The ATM Forum

Technical Committee

Physical Layer Control

AF-PHY-0134.000

October, 1999

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Contact:

John Mick, Chair
Phone: +1 408-654-6721
Email: mick@idt.com

Alan Berenbaum, Editor
Phone: +1 908-582-3655
Email: adb@bell-labs.com

Table of contents

List of Figures

List of Tables

1 Introduction

1.1 Document Purpose and Background

This document describes several means of implementing a management interface for a *Universal Test & Operations PHY Interface for ATM* (UTOPIA) data path interface [1]. These implementations are not a mandatory part of any UTOPIA specification, but are provided as a guideline to further extend the scope of interoperability for PHY devices provided by UTOPIA specifications. ATM and PHY layers are fully conformant to the UTOPIA specifications if they implement the main parts of the UTOPIA specification, even if they incorporate management interfaces that differ from that described in this document

The management interfaces described here can be applied to a Level 1 UTOPIA interface, a Level 2 UTOPIA interface, or any higher level UTOPIA interface that may be approved in the future.

This document assumes familiarity with the Level 1 UTOPIA document [1] and the Level 2 UTOPIA document [2]. Sections 2, 3 and 4 are taken directly from Appendix 2 of the Level 2 UTOPIA document.

1.2 Scope

According to UTOPIA level 1 specification (V2.01, March 21, 1994), the management interface of a PHY device is connected to a Management Entity, as shown in Figure 1.1, and allows the management entity to access status and control information within the PHY device. The management interface may be a physical connection that is implemented via a serial or parallel hardware interface as suggested in sections 2 and 3.. Alternatively, the management interface may be a logical connection carried through the ATM layer on a particular virtual circuit. This allows the management entity to reside remotely from the PHY device and control multiple PHY devices. Section 5 of this document describes minimum requirements for an in-band message format carrying PHY device management information between the management entity and the PHY device.

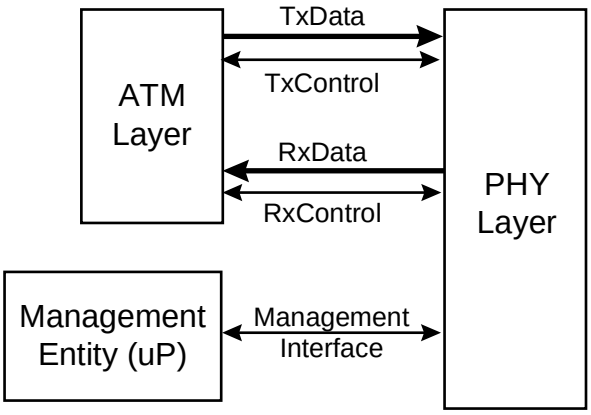

Figure 1.1: UTOPIA Interface Diagram

2 Serial interface

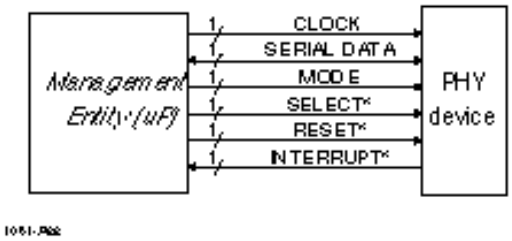

Figure 2.1: Serial management interface according to the UTOPIA Interface Diagram

2.1 Signals

The following signals are defined as required (**R**) for the serial management interface of a PHY device. Signals and their usage are described in Table 2.1. The terms uP⇒P and uP⇐P (column DIR, Direction) define an unidirectional signal from the microprocessor to the PHY device and vice versa. uP⇔P defines a bi-directional signal between the microprocessor and the PHY device.

Table 2.1: List of interface signals (serial interface)

Signal	DIR	Description
DATA	uP⇔P	Signal DATA carries the bit-serial data, address and two control bits. Any information is read in at the positive (rising) edge of CLOCK. Any information is output with the positive (rising) edge of CLOCK. For writing into the PHY device (control bits, address bits, write data) DATA is only valid if SELECT* is LOW at the same clock edge. For a data byte transfer out of the PHY device, DATA is valid at the next rising clock edge when SELECT* is LOW.
CLOCK	uP⇒P	Free running clock signal.
MODE	uP⇒P	Distinction between an address and control/data transfer. LOW: Control or data transfer HIGH: Address transfer.
SELECT*	uP⇒P	Signal SELECT* selects a PHY device and enables the transfer of address, control and data. LOW: PHY device is selected. HIGH: PHY device is not selected and ignores the signals DATA and MODE. PHY device holds its output driver of DATA in high impedance state (input mode).
RESET*	uP⇒P	LOW: PHY device is forced to reset. The output driver of DATA is set into its high impedance state (input mode). RESET must be LOW at least for one clock cycle. HIGH: PHY device does not reset.
INTR*	uP⇐P	Level sensitive interrupt signal generated by the PHY device. Open-drain active low output. INTR* must be LOW at least for one clock cycle.

SELECT, CLOCK and RESET must be driven HIGH or LOW by the microprocessor at any time. MODE and DATA may float as long as no PHY device is selected (SELECT* is HIGH, microprocessor may hold its output driver of DATA and MODE in high impedance state)

2.2 Operation

A transfer of data (transfer cycle) is initiated by asserting SELECT* to LOW. First, two control bits and (optional) the address information is transferred (written) into the PHY device. Second, data information is transferred (read/write). Thus, a transfer cycle is as follows:

* Asserting SELECT* from HIGH to LOW initiates the transfer cycle. The PHY device detects SELECT* to be LOW at the next rising (positive) clock edge (edge #1). MODE and DATA are don't care (may float) at clock edge #1.
* On the next rising clock edge (edge #2), the signal DATA defines the direction of the following data transfer by control bit C1 (control bit C1 LOW: read cycle, HIGH: write cycle). MODE should be LOW at this clock edge #2 if no new address is transferred in front of the first data transfer, otherwise MODE must be HIGH.
* On the next rising clock edge (edge #3), the signal DATA defines the address increment mode by control bit C2:
  - LOW: Automatic address increment mode.
    Multiple read/write byte operations are executed on successive byte locations (starting at a given address), e.g. used to read/write from/to successive configuration registers. The internal address register is automatically incremented after each data byte transfer. The PHY device must define its behavior in case of address wrap around. Read/write transfers from/to byte locations not implemented in the PHY device are dummy cycles.
  - HIGH: Single address mode.
    Multiple read/write operations are executed at the same address (e.g. read/write from/to a FIFO port).

  In any case, control bits C1 and C2 must be transferred. For a single data byte transfer, both address modes can be applied.

  At this considered clock edge #3, the PHY device may preset (preload) its internal address register to a predefined value (reload the address register with a new (fix or programmable) value). Otherwise the address register stays unchanged. Conditions for preloading is up to individual PHY device designs. This is independent whether the microprocessor writes address information in front of the first data transfer or not.

  If MODE is LOW at clock edge #2 or clock edge #3, no new address is written into the PHY device in front of the first data transfer. Data byte transfer starts after one additional clock cycle as described below.

