ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Proposal for Constant Latency MII to 8N/8N+1 Encoding in the 100BASE-T1L PCS

Brian Murray
Jacobo Riesco
Philip Curran

Motion #n

Motion #n

Move that the IEEE P802.3dg Task Force adopt slides 3 to 8 + 10 of Murray_3dg_01_11132024.pdf

M: Brian Murray

S:

Technical (>75%)

Introduction

- This a proposal for the constant latency MII to 8N/8N+1 encoding used in the 100BASE-T1L PCS

Control Codes and Mode Encoding/Decoding Table

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

3	4	5	6	7		
Mode M(n)[0:1]		**Control Code C(n)[0:2]**			**Symbol**	**Definition**
0	-	0	0	0	Q	Sequence Ordered Set Control Code
0	-	0	0	1	E	Transmit Error Propagation
0	-	0	1	0	I	Normal Inter-Frame (Idle) with loc_phy_ready = OK
0	-	0	1	1	**Su**	Start of Packet with leading Idle/LPI
0	-	1	0	0	Tp	End of Packet
0	-	1	0	1	L	Assert Low Power Idle (LPI)
0	-	1	1	0	Ix	Normal Inter-Frame (Idle) with loc_phy_ready = NOT_OK
0	-	1	1	1	Sp	Start of Packet
1	0	0	0	0	**Tu**D0	Dribble Nibble, Data = 0x0
1	0	0	0	1	TuD8	Dribble Nibble, Data = 0x8
1	0	0	1	0	TuD4	Dribble Nibble, Data = 0x4
1	0	0	1	1	TuDC	Dribble Nibble, Data = 0xC
1	0	1	0	0	TuD2	Dribble Nibble, Data = 0x2
1	0	1	0	1	TuDA	Dribble Nibble, Data = 0xA
1	0	1	1	0	TuD6	Dribble Nibble, Data = 0x6
1	0	1	1	1	TuDE	Dribble Nibble, Data = 0xE
1	1	0	0	0	TuD1	Dribble Nibble, Data = 0x1
1	1	0	0	1	TuD9	Dribble Nibble, Data = 0x9
1	1	0	1	0	TuD5	Dribble Nibble, Data = 0x5
1	1	0	1	1	TuDD	Dribble Nibble, Data = 0xD
1	1	1	0	0	TuD3	Dribble Nibble, Data = 0x3
1	1	1	0	1	TuDB	Dribble Nibble, Data = 0xB
1	1	1	1	0	TuD7	Dribble Nibble, Data = 0x7
1	1	1	1	1	TuDF	Dribble Nibble, Data = 0xF

* From Lo_3dg_01a_0724
Modified

- M(n)[0:1] = 00 – No more control codes
- M(n)[0:1] = 01 – More control codes
- M(n)[0] = 1 - 16 x Tu control codes
  - More control codes implied

Sequence Ordered Sets

- Optional support for sequence ordered sets
- SEQen bit added to the InfoField to allow support for sequence ordered sets to be negotiated with the link partner
  - If either link partner has SEQen = 0 then sequence ordered sets are disabled
- If sequence ordered sets are disabled
  - If the PHY receives a sequence ordered set on the MII the PHY Tx will encode it as Idles
  - If the PHY Rx receives a sequence ordered set control code (Q) at the decoder, this is not expected or supported and is treated like any other error
    - Following the error handling rules in such cases a False Carrier will be signalled at the MII
- If sequence ordered sets are enabled, they are transparently conveyed between the local and remote MII

Normal Inter-Frame (Idle)

- Keep two versions of Normal Inter-Frame
  - **I** when loc_phy_ready = OK or **Ix** when loc_phy_ready = NOT_OK
  - As in Clause 97
- Normal Inter-Frame encoding indicates whether the PHY is ready to receive data or not (loc_phy_ready = OK/NOT_OK)
  - Decoded as rem_phy_ready = OK/NOT_OK in the link partner
  - Synchronizes link_status = OK between link partners
  - Cannot rely on sequence ordered sets for this synchronization as existing MACs do not support them

Encoding Rules for Error Handling

- Packets are delimited with Start of Packet and End of Packet symbols
  - **Sp** symbol used for packets starting on even cycles - aligned
  - **Su** (was Cs) for packet starting on odd cycles - unaligned
    - The start of packet is always encoded using Sp or Su
    - If transmit error propagation is encoded at the MII during the start of packet, an error is encoded in the following octet
  - **Tp** symbol used for packets ending on even cycles - aligned
  - 16 x **Tu** (was CD) symbols for packets ending on odd cycles – unaligned
    - The end of packet is always encoded using Tp or Tu

