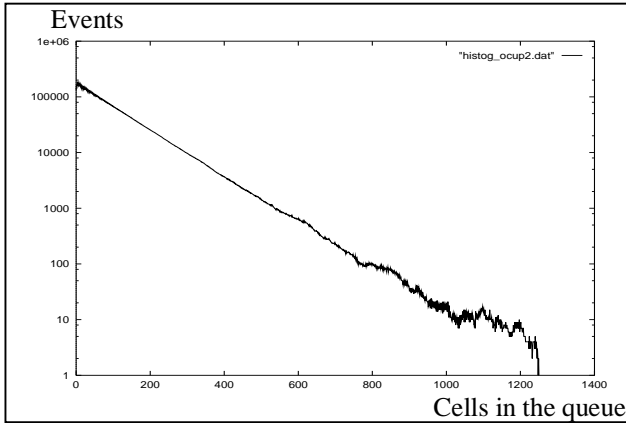
Figure 4: Histogram of cells in queue

The simulations have been performed with different number of simultaneous traffic sources (1024,512,256); the resulting queue histograms show that the number of cells in each queue follow an exponential distribution which mean value depends on the queue throughput, as shown in Figure 4.
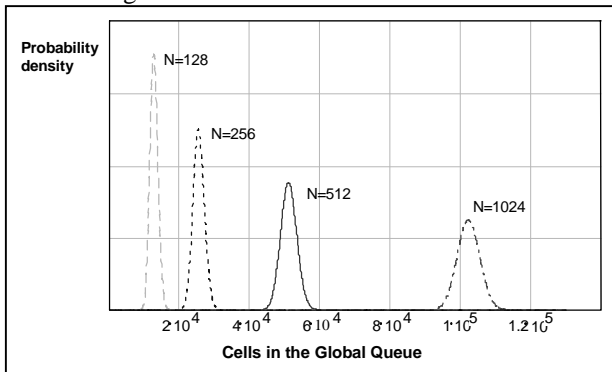

Figure 5: Gamma distribution for a different number of connections

The behaviour of the whole queue (as aggregate of 1024, 512 or 256 individual ones) can be calculated analytically using the random variable properties: the sum of independent exponential random variables is a Gamma distribution which mean value is the sum of the mean values. Figure 5 shows the Gamma distribution calculated for 1024, 512 and 256 exponential distributions with a mean value of 100 cells. All the analytical results and figures have been obtained using MathCad.

Figure 6 shows the number of cells required in the global queue to insure a Cell Loss Probability (CLP) of 10^{-9} (the quality requirement), calculated over the Gamma distribution associated with 256 connections for a range of individual queue mean values.

And finally, Figure 7 presents a set of curves that represent the number of cells to be stored in the queues required to insure a given CLP against the number of connection supported. Each curve is associated with a individual queue mean value M.
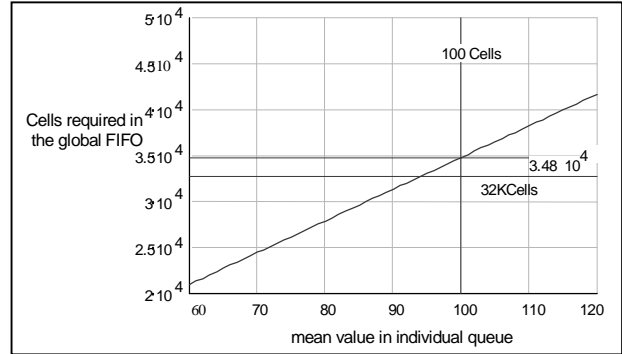

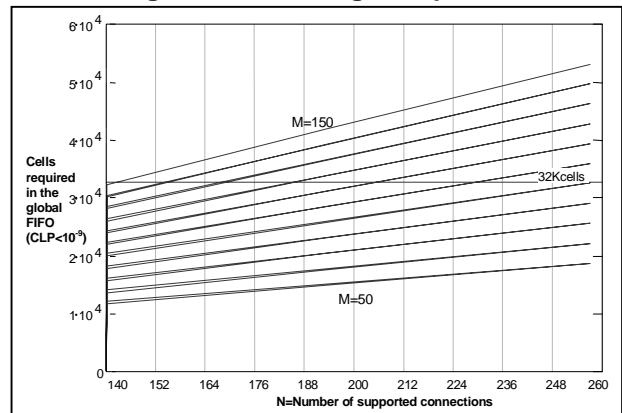**Figure 6: FIFO length required**

Figure 7: FIFO length versus number of connection supported.

