

# Asynchronous Transfer Mode Cell Delineator Implementations

G.E. Griffith<sup>1</sup>, T. Arslan<sup>1,2</sup>, A. T. Erdogan<sup>2</sup>

<sup>1</sup>The Institute for System Level Integration, Livingston, Scotland, UK.

<sup>2</sup>Department of Electronics & Electrical Engineering, The University of Edinburgh, Scotland, UK.

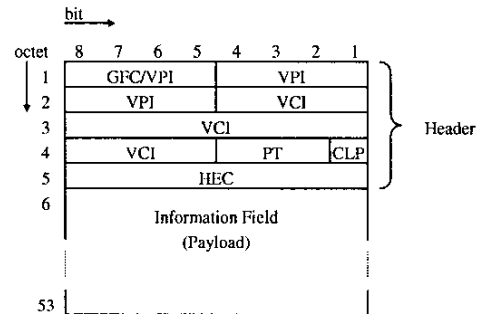
**Abstract** - Several algorithms for performing cell delineation on a bit-serial and octet-parallel basis are considered. Four algorithms were implemented in total, two bit-serial and two octet-parallel, for the design of cell delineators through to netlist generation. Analysis is provided for post-synthesis designs detailing speed, area and power parameters for each implementation at an input data rate of 160Mbps and 1280Mbps.

## 1. INTRODUCTION

Asynchronous Transfer Mode (ATM) is based on cell switched networks where data is formed into small fixed length packets, which are its basic unit of transportation. The term asynchronous refers to the fact that cells assigned to a particular channel occur at irregular intervals, as and when they are required. ATM is based on a form of Asynchronous Time Division Multiplexing (ATDM), however a significant distinction is that ATM uses fast packet-switching techniques in order to maintain desired data rates. An ATM cell consists of two parts, a 5-octet header and a 48-octet information field (commonly referred to as its payload) as illustrated in Figure 1. Header information is used to determine the method for routing the cell through the network, while the payload stores user defined data. Such a small cell size, resulting in the overheads occupying around 10% of the cell means that the transportation mechanism is pretty inefficient. The choice of cell size represents a compromise between the needs of data traffic and those of real-time (such as voice and video) traffic. It would be desirable to waste a smaller fraction of channel capacity in transmitting headers, however a larger payload takes longer to fill thus increases delays, which might be unsuitable for real-time data. The ATM transportation mechanism is defined by the International Telecommunications Union (ITU) and is detailed in a number of recommendation documents, see [1].

Cell Delineation is the process that allows identification of ATM cell boundaries from data received at an interface to a channel [2]. Traditionally, framing of packets in digital transmission technologies employ a Framing Word which is a unique synchronisation sequence located at a specific point within a packet that can be used to indicate the frame boundary. Since ATM uses a relatively small packet size, 53-bytes, compared with other transmission technologies, employing a Framing Word would prove to be costly. Instead, ATM makes use of the Header Error Control (HEC) byte within the cell header, which is produced using a popular error coding scheme, the Cyclic Redundancy Check (CRC). The HEC byte also has a dual purpose by providing single-bit error correction and multi-bit error detection over the header bits. This eliminates the requirement for a specific Framing Word but also complicates the process of detecting ATM cells.

The main goal of this work is to investigate and design an ATM cell delineator through to synthesis. Several algorithms exist in performing cell delineation, which have been developed through research and are detailed in a number of papers [3]-[7]. Several algorithms for performing cell delineation on a bit-serial and octet-parallel basis are considered. Four algorithms are identified in total, two bit-serial and two octet-parallel. These algorithms are investigated and implemented through to netlist generation. Analysis is provided for post-synthesis designs detailing speed, area and power parameters for each implementation at input data rates of 160Mbps and 1280Mbps.



GFC Generic Flow Control VPI Virtual Path Identifier  
VCI Virtual Channel Identifier PT Payload Type  
CLP Cell Loss Priority HEC Header Error Control

Figure 1: ATM Cell Structure

## 2. CELL DELINEATION

The process of detecting ATM cells within a channel is known as cell delineation and is based on validation of the HEC byte in order to extract cell boundaries. ATM cells may be carried in two formats [5]:

- i. As a continuous stream of cells in a cell-based format (Cell-based Physical Layer).
- ii. As cells carried within a SDH (Synchronous Digital Hierarchy)-based frame structure (SDH-based Physical Layer).