  If MODE is HIGH at clock edge #2 and clock edge #3, the most significant bit of a new address is read in at the next rising clock edge (edge #4) by the PHY. In general, MODE must be HIGH for two clock cycles prior to any first transferred (most significant) address bit.
* Remaining address information is read in by the PHY device on the following rising clock edges as long as MODE is HIGH. The number of address bits written into the PHY device may be less then the size of the internal address register. The new received address information overwrites (least significant bit adjusted) the correspondent part of the address register. The number of address bits written into the PHY device may be more then the size of the internal address register. Only the least recently received address information overwrites (least significant bit adjusted) the address register.
* For read and write transfers, the output of data is started one clock cycle after MODE is detected to be LOW (there is always a one clock cycle gap between the last received address bit and the first (read/write) data bit). Then data bits are output and read in on every following rising clock edge. Data bytes are read/written in units of 8 data bits (most significant data bit first). Multiple data byte transfers are executed as long as SELECT* is LOW.
* SELECT* should stay LOW at least for one data byte transfer. SELECT* must go HIGH only on a data byte boundary to end a transfer. Reading/writing the least significant bit D0 of a data byte and detecting SELECT* to be HIGH terminates the transfer. If SELECT* is deasserted to HIGH not on a data byte boundary, the current transfer must be aborted immediately. Only data bytes which have been completely read/written from/to the PHY device are valid.
* SELECT* must be HIGH between two transfer cycles at least for one clock cycle.
* During a transfer cycle, a new address can be written into the PHY device in front of each data byte transfer. In case the PHY device reads/writes the least significant bit D0 of a data byte and detects MODE to be HIGH, the transfer of a new address into the PHY device is started after one additional clock cycle (note, MODE must be HIGH for two clock cycles prior to the first transferred address bit). After the address transfer, data byte transfer is resumed as described above.

2.3 Examples

In the following examples, it is assumed that a microprocessor reads/writes from/into a PHY device. All examples start with no data transfer being active.

2.3.1 Read Operation

Figure 2.2 shows an example of a single data byte read transfer, the transfer of three address bits and the preloading of the internal address register. The individual rising clock edges are numbered from 1 to 16. The following Table 2.1 describes the actions at these rising clock edges. The internal address register holds 55h (01010101b) from a previous transfer cycle. Furthermore, this examples shows MODE be either tri-stated or don't care if no PHY device is selected.

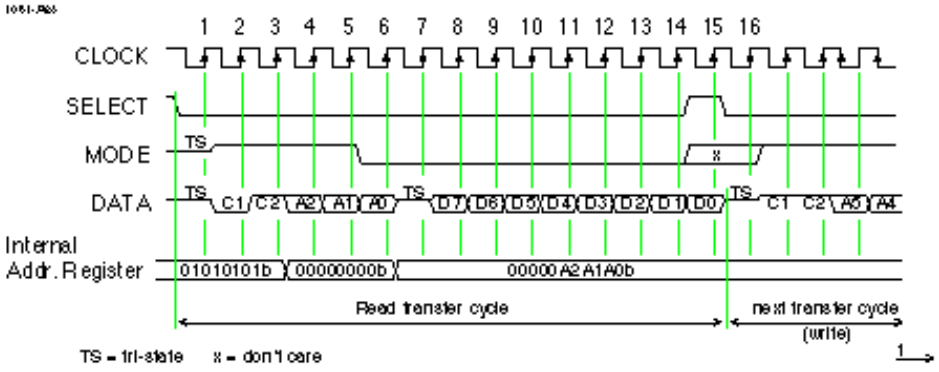

Figure 2.2: Single data byte read transfer

Table 2.2: Single data byte read transfer

Clock #	Description
1	The PHY device detects SELECT* to be LOW. It awaits the first control bit C1 at (next) rising clock edge #2. MODE and DATA are don't care at this clock edge. The Microprocessor outputs the first control bit (C1) and MODE with this rising clock edge.
2	The PHY device reads control bit C1. It is LOW, thus read cycle. The Micropro-cessor outputs the second control bit (C2) with this rising clock edge.
3	The PHY device reads in control bit C2. It is HIGH, thus single address mode. MODE is HIGH for two clock cycles, thus PHY device awaits first address bit at (next) rising clock edge #4. In this example, the internal address register is preloaded with 00000000b (00h) with this rising clock edge.
4,5,6	The PHY device reads in address bits A2, A1, A0.
6	The PHY device detects MODE to be LOW. Thus (next) rising clock edge #7 is not used and The PHY device will output the first data bit with rising clock edge #7, valid for the microprocessor at clock edge #8. The microprocessor must switch its output driver of DATA into high impedance state (input mode) with this rising clock edge #6. The received address bits A2, A1, A0 overwrite the three least significant bits of the address register. The internal address register now holds the new address 00000A2A1A0b.
7	The PHY device outputs data bit D7 with this rising clock edge (MSB first).
8	The microprocessor reads in data bit D7 at this clock edge. The PHY device outputs data bit D6 with this rising clock edge.
9, 10, 11 12, 13, 14	The microprocessor reads in data bits D6 to D1 at these clock edges. The PHY device outputs data bits D5 to D0 with these rising clock edges.
14	The microprocessor reads only one data byte, thus it deasserts SELECT* to HIGH with this clock edge.
15	The microprocessor reads in the data bit D0. The PHY device detects SELECT* to be HIGH. Thus, the PHY device ends transfer at this clock edge and switches its DATA output driver into a high impedance state (input mode).

Figure 2.3 shows an example of a multiple data bytes read transfer. The internal address register holds 3Ah from a previous transfer cycle. Control bit C2 is LOW, thus data bytes are read from successive byte locations inside the PHY device. Note, the end of the transfer is controlled by the microprocessor only by the signal SELECT. In this example, with rising clock edge #3, the internal address register is preloaded with 20h. Three address bits are written into the PHY device (A2,A1,A0 = 101b, 05h). Thus, with rising clock edge #6 the address register holds 25h. To address the successive bye locations, the internal address register is incremented (in this example) one clock cycle prior to the next data transfer.

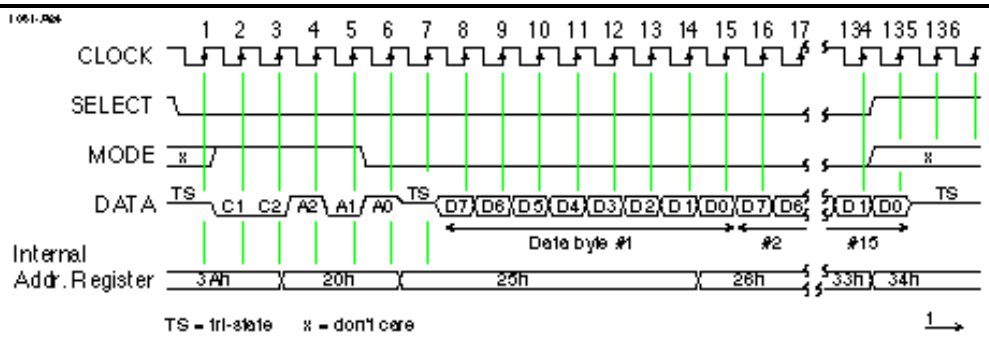

Figure 2.3: Multiple data byte read transfers (successive byte locations)

2.4 Write Operation

Figure 2.4 shows an example of a multiple data bytes write transfer, aborted by the microprocessor during the transfer of the second data byte. New address information is written into the PHY device between the first and the second data byte. The individual positive clock edges are numbered from 1 to 20. Table 2.2 describes the actions at these clock edges. The internal address register holds 41h from a previous transfer cycle. Control bit C2 is LOW, thus data bytes are written into successive byte locations inside the PHY device.

