

# Investigation into Programmability for Layer 2 Protocol Frame Delineation Architectures

Ciaran Toal, Sakir Sezer  
Institute of Communications and Information Technology,  
Queen's University Belfast,  
Queen's Road, Belfast, Northern Ireland  
Ciaran.Toal@ee.qub.ac.uk

## Abstract

*This paper presents the design and study of reconfigurable architectures for two data-link layer frame delineation techniques used for ATM and GFP. The architectures are targeted to Altera Stratix II FPGA technology and are investigated in terms of performance and area. This work addresses the potential for incorporating programmability into custom purpose architectures that could enable the same processing hardware to be used for processing multiple protocols.*

## 1. Introduction

In communication networks, the physical layer is responsible for the transmission of raw bit streams between a source and a destination. Framing is an essential process of data transmission and it is important for any data-link layer protocol to provide a mechanism for packet boundary recognition. Network-layer packets such as the Internet Protocol (IP) typically do not have a mechanism in place that determines the start and end of packets within streamed data.

Frame delineation is a key function of the framing process of data-link layer protocols, such as Ethernet, PPP, GFP, HDLC, SDLC and ATM. A number of frame delineation mechanisms have been adopted by the standard. Some of these protocols utilise mechanisms that are based on unique bit patterns, such as the PPP flag "0111110", which indicates the start and/or end of each frame [1]. ATM and the recently emerged link layer protocol, the Generic Frame Procedure (GFP), use cyclic coding for Header Error

Check (HEC) and frame delineation. Cyclic code based frame delineation requires a complex Cyclic Redundancy Check (CRC) computation circuit for error and frame boundary detection. The advantage of this technique is that the frame payload does not need to be modified before transmission and after reception unlike HDLC or PPP which must "escape" their frame delineation pattern to prevent valid payload data from being mistaken as a frame boundary indicator.

Frame delineation mechanisms for ATM and GFP are investigated. For each of these protocols, optimised 32-bit frame delineation circuit architectures are designed and their performance analysed. The designs of the different protocol frame delineation circuits are broken down to their fundamental processing blocks and are cross correlated to examine and understand possible programmability that could be implemented into one circuit with the target of developing a multi-protocol frame delineation architecture.

In this paper, section 2 presents the ATM frame delineation architecture. A byte-by-byte parallel HEC hunt circuit is designed for ATM over SONET/SDH physical layer transmission. GFP is examined and presented in section 3. Section 4 takes a step back and examines the low level functions that make up each of the 2 investigated frame delineation circuits in order to establish the feasibility of deriving a programmable frame delineation architecture for both protocols. Section 5 presents the design and implementation of two programmable architectures using Altera Stratix II FPGA technology and analyses the synthesis results in terms of programmability, throughput performance and hardware cost.

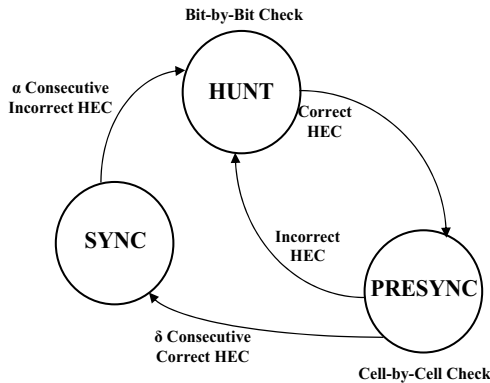
## 2. ATM Frame Delineation

The ATM Frame delimiter is based on HEC (header error check) cyclic coding [10], [11]. ATM Cell delineation is specified by the ITU-T in recommendation I.432 [5].

The ATM cell consists of 5 header bytes. The first four bytes contain information related solely for routing the protocol. The 5<sup>th</sup> byte contains the HEC field which is calculated from the first 4 bytes of the header.

When an ATM cell is received, the HEC value is again calculated from the first 4 header bytes and compared with the fifth byte. In the absence of errors, both values are identical and the cell boundary is assumed to be located.