### 2.1 Serial Cell Delineation Algorithms

Serial algorithms are generally associated with cell-based format since SDH-based frame structures provide knowledge of byte-boundaries of contained cells allowing the use of octet-parallel algorithms. Cell delineation algorithms utilise the correct HEC byte of an uncorrupted cell for boundary detection and operates as follows: The receiver initially operates in the HUNT state and searches for a valid header, see Figure 2. Since byte-boundaries are unknown, this occurs on a bit-by-bit basis over the previous 40-bits. Upon detecting

a valid HEC the receiver enters the PRESYNC state and cell delineation is performed every 53 bytes (cell-by-cell). This process continues until the HEC has been consecutively confirmed  $\delta$  times, at which point the receiver enters the SYNC state. If an incorrect HEC is found while operating in PRESYNC, the process returns to HUNT. While in the SYNC state, if an incorrect HEC is consecutively obtained  $\alpha$  times cell delineation is lost. Recommended values for these parameters are,  $\alpha = 7$  and  $\delta = 8$ .

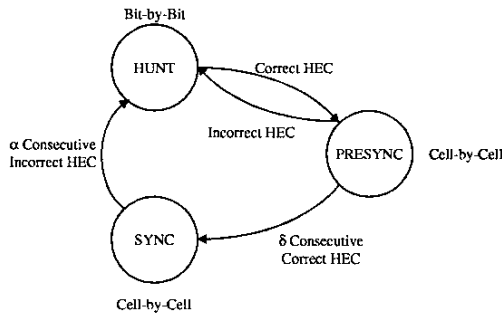


Figure 2: Cell Delineation State Diagram [2]

HUNT state searches on a bit-by-bit basis while all other states search cell-by-cell. The HUNT state is therefore the most time critical and determines the data processing rate of the cell delineation mechanism. There are several methods of performing bit-serial cell delineation, all based around a linear feedback shift register (LFSR) circuit, which consist of shift registers connected with feedback through exclusive-or (XOR) gates.

### 2.1.1 Fixed Boundary CRC Searching

The Fixed Boundary CRC searching method computes an 8-bit syndrome for a fixed 40-bit period by assuming the cell begins at an arbitrary bit location. Continuous detection of valid headers requires 40 CRC detectors in parallel. This method may be used in conjunction with any of the widely detailed standard serial LFSR based CRC detector circuits. Detailed in [5] is a Polynomial Divider Circuit implementation for syndrome calculation. The circuit, based on a generator polynomial  $G(x) = x^8 + x^2 + x^1 + 1$  can be seen in Figure 3. The contents of the shift registers are initially all set to 0, then a 40-bit message,  $I(x)$  is shifted in a bit at a time. This calculates the remainder resulting from the division of  $I(x)$  by  $G(x)$ , thus the contents of the shift holds the syndrome calculated. For successive calculations, the contents of the shift registers must be reset to zero.

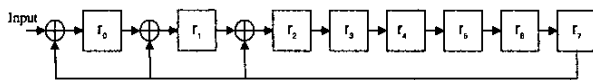


Figure 3: Serial Polynomial Divider Circuit [5]

### 2.1.2 Moving Window CRC Searching

The disadvantage of the Fixed Boundary CRC searching method is that it requires 40 CRC detector circuits in parallel for continuous detection. An alternative to this is the Moving Window CRC searching method that required a single modified CRC detector. Once again this method may be based

around any of the widely detailed standard serial LFSR based CRC detector circuits. The authors in [3] presented an efficient design which implements a serial moving window CRC detector based around a Polynomial Divider Circuit for continuous cell detection rather than the 40 needed to implement the equivalent fixed boundary CRC circuit. This detection method works by sliding the 40-bit detection window 1-bit at a time through the incoming bit-stream, entering the new incoming bit into the CRC detector while removing the effect of the bit that occurred 40 cycles previously. Figure 4 shows the Polynomial Divider based moving window CRC circuit.