Note: Read and write transfer cycle timings are in principle the same. Except that phases marked x (don't care, DATA, Figure 2.4) in the write cycle are tri-stated in the read cycle. In the read cycle, this allows the switch over of the low impedance driver side from the microprocessor to the PHY device and vice versa.

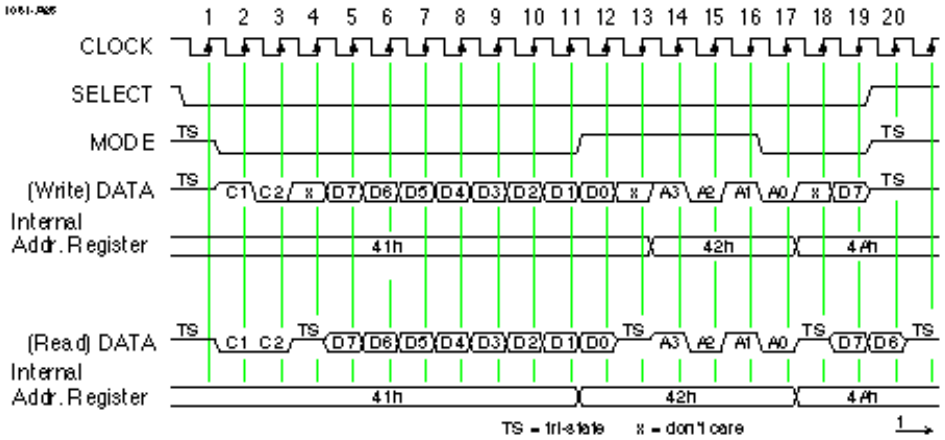

Figure 2.4: Multiple data byte write (read) transfer (aborted)

Table 2.3: Single data byte write transfer (part 1 of 2)

Clock #	Description
1	The PHY device detects SELECT* to be LOW. It awaits the first control bit C1 at (next) rising clock edge #2. MODE and DATA are don't care at this clock edge. The microprocessor outputs the first control bit (C1) and MODE with this rising clock edge.
2	The PHY device reads control bit C1. It is HIGH, thus write cycle. The micropro-cessor outputs the second control bit (C2) with this rising clock edge. MODE is LOW, thus no address in front of first data transfer.
3	The PHY device reads control bit C2. It is LOW, thus automatic address increment mode. MODE is LOW, thus no address in front of first data transfer. In this example, no preload of the internal address register at this clock edge. The microprocessor may output first data bit (MSB, D7) already with this rising clock edge.

4	Not used, DATA is don't care. The microprocessor must output first data bit (MSB, D7) latest with this rising clock edge.
5	The PHY device reads in data bit D7. The microprocessor outputs data bit D6 with this rising edge.
6,7,8,9, 10, 11	The PHY device reads in data bits D6 to D1 at these clock edges. The micropro-cessor outputs data bits D5 to D0 on these edges. With rising clock edge #11, MODE is set HIGH by the microprocessor, indicating an address transfer after this data byte transfer (detected by the PHY device at clock edge #12).
11	With rising clock edge #11, the PHY device increments the internal address register to be prepared for the next byte transfer (control bit C2 was LOW: automatic address increment mode for this transfer cycle).
12	The PHY device reads in data bit D0. The PHY device detects MODE to be HIGH. Thus, the PHY device prepares itself for an address transfer (it reads in new address starting with rising clock edge #14). The microprocessor may output the most significant address bit (in this example: A3) already with this rising clock edge.
13	DATA is don't care. The microprocessor must output most significant address bit (in this example: A3) latest with this rising clock edge.
13	PHY device: data is written in the into byte location at address 41h. To address the successive byte locations, the internal address register is incremented (in this example) one clock cycle after a complete data transfer.
14,15,16 17	The PHY device reads address bits A3, A2, A1, A0.
17	PHY device: received address bits A3, A2, A1, A0 overwrite the four least significant bits of the address register. Thus, the internal address register now holds new address 4Ah.
17	The microprocessor may output first data bit (D7, MSB) already with this rising clock edge.
18	Not used, DATA is don't care. The microprocessor must output first data bit (D7, MSB) latest with this rising clock edge.

Table 2.3: Single data byte write transfer (part 2 of 2)

Clock #	Description
19	The PHY device reads in data bit D7. The microprocessor aborts transfer cycle by deasserting SELECT* to HIGH with this rising clock edge. Thus, it outputs don't care instead of data bit D6 with this rising edge.
20	The PHY device detects SELECT* to be HIGH and aborts transfer too. No data is written into byte location 4Ah.

2.5 A.C. characteristics

The A.C. characteristics for the serial management interface are based on the timing specification for the receiver side of a signal, see Figure 2.5. This is analogous to the timing specification sections of UTOPIA specifications. The setup and the hold times are defined with regard to a rising (positive) clock edge. Taking the actual used clock frequency into account (e.g. up to the max. frequency), the corresponding (min. and max.) transmit side "clock to output" propagation delay specifications can be derived. The timing references (tT5 to tT12) are according to Tables 2.4 to 2.6.

The notes in the following sections 2.5.1 to 2.5.3 refer to section 4.5 of this document. The common conditions for the A.C. characteristics of the serial interface are described in section .4..

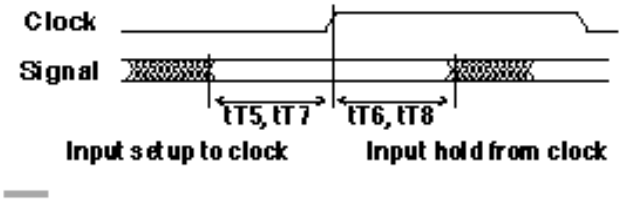

Figure 2.5: Setup and hold time definition (serial management interface)

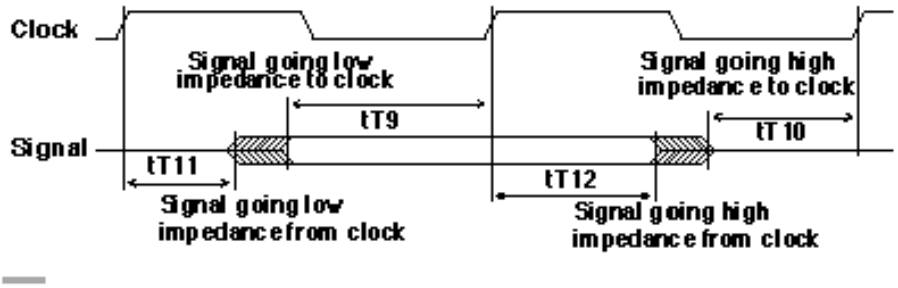

Figure 2.6: Tri-state timing definition (serial management interface)

2.5.1 25 MHz interface

Table 2.4: 25 MHz timing (serial interface)

