Adaptive Clock Recovery and Jitter Control in ATM Networks

By

Ammar Muhiyaddin

B.Sc. Computer Engineering, KFUPM, Dhahran, Saudi Arabia, 1991.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 1995

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date Dec 1, 1995

Abstract

Transporting continuous bit rate (CBR) or real-time periodic traffic is one of the major services that ATM-based B-ISDN technology is promising to provide. This service requires the receiver to preserve the original inter-cell spacing. However, statistical multiplexing and buffering in the ATM transport networks can introduce significant jitter in the inter-arrival period of the cell stream, thus degrading the quality of the cell play-back at the receiver. An additional complication in plesiochronous networks is the absence of the transmitter clock frequency at the receiver. Therefore, the receiver must be capable of extracting the frequency of the transmitter clock and removing the jitter from the arriving cell stream. This thesis provides a thorough treatment of the clock recovery and jitter removal problems for CBR traffic in ATM networks, and proposes a new practical design of a receiver unit for handling multirate CBR traffic. The design proposed complies with the ATM standards. Our design employs a number of control parameters that can be varied to optimize the operation of the receiver and to provide high adaptability to rapidly changing input cell traffic. The proposed scheme is based on monitoring the fluctuation in the receiver buffer occupancy to derive a jitter free receiver clock. The hardware design has been specified and simulated extensively using VHDL (a hardware description language), and the simulation results show that our design is robust and very effective in removing cell delay jitter and restoring the original CBR stream.

Table of Contents

List of Figures

List of Tables

Acknowledgment

I wish to thank my supervisor, Dr. Hussein Alnuweiri, for introducing me to the thesis topic that best suited my research interests. His guidance, continuous support, and patience throughout this work are most appreciated. This work was supported by the GREAT Scholarship provided by British Columbia Science Council and PMC-Sierra, Inc. I would like to thank my colleagues in the VLSI and the communication labs for being helpful and understanding. I can not thank enough my wonderful parents for everything they did and still do. Many thanks go to my wife whose presence in my life made me able to go through this work. My brothers, sisters, and true friends are greatly appreciated for their encouragement which kept me going. Lastly, I thank God for blessing me with knowledge and energy to pursue this goal, and I thank Him for providing me with all of those who helped me.

1 Introduction

1.1 Background

1.1.1 ATM Technology

In this rapidly changing information age, the demand for new services is ever increasing. Some of the services planned for the near future are High-Definition Television (HDTV), video teleconferencing, video-on-demand, and virtual reality [20] [27]. However, many challenging services are still emerging and they require very reliable and flexible systems to support their demands. This has given rise to the Broadband Integrated Services Digital Network (B-ISDN). B-ISDN is a standardized public switched telecommunications network infrastructure that supports both narrowband and broadband services on a single flexible network platform and interfaces [16] [25].

The promise of B-ISDN is to provide cost-effective support for services with both constant and variable rates including data, voice, still and moving pictures and, with a particular emphasis, multimedia applications. The bandwidth ranges needed for some of the B-ISDN applications are shown in table 1.1.1 [20]. However, one of the challenging tasks in developing B-ISDN networks is meeting the Quality Of Service (QOS) requirements for a wide range of envisioned services. The QOS requirements are the user's view of a service and, in many instances, they are hard to define since there are different types of users, and different types of applications [25]. Although significant progress has been made towards the development of B-ISDN technology, more work is needed to define and implement

Service	Bandwidth (Mbit/s)
Data transmission	1.5 to 130
Document transfer/retrieval	1.5 to 45
Videoconference/videotelephony	1.5 to 130
Broadband videotex/video retrieval	1.5 to 300
TV distribution	30 to 130
HDTV distribution	130

Table 1.1.1 Broadband Services and Bandwidth

the different measures required to ensure a high reliability of the network and to provide adequate service guarantees.

Asynchronous Transfer Mode (ATM) is considered by both the telecommunications and information processing communities to be an ultimate solution for B-ISDN. ATM combines the advantages of both circuit and packet-switching techniques which results in low overhead and processing, a low and bounded transfer delay, and flexible bit rates assigned for individual connections. This allows all services to be transported and switched in a common digital format. The ATM standards are defined by the International Telecommunication Union-Telecommunications Standardization Sector (ITU-T) which is a specialized United Nations agency charged with the consultation of global telecommunications. The ITU-T was created on March 1, 1993 to replace the International Telegraph and Telephone Consultative Committee (CCITT) which ceased to exist by the end of February 1993. In the absence of the ITU-T recommendations, a group of vendors and users known as the ATM Forum develop some interim standards for some aspects of the ATM [27].

ATM is a connection-oriented network, where a virtual connection must be established among the communicating parties prior to the transfer of user information. Moreover, ATM

supports both connection-oriented and connectionless services. In ATM, user information is packetized and transmitted among the communicating parties. The information is divided into fixed-size packets called ATM cells. The size of a single ATM cell is 53 bytes, carrying 48 bytes in the information field and a 5–byte header. ATM uses statistical multiplexing to combine the cells of the different users onto the network, and it allows point-to-point communication as well as multicasting.

The four main categories of the traffic over the ATM network are the Constant Bit Rate (CBR), the Variable Bit Rate (VBR), the Available Bit Rate (ABR), and the Unspecified Bit Rate (UBR) services. The CBR service supports applications that require an uninterrupted flow of information such as voice [25]. Therefore, the QOS requirements of this service have to be strictly met. The VBR service is designed to support applications that have bursty nature and should take advantage of the idle periods between information transmissions. An example of such applications is the Moving Picture Experts Group-2 (MPEG-2) video [27].

The ABR service is intended to support economically applications with vague requirements for throughput and delays, such as data traffic applications. These applications may use trial and error to narrow down their vague QOS requirements. They depend on the feedback from the network to the traffic source to avoid cell loss and provide certain guarantees to the user. The UBR service is intended for applications with minimal service requirements, such as file transfers or electronic-mail submitted in the background of a workstation. [3]

1.1.2 Cell Delay Variation in Real Time Services

The class of CBR services is one of the services considered by the standardization committees to be used over the ATM networks. CBR services generate traffic as a continuous

Service	Bandwidth (Kbps)
Telephony, fax, data retrieval	64
Group IV fax	64
Video telephony (low quality)	128
Hi-Fi Stereo	1,400
Proprietary fax	1,500
Standard TV	140,000

Table 1.1.2 Bandwidth Requirements for CBR Services

bit stream and at a constant rate. Examples of such services include voice, video, and audio, with bandwidth requirements shown in table 1.1.2 [25] [26]. However, the future CBR applications are expected to need larger bandwidth capacities.

An example of CBR applications is voice telephony which is transmitted at a rate of 64 Kbps as specified by the CCITT Recommendation G.711 [17]. The analog voice signal is sampled every 125 *us* and the signal amplitude is quantized into 256 steps. This results in a Pulse Code Modulation (PCM) frame length of 8 bits. ATM cells of CBR services have 47 bytes of pure information in their payloads (since one byte is used for error detection and cells sequence checking). Therefore, the ATM cell payload (47 bytes) is capable of transmitting 5.875 *ms* of voice. Transmitting voice over ATM networks may add more delay to the cells due to packetization, depacketization, multiplexing, buffering and the other transfer delays. Therefore, as one of the QOS requirements for voice applications, the tolerable delay in the one-way trip of voice cells is considered to be about 25 *ms*. If the delay exceeds this value, echo cancellation techniques should be employed [20].

To ensure the proper exchange of information between the different communicating entities, the transmitter (source) and receiver (destination) should be synchronized. This is

because the transmitter and the receiver may use different clock frequencies for writing to and reading from the network, respectively. A problem may arise if the destination reading clock is slower or faster than the average rate at which the network delivers the packets. The former situation will cause overflow in the destination buffer while the latter will cause depletion or underflow in the destination buffer.

Moreover, although ATM provides a very flexible information transfer with respect to bit rate, the QOS remains a crucial requirement. Two of the main performance parameters that have to satisfy the negotiated QOS requirements for real-time information transfer are the overall end-to-end delay of cells, and the Cell Delay Variation (CDV) which is commonly referred to as *jitter* [13] [23] [33].

Figure 1.1.1 shows an example of an ATM network that produces jitter. The transmitter generates the cells in a periodic CBR fashion. Due to the statistical nature of the transporting cells across the network, the arriving cells have a random difference in the arrival time. Different parts of the network may contribute to the overall cell jitter. The jitter components could include the variation in the delay at the multiplexer and the buffers of the intermediate network nodes, and the variation in the delay that may occur due to the congestion in the different network entities [30]. A scheme should exist to control the jitter and remove its effect at the destination.

1.2 Thesis Objectives

The main goal of this thesis is to find a practical solution for the adaptive clock recovery and jitter removal problem in ATM networks. In order to achieve this goal, the thesis will:

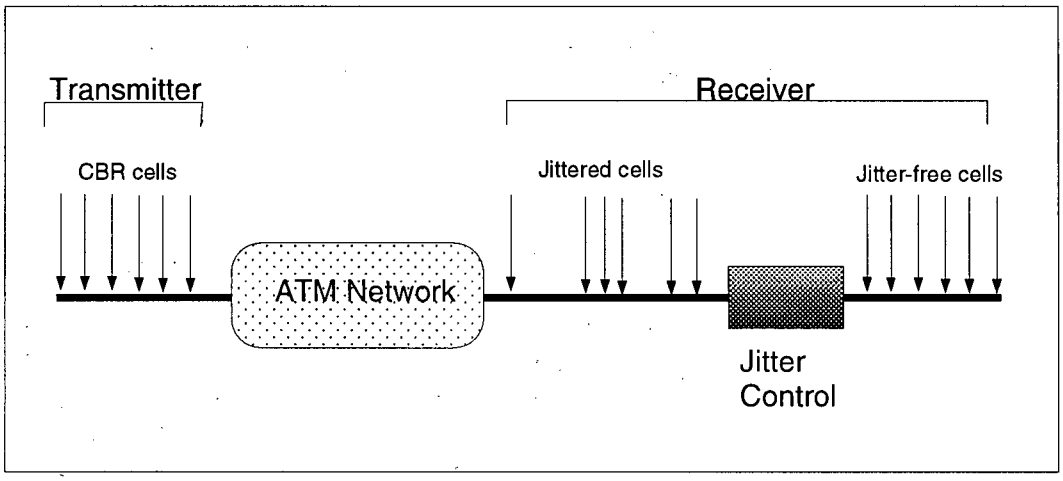

Figure 1.1.1 Jitter Produced by an ATM Network

1. Provide characterization of clock recovery techniques for ATM CBR traffic that control the jitter at the destination, and synchronize the transmitter and the receiver.
2. Propose an enhanced clock recovery scheme based on buffer-level monitoring that complies with the ITU-T standards. The proposed scheme should be robust enough to handle the jitter problem, and to reduce the cost of the buffer by properly controlling its size.
3. Provide full specification of the clock recovery system and develop a complete behavioral description of the hardware implementation. The main design tool used is VHDL (Very-high-speed-integrated-circuit Hardware Description Language) with supporting computer-aided design (CAD) tools.
4. Model the cell delay variation in CBR traffic, and use this model to simulate and analyze the system performance under different adaptation schemes.
5. Identify the work needed to extend the design and the implementation goals.

1.3 Previous Work

The different techniques used for the synchronization between the source and the destination depend on the operation of the network. Synchronized network operation assumes that there is a common clock frequency for the network which is available to all the different parties. On the other hand, a plesiochronous network operation means that the transmitter and the receiver do not have a common network reference clock but the different entities can assume a nominal clock value that can be changed in the synchronization process. Different approaches have been suggested in the literature for solving the synchronization problem [10] [12] [15] [18] [19] [21], but no references were found for actual hardware implementations. Only few commercial products provide some sort of hardware solution to the problem. These products include the TranSwitch COBRA and the MAZ Internetworking Unit DAS 112 [4].

The Synchronous Residual Time Stamp (SRTS) [21] is a standardized scheme used in synchronized networks where a reference clock is available at the transmitter and the receiver of the network. The Residual Time Stamp is used to measure and transmit information about the frequency difference between the network clock and the clock used at the entity. This method can not be used in networks where a reference clock is not available. Another method for network synchronization is cell spacing, a technique that handles early cell arrivals by ensuring a minimum spacing between two consecutive cells on a connection, but it does not handle late cell arrivals [18] [19].

The negative retiming stuffing technique [15] claims to be able to remove the cell delay variation from the cell stream at the receiver. However, this method does not comply with the standards set for the receiver in an ATM network and requires more complex implementations.

The most suitable scheme found for solving the clock recovery and jitter control problems at the receiver is done by monitoring the buffer level at the receiver [10] [12]. Variations in the buffer level give an indication of the difference between the transmitter and the receiver clock frequencies, and consequently can be used in adjusting the receiver clock frequency to match that of the transmitter.

1.4 Thesis Overview

This thesis is organized as follows. Chapter 2 provides an introduction to delay and jitter in ATM networks. It reviews the ATM protocol reference model and describes the layer that handles the clock recovery and jitter control problems in periodic real-time traffic. The performance parameters needed to ensure a reliable service will be outlined, and jitter in ATM networks will be formally defined. Finally, a survey of the proposed techniques for handling synchronization and jitter control problems in ATM networks will be provided.

Chapter 3 introduces the adaptive clock recovery and jitter control technique. It explains the two algorithms used for estimating the frequency difference between the transmitter and the receiver clocks, and for updating the receiver clock frequency. It provides the design parameters that can be used to control the operation of the algorithms. Chapter 4 explains in detail the hardware model of the adaptive clock recovery system, and provides the full hardware specifications of the system. Chapter 5 describes the input traffic generator used to simulate the jittered input cell stream, and presents extensive simulation results. Chapter 6 concludes the work presented and provides some suggestions for future work.

2 Delay and Jitter in ATM Networks

Meeting the QOS requirements of CBR traffic is very crucial for ensuring the proper operation of many real-time applications. The cells end-to-end delay and jitter in ATM networks are key QOS elements. Therefore, they must be well defined in order to analyze, characterize, and propose solutions that meet such QOS requirements.

In this chapter, the ATM reference model will be briefly overviewed, and details of the adaptation layer responsible for handling the CBR services will be presented. Then, performance guarantees and the different parameters that affect cell transfer performance will be discussed. Formal definitions of the cell delay variation will be given, and the different techniques proposed for solving the synchronization and jitter problem will be reviewed.

2.1 The ATM Protocol Reference Model

The ATM protocol reference model [6] is shown in figure 2.1.1. The protocol consists mainly of three layers, the Physical Layer, the ATM Layer, and the ATM Adaptation Layer (AAL).

2.1.1 The Physical Layer

The physical layer transports the ATM cells between two ATM entities. The functions of the physical layer are grouped into the Physical Media (PM) and the Transmission Convergence (TC) sublayers. The PM sublayer includes the physical medium dependent functions. This sublayer provides bit transmission capabilities including bit transfer and bit alignment. Moreover, the PM sublayer handles the bit timing functions such as the generation

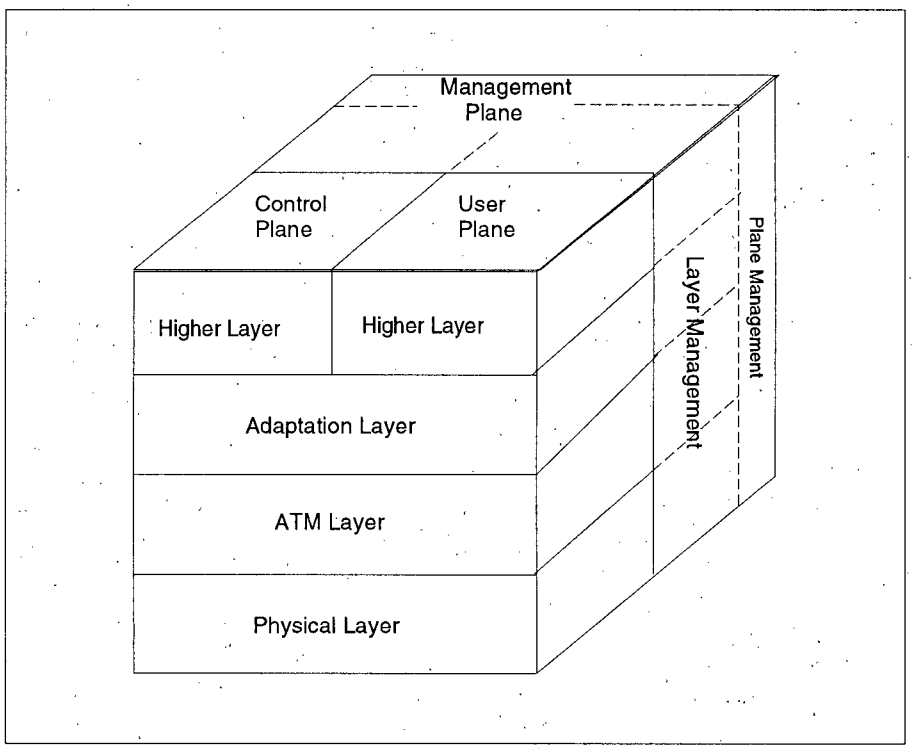

Figure 2.1.1 ATM Protocol Reference Model

and reception of waveforms suitable for the medium, the insertion and extraction of bit timing information, and line coding if required. It is also responsible for the electrical-to-optic and optic-to-electrical transformations. The physical medium used in this sublayer can be an optical fiber, a coaxial, or a twisted pair cable.

The TC sublayer performs the ATM cell Header Error Control (HEC) sequence generation in the transmit direction and the HEC verification at the receiving end. Another TC sublayer function is frame and cell delineation which means determining the cell boundaries from the stream received from the PM sublayer. The TC sublayer inserts idle cells in the sending direction and suppresses them at the receiver in order to synchronize the rate of the ATM cells to the payload capacity of the transmission link.

Some of the existing physical interfaces are [25]:

- SONET STS-3: The STS-3 is a 155.520 Mbps interface that supports the ATM cells by providing a physical payload envelope through a framing structure. The frame also includes overhead bytes associated with Operations and Maintenance (OAM) functions. The payload size per frame is 2,340 bytes and the frame is not an integer multiple of the cell size.
- DS3: The DS3 is a 44.736 Mbps interface. It consists of a 125 μsec frame that includes 12 ATM cells, each preceded by a 4–byte header.
- The 100 Mbps Multimode Fiber Interface: This interface is used in private networks and is designed around the Fiber Distributed Data Interface (FDDI) standard.
- The 155 Mbps Multimode Fiber Interface: This is a 155 Mbps interface with a frame size of 27 ATM cells. One cell is used for delimiter and OAM functions and the rest are user cells.

2.1.2 The ATM Layer

The ATM layer is a connection-oriented layer that provides its services based on pre-established connections with its peer entities. The ATM layer is common to all services. It provides the transparent transfer of fixed sized ATM cells between communicating upper layer entities.

In the sending direction, after receiving the cell information from the AAL, the ATM layer generates and appends the cell header (except the HEC value). The ATM layer multiplexes cells from different sources into one cell stream. This layer is also responsible for routing these cells through the switching and cross-connect nodes. During the establishment of the

ATM connection, the parameters needed are negotiated. These parameters can include cell loss, end-to-end delay, jitter, throughput, and some other traffic parameters. Therefore, the ATM layer supervises the cell flow control to ensure that the connections stay within the limits negotiated at the call establishment stage in order to provide shared medium access.

At the receiving end, the ATM layer demultiplexes the cells by splitting the arriving cell stream into the different flows for the destinations. The ATM cell header is extracted and the cell information field is passed to the AAL [2] [25].

2.1.3 The ATM Adaptation Layer

The simplicity and the flexibility of the ATM layer is achieved by leaving out some services needed in providing the QOS requirements for the different applications of B-ISDN. This is because not all the applications require the same types of services. Therefore, considering the commonality between the existing and the expected applications in B-ISDN networks, a small number of classes have been created to group the functionalities required by the various service classes. The AAL provides each service class with the functionalities required to reach the QOS requirements [25]. Subsequently, the AAL enhances the service provided by the ATM layer to support functions required by the next higher layers. The term Adaptation Layer is intended to describe the process of adapting data into a form suitable for ATM [27].

Four classes are defined to categorize the different B-ISDN services:

- Class A represents the CBR connection-oriented services with a timing relation between the transmitter and the receiver.

- Class B represents the VBR connection-oriented services with a timing relation between the transmitter and the receiver.
- Class C represents the VBR connection-oriented services with no timing relation between the transmitter and the receiver.
- Class D represents the VBR connectionless services with no timing relation between the transmitter and the receiver.

The standardization committee in the ITU-T [21] has defined five types of AAL protocols corresponding to the four classes discussed above. The AAL–0 protocol is not well defined yet in the standards. However, it is essential to assume its presence since it defines the case when no adaptation layer is needed, and the information field contents are passed directly from the ATM layer to the higher layers. The AAL–1 protocol supports applications that require CBR services (class A) such as voice telephony. Certain applications (such as VBR video and audio) require connection-oriented services with a timing relation between the communicating entities. Such applications (class B) are supported by the AAL–2 protocol.

The AAL–3/4 protocol was designed to support non-real-time applications providing framing services for connection-oriented and connectionless data protocols. The data communication community decided not to use the AAL–3/4 protocol because of the concerns about performance, implementation, and overhead. Therefore, AAL–5 was developed as an efficient protocol for data communication applications to reduce the overhead and the complexity of the AAL–3/4 [27].

Some other protocols will possibly be defined in the future [11]. In this thesis, the problems to be discussed occur in real-time CBR applications and, therefore, the AAL–1

layer will be described in more details.

2.2 The AAL–1 LAYER

In this type of AAL, CBR information is transferred between the transmitter and the receiver, between which a connection has been set up. Moreover, a timing relation between the transmitter and the receiver has to be established. Some of the examples of this service are high-quality audio, video, and telephony [25]. Moreover, by providing circuit emulation, a CBR service can be used to replace leased-lines [27].

Some of the functions that may be performed in the AAL-1 are [21]:

1. Segmentation and reassembly of user information.
2. Handling of cell delay variation.
3. Handling of cell payload assembly delay.
4. Handling of lost and misinserted cells.
5. Source clock frequency recovery at the receiver.
6. Recovery of the source data structure at the receiver.
7. Monitoring of user information field for bit errors and possible corrective actions.

The AAL-1 is divided into two sublayers, namely the Segmentation And Reassembly (SAR) sublayer, and the Convergence Sublayer (CS). In general, the main functions of the SAR sublayer are segmenting the higher layer data into fixed size cells (48 bytes) to be sent to the ATM layer, and at the receiving end, the reassembly of the information fields into data for higher layers. However, in AAL–1, the functions of the SAR sublayer are very basic since no segmentation or reassembly of cells take place at this sublayer. These functions are

handled mainly by the CS, and therefore, the SAR sublayer in AAL–1 is just for keeping consistency in the terminology. The CS provides the AAL service at the Service Access Points (AAL-SAP) and, therefore, is service dependent.

2.2.1 Segmentation And Reassembly Sublayer

At the transmitting side, the SAR sublayer accepts a 47 octet block of data from the CS to form the SAR Service Data Unit (SAR-SDU). The SAR sublayer then adds a 1 octet header to each block to form the SAR Protocol Data Unit (SAR_PDU) which is shown in figure 2.2.2.

In addition to the data block, the SAR sublayer receives a Sequence Number (SN) value (4 bits) from the CS. The SN field is divided into a 3–bit Sequence Count (SC) field and a 1–bit Convergence Sublayer Indicator (CSI) field. The SC is used for lost or misinserted SAR payloads, and the CSI is used to indicate the existence of the CS [5]. The SC and CSI fields are protected against bit errors by a 4–bit Sequence Number Protection (SNP) field which is capable of multiple bit error detection and single bit error correction. The SNP field is subdivided into two subfields for providing a two-level protection. The Cyclic Redundancy Check (CRC) code is a 3–bit subfield used to protect the SN field, and the resulting 7–bit codeword is protected by an even parity check bit.

In addition to accepting the 47 octet from the CS, the SAR sublayer computes the CRC value at the transmitting side and inserts it in the header. Then the transmitter inserts the even parity bit to form the full SAR-PDU.