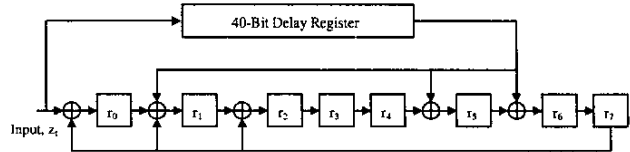


Figure 4: Serial Moving Window Polynomial Divider Circuit [3]

The shift registers and 40-bit delay registers are all initially set to zero. For the first 40-bits the circuit operates as a normal Polynomial Divider Circuit. If the operation happens to start at the beginning of the first header octet then the contents of the shift registers (i.e. the syndrome) after 40-bits would be zero, indicating a valid header. However, if the operation starts at any other 423 bit locations (53-byte ATM cell = 424 bits), the registers will most probably be non-zero after 40-bits, indicating an invalid header. Having detected the first header, there is no need to reset the shift registers to zero at the beginning of the next check as required with the fixed boundary CRC searching method.

### 2.2 Octet-Parallel Cell Delineation Algorithms

SDH-based frame structure enables the utilisation of possibly pre-existing transmission equipment by allowing ATM cells to be carried in the payload portion of an SDH STM-1 frame. This provides knowledge of byte-boundaries of contained cells and enables cell delineation to take place on a byte-by-byte rather than bit-by-bit basis. With today's high-speed transmission links, performing cell delineation on a bit-serial basis might prove to be too much of a bottleneck. The data rates required are becoming too fast for clock frequencies to keep up if processed serially, thus alternative architectures must be considered. Generally this is achieved through parallelism, processing 8 bits, 16 bits or even 32 bits at a time. This can significantly increase the data processing rate with only a minimal increase in hardware complexity, however this can only be achieved if knowledge of byte-boundaries are available. The parallel approach to speed up the CRC algorithm is to perform a number of operations in a single clock cycle. This section details octet-parallel algorithms for cell delineation and shows the speedup that can be achieved when byte-boundaries within a channel are known.

#### 2.2.1 Fixed Boundary CRC Searching

Section 2.1.1 showed how the Fixed Boundary method was achieved on a bit-serial basis. Knowledge of byte-boundaries allows the input to be considered as a byte-stream rather than a bit-stream, resulting in continuous header detection and

validation requiring only 5 parallel CRC detectors as opposed to 40 serial CRC detectors. Fixed Boundary CRC searching may make use of the parallel implementations of either *Polynomial Divider Circuit* or *CRC Register Circuit*.

### 2.2.2 Moving Window CRC Searching

The authors in [5] present a parallel approach for CRC calculation based on a moving window CRC detector introduced in [4]. Referring back to the implementation of the serial moving window CRC detector (Section 2.1.2), this detection method works by entering the new incoming bit into the CRC detector while removing the effect of the bit that occurred 40 cycles previously. In the parallel approach the input is a byte-stream rather than a bit-stream therefore the implementation must slide through the incoming data an octet at a time, entering the new incoming octet while removing the effect of the octet that occurred 5 cycles previously.

## 3. IMPLEMENTATION

### 3.1 Serial Fixed Boundary Implementation

The cell delineator was implemented in synthesisable RTL Verilog as an FSM comprising of 3 states, HUNT, PRESYNC and SYNC, with states changing as described in Section 2.2. The Serial Fixed Boundary Polynomial Divider Circuit requires 40 CRC detectors for continuous searching in the HUNT state. It would be wasteful to repeat the same code 40 times, thus the HUNT state was partitioned into a separate sub-module (HUNT-CRC), which produced an output signal at the end of a 40-bit period to indicate whether a correct header (syndrome = 0) was detected. If the number of bits entered was less than 40 the output was to be low, rising on the 40th bit if the syndrome calculated was zero, and staying low if not. The module was designed to reset all registers after the 40th bit so the same module could perform successive header detection. The HUNT sub-module instantiates HUNT-CRC block 40 times, each searching over a different 40-bit period simultaneously to provide continuous searching. For this to occur the 40-bit search window for each CRC detector is staggered by 1-bit from the previous. To allow this an enable signal was incorporated into the design, which would keep the sub-module in a reset state until required to begin searching. In addition, a second enable signal was provided which keeps all CRC detectors in a reset state while the cell delineator is not operating within the HUNT state.

The function of the HUNT sub-module is to indicate to the control block whether a correct syndrome was detected in any of the 40 CRC detectors. Another function is to stagger the enable signals for each CRC sub-module by 1-bit from the previous, allowing continuous detection.