Signal	DIR	Item	Description	Min.	Max.
CLOCK	uP⇒P	f1	TxClk frequency (nominal)	0	25 MHz
		tT2	TxClk duty cycle	40%	60%
		tT3	TxClk peak-to-peak jitter	-	5%
		tT4	TxClk rise/fall time	-	4ns
MODE, SELECT*, RESET*	uP⇒P	tT5	Input setup to CLOCK	10ns	-
		tT6	Input hold from CLOCK	1ns	-
INTR*	uP⇐P	tT7	Input setup to CLOCK	10ns	-
		tT8	Input hold from CLOCK	1ns	-
		tT9	Signal going low impedance to CLOCK	10ns	-
DATA	uP⇔P	tT10	Signal going high impedance to CLOCK(1)	0ns	-
		tT11	Signal going low impedance from CLOCK	1ns	-
		tT12	Signal going high impedance from CLOCK	1ns	-

2.5.2 33 MHz interface

Table 2.5: 33 MHz timing (serial interface)

Signal	DIR	Item	Description	Min.	Max.
CLOCK	uP⇒P	f1	TxClk frequency (nominal)	0	33 MHz
		tT2	TxClk duty cycle	40%	60%
		tT3	TxClk peak-to-peak jitter	-	5%
		tT4	TxClk rise/fall time	-	3ns
MODE, SELECT*, RESET*	uP⇒P	tT5	Input setup to CLOCK	8ns	-
		tT6	Input hold from CLOCK	1ns	-
INTR*	uP⇐P	tT7	Input setup to CLOCK	8ns	-
		tT8	Input hold from CLOCK	1ns	-
DATA	uP⇔P	tT9	Signal going low impedance to CLOCK	8ns	-
		tT10	Signal going high impedance to CLOCK(1)	0ns	-
		tT11	Signal going low impedance from CLOCK	1ns	-
		tT12	Signal going high impedance from CLOCK	1ns	-

2.5.3 50 MHz interface

Table 2.6: 50 MHz timing (serial interface)

Signal	DIR	Item	Description	Min.	Max.
CLOCK	uP⇒P	f1	TxClk frequency (nominal)	0	50 MHz
		tT2	TxClk duty cycle	40%	60%
		tT3	TxClk peak-to-peak jitter	-	5%
		tT4	TxClk rise/fall time	-	2ns
MODE, SELECT*, RESET*	uP⇒P	tT5	Input setup to CLOCK	4ns	-
		tT6	Input hold from CLOCK	1ns	-
INTR*	uP⇐P	tT7	Input setup to CLOCK	4ns	-
		tT8	Input hold from CLOCK	1ns	-
DATA	uP⇔P	tT9	Signal going low impedance to CLOCK	4ns	-
		tT10	Signal going high impedance to CLOCK(1)	0ns	-
		tT11	Signal going low impedance from CLOCK	1ns	-
		tT12	Signal going high impedance from CLOCK	1ns	-

3 Parallel interface

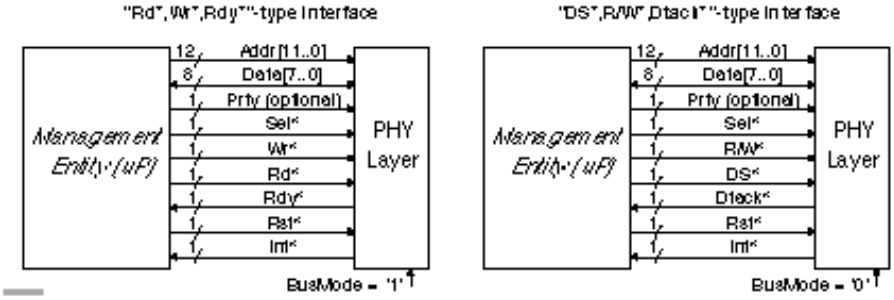

Figure 3.1: Parallel management interface according to the UTOPIA Interface Diagram

3.1 Signals

Table 3.1: List of optional interface signals (parallel interface)

Signal	DIR	Description
Prty	uP⇔P	Prty is optional (**O**) bidirectional odd Data parity.

Table 3.2: List of required interface signals (parallel interface)

Signal	DIR	Description
BusMode	uP⇒P uP⇒A	Input to ATM and PHY to select the mode of operation of the management interface. BusMode = '1' provides a <Rd*;Wr*;Rdy*> style (Intel compatible) interface; and BusMode = '0' provides a <DS*;R/W*;Dtack*> style (e.g. Motorola 68K, Intel i960 compatible) interface.
Addr[11..0]	uP⇒P	MSB is Addr[11]. The upper 64 bytes are reserved for up to 32 address pointers to registers describing each of the attached PHY devices.
Data[7..0]	uP⇔P	Byte-wide bidirectional data bus. MSB is Data[7].
Sel*	uP⇒P	Select. Active low enable signal used to validate the Addr bus for read and write transfers.
Rd* or DS*	uP⇒P	Read, or Data Strobe. If BusMode = '1', the active low Rd* input is LOW to enable read data from the addressed location onto the Data bus. If BusMode = '0' the active low DS* input is LOW to enable read data from the PHY layer, or strobe write data into the PHY layer.
Wr* or R/W*	uP⇒P	Write, or Read/Write. If BusMode = '1', the active low Wr* input is LOW to write data from the Data bus into the addressed location. If BusMode = '0', this input defines the access as a read if '1' or a write if '0'.
Rdy* or Dtack*	uP⇐P	Ready or Data Acknowledge. Tri-state acknowledge signal LOW to end data transfers over the Data bus. For either BusMode, Rdy*/Dtack* is LOW to complete a transfer.
Rst*	uP⇒P	Reset. Active low (level sensitive) input LOW to reset the PHY layer.
Int*	uP⇐P	Interrupt. Level sensitive interrupt signal LOW by the PHY layer. For either BusMode, Int* is an open-drain active low output.

For the parallel management interface of a PHY device, the signals in Table 3.2 are defined as required (**R**), the signals in Table 3.1 are defined as optional (**O**).

In Tables 3.1 and 3.2 the terms uP⇒P and uP⇐P (column DIR, Direction) define an unidirectional signal from the microprocessor to the PHY layer and vice versa. uP⇔P defines a bi-directional signal between the microprocessor and the PHY layer (the same applies for the ATM layer, uP⇒A).

3.2 Operation

3.2.1 Overview

This interface is designed to support most microprocessors. A BusMode pin selects one of two timing interfaces, Intel-compatible or Motorola-compatible. A ready/data_ack control is provided to support asynchronous or synchronous transfer cycles.
There are 3 control lines for effecting transfers. BusMode = '0' provides a read/write selector, a data strobe and a data acknowledge. BusMode = '1' provides a read strobe, a write strobe and a ready acknowledgement. The following text uses BusMode = '1' signal names, with BusMode = '0' names in parentheses.

3.2.2 Timing

For the parallel interface, the common conditions for the A.C. characteristics are described in section 4 of this document.

3.2.2.1 Write Cycles

The ATM layer performs write cycles by driving the address onto **Addr**, driving **Data** with the byte to be written, and asserting the appropriate strobes. The PHY layer will drive **Rdy*** (**Dtack*)** valid within the specified time, and assert **Rdy*** (**Dtack*)** to signal completion of the transfer. The PHY layer must tri-state **Rdy*** (**Dtack***) and **Data** whenever it is not selected.
For BusMode = '1' the data is strobed by asserting the **Wr*** signal, with **Rd*** HIGH. For BusMode = '0' the data is strobed by asserting **DS*** and setting **R/W*** to a '0'. The following PHY device timings in Table 3.3 apply to Figure 3.2.

