6 Phy layer

6.1 Phy layer overview

The phy layer interfaces to the physical layer, performs 8b10b coding, and manages the phy reset sequence.

6.2 Encoding (8b10b)

6.2.1 Encoding overview

All data bytes transferred in SAS are encoded into 10 bit data characters using 8b10b coding. Additional characters not related to data bytes are called control characters.

All characters transferred in SAS are grouped into four byte sequences called dwords. A primitive is a dword whose first byte is a control character and remaining three bytes are data characters.

Primitives are defined with both negative and positive starting disparity. SAS defines numerous primitives starting with the K28.5 control character. Table 29 shows special character usage in SAS and SATA.

Table 29 — Special character usage

First character	Usage in SATA (informative)	Usage in SAS
K28.3	All primitives except ALIGN	Primitives used only inside STP connections
K28.5	ALIGN	ALIGN and all primitives defined in this standard
Dxx.y	Data	Data

Primitives are defined in ~~7.1~~7.1.

A data dword is a dword starting with a data character.

Disparity shall be maintained separately on each physical link. Expander devices shall convert incoming 10 bit characters to 8 bit bytes and generate the 10 bit character with correct disparity for the output physical link. Physical links may or may not begin operation with the same disparity after the reset sequence.

6.2.2 8b10b coding introduction

Information to be transmitted across a physical link shall be encoded eight bits at a time into an 10-bit transmission character and then transmitted serially bit-by-bit across the link. Information received over the link shall be collected ten bits at a time, and those transmission characters that are used for data, called data characters, shall be decoded into the correct eight-bit codes. The 10-bit transmission code supports all 256 eight-bit combinations. Some of the remaining transmission characters, referred to as special characters, are used for functions that are to be distinguishable from the contents of a frame.

The encodings defined by the transmission code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the special characters of the transmission code contain a distinct and easily recognizable bit pattern (a comma) which assists a receiver in achieving word alignment on the incoming bit stream.

6.2.3 8b10b encoding notation conventions

This subclause uses letter notation for describing information bits and control variables. Such notation differs from the bit notation specified by the remainder of this standard. The following text describes the translation process between these notations and provides a translation example. It also describes the conventions used to name valid transmission characters. This text is provided for the purposes of terminology clarification only.

An unencoded information byte is composed of eight information bits A, B, C, D, E, F, G, H and the control variable Z. This information is encoded into the bits a, b, c, d, e, i, f, g, h, j of a 10-bit transmission character.

An information bit contains either a binary zero or a binary one. A control variable has either the value D or the value K. When the control variable associated with an unencoded information byte contains the value D, that byte is referred to as a valid data byte. When the control variable associated with an unencoded information byte contains the value K, that byte is referred to as a special code.

Editor's Note 26: define MSB as bit 7 and LSB as bit 0 here?

The information bit labeled A corresponds to bit 0 in the numbering scheme of this specification, B corresponds to bit 1, and so on, as shown in table 30. The control variable is typically not specified. When the control variable is not specified, this standard assumes its value to be D (data).

Table 30 — Bit designations

Bit notation	7	6	5	4	3	2	1	0	Control variable
Unencoded bit notation	H	G	F	E	D	C	B	A	Z

Each valid transmission character has been given a name using the following convention:

Zxx.y

Where:

a) Z is the control variable of the unencoded information byte. The value of Z is used to indicate whether the transmission character is a data character (Z = D) or a special character (Z = K);
b) xx is the decimal value of the binary number composed of the bits E, D, C, B, and A of the unencoded information byte in that order; and
c) y is the decimal value of the binary number composed of the bits H, G, and F of the unencoded information byte in that order.

Table 31 shows the conversion from byte notation to the transmission character naming convention described above.

Table 31 — Conversion example

Byte notation	BCh
Bit notation	7 6 5 4 3 2 1 0 Control 1 0 1 1 1 1 0 0 K
Unencoded bit notation	H G F E D C B A Z 1 0 1 1 1 1 0 0 K
Unencoded bit notation reordered to conform with Zxx.y naming convention	Z E D C B A H G F K 1 1 1 0 0 1 0 1
Transmission character name	K 28 . 5

Most Kxx.y combinations do not result in valid transmission characters within the 8b10b transmission code. Only those combinations that result in special characters as specified by table 33 are considered valid.

6.3 Character encoding and decoding

6.3.1 Introduction

This subclause describes how to select valid transmission characters (encoding) and check the validity of received transmission characters (decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the document (i.e., primitives and frames).

6.3.2 Transmission order

Within the definition of the 8b10b transmission code, the bit positions of the transmission characters are labeled a, b, c, d, e, i, f, g, h, and j. Bit a shall be transmitted first, followed by bits b, c, d, e, i, f, g, h, and j, in that order. Bit i shall be transmitted between bit e and bit f, rather than in the order that would be indicated by the letters of the alphabet.

Characters within primitives shall be transmitted sequentially beginning with the special character used to distinguish the primitive (e.g., K28.5 or K28.3) and proceeding character by character from left to right within the definition of the primitive until all characters of the primitive are transmitted.

The contents of a frame shall be transmitted sequentially beginning with the primitive used to denote the start of frame (SOF delimiter) and proceeding character by character from left to right within the definition of the frame until the primitive used to denote the end of frame (EOF delimiter) is transmitted.

6.3.3 Valid and invalid transmission characters

6.3.3.1 Definitions

Table 32 and table 33 define the valid data characters (Dxx.y characters) and valid special characters (Kxx.y characters), respectively, and shall be used for both generating valid transmission characters (encoding) and checking the validity of received transmission characters (decoding). Each Valid-Data-Byte or special code entry has two columns that represent two (not necessarily different) transmission characters, corresponding to the current value of the running disparity (Current RD - or Current RD +). Running disparity is a binary parameter with either the value negative (-) or the value positive (+). The running disparity at the beginning of an primitive is the beginning running disparity (beginning RD).

After powering on, the transmitter shall initialize the Current RD to negative. Upon transmission of any transmission character, the transmitter shall calculate a new value for its running disparity based on the contents of the transmitted character.

After powering on or exiting diagnostic mode (the definition of diagnostic mode is beyond the scope of this standard), the receiver should assume either the positive or negative value for its initial running disparity. Upon reception of any transmission character, the receiver shall determine whether the transmission character is valid or invalid according to the following rules and shall calculate a new value for its running disparity based on the contents of the received character.

The following rules for running disparity shall be used to calculate the new running disparity value for transmission characters that have been transmitted (i.e. transmitter’s running disparity) and that have been received (i.e. receiver’s running disparity).

Running disparity for a transmission character shall be calculated on the basis of sub-blocks, where the first six bits (‘abcdei’ b) form one sub-block (six-bit sub-block) and the second four bits (‘fghj’ b) form the other sub-block (four-bit sub-block). Running disparity at the beginning of the six-bit sub-block is the running disparity at the end of the last transmission character. Running disparity at the beginning of the four-bit sub-block is the running disparity at the end of the six-bit sub-block. Running disparity at the end of the transmission character is the running disparity at the end of the four-bit sub-block.

Running disparity for the sub-blocks shall be calculated as follows:

a) Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111b, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011b.
b) Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000b, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100b.
c) Otherwise, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

All sub-blocks with equal numbers of zeros and ones are disparity neutral. In order to limit the run length of zeros or ones between sub-blocks, the 8b10b transmission code rules specify that sub-blocks encoded as 000111b or 0011b are generated only when the running disparity at the beginning of the sub-block is positive; thus, running disparity at the end of these sub-blocks shall also be positive. Likewise, sub-blocks containing 111000b or 1100b are generated only when the running disparity at the beginning of the sub-block is negative; thus, running disparity at the end of these sub-blocks shall also be negative.

Table 32 defines the valid data characters (Dxx.y characters).

Table 32 — Valid data characters (part 1 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D00.0	000 00000	100111 0100	011000 1011	D00.1	001 00000	100111 1001	011000 1001
D01.0	000 00001	011101 0100	100010 1011	D01.1	001 00001	011101 1001	100010 1001
D02.0	000 00010	101101 0100	010010 1011	D02.1	001 00010	101101 1001	010010 1001
D03.0	000 00011	110001 1011	110001 0100	D03.1	001 00011	110001 1001	110001 1001
D04.0	000 00100	110101 0100	001010 1011	D04.1	001 00100	110101 1001	001010 1001
D05.0	000 00101	101001 1011	101001 0100	D05.1	001 00101	101001 1001	101001 1001
D06.0	000 00110	011001 1011	011001 0100	D06.1	001 00110	011001 1001	011001 1001
D07.0	000 00111	111000 1011	000111 0100	D07.1	001 00111	111000 1001	000111 1001
D08.0	000 01000	111001 0100	000110 1011	D08.1	001 01000	111001 1001	000110 1001
D09.0	000 01001	100101 1011	100101 0100	D09.1	001 01001	100101 1001	100101 1001
D10.0	000 01010	010101 1011	010101 0100	D10.1	001 01010	010101 1001	010101 1001
D11.0	000 01011	110100 1011	110100 0100	D11.1	001 01011	110100 1001	110100 1001
D12.0	000 01100	001101 1011	001101 0100	D12.1	001 01100	001101 1001	001101 1001
D13.0	000 01101	101100 1011	101100 0100	D13.1	001 01101	101100 1001	101100 1001
D14.0	000 01110	011100 1011	011100 0100	D14.1	001 01110	011100 1001	011100 1001
D15.0	000 01111	010111 0100	101000 1011	D15.1	001 01111	010111 1001	101000 1001
D16.0	000 10000	011011 0100	100100 1011	D16.1	001 10000	011011 1001	100100 1001
D17.0	000 10001	100011 1011	100011 0100	D17.1	001 10001	100011 1001	100011 1001
D18.0	000 10010	010011 1011	010011 0100	D18.1	001 10010	010011 1001	010011 1001
D19.0	000 10011	110010 1011	110010 0100	D19.1	001 10011	110010 1001	110010 1001
D20.0	000 10100	001011 1011	001011 0100	D20.1	001 10100	001011 1001	001011 1001
D21.0	000 10101	101010 1011	101010 0100	D21.1	001 10101	101010 1001	101010 1001
D22.0	000 10110	011010 1011	011010 0100	D22.1	001 10110	011010 1001	011010 1001
D23.0	000 10111	111010 0100	000101 1011	D23.1	001 10111	111010 1001	000101 1001
D24.0	000 11000	110011 0100	001100 1011	D24.1	001 11000	110011 1001	001100 1001
D25.0	000 11001	100110 1011	100110 0100	D25.1	001 11001	100110 1001	100110 1001
D26.0	000 11010	010110 1011	010110 0100	D26.1	001 11010	010110 1001	010110 1001
D27.0	000 11011	110110 0100	001001 1011	D27.1	001 11011	110110 1001	001001 1001
D28.0	000 11100	001110 1011	001110 0100	D28.1	001 11100	001110 1001	001110 1001
D29.0	000 11101	101110 0100	010001 1011	D29.1	001 11101	101110 1001	010001 1001
D30.0	000 11110	011110 0100	100001 1011	D30.1	001 11110	011110 1001	100001 1001
D31.0	000 11111	101011 0100	010100 1011	D31.1	001 11111	101011 1001	010100 1001
D00.2	010 00000	100111 0101	011000 0101	D00.3	011 00000	100111 0011	011000 1100
D01.2	010 00001	011101 0101	100010 0101	D01.3	011 00001	011101 0011	100010 1100
D02.2	010 00010	101101 0101	010010 0101	D02.3	011 00010	101101 0011	010010 1100
D03.2	010 00011	110001 0101	110001 0101	D03.3	011 00011	110001 1100	110001 0011
D04.2	010 00100	110101 0101	001010 0101	D04.3	011 00100	110101 0011	001010 1100

Table 32 — Valid data characters (part 2 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D05.2	010 00101	101001 0101	101001 0101	D05.3	011 00101	101001 1100	101001 0011
D06.2	010 00110	011001 0101	011001 0101	D06.3	011 00110	011001 1100	011001 0011
D07.2	010 00111	111000 0101	000111 0101	D07.3	011 00111	111000 1100	000111 0011
D08.2	010 01000	111001 0101	000110 0101	D08.3	011 01000	111001 0011	000110 1100
D09.2	010 01001	100101 0101	100101 0101	D09.3	011 01001	100101 1100	100101 0011
D10.2	010 01010	010101 0101	010101 0101	D10.3	011 01010	010101 1100	010101 0011
D11.2	010 01011	110100 0101	110100 0101	D11.3	011 01011	110100 1100	110100 0011
D12.2	010 01100	001101 0101	001101 0101	D12.3	011 01100	001101 1100	001101 0011
D13.2	010 01101	101100 0101	101100 0101	D13.3	011 01101	101100 1100	101100 0011
D14.2	010 01110	011100 0101	011100 0101	D14.3	011 01110	011100 1100	011100 0011
D15.2	010 01111	010111 0101	101000 0101	D15.3	011 01111	010111 0011	101000 1100
D16.2	010 10000	011011 0101	100100 0101	D16.3	011 10000	011011 0011	100100 1100
D17.2	010 10001	100011 0101	100011 0101	D17.3	011 10001	100011 1100	100011 0011
D18.2	010 10010	010011 0101	010011 0101	D18.3	011 10010	010011 1100	010011 0011
D19.2	010 10011	110010 0101	110010 0101	D19.3	011 10011	110010 1100	110010 0011
D20.2	010 10100	001011 0101	001011 0101	D20.3	011 10100	001011 1100	001011 0011
D21.2	010 10101	101010 0101	101010 0101	D21.3	011 10101	101010 1100	101010 0011
D22.2	010 10110	011010 0101	011010 0101	D22.3	011 10110	011010 1100	011010 0011
D23.2	010 10111	111010 0101	000101 0101	D23.3	011 10111	111010 0011	000101 1100
D24.2	010 11000	110011 0101	001100 0101	D24.3	011 11000	110011 0011	001100 1100
D25.2	010 11001	100110 0101	100110 0101	D25.3	011 11001	100110 1100	100110 0011
D26.2	010 11010	010110 0101	010110 0101	D26.3	011 11010	010110 1100	010110 0011
D27.2	010 11011	110110 0101	001001 0101	D27.3	011 11011	110110 0011	001001 1100
D28.2	010 11100	001110 0101	001110 0101	D28.3	011 11100	001110 1100	001110 0011
D29.2	010 11101	101110 0101	010001 0101	D29.3	011 11101	101110 0011	010001 1100
D30.2	010 11110	011110 0101	100001 0101	D30.3	011 11110	011110 0011	100001 1100
D31.2	010 11111	101011 0101	010100 0101	D31.3	011 11111	101011 0011	010100 1100
D00.4	100 00000	100111 0010	011000 1101	D00.5	101 00000	100111 1010	011000 1010
D01.4	100 00001	011101 0010	100010 1101	D01.5	101 00001	011101 1010	100010 1010
D02.4	100 00010	101101 0010	010010 1101	D02.5	101 00010	101101 1010	010010 1010
D03.4	100 00011	110001 1101	110001 0010	D03.5	101 00011	110001 1010	110001 1010
D04.4	100 00100	110101 0010	001010 1101	D04.5	101 00100	110101 1010	001010 1010
D05.4	100 00101	101001 1101	101001 0010	D05.5	101 00101	101001 1010	101001 1010
D06.4	100 00110	011001 1101	011001 0010	D06.5	101 00110	011001 1010	011001 1010
D07.4	100 00111	111000 1101	000111 0010	D07.5	101 00111	111000 1010	000111 1010
D08.4	100 01000	111001 0010	000110 1101	D08.5	101 01000	111001 1010	000110 1010
D09.4	100 01001	100101 1101	100101 0010	D09.5	101 01001	100101 1010	100101 1010
D10.4	100 01010	010101 1101	010101 0010	D10.5	101 01010	010101 1010	010101 1010
D11.4	100 01011	110100 1101	110100 0010	D11.5	101 01011	110100 1010	110100 1010
D12.4	100 01100	001101 1101	001101 0010	D12.5	101 01100	001101 1010	001101 1010
D13.4	100 01101	101100 1101	101100 0010	D13.5	101 01101	101100 1010	101100 1010
D14.4	100 01110	011100 1101	011100 0010	D14.5	101 01110	011100 1010	011100 1010
D15.4	100 01111	010111 0010	101000 1101	D15.5	101 01111	010111 1010	101000 1010
D16.4	100 10000	011011 0010	100100 1101	D16.5	101 10000	011011 1010	100100 1010
D17.4	100 10001	100011 1101	100011 0010	D17.5	101 10001	100011 1010	100011 1010
D18.4	100 10010	010011 1101	010011 0010	D18.5	101 10010	010011 1010	010011 1010
D19.4	100 10011	110010 1101	110010 0010	D19.5	101 10011	110010 1010	110010 1010
D20.4	100 10100	001011 1101	001011 0010	D20.5	101 10100	001011 1010	001011 1010
D21.4	100 10101	101010 1101	101010 0010	D21.5	101 10101	101010 1010	101010 1010
D22.4	100 10110	011010 1101	011010 0010	D22.5	101 10110	011010 1010	011010 1010
D23.4	100 10111	111010 0010	000101 1101	D23.5	101 10111	111010 1010	000101 1010
D24.4	100 11000	110011 0010	001100 1101	D24.5	101 11000	110011 1010	001100 1010
D25.4	100 11001	100110 1101	100110 0010	D25.5	101 11001	100110 1010	100110 1010
D26.4	100 11010	010110 1101	010110 0010	D26.5	101 11010	010110 1010	010110 1010
D27.4	100 11011	110110 0010	001001 1101	D27.5	101 11011	110110 1010	001001 1010
D28.4	100 11100	001110 1101	001110 0010	D28.5	101 11100	001110 1010	001110 1010
D29.4	100 11101	101110 0010	010001 1101	D29.5	101 11101	101110 1010	010001 1010

Table 32 — Valid data characters (part 3 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D30.4	100 11110	011110 0010	100001 1101	D30.5	101 11110	011110 1010	100001 1010
D31.4	100 11111	101011 0010	010100 1101	D31.5	101 11111	101011 1010	010100 1010
D00.6	110 00000	100111 0110	011000 0110	D00.7	111 00000	100111 0001	011000 1110
D01.6	110 00001	011101 0110	100010 0110	D01.7	111 00001	011101 0001	100010 1110
D02.6	110 00010	101101 0110	010010 0110	D02.7	111 00010	101101 0001	010010 1110
D03.6	110 00011	110001 0110	110001 0110	D03.7	111 00011	110001 1110	110001 0001
D04.6	110 00100	110101 0110	001010 0110	D04.7	111 00100	110101 0001	001010 1110
D05.6	110 00101	101001 0110	101001 0110	D05.7	111 00101	101001 1110	101001 0001
D06.6	110 00110	011001 0110	011001 0110	D06.7	111 00110	011001 1110	011001 0001
D07.6	110 00111	111000 0110	000111 0110	D07.7	111 00111	111000 1110	000111 0001
D08.6	110 01000	111001 0110	000110 0110	D08.7	111 01000	111001 0001	000110 1110
D09.6	110 01001	100101 0110	100101 0110	D09.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 33 defines the valid special characters (Kxx.y characters).

Table 33 — Valid special characters

Special code name	Current RD - abcdei fghj	Current RD + abcdei fghj
K28.0	001111 0100	110000 1011
K28.1	001111 1001	110000 0110
K28.2	001111 0101	110000 1010
K28.3	001111 0011	110000 1100
K28.4	001111 0010	110000 1101
K28.5	001111 1010	110000 0101
K28.6	001111 0110	110000 1001
K28.7	001111 1000	110000 0111
K23.7	111010 1000	000101 0111
K27.7	110110 1000	001001 0111
K29.7	101110 1000	010001 0111
K30.7	011110 1000	100001 0111

Only K28.5 and K28.3 are used in this standard.

6.3.3.2 Generating transmission characters

The appropriate entry in the table shall be found for the valid data byte or special code for which a transmission character is to be generated (encoded). The current value of the transmitter's running disparity shall be used to select the transmission character from its corresponding column. For each transmission character transmitted, a new value of the running disparity shall be calculated. This new value shall be used as the transmitter's current running disparity for the next valid data byte or special code to be encoded and transmitted.

6.3.3.3 Validity of received transmission characters

The columns in table 32 and table 33 corresponding to the current value of the receiver's running disparity shall be searched for each received transmission character. If the received transmission character is found in the proper column, then the transmission character shall be considered valid and the associated data byte or special code determined (decoded). If the received transmission character is not found in the proper column, then the transmission character shall be considered invalid and the dword containing the character shall be considered an invalid dword. Independent of the transmission character's validity, the received transmission character shall be used to calculate a new value of running disparity.