At the receiving end, the SAR sublayer receives a 48 octet block of data from the ATM layer, separates the SAR-PDU header, and passes the 47 octet block of data to the CS. The

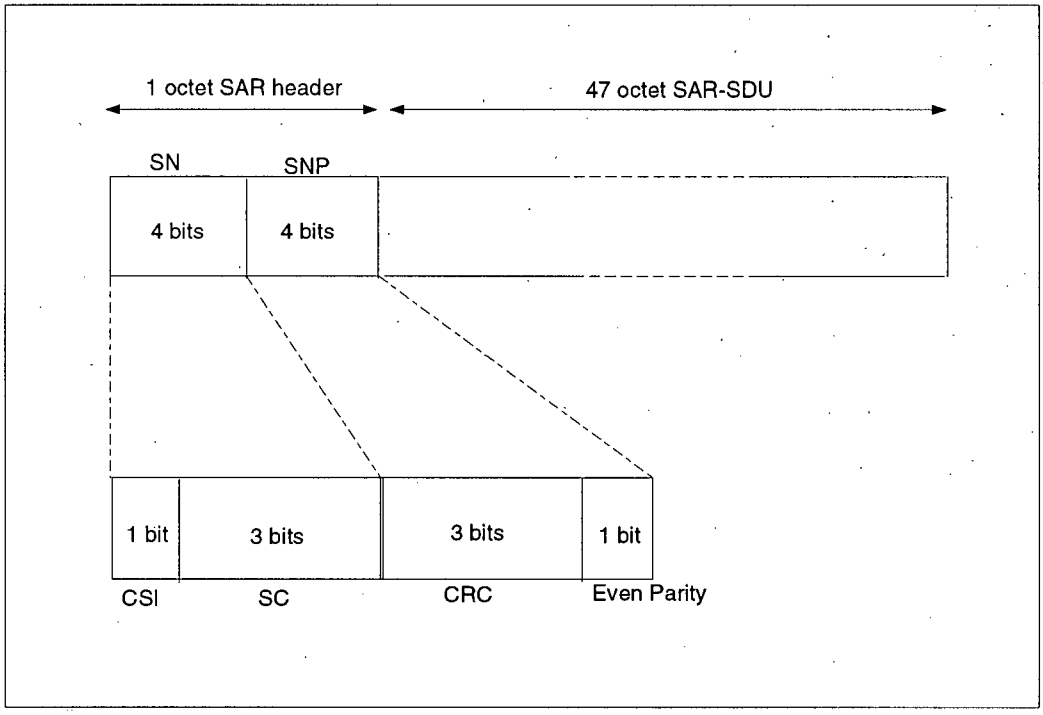

Figure 2.2.2 SAR-PDU for AAL1

SAR also passes the SN (SC and CSI) value to the CS together with the SN check status informing the receiving CS wether the SN is valid or not.

2.2.2 Convergence Sublayer

The particular service to be supported strongly determines the functions that exist in the CS. Therefore, the ITU-T Recommendation I.363 [21] defines the CS functions for the synchronous and asynchronous circuit transport, video, voice, and high quality audio signal transport. The following is a general description of the CS functions, which may or may not exist in the different applications.

The CS handles the cell delay variations for delivering the Service Data Units (CS-SDU) to an ALL–1 user at a constant bit rate. A buffer is used to support this function. Moreover, the CS can provide a mechanism for recovering the source clock frequency at the

destination end. For the CS to handle the cell delay variations, the recovery of the source clock frequency, and some other functions, the CS needs some timing relation between the source and the destination. This relation can be explicit by using a time stamp transmitted by the source and used at the receiver to recover its clock (when a common network clock frequency is known to both the transmitter and the receiver), or it can be implicit as in observing the fluctuation in the destination buffer level.

Detection of cell loss and misinsertion can be done at the CS using the sequence count value and its error check status provided by SAR. Additionally, to ensure the high quality in some applications, CS provides Forward Error Correction (FEC) to protect against bit errors. The CS may also provide some users with the transfer of structured information between the source and the destination.

2.3 Performance Parameters

Since meeting the QOS requirements of multiple services is one of the most important issues in ATM networks, certain parameters have to be defined to measure the QOS requirements of different applications. Moreover, the types of guarantees an application may receive should be clearly specified.

2.3.1 Performance Guarantees

Three related properties of a flow (cell stream carried on the same connection) need to be guaranteed in order to have a reliable application: the worst-case loss rate, the worst-case bandwidth, and the worst-case delay. Each one of these properties implies

different requirements depending on the service used. For example, jitter is a very important requirement implied by the worst-case delay for real-time applications using CBR traffic.

An ATM network may offer either statistical or deterministic guarantees for these properties. Deterministic guarantees resemble a confirmed promise. Examples include the promise that all data will arrive within a specific delay after being transmitted, the flow will have access to a specific bandwidth all the time, and the cell loss will not exceed a given value. Statistical guarantees promise that only a portion (high percentage) of the properties is guaranteed. A high portion of data will arrive within a specific delay, the average bandwidth available to the flow is x bits per second, and the possibility of having a certain cell loss is guaranteed for a percentage of the cell stream, are all examples of statistical guarantees. [27]

2.3.2 Cell Transfer Performance Parameters

Since different applications may have different QOS requirements, standardization bodies such as the American National Standards Institute (ANSI) for Telecommunications and the ATM Forum proposed many performance parameters that could be used by the different applications. Some of the performance parameters that are recommended by the ATM Forum and the ANSI for Telecommunications (T1.511–1994) [31] are:

1. Peak Cell Rate (PCR) is the maximum rate at which cells can be transmitted.
2. Sustained Cell Rate (SCR) is the average cell rate over a long period of time.
3. Minimum Cell Rate (MCR) is the minimum rate desired by the user.
4. Cell Error Ratio (CER) is the ratio of the total number of errored cells to the total number of errored and successfully transferred cells.

5. Cell Loss Ratio (CLR) is ratio of the total number of lost cells to the total number of cells that are errored, lost, or successfully transmitted.
6. Cell Misinsertion Rate (CMR) is the number of misinserted cells in a given time interval.
7. Severely Errored Cell Block Ratio (SECBR) is the ratio of the total number of severely errored cell blocks to the total number of cell blocks.
8. Cell Transfer Delay (CTD) is the difference between the arrival time of a cell at the destination, and the transmission time of the cell from the source. The CTD includes two important parameters:
   a. Mean cell transfer delay is the arithmetic average of a specified number of cell transfer delays
   b. Cell delay variation is the variability in the pattern of cell arrival events.

2.4 Cell Delay Variation

2.4.1 Cell Delay in ATM Networks

Signals experience delay when they traverse the network from the source terminal equipment to the destination terminal equipment. This is called the end-to-end delay. It arises from the transmission length and the delay in the different network entities. For an end user of the AAL service, the delay encountered may have the following components [31]:

1. Coding and decoding delay.
2. Segmentation and reassembly delay which includes,
   a. Segmentation delay in the AAL of the sending side.
   b. Buffering delay in the AAL of the receiving side to eliminate the jitter.

c. Reassembly delay in the AAL of the receiving side.

3. Cell transfer delay which includes,

a. Inter-ATM transmission delay.

b. ATM node processing delay which results from queueing, switching, routing, etc.

Some of the delay components do not have a fixed value and may change stochastically from one cell to another in the same cell stream. This jitter may alter the traffic characteristics of an ATM connection. In general, jitter refers to the distortion in the interval times of a CBR cell stream. A precise definition of CDV is controversial [28]. The following are the definitions introduced in the (T1.511–1994) ANSI standard for Telecommunications [31]. In this standard, the CDV can be defined as a 1–point CDV, or a 2–point CDV.

2.4.2 1–point CDV

The definition of the 1–point CDV is based on observing a sequence of consecutive cell arrivals at a single boundary. It describes the variability in the pattern of the cell arrival events at a boundary with reference to the negotiated peak cell rate. This variability includes the one present at the customer equipment and the cumulative effects of variability in all connection sections between the cell source and the boundary.

At a section boundary B (see figure 2.4.3), the 1–point CDV, *y(k)*, for a cell k is the difference between the cell's reference arrival time, *c(k)*, and the actual arrival time, *a(k)*:

$$y(k) = c(k) - a(k) \tag{1}$$

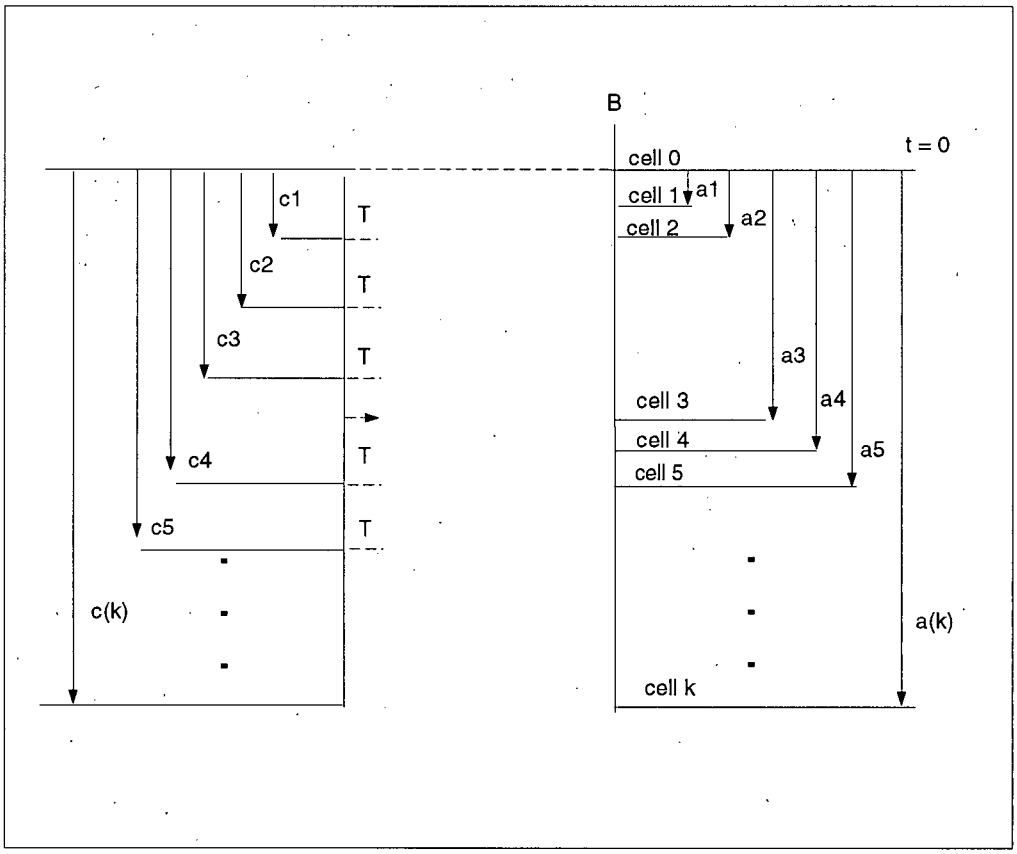

Figure 2.4.3 Cell Delay Variation: 1–Point Definition

The reference cell arrival time *c(k)* is defined as follows:

$$c(0) = a(0)c = 0$$

$$c(k+1) = \begin{cases} c(k) + T & when\ \ c(k) \geq a(k) \\ a(k) + T & otherwise \end{cases} \qquad (2)$$

An early cell arrival (positive value of the 1–point CDV) corresponds to cell clumping, and a late cell arrival (negative value of the 1–point CDV) corresponds to a gap in the cell stream.

2.4.3 2–point CDV

The definition of the 2–point CDV is based on observing the corresponding cell arrivals at two boundaries. It describes the variability in the pattern of cell arrival events at a cell

output boundary B_j with reference to the pattern of corresponding events at a cell input boundary B_i. The 2–point CDV includes only the variability introduced between the two specified boundaries.

Between the two boundaries B_i and B_j (see figure 2.4.4), the 2–point CDV, *v(k)*, for a cell *k* is the difference between the absolute cell transfer delay, *x(k)*, of cell *k* and a defined reference cell transfer delay, *d(1,2)* between the same two boundaries:

$$v(k) = x(k) - d(1,2) \tag{3}$$

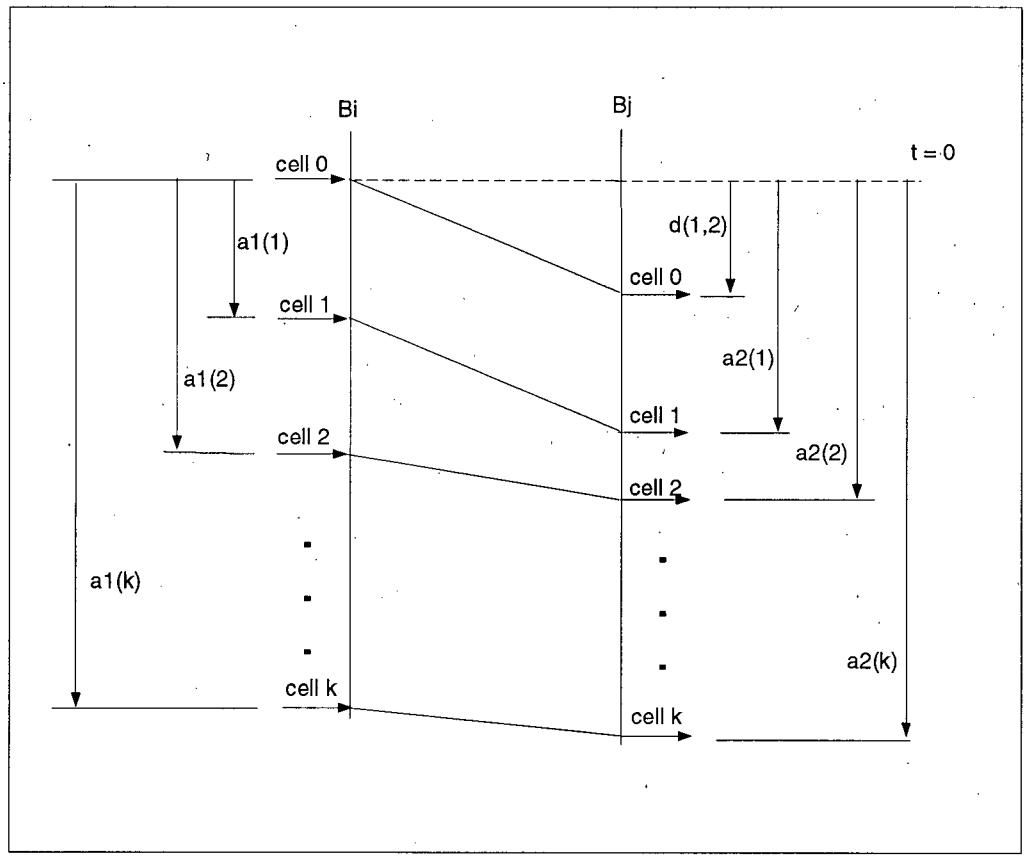

Figure 2.4.4 Cell Delay Variation: 2–Point Definition

The absolute cell transfer delay, *x(k)*, of cell k between B_i and B_j is the difference between the actual arrival time of the cell at B_j, *a2(k)*, and the actual arrival time of the cell at B_i, *a1(k)*:

$$x(k) = a2(k) - a1(k) \tag{4}$$

The reference cell transfer delay, *d(1,2)*, between two boundaries B_i and B_j is the absolute cell transfer delay experienced by cell 0 between the same boundaries:

$$d(1,2) = a2(0) - a1(0) \tag{5}$$

A positive value of a 2–point CDV corresponds to a cell transfer delay greater than that experienced by the reference cell. On the other hand, a negative value of a 2–point CDV corresponds to a cell transfer delay less than that experienced by the reference cell.

2.5 Survey of Related Work

Different techniques have been proposed in the literature to characterize the behavior of a system that can be used at the receiver side to recover the transmitter clock and remove the jitter from the incoming cell stream. The main methods are summarized below.

2.5.1 Synchronous Residual Time Stamp (SRTS)

This technique assumes that the same network reference clock is available at the source and the destination to synchronize both entities. It uses the Residual Time Stamp (RTS) to measure and transmit information about the frequency difference between the network clock and the clock used at the entity.

To give a description of this technique, following notations need to be introduced:

- *fs* is the service clock frequency.
- *fn* is the network clock frequency.
- *fnx* is the derived network clock frequency.
- *N* is the period of the RTS in cycles of *fs*.
- *T* is the period of the RTS in seconds.
- *M(Mnom, Mmax, Mmin)* is the number of *fnx* cycles within a (nominal, maximum, minimum) RTS period.
- *Mq* is largest integer smaller than or equal to *M*.

The time stamping method is shown in figure 2.5.5. The number of derived network clock cycles *Mq* is obtained at the transmitter in a fixed duration *T* measured by *N* service clock cycles. *Mq* is made up of a nominal part *Mnom* and a residual part. *Mnom* corresponds to the nominal number of *fnx* cycles in *T* seconds, and it is fixed for the service. Therefore, *Mnom* can be assumed to be available at the receiver. The residual part of *Mnom* conveys the frequency difference information and it is transmitted to the receiver. It is represented by means of the RTS, whose generation is shown in figure 2.5.6. The output of the counter is sampled every *N* service clock cycles to give a 4–bit size RTS value.

By knowing the residual value of *Mq*, i.e. the RTS value, and the nominal value of the *Mq* at the receiver, *Mq* is completely known. Then it is used to produce a reference timing signal for a phase-locked loop to obtain the proper service clock. For more details on this topic, the reader is referred to [21] and [25]. Although this method is approved by the ATM

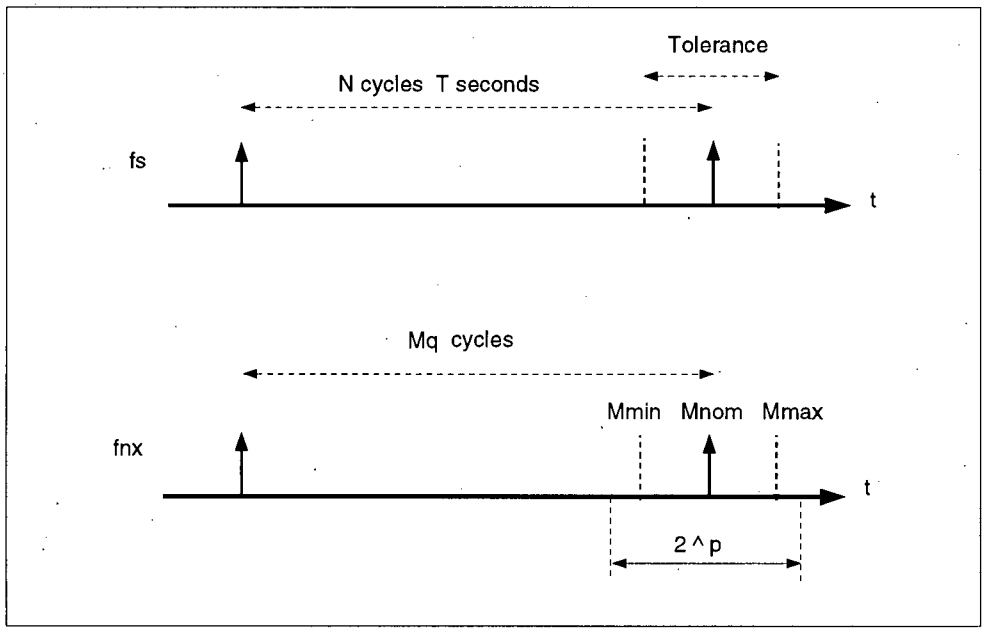

Figure 2.5.5 The Concept of Residual Time Stamp

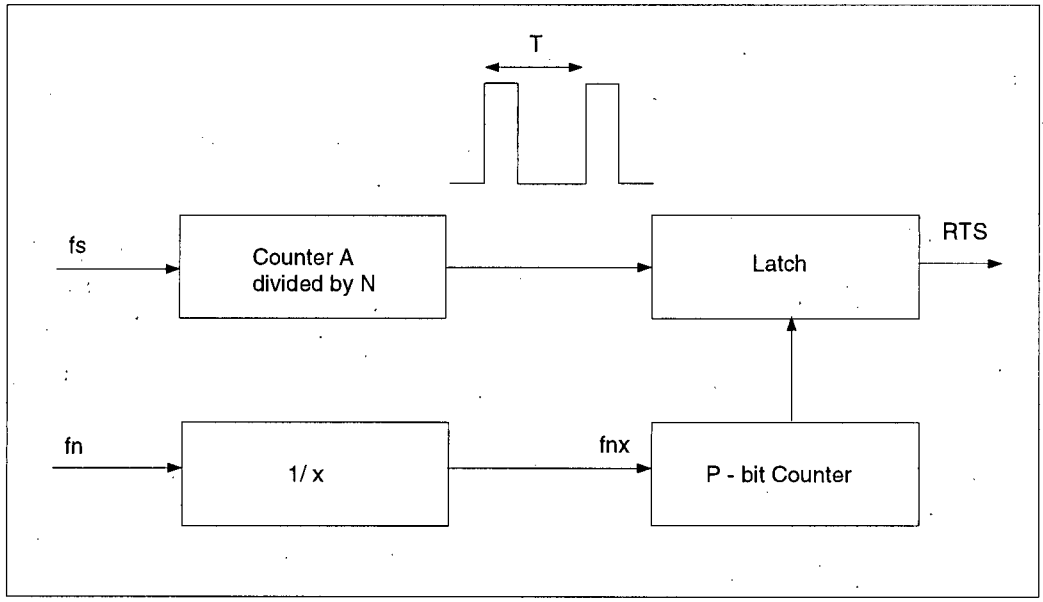

Figure 2.5.6 Generation of RTS

standardization bodies, it can not be used in plesiochronous networks where no common network clock is available to the different communicating entities.

2.5.2 Negative Retiming Stuffing

The concept of bit stuffing is used mainly with Time Division Multiplexing (TDM). In TDM, if the channel input rate is lower than the multiplexer channel rate, the buffer tends to deplete and stuff-bits are added (positive stuffing). If the input rate is higher, the buffer tends to overflow and here information bits are subtracted (negative stuffing) from the channel input and transmitted as part of the overhead channel. To positively stuff a bit, a request is made to stuff a time slot with a dummy stuff bit. At the next available opportunity, the read clock is inhibited for a single clock pulse allowing a stuff bit to be inserted in the synchronous channel while the asynchronous channel input continues to fill the buffer [29].

The negative retiming stuffing approach is divided mainly into two steps. First, signal retiming is done by adjusting the difference between the transmitter and the receiver clocks using negative stuffing. Then, a Phase-Locked Loop (PLL) is used to remove the stuffing and smooth the final output signal. A simple negative stuffing control algorithm takes place after frames M_m, M_{2m}, M_{3m}..... for mP consecutive frames where m is a large enough integer, M_m is the first frame by the end of which the $(nM_m + m)th$ packets has completely arrived, and n is the number of packets in a transmitted frame. This algorithm makes sure that there are enough intermediate frames within which negative stuffing can be done to digest the m extra packets. This scheme will transform the large and fast packet arrival jitter into small and slowly varying frequency wander due to the stuffing itself.

Another uniform algorithm depends on the average percentage (ρ) of having negative stuffing. It claims that if ρ is estimated precisely, the results will be equivalent to those

of asynchronous TDM and the large jitter will be removed. For more details, the reader is referred to [15].

Although the above algorithms do not depend on the buffer level in adjusting the destination clock frequency, it does not follow the ATM standardization committee recommendations. For example, it violates the ATM frame formats since it is assuming that extra bits are added to the frame for retiming.

2.5.3 Cell Spacing

This method proposes a solution when cells arrive too close together according to the peak emission period. This is called the clumping effect. A space controller device is used to monitor the incoming cells. If two cells arrive in a period smaller than the peak rate period, the controller ensures a minimum spacing between any two consecutive cells on a connection which is equal to the peak emission period of the connection.

This method can reduce the magnitude of clumping but does not solve the problem of cell gaps (dispersion) in which consecutive cells may have an inter-arrival period larger than the peak emission period [18] [19].

2.5.4 Buffer Occupancy

In this scheme, a plesiochronous network operation is assumed. The destination clock is adjusted according to the buffer level. This is based on the fact that the buffer level is an indication of the difference between the transmitter and the receiver clocks. The clock adjustment can be done by using a Digital Phase-locked Loop (DPLL) [10]. A simple representation of this technique is shown in figure 2.5.7.

