The ATM Forum
Technical Committee

UTOPIA
Specification
Level 1, Version 2.01

af-phy-0017.000

March 21, 1994

UTOPIA, An ATM-PHY Interface Specification
Level 1, Version 2.01
March 21, 1994

The ATM Forum
2570 W. El Camino Real
Suite 304
Mountain View, CA 94040
Telephone: +1.415.949.6700
Fax: +1.415.949.6705
E-mail: info@atmforum.com

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Document Purpose

This document specifies the *Universal Test & Operations PHY Interface for ATM* (UTOPIA) data path interface. UTOPIA defines the interface between the *Physical Layer* (PHY) and upper layer modules such as the ATM Layer, and various management entities. The definition allows a common PHY interface in ATM subsystems across a wide range of speeds and media types. This definition covers connection to devices supporting ATM PHY specifications from sub-100 Mbps to 155 Mbps, and provides guidelines for 622 Mbps.

1.2 Document Scope

The UTOPIA *data path* specification defines two interface signal groups and general ATM and PHY layer device capabilities. The two interface signal groups are: Transmit and Receive. The device capability specification defines minimum capabilities primarily of the PHY layer device, but secondarily the ATM (SAR) device also. The definition of minimum device capabilities is required to allow different PHY layer devices to interface to a common ATM layer device. Figure 1 shows the relationship of this interface to ATM and PHY components of an ATM subsystem. The shaded interface groups are defined by this specification.

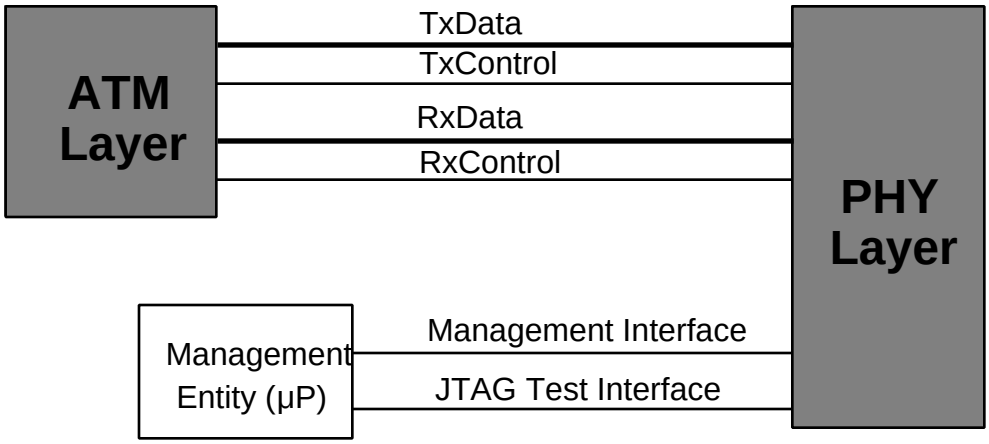

Figure 1. UTOPIA Interface Diagram

1.3 Current Level

This level of the specification covers the following two scenarios.

1. an 8-bit wide data path, using an octet-level handshake, operating up to 25 MHz, with a single PHY device
2. an 8-bit wide data path, using a cell-level handshake, operating up to 25 MHz, with a single PHY device

2. Motivation and Goals

The fundamental goal of this specification is to define a common, standard interface between ATM and PHY layers of ATM subsystems. The motivations for such a proposal are given below. All ATM developers wish to leverage their investment cost and development effort. There are multiple PHY layers defined for ATM (defined by ANSI, the ITU and the ATM Forum). Each of the PHY layers is potentially supportable by a common ATM layer. Consequently, one motivation of this proposal is to leverage ATM development across multiple PHY types.

The ATM-PHY interface is not a UNI (User Network Interface) nor NNI (Network Network Interface) specification, so is not directly relevant to interoperability. Therefore, no one is directly required to adhere to any specific interface definition. However, the economic advantages of multiple PHY layer parts that are interface compatible cannot be ignored. Correspondingly, another motivation for this is to encourage the adoption of this data path interface, implementable as a chip-internal, chip-to-chip, or even board-to-board interface.