After a deep study of the above simulations and analytical calculations, the number of connections (queues) supported by the ATS was fixed to 256, big enough for the premises of the PNAP project, and the external memory would have to store at least 40000 cells. Unfortunately commercial memories in a single IC could not hold so much information. Therefore a memory capable of storing 32 Kcells was chosen, assuming the CLP did not increase very much since the source models for the system simulations were too pessimistic: in the simulations performed with the prototype the CLP obtained with this amount of memory was less than 10^{-13}.

5: The ATS architecture

Figure 8 shows the high level architecture of the ATS circuit. The internal blocks are briefly described in this section.

5.1: Input module

The input module receives the cells in a UTOPIA level 2 standard format at 155 Mb/s [10] (8 bits in parallel at a frequency of 19.44 MHz). The connection identifier is stored in the HEC field of the ATM cell header [6], because this field is useless at the ATM layer.
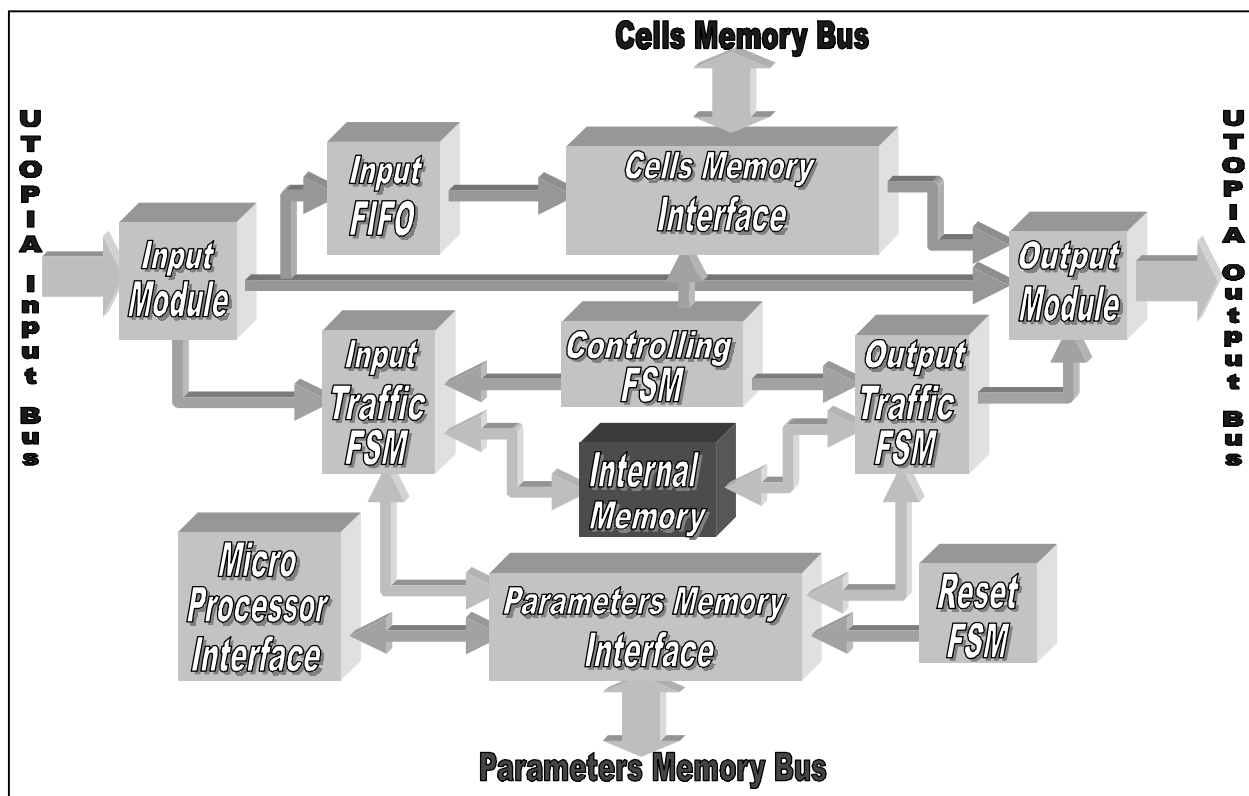
Figure 8: ATS Architecture