The HEC field is calculated as a remainder of the modulo-2 division of the first 4 header bytes with the CRC generator polynomial  $G(x) = 1+x+x^2+x^8$ .



**Figure 1. ATM Cell Delineation State Diagram.**

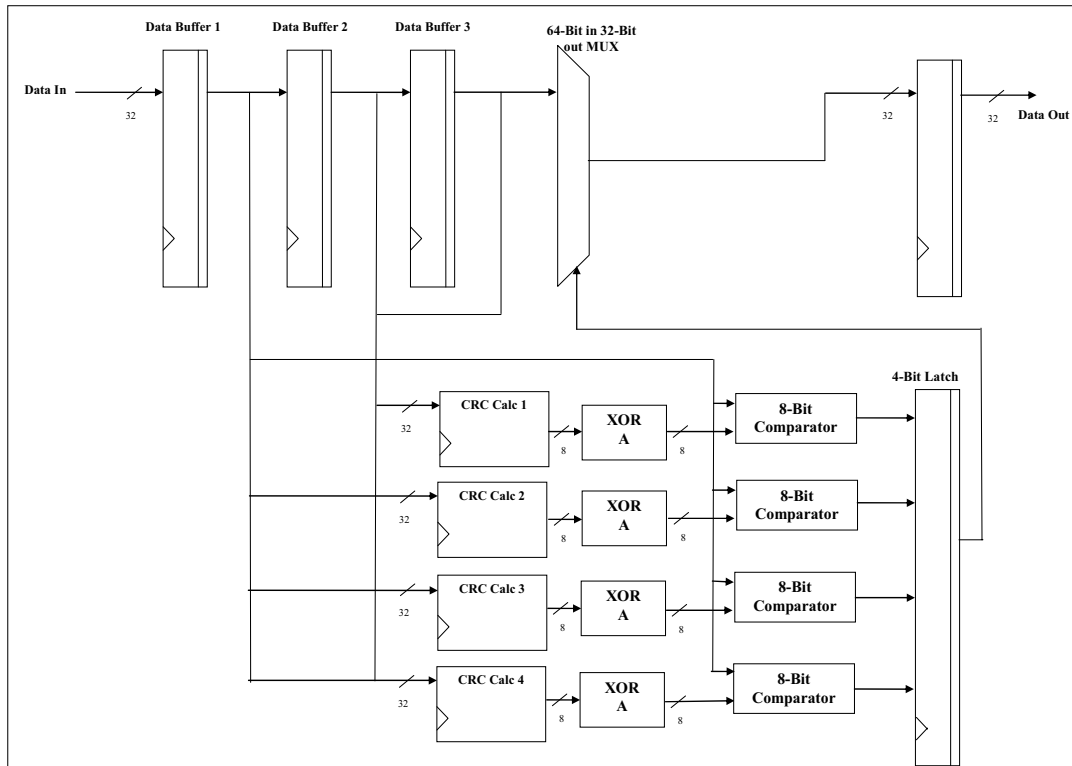
In the ITU-T recommendation I.432, for the SDH-based physical layer, values of  $\alpha = 7$  and  $\delta = 6$  are suggested. For the cell-based physical layer, values of  $\alpha = 7$  and  $\delta = 8$  are suggested.

ATM cell synchronisation is a sequential process in accordance with the state graph in figure 1. The receiver initially operates in the HUNT state and

assumes no knowledge of the next incoming frame boundary. Incoming data is streamed through the CRC computation circuit. Once 4 bytes have been processed by the CRC circuit, the receiver checks if the computed 8-bit CRC value is equal to the next incoming 8 bits i.e. the HEC field in the frame header. If there is a match the system enters the PRESYNC state, otherwise it continues checking incoming data. The comparison of the computed CRC value with a possible HEC field must be carried out for each byte entering the computation circuit. If a correct HEC pattern is detected, the synchronisation state machine moves to the PRESYNC state and checks subsequent cells for matching HEC fields. If it receives  $\delta$  consecutive correct HEC fields it enters the SYNC state. During the PRESYNCH phase, the synchronisation circuit will return back to HUNT state if a single incorrect HEC is found. Once in the SYNC state, the system can only return to HUNT if  $\alpha$  consecutive incorrect HEC fields are received.