The PRESYNC and SYNC states are only required to perform the CRC function once per cell over the assumed 40-bit cell header, thus only a single CRC detector is required for each state. Since the CRC function had already been partitioned into a separate stand-alone sub-module it was reused instead of burying the same functions into the PRESYNC and SYNC blocks. The CRC sub-module however required a slight modification, which was to include an additional signal to indicate whether the syndrome detected over the 40-bit period was incorrect. This signal was included due to the need for PRESYNC and SYNC states to change on an incorrect

syndrome or sequence of incorrect syndromes detected. The HUNT state on the other hand disregards all incorrect headers and continues searching; thus, this signal is not required. The PRESYNC and SYNC sub-modules were therefore used for controlling the operations while in these states. A block diagram showing the system partitioning and communicating signals may be seen in Figure 5.

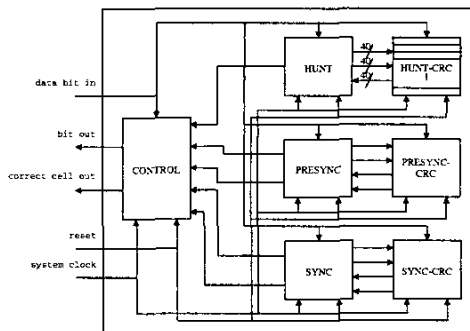


Figure 5: System Partitioning for Serial Fixed Boundary

### 3.2 Serial Moving Window Implementation

The second algorithm chosen for implementation was the Serial Moving Window Polynomial Divider Circuit. Referring back to Section 3.1, the same partitioning was developed for this design, however the HUNT and CRC sub-modules were replaced with a single HUNT sub-module as can be seen in Figure 6. The reason for implementing the HUNT state of the Fixed Boundary design as two sub-modules was that 40 CRC detectors were required and therefore a single CRC detector could be instantiated 40 times within another module containing the control logic. The Moving Window algorithm is completely different in concept to Fixed Boundary and only required a single modified CRC detector. This resulted in all HUNT state logic being contained in a single sub-module.

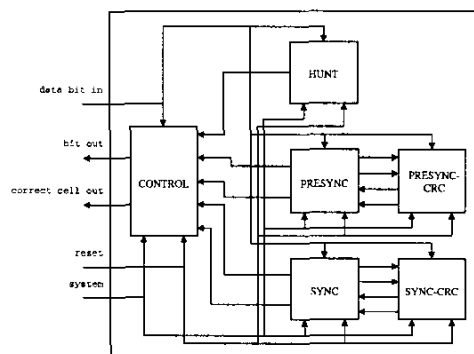


Figure 6: System Partitioning for Serial Moving Window

In implementing the Serial Moving Window Polynomial Divider Circuit based cell delineator much of the Fixed Boundary design could be reused. The only block required to be designed was the HUNT sub-module. The function of the HUNT sub-module was to detect whether the previous 40 input bits comprised of a valid cell header utilising the Moving Window method for cell delineation. The Moving Window detection method, unlike the Fixed Boundary, only requires a single CRC detector. This meant that the HUNT

sub-module was shorter and simpler to code than for the Fixed Boundary design. The general coding of the sub-module was based on the CRC sub-module (Section 3.1), however the function for performing the algorithm was re-coded to reflect the new design. The output from the HUNT sub-module remains low for the first 39 input bits while operating in the HUNT state and then rises if a syndrome of zero is calculated, indicating that the previous 40 bits comprised a valid header. The sub-module is only enabled while operating in the HUNT state, allowing a syndrome to be calculated, and is held in a reset state for all other operating states.

### 3.3 Octet-Parallel Fixed Boundary Implementation

The major differences between the bit-serial and octet-parallel cell delineators are the interfaces between modules, due to a byte rather than a bit being presented per clock period, and the type of algorithm incorporated. All the control signalling are however identical apart from minor changes to reflect that processes are completed on an octet rather than bit basis. The same partitioning strategy as developed for the serial implementations could therefore be used. The blocks were designed to operate in the same manner as the serial implementation, see Section 3.1. The alterations required in adapting the serial design to the octet-parallel implementation were only minor. Obviously the data ports and data handling registers had to be modified to a byte in length, while the number of instantiations of the CRC module needed to be reduced from 40 to 5. All the control logic was left the same apart from the number of enable signal to be generated were also reduced from 40 to 5.