This new value shall be used as the receiver's current running disparity for the next received transmission character.

Detection of a code violation does not necessarily indicate that the transmission character in which the code violation was detected is in error. Code violations may result from a prior error that altered the running

disparity of the bit stream but did not result in a detectable error at the transmission character in which the error occurred. The example shown in table 34 exhibits this behavior.

Table 34 — Delayed code violation example

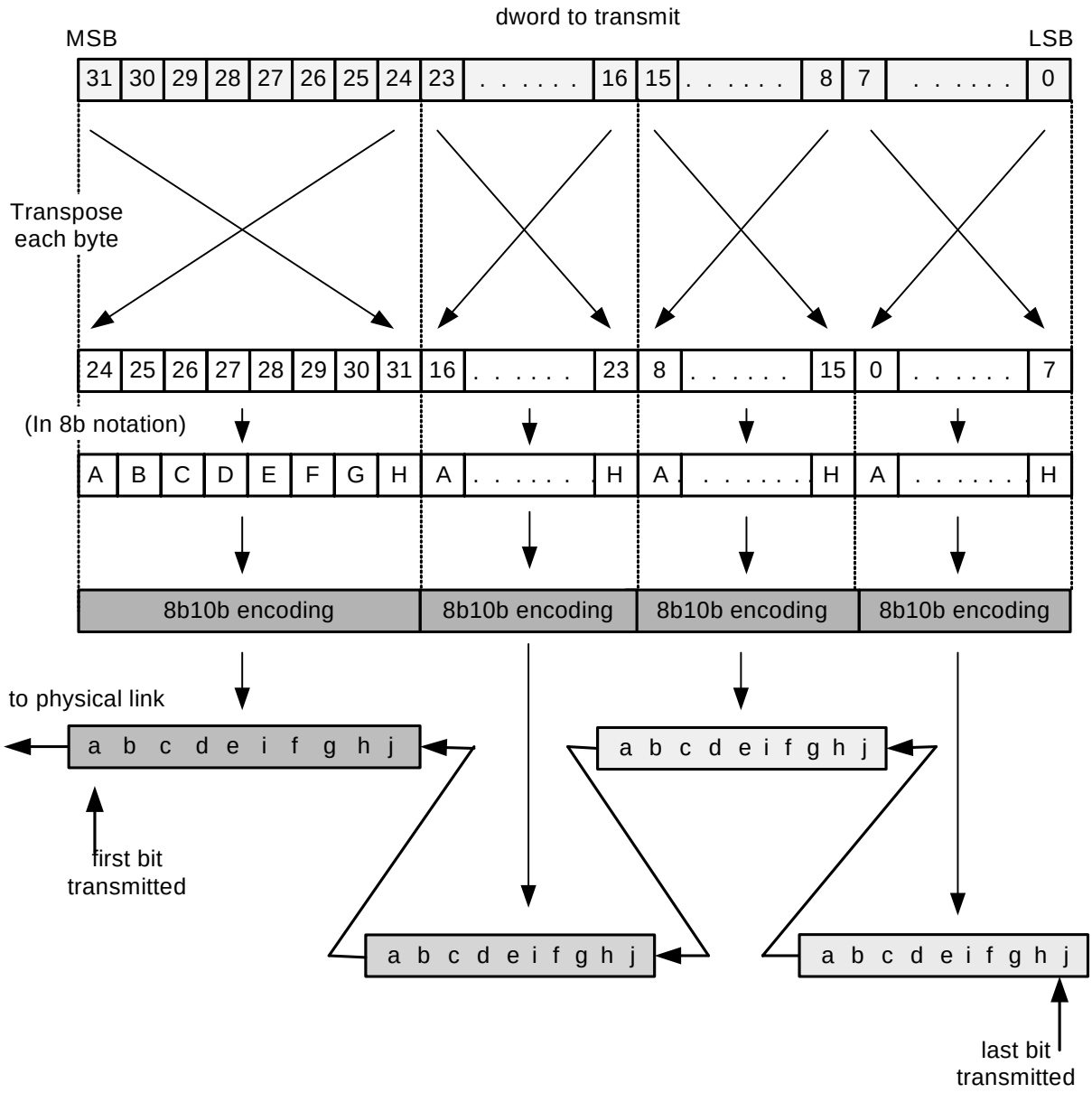
	RD	Character	RD	Character	RD	Character	RD
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Bit stream after error	-	101010 1011	+	010101 0101	+	111010 1010	+
Decoded character stream	-	D21.0	+	D10.2	+	Code violation	+

6.4 Bit order

Dwords transmitted in a STP connection shall be transmitted in the bit order specified by SATA.

Dwords for other types of connections and outside of connections shall be transmitted in the bit order in figure

40.


dword to transmit
MSB
LSB
31 30 29 28 27 26 25 24 23 . . . . . 16 15 . . . . . 8 7 0
Transpose each byte
24 25 26 27 28 29 30 31 16 . . . . . 23 8 . . . . . 15 0 7
(In 8b notation)
A B C D E F G H A . . . . . H A . . . . . H A . . . . . H
8b10b encoding
8b10b encoding
8b10b encoding
8b10b encoding
to physical link
a b c d e i f g h j
a b c d e i f g h j
a b c d e i f g h j
a b c d e i f g h j
first bit transmitted
last bit transmitted

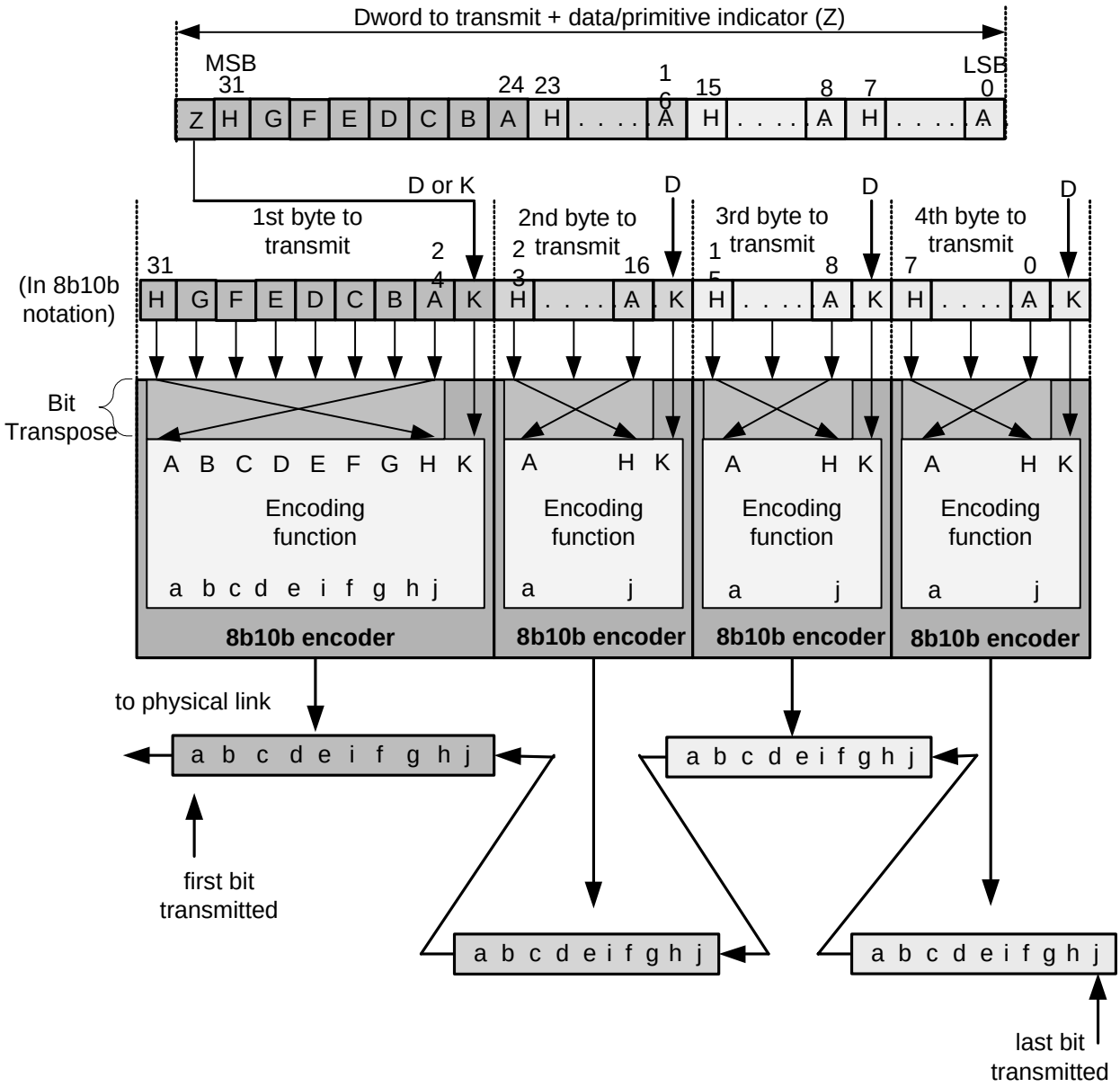
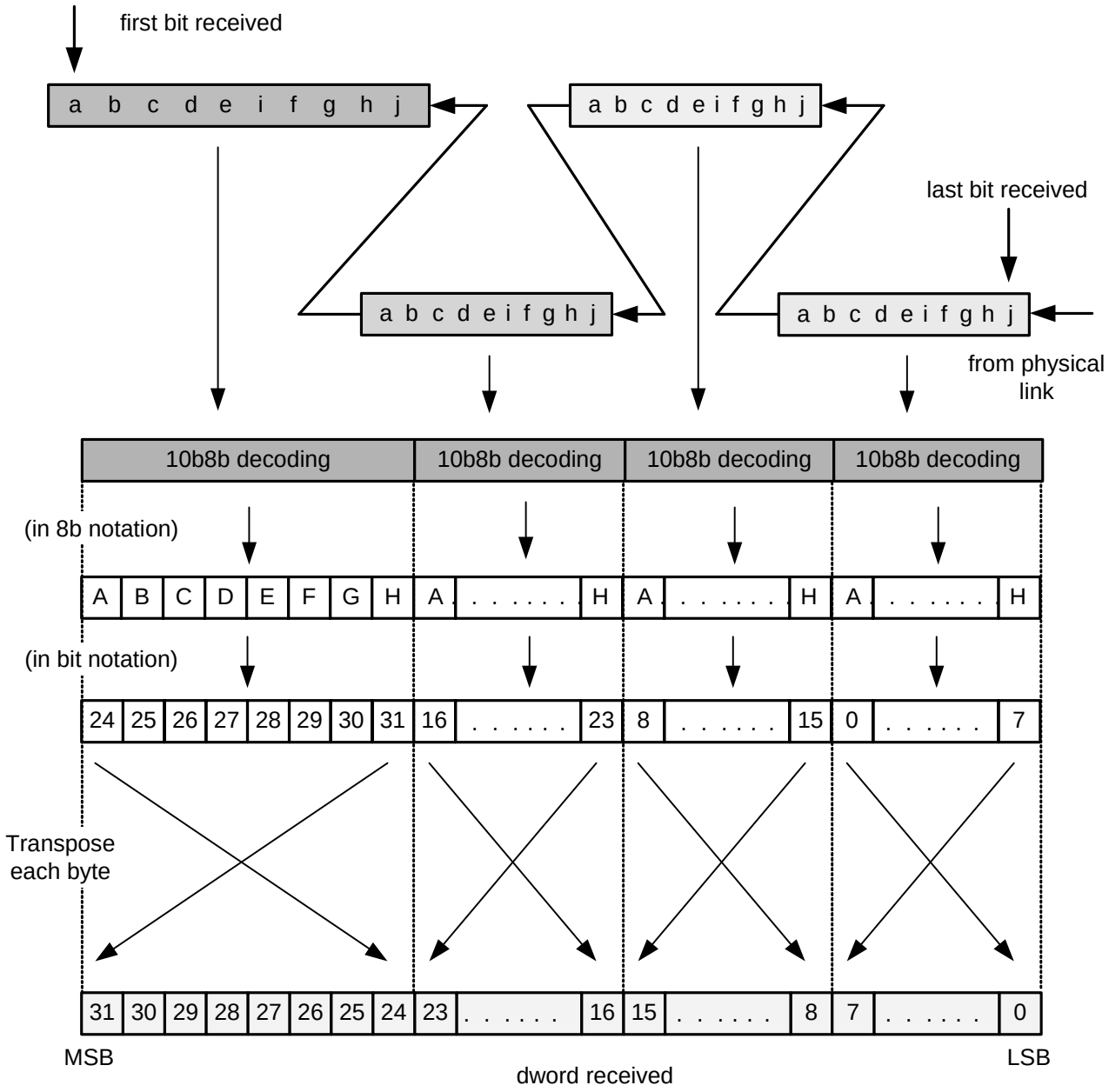


Figure 40 — SAS bit transmission logic

Figure 41 shows the SAS bit reception order.



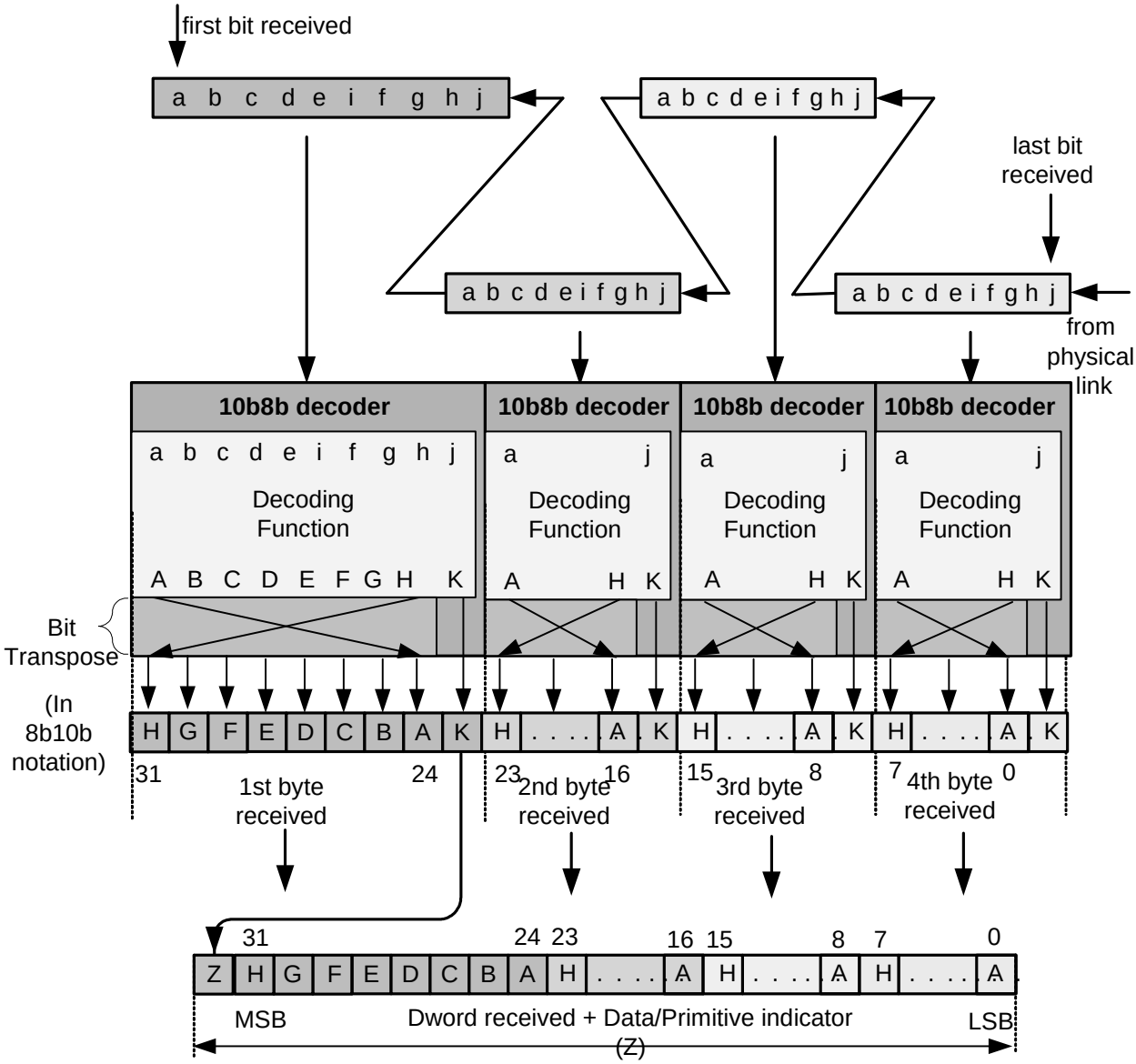


Figure 41 — SAS bit reception logic

6.5 Out of band (OOB) signals

Out of band (OOB) signals are low-speed signal patterns detected by the phy that do not appear in normal data streams. They consist of defined amounts of idle time followed by ALIGN bursts (repeated sequences of ALIGNs). During the burst time, ALIGN (0) primitives are transmitted repeatedly. During the idle time, the transmitter output shall meet the requirements described in ~~5.7.5~~5.7.5. The signals are differentiated by the length of idle time between the ALIGN bursts.

Editor's Note 27: change to "burst time" throughout rather than ALIGN burst throughout

SATA defines two OOB signals: COMINIT/COMRESET and COMWAKE. COMINIT and COMRESET are used in this standard interchangeably. SAS devices identify themselves with an additional SAS-specific OOB signal called COMSAS.

Table 35 defines the timing specifications for OOB signals.

Table 35 — OOB signal timing specifications

Parameter	Minimum	Nominal	Maximum	Comments
Unit interval during OOB (UI(OOB)) [a]	666,600 ps	666,667 ps	666,734 ps	The time basis for ALIGN bursts and idle times used to create OOB signals. Based on 1,5 Gbps clock tolerance.
COMSAS detect timeout	13,65 µs			The minimum time a receiver shall allow to detect COMSAS after transmitting COMSAS. Derived from: UI(OOB) x 512 x 40

[a] UI(OOB) is different than that defined in SATA; SAS has tighter clock tolerance. This is fixed value equal to the UI for G1, regardless of the actual transfer rate being used to create the burst time.

Editor's Note 28: work on UI(OOB) note to explain that it's not variable

Table 36 describes the OOB signal transmitter requirements for the burst time, idle time, and negation times that comprise each OOB signal.

Table 36 — OOB signal transmitter requirements

Signal	Burst time	Idle time	Negation time
COMWAKE	160 UI(OOB)	160 UI(OOB)	280 UI(OOB)
COMINIT/RESET	160 UI(OOB)	480 UI(OOB)	800 UI(OOB)
COMSAS	160 UI(OOB)	1 440 UI(OOB)	2 400 UI(OOB)

To transmit an OOB signal, a transmitter shall repeat these steps six times:

1) transmit idle for an idle time; and
2) transmit an ALIGN burst.

It shall then transmit idle for an OOB signal negation time.

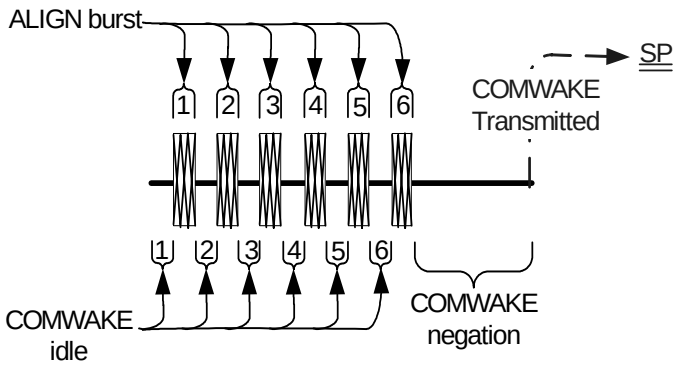
The ALIGNs used in OOB signals are not required to be at generation 1 (G1) rates (i.e., 1,5 Gbps), as this rate may not be supported in future generations of SAS devices. The ALIGNs are only required to generate an envelope for the detection circuitry, as required for any signaling that may be AC coupled. If G2 ALIGNs are used, the number of ALIGNs doubles compared with G1 ALIGNs.

A SAS transmitter should transmit ALIGNs at the G1 rate to create the burst portion of the OOB signal, but may transmit ALIGNs at its slowest supported rate if it does not support the G1 rate and shall not transmit them at a rate faster than its slowest supported rate.