This module handles the UTOPIA level 2 protocol, extracts the connection identifier and converts the input data from 8 to 32 bits in parallel. It also separates the cells of real time connections; which are transmitted to the output module without queuing, in order to avoid any transfer delay.

5.2: Input FIFO

The cells are temporarily stored in a FIFO, in order to synchronise them with the internal cell synchronism signal. The cells belonging to real time connections are not stored in the input FIFO.

5.3: Cells memory interface

The queuing system is implemented in an external dynamic SGRAM, which allows burst accesses. Each cell time slot (53 input clock cycles) a cell coming from the input FIFO is written, and another cell is read and delivered to the output module. The cells memory interface controls these accesses.

5.4: Parameters memory interface

The ATS algorithm needs the following segments:

- **Linked list control segment**. It contains a pointer (address) to the next cell in the queue for each cell slot in the cells memory.

- **Pending services segment**. It stores the connection identifier of the queue to be served in each cell time slot.
- **Occupied cell time slots segment**. It has one bit per each cell time slot indicating if the cell slot is free or assigned to serve the connection recorded in the pending services segment.
- **Queues descriptors segment**. This segment contains the parameters associated to the individual queues control such as:
  - Pointer to the first cell in the queue.
  - Pointer to the last cell in the queue.
  - Number of cells in the queue.
  - Maximum number of cells in the queue (programmable).
  - Shaping parameters (programmable).

The parameters memory interface takes care of the access to the parameters memory. There are several modules that use this memory:

- Input traffic finite-state machine.
- Output traffic finite-state machine.
- Reset finite-state machine.
- Microprocessor interface.

5.5: Microprocessor interface

It is an asynchronous parallel interface to a Motorola 68000 family microprocessor. Via the microprocessor interface module the circuit can be initialised and

programmed. The queuing parameters can be set by writing internal registers.

5.6: Internal memory

The simulations have shown that the occupancy level of the pending services segment around the current cell time slot is very high, and too many accesses to the occupied cell time slots segment had to be performed to find a free one. To reduce this number of accesses, the shaper circuit has an internal memory that holds information about the occupancy the latter segment.

5.7: Input traffic finite-state machine

This module processes the incoming cells, and administrates the queues system (pointers, number of cells, etc.).

5.8: Output traffic finite-state machine

It controls the outcoming cell flow. It schedules the time when a connection has to be served (according to the pre-programmed shaping parameters) and updates the queue structure and parameters.

5.9: Reset finite-state machine

It initialises the linked list structure at power on.

5.10: Controlling finite-state machine

This is the central control module of the circuit. It contains the time counter and generates the control signals for the rest of modules. It is also responsible for the internal synchronisation.
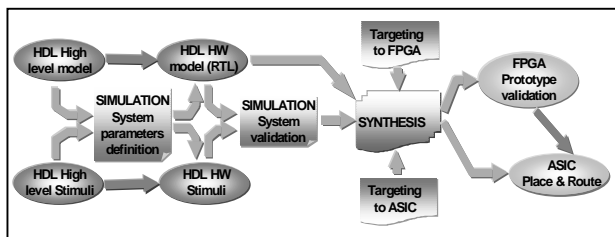
6: Design methodology and tools