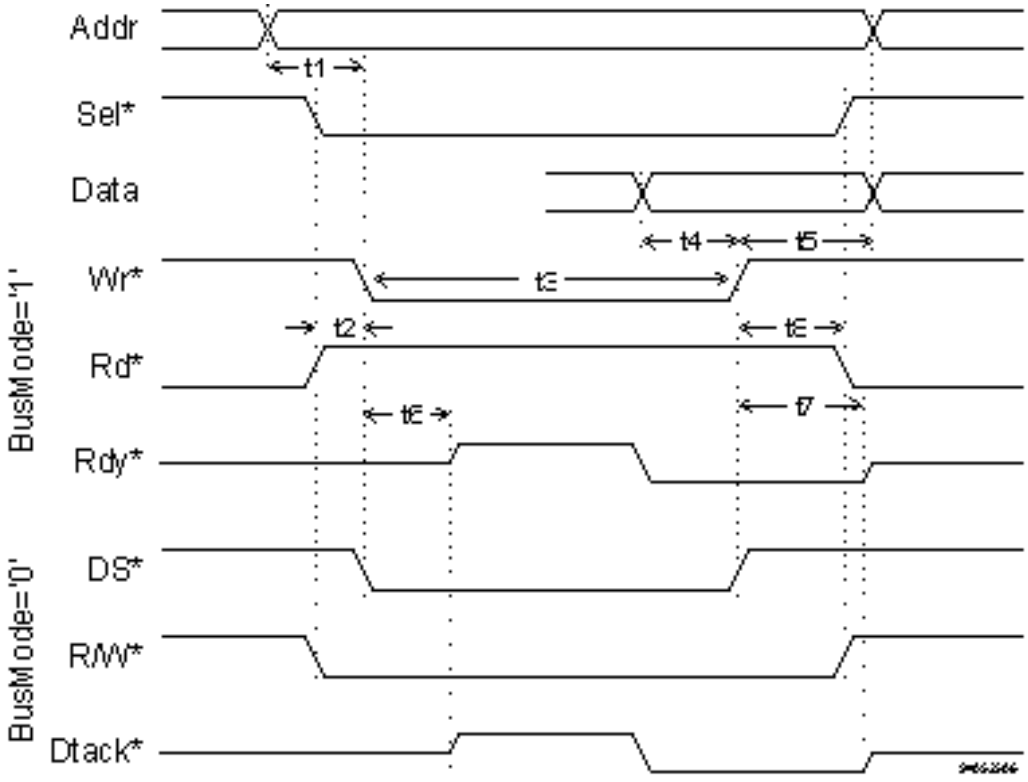

Figure 3.2: Single data byte write transfer (parallel interface)

Table 3.3: Transmit timing (parallel interface)

Item	Description	Min.	Max.
t1	Addr setup to Wr* (DS*) assertion	10 ns	
t2	Sel* (Sel*, R/W*) setup to Wr* (DS*) assertion	5 ns	
t3	Wr* (DS*) pulse width	50 ns	
t4	Data setup to Wr* (DS*) deassertion	15 ns	
t5	Addr, Data hold from Wr* (DS*) deassertion	4 ns	
t6	Rdy* (Dtack*) valid from Wr* (DS*) assertion		15 ns
t7	Rdy* (Dtack*) tri-state from Wr* (DS*) deassertion		10 ns
t8	Sel* (Sel*, R/W*) hold from Wr* (DS*) deassertion	0 ns	

3.3 Read Cycles

The ATM layer performs read cycles by driving the address onto **Addr**, and asserting **Sel*** and the appropriate strobes. The PHY layer will drive **Rdy*** (**Dtack*)** valid within the specified time, drive **Data** with the requested byte, and assert **Rdy*** (**Dtack*)** to signal completion of the transfer. The PHY layer must tri-state **Rdy*** (**Dtack*)** and **Data** whenever it is not selected.
For BusMode = '1' the data is strobed by asserting the **Rd*** signal, with **Wr*** HIGH. For BusMode = '0' the data is strobed by asserting **DS*** and setting **R/W*** to a '1'. The following PHY device timings in Table 3.4 apply to Figure 3.3.

Figure 3.3: Single data byte read transfer (parallel interface)

Table 3.4: Receive timing (parallel interface)

Item	Description	Min.	Max.
t1	Addr setup to Rd* (DS*) assertion	10 ns	
t2	Sel* (Sel*, R/W*) setup to Rd* (DS*) assertion	5 ns	
t3	Rd* (DS*) pulse width	50 ns	
t4	Data valid from Rdy* (Dtack*) assertion		10 ns
t5	Addr hold from Rd* (DS*) deassertion	4 ns	
t6	Rdy* (Dtack*) valid from Rd* (DS*) assertion		15 ns
t7	Rdy* (Dtack*) tri-state from Rd* (DS*) deassertion	10 ns	
t8	Sel* (Sel*, R/W*) hold from Rd* (DS*) deassertion	0 ns	
t9	Data invalid/tri-state from Rd* (DS*) deassertion	15 ns	

4 Common A.C characteristics and D.C. characteristics

4.1 Common parameters for A.C. characteristics

Table 4.1 lists the common parameters for the A.C. characteristics. Notes (3), (15), (16) are listed in section 4.3.

Table 4.1: Common parameters for A.C. characteristics

Parameter	Typ.	Min.	Max.
Input capacitance of input or input/output signal (pin)	10pF	-	
Input and output timing reference level	1.4V	-	-
Temperature range (Tambient) (15)	-	0 to +70°C	-
Power supply VDD (3)	+5V or +3,3V	-	-
Power supply Tolerance (16)	-	±5%	-

RxClk and TxClk peak-to-peak jitter, (for one cycle)	measured from one rising edge to the next rising edge
RxClk and TxClk rise time	measured at transmit side (driver), unloaded measured between 10% and 90% levels of VOH
RxClk and TxClk fall time	measured at transmit side (driver), unloaded measured between 90% and 10% levels of VOH

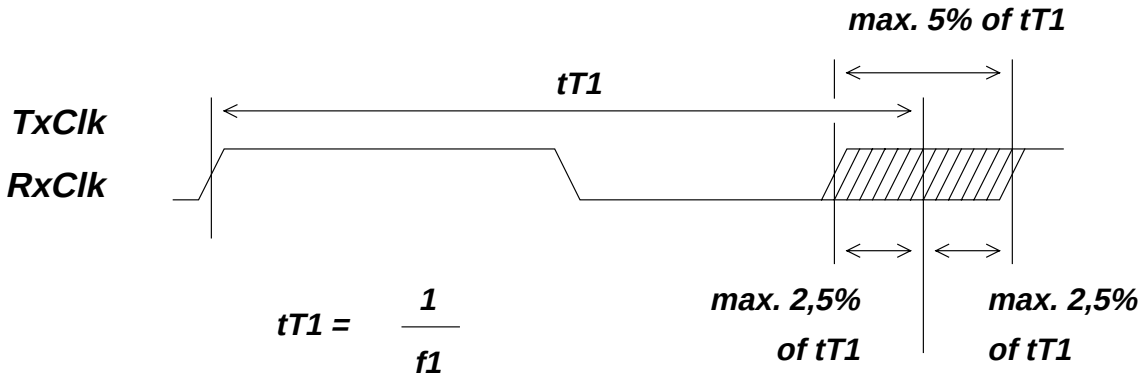

Figure 4.1: Clock peak-to-peak jitter

The following sections 4.2 and 4.3 list the D.C. characteristics for the serial and parallel management interface. The parameters assume, that the PHY device and management entity (microprocessor) are located close to each other, minimizing transmission line effects.