Figure 42 describes OOB signal transmission by the SP transmitter. The COMWAKE Transmitted, COMINIT

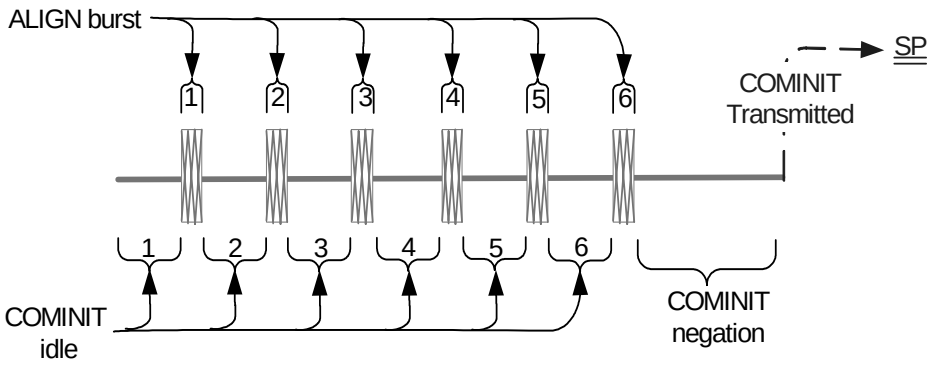
Transmitted, and COMSAS Transmitted parameters are sent to the SP state machine.

COMWAKE

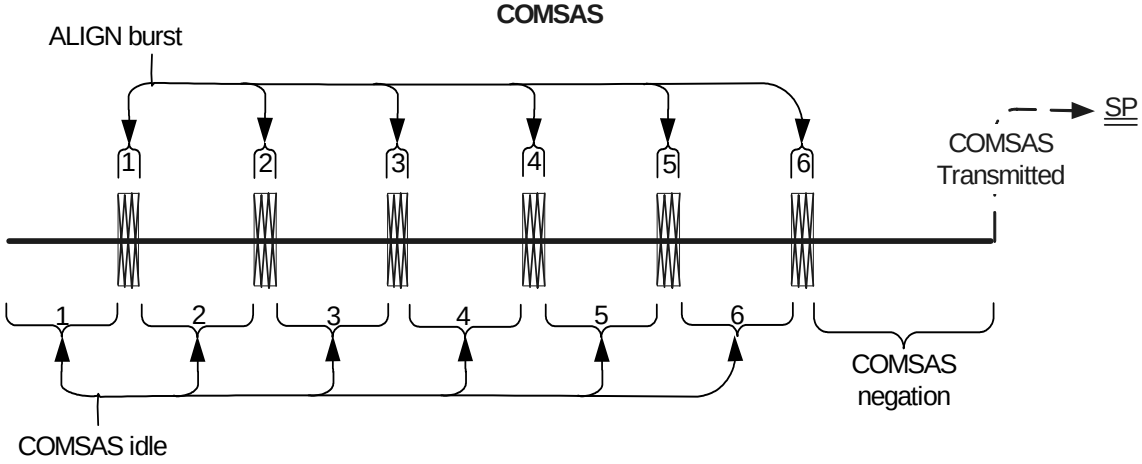


Note: Idle time is shown here as a neutral signal for visual clarity only.

COMRESET/COMINIT



COMSAS



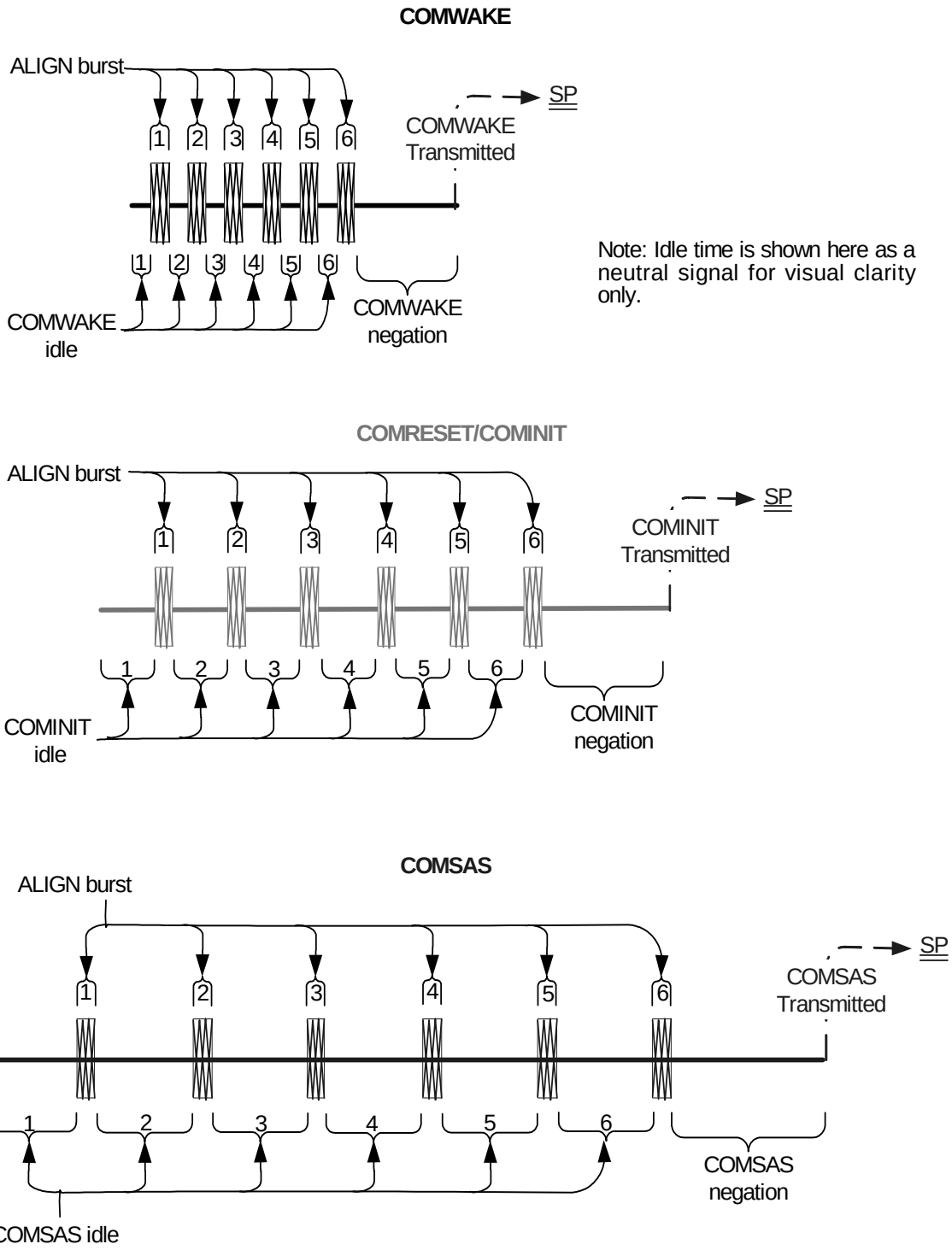


Figure 42 — OOB signal transmission

Table 37 describes the OOB signal receiver requirements for the idle times and negation times. The burst time is not used to distinguish between signals.

Table 37 — OOB signal receiver requirements

Signal	Idle time			Negation time
	may detect	shall detect	shall not detect	shall detect
COMWAKE	55 ns to 175 ns	101,3 ns to 112 ns	< 55 ns or > 175 ns	175 ns
COMINIT/ COMRESET	175 ns to 525 ns	304 ns to 336 ns	< 175 ns or > 525 ns	525 ns
COMSAS	525 ns to 1 575 ns	911,7 ns to 1008 ns	< 525 ns or > 1 575 ns	1 575 ns

A receiver shall detect an OOB signal after receiving four consecutive idle time/ALIGN burst pairs (see figure 43). It is not an error to receive more than six idle time/ALIGN burst pairs. A receiver shall not detect the same OOB signal again until it has detected a different OOB signal (e.g., if the idle time changes) or has detected lack of transitions for a time greater then the proceeding idle time (i.e., a COMINIT negation time for a COMINIT idle time or a COMSAS negation time for a COMSAS idle time).

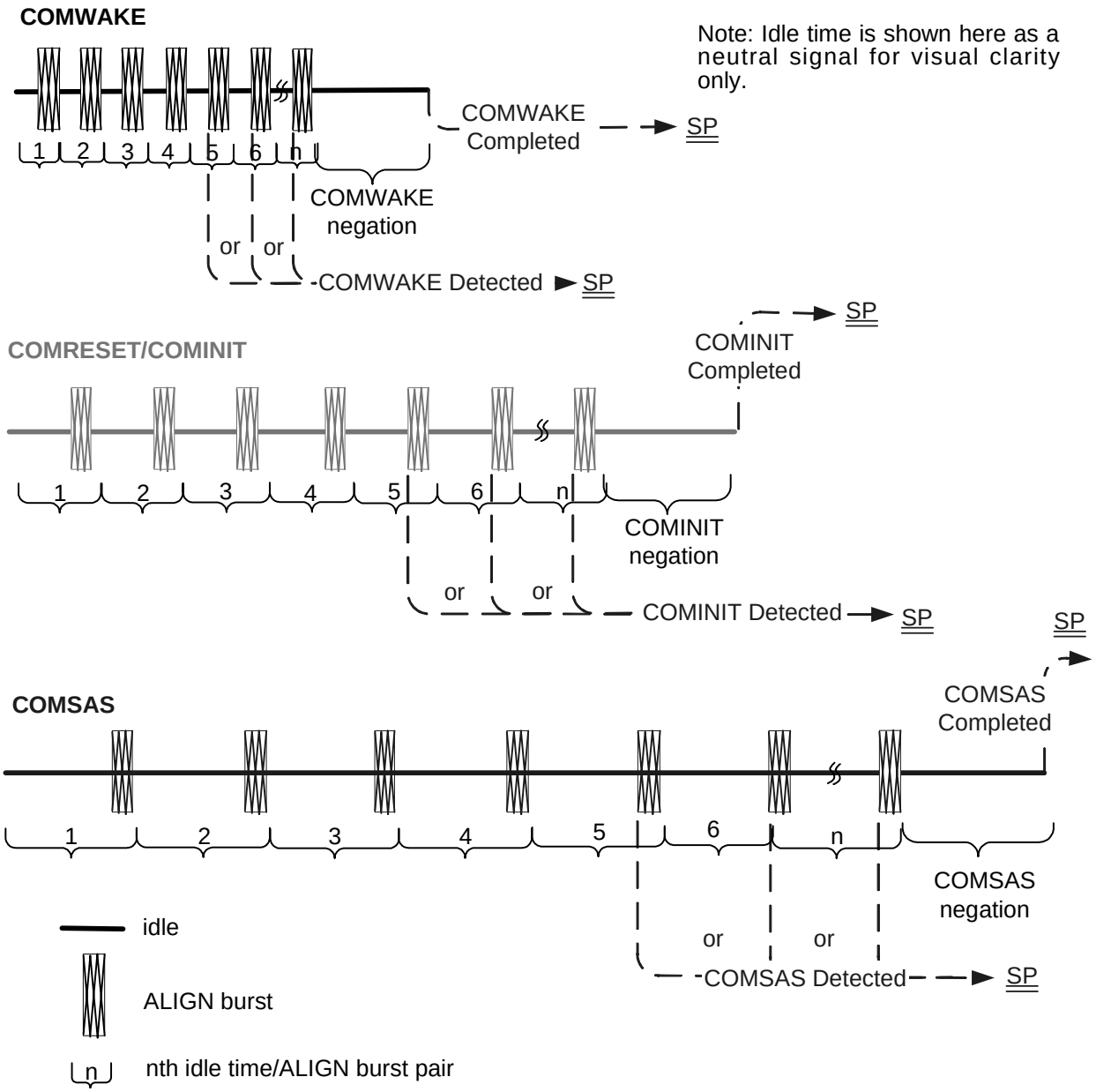
Editor's Note 29: might be best to have error for COMSAS to exceed 6 but allow COMINIT and COMWAKE to exceed...

A SAS receiver shall detect OOB signals comprised of ALIGNs transmitted at any rate up to its highest supported rate. This includes rates below its lowest supported rates (e.g., a SAS receiver supporting only 3,0 Gbps needs to detect 1,5 Gbps based ALIGNs to interoperate with a SAS transmitter supporting both 1,5 Gbps and 3,0 Gbps).

Figure 43 describes SAS OOB signal detection by the SP receiver. The COMWAKE Detected, COMWAKE Completed, COMINIT Detected, COMINIT Completed, COMSAS Detected, and COMSAS Completed

parameters are sent to the SAS phy (SP) state machine to indicate that an OOB signal has been partially or

fully detected.



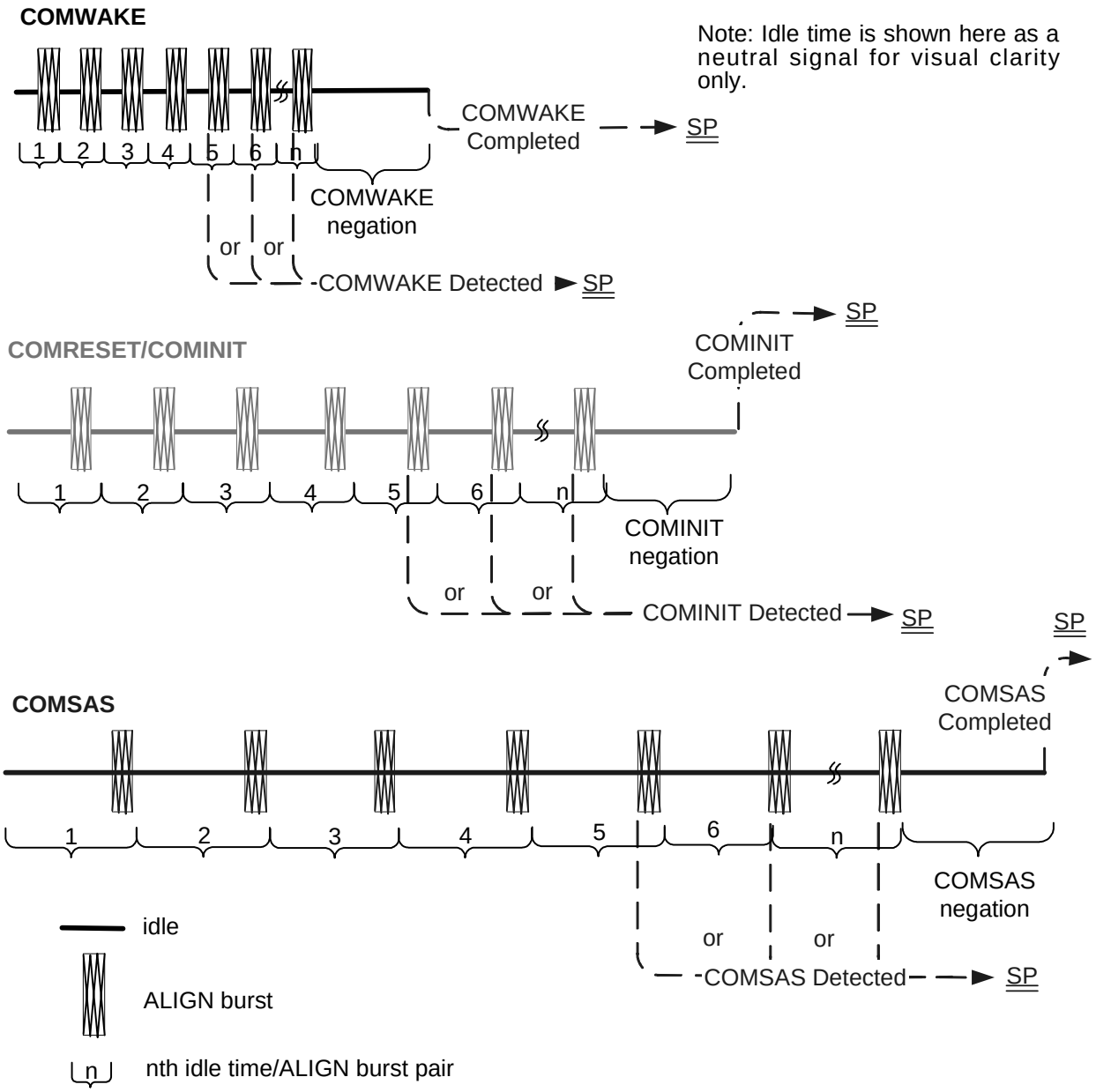


Figure 43 — OOB signal detection

Expanders shall not pass OOB signals. An expander device shall run the link reset sequence independently on each physical link (i.e., from initiator phy to expander phy, expander phy to expander phy, or expander phy to target phy).

6.6 Phy reset sequences

6.6.1 Overview

The phy reset sequence consists of an OOB sequence and a speed negotiation sequence.

The SAS phy reset sequence shall only affect the phy, not the port or device containing the phy or other phys in the same port or device.

After a HARD_RESET, a device should start the phy reset sequence within 250 ms.

6.6.2 SATA phy reset sequence (informative)

6.6.2.1 SATA OOB sequence (informative)

Figure 44 shows the SATA OOB sequence between a SATA initiator device (i.e., SATA host) and target device (i.e, SATA device). The SATA OOB sequence is defined by SATA; see SATA for detailed requirements.

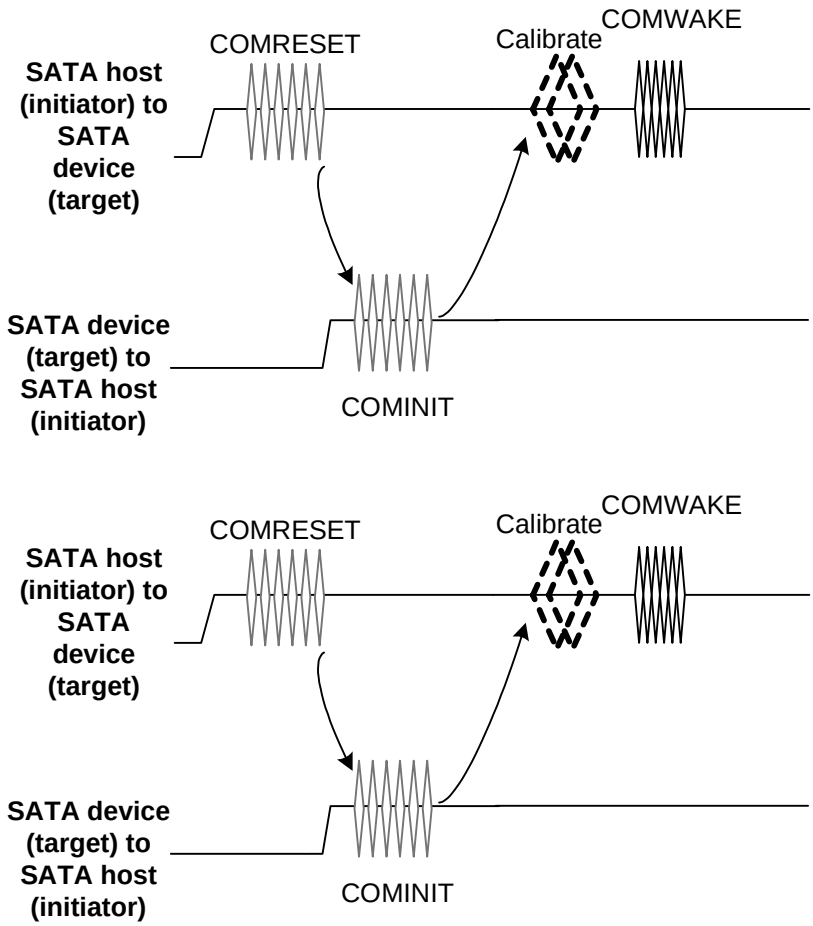


Figure 44 — SATA OOB sequence (informative)

6.6.2.2 SATA speed negotiation sequence (informative)

Figure 45 shows the speed negotiation sequence between a SATA initiator device and target device. The SATA speed negotiation sequence is defined by SATA; see SATA for detailed requirements.