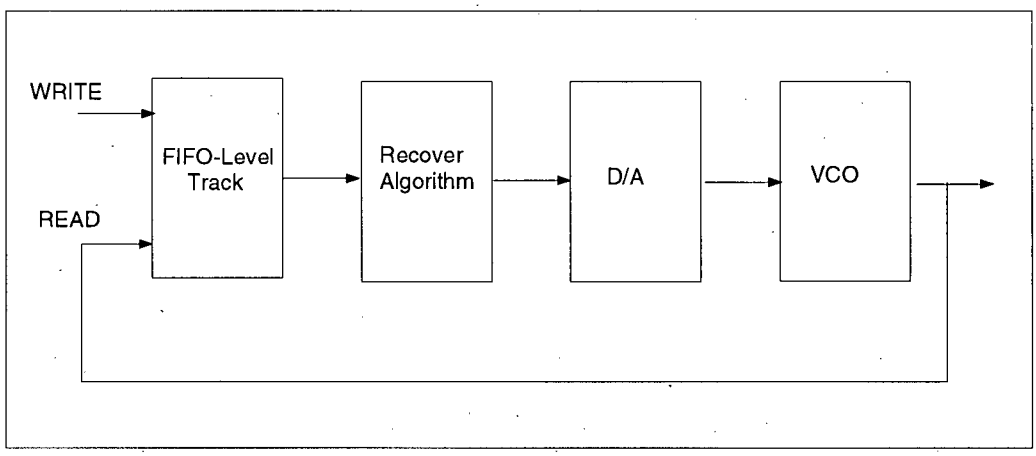

Figure 2.5.7 Functional Diagram of a Simple DPLL

The FIFO-level track (FLT) block and the recover algorithm (RA) block function as a phase detector between the Voltage Controlled Oscillator (VCO) output signal and the input signal. The FLT block keeps the current level of the destination buffer, and passes it to the RA block. The RA block depends on its history and on the current level of the buffer to produce a signal to increase or decrease the speed of the read clock. The signal passes through a Digital-to-Analog (D/A) convertor to produce the appropriate signal of the VCO to adjust the READ signal of the destination buffer.

Different versions of the clock recovery technique that are based on the buffer level at the destination exist. A more sophisticated adaptive technique will be proposed and analyzed in chapter 3. The proposed technique is an enhancement of the two-time scale algorithm of Singh, et. al. [12] which contains a fast and a slower time sequences. The faster time sequence is used by the frequency estimation algorithm, and the slower time sequence is used by the adaptive clock adjustment algorithm.

3 Adaptive Clock Recovery

In real-time CBR applications, the convergence sublayer of the AAL–1 layer at the receiver is responsible for the recovery of the transmitter clock and removing the jitter from the arriving cell stream. However, in the absence of an explicit information about the relationship between the frequencies of the transmitter and the receiver, it is very hard for the CS to do its necessary functions.

In the adaptive clock recovery scheme, the network is assumed to be plesiochronous where the clock frequency of the transmitter is transparent to the receiver. The receiver writes the incoming cells into its buffer, and then reads them out using a local read clock. The fill level of the buffer provides an indication about the difference between the transmitter and the receiver clocks, and can be used to control the frequency of the local read clock as shown in figure 3.1.

The local clock frequencies at the transmitter and the receiver are assumed to have the same nominal values, but may differ in their part per million (ppm) values [12]. This assumption is needed to carry out the analysis of the scheme, however, the simulation results show that even larger differences between the two clock frequencies can be eliminated by the receiver being able to adjust its frequency very closely to that of transmitter.

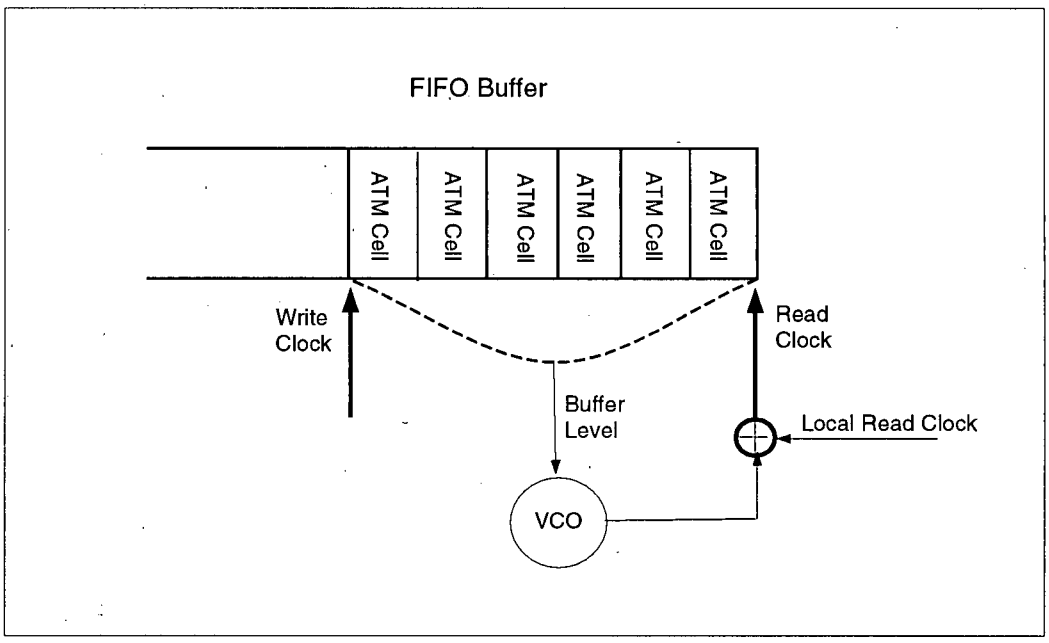

Figure 3.1 General Adaptive Clock Recovery Method

The adaptive clock recovery method proposed in this chapter is an enhancement of the method previously described by Singh, et. al. [12]. Our method exercises better buffer size control to reduce the system cost, and to minimize the delay that results from large buffering time.

3.1 Preliminaries

The rate of the cell stream arriving at the destination FIFO buffer is a measure of the source clock frequency, while the rate of the cell stream leaving the destination buffer represents the destination clock frequency. The main idea behind the adaptive clock recovery scheme is to observe the fluctuation of the buffer level over a sufficiently long period of time, then use the buffer-fill rate information to cancel the effect of jitter. The main contributor to the fluctuation in the buffer level is the frequency difference between the transmitter and

the receiver clocks. If a good estimate of this difference is available and the frequency of transmitter clock is known approximately, then the estimate can be used to adjust the receiver clock frequency.

Before introducing the estimation and adjustment algorithms, the mathematical quantities used in the analysis are defined below:

- $\phi(k)$ is the destination buffer level function at time k which indicates the filling of the buffer in the positive or negative direction from its middle zero position. In other words, $\phi(k)$ is the number of ATM cells in the destination buffer at time k.
- f_1 is the CBR source clock frequency at which the cells are generated uniformly. It is measured by the "cells per second" unit.
- T is the period between the generation of two cells, i.e. $T = 1/f_1$.
- $f_2(k)$ is the destination clock frequency at which the cells are removed from the destination buffer, and it is also measured in cells per second. To avoid the destination buffer overflow or underflow, this frequency changes with time (k) according to the adjustment algorithm.
- *T(k)* is the period of cell removal (read clock period) from the destination buffer, i.e. $T(k) = 1/f_2(k)$.
- $\Delta f(k)$ is the difference between the source and the destination clock frequencies at time k, i.e. $\Delta f(k) = f_1 - f_2(k)$.
- *n(k)* is the random fluctuation in the source clock frequency (represented by the cell stream arriving at the destination buffer) which carries the jitter characteristics. The definition of *n(k)* is similar to that of the 1–point CDV mentioned earlier.

- *d(k)* is the jitter measured in number of cells over a sampling period *T(k)*, i.e. *d(k)* = *T(k)n(k)*.
- *j(i)* is the length of an estimation time interval.
- α is the adaptation rate parameter.
- ϵ is the adjustment period control parameter.
- γ is the buffer-size control parameter.
- ρ is the buffer size limits.

The cells that traverse the network from end-to-end suffer from stochastic delay. When these cells arrive at the destination, they represent a noisy value of the source clock frequency which can be represented by $f_1 + n(k)$. The value of $\phi(k)$ fluctuates randomly depending on the frequency difference between the incoming cell stream and the outgoing cell stream from the buffer, i.e. $\phi(k)$ depends on $f_1 - f_2(k) + n(k) = \Delta f(k) + n(k)$. From the conservation law, the buffer level at time *k+1* must equal the buffer level at time *k* plus the contribution of the random frequency difference during the sampling period *T(k)*. Therefore,

$$\phi(k+1) = \phi(k) + T(k)\ (\Delta f(k) + n(k)) \tag{6}$$

which results in

$$\Delta\phi(k) = \phi(k+1) - \phi(k) = T(k)\Delta f(k) + d(k) \tag{7}$$

Modeling the cell jitter *d(k)* in an ATM network is the most difficult part in the design and analysis of the clock recovery schemes. Some preliminary analysis show that the cell jitter could be a correlated sequence which depends on the loading of the network, relative rates of the input traffic being multiplexed in comparison to the multiplexed line rate, service

disciplines at the multiplexers and the switches, the type of connection and others. Therefore, the analysis presented next will not use the statistics of jitter, instead it uses time-averaging estimation algorithm making use of the fact that the expected value of the jitter over a statistically long period is 0 and the assumption that there is no cell loss in the network.

3.2 Clock Recovery Algorithms

The random fluctuation in the buffer level depends on the difference between the source and the receiver clock frequencies, and on the cell jitter (equation (7)). However, since the frequencies share a nominal value and differ only in there ppm values, the frequency difference value $\Delta f(k)$ is very small compared to the jitter term *d(k)* which may range from 0 to few cells from one instant to another. Therefore, we can assume that the change in the buffer level $\phi(k)$ is due mainly to cell jitter *d(k)*. Therefore, to extract the value of $\Delta f(k)$, we have to eliminate the effect of *d(k)* by observing the set $\Delta\phi(k)$ over large and time-variant intervals. The larger the value of the time period, the better the estimations of the frequency difference since the cancellation of the jitter effect will be more effective. However, this would require larger buffer size, and adds more delay to the received cells. Therefore, the length of the time periods should be carefully selected and controlled.

To be able to observe the data set for $\phi(k)$ and adjust the frequency accordingly, two time scales (figure 3.2.2) are introduced:

1. A fast time sequence ($k \geq 0$) in which the observation data set $\phi(k)$ is collected. This sequence defines the sampling instants in the estimation period.

2. A slow time sequence ($m \geq 0$) in which the estimated frequency difference $\Delta\hat{f}(m)$ is computed and the destination frequency $f_2(m)$ is adjusted. This sequence defines the boundaries of the adjustment periods.

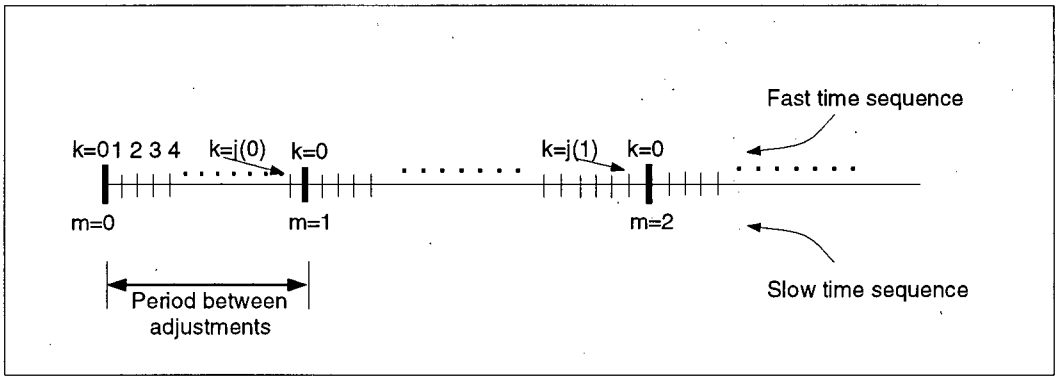

Figure 3.2.2 The Two-Time Scale Model

The length of the time period *j(m)* is time-varying. In each time period, there are a total of *j(m)+1* samples of $\phi(k)$ taken at constant times. One good sampler is the read clock frequency from the destination buffer $f_2(k)$, and therefore, the length of the interval $[k, k+1]$ equals $T(k) = \frac{1}{f_2(k)}$ which stays constant in each time period and changes only at the boundaries of the adjustment period.

The time-varying time period *j(m)* is given by

$$j(m+1) = \frac{1}{1-\alpha}\, j(m) \quad or \quad j(m) = \frac{1}{(1-\alpha)^m}\, j(0) \tag{8}$$

where $0 < \alpha < 1$.

This shows that the size of the interval increases as time progresses. However, this could not happen indefinitely. The length of the time period should be fixed after a number of iterations. The length should be long enough to produce a good estimation, and suitable for the algorithm to recover if a sudden change occurs in the source clock frequency.

The new value of the time period length found in equation (8) is chosen in a way that is related to the adaptation factor α since the value of *j(m)* is closely related to that of $\Delta f(k)$. For example, at the beginning of the algorithm and when the frequency difference is large, the time intervals should be small enough so that more frequent adjustments in the destination frequency can be taken, perhaps at the cost of poor estimate.

3.2.1 Frequency Estimation Algorithm

In this algorithm, the observation set $\phi(k)$, $0 \leq k \leq j(m)$ is processed to filter out the effects of the cell jitter and obtain the frequency difference estimate $\Delta\hat{f}(m)$ for each interval [*m, m+1*). Since CBR services are assumed, during the interval [*m, m+1*), the value of f_1 is constant (without jitter), and the value of $f_2(m)$ is also constant since the destination frequency does not change during the estimation process. Therefore, the value of $\Delta f(k)$ is constant in the interval [*m, m+1*) and the buffer state (equation (7)) can be written as

$$\Delta\phi(k) = \frac{\Delta f(m)}{f_2(m)} + d(k) \tag{9}$$

The frequency difference in an interval [*m, m+1*) can be estimated by finding the sample mean of all the available observations. This interval has a total of *j(m)* samples, and therefore, the estimated frequency difference will be:

$$\Delta\hat{f}(m) = \frac{1}{j(m)}\left[\left(\sum_{k=0}^{j(m)-1} \Delta\phi(k) f_2(m)\right) - \left(\sum_{k=0}^{j(m)-1} f_2(m) d(k)\right)\right] \tag{10}$$

Since averaging the zero mean jitter samples *(d(k))* will cancel them out, the second term of equation (10) will approach zero and the estimated difference in frequency will be:

$$\Delta\hat{f}(m) = \frac{f_2(m)}{j(m)} \sum_{k=0}^{j(m)-1} \Delta\phi(k) \tag{11}$$

To save the long computation time (subtractions and summation) needed by the cumulative sum $\sum_{k=0}^{j(m)-1} \Delta\phi(k)$ expression, another estimation is needed. This expression reflects the fluctuation in the buffer level in the estimation period. Therefore, equation (11) can be rewritten as:

$$\Delta\hat{f}(m) = \frac{f_2(m)}{j(m)}(\phi(m+1) - \phi(m)) \tag{12}$$

where $\phi(m+1)$ is the value of the destination buffer level function at the end of the estimation period just before the new estimation period starts at time *m+1*.

3.2.2 Frequency Adjustment Algorithm

The estimated frequency difference ($\Delta\hat{f}(m)$) obtained in the estimation algorithm is used as an input to the frequency adjustment algorithm to adjust the destination clock frequency. The adjustment algorithm updates the destination clock frequency at the end of every interval [*m, m+1*). A simple way for the update of the destination frequency is to add a weighted value of $\Delta\hat{f}(m)$ to $f_2(m)$ to obtain $f_2(m+1)$, i.e.

$$f_2(m+1) = f_2(m) + \alpha\Delta\hat{f}(m) \tag{13}$$

for some $0 < \alpha < 1$.

Substituting the value of $\Delta\hat{f}(m)$ from equation (12) into the adjustment formula of equation (13), results in

$$f_2(m+1) = \frac{f_2(m)}{j(m)}[j(m) + \alpha(\phi(m+1) - \phi(m))], \tag{14}$$

when expressed in terms of time periods, the above equation will be

$$T2(m+1) = T2(m)\frac{j(m)}{j(m) + \alpha((\phi(m+1) - \phi(m)))} \tag{15}$$

The above equations show that the adjusted clock frequency at the destination depends mainly on the buffer level fluctuation which is represented by the difference between the buffer levels at the beginning and the end of the estimation period. Equation (15) clearly shows that if the buffer level increases during an estimation period (i.e. $\Delta\phi(m) = \phi(m+1) - \phi(m)$ is positive), the updated clock period will become smaller, which means a higher read clock frequency. This is the desired response since the increase in the buffer level indicates a slower read frequency at the destination buffer which should be increased to match that of the transmitter clock. Similarly, if the buffer level decreases during the estimation period (i.e. $\Delta\phi(m)$ is negative), the updated clock period will be larger resulting in a decrease in the read clock frequency at the destination. This situation occurs when the destination clock frequency is faster than that of the transmitter which will eventually deplete the destination buffer. To prevent buffer depletion (or underflow), the read clock should be slowed down so as to make it as close as possible to the transmitter clock.

3.2.3 Design Parameters

The estimation and adjustment algorithms are characterized by a number of parameters which control the clock adaptation rate, estimation period length, and the maximum buffer size. Such design parameters must be well defined to ensure the smooth operation. The following are the parameters suggested for the design:

1. Adaptation rate parameter (α): this parameter is used to control the size of the estimation time interval and how quick is the response of the system. From equation (8), one can see that as time progresses, $\Delta f(m)$ diminishes exponentially at a rate of $(1-\alpha)$. Moreover, larger α implies:

— faster response (convergence of $\Delta f(m)$ to 0)

— higher residual variance (fluctuation in the value of $f_2(m+1)$)

This is because the value of $f_2(m+1)$ can be determined quickly at the expense of not having accurate results, and higher residual variance. Finally, the value of α depends on many factors such as the jitter statistics (which are not taken care of in the previous analysis), and the nominal destination clock frequency.

2. Estimation period control parameter (ϵ): After a certain time period, the length of the estimation period *j(m)* should become constant. This happens when the estimated frequency difference $\Delta\hat{f}(m)$ is smaller than the pre-defined parameter ϵ whose unit is cells per seconds. Therefore, if $\left|\Delta\hat{f}(m_i)\right| \leq \varepsilon$, we set $j(m) = j(m+1)$ for all $m \geq m_i$.
3. Buffer size control parameter (γ) whose unit is cells. This parameter will be discussed in detail in section 3.2.4.
4. Buffer size limits (ρ): Sometimes we have to reset the entire algorithm during its operation because the clocks at the transmitting nodes are switched or replaced. Therefore, we set limits on the buffer size ($\pm\rho$) depending on different factors such as the maximum expected fluctuation in $\phi(k)$ due to jitter, and the expected values of $j(m), \Delta f(m)$, and $f_2(m)$. So, anytime $|\phi(k)| \geq \rho$, we reset the algorithm.

3.2.4 Buffer Size Control

Even though the analytical results show that the read clock frequency of the destination buffer adjusts according to the fluctuation in the buffer level, experimental data show that the destination buffer can grow to a large size even under relatively mild cell jitter. Therefore, in addition to the average change in the buffer level during an estimation period, the updated

read frequency depends on the current level of the buffer. The parameter γ is introduced to monitor the buffer level and its average change, and tries to keep the buffer level steady. The parameter γ is introduced to the adjustment equation (15) which results in

$$T2(m+1) = T2(m)\frac{j(m)}{j(m) + \alpha[(\phi(m+1) - \phi(m)) + \gamma]} \tag{16}$$

The parameter γ (whose unit is cells) changes between a negative value and a positive value to adjust the read frequency of the buffer depending on the buffer fill-level. Since it is very desirable to minimize the cost of the buffer by minimizing its size, we should try to keep as few cells as possible in the buffer. If the buffer level increases above a certain threshold and stays at that level, then this indicates that the read frequency is relatively steady but unable to decrease the buffer level. Therefore, parameter γ takes a positive value showing that the buffer level is increasing and the read clock frequency is forced to increases and read the extra cells. This will reduce the buffer size, and the buffering delay suffered by the arriving cells at the destination.

Similarly, if the buffer level is going below a certain threshold, parameter γ takes a negative value indicating that the buffer level is decreasing. Therefore, parameter γ forces the read clock of the destination buffer to slow down. The threshold which controls the parameter γ should be defined beforehand. An ideal value of the desired buffer level threshold should be a little larger than one (5 cells for example) so that the system will be able to respond and adjust the read clock before its buffer depletes or overflows.

3.3 Chapter Summary

In this chapter, the adaptive clock recovery technique was introduced. The technique

consists of two main algorithms, one for estimating the difference between the transmitter and the receiver clock frequencies, and the other for adjusting the receiver read clock frequency. The parameters that are used to control the operation of the algorithms were also identified. The analysis presented shows that this scheme will able to recover the transmitter clock and remove the jitter from the arriving cell stream at the receiver.

4 System Design and Specifications

Considering the multirate behavior of ATM networks, there is a need to support multirate CBR services using a single unit. Our goal is to develop a clock recovery and jitter control unit that operates over a wide range of rates such as the 1.55 Mbps DS1, the 44.736 DS3, and other possible rates.

The approach adopted for building the clock-recovery unit is based on developing a top-down design of the model. Initially, a state diagram is used to specify the model by splitting its operation into three interacting processes. Then, flowcharts are used to describe the processes and show the flow of data and the control signal generation in the system. The behavioral architecture of the model is specified using the VHDL hardware description language, and system behavior is tested and simulated using different CAD tools such as Leapfrog^{TM} from Cadence [32], and Synopsys [1]. In this chapter, the specification of the system that performs the transmitter clock recovery and jitter removal from the incoming cell stream at the receiver will be presented, and the full behavioral description of the system hardware will be given.

4.1 Model

The proposed clock-recovery and jitter-control system consists mainly of three interacting processes: ATM cell Detection Queuing and Dequeuing (DQD) process, Length of Estimation Period (LEP) process, and the Frequency Estimation and Adjustment (FEA) process. These processes run concurrently and interactively as shown in figure 4.1.1.

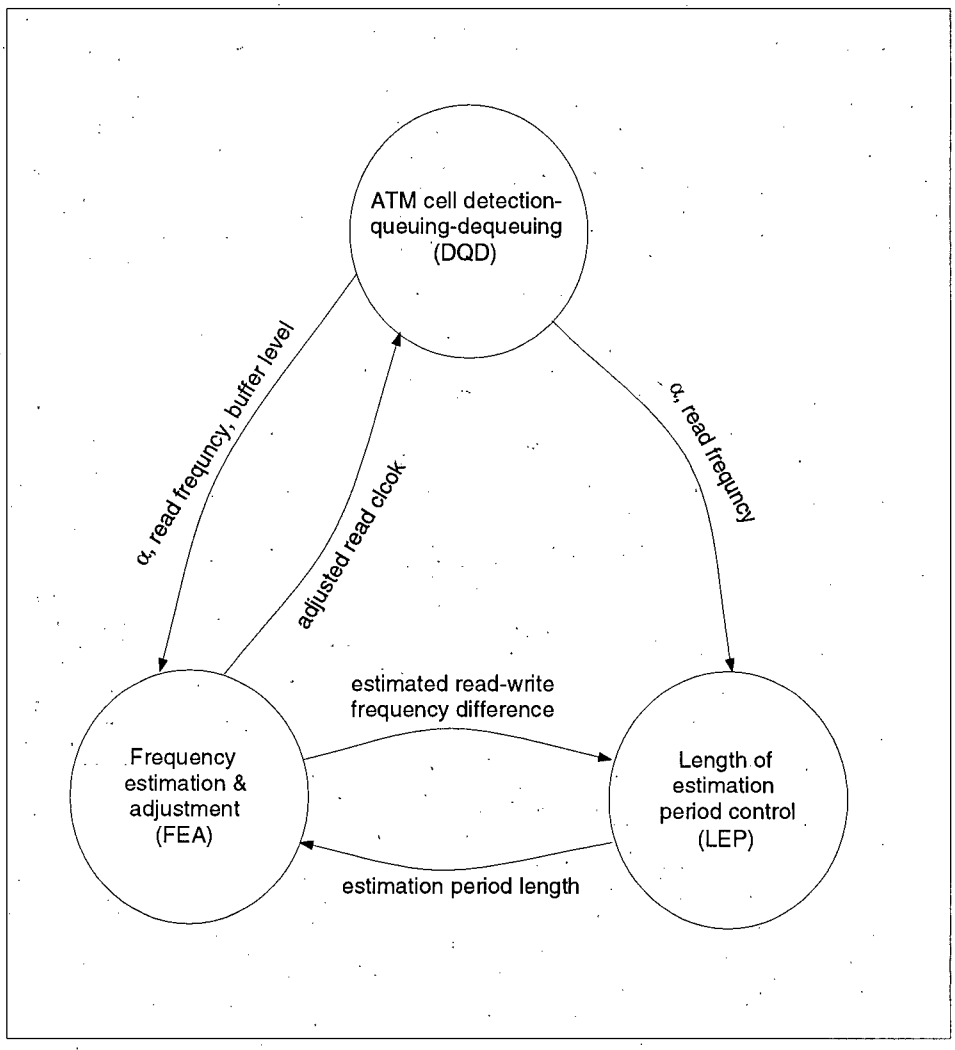

Figure 4.1.1 Processes of the Clock Recovery and Jitter Control System

The DQD process removes the jitter from the received jittered cell stream and produces a cell stream with a cell inter-arrival period very close to the cell inter-departure period at the transmitter. The LEP process indicates the times at which the adjustment algorithm takes action, and calculates the time periods during which the estimation algorithm is updated. The FEA process produces the updated read clock frequency of the destination buffer to eliminate cell jitter that results from the ATM network.

