



Melissa Chang

摘要

本应用手册概述了时域反射法 (TDR) 如何帮助解决基于以太网的通信系统的各种电缆故障难题。本应用手册介绍了使用 DP83867 和 DP83869 的 TDR 功能在系统中实现电缆诊断功能的过程。

内容

1 时域反射法	2
1.1 示例连接.....	2
1.1.1 开路电缆.....	2
1.1.2 短路电缆.....	3
2 DP83867 和 DP83869 TDR 实现	4
2.1 TDR 配置.....	4
2.2 TDR 算法.....	5
2.2.1 TDR 算法示例流程.....	7
2.2.2 TDR 算法 Matlab 示例.....	8
3 总结	9
4 参考文献	10

插图清单

图 1-1. 开路电缆.....	2
图 1-2. 短路电缆.....	3
图 2-1. DP83867 和 DP83869 TDR 算法.....	6
图 2-2. TDR 算法示例流程图.....	7

表格清单

表 2-1. TDR 寄存器写入.....	4
表 2-2. TDR 寄存器结果.....	4
表 2-3. 5x3 矩阵格式.....	5
表 2-4. 初始变量.....	5
表 2-5. Prop_dly 值.....	5
表 2-6. TDR 数据表示例.....	7

商标

所有商标均为其各自所有者的财产。

1 时域反射法

TDR 仅适用于双绞线连接。TDR 涉及在 TX 和 RX 对上发送脉冲并观察任一对上的结果。通过测量电压幅度、极性和时间间隔，PHY 可以确定故障的性质和位置。DP83867/DP83869 TDR 发生器在 TX 和 RX 通道上发送脉冲，然后监测这两个通道以观察反射。该发生器一次向一个通道发送一个脉冲，如果在另一个通道上观察到反射，则 PHY TDR 会意识到导线已交叉。DP83867/DP83869 可为每个发送和接收通道检测一个峰值。TDR 可用于检测以下故障：

- 电缆开路
- 电缆短路
- TX/RX 对交叉接线
- 阻抗不连续

TDR 只能在链路断开时使用。

1.1 示例连接

下文说明了可使用 TDR 的示例连接。

1.1.1 开路电缆

开路电缆很容易诊断，因为它会产生强烈的反射。在另一个通道上不会观察到反射。开路导致的反射与发送的脉冲（正极性）同相。任何类型的电感阻抗不连续都会产生同相反射，幅度取决于阻抗不连续的程度。