A second goal is to allow expansion of the interface FIFO's using industry standard devices. Due to the differing data rates of the various ATM PHY layers, it is necessary to provide rate matching buffers (i.e. FIFO's) between ATM and PHY layers.

A third goal is to define an interface that supports a streaming mode of operation, where both the ATM and PHY layers have the capability to throttle the actual transfer rate.

A fourth goal is to support rates from sub-100 Mbps, up to 155 Mbps with a common interface, requiring only an 8-bit wide data path. There is also an intention to provide support for higher rates (e.g. 622 Mbps) by extending the data path width and increasing the transfer rate. A proposed method for such an extension is described in section 6[1].

[1]It is recognized that this document (Level 1) does not fully address this issue; the text is for guidance only.

3. Functional Description

3.1 Conventions

By convention, the interface where data flows from ATM to PHY layers is labeled the *Transmit Interface*, and the interface where data flows from PHY to ATM layers is labeled the *Receive Interface.*

All signals are active high, unless denoted via a trailing "*" after the signal name, e.g.

SIGNAL_1	Active High
SIGNAL_2*	Active Low

3.2 Interface Rates

Transmit and Receive transfers are synchronized via their respective interface transfer clock. With an 8-bit data path and a maximum clock rate of 25 MHz, this interface supports rates from sub-100 to 155 Mbps. Example interfaces are:

1. 155.52 Mbps (SONET/OC-3c)
2. 155.52 Mbps (8B/10B block coded)
3. 100 Mbps (4B/5B TAXI)
4. 44.736 Mbps (DS-3)
5. 51.84 Mbps (OC-1)

Higher rates (e.g. 622 Mbps) may be supported by extending the data path to 16 bits, and using a faster transfer clock. Guidance for a proposed 16-bit extension is described in section 6.

3.3 Device Capabilities3.3.1 Synchronization

The transfer of data at the octet level is via separate Transmit and Receive transfer synchronizing clocks. This specification assumes that the PHY layer must accept both Transmit and Receive transfer synchronizing clocks from the ATM layer. The reason for this is to allow the ATM layer to remain constant across different PHY layers (e.g. for use with different PHY daughter-boards).
The transfer of data is synchronized at the cell level via a *Start Of Cell* signal. This signal is asserted when the data transfer path contains the first byte of a cell. The use of octet-level or cell-level status signals for flow control is possible.

3.3.2 Rate Matching

Given that the PHY layer must accept transfer synchronizing clocks from the ATM layer, the PHY layer will require rate matching buffers, i.e. FIFO's. With the use of FIFO's, flow control signals are provided to allow both ATM and PHY to throttle the transfer rate. There are two methods of flow control : octet-level and cell-level. The same flow control signal is used, but has a slight semantic difference, depending upon the flow control method.

The Receive Interface transfers data only when the ATM layer requests it by asserting an *Enable* signal. The interface also provides an *Empty/Cell Available* signal from the PHY to allow PHY layer rate control. A false *Empty* signal indicates a valid receive octet and provides octet-level flow control. A true *Available* signal indicates the availability of a whole cell and provides cell-level flow control.

The Transmit Interface transfers data only when the ATM layer requests it by asserting an *Enable* signal. The interface also provides a *Full/Cell Available* signal from the PHY to allow PHY layer rate control. The *Full* signal is asserted if more than 4 additional writes would cause a PHY buffer overflow. The *Available* signal indicates there is enough space in the transmit FIFO to hold a complete cell.

3.3.3 Cell Processing

In an ATM-PHY interface, there arises the question of where cell HEC processing is performed/terminated[2]. This specification assumes as a minimum, that the PHY layer will perform HEC processing as specified in ITU-TS Recommendation I.432. Cell delineation, which can be based upon the HEC is a PHY layer function. However, for backward compatibility, cell transfers between ATM and PHY layers provide a "user-defined" field within the data stream. This field may be utilized to transfer the HEC. Figure 2 shows the transfer format of cells for 8-bit mode. See section 6 for details of 16-bit mode.

