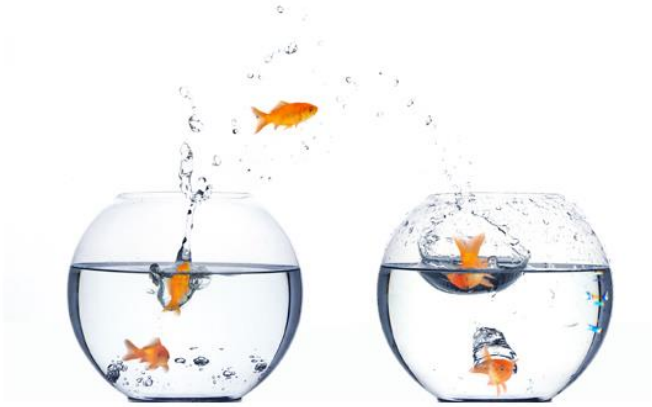
TERADYNE TIM 2017

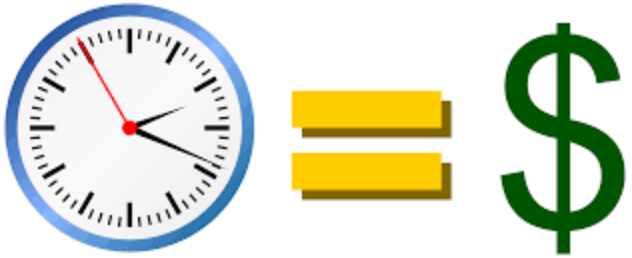
LEGACY TEST PROGRAM TRANSLATOR

Lin Yang & Sid Fluhrer
Alliance Support Partners, Inc.