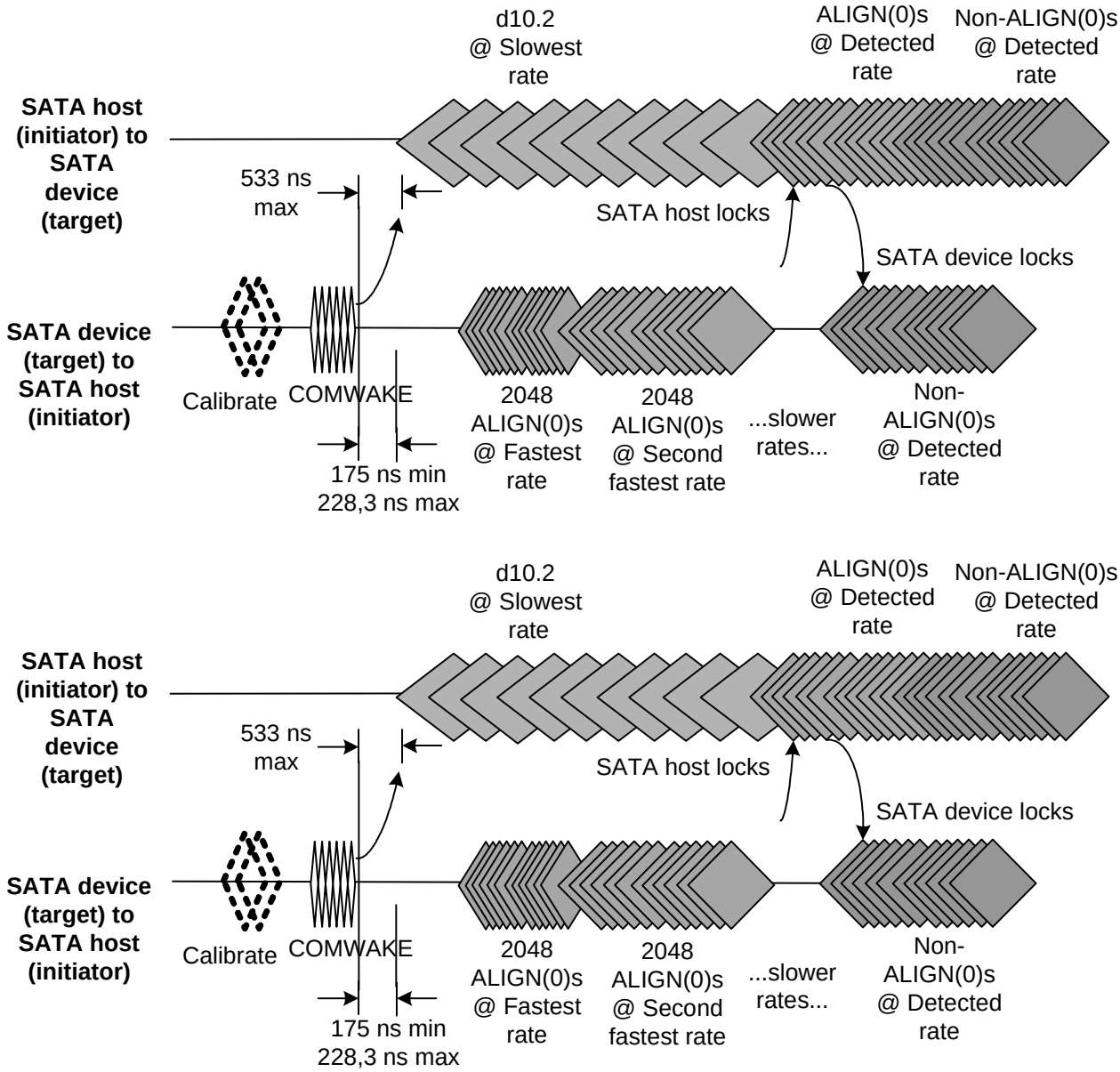


Figure 45 — SATA speed negotiation sequence (informative)

6.6.3 SAS to SATA phy reset sequence

SAS initiator devices and expander devices may be attached to SATA target devices.

To initiate a phy reset sequence a SAS phy shall:

1) transmit a COMINIT; and
2) in response to a COMINT, transmit a COMSAS.

The COMSAS identifies the phy as a SAS phy instead of a SATA phy for SAS to SAS attachments.

If a SATA phy is attached to the link it either:

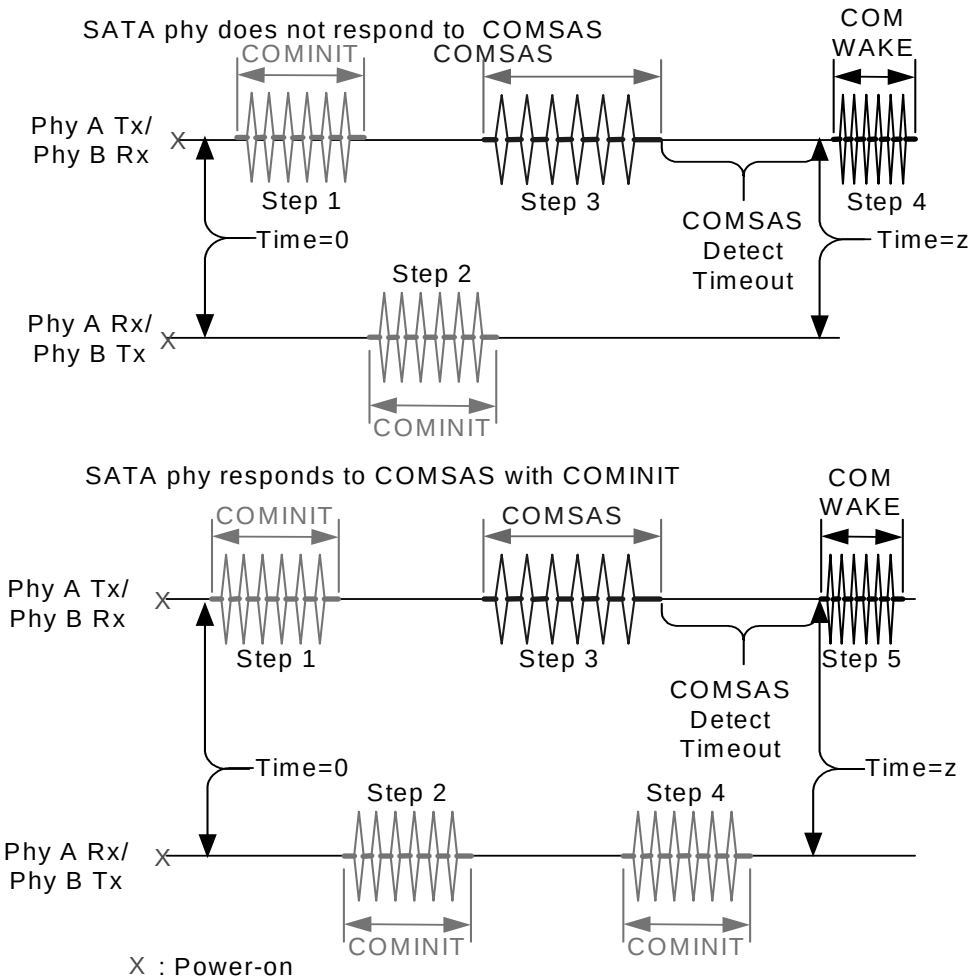
a) misinterprets the COMSAS to be a COMRESET and responds with a COMINIT; or
b) ignores the COMSAS and provides no response within a COMSAS detect timeout.

Either response indicates to the SAS phy that a SATA phy is attached. As a result the SAS phy shall initiate a SATA reset sequence (i.e., transmit a COMRESET).

Figure 46 shows a reset sequence between a SAS phy and a SATA phy (i.e., between an expander device and a SATA target device, or between a SAS initiator device and a SATA target device). The two possible cases are presented. The first case is that the SATA phy ignores the COMSAS and provides no response within a COMSAS detect timeout. The second case is that a legacy SATA phy misinterprets the COMSAS to be a COMRESET and responds with a COMINIT. The SAS phy state machine treats these two cases the

same, and determines that a SATA phy is attached after a COMSAS detect timeout. The normal SATA reset

sequence shall be entered starting with COMWAKE.



X : Power-on
Time 0 : SAS phy reset sequence begins
Time z : Post-COMRESET/COMINIT SAS-SATA reset sequence begins

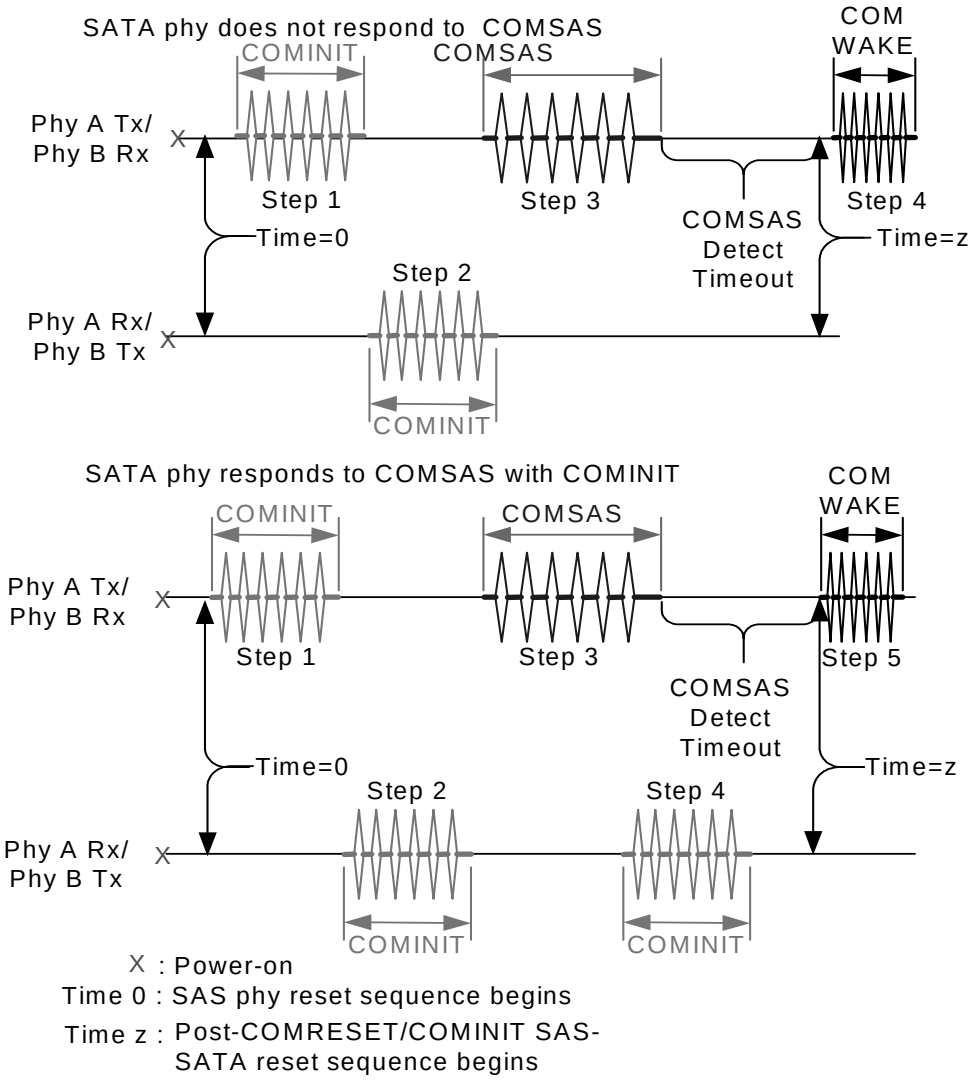


Figure 46 — SAS to SATA OOB sequence

6.6.4 SAS to SAS phy reset sequence

6.6.4.1 SAS OOB sequence

To initiate a SAS OOB sequence a SAS phy shall transmit a COMINIT. If there is no COMINIT or a COMSAS recieved within a hot-plug timeout then the SAS phy shall transmit another COMINIT.

Editor's Note 30: move time z in the SAS to SATA figure above to the start of speed negotiation, not before COMWAKE

On receipt of a COMINIT a SAS phy shall either:

a) transmit a COMINIT, if the receiving SAS phys has not yet transmitted a COMSAS, followed by a COMSAS; or
b) transmit a COMSAS, if the receiving SAS phys has transmitted a COMINIT.

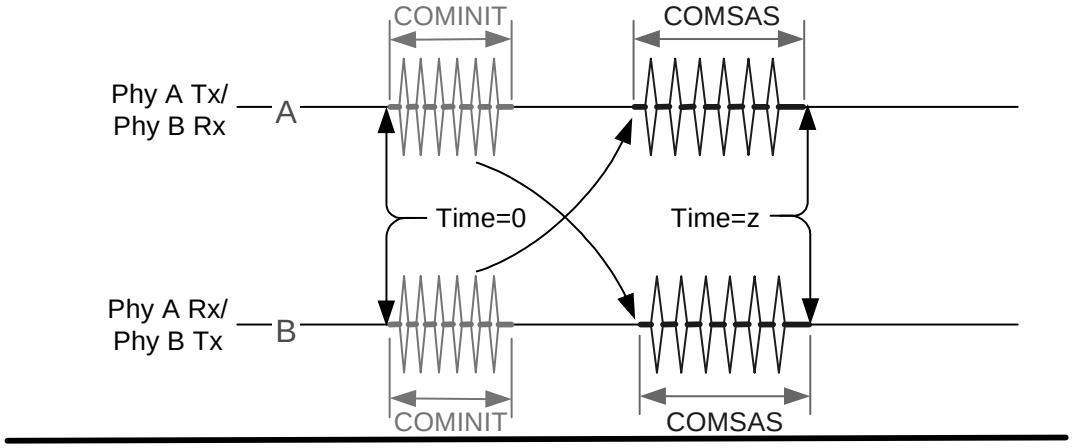
After completing the transmission of a COMSAS and the successful receipt a COMSAS the SAS OOB sequence is complete and the SAS speed negotiation sequence begins.

A SAS device shall distinguish between COMINIT and COMSAS and continue with a SAS speed negotiation sequence after completing the SAS OOB sequence.

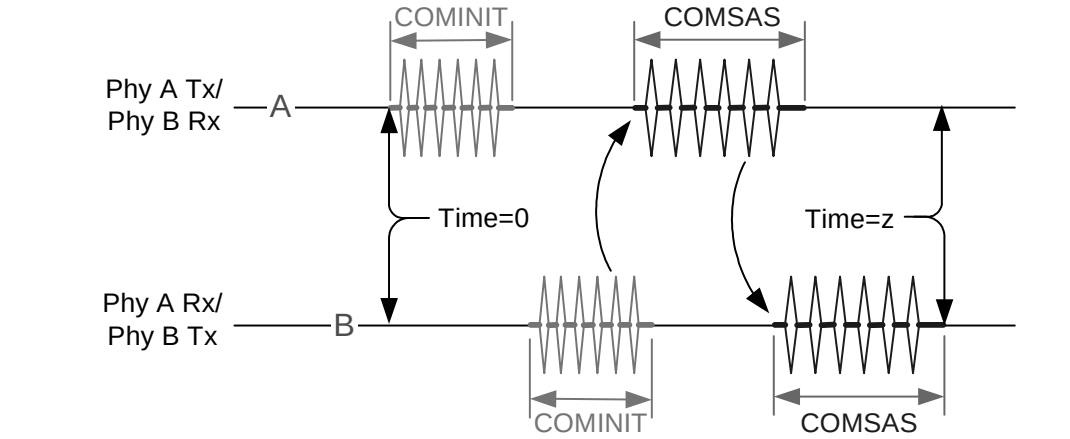
Figure 47 shows several different SAS OOB sequences between a SAS phy A and SAS phy B, with SAS phy

A starting the SAS OOB sequence before, after, or at the same time as SAS phy .

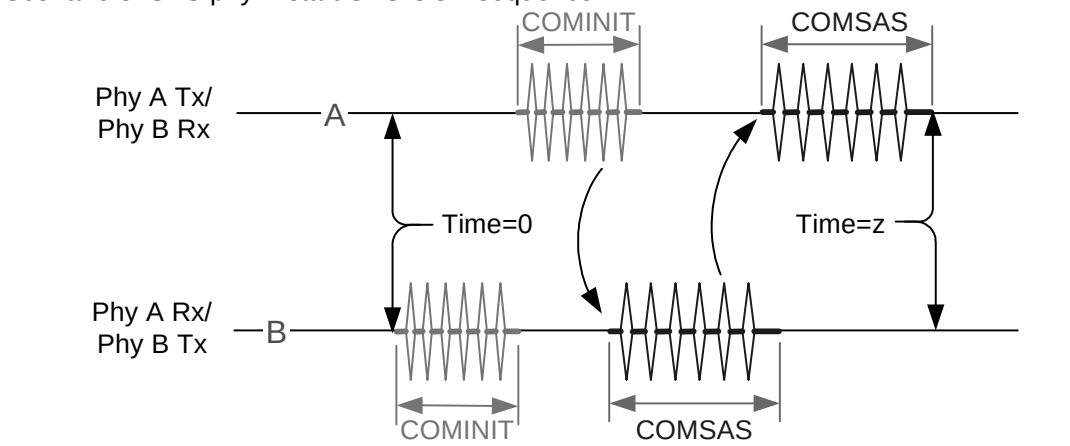
Scenario 1: Both SAS phys start SAS OOB sequence at same time



Scenario 2: SAS phy A starts SAS OOB sequence



Scenario 3: SAS phy B start SAS OOB sequence

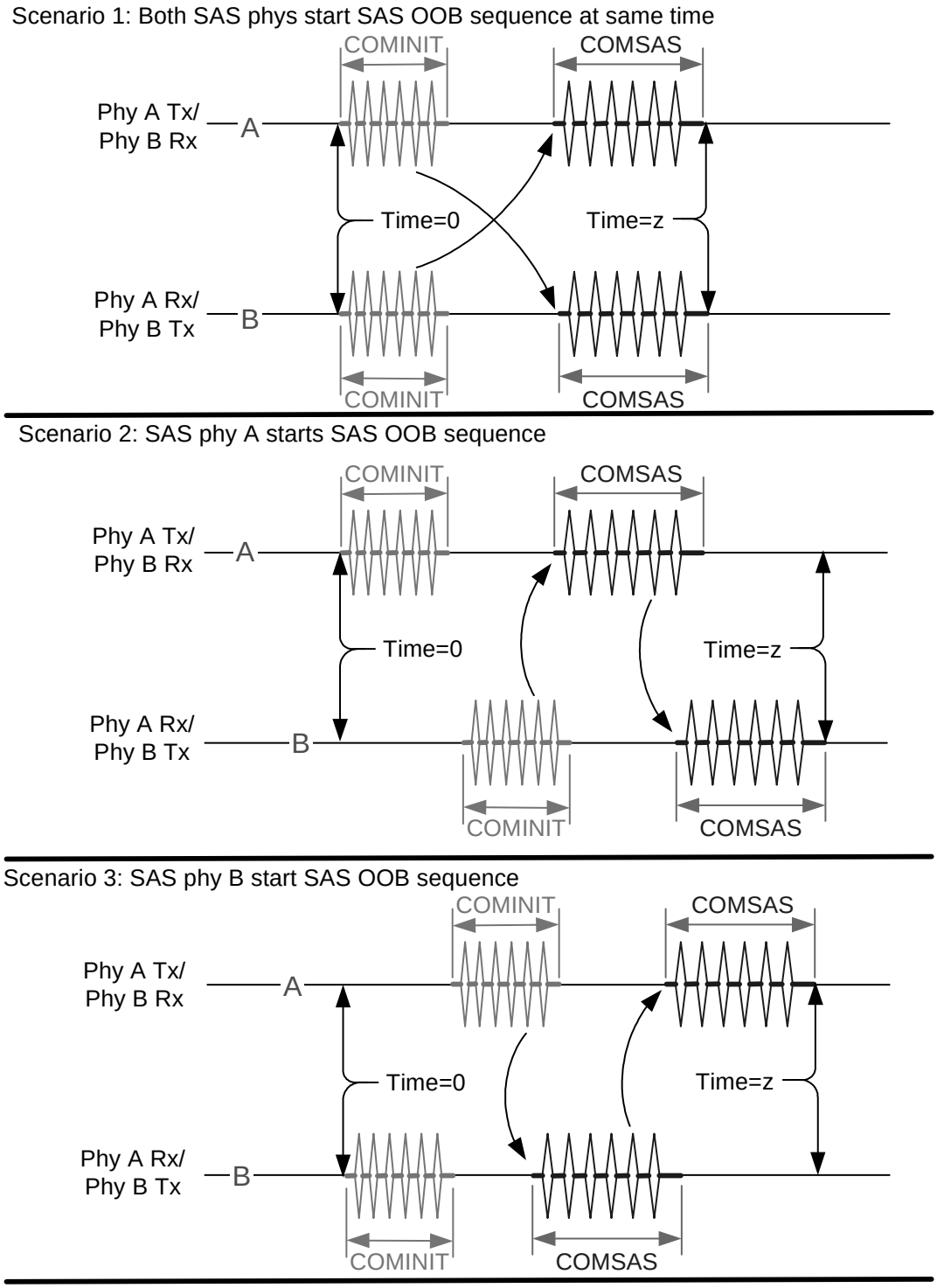


A : SAS A Power-on

B : SAS B Power-on

Time 0: SAS phy reset sequence begins

Time z: SAS speed negotiation sequence begins



A : SAS A Power-on

B : SAS B Power-on

Time 0: SAS phy reset sequence begins

Time z: SAS speed negotiation sequence begins

Figure 47 — SAS to SAS OOB sequence

6.6.4.2 SAS speed negotiation sequence

The SAS speed negotiation sequence is a peer-to-peer negotiation technique that does not assume initiator device and target device roles like the SATA speed negotiation sequence. The sequence consists of a set of speed negotiation windows for each link rate, starting with 1,5 Gbps, then 3,0 Gbps, then the next rate. The length of the speed negotiation sequence is determined by the number of link rates supported by the phys.