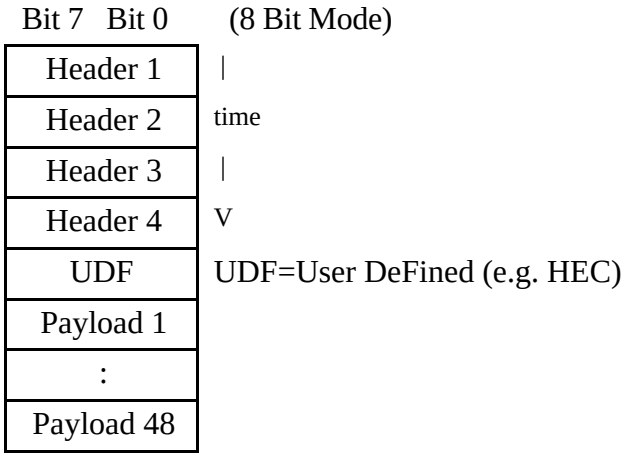

Figure 2. UTOPIA Cell Format

3.4 Relationship of Signals

The general paradigm for the relationship between data flow and enable signals is as follows:

1. when enable and data flow in the same direction, the enable corresponds to data during the current cycle,

2. when enable and data flow in opposite directions, the enable corresponds to data at the end of the next cycle.

The Start Of Cell signal is asserted to indicate when the first octet of a cell is present. The relationship of data to transfer clock is defined in the following sections.

[2]Cell HEC processing includes HEC generation, checking, and cell delineation.

4. Transmit Interface

4.1 Signals

This section defines the signals of the Transmit interface. The following Operation and Timing section provides the discipline for each signal. For specification purposes timing is described for a system where both ATM and PHY layers use the low-to-high transition of TxClk to sample and generate signals. An implementation may choose to use some other reference point for some PHY layer signals, but this is not described.

The following signals are defined as required (**R**) for the Transmit interface.

TxData[7..0]
: Data. Byte-wide true data driven from ATM to PHY layer. TxData[7] is the MSB.

TxSOC
: Start Of Cell. Active high signal asserted by the ATM layer when TxData contains the first valid byte of the cell.

TxEnb*
: Enable. Active low signal asserted by the ATM layer during cycles when TxData contains valid cell data.

TxFull*/TxClav
: Full/Cell Available. For octet-level flow control, TxFull* is an active low signal from PHY to ATM layer, asserted by the PHY layer to indicate a maximum of four more transmit data writes will be accepted[3]. For cell-level flow control, TxClav is an active high signal from PHY to ATM layer, asserted by the PHY layer to indicate it can accept the transfer of a complete cell.

TxClk
: Data transfer/synchronization clock provided by the ATM layer to the PHY layer for synchronizing transfers on TxData.

The following signals are defined as optional (**O**) for the Transmit interface.

TxPrty
: Parity. TxPrty is the odd parity bit over TxData[7:0], driven by the ATM layer.

TxRef*
: Transmit Reference. Input to the PHY layer for synchronization purposes (e.g. 8 kHz marker, frame indicator, etc.).

4.2 Operation and Timing

The Transmit interface is controlled by the ATM layer. The ATM layer provides an interface clock to the PHY layer for synchronizing all interface transfers. This convention requires the PHY layer to incorporate rate-matching buffers (i.e. a FIFO).

The transmit interface has data flowing in the same direction as the ATM enable. The ATM transmit block generates all output signals, and samples all input signals on the rising edge of TxClk.

---

[3]Four more write cycles represent at most 4 more octets for an 8-bit interface, and 8 more octets for a 16-bit interface.

Transmit data is transferred from the ATM layer to PHY layer via the following procedure. The PHY layer indicates it can accept data using the TxFull*/TxClav signal, then the ATM layer drives data onto TxData and asserts TxEnb*. The PHY layer controls the flow of data via the TxFull*/TxClav signal.

4.2.1 Octet-Level Handshake

During a time period termed the *transmit window*, the PHY layer stores data from TxData on the low-to-high transition of TxClk, if TxEnb* is asserted. The transmit window exists from the time that the PHY layer indicates it can accept data by deasserting TxFull*, until 4 **valid** write cycles after the PHY layer asserts TxFull*. The PHY layer may assert TxFull* at any time, and while asserted this indicates that the ATM layer may transfer no more than 4 data words on TxData.