4.2 Serial interface

Table 4.2: D.C. characteristics (serial interface)

Symbol	Parameter	Min.	Max.	Conditions
VIL	Input LOW voltage	-0,3V	+0.8V	(3)
VIH	Input HIGH voltage	+2.0V	VDD + 0.3V	(4)
VOH	Output or bi-directional HIGH voltage	+2.4V	-	IOH ≥ |-4mA| (5)
VOL	Output or bi-directional LOW voltage	-	+0,5V	IOL ≥ +4mA (6)
IOH	Output current at HIGH voltage	-4mA	-	VOH ≥ +2.4V (7)
IOL	Output current at LOW voltage	+4mA	-	VOL ≤ +0.5V (8)
IIH	Input current at HIGH voltage	-	-	(9)
IIL	Input current at LOW voltage	-	-	(9)

4.3 Parallel interface

Table 4.3: D.C. characteristics (parallel interface)

Symbol	Parameter	Min.	Max.	Conditions
VIL	Input LOW voltage	-0,3V	+0.8V	(3)
VIH	Input HIGH voltage	+2.0V	VDD + 0.3V	(4)
VOH	Output or bi-directional HIGH voltage	+2.4V	-	IOH ≥ |-8mA| (10)
VOL	Output or bi-directional LOW voltage	-	+0,5V	IOL ≥ +8mA (11)
IOH	Output current at HIGH voltage	-8mA	-	VOH ≥ +2.4V (12)
IOL	Output current at LOW voltage	+8mA	-	VOL ≤ +0.5V (13)
IIH	Input current at HIGH voltage	-	-	(9)
IIL	Input current at LOW voltage	-	-	(9)

4.4 Common D.C. characteristics

Table 4.4: Common parameters for D.C. characteristics

Parameter	Typ.	Min.	Max.
Temperature range (Tambient) (14)	-	0 to +70°C	-
Power supply VDD (2)	+5V or +3,3V	-	-
Power supply Tolerance (15)	-	±5%	-

4.5 Notes

(1) This gives one clock period time for the active driver to switch in the high impedance state (signal going high impedance 0ns *in front of next* CLOCK).

(2) This specification defines either VDD = +5V or VDD = +3,3V. It assumes the same VDD on the transmit and receive side of a signal. It does not consider a mixed VDD (+5V/+3,3V) configuration.

(3) A PHY device with VIL, min. ≤ -0.3V is compliant to this specification.

(4) A PHY device with VIH, max. ≥ VDD+0.3V is compliant to this specification.

(5) Negative current flows out of the considered node (out of the PHY device pin). A PHY device with VOH, min. ≥ +2.4V (and IOH ≥ |-4mA|) is compliant to this specification.

(6) Positive current flows into the considered node (into the PHY device pin). A PHY device with VOL, max. ≤ +0.5V (and IOL ≥ +4mA) is compliant to this specification.

(7) Negative current flows out of the considered node (out of the PHY device pin). IOH, min. defines the minimal required IOH value for the driver.
A PHY device with IOH, min. ≥ |-4mA| (and VOH ≥ +2.4V) is compliant to this specification.

(8) Positive current flows into the considered node (into the PHY device pin). IOL, min. defines the minimal required IOL value for the driver.
A PHY device with IOL, min. ≥ +4mA (and VOL ≤ +0,5V) is compliant to this specification.

(9) To allow several technologies, no values for IIH and IIL are specified. An input can sink and source.

(10) Negative current flows out of the considered node (out of the PHY device pin). A PHY device with VOH, min. ≥ +2.4V (and IOH ≥ |-8mA|) is compliant to this specification.

(11) Positive current flows into the considered node (into the PHY device pin). A PHY device with VOL, max. ≤ +0.5V (and IOL ≥ +8mA) is compliant to this specification.

(12) Negative current flows out of the considered node (out of the PHY device pin). IOH, min. defines the minimal required IOH value for the driver.
A PHY device with IOH, min. ≥ |-8mA| (and VOH ≥ +2.4V) is compliant to this specification.

(13) Positive current flows into the considered node (into the PHY device pin). IOL, min. defines the minimal required IOL value for the driver.
A PHY device with IOL, min. ≥ +8mA (and VOL ≤ +0,5V) is compliant to this specification.
(14) A PHY device with a temperature range of at least 0 to +70°C is compliant to this specification.
(15) A PHY device with a power supply voltage tolerance ≥ 5% is compliant to this specification.

5 In-band PHY Management Message Format

An in-band management interface uses the ATM data stream across the UTOPIA port itself to communicate with the management agent in the PHY device. This is shown in Figure 5.1. A particular ATM cell header pattern identifies the management cell stream. These cells are filtered from the ATM cell stream and not passed over the physical media, but are routed to the PHY's internal management agent instead. Likewise, responses from the management agent are identified by the same ATM cell header and inserted in the return ATM cell stream. Such an in-band management interface allows remote configuration and supervision of PHY devices.

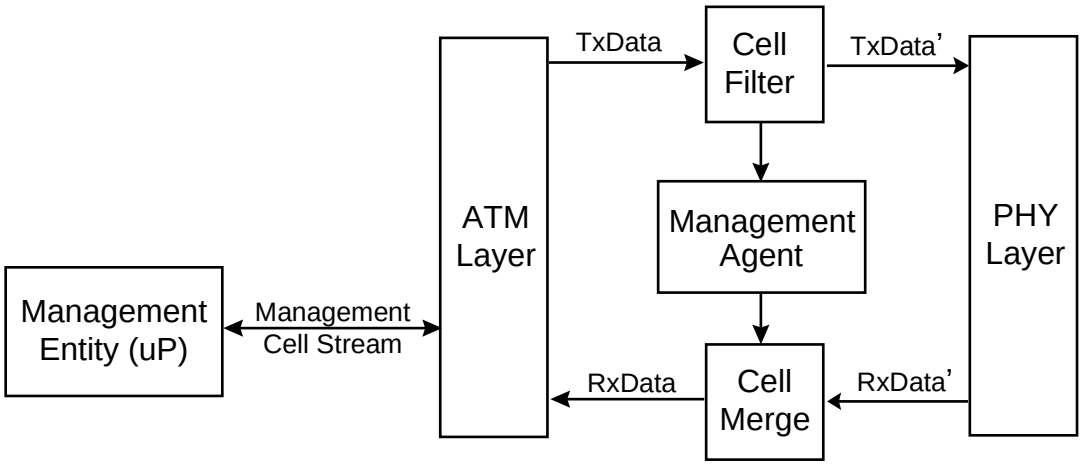

Figure 5.1: In-band PHY Management Interface

The in-band PHY management protocol is implemented via a sequence of messages between a management entity within the network and a management agent within the PHY device. Each PHY management message is carried in a single ATM cell. Single cell messages do not require SAR function in the PHY device and reduce the protocol's impact on the rest of the ATM network. PHY management messages are identified by an ATM header with a particular combination of VPI, VCI, and PTI values. Every message carries a CRC to prevent damage by corrupted messages. Any PHY management message with an invalid CRC is ignored.