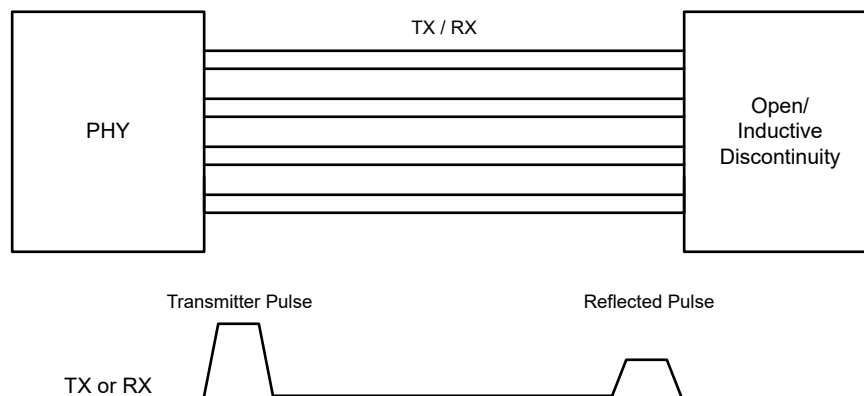


图 1-1. 开路电缆

1.1.2 短路电缆

短路的电缆也会产生强反射，但这种反射与原始脉冲（负极性）异相。任何类型的电容阻抗不连续都会产生异相反射。幅度取决于阻抗不连续的程度。

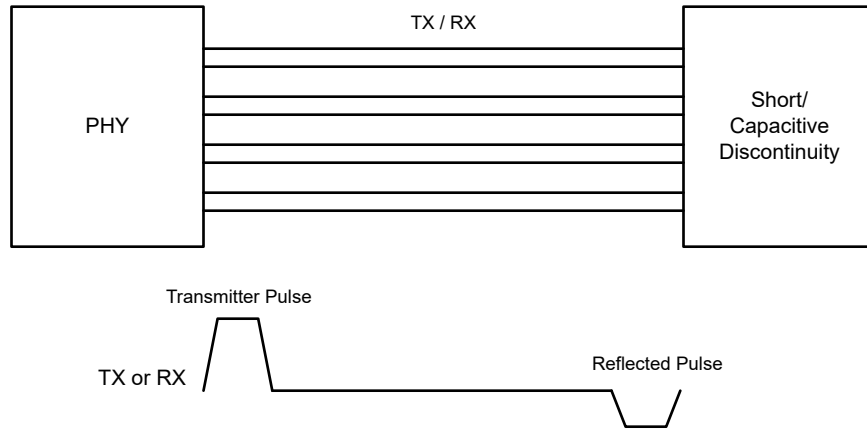


图 1-2. 短路电缆

2 DP83867 和 DP83869 TDR 实现

本文档中将讨论 DP83867 和 DP83869 中的 TDR 实现。

2.1 TDR 配置

要手动运行 TDR，必须在配置寄存器之前关闭链路。若要使用 TDR 功能，请遵循如下步骤：

1. 配置寄存器，如表 2-1 所示。要了解如何配置扩展寄存器，请查看 [DP83867](#) 或 [DP83869](#) 数据表。

表 2-1. TDR 寄存器写入

寄存器地址	值
0x0189	0x0000
0x0186	0x294A
0x0187	0x0A9B
0x001E	0x0003

2. 运行之前的配置后，检查 $0x001E[1] = 1$ 以确认 TDR 已成功完成。然后，读取以下扩展寄存器：

表 2-2. TDR 寄存器结果

寄存器
0x0190
0x0191
0x0192
0x019a
0x019b
0x019c
0x01a5

3. 有关如何处理从寄存器收集的数据以确定电缆故障类型和位置的信息，请参阅下一节“[TDR 算法](#)”。

2.2 TDR 算法

本节介绍如何处理 TDR 寄存器数据。以下所有步骤都必须在代码中执行。

1. 定义 5x3 矩阵，如表 2-3 所示。然后，将每一列定义为一个数组，并分别命名为 `peak_index`、`peak_value` 和 `peak_sign`。

表 2-3. 5x3 矩阵格式

peak_index	peak_value	peak_sign
0190[7:0]	019a[7:0]	01a5[0]
0190[15:8]	019a[15:8]	01a5[1]
0191[7:0]	019b[7:0]	01a5[2]
0191[5:8]	019b[15:8]	01a5[3]
0192[7:0]	019c[7:0]	01a5[4]

2. 初始化以下变量：

表 2-4. 初始变量

变量	类型	初始值	说明
i	int	5	矩阵的当前行
阈值	int	10	为了检测到故障， <code>peak_index</code> 必须超过的阈值
threshold2	int	24 或 17	<code>peak_index</code> 必须超过这一阈值，才会在前两次迭代中检测到故障。 DP83867 : threshold2 = 24 DP83869 : threshold2 = 17
fault_detected	bool	否	定义是否检测到故障
fault_location	float	0	故障位置
prop_dly	float		传播延迟变量，具体取决于电缆类型和 EVM。有关更多信息，请参阅“Prop_dly 值”表。
失调电压	int	16	用于计算故障位置的变量
故障[fault]	int	1	定义故障是短路 (1) 还是开路 (0) 故障

表 2-5. Prop_dly 值

电缆类型	Prop_dly (DP83867)	Prop_dly (DP83869)
Cat5/5e	5.35	5.1
Cat6	5	5
Cat7	4.6	4.6