The ATM layer must deassert TxEnb* within 4 data writes of TxFull* assertion and must not reassert TxEnb* until TxFull* is detected deasserted.[4] Asserting TxEnb* outside the transmit window is an error, and the PHY layer will ignore such writes. Inside the transmit window the ATM layer may assert and deassert TxEnb* as required.

4.2.2 Cell-Level Handshake

The cell-level handshake is identical to the octet-level handshake except for one difference, namely that once TxClav is asserted, the PHY layer must be capable of accepting the transfer of a whole cell. TxEnb* can be used by the ATM Layer to control the flow of data at an octet level (just as for octet-level handshake mode). To ensure that the ATM Layer does not cause transmit overrun, the PHY Layer must deassert TxClav at least 4 cycles before the end of a cell if it cannot accept the immediate transfer of the subsequent cell.

4.2.3 Examples

The timing sequences may be summarized as: TxEnb* asserted indicates valid ATM data available in the current cycle, TxFull* asserted indicates the PHY is unable to accept new data after 4 more write cycles, TxClav deasserted indicates the PHY is unable to accept another cell transfer after the current one.

Examples of Transmit interface operation are shown in Figures 3 to 5. Figures 3 and 4 show timing for an octet-level and cell-level handshake respectively. Figure 5 shows timing for the TxFull* and TxClav boundary conditions.

[4]This rule allows the ATM layer to exactly determine during which cycles data was transferred on TxData.