ASP BACKGROUND

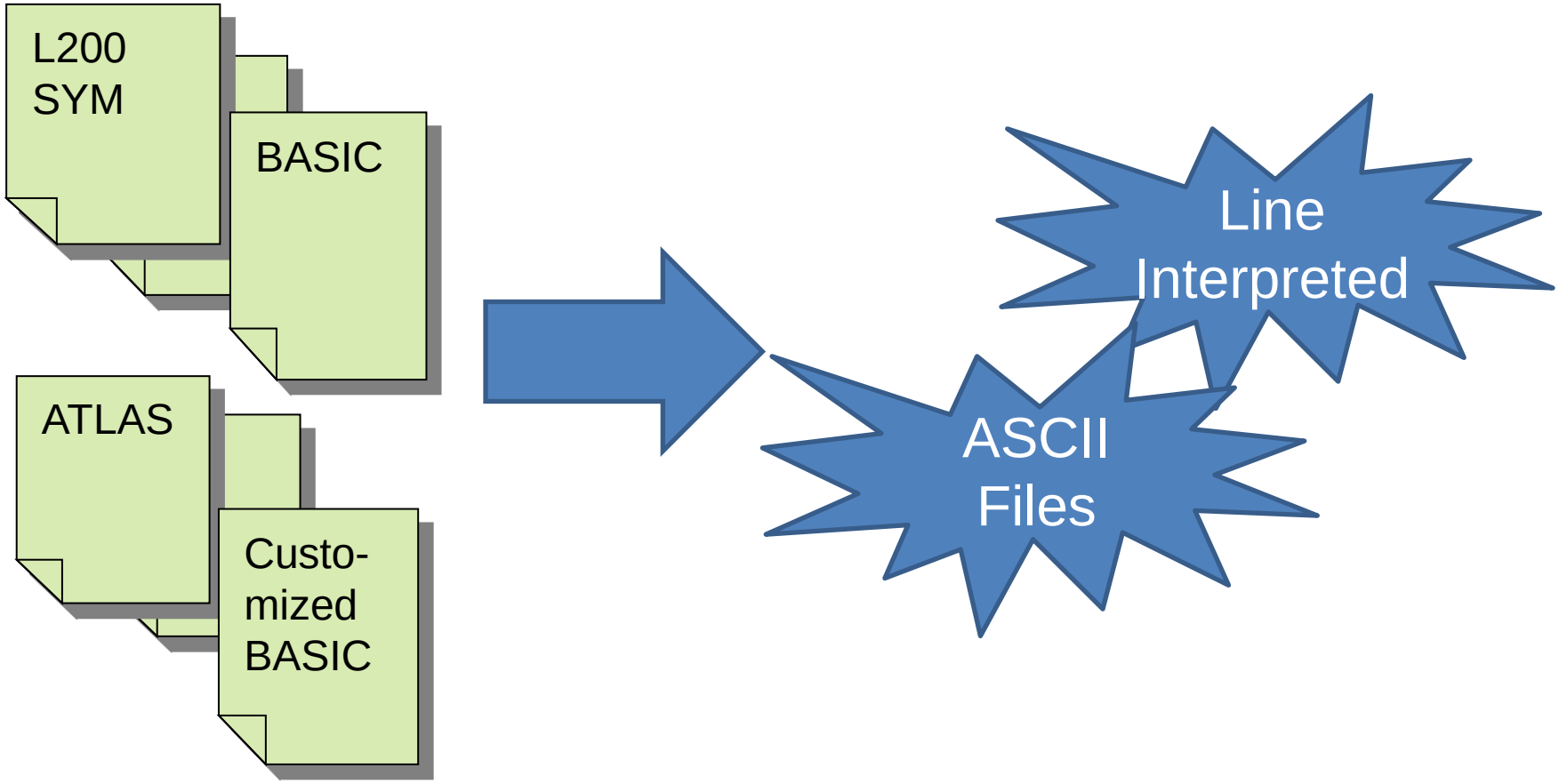


- Rehosting TPS is our specialty
- Especially large number of TPS
- Efficiency is very important!



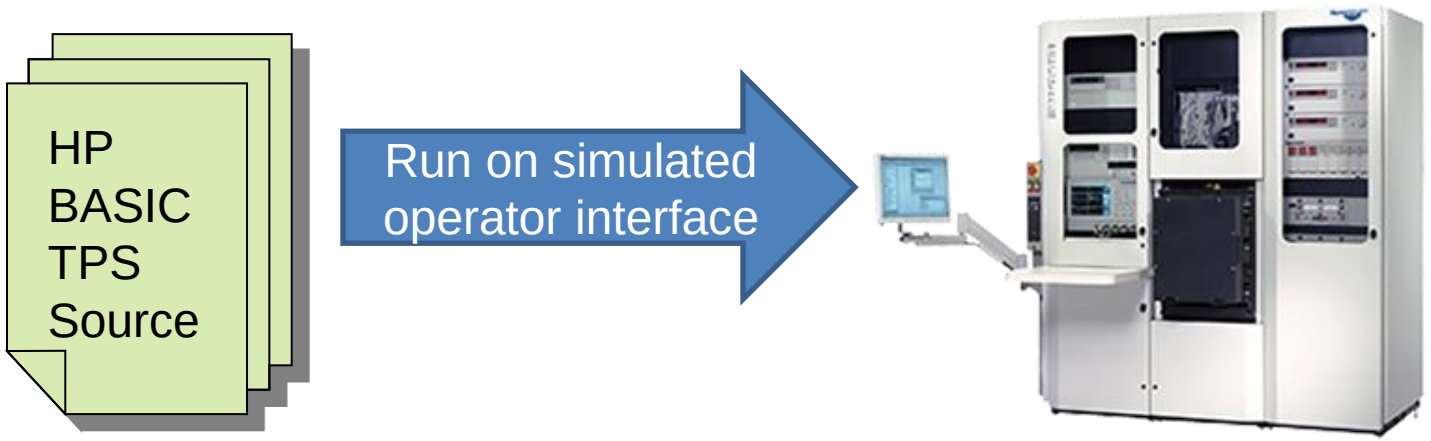
WHAT DO LEGACY TESTERS USE?

- We are rehosting test program developed in 1970s,1980s. Most of what we have seen are line interpreted ASCII test programs.