3. 使用以下算法编写一个程序。如需参考，请参阅 [MATLAB 示例代码](#)。

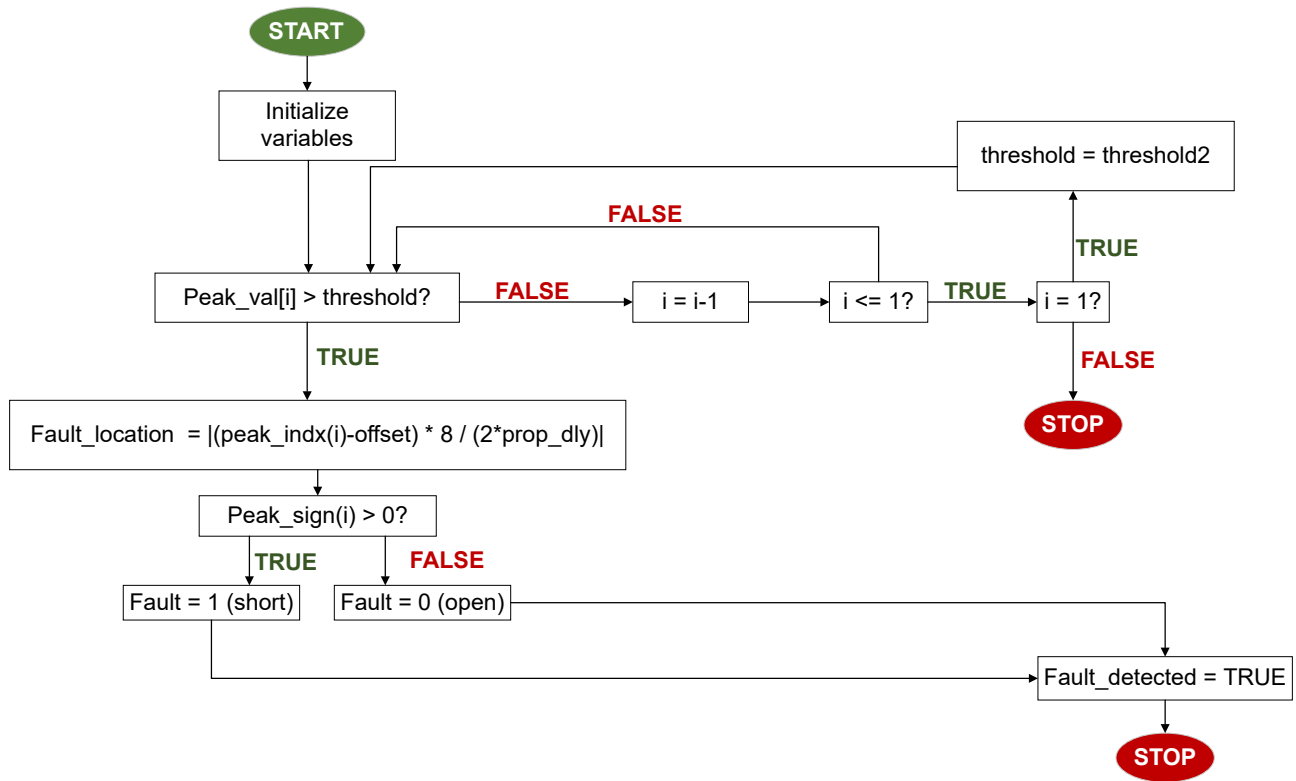


图 2-1. DP83867 和 DP83869 TDR 算法

4. 故障类型位于 *Fault* 变量中，故障位置存储在 *fault_location* 中。

2.2.1 TDR 算法示例流程

表 2-6 展示了 100m CAT5e 电缆的 TDR 结果。

表 2-6. TDR 数据表示例

peak_index (0x180)	peak_value (0x185)	peak_sign (0x18A)
F	2D	0
93	11	0
0	0	0
0	0	0
0	0	0

该算法首先检查数组 *peak_val* 的最后一个元素是否大于电流阈值 10。如果不大于 10，则移至下一行。如果大于 10，该算法将在数组 *peak_indx* 中查找相应的行，并使用该值计算故障位置。然后，该算法将查看 *peak_sign* 的相应行以确定故障类型。0 表示开路故障，非零值表示短路故障。