The implementation of the bit-serial and parallel ATM HEC check architectures have been presented by G.E. Griffith et al [3], Suh Chung-Wook et al [9], Ng. Leong Seong et al [6] and A. Maniatopoulos. Chung-Wook's investigation is based on a HEC check implementation for a 16-bit data path targeting at a throughput rate of 622 Mbps for ATM over SONET. Leong Seong's investigation explores an 8, 16 and 32-bit CRC computation architecture for the ATM HEC hunt. Both investigations emphasise mainly the CRC computation of the HEC hunt circuit and targets a solution only for octet based cell transmission (SONET/SDH).

The 32-Bit ATM HEC hunt circuit is shown in Figure 2. It is a pipelined architecture and consists of 4 32-bit in/ 8-bit out CRC calculators and 4 8-bit comparators. The circuit basically requires one CRC calculator and one comparator for each incoming byte.



**Figure 2. 32-Bit ATM Receiver Frame Delineation Architecture.**

### 3. GFP Frame Delineation

GFP Frame delineation is specified by the ITU-T in recommendation G.7041 [7], [4]. GFP deploys a HEC based frame delimiter mechanism in a similar manner as ATM [8], [2], [12], [13]. The 32-bit GFP frame delineation circuit utilises 4 CRC HEC calculators and 4 16-bit comparators to accommodate the wide datapath, a Payload Length Indicator (PLI) frame counter, frame synchronisation state machine and a single bit error correction mechanism.

cHEC (Core Header Error Check) field is calculated from the first 2 bytes of the core header i.e. the Payload Length Indicator. The calculated cHEC field is used for frame delimitation/synchronisation and is located at the third and fourth byte positions of the GFP core header. When a GFP frame is received the cHEC is again calculated from the first 2 core header bytes and compared with the third and fourth bytes. In the absence of errors, both values are identical and the frame boundary is assumed to be located. The cHEC field is calculated as a remainder of the modulo-2 division of the PLI field with the CRC generator polynomial  $G(x) = 1+x^5+x^{12}+x^{16}$ . One major difference between the GFP and ATM specifications is that GFP always hunts data byte-by-byte.

GFP frame synchronisation state graph is similar to that of ATM. The receiver initially operates in the HUNT state and assumes no knowledge of the next incoming frame boundary. The received data is streamed through the CRC computation circuit. Once 2 bytes have been processed by the CRC circuit, the receiver checks if the computed 16-bit CRC value is equal to the next incoming 16 bits i.e. the cHEC field in the frame core header. If there is a match the system enters the PRESYNC state, otherwise it continues checking incoming data byte-by-byte.

The 32-bit architecture is shown in figure 3. The design requires 4 16-bit In/16-bit Out CRC units and 4 16-bit comparator units. Every clock cycle, 4 new bytes of data are scanned in. The circuit is designed to locate a possible cHEC on all 4 input byte locations. The first positive match between the PLI field CRC remainder and the subsequent transmitted CRC field found by a comparator unit (i.e. a located cHEC) is latched. This latched signal controls what is essentially a 4-byte window gate enabling 4 consecutive bytes of a possible 7 to be routed through to the output.

The deployed error correction technique is a ROM based RS lookup table implementation. Due to the small number of entries, ROM based logic synthesis on FPGA presents a more efficient solution than a RAM

based implementation, overcoming memory addressing issues and resulting in a reasonably small circuit. The key advantage of synthesizing a ROM table is the portability to other technologies in form of a technology

independent IP core. The error correction circuit is able to correct any single bit error in one clock cycle.