The DQD process receives the jittered ATM cell stream from the network, and outputs the jitter-free cell stream with a cell inter-arrival period very similar to that of the original cell stream. The flowchart of the DQD process is shown in figure 4.1.2. This process consists of two concurrent sub-processes running asynchronously, one for detecting and buffering (or queuing) new incoming cells, and the other is for reading (or dequeuing) the buffered cells out of the buffer. These operations are supported by an asynchronous FIFO buffer that allows concurrent writing and reading into and from the buffer, respectively.

Referring to figure 4.1.2, in the cell detection and buffering sub-process, whenever a new ATM cell arrival at the receiver is detected and the buffer is not full, the cell is inserted into the FIFO buffer and its level is incremented by one. In the cell dequeuing sub-process, if it is the time for a new cell to be read out of the buffer (as indicated by $f_2(m)$) and the buffer is not empty, then the cell is dequeued from the buffer and the buffer level is decremented by one. This process employs the updated value of the read frequency ($f_2(m)$) which is generated by the FEA process to be described later.

The LEP process performs two important tasks:

1. Indicating the times at which the adjustment algorithm (operating at the slower time sequence) takes action.
2. Calculating the time periods during which the estimation algorithm (operating at the faster time sequence) is updated.

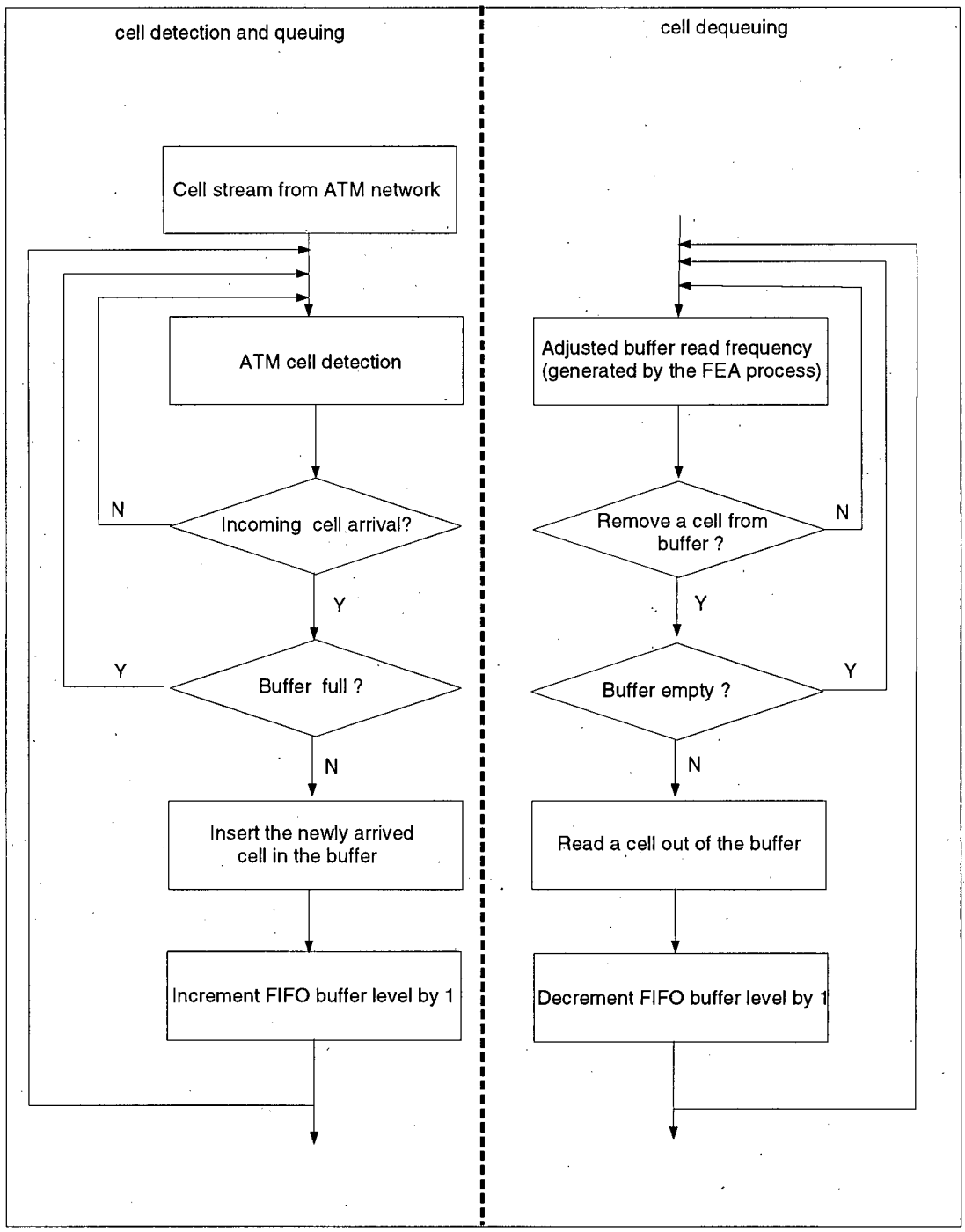

Figure 4.1.2 ATM Cell Detection, Queuing, and Dequeuing

The main functions of this process are shown as a flowchart in figure 4.1.3. When the system is initialized, the process starts with a pre-specified value for the estimation period length, *j(0)*. A counter (EPC) is used to keep track of the length of the estimation period (*j(m)*). The EPC is updated by the read-clock of the destination buffer ($f_2(m)$), and it is incremented whenever a cell is read out of the buffer. When the value of the EPC reaches the given estimation period length (*j(m)*), a signal is sent to the FEA process to end the current estimation algorithm, and begin the adjustment algorithm and a new estimation period.

A decision should be made on whether to adjust the length of the next estimation period or keep it the same as the current one. This depends on how stable the system is, a quality measured by the value of the estimated frequency difference ($\Delta\hat{f}(m)$) between the transmitter and the receiver clocks (from the FEA process). If this difference is sufficiently small (i.e. $\left|\Delta\hat{f}(m_1)\right| \leq \varepsilon$ cells/sec), the system is considered to be stable, and consequently, the length of the current estimation period will be used for the next estimation period. Otherwise, a new value for the length of the estimation period should be calculated according to equation (8) in section 3.2.

The following signals are required by LEP process:

1. The current read frequency ($f_2(m)$, where m is the index of the slower time sequence) from the DQD process is used to control (mainly increment) the EPC.
2. The estimated frequency difference ($\Delta\hat{f}(m)$) from the FEA process is employed to keep the length of adjustment period constant after a certain number of iterations.
3. The value of the clock adaptation rate, α, from the DQD process is needed to control the length of the individual estimation time periods (refer to equation (8) in section 3.2).

The FEA process is controlled by the length of the estimation period whose value is passed to this process by the LEP process. The main steps of this process are illustrated in figure 4.1.4. At the beginning of the estimation period, the buffer-level value ($\phi(m)$) is stored in a register to compare its value to that of the buffer level at the end of the estimation period ($\phi(m+1)$). The difference between these two values ($\phi(m+1) - \phi(m)$) represents the average change in the buffer level during the current estimation period, and it is denoted in the flowchart by Avg_buf. This provides a good estimation of the difference $\Delta\hat{f}(m)$) between the transmitter and the receiver clock frequencies, f_1 and f_2, respectively.

Before adjusting the read frequency of the destination buffer, a decision is made about its size. If the buffer size stays large (i.e. above a certain threshold ϕ_{th1}) and the average change in its level is small (i.e. $|\phi(m+1) - \phi(m)| \leq \phi_{th2}$), then this indicates that the read and write clock frequencies are very close to each other and the read frequency is not able to reduce the level of the buffer. An increased buffer-size implies more delay added to the cells arriving at the destination as well as increased buffer cost. If the above situation is encountered, the parameter γ (default value is zero) is assigned a positive value (for example, two cells) to force an increase in the read clock frequency of the buffer.

On the other hand, if the buffer level becomes lower than the threshold (i.e., $\phi(m) < \phi_{th1}$), the system detects that the read clock is faster than the write clock. Therefore, γ is assigned a negative value (e.g. -2 cells) to force the read clock to slow down. Then, the FEA process adjusts the read clock frequency of the buffer (according to equation (16) of section 3.2.4) so as to control the buffer level while trying to keep the value of f_2 as close as possible to that of the write clock frequency of the transmitter (f_1). Therefore, the FEA

process will eliminate cell jitter that results from the network. This process requires the values of the adaptation rate parameter (α), the buffer level ($\phi(m)$) , and the current read frequency ($f_2(m)$) to be passed from the DQD process to govern the rate of clock adaptation.

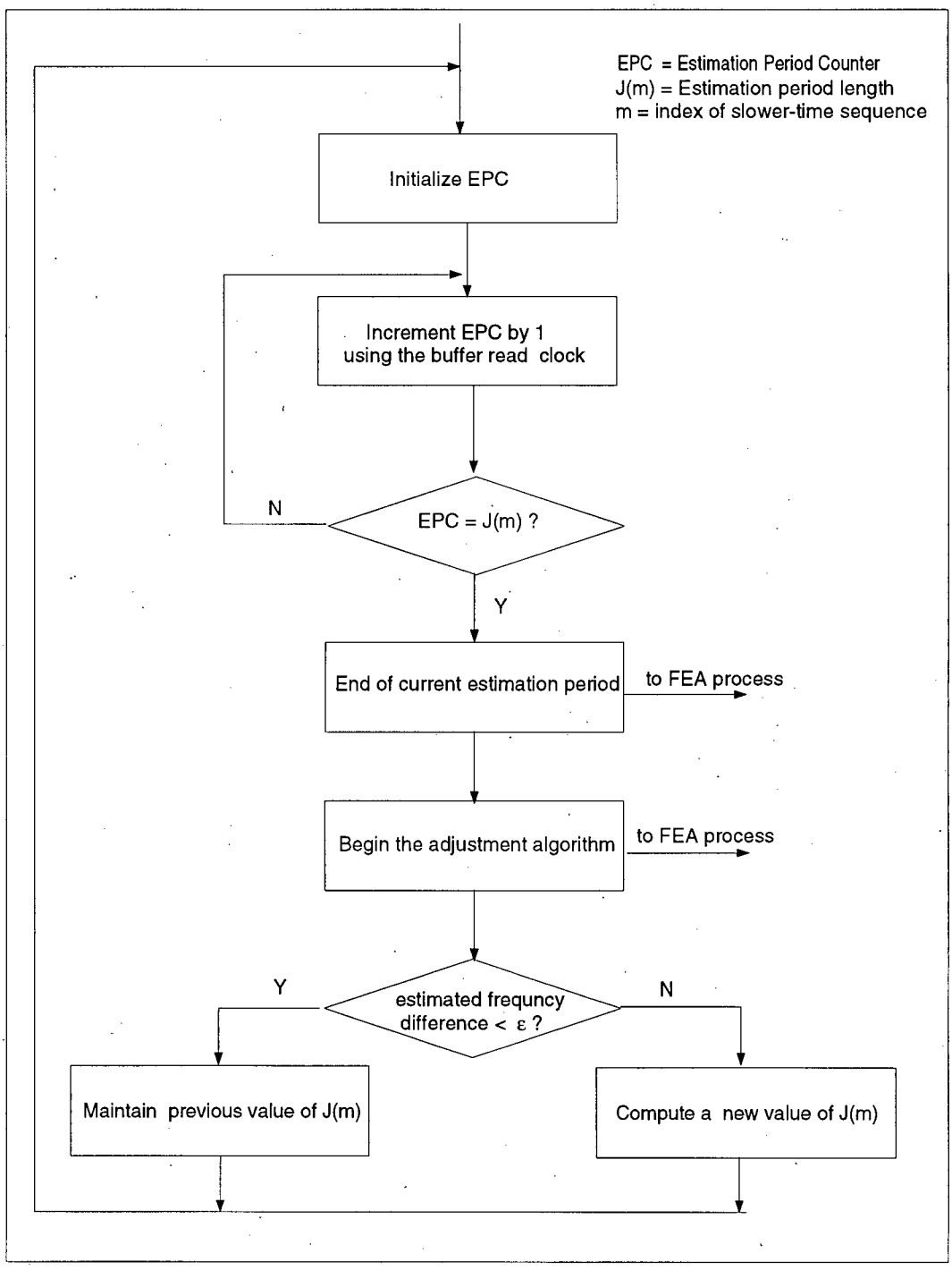

Figure 4.1.3 Length of Estimation Period Control

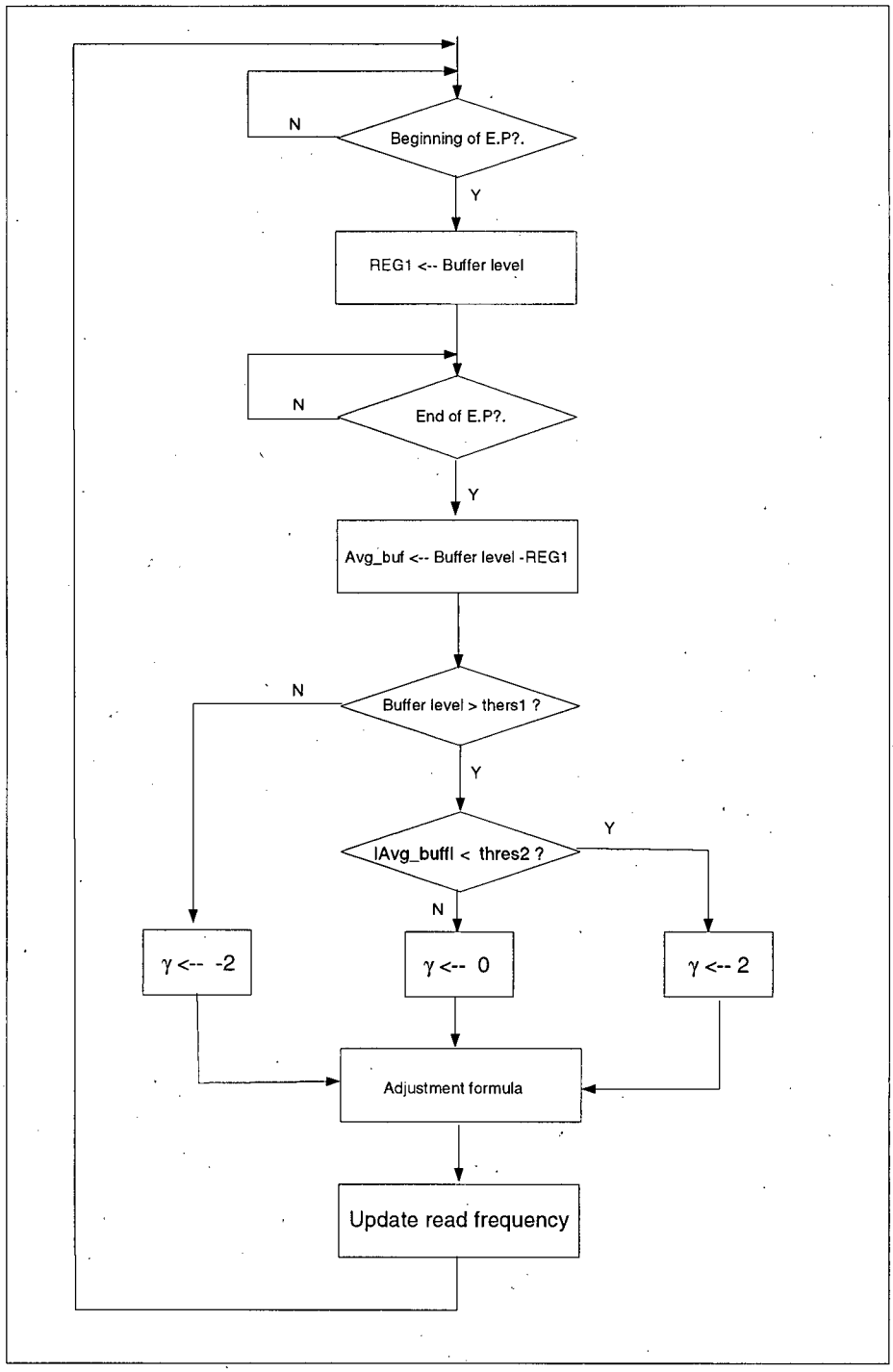

Figure 4.1.4 Frequency Estimation and Adjustment

4.2 Hardware Implementation

The block diagram of the clock recovery and jitter control system is shown in figure 4.2.5. The basic blocks are specified and implemented in VHDL at the behavioral level, and they represent the three processes that constitute the system (see figure 4.1.1). The following sections provide implementation details of the basic building blocks. VHDL related issues are discussed at the end of the chapter.

4.2.1 Cell Detection, Queuing, and Dequeuing Block

The cell detection, queuing, and dequeuing process consists of four main hardware blocks, the asynchronous FIFO buffer, the cell arrival detection block, the read clock generator, and the up-down counter. The cell detection block monitors the incoming cell stream. When a cell with a destination address that matches the receiver address is detected, the New-cell signal is generated indicating a new cell arrival. If the FIFO buffer is not full, it is enabled to receive the new cell. The description of the cell detection and delineation can be found in ITU-T Recommendation I.432 [22]. The cell detection circuitry is normally implemented at the physical layer (ATM/SONET interface) and the ATM layer (ATM header removal). However, we have simulated the output of this block using a cell traffic generator that produces jittered cell inter-arrival periods. The traffic generator will be described in detail in the next chapter.

The New-cell signal is used to increment the up-down counter to indicate that the buffer level, or the number of cells in the buffer, has increased by one. Therefore, the value of the up-down counter continuously tracks the buffer level that will be used in the frequency estimation and adjustment process.

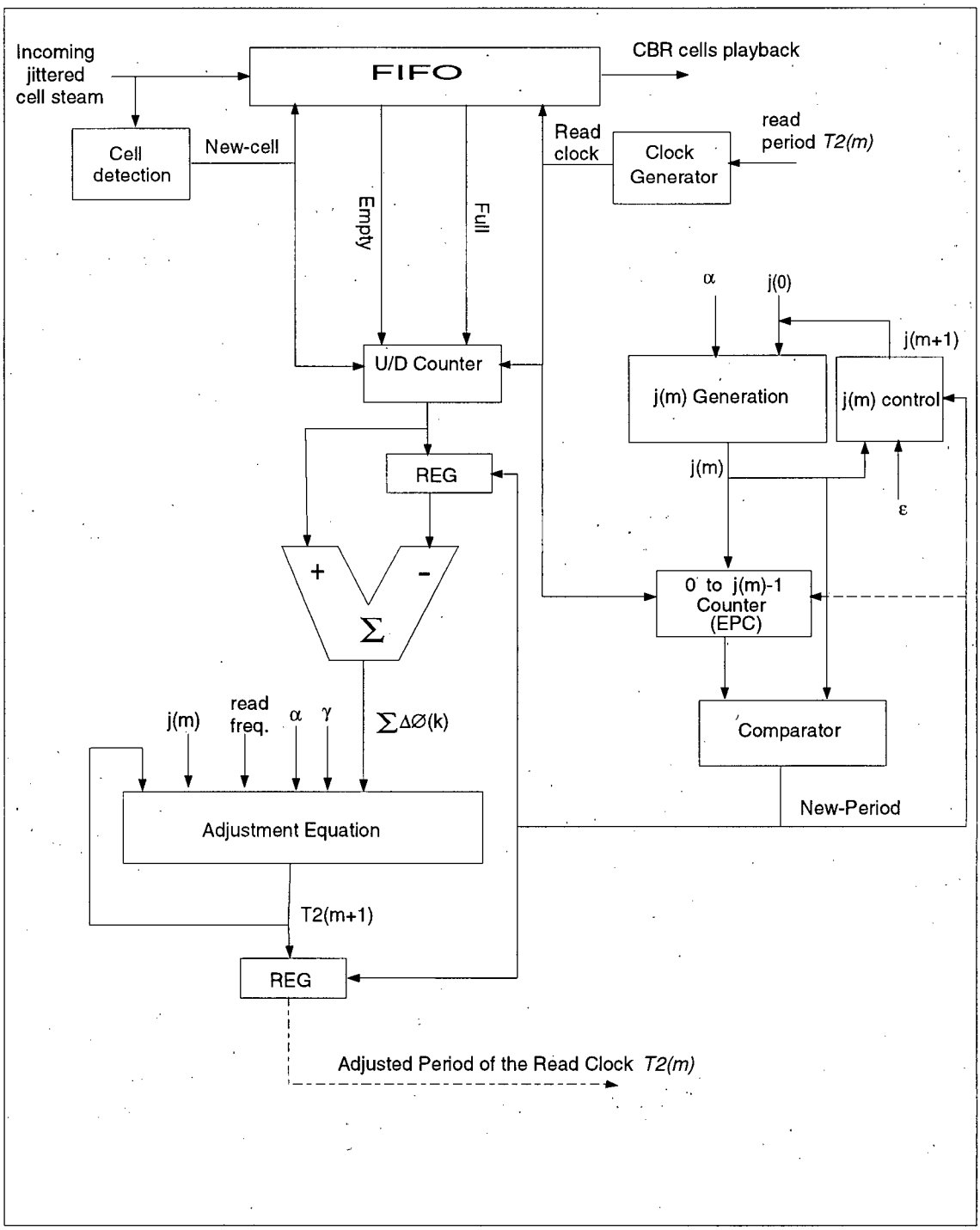

Figure 4.2.5 Block Diagram of the Adaptive Clock Recovery and Jitter Control Scheme

The read clock generator receives the updated period from the FEA process to produce the clock which is used to read the cells from the buffer. The final implementation may involve a phase lock loop if necessary to keep the read clock frequency locked to its stable value. The cell stream read out of the buffer should have an inter-arrival period very close to that of the transmitter. Therefore, the CBR cell stream will be restored. Whenever a cell is read out of the buffer, the up-down counter is decremented by one to update the value of the buffer level.

The buffer used to store the new arriving cells at the receiver is an asynchronous FIFO buffer. The design of the buffer is based on a logical circular queue, with two pointers keeping track of the positions to insert a new incoming cell, and to read a cell from the buffer. This buffer can be implemented using a static RAM. We assume that this process generates the initial value of α which is passed to the other two processes.

4.2.2 Estimation-Period Length Control Block

This process produces the signal that controls the beginning and the end of the estimation periods, and the times at which action is taken to adjust the read frequency of the destination buffer. This block consists of four main components, the estimation period length *j(m)* generator, the EPC counter, the *j(m)* controller, and a comparator. The initial value of the estimation period length *j(0)* is predefined for this process. The estimation period length is a multiple of the number of the cells read out of the destination buffer. The *j(m)* generation block calculates the value of the next estimation period length according to equation (8). If a faster response of the system is required (i.e. α is large), the length of the estimation period will be short enough to enable the system to take rapid adjustment actions.

The counter is used to keep track of the estimation period length. It is incremented by the read clock of the buffer when a cell has been read out of the destination buffer. Whenever the value of the counter becomes equal to the given estimation period length, a signal called New-Period is generated by the comparator. The positive edge of the New-Period signal indicates the end of the current estimation period and the time to take the adjustment actions, and the negative edge of the signal indicates the beginning of a new estimation period.