Decoding Rules for Error Handling

- PCS receive functions or Multi-G RS functions are incorporated into the PHY decoder
  - RX_DV will be asserted in response to the reception of a start of packet control code if the previous nibble was a normal inter-frame
  - RX_DV will be de-asserted when a control code other than an error is received
    - When RX_DV is de-asserted because of a control code other than end of packet, RX_ER will be asserted before RX_DV is de-asserted
  - If a symbol other than Idle, LPI, start of packet or sequence ordered set control code (if supported), is received following an idle nibble or a sequence ordered set (if supported), then a false carrier indication is encoded onto the MII RX
    - False carrier is held until an Idle, LPI or a sequence ordered set control code (if supported) is received
    - This is the same as 100BASE-X, 1000BASE-X, 1000BASE-T, 100BASE-T1, 10BASE-T1L and 10BASE-T1S (Clauses 24, 36, 40, 96, 146 and 147)

8N/(8N+1) Encoding

- Defined by the following pseudo-code (Modifications from Clause 97, per [Lo_3dg_01a_0724](Lo_3dg_01a_0724)), where N is the number of octets encoded in a block
  - N = 8 when the Reed-Solomon FEC is used and N = 2 when it is not used
  - Octets within a block are numbered using and increasing index n, from 0 to N-1, with n = 0 being the first octet of the block presented on the MII interface.

```
TC[n]           : 0 if octet n is encoded as a data packet octet (the octet n contains two MII data nibbles, TXD[2n][0:3] and  TXD[2n+1][0:3]); 1 otherwise.
TC[-1]          : 1 by definition
TD[n][0:7]      : MII octet n (TD[n][0:3] = TXD[2n][0:3], TD[n][4:7] = TXD[2n+1][0:3]) if TC[n] = 0
B[0:8N]         :  8N+1 block. Bit 0 transmitted first
OR(n)           : Bitwise OR of TC[n:N-1]
OR(N)           : 0 by definition
NEXT(n)[0:2]    : Bit position of lowest bit in TC[n:N-1] that is a 1. Bit 2 is MSB
C(n)[0:2]       : MII control code n, corresponding to MII data nibbles 2n, 2n+1 as per control codes and mode encoding table
M(n)[0:1]       : MII mode n, corresponding to MII data nibbles 2n, 2n+1
                  M(n)[0] = 1 if encoded symbol is CD; else 0
                  M(n)[1] = TXD[2n][0] if encoded symbol is CD; else OR(n+1)

B[0]            = OR(0)
B[8n+1:8n+3]    = TD[n][0:2]     if OR(n) = 0
                  NEXT(n)[0:2]   else if TC[n-1] = 1
                  TD[n-1][5:7]   else
B[8n+4:8n+5]    = TD[n][3:4]     if OR(n) = 0
                  M(n)[0:1]      else if TC[n] = 1
                  TD[n][0:1]     else
B[8n+6:8n+8]    = TD[n][5:7]     if OR(n) = 0
                  C(n)[0:2]      else if TC[n] = 1
                  TD[n][2:4]     else
```

8N/(8N+1) Encoding – Proposed Text

- This text is taken from page 8 of Lo_3dg_01b_1124

The N octets are mapped to 8N+1 bits as described in the following pseudo code, where N = 2 for low latency mode and N = 8 for long reach mode.

```
N = number of octets encoded into block.
Octets numbered n = 0,1,2,…,N-1.

octet 0 is the first one presented to the encoder.

octet -1 is by definition not a Tu* symbol
TC[-1] = 1 by definition

if octet n is a data symbol and octet n-1 is not a Tu* symbol then
    TC[n] = 0
else
    TC[n] = 1

NEXT(n)[0:2] = bit position of lowest bit in TC[n:N-1] that is a 1. Bit 2 is
MSB.

NEXT(n)[4] = 0 if Bitwise SUM of TC[n:N-1] = 1, else 1

if TC[n] = 1 then
    TD[0:2] is undefined
    if octet n is one of the Tu* symbols then
        TD[n][3:7] = {Mode[0:1], Control[0:2] of the corresponding control
        symbol as defined in Table ZZZ-B.
    else
        TD[n][4] = NEXT(n)[4]
        TD[n][3, 5:7] = {Mode[0], Control[0:2]} of the corresponding
        control symbol as defined in Table ZZZ-B.
else
    TD[n][0:7] = octet n {first nibble TXD[0:3], second nibble TXD[0:3]}


B[0:8N] is the 8N+1 block. Bit 0 transmitted first.

OR(n) = Bitwise OR of TC[n:N-1]

B[0] = OR(0)

B[8n+1:8n+3] = TD[n][0:2] – if OR(n) = 0
               NEXT(n)[0:2] – if OR(n) = 1 AND TC[n-1] = 1
               TD[n-1][5:7] – if OR(n) = 1 AND TC[n-1] = 0

B[8n+4:8n+8] = TD[n][3:7] – if OR(n) = 0
               TD[n][3:7] – if OR(n) = 1 AND TC[n] = 1
               TD[n][0:4] – if OR(n) = 1 AND TC[n] = 0
```

Editor's Note:
I do not think an equivalent to clauses 97.3.2.2.6 is needed since this is concisely covered as a combination in Tables ZZZ-A and ZZZ-B.