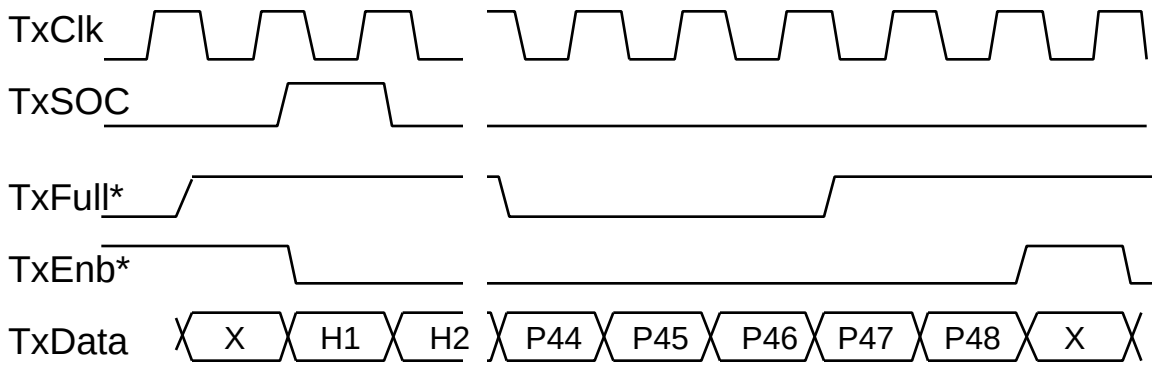

Figure 3. Transmit Timing for Octet-Level Handshake

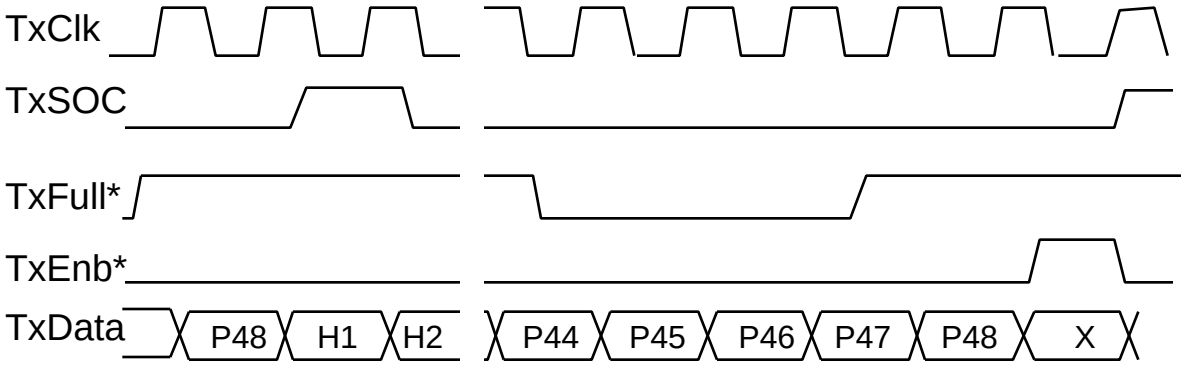

Figure 4. Transmit Timing for Cell-Level Handshake

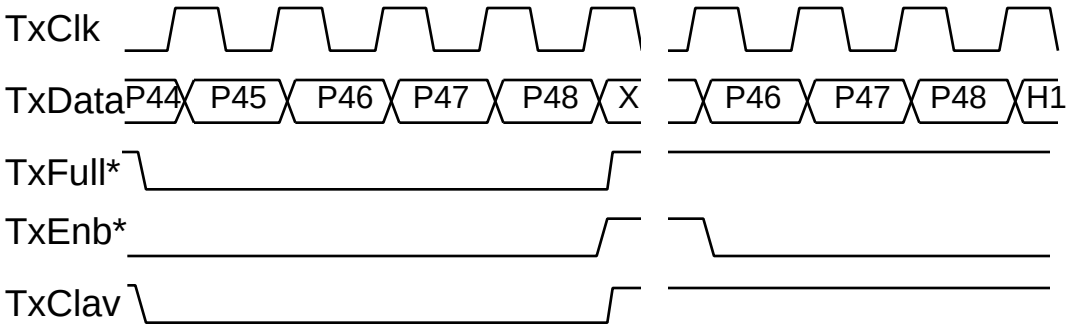

Figure 5. TxFull*/TxClav Timing

4.2.4 Timing Specifications

The following ATM and PHY Layer timing requirements/specifications apply to the Transmit interface (note where TxData is referenced, the same timing applies to TxData, TxPrty, and TxSOC).

Table 1. Transmit Timing

Item	Description	Applies To	Min	Max
tT1	TxClk frequency	TxClk	0	25 MHz
tT2	TxClk duty cycle	TxClk	40 %	60 %
tT3	output delay from TxClk	TxData, TxEnb*, TxRef*	1 ns	20 ns
tT4	input setup to TxClk	TxFull*/TxClav	10 ns	
tT5	input hold from TxClk	TxFull*/TxClav	1 ns	
tT6	TxRef* high or low pulse width	TxRef*	1 TxClk	1 TxClk

5. Receive Interface

5.1 Signals

This section defines the signals of the Receive interface. The following Operation and Timing section provides the discipline for each signal. For specification purposes timing is described for a system where both ATM and PHY layers use the low-to-high transition of RxClk to sample and generate signals. An implementation may choose to use some other reference point for some PHY layer signals, but this is not described.

The following signals are defined as required (**R**) for the Receive interface.

RxData[7..0]
: Data. Byte-wide data driven from PHY to ATM layer. RxData[7] is the MSB. Note, to support multiple PHY configurations, it is recommended that RxData be tri-stateable, enabled only when RxEnb* is asserted.

RxSOC
: Start Of Cell. Active high signal asserted by the PHY layer when RxData contains the first valid byte of a cell. Note, to support multiple PHY configurations, it is recommended that RxSOC be tri-stateable, enabled only in cycles following those with RxEnb* asserted.

RxEnb*
: Enable. Active low signal asserted by the ATM layer to indicate that RxData and RxSOC will be sampled at the end of the next cycle. Note, to support multiple PHY configurations, RxEnb* should be used to tri-state RxData and RxSOC PHY layer outputs. RxData and RxSOC should be enabled only in cycles following those with RxEnb* asserted.

RxEmpty*/RxClav
: Empty/Cell Available. For octet-level flow control, RxEmpty* is an active low signal asserted by the PHY layer to indicate that in the current cycle there is no valid data for delivery to the ATM layer. For cell-level flow control, RxClav is an active high signal asserted by the PHY layer to indicate it is has a complete cell available for transfer to the ATM layer. In both cases, this signal indicates cycles when there is valid information on RxData/RxSOC.