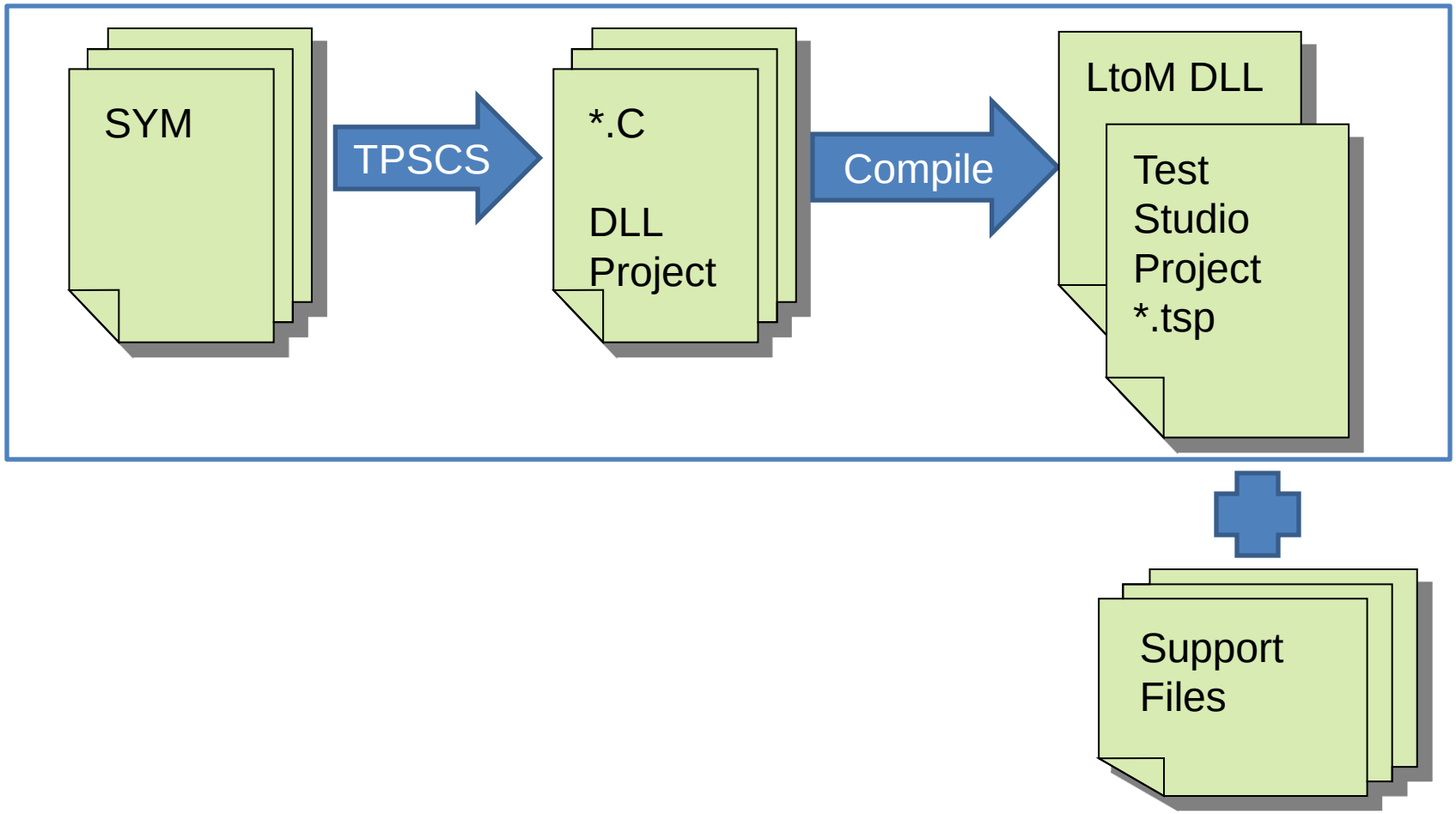
YOU CAN CHOOSE NOT TO TRANSLATE

- Some people choose not to translate or rewrite the test program
- Example : Teradyne Common Test System (CTS)
  - Supported 300 HP-BASIC style TPS
  - Developed interpreter software to run legacy test programs on new hardware
  - Development cost is high (multiple years effort), recursive testing required when interpreter update



REHOST WITH TPSCS

- Teradyne TPS Convert Studio is a complete process to rehost L200 TPS to S9100 platform



HOW ABOUT NON-L200 SOURCE?

- And there is no existing translator?
- And there are lots of them?