Editor's Note 31: change acronyms to 3 letters no T at the end?

Figure 48 defines the speed negotiation window, including:

a) speed negotiation window time (SNWT);
b) rate change delay (RCD);
c) speed negotiation transmit time (SNTT); and
d) speed negotiation lock time (SNLT), a subset of the SNTT used by the receiver.

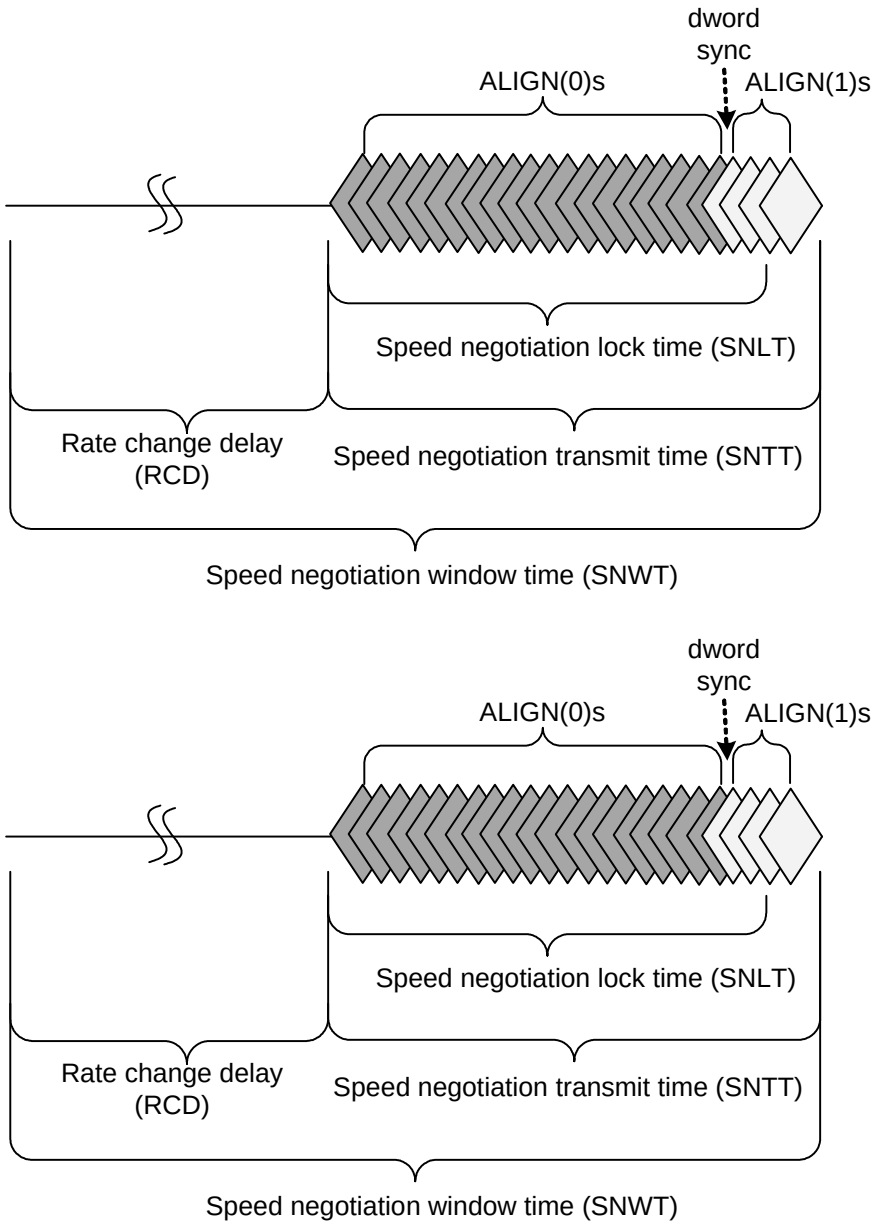


Figure 48 — SAS speed negotiation window

Table 35 defines the timing specifications for the SAS speed negotiation sequence.

Table 38 — SAS speed negotiation sequence timing specifications

Parameter	Minimum	Nominal	Maximum	Comments
Hot-plug timeout	10 ms	100 ms	500 ms	How often a device should retransmit COMINIT to detect if a device has been attached.
Rate change delay (RCD)		750 000 UI(OOB)		The time the transmitter shall transmit idle between rates during speed negotiation. Used by the transmitter and receiver to calculate the width of the speed negotiation window.
Speed negotiation transmit time (SNTT) for transmitter		163 840 UI(OOB)		The time during which ALIGN (0) or ALIGN (1) is transmitted at each rate during the speed negotiation sequence. Derived from: UI(OOB) x 4 096 x 40.
Speed negotiation transmit time (SNTT) for receiver	109,22 µs	109,23 µs	109,24 µs	The time during which ALIGN (0) or ALIGN (1) is received at each rate during negotiation. Derived from: UI(OOB) x 4 096 x 40.
Speed negotiation lock time (SNLT) for transmitter		153 600 UI(OOB)		The maximum time during the speed negotiation window for a transmitter to reply with ALIGN (1). Derived from: UI(OOB) x 3 840 x 40
Speed negotiation lock time (SNLT) for receiver	102,39 µs	102,40 µs	102,41 µs	The maximum time during the speed negotiation window for a receiver to detect ALIGN (0) or ALIGN (1) and reply with ALIGN (1). Derived from: UI(OOB) x 3 840 x 40.
Speed negotiation window time (SNWT)	609,166 µs	609,227µs	609,288 µs	The duration of a speed negotiation window. Derived from: RCD + SNTT.
ALIGN detect timeout			880 µs	The maximum time during SATA speed negotiation that an initiator phy receiver shall allow for an ALIGN to be received after detecting the COMWAKE negation time.

[a] This is not the same as UI(OOB) defined in SATA; SAS has tighter clock tolerance.

The speed negotiation window shall consist of the following transmission sequence for each speed negotiation window:

1) a transmission of idle for an RCD; and
2) if the phy supports the link rate, a transmission of ALIGNs at that link rate for the remainder of entire speed negotiation window. If the phy does not supports the link rate, transmission of idle for the remainder of entire speed negotiation window.

If the receiving phy supports the link rate, it shall attempt to synchronize on an incoming series of dwords at that rate for the SNLT. The received dwords may be ALIGN (0) or ALIGN (1) primitives. If the phy achieves dword synchronization within the SNLT, it shall change from transmitting ALIGN (0) primitives to transmitting ALIGN (1) primitives for the remainder of the SNTT (i.e., the remainder of the SNWT). If the device does not achieve dword synchronization within the SNLT, it shall continue transmitting ALIGNs for the remainder of the SNTT (i.e., the remainder of the SNWT).

At the end of the SNTT, if a phy is both transmitting and receiving ALIGN (1) primitives, it shall consider that link rate valid. The phy shall then proceed to the next speed negotiation window. A phy shall participate in all speed negotiation windows:

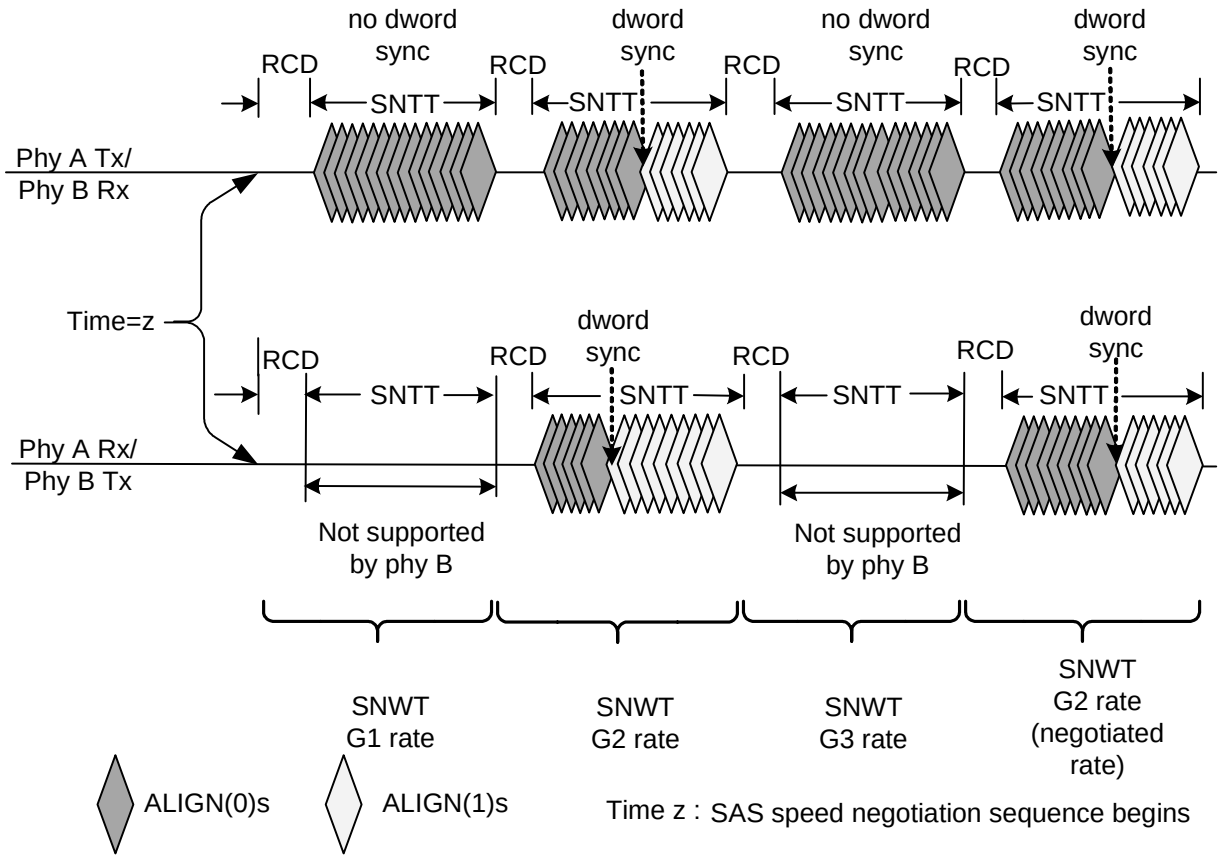
a) up to its highest supported link rate plus one; or
b) until it runs a speed negotiation window that does not detect a valid link rate.

If the phy has detected a valid link rate in the previous speed negotiation window, it shall enter the final speed negotiation window using the highest previously successful link rate.

Figure 49 shows speed negotiation between a SAS phy A that supports G1 thru G3 link rates and a SAS phy B that only supports the G2 link rate. Both phys run the G1 speed negotiation window (supported by phy A but not by phy B, so invalid), the G2 speed negotiation window (valid), and the G3 speed negotiation window

(supported by phy A but not by phy B, so invalid), that phy then selects G2 for the final speed eegotiation

eindow -to establish the negotiated rate.



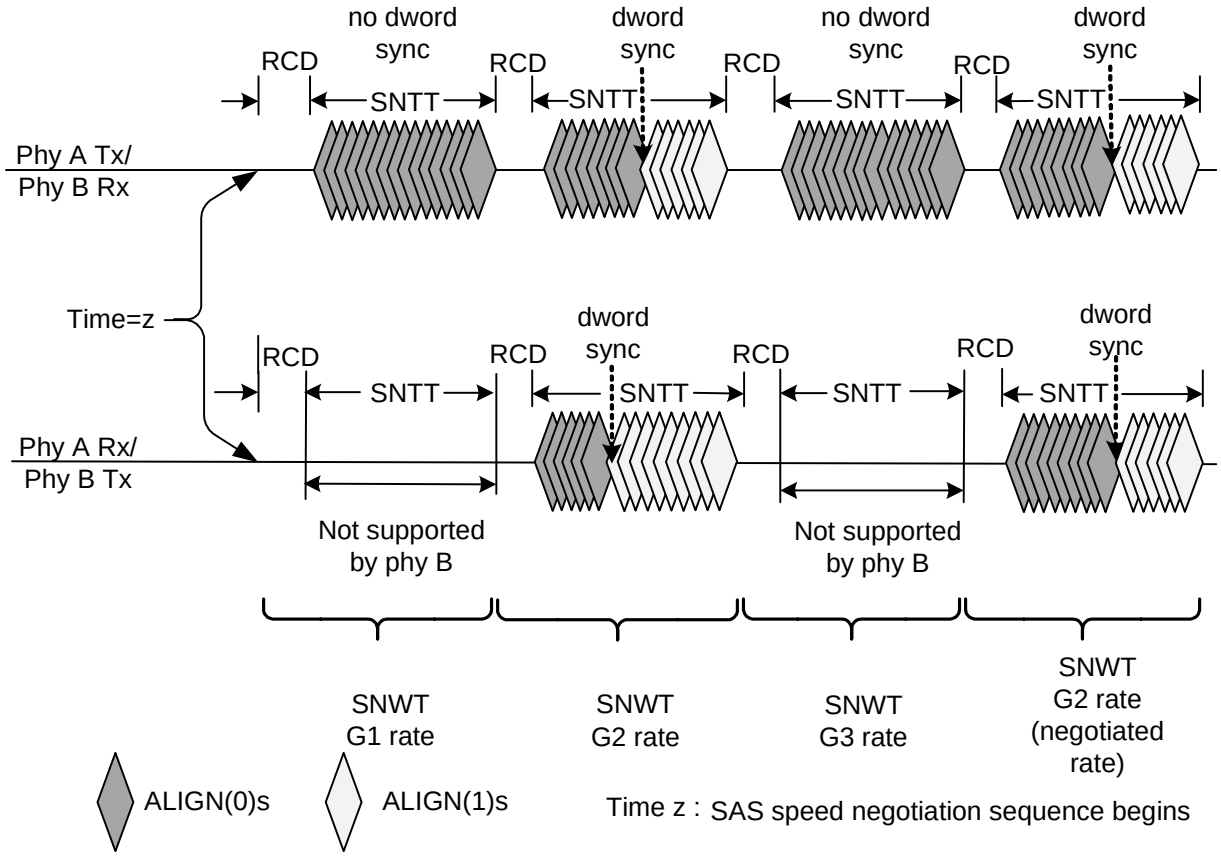


Figure 49 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G2 only)

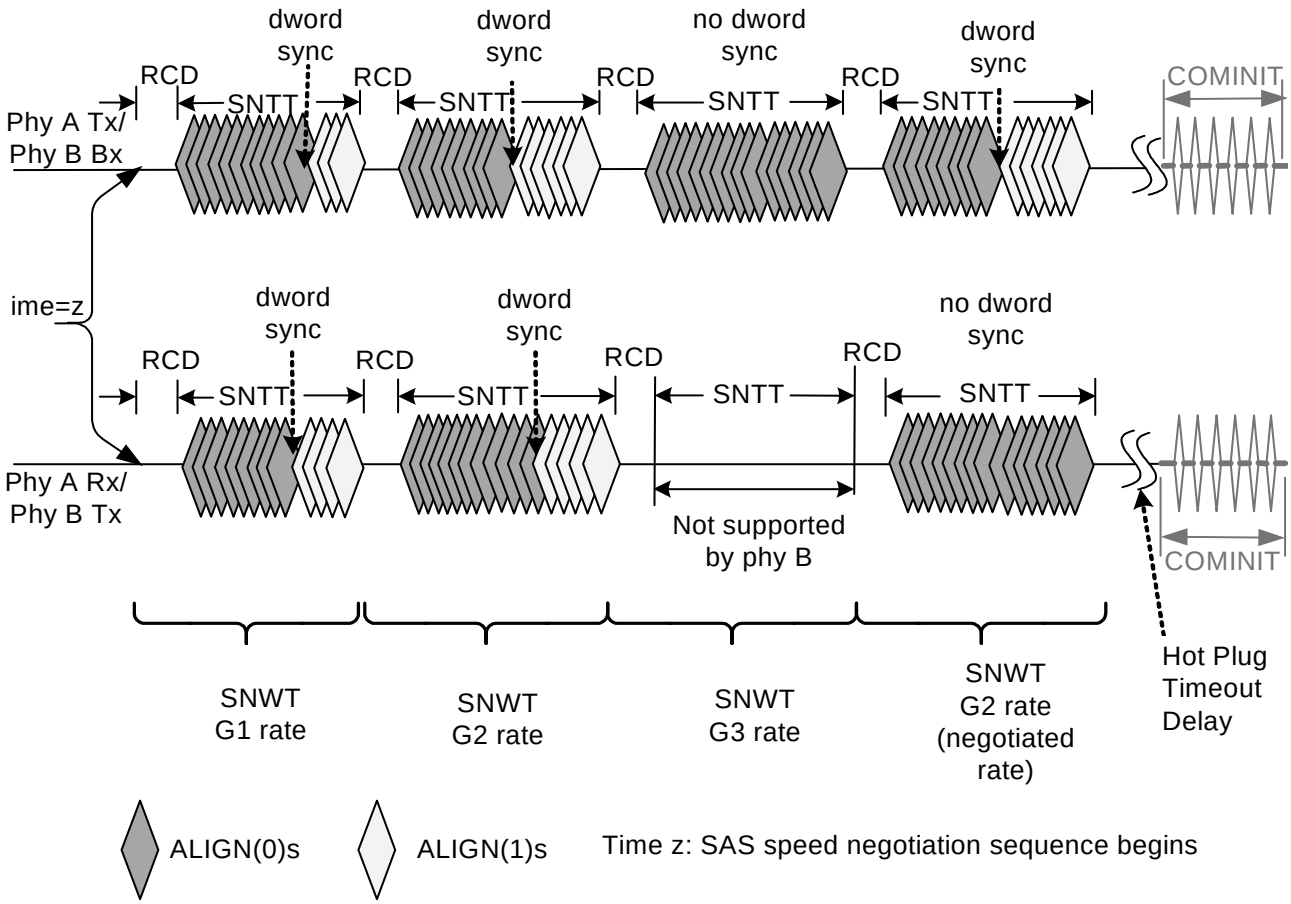
If the SAS phy's PLLs does not obtain dword synchronization during the final speed negotiation window the SAS phy speed negotiation fails and the phy reset sequence shall be retried. This may be counted and reported in the PHY RESET PROBLEM field in the SMP REPORT PHY ERROR LOG page ~~(see 10.3.1.6)~~(see 10.3.1.6) and the REPORT PHY ERROR LOG log page ~~(see 10.1.2.1)~~(see 10.1.3.1).

Any time a SAS phy fails speed negotiation, it shall wait the hot-plug timeout before attempting a retry.

Figure 50 shows the same speed negotiation sequence as in figure 49 when SAS phy B does not obtain dword synchronization during the final speed negotiation window. If this occurs, the handshake is not

complete and the OOB sequence shall be retried starting with COMINIT, forcing the phy to retry the whole

reset sequence.