Figure 9: ATS Design Methodology

6.1: Top level design

The ATS was designed following a top-down methodology (see Figure 9)[11]. The Behaviour of the system has been modelled and simulated using Verilog HDL. Very complex Verilog test benches were written and used for the simulations in every design step down to the post-layout phase (Cadence- Cell3). The IC was validated by an FPGA prototype. Automatic synthesis (Synopsys Design Compiler) has been applied to produce a standard cell netlist (HCMOS6 from SGS Thomson).
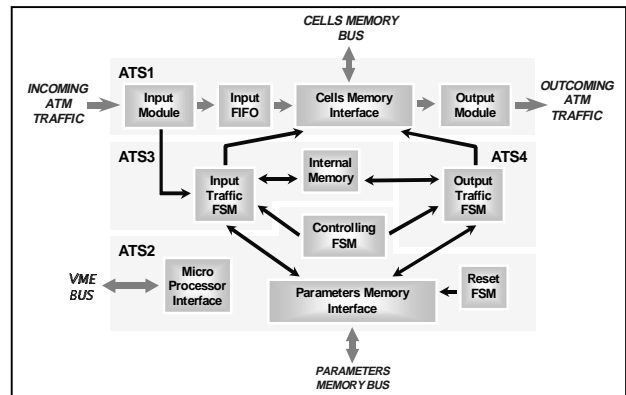
6.2: FPGA prototype

Figure 10: ATS mapping to 4 FPGA's.

The HDL model of the ATS circuit was targeted to the XILINX 4000E family and synthesised using Cadence Synergy. The system had to be divided in four blocks as shown in Figure 10, and the statistics of the resulting FPGA devices are displayed in Table 1.

Device	Part	CLB	Used	Pins	Used	Package
ATS1	4008E	324	312(96%)	144	132	208PQFP
ATS2	4006E	256	231(90%)	128	121	208PQFP
ATS3	4010E	400	369(92%)	160	152	208PQFP
ATS4	4010E	400	389(97%)	160	144	208PQFP

Table 1: FPGA devices characteristics

This prototype has been successfully employed and tested in a real ATM equipment developed in Telefónica I+D (ATMA, ATM Adapter). The complete system is shown in Figure 11.
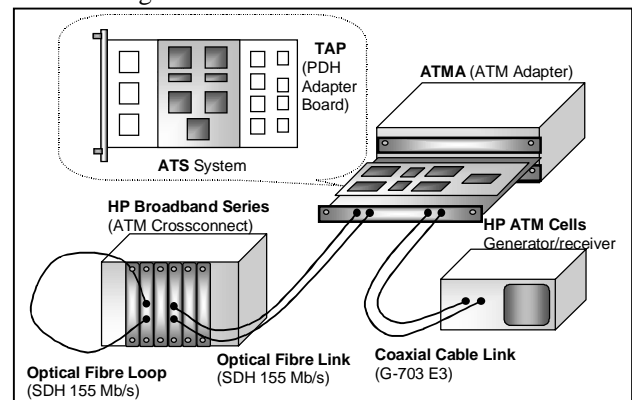

Figure 11: ATMA development system

6.3: Place and route

The layout was carried out in a standard cell fashion with a semi-automatic tool environment: Unicad Cell3 kit from SGS-Thomson based on the Cadence tools. A Clock tree synthesis approach was followed in order to minimise the clock skew. To do that, shell scripts were available that could automatically construct a clock buffer tree, equalising perfectly all different clock branches and sub-branches. To meet all the timing constraints, in-place optimisation was used using SDF back-annotation into Synopsys Design Compiler.

6.4: Test strategy

A Full Scan strategy was applied: eight different scan chains were obtained. A proprietary test tool was used, PLATON [12], to insert the scan flip-flops, interconnect the scan chains taking into consideration the placement of the flip-flops and generate the test vectors. The fault coverage obtained with the latter is around 96%. To test the memories, a dedicated BIST strategy was conceived: A set of LFSRs and MISRs blocks were designed and validated. Input patterns are fed to the memories via specially provided multiplexers at the I/O of the circuit.