The *j(m)* control block controls the number of iterations after which the value of the estimation period length becomes fixed. This is based on the parameter ϵ. If the estimated difference between the transmitter and receiver frequencies is sufficiently small ($\left|\Delta\hat{f}(m_i)\right| \leq \varepsilon$), the length of the estimation period stabilizes so that it will not increase indefinitely. Therefore, if there is a sudden change in the transmitter frequency (which does not cause the resetting of the whole algorithm), the receiver frequency will be able to adjust its value in a reasonable time.

4.2.3 Frequency Estimation and Adjustment Control Block

The frequency estimation and adjustment control block consists mainly of latches, a subtracter, and a controller that implements the adjustment algorithm. The subtracter and its input latch are used to compute the difference $(\phi(m+1) - \phi(m))$ which represents the fluctuation in the buffer level in the observation (estimation) period. The latch maintains the first value of the buffer level, $\phi(m)$, at the beginning of the estimation period (at the falling edge of the New-Period signal). The subtracter calculates the difference between the values of the current buffer level and the initial buffer level. To simplify the implementation, only the last value of the buffer level read just before the new estimation period starts (specifically,

the value $\phi(m+1))$ is recorded and used in finding the adjusted read clock frequency. This is taken care of by the latch at the output of the adjustment block which is enabled at the end of the current estimation period (the rising edge of the New-Period signal). Therefore, the correct value of $(\phi(m+1) - \phi(m))$ will be available.

The adjustment block basically realizes equations (12) and (16). This block calculates the estimated frequency difference between the source and the destination, and also computes the adjusted read clock period of the destination buffer. The calculation is done whenever a change happens in the buffer level. This will help in getting a better approximation of the read clock frequency at the expense of extra computational time. The register at the output of the adjustment block ensures that only the updated frequency is passed to the cell DQD process to read the cells out of the receiver buffer. This is done at the end of the estimation period when the adjustment action is required.

4.2.4 Design using VHDL

The specification and implementation of the adaptive clock recovery and jitter control circuitry are done in VHDL. VHDL is a hardware description language capable of specifying digital hardware at multiple levels of representation ranging from highly abstract behavioral descriptions to low-level structural descriptions. Therefore, VHDL provides a very useful system design tool since, by using a high level of abstraction for the device descriptions, the designer can deal efficiently with more complex applications, and with complete mastery of the technology. Moreover, VHDL can be used to perform different tasks of the design process such as:

1. System Specification: done by stating the requirements of a system in an unambiguous manner including functional and timing constraints.
2. Simulation: done by executing a system description to check its timing behavior.
3. Design Synthesis: done by translating the design from behavioral description to structural description where each element represents a predefined electronic source (basically creating a netlist).
4. Verification: done by comparing two descriptions at different levels of abstractions.
5. System documentation: done by describing a system in an informal way.

One of the most important features of VHDL is that it is becoming the standard which is portable between different platforms, and is used almost everywhere in the commercial and academic fields.

Figure 4.2.6 is a sample output of the adaptive clock recovery and jitter control system generated by Leapfrog^{TM}, a CAD tool from Cadence [32]. The signals shown are selected to illustrate the following outputs:

1. New_Cell is the signal produced from the jittered cell stream arriving at the receiver. It enables the buffer to insert a new cell.
2. Read_Clk is the smoothed read signal that is produced by the hardware system after jitter removal and it should be as close as possible to the original clock of the transmitter.
3. Full_Sig and Empty_Sig indicate if the buffer is full or empty.
4. *J(m)* shows the length of the estimation period in terms of the number of cells read out of the destination buffer.

5. New_Period indicates the times at which the estimation and the adjustment algorithms take action.
6. R_T2_Plus shows the period of the read clock in *ns* as it adjusts to its steady state.

4.3 Chapter Summary

This chapter introduced the hardware design and specifications of the system that handles clock recovery and provides jitter control for ATM CBR services. The main processes used have been identified and described in detail. Also, a block diagram was presented to show the various hardware components that generate the control signals for the system. Finally, some aspects of the VHDL implementation and simulation were discussed.

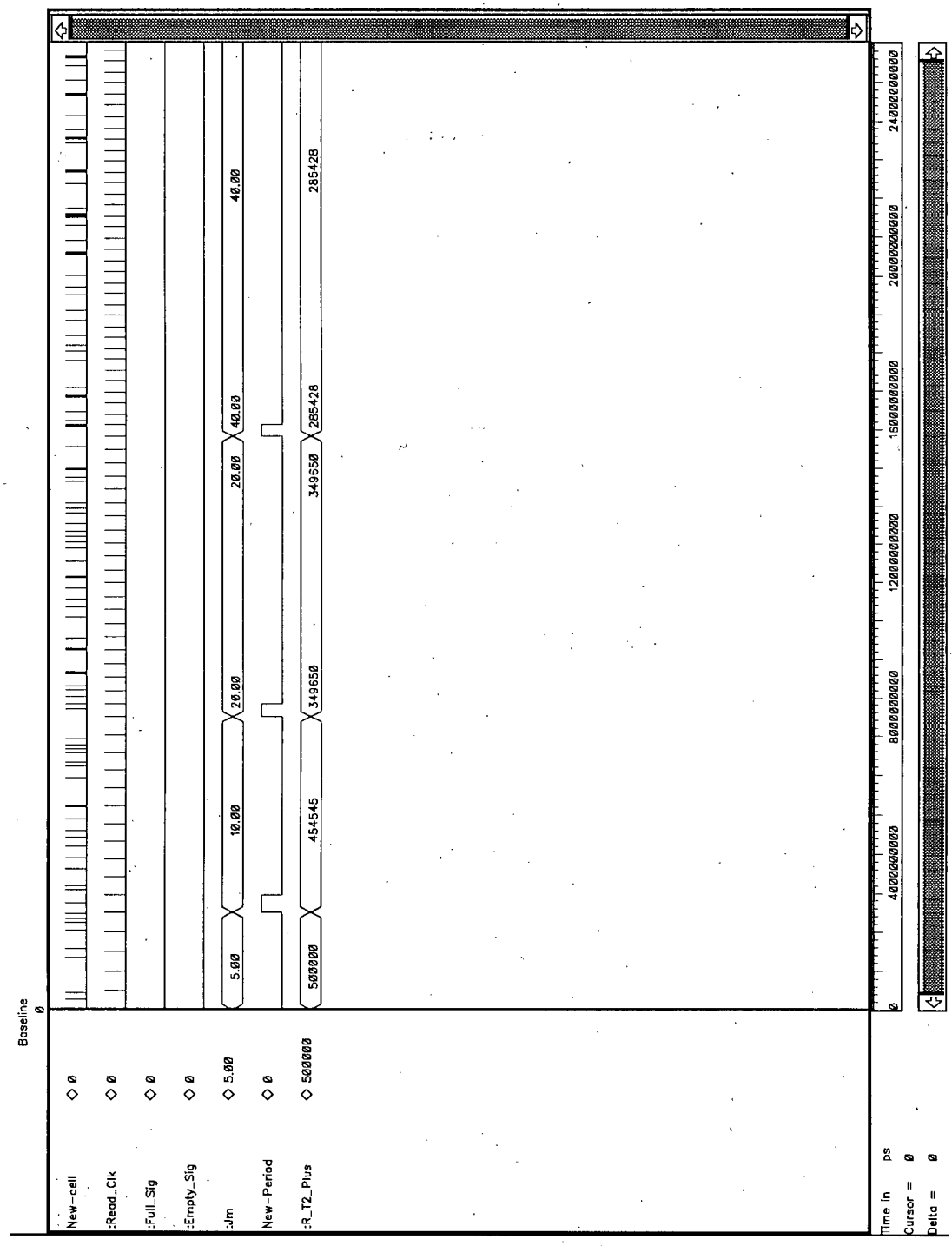

Figure 4.2.6 Sample Leapfrog™ Simulation Output

5 Performance and Simulation Results

5.1 Traffic Generator

Simulation is an essential tool in understanding and evaluating hardware units operating in a statistical environment such as ATM. To simulate our clock-recovery scheme in an ATM network environment, a number of network parameters and related performance guarantees must be taken into account. The sources of cell delay variation can be modeled in a number of ways. One way is to use a detailed queuing model whereby ATM nodes are modeled by a network of queues [8] [9] [14]. However, for the purpose of testing the proposed hardware scheme, it suffices to base our simulation on worst-case end-to-end network behavior. The main rationale in this case is that if the hardware unit is successful in recovering jitter under worst case traffic conditions, then it will be equally or even more successful with smoother cell traffic. In this chapter, we derive a simple yet effective input traffic generator for jittered CBR traffic which is used in our simulation model to evaluate the performance of the proposed hardware.

5.1.1 Network Parameters

The traffic generator uses some network parameters as variables that can be changed according to the transmission session scenario. These parameters are:

1. d_{max} is the maximum delay that a cell can encounter from the time it is transmitted to the time it reaches the receiver. If the delay suffered by a certain cell exceeds this value, the receiver assumes that the transmitted cell is lost and a dummy cell is inserted

to compensate for the time slot occupied by the original cell. Note that d_{max} is a performance guarantee parameter that may depend on many factors such as the number of switching hops between the source and the destination, traffic intensity, etc.

2. d_{min} is the minimum delay that a cell can suffer from the time it is transmitted to the time it reaches the receiver. If a cell delay is less than the minimum delay, then it could be a misinserted cell (due to an error in the cell header) that has to be discarded. d_{min} represents the best case delay scenario in which a cell does not encounter any contention or buffering delays, and where most of the delay is due to transmission links.
3. Transmission (inter-departure) period is the time elapsed between the transmission of two consecutive ATM cells. For the *peak rate* (in bits per second), the transmission time T is:

$$T = \frac{48 * 8}{Peak\ Rate} \quad (seconds) \tag{17}$$

This is because the AAL layer receives 48 bytes of the cell (after removing the ATM header).

4. $d(i)$ is the random end-to-end delay of the ith cell.

5.1.2 Traffic Modeling

The traffic generator produces cell inter-arrival periods in a cell stream. The changing inter-arrival periods (jitter) is due to the random delay each cell encounters while passing through the network. Since we assume that there is no cell loss, the end-to-end random delay for the cells generated should satisfy certain minimum conditions:

1. The end-to-end delay of any given cell i should not exceed the maximum delay, i.e.

$$d(i) \leq d_{max} \tag{18}$$

2. The end-to-end delay of the cell should not be less than the minimum end-to-end network delay, i.e.

$$d(i) \geq d_{min} \tag{19}$$

3. The cells should arrive in the right sequence (an assumption guaranteed by the ATM standards). This means that the arrival time of cell *i*, denoted by $a(i)$, should be larger than the arrival time of cell *i-1*, i.e.

$$a(i) > a(i-1) \tag{20}$$

The transmission time of the *i*th cell, denoted by *s(i)*, is generated by multiplying the cell sequence number, *i*, by the inter-departure period *T*. The arrival time of the cell *i* equals the transmission time of the cell, *s(i)*, added to the end-to-end delay experienced by this cell, i.e.

$$a(i) = s(i) + d(i) \tag{21}$$

Equations (20) and (21) imply:

$$d(i) + s(i) > a(i-1) \tag{22}$$

In other words,

$$d(i) > a(i-1) - s(i) \tag{23}$$

Therefore, the random end-to-end delay assigned to a cell by the traffic generator must satisfy equations (18), (19), and (23). The above equations can be combined to provide upper and lower bounds on the end-to-end delay of cell i as follows,

$$max\{d_{min}, a(i-1) - s(i)\} < d(i) \leq d_{max} \tag{24}$$

Thus, the legal period (LP) during which the ith cell arrival should occur is

$$LP = d_{max} - max\{d_{min}, a(i-1) - s(i)\} \tag{25}$$

In this case, the end-to-end delay encountered by cell i is given by:

$$d(i) = max\{d_{min}, a(i-1) - s(i)\} + r(i).LP \tag{26}$$

where $0 < r(i) < 1$ is a random number representing the additional delay of the ith cell. As a result, the arrival time of the ith cell will be computed according to equation (21). From equations (24) and (26), we can see that the delay of the ith cell is uniformly distributed between $max\{d_{min}, a(i-1) - s(i)\}$ and d_{max}. The inter-arrival period (*IP(i)*) between two consecutive cells, i and *i-1* will be

$$IP(i) = a(i) - a(i-1) \tag{27}$$

Equation (27) is used to generate the jittered inter-arrival periods between the cells, which represents the input to our hardware unit.

5.1.3 Inter-Arrival Period Generation

Many applications use DS1 (1.55 Mbps) as the transmission link. Therefore, in the simulations presented next, the parameters of the traffic model are chosen to simulate a

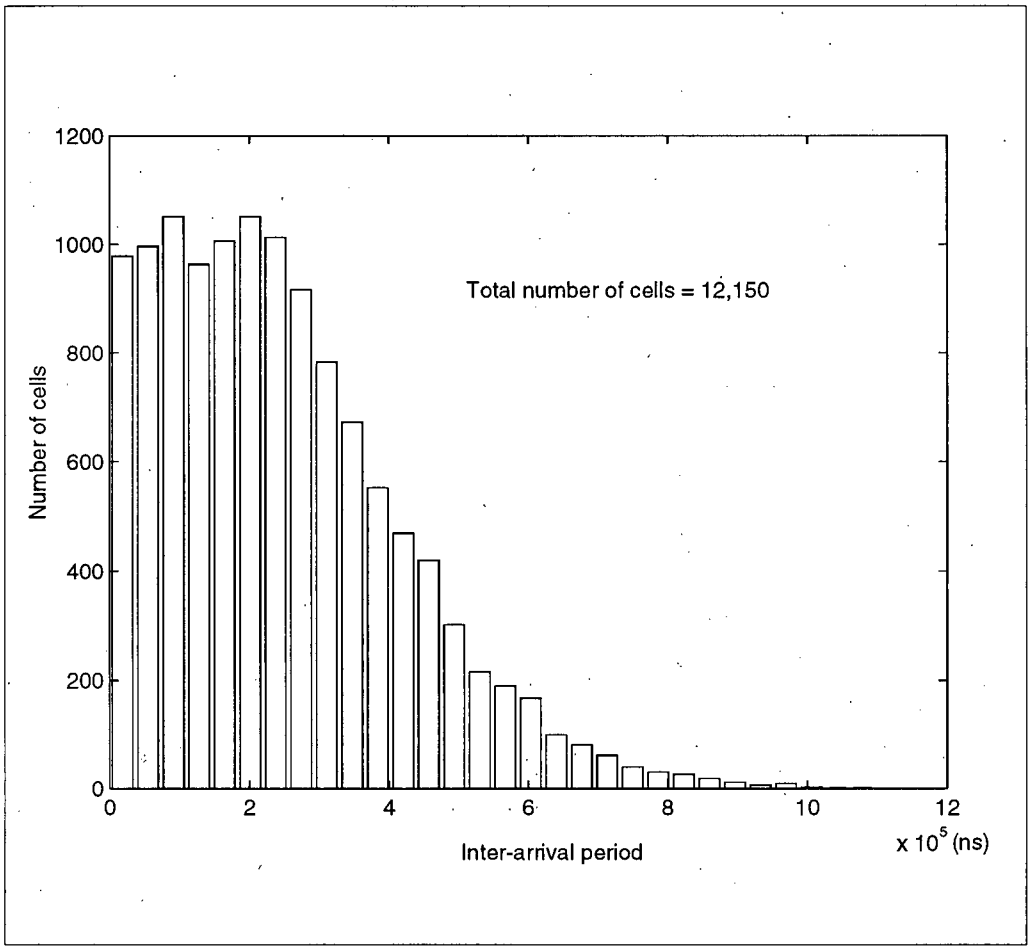

Figure 5.1.1 Inter-Arrival Periods Sample Histogram

DS1 transmission session. Appendix A presents the VHDL code that implements this traffic generator.

Figure 5.1.1 shows a sample histogram of the inter-arrival periods produced by the traffic generator according to equation (27).

It should be noted that the generated cell stream has the following properties:

1. The number of cells generated is 12,150 cells, over a period of 3 seconds.
2. d_{max} = 1 ms, and d_{min} = 0.1 ms.

3. The inter-departure period = 247,000 *ns* corresponding to a peak rate of 1.55 Mbps. This can be found by using equation (17), resulting in $T = \frac{48*8\ bits}{1.55\ Mbps} = 247,000\ .\ ns.$
4. The overall mean of the jittered inter-arrival periods is 247,050 *ns*. This shows that we practically have a zero-mean jitter process.
5. The maximum inter-arrival period = 1,097,419 *ns*, and the minimum inter-arrival period = 23 *ns*.

5.1.4 Random Number Generator

The random number generator (RNG) used to produce the values of *r(i)* in equation (26) is called Linear-Congruential Generator, and it is proven to be suitable for simulations similar to the ones used in our traffic generator. The function used in the RNG is deterministic and not fully random, i.e. it is pseudo-random. It is random in the sense that the numbers it generates would pass statistical tests for randomness. Pseudo-random generators are very suitable for our simulation since it is often desirable to be able to repeat a simulation session in the same manner it was done before.

Given a random seed, the RNG outputs a sequence of random numbers distributed uniformly between 0 and 1. The function used meets the main properties of a desired generator function since it is efficiently computable, generates a wide range of numbers, and the successive values it produces are independent and uniformly distributed. [24]

5.2 Simulation Results

To understand the system response and the effect of the different design parameters used, extensive simulations have been conducted. For example, investigating the adaptation rate of

the read clock can be achieved by running many simulation sessions using different values of α, while maintaining the buffer size control parameter (γ) and the estimation period control parameter (ϵ) constants. The simulation plots are compiled at the end of the chapter in figures 5.2.7 to 5.2.17. Figure 5.2.6 shows an enlarged portion of figure 5.2.7 to illustrate the fluctuation in the inter-arrival period, the read clock period, and buffer level. In these figures, *T1* represents the inter-arrival period and *T2(m)* represents the adjusted read clock period. The initial value of *T2(m)* is set to 500,000 *ns* to clearly show the read clock adaptation, and the initial length of the estimation period *j(0)* is 5 cells.

5.2.1 Effect of the Adaptation Rate Factor (α)

Figures 5.2.9, 5.2.11, and 5.2.16, show the response of the clock-recovery system when the value of α changes between 0.75, 0.5, and 0.25, respectively. The parameter γ is used in these sessions (i.e., γ can be 2, 0, or —2 cells as explained later), and the value of ϵ is 3 cells. The parameters α and γ are used in the adjustment algorithm as was shown in equation (16).

The effect that parameter α has on the adaptation rate of the read clock of the destination buffer can be seen in figure 5.2.2. If the value of α is large (α=0.75), the adaptation is more aggressive which means that the buffer read frequency converges to its steady-state value much faster than if a smaller value of α is used. This can be seen in table 5.2.1, where *T2(m)* is the period of the read clock of the destination buffer. After 350 *ms*, the period of the read clock reaches a closer value to the ideal period (247,000 *ns*) when α is large.

However, the fast adjustment is achieved at the price of poor estimation during certain periods. For example, from figure 5.2.2 and table 5.2.1, one can see that after 200 *ms*, the value of the read clock when α=0.5 has a better estimate (closer to 247,000 *ns*) than that

	α = 0.25	α = 0.5	α = 0.75
T2(m) after 200,000,000 *ns*	293,353 *ns*	285,428 *ns*	292,290 *ns*
T2(m) after 350,000,000 *ns*	276,651 *ns*	262,463*ns*	254,165 *ns*

Table 5.2.1 Effect of α on the Adjustment Rate
(Steady State Period of *T2(m)* = 247,000 *ns*)

when α=0.75. This is due to the fact that when α=0.75, the time between two consecutive adjustment actions is longer and the system does not respond to the changes in the buffer level as frequently as when α=0.5. Although the system takes more adjustment steps when α is small (α=0.25), the changes in the period of the read clock are very small which result in slower adjustments.

5.2.2 Effect of the Buffer Size Control Parameter (γ)

Figures 5.2.8, 5.2.12 and 5.2.15 show the response of the system when the parameter γ is not employed (γ=0) for the cases when only the adaptation factor α is used to control the changes in read frequency. In these figures, the value of α varies over the values 0.75, 0.5, and 0.25, and the value of ϵ is 3. Figure 5.2.3 shows that the buffer will deplete after a short time since γ, who is in charge of monitoring the buffer level, is not used.

Figures 5.2.9, 5.2.11 and 5.2.16 show the response of the same system described above but with the use of parameter γ (γ is 2, 0, or –2). As discussed earlier, the parameter γ is used to monitor the buffer level. When the buffer level is higher than a preset threshold and the read clock is not able to lower the buffer fill level (due to the closeness of the read

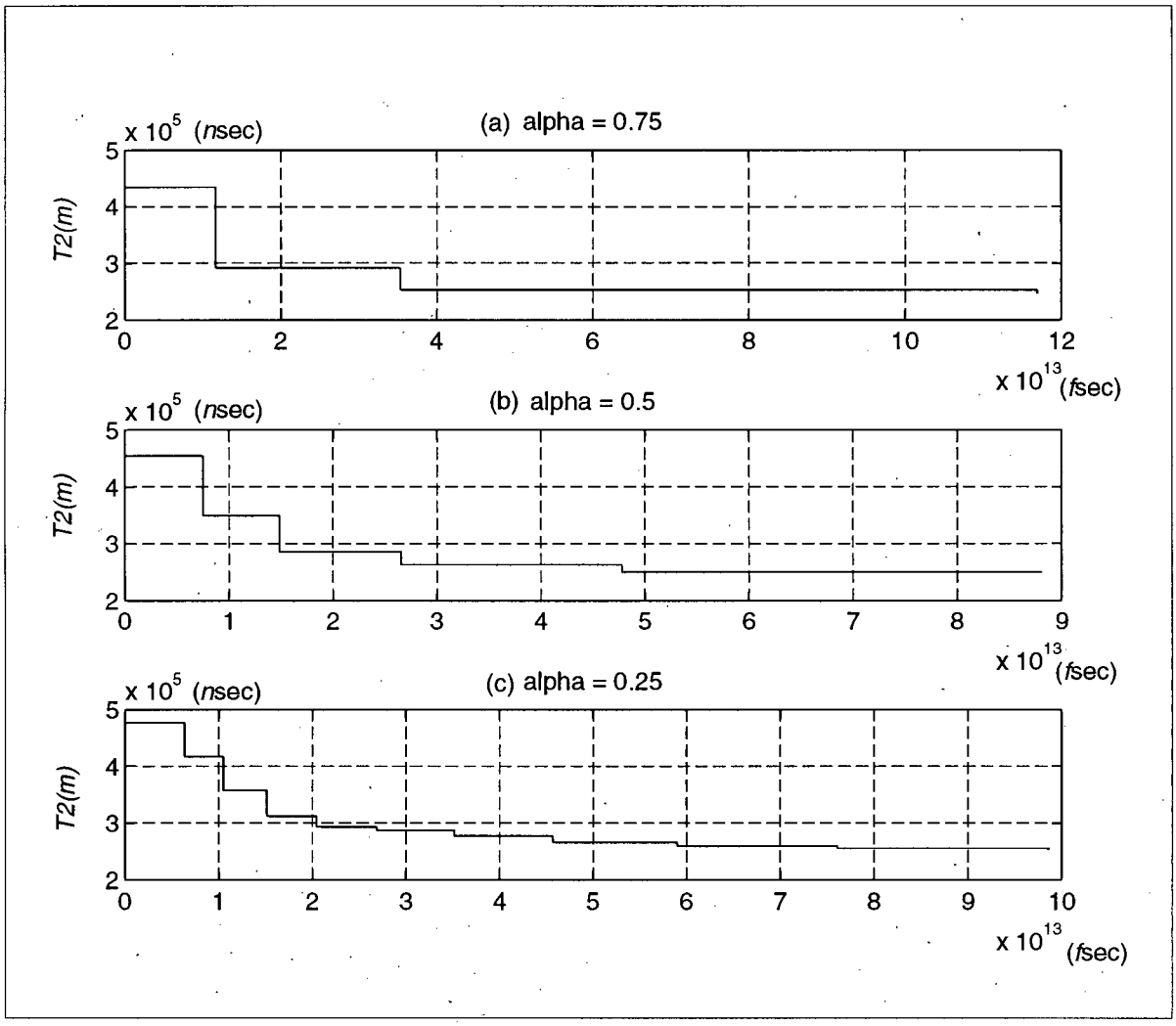

Figure 5.2.2 The Effect of α on the Period of the Read Clock

and write clock frequencies), γ is assigned a positive value (e.g. γ=2 cells) to increase the frequency of the read clock. This will reduce the buffer level. The effect of changing the positive value of γ on the drop of the buffer level can be seen in figure 5.2.4.