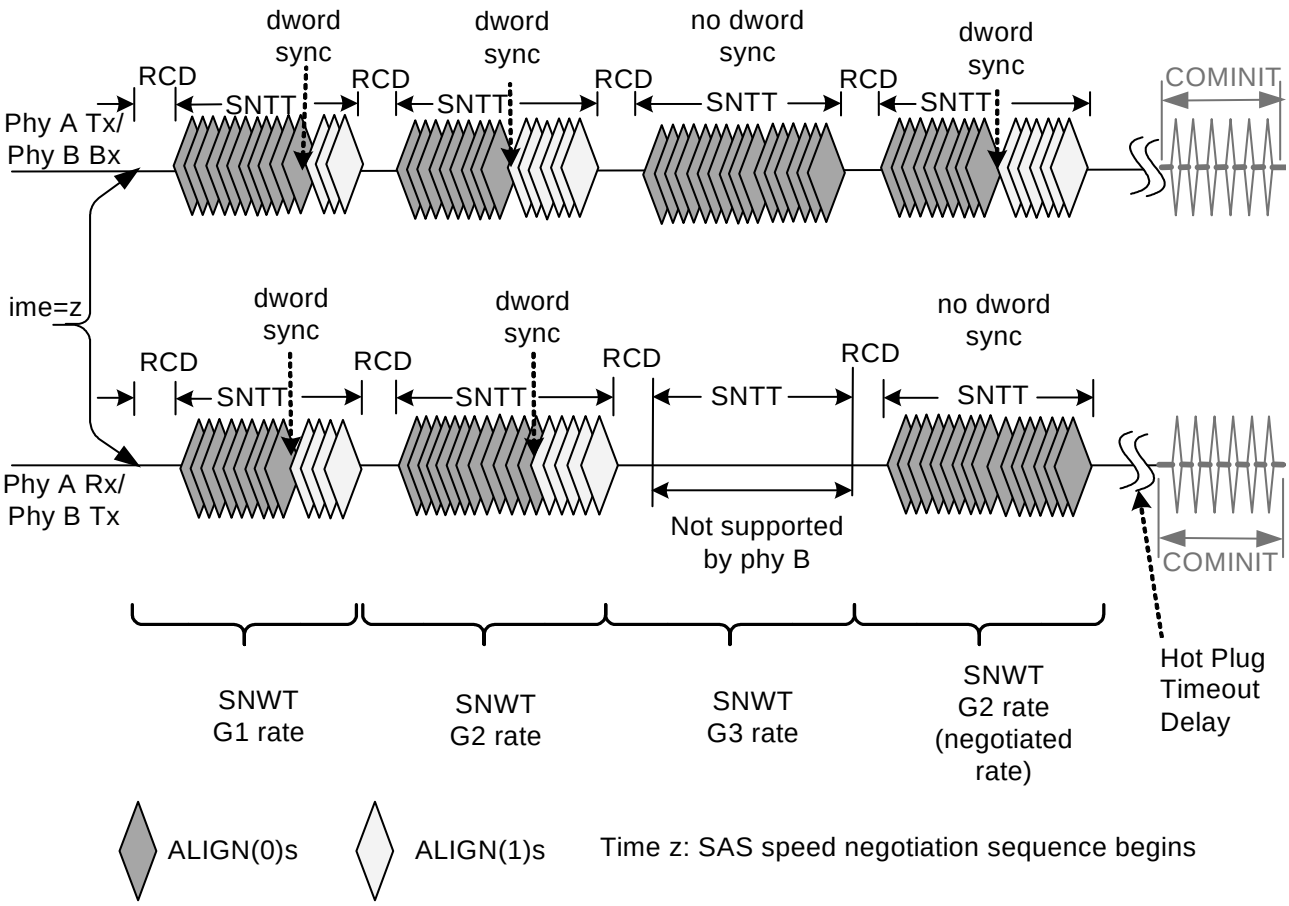


Figure 50 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2) with failure

For more examples of speed negotiaions between devices that support various speeds see ~~Annex A~~Annex A.

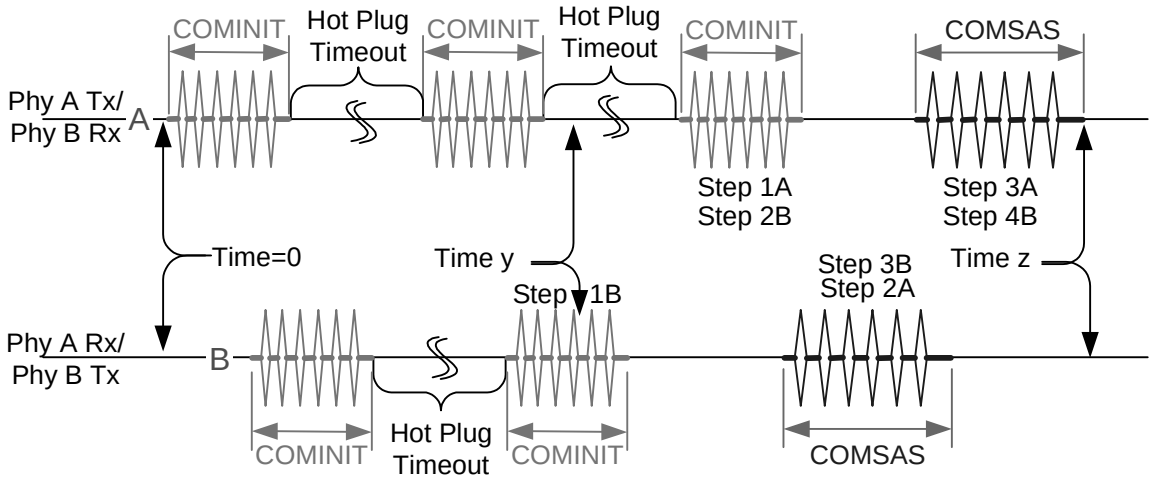
6.6.5 Phy reset sequence after device is attached

Since SATA and SAS signal cable connectors do not include power lines, it is not possible to detect the physical insertion of the signal cable connector into a receptacle. As a result, SAS phys should periodically transmit a COMINIT sequence if they have not detected a COMINIT sequence within a hot-plug timeout.

Editor's Note 32: work on first sentence

Figure 51 shows how two SAS phys complete the phy reset sequence if the phys are not attached at power on. In this example, SAS phy B is attached to SAS phy A some time before SAS phy B's second hot-plug timeout occurs. SAS phy B's OOB detection circuitry detects a COMINIT after the attachment, and therefore SAS phy B transmits the COMSAS sequence, since it has both transmitted and received a COMINIT

sequence. Upon receiving COMSAS, SAS phy A transmits its own COMSAS sequence, bypassing the normal requirement that COMINIT be both transmitted and received. The SAS speed negotiation sequence follows.



A : SAS A Power-on

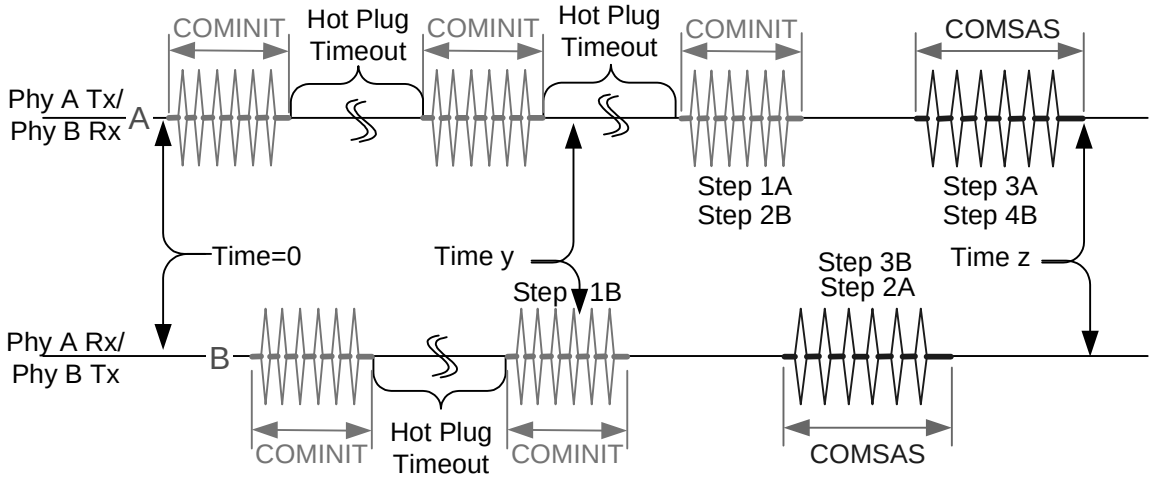
Time y : SAS phy A attached to SAS phy B

Time z : SAS phy A and SAS phy B start the SAS speed negotiation sequence

B : SAS B Power-on

Step xA : The SAS OOB sequence at SAS phy A

Step xB : The SAS OOB sequence at SAS phy B



A : SAS A Power-on

Time y : SAS phy A attached to SAS phy B

Time z : SAS phy A and SAS phy B start the SAS speed negotiation sequence

B : SAS B Power-on

Step xA : The SAS OOB sequence at SAS phy A

Step xB : The SAS OOB sequence at SAS phy B

Figure 51 — Hot-plug and the phy reset sequence

6.7 SAS phy (SP) state machine

6.7.1 Overview

The SAS phy (SP) state machine controls the phy reset sequence. The SP state machine consists of three sets of states:

a) OOB sequence (OOB) states;
b) SAS speed negotiation (SAS) states; and
c) SATA host emulation (SATA) states.

The SP states are:

d) SP1:OOB_COMINIT (see 6.7.2.1);
e) SP2:OOB_AwaitCOMX (see 6.7.2.2);
f) SP3:OOB_AwaitCOMINIT_Sent (see 6.7.2.3);
g) SP4:OOB_COMSAS (see 6.7.2.4);
h) SP5:OOB_AwaitCOMSAS_Sent (see 6.7.2.5);
i) SP6:OOB_AwaitNoCOMSAS (see 6.7.2.6);
j) SP7:OOB_AwaitCOMSAS (see 6.7.2.7);
k) SP8:SAS_Start (see 6.7.3.1);
l) SP9:SAS_RateNotSupported (see 6.7.3.2);
m) SP10:SAS_AwaitALIGN (see 6.7.4.3);
n) SP11:SAS_AwaitALIGN1 (see 6.7.3.4);
o) SP12:SAS_AwaitSNW (see 6.7.3.5);
p) SP13:SAS_Pass (see 6.7.3.6);
q) SP14 SAS_Fail (see 6.7.3.7);
r) ~~SP18~~SP15:SAS_PHY_Ready (see 6.7.3.8);
s) ~~SP19~~SP16:SATA_COMWAKE (see 6.7.4.1);
t) ~~SP20~~SP17:SATA_AwaitCOMWAKE (see 6.7.4.2);
u) ~~SP21~~SP18:SATA_AwaitNoCOMWAKE (see 6.7.4.3);
v) ~~SP22~~SP19:SATA_AwaitALIGN (see 6.7.4.4);
w) ~~SP23~~SP20:SATA_AdjustSpeed (see 6.7.4.5);
x) ~~SP24~~SP21:SATA_Transmit_ALIGN (see 6.7.4.6);
y) ~~SP26~~SP22:SATA_PHY_Ready (see 6.7.4.7);
z) ~~SP27~~SP23:SATA_PM_Partial (see 6.7.4.8); and
aa) ~~SP28~~SP24:SATA_PM_Slumber (see 6.7.4.9).
ab) SP25:SATA_DelaySpin(see xxx).

The SP state machine shall start in the SP0:SAS_PowerOn state after:

a) a power on event; or
b) receiving a PhyReset request from the management layer (e.g., from the SMP PHY CONTROL function).

~~Editor's Note 33: renumber states now that several have been deleted~~

The SP state machine sends the following parameters to the DWS state machines:

a) PhyNotReady;
b) PhyReady (SAS); and
c) PhyReady (SATA).

The SP state machine interacts with separate SP transmitter logic which transmits OOB signals.

Parameters sent to the SP transmitter include:

a) Transmit COMINIT;
b) Transmit COMSAS;
c) Transmit COMWAKE;

d) Transmit D10.2s;
e) Set Rate (Physical Link Rate);
f) Transmit ALIGN0s; and
g) Transmit ALIGN1s.

Parameters received from the SP transmitter include:

h) COMINIT Transmitted;
i) COMSAS Transmitted; and
j) COMWAKE Transmitted.

The SP state machine interacts with separate SP receiver logic which receives OOB signals.

Parameters received from the SP receiver include:

k) COMINIT Detected;
l) COMSAS Detected;
m) COMWAKE Detected;
n) COMINIT Completed;
o) COMSAS Completed;
p) COMWAKE Completed;
q) ALIGN0 Detected; and
r) ALIGN1 Detected.

The SP State Machine interacts with the upper level SMP protocol layer to control Delayed Spin for SATA devices.

The parameters sent to the upper level SMP protocol are:

s) SpinupHold
t) Broadcast Event Notify

The parameter received from the upper level SMP protocol layer is:

u) SMP Reset

The SP State Machine interacts with the upper level SMP protocol to control Delayed Spin for SATA devices.

The parameters sent to the upper SMP layer is

Spin Pause

The SAS phy layer shall maintain the following timers:

a) OOB timeout timer;
b) hot-plug timeout timer;
c) RCD timer;
d) SNLT timer;
e) SNWT timer; and
f) COMSAS detect timeout timer.

Editor's Note 34: use only acronyms

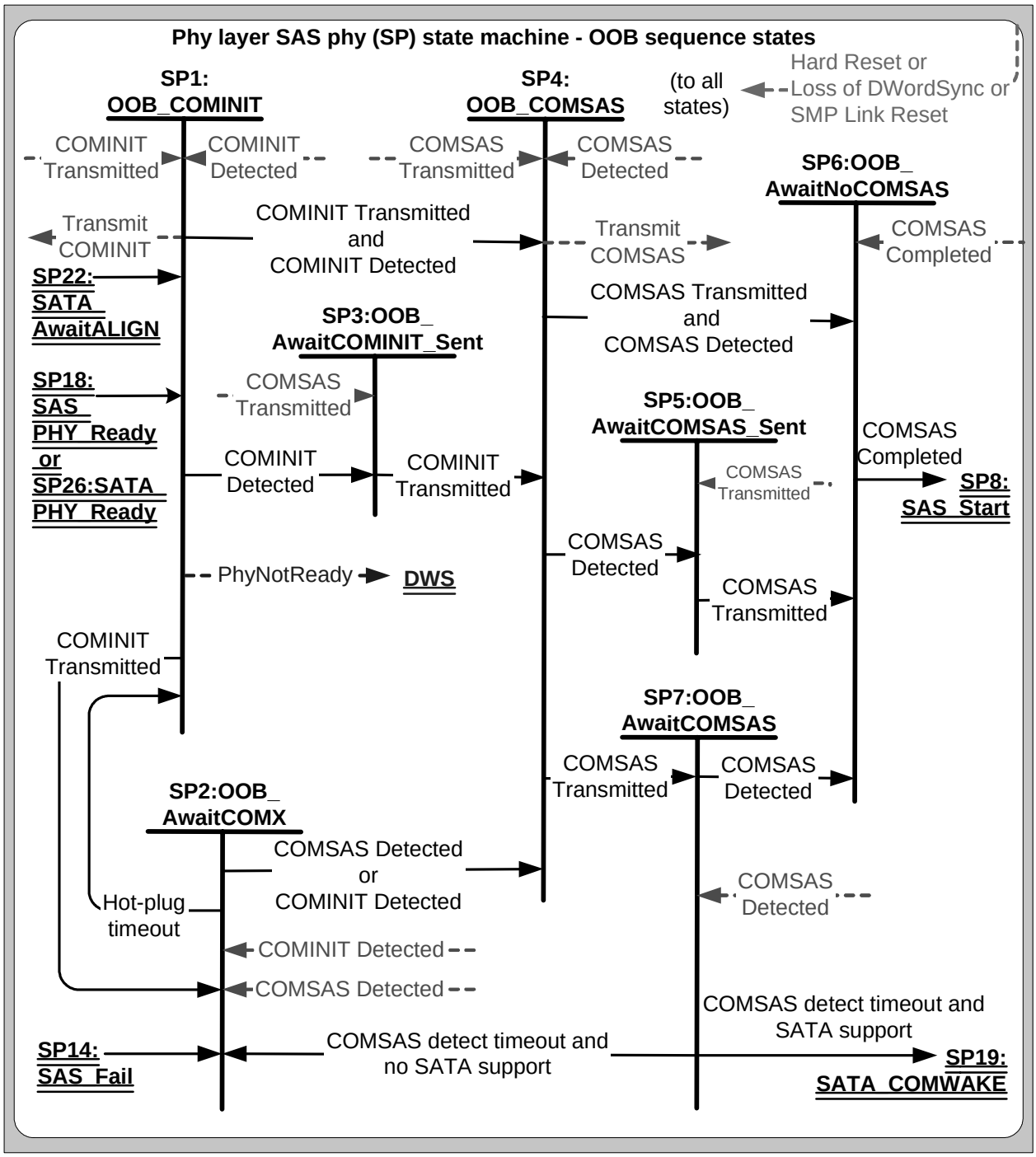
Editor's Note 35: fix await vs wait in text

~~Editor's Note 36: OOB states have green to/from transmitter/receiver. Since transition conditions are labeled, could drop them and leave them implied. Could drop the transition condition labels,~~

~~but that makes the picture less useful.~~

6.7.2 OOB sequence states

Figure 52 shows the OOB sequence states.



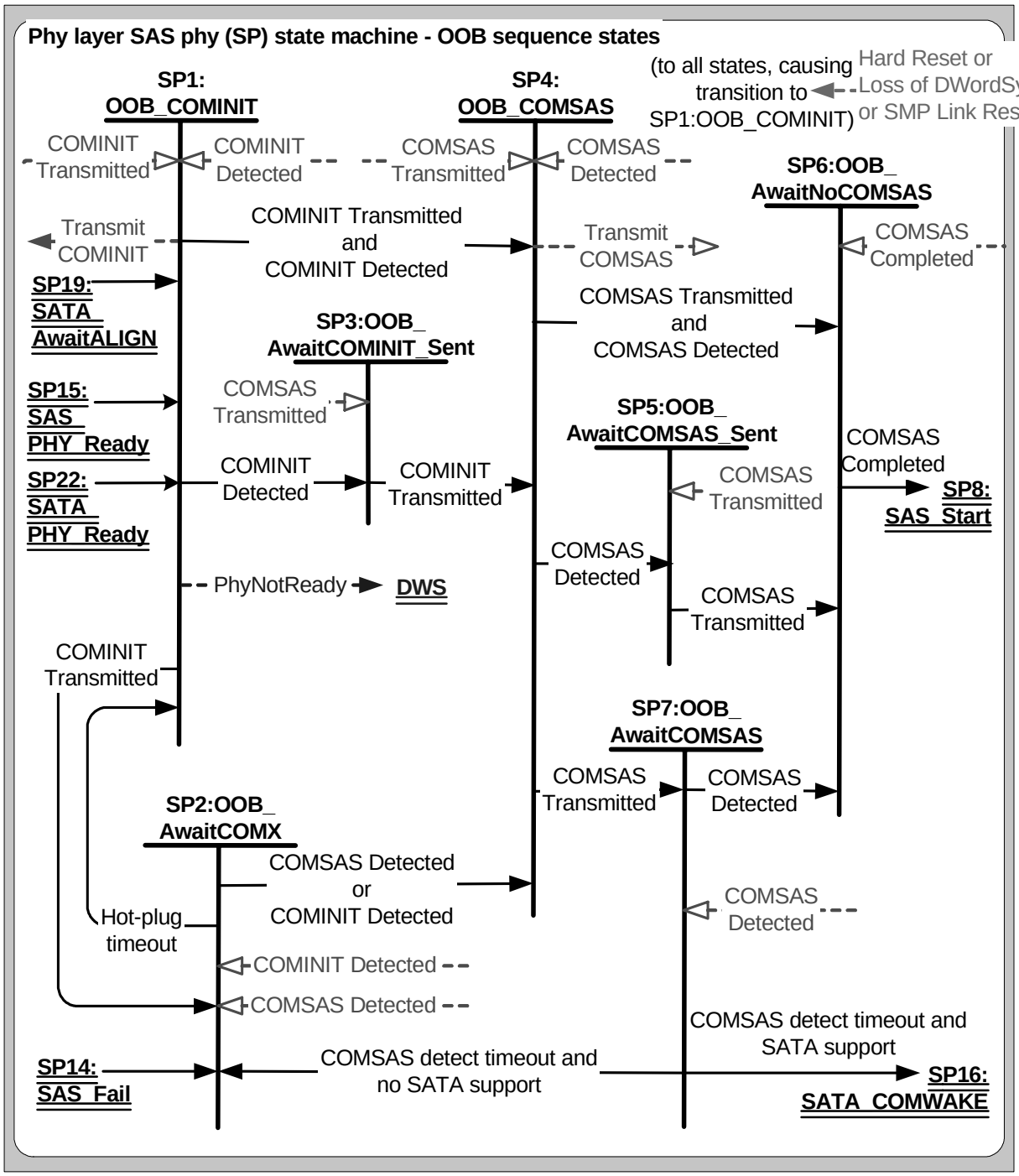


Figure 52 — SAS phy (SP) state machine - OOB sequence states

6.7.2.1 SP1:OOB_COMINIT state

6.7.2.1.1 State description

This state is the initial state for the state machine.