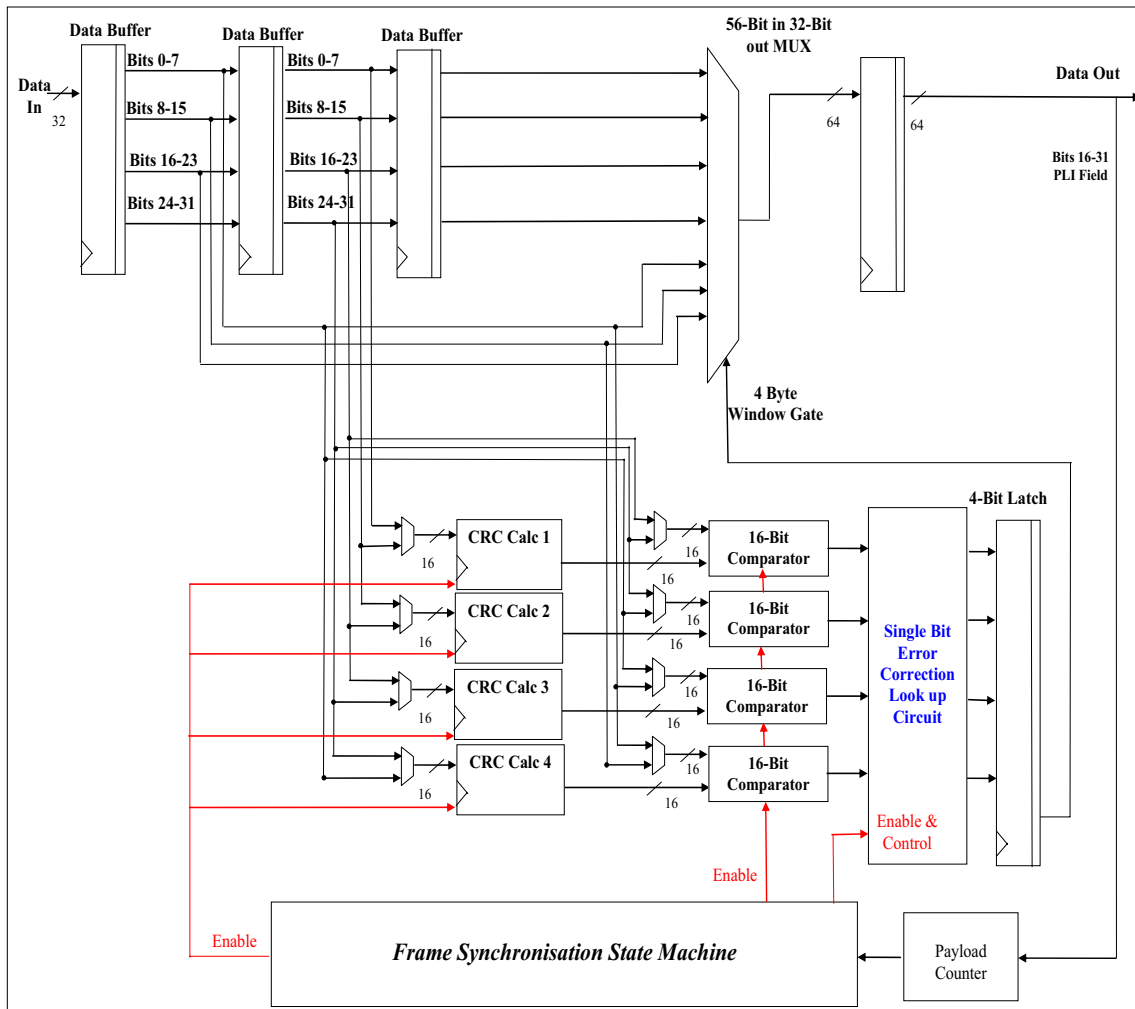


Figure 3. 32-Bit GFP Receiver Frame Delineation Architecture.

#### 4. Programmable Frame Delineation

The ATM and GFP frame delineation circuit implementations have been explored to determine the feasibility of deriving a single programmable frame delimiter architecture that can support both protocols with high data throughput rates.

The analysis suggests that there is no simple method of implementing both the frame delineation processes within the same programmable circuit. For example, despite the fact that ATM and GFP are based on the

same principle, using CRC HEC computation, their low level architectures are significantly different.