When the buffer level falls below the preset threshold, γ is assigned a negative value (e.g. γ = –2 cells) to slow down the read frequency and increase (or maintain) the buffer level. The effect of using parameter γ on the destination buffer level can be seen in figure

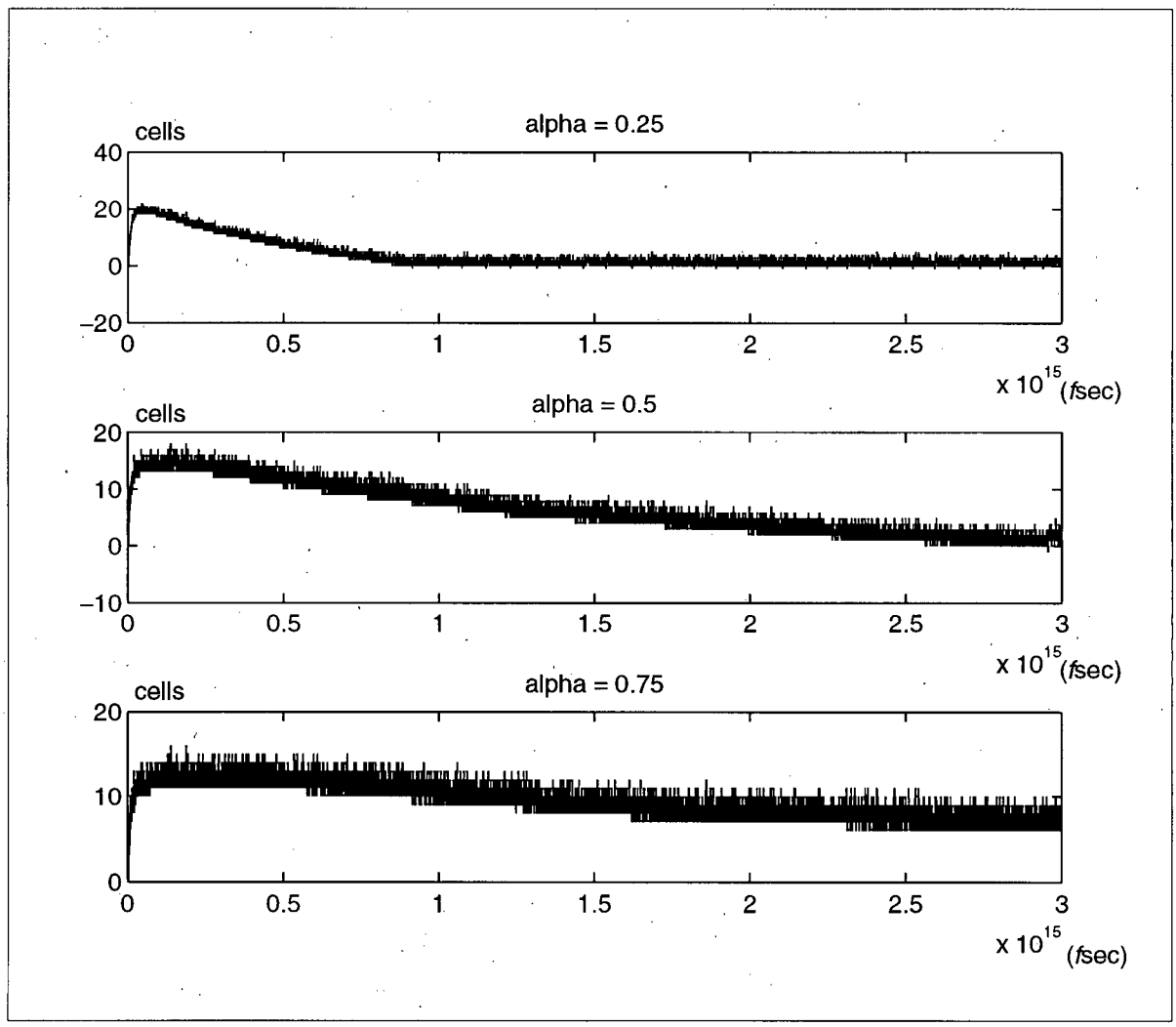

Figure 5.2.3 Effect of not Employing Parameter γ on the Buffer Level

5.2.5 where α is varied over the values 0.25, 0.5, and 0.75, and the value of ϵ is set to 3. This can be compared to figure 5.2.3. It is worth mentioning that changing the value of γ does not have a dramatic effect on the adjustments of the read clock period.

5.2.3 Effect of the Estimation Period Control Parameter (ϵ)

The parameter ϵ controls the length of the estimation period to prevent it from increasing indefinitely. Waiting too long for the adjustment can decrease the system robustness to

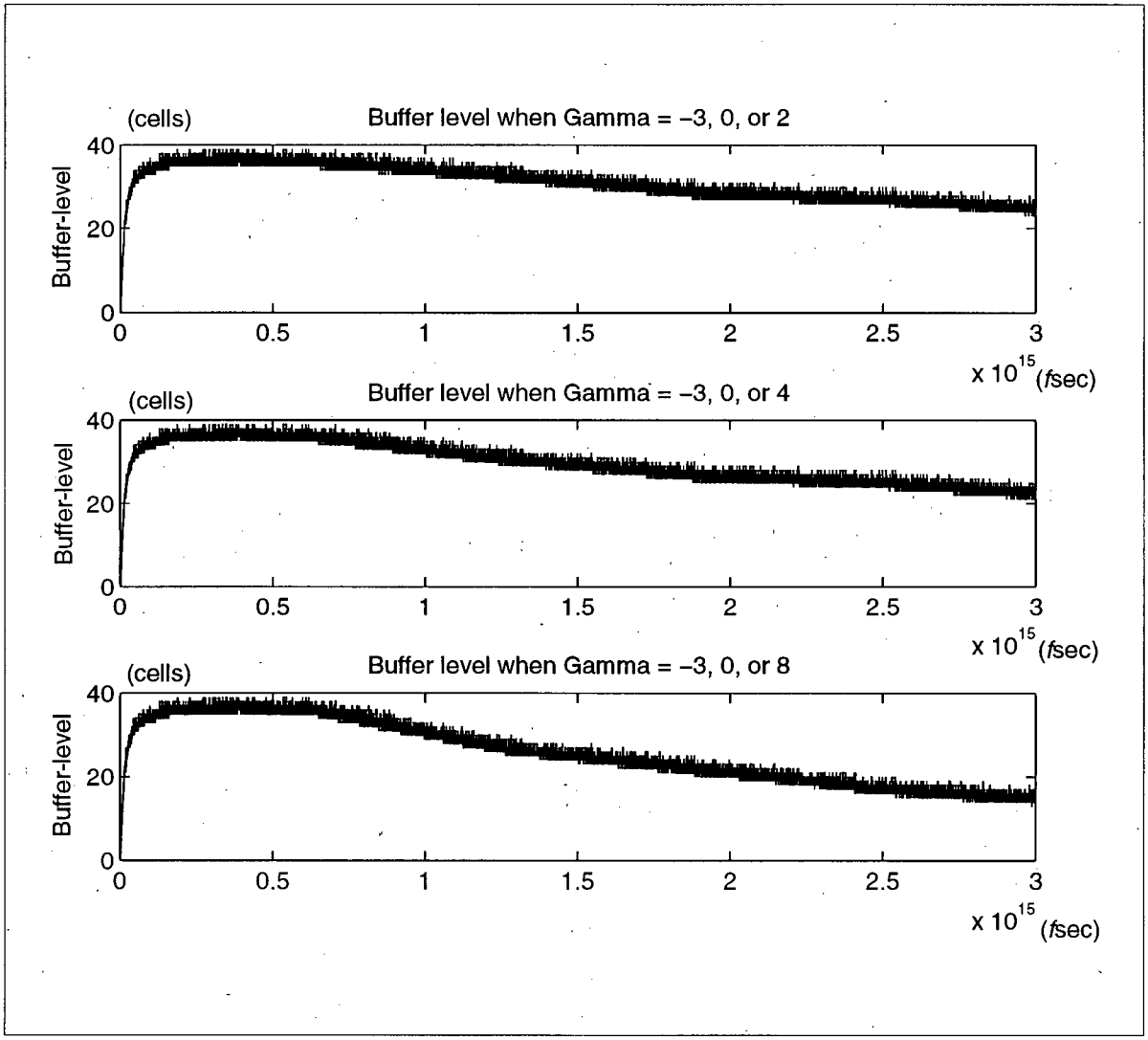

Figure 5.2.4 Effect of γ on the Drop of the Buffer Level

suddenly changing jitter statistics. Therefore, when the estimated frequency difference is smaller than ϵ cells/sec, (i.e. $\left|\Delta\hat{f}(m_1)\right| \leq \varepsilon$), we stop increasing the length of the estimation period and we maintain the length of the current one.

Figures 5.2.11 and 5.2.10 show the response of the system when the value of ϵ changes between 3 and 10, while α is 0.5, and γ is 2, 0, or –2. Moreover, figures 5.2.16 and 5.2.17 show the response of the system when the value of ϵ changes between 3 and 10 for the cases

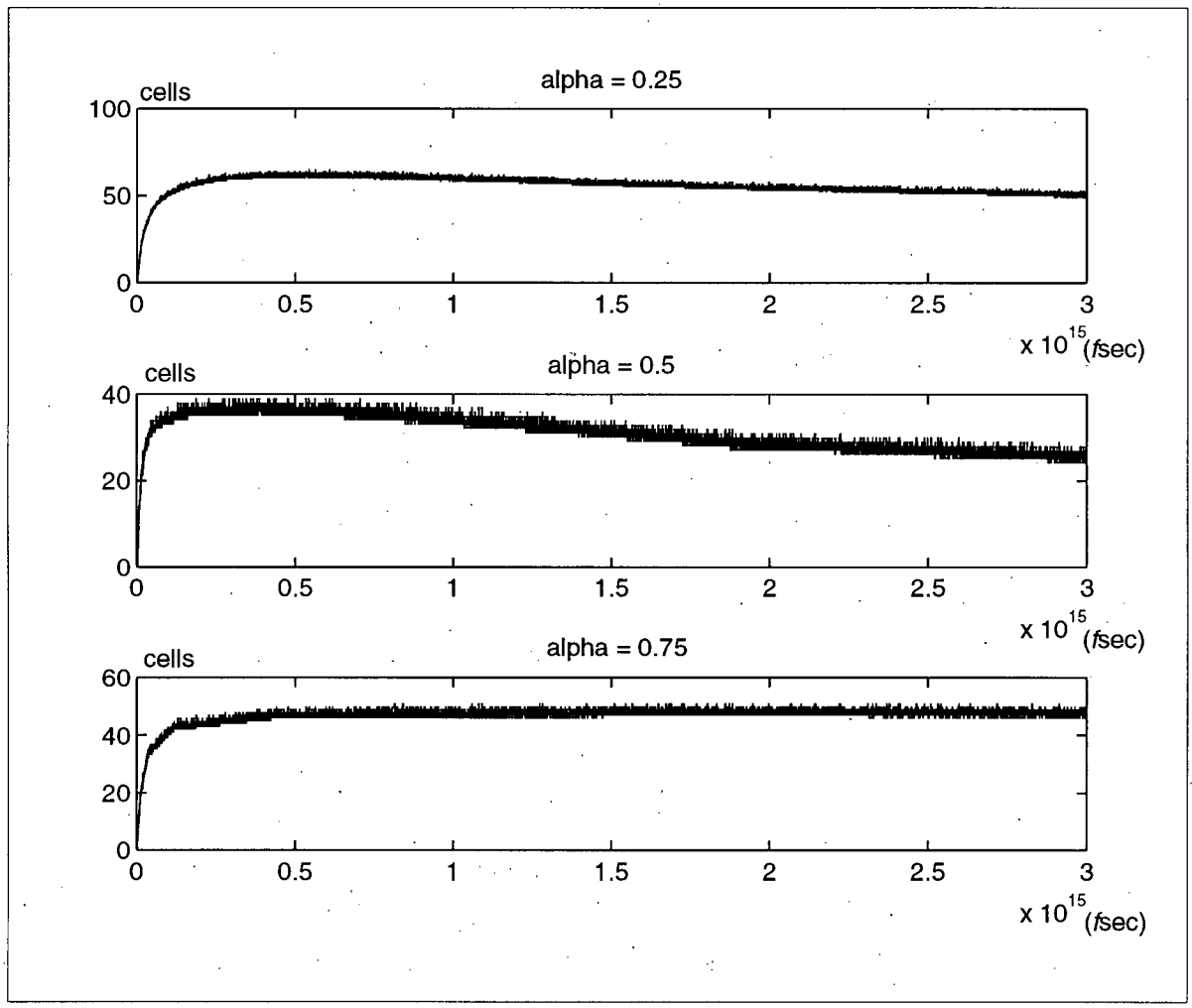

Figure 5.2.5 Effect of Employing Parameter γ on the Buffer Level

when α is 0.25 and γ is varied. One can see that changing ϵ has little effect on the buffer level and on the adaptation rate of the read clock. However, if checking against ϵ is not employed (or if ϵ is very small), the estimation (and subsequently the adjustment) period will grow large as $\Delta\hat{f}(m)$ becomes small. As a result, the clock recovery system will become less responsive to sudden changes in cell jitter leading to buffer overflow or underflow.

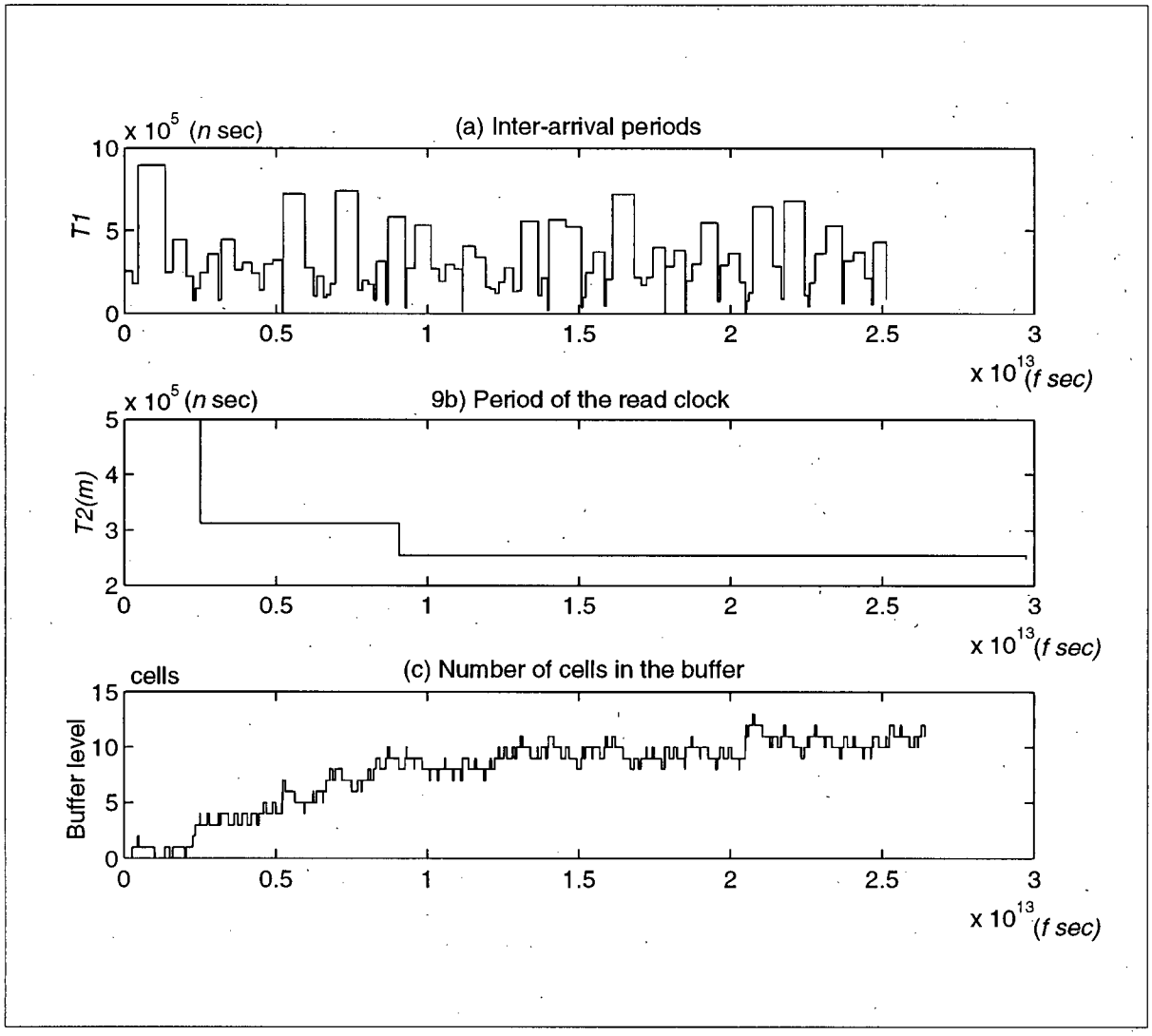

Figure 5.2.6 Enlarged Portion of the System Response when α=0.75, ϵ =10 and γ is not Employed (γ=0)

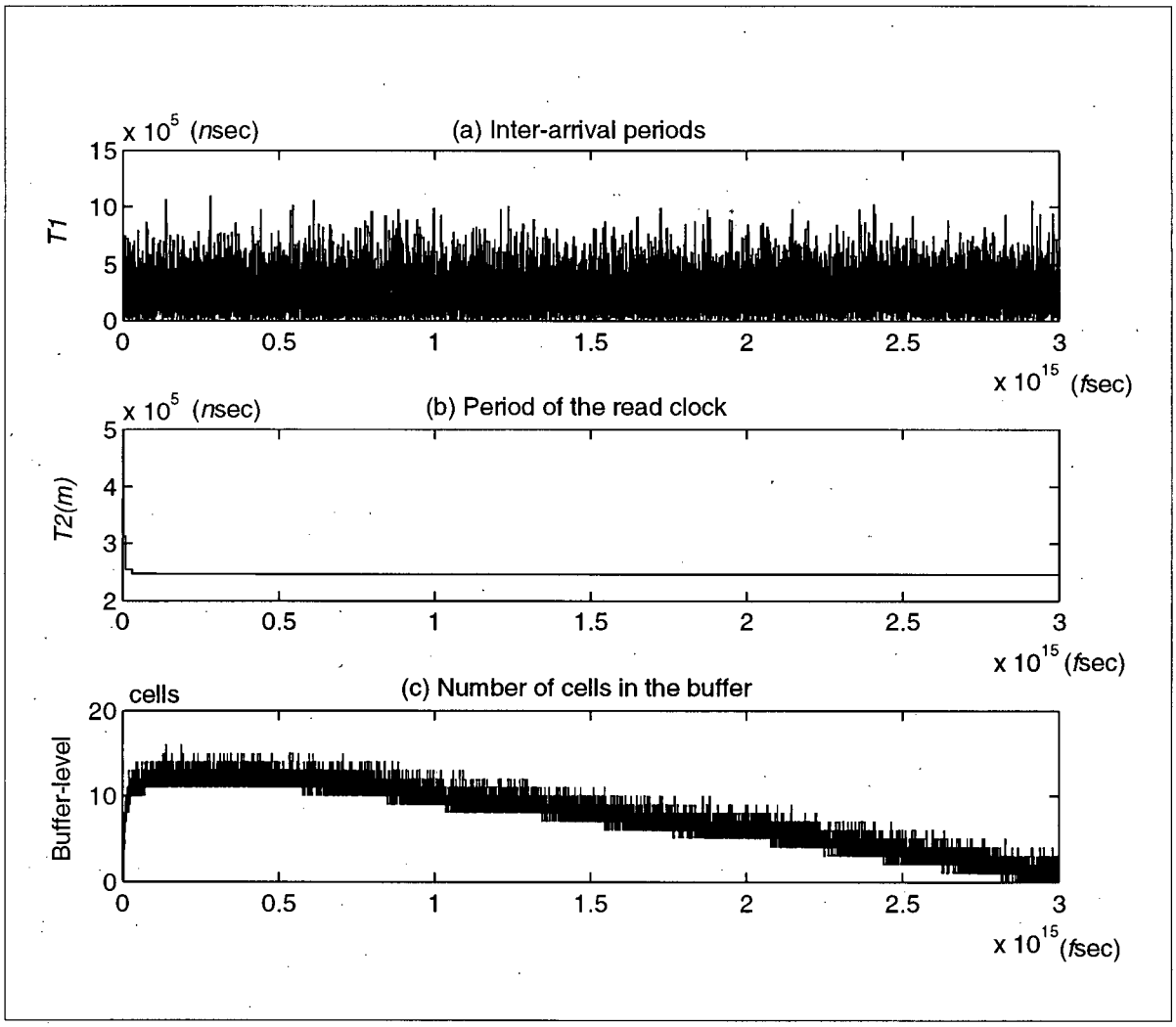

Figure 5.2.7 System Response when α=0.75, ϵ =10 and γ is not Employed (γ=0)

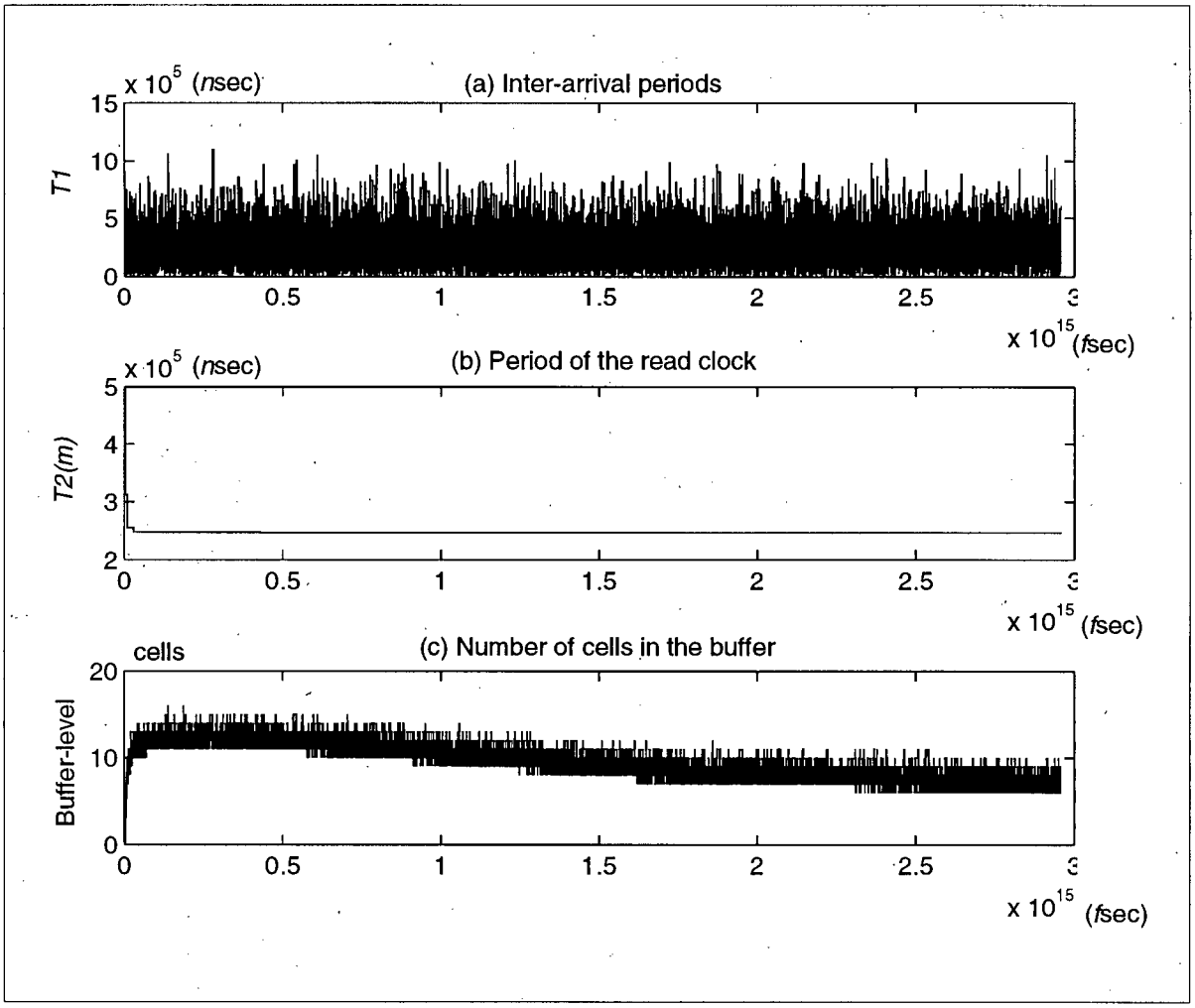

Figure 5.2.8 System Response when α=0.75, ϵ =3 and γ is not Employed (γ=0)