RxClk
: Clock. Transfer/synchronization clock from the ATM layer to the PHY layer for synchronizing transfers on RxData.

The following signals are defined as optional (**O**) for the Receive interface.

RxPrty
: Parity. RxPrty is odd parity for RxData[7:0], driven by the PHY layer.

RxRef*
: Receive Reference. Output from the PHY layer for synchronization purposes (e.g. 8 kHz marker, frame indicator, etc.).

5.2 Operation and Timing

The Receive interface is controlled by the ATM layer. The ATM layer provides an interface clock to the PHY layer to synchronize all transfers. This convention requires the PHY layer to incorporate rate-matching buffers (i.e. a FIFO).

The receive interface has data flowing in the opposite direction to the ATM layer enable. The ATM receive block generates all output signals, and samples all input signals on the rising edge of RxClk.

Receive data is transferred from the PHY layer to ATM layer via the following procedure. The PHY layer indicates it has valid data, then the ATM layer asserts RxEnb* to read this data from the PHY layer. The PHY layer indicates valid data (thereby controlling the data flow) via the RxEmpty*/RxClav signal.

5.2.1 Octet-Level Handshake

The PHY layer indicates it has valid data in any cycle by deasserting RxEmpty*. The PHY layer may assert and deassert RxEmpty* at any time[5]. The ATM layer indicates that it wants to read PHY data by asserting RxEnb*. The ATM layer may assert and deassert RxEnb* at any time.[6]
The cycles during which RxEnb* is asserted constitute a *read window*. During a read window the PHY layer reads data from its internal FIFO and presents it on RxData/RxSOC on each low-to-high transition of RxClk.

Asserting RxEnb* while RxEmpty* is asserted is not an error but the value of RxData is undefined.

5.2.2 Cell-Level Handshake

The cell-level handshake is identical to the octet-level handshake except for one difference, namely that once RxClav is asserted, the PHY layer must be able to transfer a whole cell. RxClav has the same timing as RxEmpty*. This means that in the cycle following the one with the final octet of the cell, RxClav asserted indicates there is a new cell to transfer, while RxClav deasserted indicates there is no new cell to transfer.

5.2.3 Examples

The timing sequences may be summarized as: RxEnb* forces a data read from the PHY layer when RxEmpty* is deasserted, the PHY layer drives data during a cycle after one at which RxEnb* was asserted and RxEmpty* was deasserted.

Examples of Receive interface operation are shown in Figures 6 and 7. Figure 6 shows timing applicable to both octet and cell-level handshake methods. Figure 7 shows timing for RxEmpty*/RxClav boundary conditions.

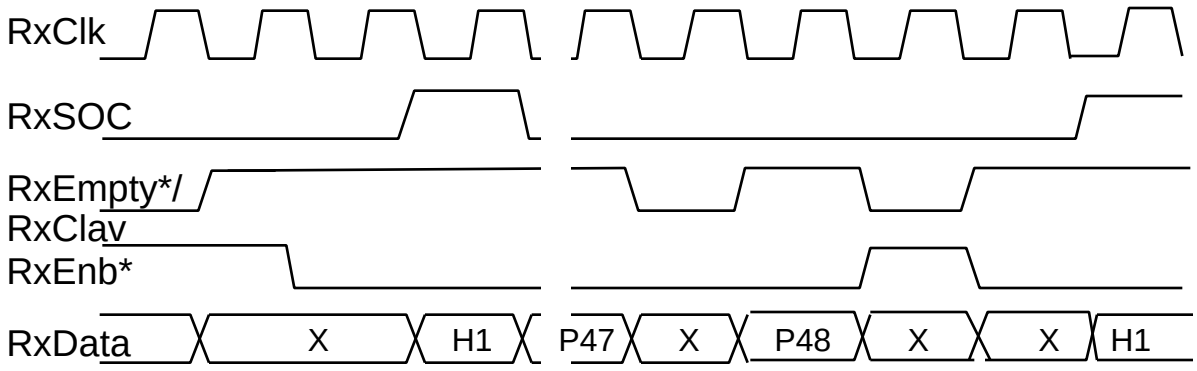

Figure 6. Receive Timing for Octet or Cell-Level Handshake

[5]An implementation could deassert RxEmpty* after several bytes are available (e.g. a cell header has been received and checked).
[6]This means that the ATM layer can choose to permanently assert RxEnb* if desired.