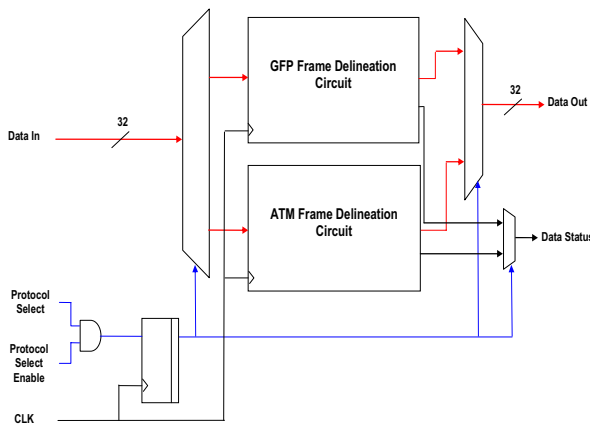
ATM is based on a CRC-8 calculation of a 32-bit data word, whereas GFP is based on a different CRC-16 polynomial division of a 16-bit data word, not to mention very different divisor polynomials. Both architectures are so different in nature that a programmable CRC computation circuit cannot be efficiently mapped onto the same hardware.

Therefore two techniques have been investigated as possible underlying technology to support

programmability of the target frame delineation architecture

- The first programmable architecture is based on the implementation of both circuits using the same hardware. In this case programmability is achieved by the selective multiplexing of the data-path between one of the two circuits.
- The second programmable architecture was based on the use of an embedded reconfigurable logic that can be configured to support a specific protocol.

Figure 4 shows the block diagram of the first and probably less elegant approach for achieving programmability. It is composed of two protocol specific (hardwired) frame delineation circuits as individual blocks with a programmable data-path that selects the required circuit for the programmed protocol.



**Figure 4. Programmable Dual-Protocol Frame Delineation Architecture.**

A high level diagram of the second circuit is shown in figure 5. It has been derived from the analysis of the low-level functional blocks of both protocol specific frame delineation architectures. The data buffers are utilised by both circuits. The comparators have been

designed so that they can be programmed to operate as 8-bit (for ATM) or as full 16-bit (for GFP). The XOR matrix structures are included as separate components for the GFP and ATM CRC calculation. This is because the XOR arrays are different. There is no advantage to be gained by attempting to reuse the XOR gates so that the one component can handle both CRC calculations.

ATM contains 4 8-bit in/32-bit out CRC engines whilst GFP contains 4 16-bit in/16-bit out CRC engines. Each output CRC bit is fabricated from different input bits. If the XOR arrays consisted of the same dimensions in terms of length and breadth then the argument could be made to implement one component with an optimised number of XOR gates along with matching multiplexers that would enable both the CRC calculations to be performed by the structure. Due to the very different XOR array structures and plus the fact that the ATM matrix is effectively 4 8\*8 XOR arrays staggered means that this option is unfeasible. The GFP error correction engine is obviously only common to the GFP and as such can only be accessed when the circuit is configured to process GFP packets.

The counter is synthesised by the synthesis tool. It is configured so that when processing GFP it reads in the 16-bit PLI value and decrements from this as bytes are received. When configured for ATM the counter always resets to 48 since this is always the size of the ATM cell. The protocol control state machine is effectively a RAM where each state is a memory address and the output of the state machine is the data stored at that memory address. The ATM and GFP state machines are stored in the same memory bank with the protocol select register effectively acting as a pointer that selects the section of memory that contains the micro-code for each protocol. The tri-state frame synchronisation, although contains the same three states for GFP and ATM i.e. HUNT, Pre-SYNC and SYNC, the  $\theta$  and  $\delta$  values are different which means that the behaviour of the two state machines is different as the output generated when in each state can be different thus meaning that the data stored in the memory address can be different.