This state shall send a Transmit COMINIT parameter to the SP transmitter and wait for COMINIT to be transmitted and/or received.

This state shall send a PhyNotReady parameter to the DWS state machine.

~~The Fallback and Success flags shall be set to zero upon entering this state.~~

~~Editor's Note 37: work on these flags - without Fallback and IncSpeed states, more need for them~~

6.7.2.1.2 Transition SP1:OOB_COMINIT to SP2:OOB_AwaitCOMX

This transition shall occur if this state receives a COMINIT Transmitted parameter from the SP transmitter.

6.7.2.1.3 Transition SP1:OOB_COMINIT to SP3:OOB_AwaitCOMINIT_Sent

This transition shall occur if this state receives a COMINIT Detected parameter from the SP receiver.

6.7.2.1.4 Transition SP1:OOB_COMINIT to SP4:OOB_COMSAS

This transition shall occur if this state receives both a COMINIT Transmitted parameter from the SP transmitter and a COMINIT Detected parameter from the SP receiver.

6.7.2.2 SP2:OOB_AwaitCOMX state

6.7.2.2.1 State description

Upon entering this state, a hot-plug timeout timer shall be initialized and enabled. The state machine waits for COMINIT, COMSAS, or a hot-plug timeout.

6.7.2.2.2 Transition SP2:OOB_AwaitCOMX to SP1:OOB_COMINIT

This transition shall occur when a hot-plug timeout occurs.

6.7.2.2.3 Transition SP2:OOB_AwaitCOMX to SP4:OOB_COMSAS

This transition shall occur if this state receives either a COMINIT Detected parameter or a COMSAS Detected parameter from the SP receiver.

6.7.2.3 SP3:OOB_AwaitCOMINIT_Sent state

6.7.2.3.1 State description

This state is entered when a COMINIT sequence has been detected but the COMINIT initiated in SP1:OOB_COMINIT has not been completely transmitted.

This state waits for COMINIT to be transmitted.

6.7.2.3.2 Transition SP3:OOB_AwaitCOMINIT_Sent to SP4:COMSAS

This transition shall occur if this state receives a COMINIT Transmitted parameter from the SP transmitter.

6.7.2.4 SP4:OOB_COMSAS state

6.7.2.4.1 State description

This state is reached when a COMINIT has been transmitted and detected.

This state shall send a Transmit COMSAS parameter to the SP transmitter and wait for COMSAS to be transmitted and/or detected.

6.7.2.4.2 Transition SP4:OOB_COMSAS to SP5:OOB_AwaitCOMSAS_Sent

This transition shall occur if this state receives a COMSAS Detected parameter from the SP receiver.

6.7.2.4.3 Transition SP4:OOB_COMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur if this state receives both a COMSAS Transmitted parameter from the SP transmitter and a COMSAS Detected parameter from the SP receiver.

6.7.2.4.4 Transition SP4:OOB_COMSAS to SP7:OOB_AwaitCOMSAS

This transition shall occur if this state recieves a COMSAS Transmitted parameter from the SP transmitter.

6.7.2.5 SP5:OOB_AwaitCOMSAS_Sent state

6.7.2.5.1 State description

This state waits for COMSAS to be transmitted.

6.7.2.5.2 Transition SP5:OOB_AwaitCOMSAS_Sent to SP6:OOB_AwaitNoCOMSAS

This transition shall occur if this state receives a COMSAS Transmitted parameter from the SP transmitter.

6.7.2.6 SP6:OOB_AwaitNoCOMSAS state

6.7.2.6.1 State description

This state is entered when a COMSAS sequence has been both transmitted and detected.

The state machine waits for COMSAS to be completely received.

6.7.2.6.2 Transition SP6:OOB_AwaitNoCOMSAS to SP8:SAS_Start

This transition shall occur if this state receives a COMSAS Completed parameter from the SP receiver. The COMSAS Completed parameter may be received before the SP6 state is entered.

6.7.2.7 SP7:OOB_AwaitCOMSAS state

6.7.2.7.1 State description

Upon entering this state the COMSAS detect timeout timer shall be initialized and enabled.

This state waits for COMSAS to be received or for a COMSAS detect timeout.

6.7.2.7.2 Transition SP7:OOB_AwaitCOMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur if this state receives a COMSAS Detected parameter from the SP receiver.

6.7.2.7.3 Transition SP7:OOB_AwaitCOMSAS to ~~SP19~~SP16:SATA_COMWAKE

This transition shall occur if the device supports SATA and the COMSAS detect timeout timer expires.

6.7.2.7.4 Transition SP7:OOB_AwaitCOMSAS to SAS_AwaitNoCOMX

This transition shall occur if the device does not support SATA and the COMSAS detect timeout timer expires.

6.7.3 SAS speed negotiation states

Figure 53 shows the SAS speed negotiation states, in which the SAS phy has detected that it is attached to a SAS phy, and performs the SAS speed negotiation sequence. These states are indicated by state names with a prefix of SAS.

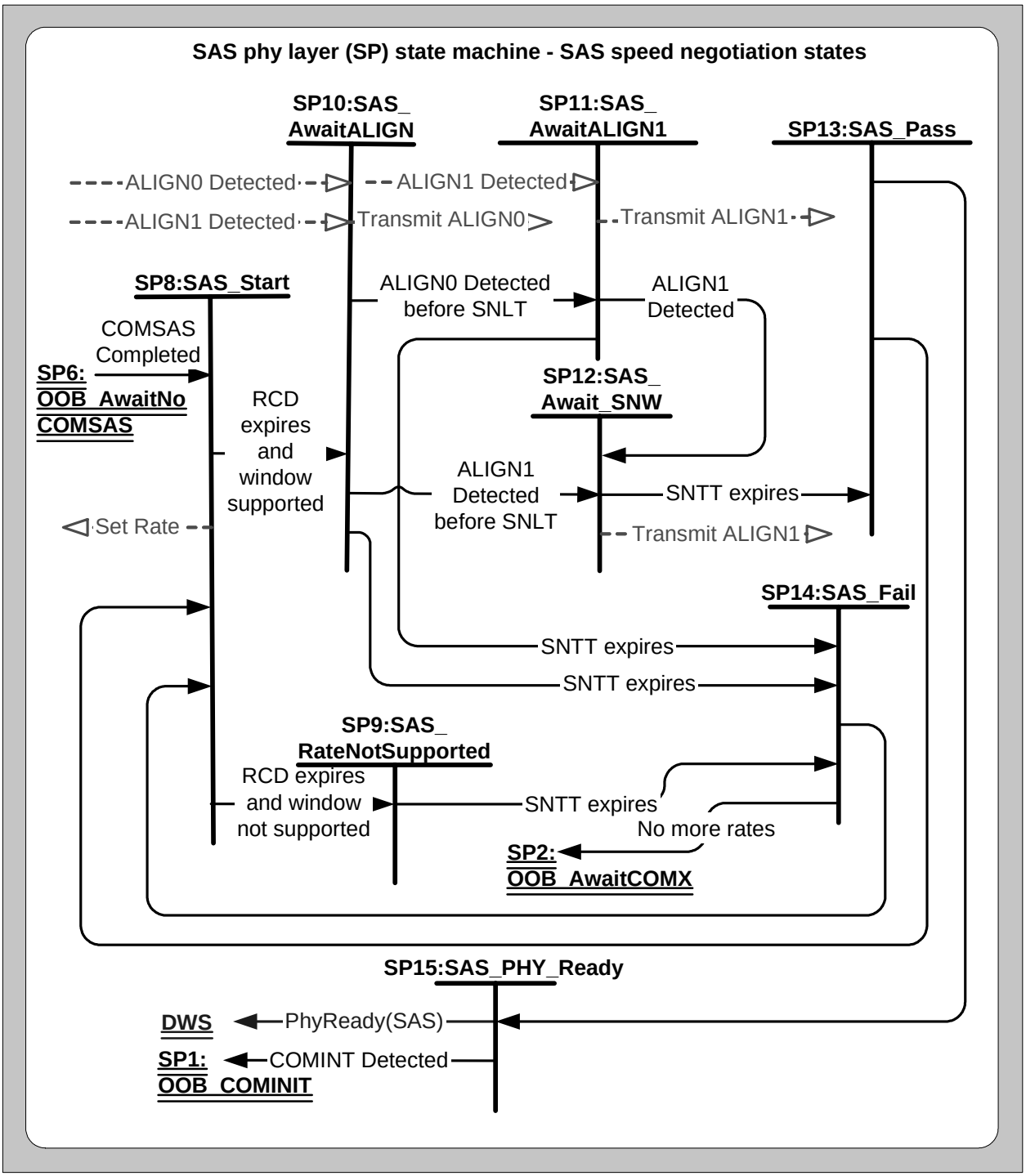


Figure 53 — SAS phy (SP) state machine - SAS speed negotiation states

6.7.3.1 SP8:SAS_Start state

6.7.3.1.1 State description

This state marks the beginning of the SAS speed negotiation process. It is used to transmit idle ~~state~~ in between SAS speed negotiation windows.

Upon entering this state, the RCD timer shall be initialized and started. This allows time required for a transmitter to switch to either the next higher or next lower supported speed. The state machine remains in this state until a RCD has elapsed.

This state shall send the Set Rate parameter to the SP transmitter selecting the next rate for attempted negotiation.

~~Editor's Note 38: more words here on what rate to select since the IncSpeed/Fallback states were removed.~~

This state sets the SAS speed negotiation rate to:

a) the lowest supported speed negotiation window if the transition into this state is from the SP6:OOB_AwaitNoCOMSAS state; or
b) to the value of the speed negotiation window received as an argument.

During this state idle shall be transmitted.

6.7.3.1.2 Transition SP8:SAS_Start to SP10:SAS_AwaitALIGN

This transition shall occur when the RCD timer expires and the rate of the current ~~rate~~ speed negotiation window is supported.

6.7.3.1.3 Transition SP8:SAS_Start to SP9:SAS_RateNotSupported

This transition shall occur when the RCD timer expires and the rate of the current ~~rate~~ speed negotiation is not supported.

6.7.3.2 SP9:SAS_RateNotSupported state

6.7.3.2.1 State description

Upon entering this state the ~~SNWT~~ SNTT timer shall be initialized and enabled. The state machine exits from this state after the ~~SNWT~~ SNTT expires.

During this state idle shall be transmitted.

6.7.3.2.2 Transition SP9:SAS_RateNotSupported to SP14:SAS_Fail

This transition shall occur after the ~~SNWT~~ SNTT expires.

6.7.3.3 SP10:SAS_AwaitALIGN state

6.7.3.3.1 State description

The state machine shall start transmitting ALIGN (0) primitives at the current rate (G1, G2, G3…).

Upon entering this state, the ~~SNWT~~ SNTT timer and SNLT timer shall be initialized and enabled.

The state machine exits this state when ~~SNWT~~ SNTT expires without the pattern synchronization, or if synchronization occurs before the SNLT expires.

6.7.3.3.2 Transition SP10:SAS_AwaitALIGN to SP11:SAS_AwaitALIGN1

This transition shall occur if this state receives an ALIGN0 Detected parameter from the SP receiver before the SNLT timer expires.

6.7.3.3.3 Transition SP10:SAS_AwaitALIGN to SP12:SAS_AwaitSNW

This transition shall occur when this state receives an ALIGN1 Detected parameter from the SP ~~receiver~~receiver before the SNLT timer expires.

6.7.3.3.4 Transition SP10:SAS_AwaitALIGN to SP14:SAS_Fail

This transition shall occur if the ~~SNWT~~ SNTT timer expires.

6.7.3.4 SP11:SAS_AwaitALIGN1 state

6.7.3.4.1 State description

This state is reached after ALIGN (0) has been both transmitted and received.

During this state ALIGN (1) shall be transmitted.

This state is exited when the ~~SNWT~~ SNTT expires or when ALIGN (1) primitives are received before the ~~SNWT~~SNLT expires.

~~**6.7.3.4.2 Transition SP11:SAS_AwaitALIGN1 to SP14:SAS_Fail**~~

~~This transition shall occur if the SNWT timer expires. This indicates that the other phy has not been able to lock at the current rate.~~

6.7.3.4.3 Transition SP11:SAS_AwaitALIGN1 to ~~SP13~~SP14:~~SAS_Pass~~SAS_Fail

This transition shall occur if ~~this state receives an ALIGN1 Detected parameter from~~ the ~~SP receiver and the SNWT~~ SNTT timer expires. This indicates that the other phy has not been able to lock at the current rate.

6.7.3.4.4 Transition SP11:SAS_AwaitALIGN1 to SP14:SAS_ AwaitSNW

This transition shall occur if this state receives an ALIGN1 Detected parameter from the SP ~~receiver~~receiver before the SNTT timer expires. This indicates that the other phy has been able to lock at the current rate.

6.7.3.5 SP12:SAS_AwaitSNW state

6.7.3.5.1 State description

This state is reached after ALIGN (1) has been both transmitted and received.

During this state ALIGN (1) shall be transmitted.

This state waits for the ~~SNWT~~ SNTT timer to expire.

6.7.3.5.2 Transition SP12:SAS_AwaitALIGN1 to SP13:SAS_Pass

This transition shall occur after when the ~~SNWT~~ SNTT timer expires.

6.7.3.6 SP13:SAS_Pass state

6.7.3.6.1 State description

~~This state is entered from state SP11:SAS_AwaitALIGN1 or SP12:SAS_AwaitSNW after an SNWT occurs. This is a single clock state that set the Success flag, indicating that there was at least one successful speed negotiation.~~

This state determines if:

a) another SAS speed negotiation window is required; and
b) the SAS speed negotiation is complete.

6.7.3.6.2 Transition SP13:SAS_Pass to SP8:SAS_Start

This transition shall occur:

a) after setting the SAS speed negotiation window rate to one greater than the current SAS speed negotiation window rate which is sent as an argument to the SN_start state; and
b) ~~This transition shall occur~~ if the state machine has not fallen ~~back, which means that it has not progressed past the breaking point so the~~ back during this current SAS speed ~~is incremented for the next attempt~~negotiation.

6.7.3.6.3 Transition SP13:SAS_Pass to ~~SP18~~SP15:SAS_PHY_Ready

This transition shall occur if ~~the~~ speed ~~negotiation has completed. This is indicated by the Fallback flag. If this flag is set it indicates that the speed~~ negotiation has progressed to where it failed and then had fallen back to the last negotiated speed and then subsequently passed.

6.7.3.7 SP14:SAS_Fail state

6.7.3.7.1 State description

~~This state is entered if ALIGN (1) primitives are not received, indicating that the other phy does not support the current rate.~~

This state determines if the SAS speed negotiation window failure occurred because:

a) the maximum SAS speed negotiation window has been attempted and there haven't been any successful negotiated rates;
b) the SAS speed negotiation failed after dropping back to the last successful SAS speed negotiation window;
c) the SAS speed negotiation has failed and there was a previous successful SAS speed negotiation; or
d) no SAS speed negotiation has previously passed and the maximum SAS speed negotiation window has not yet been attempted.

6.7.3.7.2 Transition SP14:SAS_Fail to SP2:OOB_AwaitCOMX

~~This transition shall occur when the state machine has not been successful, meaning that the max rate has been attempted and there haven't been any successful rate negotiated or when the speed negotiation fails after dropping back to the last working frequency.~~

This transition shall occur if:

a) the maximum SAS speed negotiation window has been attempted and there haven't been any successful negotiated rates; or
b) the SAS speed negotiation failed after dropping back to the last successful SAS speed negotiation window.

6.7.3.7.3 Transition SP14:SAS_Fail to SP8:SAS_Start

This transition shall occur:

a) after setting the SAS speed negotiation window to one less the current SAS speed negotiation window; and
b) if the SAS speed negotiation has failed and there was a previous successful SAS speed negotiation;

or:

a) after setting the SAS speed negotiation window to one greater than the current SAS speed negotiation window; and
b) ~~This transition shall occur if the~~ no SAS speed negotiation has ~~failed and there was a previous success or if speed negotiation has not~~ previously passed and the maximum supported ~~rate~~ SAS speed negotiation window has not yet been attempted.

Which speed negotiation window to use is sent as an argument with this transition.

6.7.3.8 ~~SP18~~SP15:SAS_PHY_Ready state

6.7.3.8.1 State description

~~This state is reached when SAS speed negotiation procedure has completed successfully.~~

This state enables the SAS phy dword synchronization state machine (DWS) to provide rule checking for dword synchronization and determination of link failure. ~~The input from DWS to indicate that Dword Synchronization is Acquired (DWordSync) and is used in the SP:18:Phy Ready and the SP26:SATA_Phy Ready states to determine if it is necessary to re-run OOB. Until this time, the DWS input is ignored.~~

~~The state machine remains in this state until a COMINIT sequence is detected or loss of DWordSync.~~

This state shall not be exited until a Dword Synchronization (Acquired) parameter is received.

After a Dword Synchronization (Acquired) parameter is received this state machine monitors for:

a) the receipt of a COMINIT; and
b) the Dword Synchronization (Loss) parameter.

While in this state dwords from the link layer are transmitted at the negotiated rate.

This state shall send a PhyReady (SAS) parameter to the DWS state machine to indicate that the link has been brought up successfully in SAS mode. While in this state, dwords from the link layer are transmitted at the negotiated rate

6.7.3.8.2 Transition ~~SP18~~SP15:SAS_PHY_Ready to SP1:OOB_COMINIT

This transition shall occur if:

a) ~~This transition shall occur if~~ a COMINIT Detected parameter is received from the SP receiver~~.~~; or

~~Editor's Note 39: or dword sync is lost...~~

b) a Dword Synchronization (Loss) parameter is received.

6.7.4 SATA host emulation states

Figure 54 shows the SATA Host emulation states, in which the SAS device has detected that it is attached to a SATA target device and behaves as if it were a SATA host, initiating the SATA speed negotiation sequence. These states are indicated by state names with a prefix of SATA_.

During SATA Host emulation, the SAS device transmits a COMWAKE sequence and then waits to receive a COMWAKE. Once the COMWAKE sequence is detected, the SAS device follows the speed negotiation sequence defined in the SATA.

The power management states defined in this specification are for SAS initiator devices that support being attached to SATA targets; SAS expander devices attached to SATA target devices do not support power management in this standard.