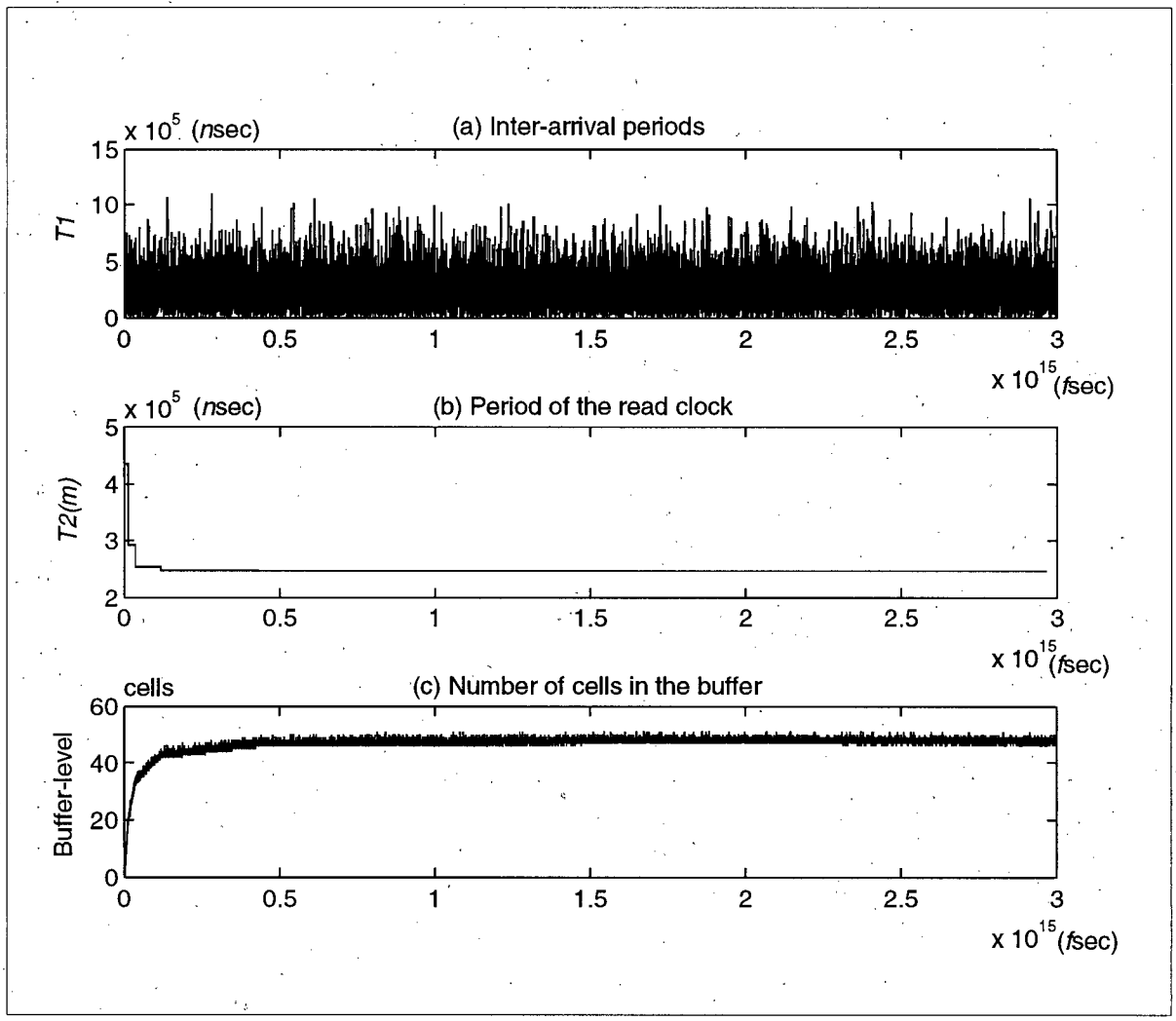

Figure 5.2.9 System Response when α=0.75, ϵ =3 and γ is Employed (γ =-2, 0, or 2)

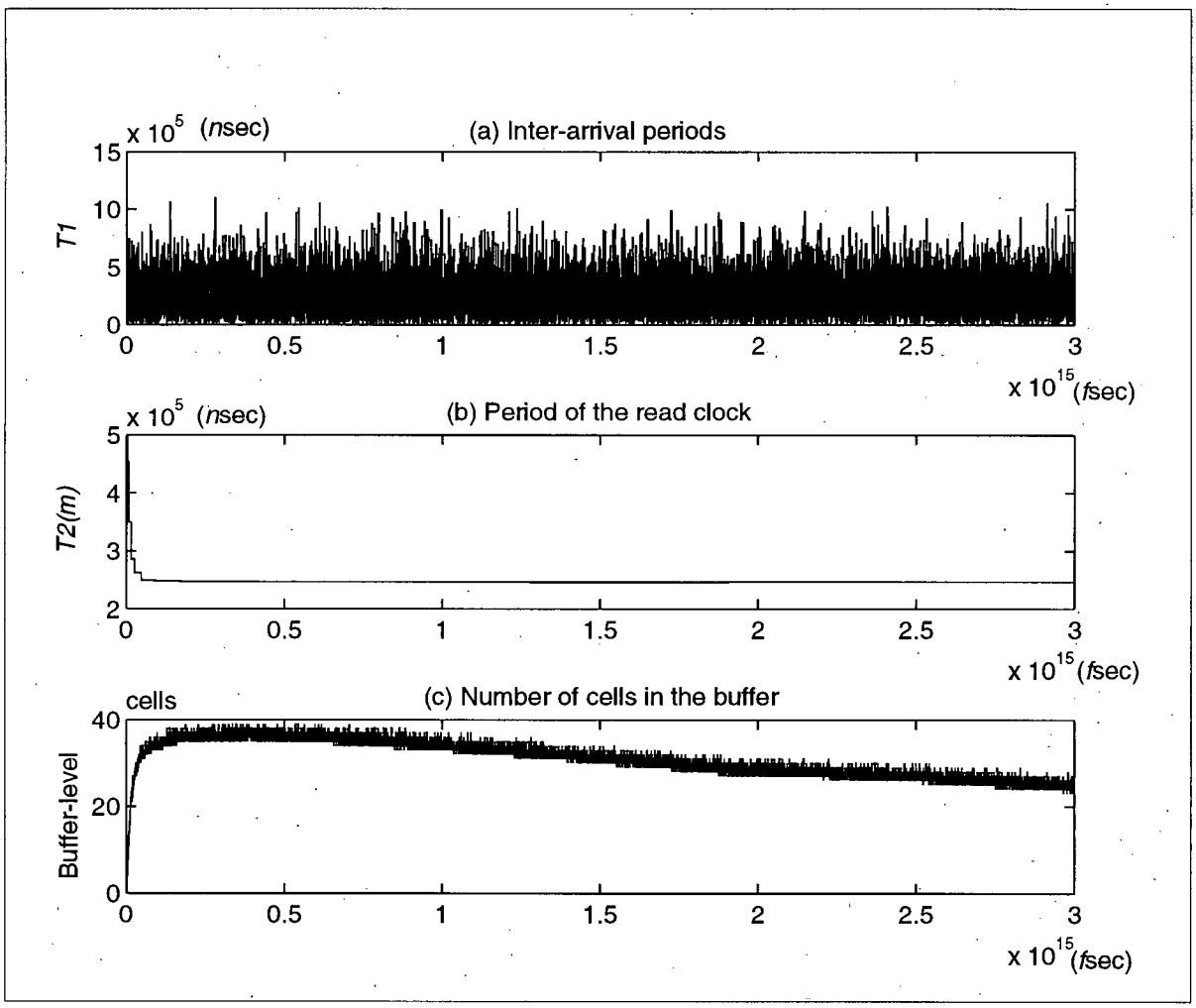

Figure 5.2.10 System Response when α=0.5, ϵ =10 and γ is Employed (γ =-2, 0, or 2)

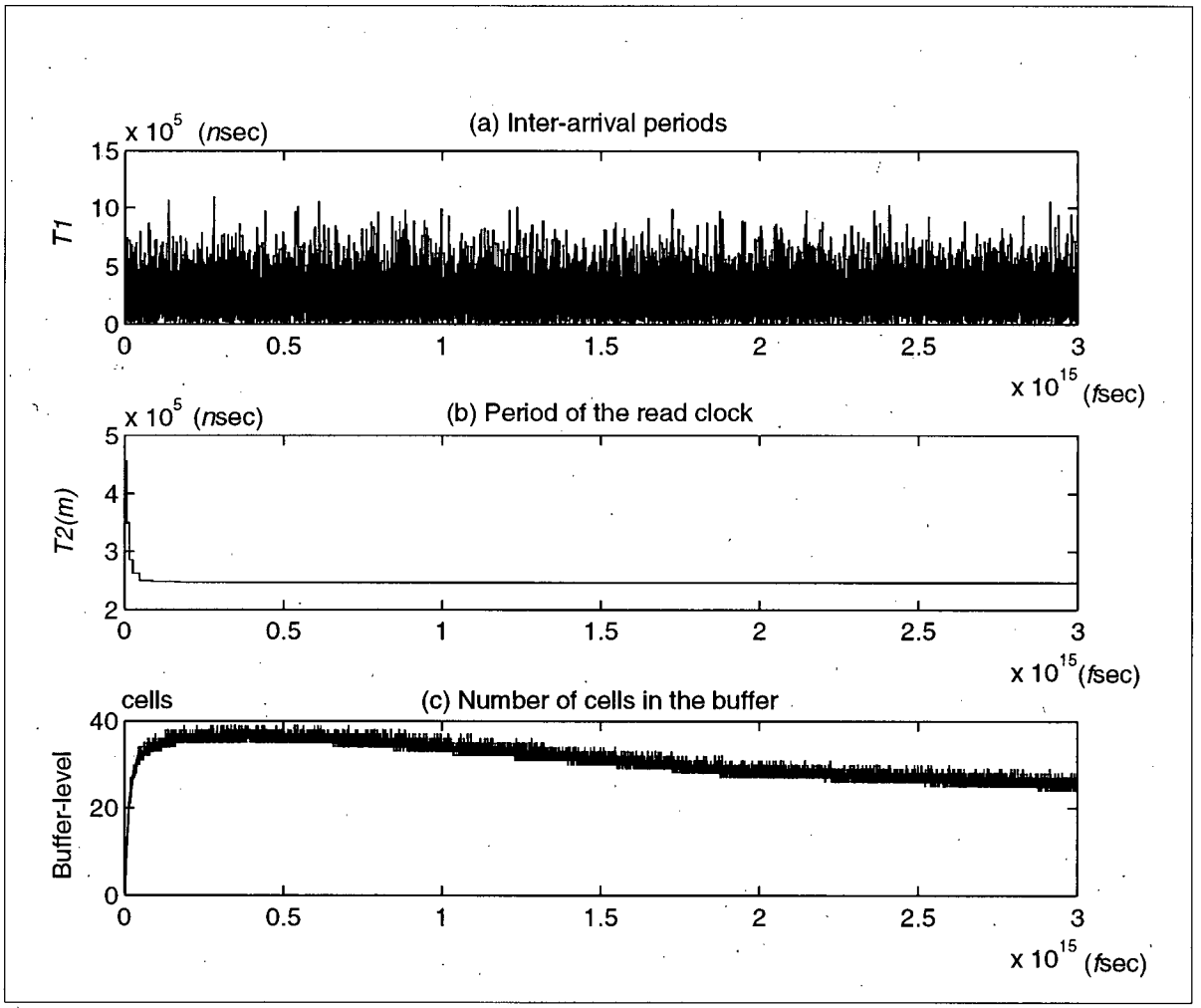

Figure 5.2.11 System Response when α=0.5, ϵ =3 and γ is Employed (γ =-2, 0, or 2)

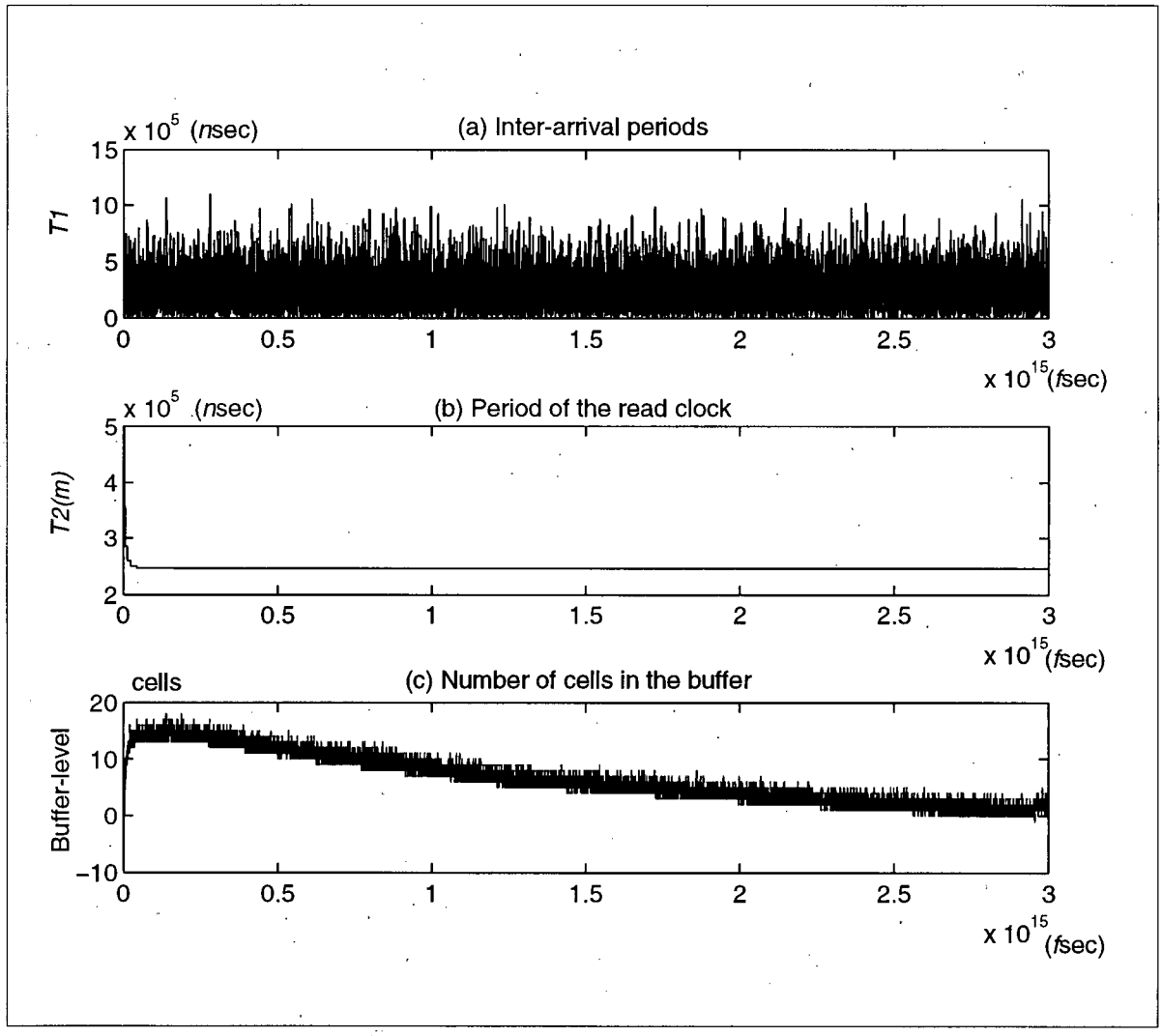

Figure 5.2.12 System Response when α=0.5, ϵ =3 and γ is not Employed (γ=0)

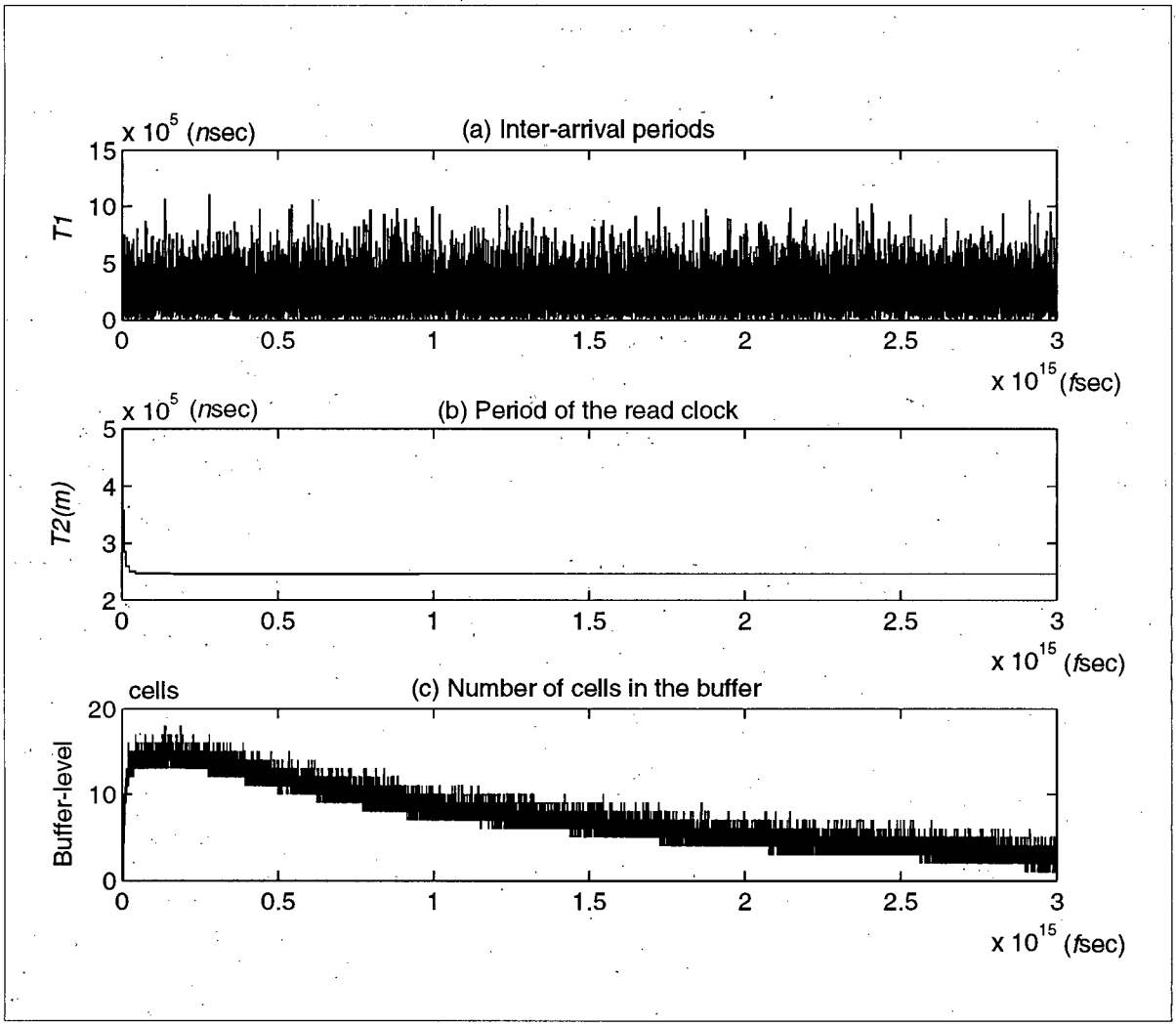

Figure 5.2.13 System Response when α=0.5, ϵ =10 and γ is not Employed (γ=0)

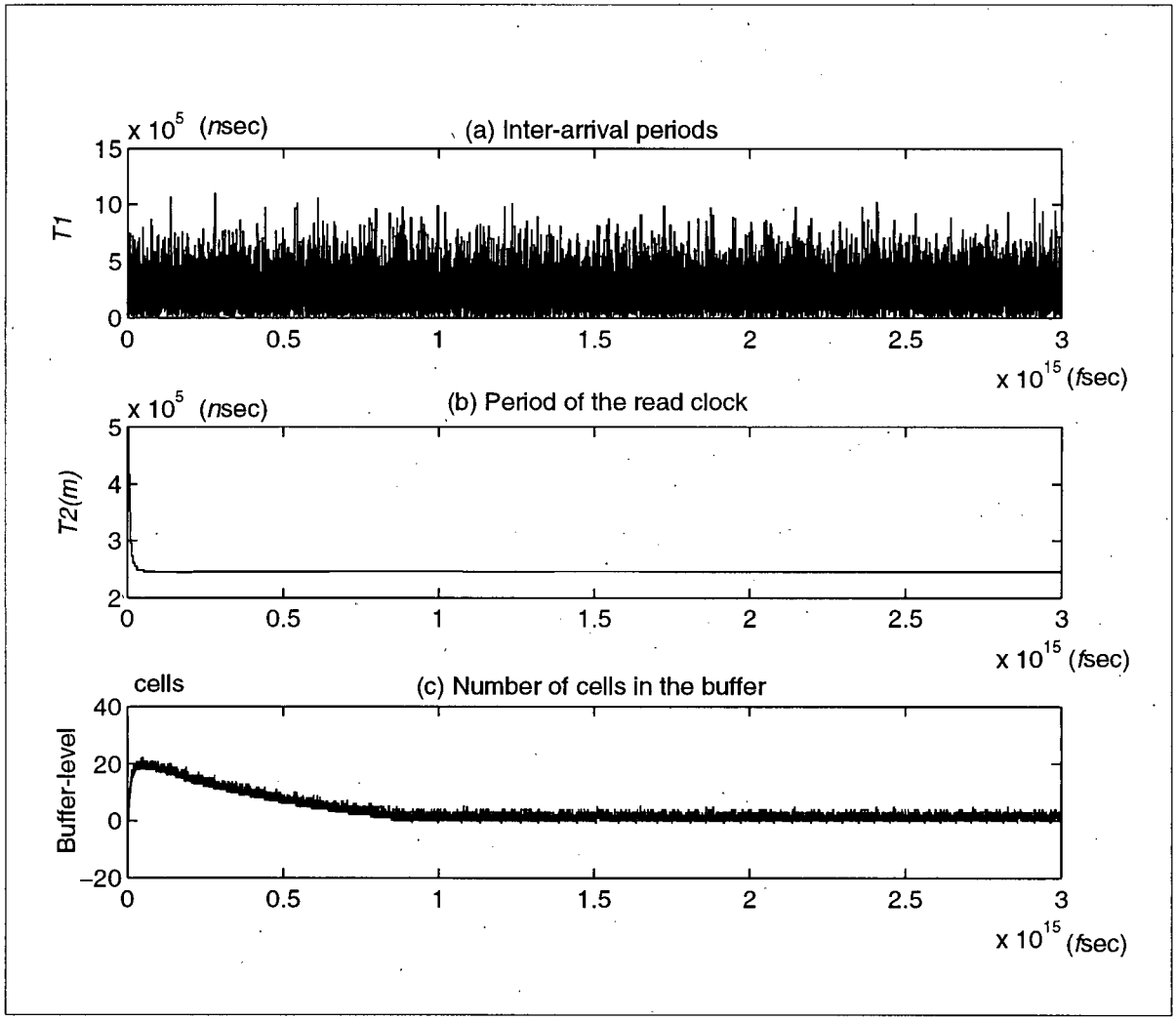

Figure 5.2.14 System Response when α=0.25, ϵ =10 and γ is not Employed (γ=0)