The management entity originates a message and the PHY device responds with an acknowledging message. Each message is acknowledged, so the management entity can insure that its commands have been properly executed. Each message contains a transaction ID, so that acknowledgments can be correlated with the outstanding messages in the management entity. PHY devices are only required to process a single outstanding message at a time, because support for multiple outstanding messages would necessitate the burden of a queue in the PHY device. Vendor-specific extensions to this protocol may allow multiple outstanding messages, but the basic protocol only requires support for a single outstanding message. PHY devices busy processing a management message are free to ignore subsequent management messages until the initial acknowledge message is sent.

The single cell message format appears in Figure 5.2. It consists of two main parts: a 44-byte message body and a 4-byte message trailer. The message trailer is further partitioned into three fields: a 16-bit transaction ID, a 6-bit type field, and a 10-bit CRC.

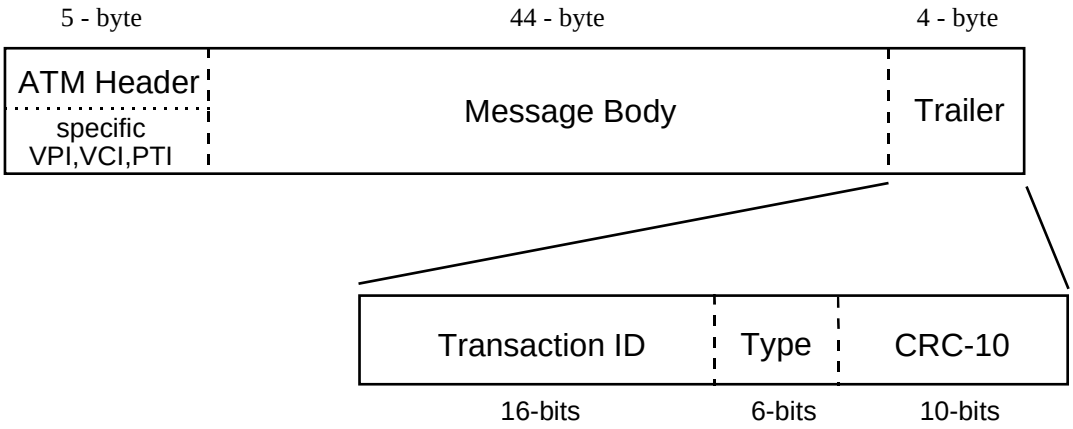

Figure 5.2: In-band PHY Message Format

The management cell header is defined to be an ATM cell header with head GFC = 0, VPI = 0, VCI = 0, CLP = 1 and PTI = 1. ATM cell headers with GFC = 0, VPI = 0, VCI = 0, and CLP = 1 are reserved for PHY layer use (as defined in I.610 and I.432.1). The management cell header pattern (with PTI = 1) is the same pattern as the F1 OAM cell header. Because F1 OAM cells are only defined to exist on the line side of the PHY device and should never appear on the drop (or ATM, i.e. UTOPIA) side of a device, they can be freely used by the in-band management interface. The management agent will absorb these cells, so they will never be passed through to the line side of the PHY device and disrupt normal F1flows between peer PHY devices. The management cell header is shown in Figure 5.3.

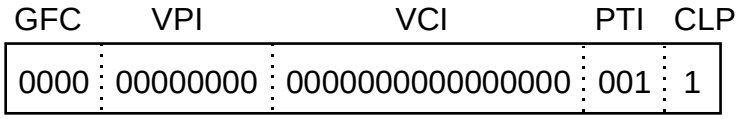

Figure 5.3: Management Cell Header

5.1 CRC Calculation

The 10-bit CRC is computed over the entire ATM cell payload excluding the last 10 bits. It is the remainder of the division (modulo 2) by the generator polynomial of the product of x^{10} and the content of the PHY management cell minus the 10-bit CRC field (namely the 44-byte message body, the 16-bit transaction ID, and the 6-bit format field, excluding the 10-bit CRC field) (374 bits). Each bit of the concatenated fields mentioned above is considered as a coefficient (modulo 2) of a polynomial of degree 373 using the first bit as coefficient of the highest order term. The CRC-10 generating polynomial is:

$$G(x) = 1 + x + x^4 + x^5 + x^9 + x^{10}$$

The result of the CRC calculation is placed with the least significant bit right justified in the CRC field. This calculation is the same CRC-10 as described in I.610 section 7.1.

5.2 Transaction ID

The 16-bit transaction ID field is initialized by the management entity to a unique value for each message. The management agent in the PHY device responds with an acknowledging message with the same 16-bit transaction ID. In this way, when the management entity receives an acknowledgment message, it can correlate it with the appropriate outstanding message. No glare situations arise since only the management entity generates transaction IDs.

5.3 Required Message Types

The 6-bit type field defines how the remaining bits in the message body are to be interpreted. Of the sixty-four possible message types, only one is defined for all devices. That is type = 0, the *Identify* message. The management entity uses the *Identify* message to determine the specific type of management agent it is communicating with and the corresponding PHY device it is controlling. The syntax and semantics of all other message types are device-specific and determined by the initial response to the *Identify* message.

5.3.1 *Identify* Message: Type = 0.

The *Identify* message is sent by the management entity when a physical device is first initialized in order to select the appropriate PHY management driver. The body of the outgoing *Identify* message contains no useful information. Only the message type field is significant.

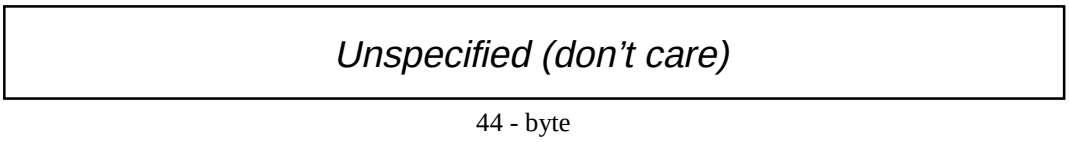

Figure 5.4: *Identify* Message Body

Upon receiving an *Identify* message, a PHY device will respond with an acknowledging *Identify* message. The type field of the acknowledging *Identify* message is the same as the *Identify* message, type = 0. Whether the message is a request for identification, or a response to such a request, the type field is the same. However, the message body of an acknowledging *Identify* message now has some useful information. The message body of a acknowledging *Identify* message contains a 3-byte organizationally unique identifier, or OUI; a 3-byte organization specific device identifier, and a 2-byte version specifier. The remainder of the message body is unspecified, though the contents may be device dependent.