In Figure 6, the diagram for RxData shows the data value for cycles during which a transfer from PHY to ATM layer is effected. At the start (prior to H1) the “X” shows no data transfer because either RxEmpty* is asserted, or the ATM layer’s read window is not open (which starts 1 cycle after RxEnb* is asserted in this figure). Although the ATM layer is sampling between P47 and P48, no data transfer is effected because RxEmpty* is asserted. The same condition applies to the “X” immediately following P48. The second “X” after P48 represents no data transfer, in this case because the read window is closed due to the deassertion of RxEnb* in the previous cycle.

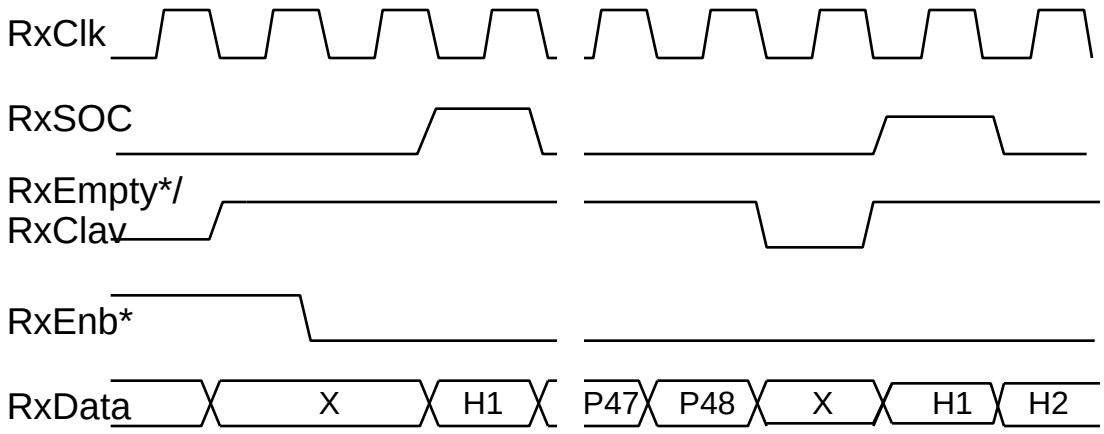

Figure 7. RxEmpty*/RxClav Timing

5.2.4 Timing Specifications

The following ATM and PHY Layer timing requirements/specifications apply to the Receive interface (note where RxData is referenced, the same timing applies to RxData, RxPrty and RxSOC).

Table 2. Receive Timing

Item	Description	Applies To	Min	Max
tR1	RxClk frequency	RxClk	0	25 MHz
tR2	RxClk duty cycle	RxClk	40 %	60 %
tR3	output delay from RxClk	RxEnb*	1 ns	20 ns
tR4	input setup to RxClk	RxData, RxEmpty*/RxClav, RxRef*	10 ns	
tR5	input hold from RxClk	RxData, RxEmpty*/RxClav, RxRef*	1 ns	
tR6	RxRef* high or low pulse width	RxRef*	1 RxClk	1 RxClk

6. 16-Bit Mode

6.1 Purpose

The standard 8-bit interface may be extended to a 16-bit data path. Accompanied with an increase in the transfer clock frequency, higher rates such as the 622 Mbps layer may be catered for. Guidelines for an extension to a 16-bit mode is defined in this section[7].

6.2 Cell Format

In 16-bit mode, 54-octet cells are transferred between ATM and PHY layers. As for 8-bit mode, a user-defined field is provided for backward compatibility. The byte arrangement is Big-Endian. Figure 8 shows the cell format for 16-bit mode.

