Application Note
Version 0.2

TLK110: TDR and ALCD Use

TLK110

Introduction

This document describes the use of the TDR and ALCD features of the TLK110 Ethernet PHY. This is a preliminary version based on early characterization and will be updated over time.

Contents

1 TDR USE

Below is the preliminary use case, this is subject to change with further characterization and device development.

In order to operate the TDR properly the following registers need to be set prior to operation:

1. Write register 171 bits (3:0) value of C(h) → set initial skip to higher values, helps to ignore self returns
2. Write register 173 bits (15:8) value of FF(h) → set cfg_tdr_seg_duration_seg5 to higher value to allow high cable range detection (up to 200m)
3. Write register 177 bits (12:8) value of 18(h) → set seg1_high_threshold higher value

The formula to convert the location from the registers to meters is:

1. Read location from the results register.
2. If (Value == C(h))
   → Range is 0-3 [m]. Due to self returns we do not have better resolution in 0-3 [m].
3. Else:
   1. Convert the value from hexadecimal to decimal = Decimal_Value
   2. IL = (Decimal_Value x 0.8621) - 8 (IL = Intermediate Location)
      Takes into account pulse width, cable propagation and offset
   3. FL = IL + ((70 – IL) x 0.01) (FL = Final Location)
      Takes into account pulse shape smearing over the cable.
   4. FL = Final Location = real length of the reflection.

Example:

Hex_Value = 33h	→ Decimal_Value = 51	
IL = (51 x 0.8621) - 8	→ 35.97m	(Intermediate Location)
FL = (35.97 + ((70 – 35.97) x 0.01)	→ 36.31m	(Final Location)

2 ALCD USE

Below is the preliminary use case, this is subject to change with further characterization and device development.

In order to operate the ALCD properly the following actions need to be performed:

1. Set efuse values to default.
    a. Run script efuse_general.txt (set all efuse bits to correct value)

2. check valid link up state

3. read CAGC
    a. read register 0x0215(h) bits 15:12

4. read DFE sum COEFF 1 to 20 (h)
    a. write register 0x0155(h) value of 8001(h)
    b. Y(h) = read register 0x0155(h) bits 11:04
    c. M(dec) = 2's comp of Y (dec). {m = 255 - HEX2DEC(Y) +1}

5. read FAGC
    a. Z(h) = read register 0x0215(h) bits 11:4
    b. P(dec) = hex2dec(z)

6. multiply FAGC x DFE sum COEFF 1 to 20 (h)
    a. X(dec) = P x M

7. use 2^{nd} degree polynomial with coefficients according to CAGC(h)

    a. estimated length = poly_coeff_2 *X^2 + poly_coeff_1 *X + offset

CAGC	poly_coeff_2	poly_coeff_1	offset
E	-2.18E-08	0.0029758	-42.007
D	-4.52E-08	0.0049168	-74.548
C	-1.47E-08	0.0024888	-21.355
B	-1.88E-08	0.0031974	-36.946
A	-1.18E-08	0.0024718	-14.336
9	-4.08E-09	0.001475	22.849
8	-7.85E-09	0.0020259	8.288
7	-6.57E-09	0.0019358	14.181
0	-5.41E-09	0.0017626	24.186
1	-7.50E-09	0.0023272	-4.1926
2	-2.21E-09	0.0011374	62.198
3	6.66E-09	-0.001359	244.56