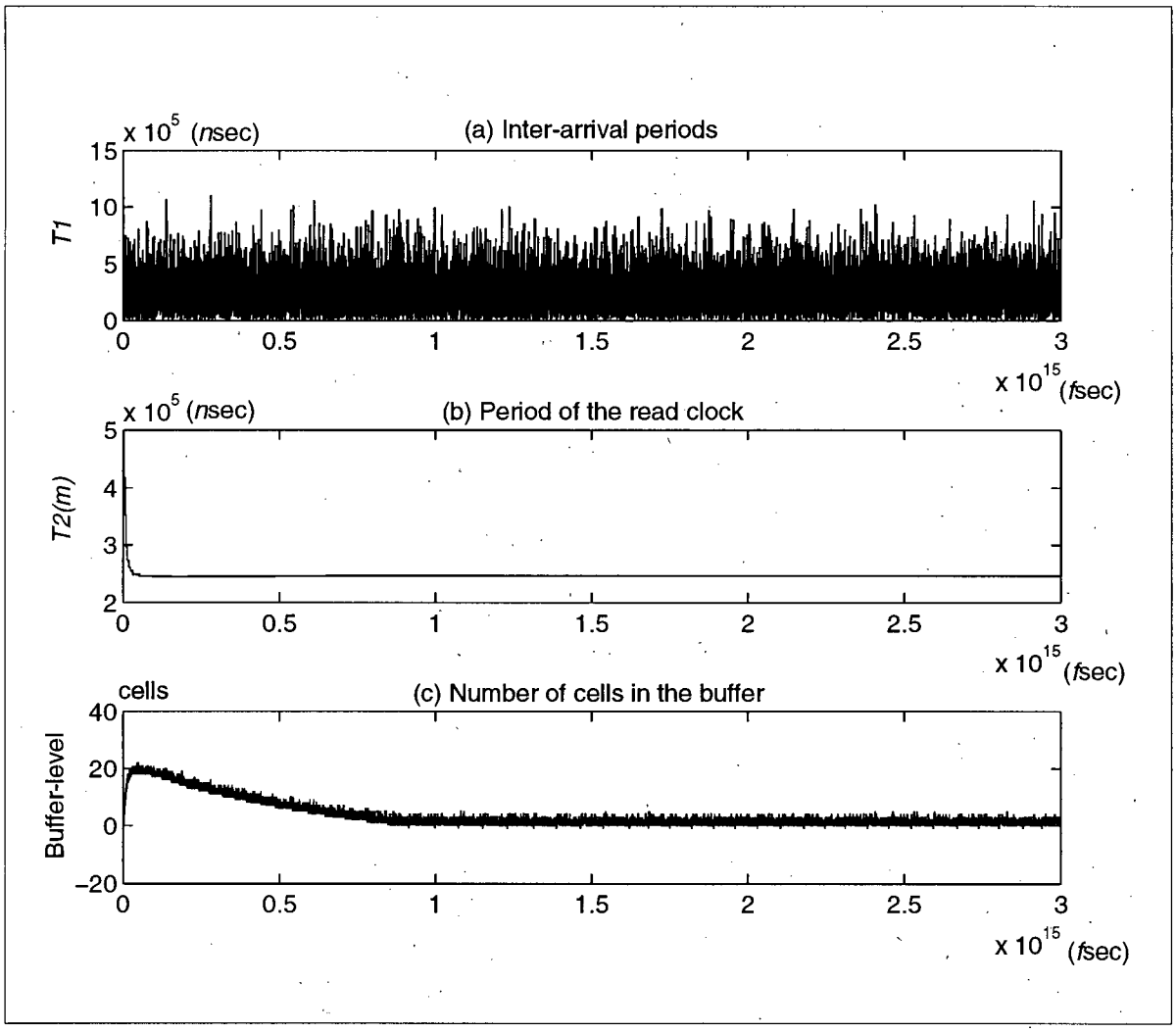

Figure 5.2.15 System Response when α=0.25, ϵ =3 and γ is not Employed (γ=0)

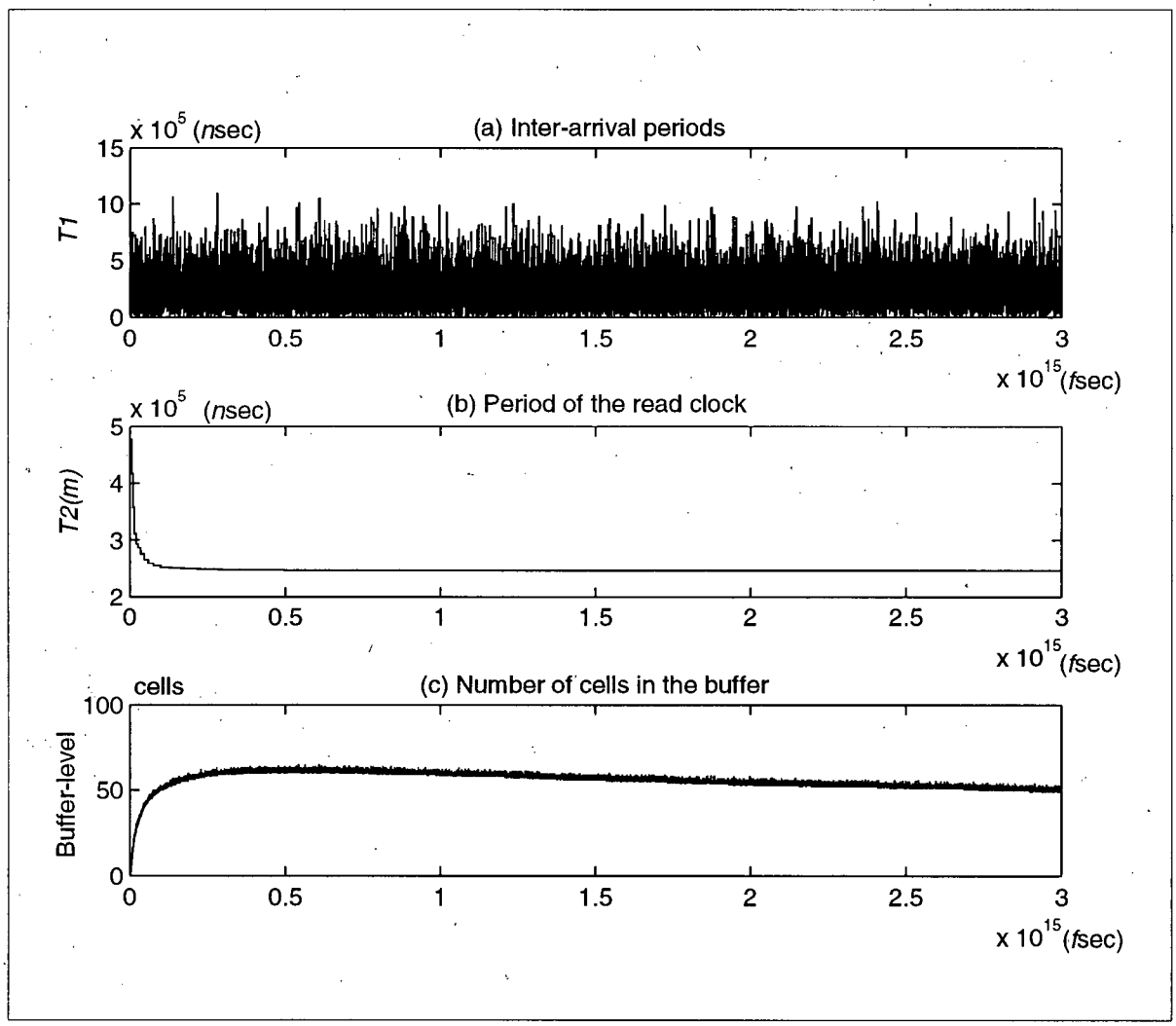

Figure 5.2.16 System Response when α=0.25, ϵ =3 and γ is Employed (γ =-2, 0, or 2)

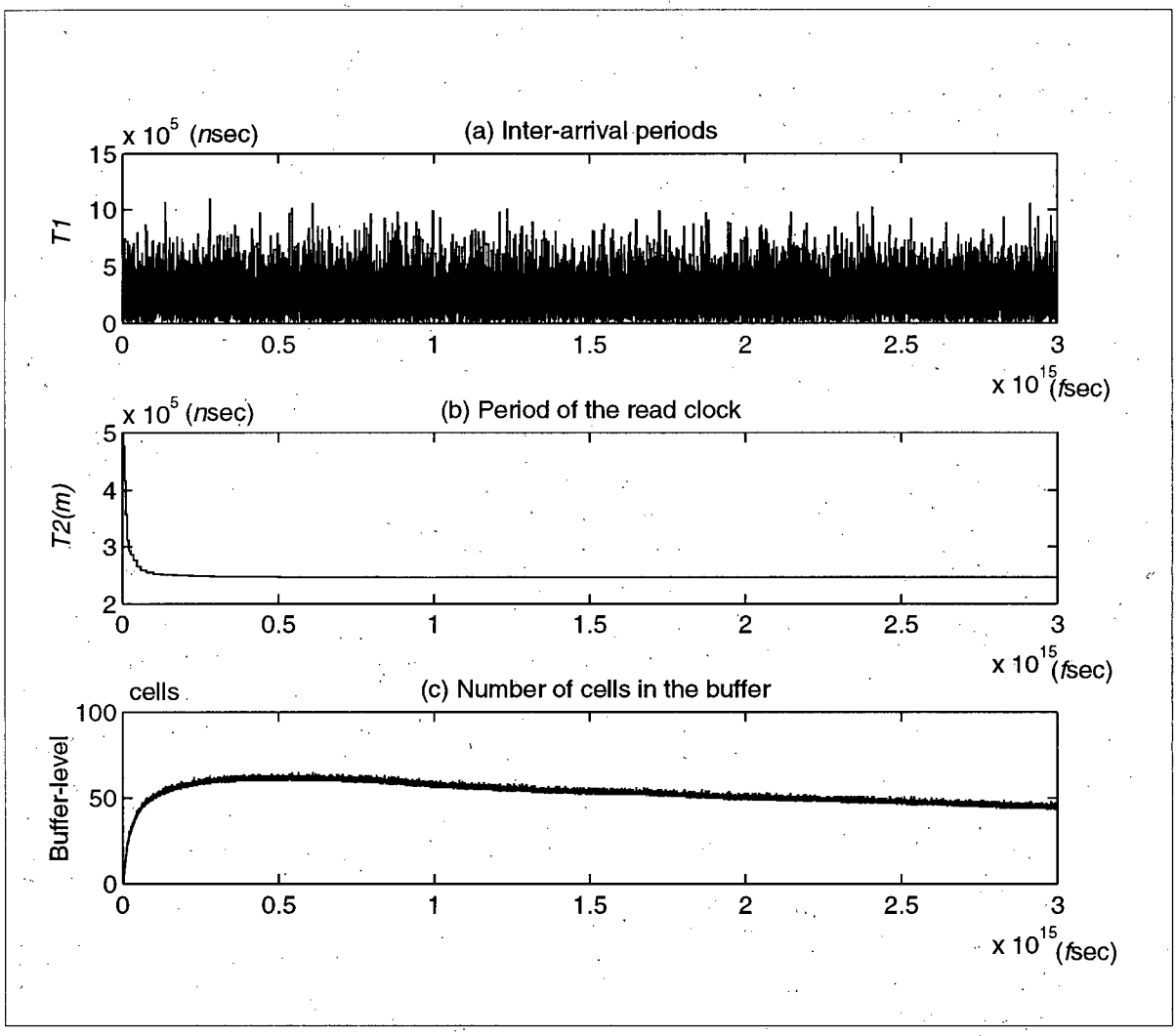

Figure 5.2.17 System Response when α=0.25, ϵ =10 and γ is Employed (γ =-2, 0, or 2)

5.3 Chapter Summary

In this chapter, the traffic generator that is used in the simulation of the adaptive clock recovery technique was presented. Then, the simulation results were shown to explain the effect of the different parameters on the performance of the system. It was shown that by carefully selecting the parameters, one can optimally recover the transmitter clock, and remove the jitter from the incoming cell stream at the receiver.

6 Conclusions and Future Work

6.1 Conclusions

The problem of jitter removal for CBR traffic over ATM has been studied, and the various techniques proposed for solving this problem have been reviewed. A major contribution of this thesis is an enhanced scheme for adaptive clock recovery and cell jitter removal at the receiver side of an ATM network. The operation state diagrams for the processes that are used in the scheme, and the flowcharts of the different processes were presented. The behavioral hardware description of the components needed in building the system was provided. The technique complies fully with the ITU-T standards. The system performance was simulated and analyzed for some particular data rates of interest to existing services. The results can be extended to other data rates.

As was shown in the simulation results of chapter 5, the adaptation rate parameter (α), and the buffer control parameter (γ) have important effect on the behavior of the system. The main problem was to determine suitable values for α and γ that will produce the best system behavior in terms of the quality of the buffer read clock, and the prevention of the buffer underflow. For the particular data rate used, the value of α that results in the best adaptation is around 0.5. This will provide smooth adaptation as well as good estimation of the read clock frequency. The employment of parameter γ is very essential to prevent the buffer depletion as well as the large growth in the buffer size. The parameter γ will attempt to keep the buffer size small, prevent it from depletion, and reduce the buffering delay of the incoming cells at the receiver.

Based on the above analysis, implementation, and simulation, the receiver in an ATM network will be able to restore the original CBR cell stream, and meet the QOS requirements negotiated by the user. Moreover, the system will be able to respond robustly to rapidly changing cell traffic.

6.2 Future Work

The cell delay jitter is a major problem for ATM CBR services, especially for high data rates. The work presented in this thesis was focused on providing an efficient engineering solution to the jitter removal problem in ATM CBR services, taking into account the practical implementation details. However, additional research is required to identify the critical performance parameters for an ATM AAL-1 receiver as a function of data rate and ATM network delays. For example, what should the initial buffer fill level be under a particular data rate and network delay.

Since the data that characterizes the traffic is mainly derived from service standards, it would be interesting to test the system behavior under realistic data generated from ATM networks. Another point of interest is the impact of the implementation technology and the architectural design styles employed in the actual realization of the clock recovery and jitter control unit. VHDL-based designs are suitable for automated hardware synthesis but do not necessarily produce optimized hardware. Additionally, careful analysis of digital VCOs must be conducted to determine how fast will the read clock adapt to the service.

Bibliography

[1] Synopsys 1076 VHDL Analyzer and Debugger Version 3.3b. Synopsys, Inc., Mt. View, CA, 1995.

[2] Bellcore. *Transport Systems Generic Requirements (TSGR): Common Requirements*. TR-NWT-000499, December 1993.

[3] F. Bonomi and K. Fendick. "The Rate-Based Flow Control Framework for the Available Bit Rate Service". *IEEE Networks*, pages 25–39, March-April 1995.

[4] CBR and AAL1 Processing. http://www.infotech.tu-chemnitz.de/www-public/prof-it/dako/atm/node3.html, September 25, 1995.

[5] T. M. Chen and S. S. Liu. *ATM Switching Systems*. Artech House, Boston, 1995.

[6] L. G. Cuthbert and J. C. Sapanel. *ATM: The Broadband Telecommunication Solution.* The Institution of Electrical Engineers, London, 1993.

[7] M. d. Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN*. Ellis Horwood, New York, 1991.

[8] J. N. Daigle. *Queueing Theory for Telecommunications*. Addison Wesley, Reading, MA, 1992.

[9] H. Kroner et. al. "Approximate Analysis of the End-to-End Delay in ATM Networks". *INFOCOM '92, Flerence, Italy*, pages 978–986.

[10] N. Almeida; et al. "End-to-End Synchronization in Packet Switched Networks".

[11] J. B. Kim; et. al. "International Standardization of B-ISDN". *Computer Networks and ISDN Systems*, No. 27, pages 5–27, 1994.

[12] R. P. Singh; et al. "Jitter and Clock Recovery for Periodic Traffic in Broadband Packet Networks". *IEEE Transactions on Communications*, Vol 42, No. 5, pages 2189–2196, May 1994.

[13] C. M. Aras; et. al. "Real Time Communications in Packet-Switched Networks". *Proceedings of the IEEE.*, Vol. 82, No. 1, pages 122–139, January 1994. Invited paper.

[14] A. K. Wong. "Queuing Analysis for ATM Switching of Continous Bit Rate Traffic — A Recursion Computation Method. *GLOBECOM '90, San Diego, CA*, pages 1438–1444.

[15] M. K. Liu. "Using Negative Stuffing Retiming for Circuit Emulation in a Packet Switching Network". *IEEE Transactions on Communications*, Vol 40, No 9, pages 1522–1531, September 1992.

[16] N. K. Cheung. "The Infrastucture for Gigabit Computer Networks". *IEEE Communications Magazine*, pages 60–68, April 1992.

[17] CCITT Recommendation G.711. *Pulse Code Modulation (PCM) of Voice Frequencies, CCITT Red Book*, Vol. III, Fascicle III.3, VIIIth plenary assembly, Spain, October 8–19, 1984.

[18] M. Gagnaire and N. Cartier. "Source Policing at the Output of a DQDB/B-ISDN Interconnection Gateway". *INFOCOM '94, Toronto, Canada*, pages 876–883.

[19] F. Guillemin and W. Monin. "Management of Cell Delay Variation in ATM Networks". *GLOBECOM '92, Flerence, Italy*, pages 128–132.

[20] R. Handel; M. Huber; and S. Schroder. *ATM Networks: Concepts, Protocols, Applications*. Addison-Wesley, Reading, MA, second edition, 1994.

[21] ITU-T Recommendation I.363. *B-ISDN ATM Adaptation Layer (AAL) Specification*, March 1993.

[22] ITU-T Recommendation I.432. *B-ISDN User-Network Interface-Physical Layer Specifications*, Rev. 1, Geneva, 1993.

[23] J. Kurose. "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks". *Computer Communication Review*, pages 6–15.

[24] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley, New York, 1991.

[25] R. O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*. Artech House, Boston, 1994.

[26] J. P. Cosmas; et al. "A Review of Voice, Data and Video Traffic Models for ATM". *European Transactions on Telecommunications and Related Technologies*, Vol. 5, No. 2, pages 139–154, March-April 1994.

[27] C. Partidge. *Gigabit Networking*. Addison-Wesley, Reading, MA, 1994.

[28] H. Saito. *Teletraffic Technologies in ATM Networks*. Artech House, Boston, 1994.

[29] D. R. Smith. *Digital Transmission Systems*. Van Nostrand Reinhold, New York, second edition, 1993.

[30] J. C. Bellamy. "Digital Network Synchronization". *IEEE Communications Magazine*, pages 70–83, April 1995.

[31] ANSI T1.511. *B-ISDN ATM Layer Cell Transfer — Performance Parameters*, 1994.

[32] HDLDESK v2.1.(1). Cadence Design Systems, Inc., San Jose, CA, 1994.

[33] Z. Wang, J. Crowcroft. "Analysis of Burstiness, and Jitter in Real-Time Communications. *SIGCOMM '93, Ithaca, NY*, pages 13–19.

Appendix A VHDL Code for the Traffic Generator

```
    library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use work.rand_pack.all;
use work.time_conv.all;
use work.time_conv2.all;

entity rand_gen_clk1 is
     port(writ_period : OUT TIME := 0 ns;
     r_period: OUT INTEGER;
     data_in : OUT INTEGER; clk : OUT BIT);
end rand_gen_clk1;

architecture impl of rand_gen_clk1 is
     SIGNAL lost : BIT := '0';
     -- max_delay is the maximum delay the cell can suffer
      CONSTANT max_delay : TIME := 1 ms;
     -- min_delay is the minimum delay the cell can suffer
      CONSTANT min_delay : TIME := 100000 ns;
     -- num_cells is the number of ATM cells to be generated during the session
```

```
    CONSTANT num_cells : INTEGER := 500000;
    – trans_period is the time elapsed between transmission of two
    – consecutive ATM cells for a certain application.
    CONSTANT trans_period : TIME := 247000 ns;

begin
    process
    – x is the output of random number generator
     variable x : real;
    – seed is a random nuber fed to the random nuber generator
     variable seed : integer := 80000000 ;
     variable index, r_mean_delay : integer := 0;
    – The Inter_arrival period between two cells
     variable IA_period : time;
    – mean delay is the random end-to-end delay encounterd by a cell
    VARIABLE mean_delay : TIME := 2 ms;
     – trans_time is the time at which the ith cell is transmitted
    VARIABLE trans_time : TIME := 247000 ns;
    VARIABLE A : TIME;
    VARIABLE legal_period : TIME;
    – prev_arriv_time is the time at which the (i-1)th cell arrived
    – assume that the first cell (cell 0) had a delay of 200,000 ns
```

```
VARIABLE prev_arriv_time : TIME := 2 ms;

begin
    -- this part generates the first cell, and IA period
    r_mean_delay := cvIntNs(mean_delay);
    IF ((mean_delay >= min_delay) AND (max_delay >= mean_delay)) THEN

        IA_period := trans_time + mean_delay - prev_arriv_time;
        prev_arriv_time := trans_time + mean_delay;
        r_period <= cvIntNs (IA_period);
        writ_period <= IA_period;
        data_in <= 1;
        lost <= '0';
        clk <= '0';
        wait for IA_period;
    ELSE

        IA_period := trans_time + max_delay - prev_arriv_time;
        prev_arriv_time := trans_time + max_delay;
        r_period <= cvIntNs (IA_period);
        writ_period <= IA_period;
        data_in <= -1;
```

```
            lost <= '1';
            clk <= '0';
            wait for IA_period;
        END IF;
        clk <= '1';
        wait for 10 ns;

        -- This is to generate the second cell and the rest of the cells
    FOR i IN 2 TO num_cells LOOP
         random(seed,x);
         trans_time := i * trans_period;
        IF (min_delay > prev_arriv_time - trans_time) THEN
         A:= min_delay;
        ELSE
         A := prev_arriv_time - trans_time;
        END IF;
        legal_period := max_delay - A;
        mean_delay := A + (x * legal_period);
        r_mean_delay := cvIntNs(mean_delay);
        trans_time := i * trans_period;
        IF ((mean_delay >= min_delay) AND
              (mean_delay >= (prev_arriv_time - trans_time)) AND
```

```
            (max_delay >= mean_delay)) THEN
            IA_period := trans_time + mean_delay - prev_arriv_time;
            prev_arriv_time := trans_time + mean_delay;
          r_period <= cvIntNs (IA_period);
          writ_period <= IA_period;
          data_in <= i;
          lost <= '0';
          clk <= '0';
          wait for IA_period;
        ELSE
          IA_period := trans_time + max_delay - prev_arriv_time;
          prev_arriv_time := trans_time + max_delay;
          r_period <= cvIntNs (IA_period);
          writ_period <= IA_period;
          data_in <= —1;
          lost <= '1';
          clk <= '0';
          wait for IA_period;
        END IF;
        clk <= '1';
        wait for 10 ns;
END LOOP;
```

```
end process;

end impl;


CONFIGURATION rand_clk1con OF rand_gen_clk1 IS

FOR impl

END FOR;

END rand_clk1con;
```

Appendix B List of Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Services Digital Network
CAD	Computer Aided Design
CBR	Continuous Bit Bate
CCITT	International Consultative Committee for Telegraphy and Telephony
CDV	Cell Delay Variation
CER	Cell Error Ratio
CLR	Cell Loss Ratio
CMR	Cell Misinsertion Rate
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
CSI	Convergence Sublayer Indicator
CTD	Cell Transfer Delay
DA	Digital-to-Analo
DPLL	Digital Phase-Locked Loop
FDDI	Fiber Distributed Data Interface
FEC	Forward Error Correction
FIFO	First In First Out
Gbps	Giga bits per second
HDTV	High Definition Television
HEC	Header Error Check
ITU-T	International Telecommunication Union-Telecommunications Standardization

Mbps	Mega bits per second
MCR	Minimum Cell Rate
MPEG-2	Moving Picture Experts Group-2
OAM	Operations and Maintenance
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
PDU	Protocol Data Unit
PLL	Phase-Locked Loop
PM	Physical Media
QOS	Quality Of Service
RAM	Random Access Memory
RNG	Random Number Generator
RTS	Residual Time Stamp
SAP	Service Access Points
SAR	Segmentation and Reassembly
SC	Sequence Count
SCR	Sustained Cell Rate
SDU	Service Data Unit
SECBR	Severely Errored Cell Block Ratio
SN	Sequence Number
SNP	Sequence Number Protection
SONET	Synchronous Optical Network
SRTS	Synchronous Residual Time Stamp
TC	Transmission Convergence
TDM	Time Division Multiplexing
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate
VCO	Voltage Controlled Oscillator

VHDL	Very-high-speed-integrated-circuit Hardware Description Language
VLSI	Very Large Scale Integration