TERADYNE TIM 2017

CASE STUDY – TRANSLATE “BASIC” TO C++ CODE

TPS REHOST –"TRACEABILITY IS MUST"

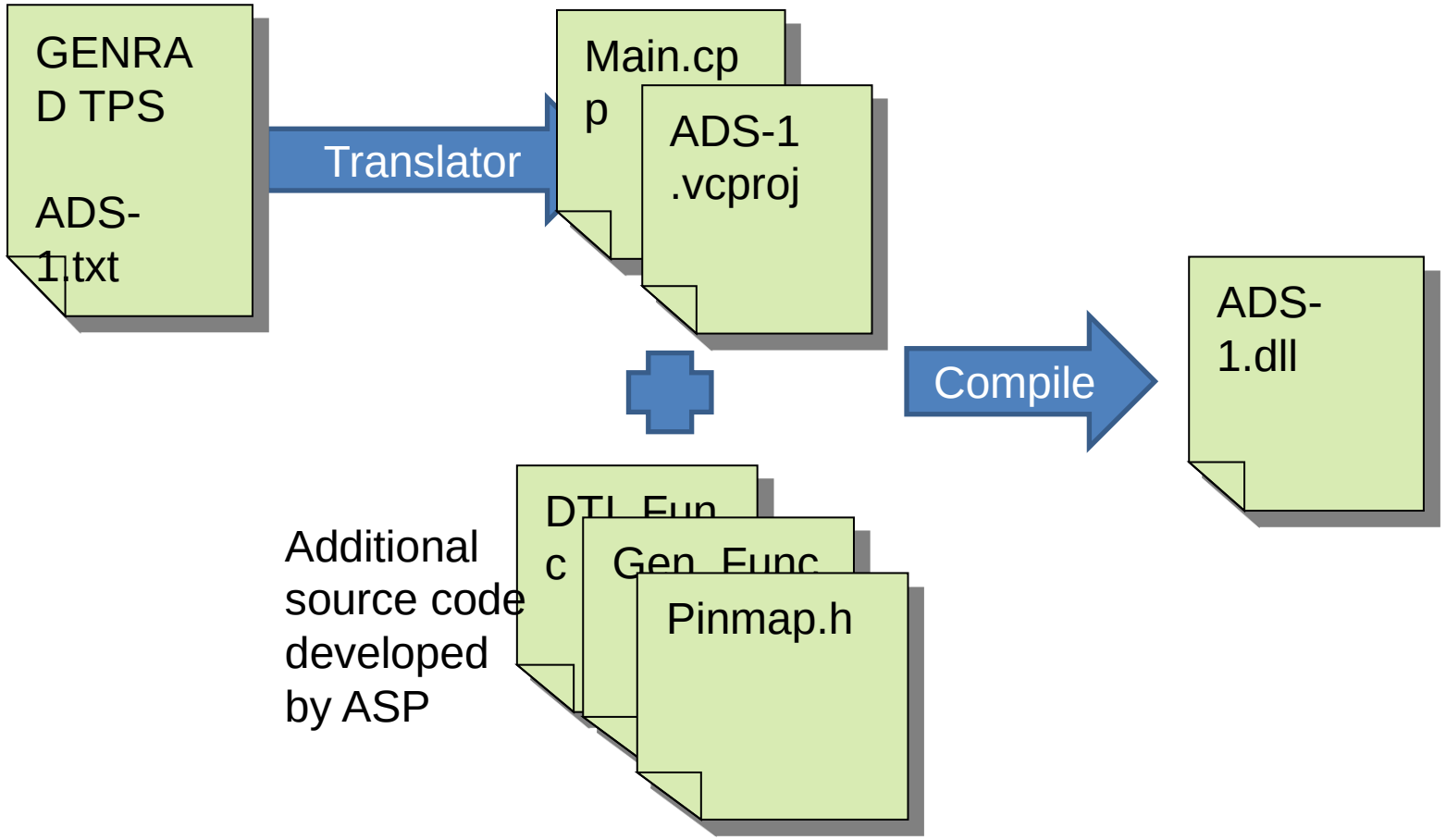
Tester Architecture

BASIC Programming language **C++**

- No time and resources to do-it-all-over; certification will take years
- No tools such as TPS Converter Studio for L200 to provide traceability
- 48 TPS, most digital
- Share one ITA

GENRAD TO S9

- Translator TPS source to Main.CPP
- Additional source files added to be compiled to test DLL



WHAT DOES THE TRANSLATOR DO?