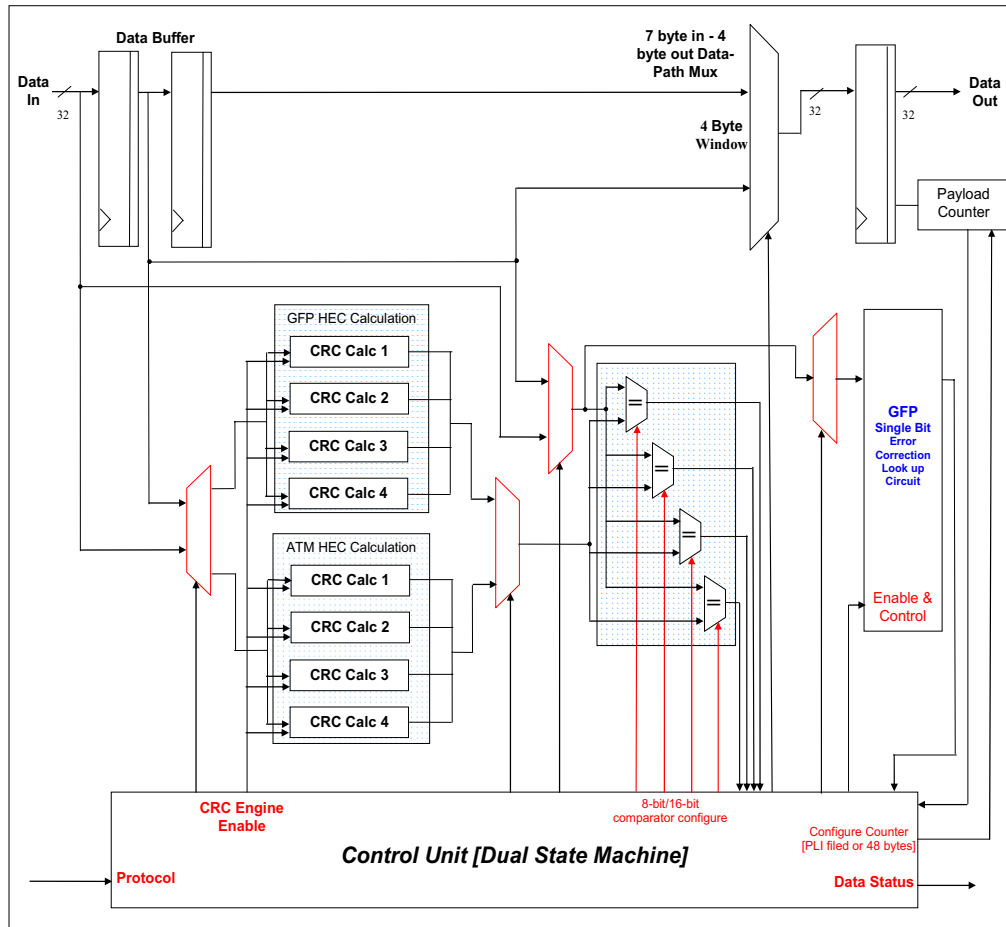


Figure 5. 32-Bit ATM/GFP Programmable Frame Delineation Architecture Incorporating Common/Programmable Elements.

## 5. Synthesis and Circuit Study

The two 32-bit frame delineation circuits have been synthesised and targeted to Altera Stratix II FPGA technology. The post-layout synthesis results are included in table 1. Speed and area performance is examined.

Table 1. 32-Bit ATM/GFP Frame delineation Circuits.

Protocol	Area			Speed	
	ALUTs	Registers	ALMs	Clock Frequency (MHz)	Data Throughput (Mbps)
GFP	547	351	361	171.89	5500.48
ATM	281	164	177	260.55	8337.6

The GFP frame delineation circuit is much slower than the ATM circuit. GFP is a much more complex

architecture than ATM. The design is impeded by the requirement of the memory correction look-up table, which not only penalises the area but also imposes a large area constraint on the circuit.

The Stratix II post-layout synthesis results for the two dual-protocol frame delineation circuits are presented in table 2.

Table 2. P<sup>5</sup> 32-bit Implementation.