### 3.4 Octet-Parallel Moving Window Implementation

In implementing the Octet-Parallel Moving Window Polynomial Divider Circuit based cell delineator much of the Fixed Boundary design could be reused. As with the serial Moving Window implementation the only block that required redesigning, having implemented the octet-parallel Fixed Boundary design, was the HUNT sub-module. All the blocks from the Octet-Parallel Fixed Boundary design were reused in this implementation apart from the HUNT sub-module. The function of the HUNT block was to detect whether the previous 5 input octets comprised a valid header utilising the Moving Window method for cell delineation. The structure of this block is similar to the serial equivalent (Section 3.2) with modifications for performing parallel cell delineation as described in Section 3.3.

## 4. RESULTS

The algorithms have been synthesised down to 0.18 $\mu$ m Alcatel CMOS technology, using Synopsys' Design Compiler. All algorithms have been subjected to identical constraints, for a clock frequency of 160MHz and minimal area. The resulting netlists were then simulated to verify the functionality of the implemented algorithms and compute their power consumption with Synopsys' Design Power, at a clock frequency of 160MHz and 1.8 volts. The results are shown in Table 1. When data processing rates are compared, octet-parallel implementations result in a speed-up of x8 (1280Mbps) compared to serial implementations (160Mbps), as might be expected from processing a byte rather than a bit at a time. However, this does not necessarily translate to an

equivalent increase in area. For example, the Octet-parallel Fixed Boundary implementation has a significant number of registers less than its serial equivalent resulting in a significant reduction in area. The Octet-parallel Moving Window circuit on the other hand has a comparable number of registers to its serial equivalent, resulting in only a slight increase in area. For serial implementations, the algorithm to choose is the Moving Window Polynomial Divider Circuit due to its less area and power consumption compared to Fixed Boundary Polynomial Divider Circuit. Similarly, for octet-parallel implementations the Moving Window Polynomial Divider Circuit achieves better results in terms of area and power consumption compared to the Fixed Boundary Polynomial Divider Circuit. Overall, the algorithm that comes out on top is the Octet-Parallel Moving Window Polynomial Divider Circuit due to its small area, high data processing rate and low power consumption.

Table 1: Post-synthesis Comparisons

Algorithm	Area (NAND gates)	Critical Path (ns)	Power (mW)
Serial Fixed Boundary	12633	5.40	21.40
Serial Moving Window	2091	3.76	3.68
Octet-Parallel Fixed Boundary	3356	3.75	5.31
Octet-Parallel Moving Window	2299	3.75	3.76

## 5. CONCLUSIONS

Four algorithms, two bit-serial and two octet-parallel, for detecting ATM cells within a channel have been implemented through to synthesis. Analysis has been provided for post-synthesis designs detailing speed, area and power parameters for each implementation at a clock frequency of 160MHz. As expected, octet-parallel implementations achieved a speed-up of x8 compared to bit-serial ones, with only a slight increase in area for the Moving Window Polynomial Divider Circuit and a significant decrease in area for the Fixed Boundary Polynomial Divider Circuit. In overall, the best performance in terms of area, data processing rate and power consumption was achieved with the Octet-Parallel Moving Window Polynomial Divider Circuit.

### ACKNOWLEDGEMENTS

Dr. A.T. Erdogan would like to acknowledge the support of the EPSRC under grant no GR/NO8332.

### REFERENCES

- ITU-T Recommendation I.361, B-ISDN ATM layer specification, 1999.
- ITU-T Recommendation I.432.1, B-ISDN user-network interface Physical layer specification: General characteristics, 1999.
- Dodds, D.E. & Du, L. 1994, ATM Framing Using CRC Byte, IEEE Conference on Communications, pp. 410-414.
- Ramabadran, T.V. & Gaitonde, S.S. 1988, A Tutorial on CRC Computations, IEEE Micro, pp. 62-75.
- Ng, S.L. & Dewar, B. 1996, Parallel realization of the ATM cell header CRC, Computer Communications, 19, pp.257-263.
- Maniatopoulos, A., Antonakopoulos, T. & Makios, V. 1996, Implementation issues of the ATM cell delineation mechanism, IEE Electronics Letters, Vol.32, No.11, pp. 963-965.
- Pei, T. & Zukowski, C. 1992, High-Speed Parallel CRC Circuits in VLSI, IEEE Trans. on Communications, Vol. 40, No. 4, pp. 653-657.