Original TPS

10050 IH 29,30,22,34,5; X

Translator

Main.cpp

```
//10050 IH 29,30,22,34,5; X
SetPinState(PIN_IH,PIN_29);
SetPinState(PIN_IH,PIN_30);
SetPinState(PIN_IH,PIN_22);
SetPinState(PIN_IH,PIN_34);
SetPinState(PIN_IH,PIN_5);
End_Of_Pattern();
XFunction(__LINE__);
```

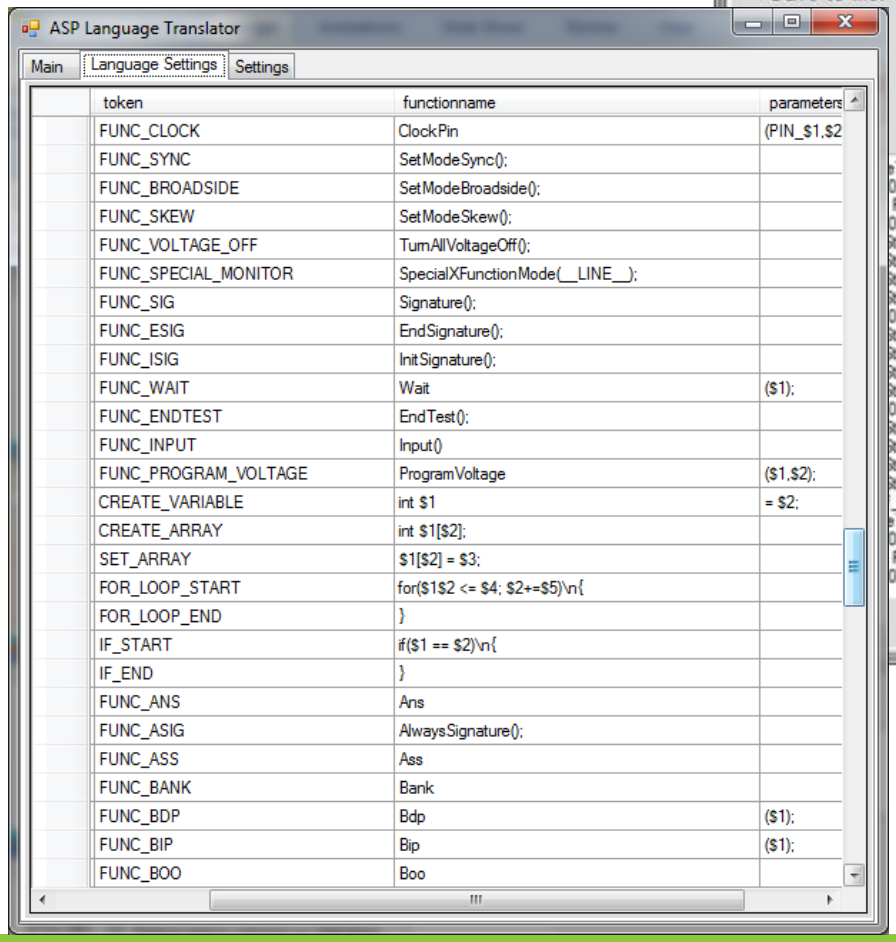
Source code level traceability

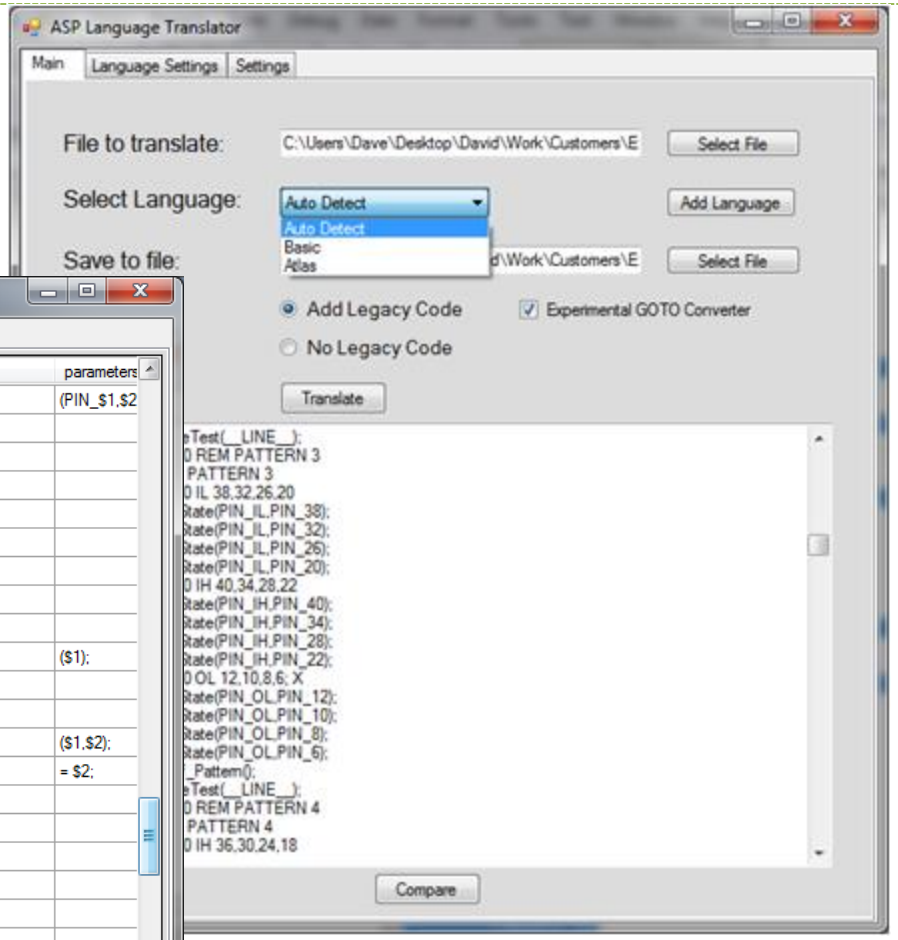
HOW DID WE IMPLEMENT THE TRANSLATOR?

- **Use GNU Tools**
- **Flex**
  - A fast lexical analyzer generator
  - It is a GNU tool for generating programs that perform pattern-matching on text
  - http://flex.sourceforge.net/