Programmable ATM/GFP Frame Delimiter	Area				Speed	
	ALUTs	Registers	ALMs	LABs	Clock Frequency (MHz)	Data Throughput (Mbps)
Common Elements	885	387	530	78	165.65	5300.8
Separate Data-Path	872	531	621	124	159.46	5102.72

The two implementations have very similar performance, and in fact the circuit that utilises

common elements is surprisingly slightly faster of the two.

Not only it is faster but it is also smaller in terms of hardware cost. Although it contains 13 more ALUTs, it requires 144 less registers, which is approximately a 50% reduction of the register cost. The overall LAB (Logic Array Block) cost is therefore reduced for the FPGA technology. It is anticipated that the reduction of the register count will contribute more significantly to the overall hardware cost reduction in terms of silicon area for a cell based technology.

The register reduction is not unexpected considering the reused components such as the buffers, comparators and pipeline stages of both circuits.

This initial analysis has produced some positively conclusive evidence as to the design style that should be followed in designing a multi-protocol processor on an ASIC or structured ASIC design in order to obtain maximum performance.

## 5. Conclusions

The primary objective of the research described in this paper was to ascertain the feasibility of implementing architectures that could handle multiple protocol frame delineation functions. Two architectures were developed that were each programmable and able to perform 32-bit frame delineation for both GFP and ATM. The first architecture is composed of the originally designed GFP and ATM circuits with a common data-path included for input and output. The desired frame delineation function is selected via multiplexers. The second circuit is a much more complex design and is composed of common low-level function blocks such as buffering registers and comparators. Function blocks that could not serve both protocols efficiently were included separately. The GFP/ATM frame delimiter architecture that contained common logic resulted in a slightly faster and smaller circuit than the classical architecture based on multiplexing data-path between protocol specific circuits. The study has produced conclusive evidence that programmability can be achieved by designing a configurable data-path of common, configurable and domain specific function blocks.

## 8. References

[1] C. Toal and S Sezer, "A 32-Bit SoPC Implementation of a P<sup>5</sup>", IEEE Symposium on Computers and Communications", Antalya , Turkey, July 2003.

- [2] P. Bonenfant, A. Rodriguez-Moral, "Generic Framing Procedure (GFP): The Catalyst for Efficient Data over Transport", IEEE Communications Magazine, May 2002.
- [3] G.E. Griffith, T. Arslan and A. T. Erdogan, "Asynchronous Transfer Mode Cell Delineator Implementations" IEEE SoC Conference", September 2003.
- [4] ITU-T Recommendation G.7041/Y.1303, "Generic Framing Procedure (GFP)", December 2003.
- [5] ITU-TS Recommendation. I.432 "B-ISDN user-network interface - Physical layer specification", June 1992.
- [6] L.S. Ng and Bill Dewar, "Parallel realization of the ATM cell header CRC" Computer Communications", 1996.
- [7] H. Qureshi, S. Ferguson, C. Scotland, "Generic Framing Procedure ITU-T G.7041 White Paper", Electronic Products Solutions Group, Telecomms Networks Test Division, Scotland, Agilent Technologies, July 2002.
- [8] M. Scholten, Z. Zhu, Enrique Hernandez-Valencia, John Hawkins, "Data Transport Applications Using GFP", IEEE Communications Magazine," May 2002.
- [9] C. W. Suh and K. S. Kim, "High-speed HEC algorithm for ATM", 1<sup>st</sup> International Conference on Information, Communications and Signal Processing", 1997.
- [10] C. Toal and S. Sezer, "The Implementation of a Scalable ATM Frame Delineation Circuits", IEEE International Conference on Telecommunications", August 2004.
- [11] C. Toal, S. Sezer, "A 10Gbps HEC HUNT Circuit for ATM over SDH/SONET", The IEE Irish Signals and Systems Conference", June 2004.
- [12] C. Toal, S. Sezer, "Exploration of GFP Frame Delineation Architectures for Network processing", IEE SoC Conference", September 2004.
- [13] C. Toal, S. Sezer, "A 10 Gbps GFP Frame Delineation Circuit with Single Bit Error Correction on an FPGA", IEEE Advanced Industrial Conference on Telecommunications", July 2005.