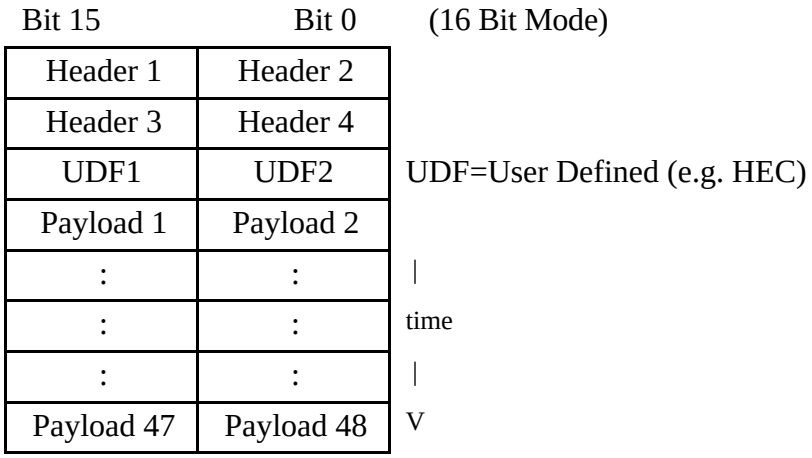

Figure 8. Cell Format in 16-bit mode

If the UDF field is utilized for the HEC, it is recommended that the HEC octet is carried in the UDF1 field. The UDF2 field may be used to provide control over the HEC for test purposes[8]. The field is also provided for backward compatibility to existing devices. HEC error statistics should be measured by PHY layers, and made accessible via a management interface register.

6.3 Signals

The 16-bit mode requires signals to be added to the 8-bit interface. These additional signals are defined below.

TxData[15:8]
: High octet of data. Most significant octet of transmit data, driven from ATM to PHY. TxData[15] is the MSB, TxData[0] is the LSB of the 16-bit data path.

TxPrty[1]
: Data path Parity. The 8-bit mode parity bit is named TxPrty[0] and combined with TxPrty[1] to provide 16-bit coverage. TxPrty[1] is the odd parity bit over TxData[15:8] and TxPrty[0] is the odd parity bit over TxData[7:0].

---

[7]It is recognized that this document (Level 1) does not fully address this issue; the text is for guidance only.
[8]This field allows either/both ATM/SAR to implement HEC, and can provide Header Control/Status functions.

RxData[15:8]

High octet of data. Most significant octet of receive data, driven from PHY to ATM. RxData[15] is the MSB, RxData[0] is the LSB of the 16-bit data path.

RxPrty[1]

Data path Parity. The 8-bit mode parity bit is named RxPrty[0] and combined with RxPrty[1] to provide 16-bit coverage. RxPrty[1] is the odd parity bit over RxData[15:8], and RxPrty[0] is the odd parity bit over RxData[7:0].

7. Summary

7.1 Transmit Interface Signals

Table 3. Transmit Interface Signals

Signal	Direction	Req./Opt	Description
TxData[7:0]	ATM to PHY	R	Data bus
TxData[15:8]	ATM to PHY	O	Data bus extension for 16-bit mode
TxPrty[1:0]	ATM to PHY	O	Data bus odd parity
TxSOC	ATM to PHY	R	Start Of Cell
TxEnb*	ATM to PHY	R	Enable data transfers
TxFull*/ TxClav	PHY to ATM	R	FIFO full/Cell Buffer Available
TxClk	ATM to PHY	R	Transfer/interface byte clock
TxRef*	ATM to PHY	O	Reference (e.g. 8 kHz)

7.2 Receive Interface Signals

Table 4. Receive Interface Signals

Signal	Direction	Req./Opt	Description
RxData[7:0]	PHY to ATM	R	Data bus
RxData[15:8]	PHY to ATM	O	Data bus extension for 16-bit mode
RxPrty[1:0]	PHY to ATM	O	Data bus odd parity
RxSOC	PHY to ATM	R	Start Of Cell
RxEnb*	ATM to PHY	R	Enable data transfers
RxEmpty*/ RxClav	PHY to ATM	R	FIFO empty/Cell Available
RxClk	ATM to PHY	R	Transfer/interface byte clock
RxRef*	PHY to ATM	O	Reference (e.g. 8 kHz)