7: Results and conclusions

The ATS layout and main statistics of the realised IC are reported below.
- Overall complexity: 38 Kgates.
- Clock frequency: 32 MHz
- Technology: 0.35 Micron CMOS, 5 level metal
- Test Fault Coverage: 96%
- Package: 208 QFP
- Power Consumption: 341 mA -> 1.125 Watts

The implementation of a CMOS 0.35 ATM Traffic Shaper was outlined on this paper. Through the use of FPGA prototyping the behaviour of the IC and the RTL HDL were validated. In fact the HDL had to be targeted and modified in order to get the best results for the FPGA synthesis and for the IC synthesis. The UNICAD Unix-Cell3 SGS-Thomson design kit was used throughout the design phase. To reduce the clock skew between flip-flops an automatic set of tools was used to generate an equalised buffer tree for the system clock. Full scan was applied to test the dies in combination with customised BIST structures, insuring good fault coverage. The IC is pad limited. In place optimisation was used with SYNOPSYS in order to meet the timing constraints after layout.
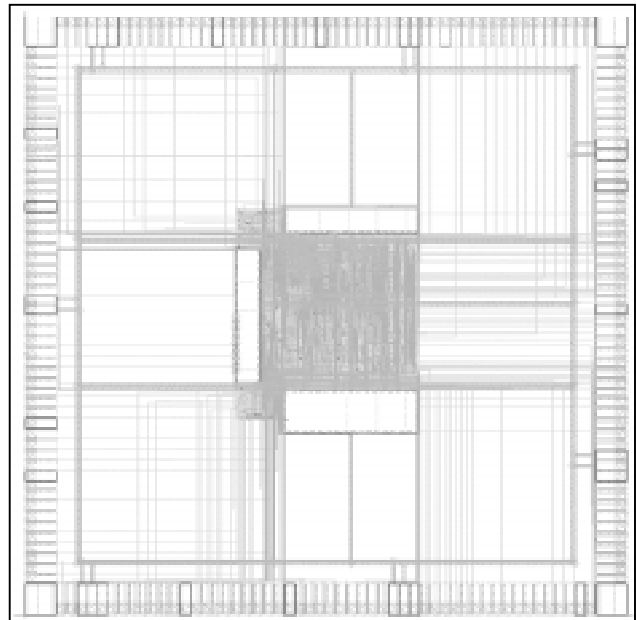

Figure 12: ATS Layout

8: References

[1] ITU Telecommunications Standardization Sector. Study Group 13. "Recommendation I.311, B-ISDN General Network Aspects". July 1995.

[2] Telefónica I+D, SGS-Thomson, Italtel. AE-102: "PNAP" Blue Book Abstract. JESSI Proposal, January 1995.

[3] ITU Telecommunications Standardization Sector. Study Group 13. "Recommendation I.361, B-ISDN ATM Layer Specification". July 1995.

[4] ITU Telecommunications Standardization Sector. Study Group 13. "Recommendation I.610, B-ISDN Operation and Maintenance Principles and Functions". November 1994.

[5] ITU Telecommunication Standardization Sector. Study Group 12. "Recommendation I.371, Traffic and Congestion Control in B-ISDN". July 1995.

[6] A. Daniele. Multicast OAM ATM IC specification. PNAP JESSI project deliverable, September 1996.

[7] The ATM Forum Technical Committee. "Traffic Management Specification". December 1995.

[8] B.G. Kim and P. Wang. ATM ABR Traffic Control Functions. Intern. Conf. On system Engineering '96.

[9] A.M. Law, W.D. Kelton. "Simulation Modeling and Analysys". McGraw-Hill, 1982.

[10] The ATM Forum Technical Committee. "UTOPIA, An ATM-PHY Interface Specification. Level 2". Version 0.95, June 1995.

[11] J. C. Diaz, P.Plaza, L Merayo et al. Design and Validation with HDL Verilog of a complex I/O Processor for an ATM Switch: The CMC. IVC'95. California, 1994.

[12] M.J. Aguado et al. Automatic Test Pattern Generation and Ensemble for very complex ASICs. DCIS'92. Toledo-Spain-1992.