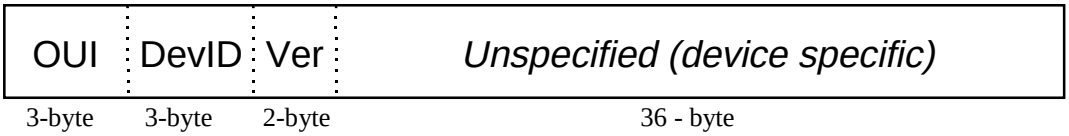

Figure 5.5: Acknowledging *Identify* Message

The IEEE[5] is the authority which assigns OUI values. Each organization is free to create device IDs to uniquely distinguish the types of PHY devices they manufacture. The version field is intended to distinguish different versions of the same PHY device type. The remainder of the acknowledging *Identify* message body may be freely used to convey device specific information.

The three byte OUI octet placement is as follows. Octet 0 of the OUI is octet 6 of the ATM cell. Octet 1 of the OUI is octet 7 of the ATM cell. Octet 2 of the OUI is octet 8 of the ATM cell. For clarification, the I/G bit of the OUI is the LSB of octet 6 of the ATM cell and the U/L bit is the second LSB of octet 6 of the ATM cell. A similar convention is used for the DevID. The most significant byte of the DevID is octet 9 of the ATM cell; the second most significant byte is octet 10 and the least significant byte is octet 11. Likewise, the most significant byte of the Ver is octet 12 in the ATM cell and the least significant byte is octet 13.

5.4 Example Message Types

Only the *Identify* message is required for interoperability, since with that message a management entity can determine the command set of a PHY device by table lookup. The message types in this section are provided only as an example of the types of messages that can be implemented for an in-band management protocol.

5.4.1 *Block Read* Message: Type = 1.

The *block read* message is used to transfer a block (40 bytes) of information from the PHY device to the management entity. The *block read* message specifies a 32-bit starting address of the 40 byte block of data. The address may point to internal dynamic state of the PHY device, or static information describing operation and use of the particular PHY device. The semantics of particular addresses is device specific.

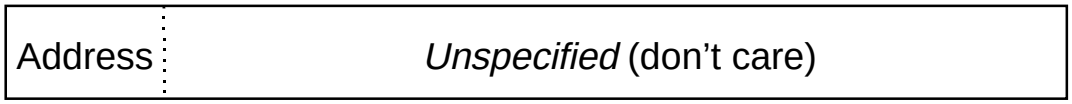

Figure 5.6: *Block Read* Message Body

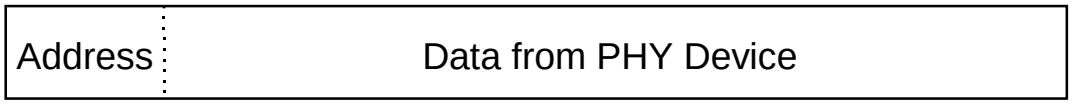

Figure 5.7: Acknowledging *Block Read* Message

5.4.2 *Block Write* Message: Type = 2.

The *block write* message is used to transfer a block (40 bytes) of information from the management entity to the PHY device. The *block write* message specifies a 32-bit starting address of the 40 byte block of data. The address may point to internal dynamic state of the PHY device, or static information describing operation and use of the particular PHY device. The *block write* message is useful for downloading programs to PHY devices. The semantics of particular addresses is device specific.

Figure 5.8: *Block Write* Message Body

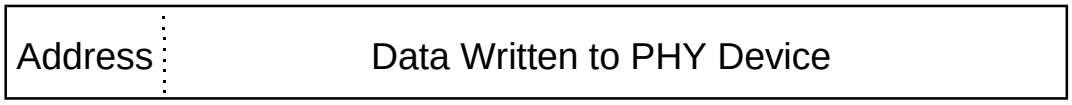

Figure 5.9: Acknowledging *Block Write* Message

Note that the acknowledging *block write* message returns the values of data written in the PHY device. The written data along with a valid CRC provide end to end verification that the correct data has been written to the PHY device.

5.4.3 *Register Read/Write* Message: Type = 3.

The *register read/write* message allows multiple registers within the PHY device to be read and written. Each register is specified by a 7-bit register address, allowing 128 addressable registers. The registers may be up to 24 bits wide. The semantics of particular register addresses is device specific. The *register read/write* message can access 11 independent registers in each message. The register read/write message body is comprised of eleven identical 4-byte commands. Each command specifies the operation, either read or write to be performed on a register and space to transfer 24-bits of data. Commands are processed sequentially left to right in the body of the message.

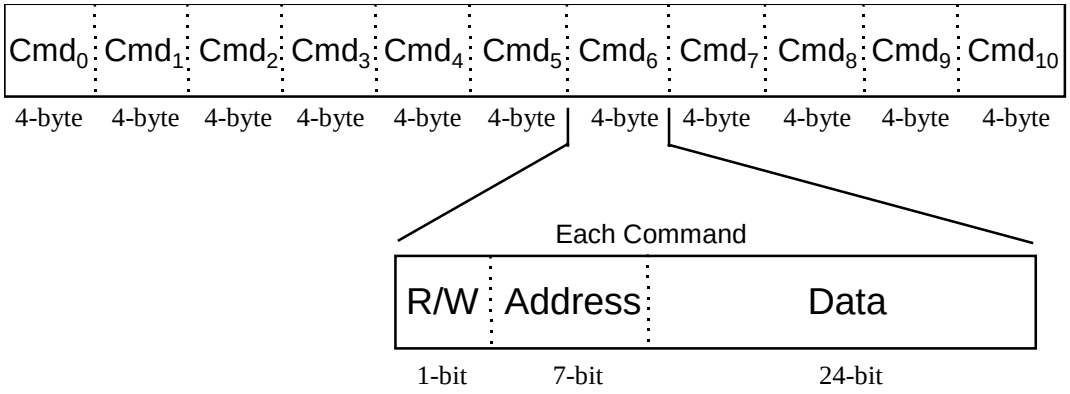

Figure 5.10: *Register Read/Write* Message

The R/W bit denotes whether the command reads (R/W = 1) or writes (R/W = 0) the register. If it is a register read command, the data field is unspecified for outgoing messages and contains the data read in the acknowledging messages. If it is a register write command, the data field contains the data to be written in the outgoing message and the data that was written in the acknowledging message. The acknowledging register read/write message returns a copy of the outgoing message body with the appropriate register read command data fields updated to the accessed registers. Returning the register addresses along with the written data and newly read data provides end to end verification.

6 Acronym List

CLP	Cell Loss Priority
CRC	Cyclic Redundancy Check
DevID	Device Identifier
GFC	Generic Flow Control
OAM	Operations And Maintenance
OUI	Organizationally Unique Identifier
PHY	Physical interface
PTI	Payload Type Identifier
SAR	Segmentation and Reassembly
UTOPIA	Universal Test & Operations PHY Interface for ATM
VCI	Virtual Channel Identifier
Ver	Version
VPI	Virtual Path Identifier

7 Normative References

[1] The ATM Forum, "UTOPIA, An ATM-PHY Interface Specification, Level 1, Version 2.01", March 21, 1994
[2] The ATM Forum, "UTOPIA Level 2, Version 1.0", June 1995
[3] I.432.1-1996, B-ISDN User-Network Interface: Physical Layer specification – General Characteristics
[4] I.610-1995, B-ISDN operation and maintenance principles and functions
[5] 802-1990, "IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture", May 31, 1990