- **Bison**
  - General-purpose parser generator
  - http://www.gnu.org/software/bison/

TERADYNE TIM 2017

WHAT DOES IT LOOK LIKE?





TERADYNE TIM 2017

CASE STUDY – TRANSLATOR TO TEST STUDIO PROJECT

PROJECT BACKGROUND

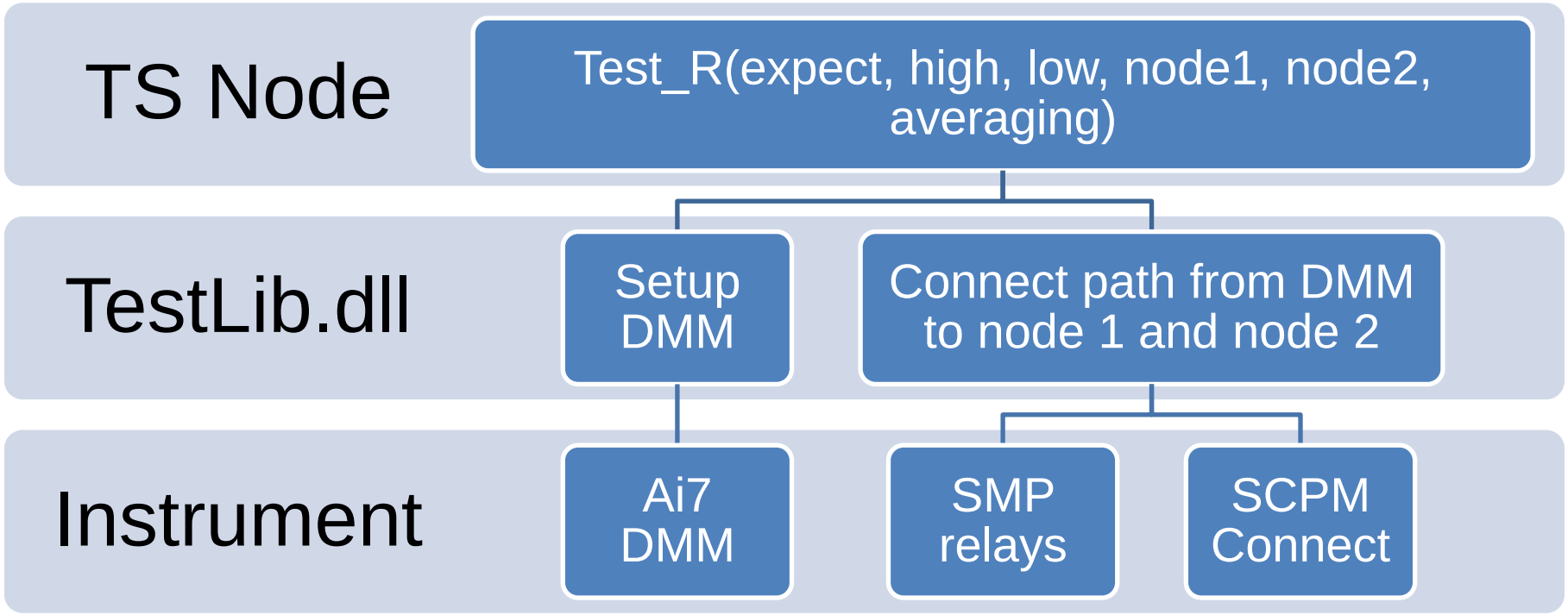
- Number of test program – 147
- Number of test adapters – 13
- Analog tests and simple digital tests only
- Duration of project – 3 years
- Location of project – oversea
- Resource to debug program – inexperienced with S9
- Customer requirement
  - Same test result as previous tester
- Traceability
  - Need to compare with legacy test result line by line