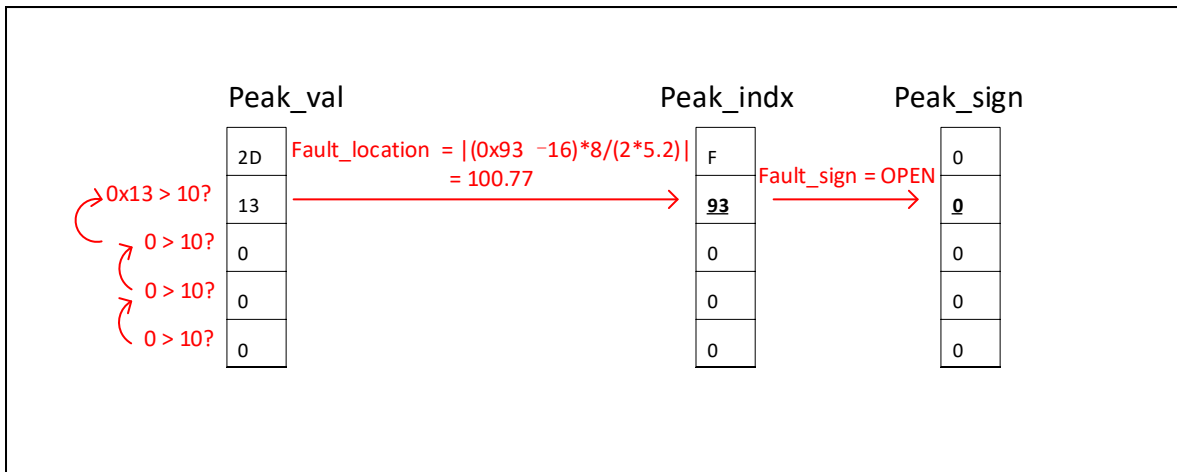


图 2-2. TDR 算法示例流程图

2.2.2 TDR 算法 Matlab 示例

以下代码是如何在 Matlab 中实现 TDR 算法的示例。程序的输入是 5x3 矩阵。

```
function [tdr_results] = tdr_869(input_matrix)

tmp = input_matrix

%iteration = tmp(1:4:end);
peak_indx = tmp(1:3:end);
peak_indx %first column in 5x3 matrix
peak_val = tmp(2:3:end);
peak_val %second column
peak_sign = tmp(3:3:end);
peak_sign %third column

thr = 10;
prop_dly = 4.6; % ns perm %propagation delay of the cable type
offset = 16;

flt_found = 0;
flt_loc = 0;
flt_sign = 0;

%% Process the TDR data from Iteration 5 to Iteration 2
for jj = 1:4 %% 1,2,3,4,5 => 5,4,3,2,1
    jj=5-jj;
    jj+1
    peak_val(jj+1)
    if peak_val(jj+1) > thr
        flt_loc = abs(((peak_indx(jj+1) - offset)*8)/(2*prop_dly));
        if peak_sign(jj+1) > 0
            flt_sign = 1;
        else
            flt_sign = 0;
        end
        flt_found = 1;
        break;
    end
end

%% Process the TDR data for Iteration 1..
%% 1st for the offset setting of 0xC..
threshold2 = 17;

if flt_found == 0
    fprintf('Peak not found in higher iterations\n')
    peak_val(1)
    if peak_val(1) > thr_seg1_2
        fprintf('peak val : %d\n', peak_val(1));
        flt_loc = abs(((peak_indx(1) - offset)*8)/(2*prop_dly));
        if peak_sign(1) > 0
            flt_sign = 1;
        else
            flt_sign = 0;
        end
        flt_found = 1;
    end
end%% Print the Results..
if flt_found == 1
    fprintf('\n');
    if flt_sign == 0
        fprintf('Fault location = %6.2f; Fault = Open\n', flt_loc);
    else
        fprintf('Fault location = %6.2f; Fault = Short\n', flt_loc);
    end
else
    fprintf('\n');
    fprintf('No Fault found\n');
end

tdr_results.flt_loc = flt_loc;
tdr_results.flt_sign = flt_sign;

return
end
```


3 总结

本应用手册介绍了 TDR 的基础知识以及如何使用 DP83867 和 DP83869 工业以太网 PHY 的 TDR 功能。

4 参考文献

- 德州仪器 (TI), [DP83867IR/CR 稳健型高抗扰性 10/100/1000 以太网物理层收发器](#) 数据表。
- 德州仪器 (TI), [DP83869HM 配备铜缆和光纤接口的高抗扰性 10/100/1000 以太网物理层收发器](#) 数据表。
- 德州仪器 (TI), [如何使用 DP83822 的 TDR 功能](#) 应用手册。
- 德州仪器 (TI), [使用 TI 以太网 PHY 解决电缆故障难题](#) 应用手册

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司