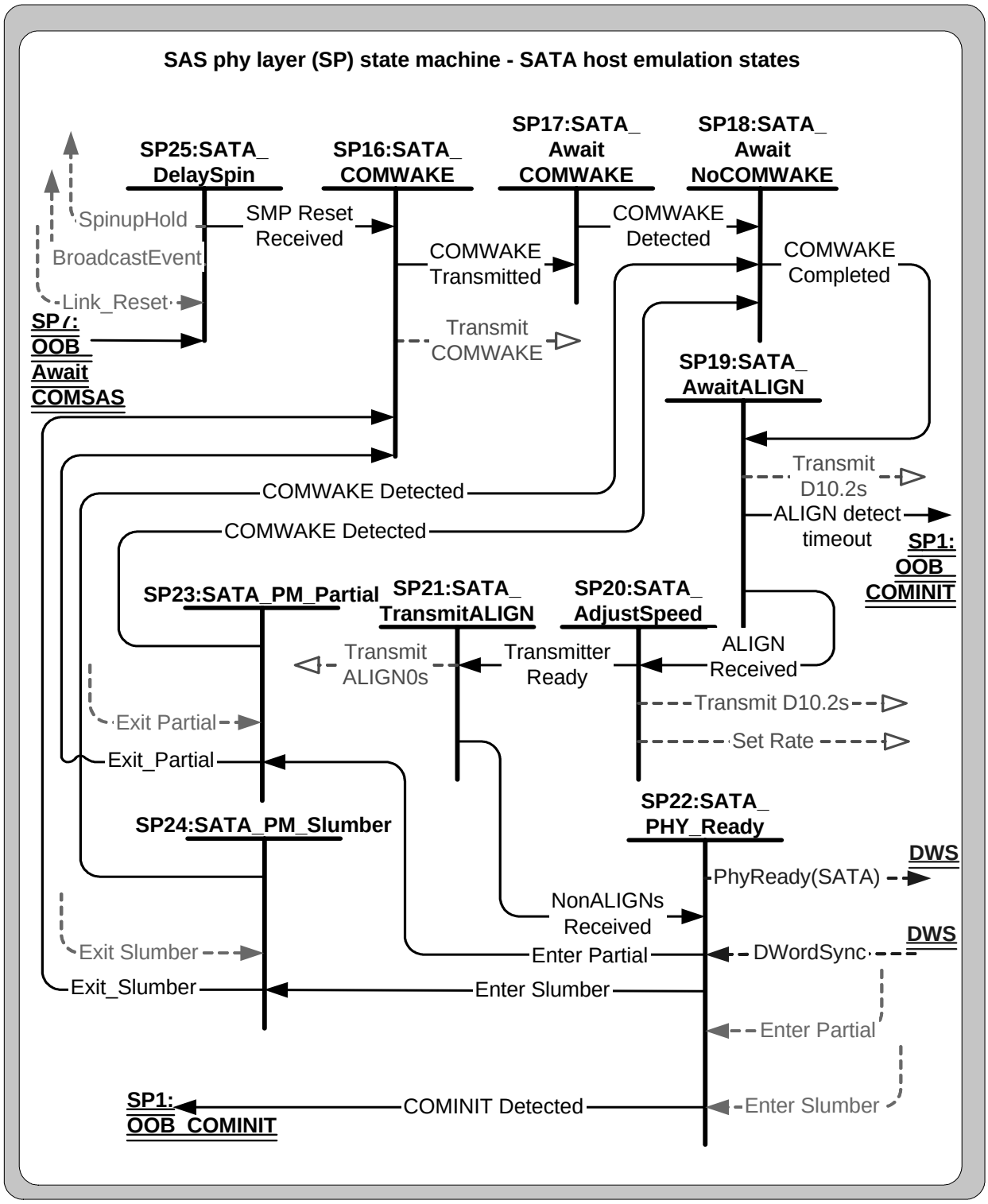


Figure 54 — SAS phy (SP) state machine - SATA host emulation states

6.7.4.1 ~~SP19~~SP16:SATA_COMWAKE state

6.7.4.1.1 State description

This state shall send a Transmit COMWAKE parameter to the SP transmitter and wait for COMWAKE to be transmitted.

6.7.4.1.2 Transition ~~SP19~~SP16:SATA_COMWAKE to ~~SP20~~SP17:SATA_AwaitCOMWAKE

This transition shall occur when this state receives a COMWAKE Transmitted parameter from the SP transmitter.

6.7.4.2 ~~SP20~~SP17:SATA_AwaitCOMWAKE state

6.7.4.2.1 State description

This state waits for COMWAKE to be received.

6.7.4.2.2 Transition ~~SP20~~SP17:SATA_AwaitCOMWAKE to ~~SP21~~SP18:SATA_AawitNoCOMWAKE

This transition shall occur when this state receives a COMWAKE Received parameter from the SP receiver.

6.7.4.3 ~~SP21~~SP18:SATA_AwaitNoCOMWAKE state

6.7.4.3.1 State description

This state wails for COMWAKE to be fully received.

6.7.4.3.2 Transition ~~SP21~~SP18:SATA_AwaitNoCOMWAKE to ~~SP22~~SP19:SATA_AwaitALIGN

This transition shall occur when this state receives a COMWAKE Completed parameter from the SP receiver.

6.7.4.4 ~~SP22~~SP19:SATA_AwaitALIGN state

6.7.4.4.1 State description

This state shall send a Transmit D10.2s parameter to the SP transmitter, start the ALIGN detect timeout timer, and wait for an ALIGN to be received or an ALIGN detect timeout.

The SAS device shall start transmitting D10.2 characters no later than 20 G1 dwords (533 ns) after COMWAKE was deasserted.

6.7.4.4.2 Transition ~~SP22~~SP19:SATA_AwaitALIGN to ~~SP23~~SP20:SATA_AdjustSpeed

This transition shall occur when this state receives an ALIGN0 Detected parameter from the SP receiver at any of its supported rates.

6.7.4.4.3 Transition ~~SP22~~SP19:SATA_AwaitALIGN to SP1:OOB_COMINIT

This transition shall occur when the ALIGN detect timeout timer expires.

6.7.4.5 ~~SP23~~SP20:SATA_AdjustSpeed state

6.7.4.5.1 State description

This state waits for the SP transmitter to adjust to the same rate of the ALIGNs that were detected by the receiver circuitry.

6.7.4.5.2 Transition ~~SP23~~SP20:SATA_AdjustSpeed to ~~SP24~~SP21:SATA_TransmitALIGN

This transition shall occur when this state receives a Transmitter Ready parameter from the SP transmitter.

6.7.4.6 ~~SP24~~SP21:SATA_TransmitALIGN state

6.7.4.6.1 State description

This state shall send the Transmit ALIGN0s parameter to the SP transmitter.

When the SP receiver detects three back-to-back non-ALIGN primitives, the state machine transitions to state ~~SP26~~SP22:SATA_PHY_Ready.

6.7.4.6.2 Transition ~~SP24~~SP21:SATA_TransmitALIGN to ~~SP26~~SP22:SATA_PHY_Ready

This transition shall occur when this state receives a NonALIGNS Received parameter from the SP receiver.

6.7.4.7 ~~SP26~~SP22:SATA_PHY_Ready state

6.7.4.7.1 State description

In this state, the SP state machine hands control over dword transmission to the DWS state machine. The SP receiver monitors the input dword stream looking for COMINIT, and the DWS state machine monitors the input dword stream looking for

This state shall send a PhyReady (SATA) parameter to the DWS state machine.

6.7.4.7.2 Transition ~~SP26~~SP22:SATA_PHY_Ready to SP1:Reset

This transition shall occur when this state receives a COMINIT Received parameter from the SP receiver, or TBD from the DWS state machine.

6.7.4.7.3 Transition ~~SP26~~SP22:SATA_PHY_Ready to ~~SP28~~SP24:SATA_PM_Partial

This transition shall occur if this state receives an Enter Partial request from the link layer.

6.7.4.7.4 Transition ~~SP26~~SP22:SATA_PHY_Ready to ~~SP27~~SP23:SATA_PM_Slumber

This transition shall occur if this state receives an Enter Slumber request from the link layer.

6.7.4.8 ~~SP27~~SP23:SATA_PM_Partial state

6.7.4.8.1 State description

Exit from this state is driven from receipt of COMWAKE or de-assertion of the partial signal.

The phy shall remember if COMWAKE was detected during ~~SP27~~SP23:SATA_PM_Partial to determine if the wakeup request was originated from the host or the phy.

6.7.4.8.2 Transition ~~SP27~~SP23:SATA_PM_Partial to ~~SP19~~SP16:SATA_COMWAKE

This transition shall occur when this state receives a Exit Partial request from the link layer.

6.7.4.8.3 Transition ~~SP27~~SP23:SATA_PM_Partial to ~~SP21~~SP18:SATA_AwaitNoCOMWAKE

This transition shall occur when this state receives a COMWAKE Detected parameter from the SP receiver.

6.7.4.9 ~~SP28~~SP24:SATA_PM_Slumber state

6.7.4.9.1 State description

Exit from this state is driven from receipt of COMWAKE or de-assertion of the slumber signal.

The phy shall remember if COMWAKE was detected during ~~SP28~~SP24:SATA_PM_Slumber to determine if the wakeup request was originated from the link layer or the phy.

6.7.4.9.2 Transition ~~SP28~~SP24:SATA_PM_Slumber to ~~SP19~~SP16:SATA_COMWAKE

This transition shall occur when this state receives a Exit Slumber request from the link layer.

6.7.4.9.3 Transition ~~SP28~~SP24:SATA_PM_Slumber to ~~SP21~~SP18:SATA_AwaitNoCOMWAKE

This transition shall occur when this state receives a COMWAKE Detected parameter from the SP receiver.

6.7.4.10 SP25:SATA_DelaySpin

6.7.4.10.1 State description

This state shall send the SpinupHold parameter to SMP protocol layer to indicate that a SATA device has been detected and is waiting for a response before it continues and starts the spin up process. This state shall also send a Broadcast Event Notify.

Exit from this state is driven from receipt of link reset from SMP.

6.7.4.10.2 Transition SP24:SATA_DelaySpin to SP16:SATA_COMWAKE

This transition shall occur when this state receives a link reset from SMP. (SMP Phy Control Request).

6.8 SAS phy dword synchronization (DWS) state machine

6.8.1 Overview

Each SAS phy includes a SAS phy dword synchronization state machine (DWS).

The DWS state machine establishes the same dword boundaries at the receiver as at the attached transmitter by searching for control characters. A receiver in the SAS phy monitors and decodes the incoming data stream and forces K28.5 characters into the first byte position to effectively perform dword alignment. The receiver continues to reestablish dword alignment by forcing received K28.5 characters into the first byte position until a valid primitive is detected. The resultant primitives, dwords and valid dword indicators (e.g., encoding error indicators) are sent to the DWS machine to enable the state machine to determine the dword synchronization policy.

After dword synchronization has been achieved, the DWS state machine monitors invalid dwords that are received. When an invalid dword is detected, it requires two valid dwords to nullify its effect. When four invalid dwords are detected without nullification, dword synchronization is considered lost.

While dword synchronization is lost, the data stream received is invalid and dwords shall not be passed to the link layer.

The DWS state machine contains the following states:

a) DWS0:AcquireSync (see 6.8.2);
b) DWS1:Valid1 (see 6.8.3);
c) DWS2:Valid2 (see 6.8.4);
d) DWS3:SyncAcquired (see 6.8.5);
e) DWS4:Lost1 (see 6.8.6);
f) DWS5:Lost1Recovered (see 6.8.7);
g) DWS6:Lost2 (see 6.8.8);
h) DWS7:Lost2Recovered (see 6.8.9);
i) DWS8:Lost3 (see 6.8.10); and
j) DWS9:Lost3Recovered (see 6.8.11).

The state machine shall start in the DWS0:AcquireSync state after power on.

The DWS state machine receives the following parameters from the SP state machine:

a) PhyNotReady;
b) PhyReady (SAS); and

c) PhyReady (SATA).

Figure 55 shows the DWS state machine.

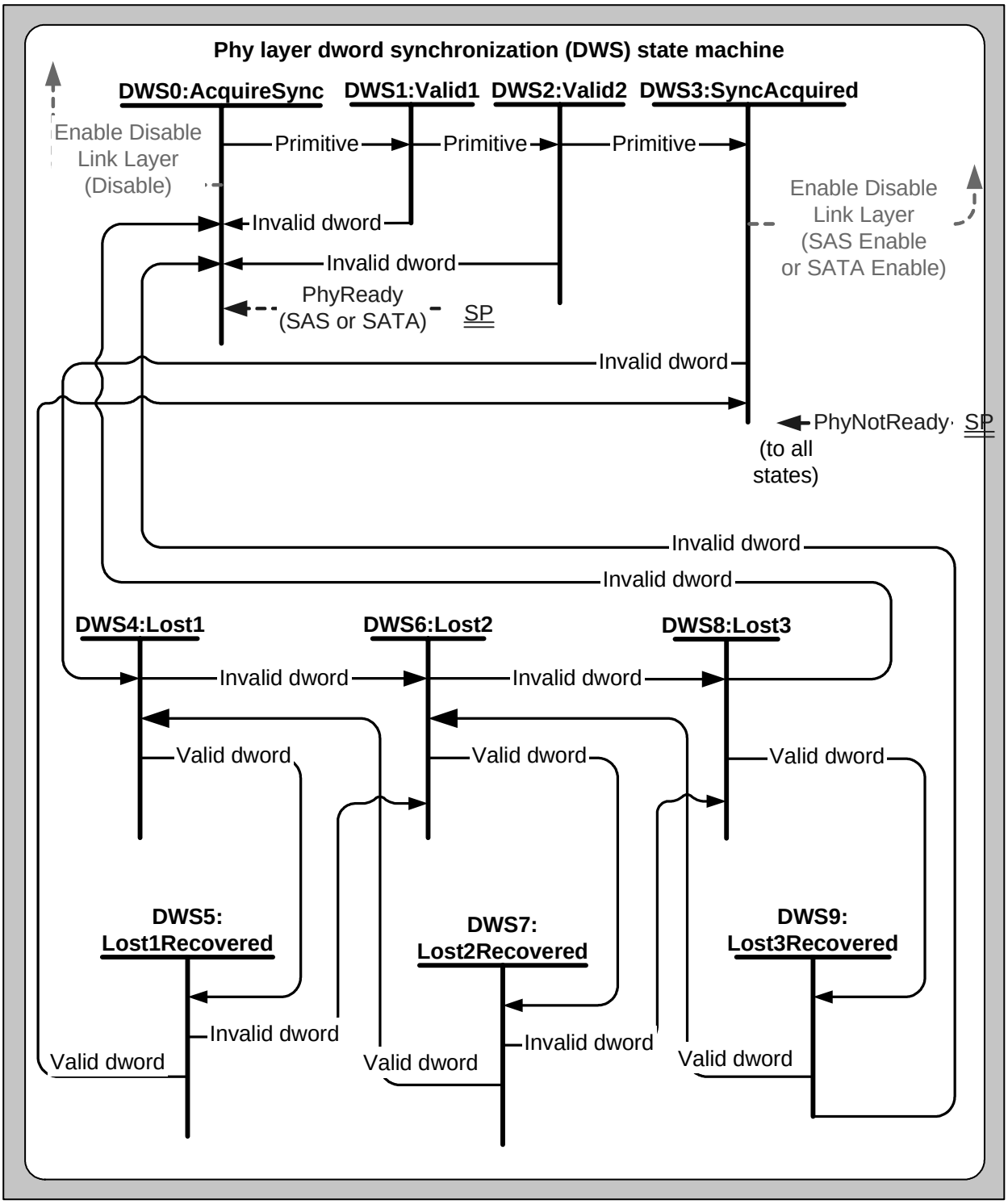


Figure 55 — Dword synchronization state machine

6.8.2 DWS0:AcquireSync state

6.8.2.1 State description

This state is entered upon power on loss ~~of~~ or previous dword ~~synchronization~~ synchronization. On entry into this state, this state shall send an Enable Disable Link Layer (Disable) confirmation to the link layer.

In this state, the receiver monitors the input data stream and forces each K28.5 character it detects into the first byte position for possible dword alignment. This is the only state that shall modify dword alignment.

6.8.2.2 Transition DWS0:AcquireSync to DWS1:Valid1

This transition shall occur when a PhyReady parameter has been received from the SP state machine and a valid primitive (e.g., a known dword starting with K28.5) is detected.

6.8.3 DWS1:Valid1 state

6.8.3.1 State description

This state is reached after one valid primitive has been detected. In this state, the receiver shall monitor the input data stream looking for a second valid primitive or an invalid dword.

6.8.3.2 Transition DWS1:Valid1 to DWS2:Valid2

This transition shall occur when a valid primitive is detected.

6.8.4 DWS2:Valid2 state

6.8.4.1 State description

This state is reached after the receiver has detected two valid primitives. In this state, the receiver shall monitor the input data stream looking for a third valid primitive or an invalid dword.

6.8.4.2 Transition DWS2:Valid2 to DWS3:SyncAcquired

This transition shall occur when a valid primitive is detected.

6.8.4.3 Transition DWS2:Valid2 to DWS0:AcquireSync

This transition shall occur when an invalid dword is detected.

6.8.5 DWS3:SyncAcquired state

6.8.5.1 State description

This state is reached when the receiver has detected three valid primitives without adjusting the dword synchronization. This state shall send one of the following confirmations to the link layer:

a) an Enable Disable Link Layer (SAS Enable) confirmation if it has received the PhyReady (SAS) parameter from the SP state machine; or
b) an Enable Disable Link Layer (SATA Enable) confirmation if is has received the PhyReady (SATA) parameter from the SP state machine.

The most recently received primitive shall be considered valid for processing by the link layer.

In this state, the receiver shall monitor the incoming data stream looking for an invalid dword, which indicates that dword synchronization might be lost.

6.8.5.2 Transition DWS3:SyncAcquired to DWS4:Lost1

This transition shall occur when an invalid dword (i.e., the first invalid dword) is detected. An expander forwarding the dword to another link shall replace the invalid dword with ERROR or SATA_ERROR (see ~~7.14.13~~7.14.13).

6.8.6 DWS4:Lost1 state

6.8.6.1 State description

This state is reached when one invalid dword has been received and not nullified. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.6.2 Transition DWS4:Lost1 to DWS5:Lost1Recovered

This transition shall occur when a valid dword is detected.

6.8.6.3 Transition DWS4:Lost1 to DWS6:Lost2

This transition shall occur when an invalid dword is detected. An expander forwarding the dword to another link shall replace the invalid dword with ERROR or SATA_ERROR (see ~~7.14.13~~7.14.13).

6.8.7 DWS5:Lost1Recovered state

6.8.7.1 State description

This state is reached when a valid dword has been received, and another valid dword will nullify the previous invalid dword. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.7.2 Transition DWS5:Lost1Recovered to DWS3:SyncAcquired

This transition shall occur when a valid dword is detected.

6.8.7.3 Transition DWS5:Lost1Recovered to DWS6:Lost2

This transition shall occur when an invalid dword is detected. An expander forwarding the dword to another link shall replace the invalid dword with ERROR or SATA_ERROR (see ~~7.14.13~~7.14.13).

6.8.8 DWS6:Lost2 state

6.8.8.1 State description

This state is reached when two invalid dwords has been received and not nullified. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.8.2 Transition DWS6:Lost2 to DWS7:Lost2Recovered

This transition shall occur when a valid dword is detected.

6.8.8.3 Transition DWS6:Lost2 to DWS8:Lost3

This transition shall occur when an invalid dword is detected. An expander forwarding the dword to another link shall replace the invalid dword with ERROR or SATA_ERROR (see ~~7.14.13~~7.14.13).

6.8.9 DWS7:Lost2Recovered state

6.8.9.1 State description

This state is reached when a valid dword has been received, and another valid dword will nullify the previous invalid dword. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.9.2 Transition DWS7:Lost2Recovered to DWS4:Lost1

This transition shall occur when a valid dword is detected.

6.8.9.3 Transition DWS7:Lost2Recovered to DWS8:Lost3

This transition shall occur when an invalid dword is detected. An expander forwarding the dword to another link shall replace the invalid dword with ERROR or SATA_ERROR (see ~~7.14.13~~7.14.13).

6.8.10 DWS8:Lost3 state

6.8.10.1 State description

This state is reached when three invalid dwords has been received and not nullified. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.10.2 Transition DWS8:Lost3 to DWS9:Lost3Recovered

This transition shall occur when a valid dword is detected.

6.8.10.3 Transition DWS8:Lost3 to DWS0:AcquireSync

This transition shall occur when an invalid dword (i.e., the fourth non-nullified invalid dword) is detected.

6.8.11 DWS9:Lost3Recovered state

6.8.11.1 State description

This state is reached when a valid dword has been received, and another valid dword will nullify the previous invalid dword. In this state, the receiver shall monitor the incoming data stream looking for a valid or invalid dword.

6.8.11.2 Transition DWS9:Lost3Recovered to DWS6:Lost2

This transition shall occur when a valid dword is detected.

6.8.11.3 Transition DWS9:Lost3Recovered to DWS0:AcquireSync

This transition shall occur when an invalid dword (i.e., the fourth non-nullified invalid dword) is detected.

6.9 Spin-up

If a SAS target device receives COMSAS during the reset sequence, it shall not spin-up until ~~a SCSI START STOP UNIT command is received with the START bit set to one~~ allowed by the ~~appropriate logical unit~~power condition state machine (see 10.1.4).

If a SAS target device supporting SATA does not receive COMSAS during the reset sequence, it shall follow SATA spin-up rules (see SATA).

NOTE 10 A SATA target device with rotating media spins up:
a) automatically after power on (allowed by SATA);
b) after its phy is enabled (allowed by SATA);
c) after the reset sequence has completed (recommended by SATA); or
d) after the Power Up in Standby flag is cleared by an application (if the ATA Power Up in Standby feature is implemented).

The ATA Power Up in Standby feature is not widely implemented, since it requires the target device to include a nonvolatile memory to remember the state of the Power Up in Standby flag. Desktop-class disk drives do not typically have nonvolatile memory storage.

Expanders attached to SATA target devices may halt the automatic phy reset sequence to delay spin-up; see ~~10.3.1.11~~10.3.1.10.

NOTE 11 Enclosures supporting both SATA-only target devices and SAS target devices may need to sequence power to each target device to avoid excessive power consumption during power on, since the SATA-only target devices may spin-up automatically after power on.