THE APPROACH

Let's make an In-circuit Tester !

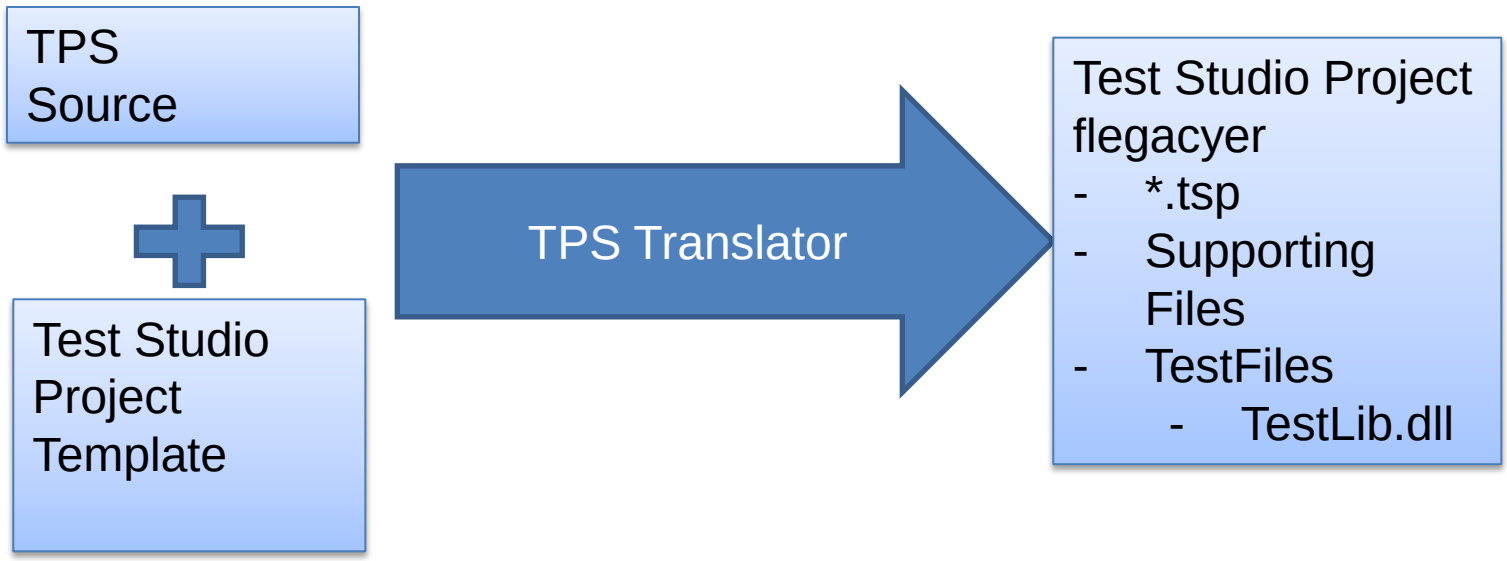
MAKE TEST LIBRARY

- Develop a DLL that export all the instructions required by the TPS
- The test library will be included in each Test Studio project



TRANSLATE THE TPS

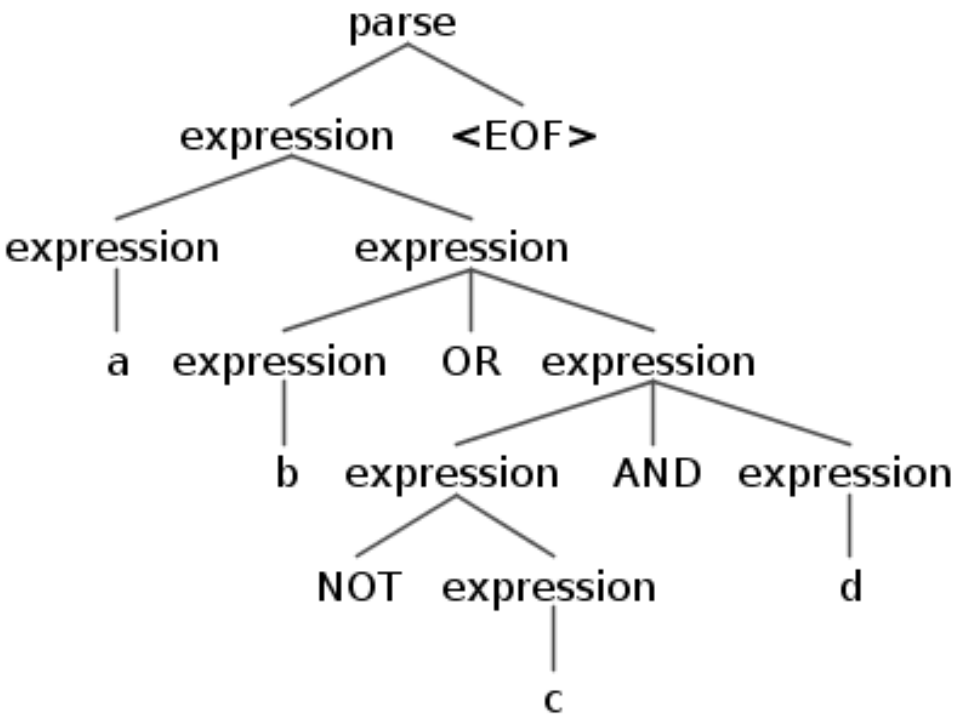
- Use TPS Translator to read the TPS source file, and build a Test Studio Project tree.
- No code compile required



TPS TRANSLATING WITH ANTLR

ANTLR

- ANTLR (ANother Tool for Language Recognition) is a powerful parser generator for reading, processing, executing, or translating text files.



TERADYNE TIM 2017

GRAMMAR

```
grammar Grammar;

/*
 * Parser Rules
 */
start                           : title bootstrap (segment endProgram)+ ;

// Title Sequence ---------------------------------------------------------------
title                           :       break? (titleStat+)? (mountInstruct NL* dismountInstruct NL*)? break?;
titleStat                       :       partNoStat                              #PartNo
                                |       BOOTSTRAP PRN deci NL* #Title_Bootstrap
                                |       TESTPACKAGE testPackage NL* #Title_TestPackage
                                |       ITA PARTNO partNumber NL* #Title_ITA
                                |       titleComment NL* #Title_Comment
                                ;
testPackage                     :       value;
allPartNumbers          :       partNumber '&' allPartNumbers
                                |       partNumber
                                |       partialNumber;
partNumber                      :       value ('-' value)*;
partialNumber           :       '-' value;
titleComment            :       (value | '-' | '.'| months)+;
months                          :       'JAN' | 'FEB' | 'MAR' | 'APR' | 'MAY' | 'JUN' | 'JUL' | 'AUG' | 'SEP' | 'OCT' | 'NOV' | 'DEC';
partNoStat                      :       PARTNO allPartNumbers NL* 'SEGMENT' deci '(' PRN deci ')' NL* break?    #Segment_PartNo
                                |       PARTNO allPartNumbers NL* #Title_PartNo;
mountInstruct           :       MOUNTUUT NL* instr1 NL* (instr2 NL*)? (instr3 NL*)?;
dismountInstruct        :       DISMOUNTUUT NL* instr1 NL* (instr2 NL*)? (instr3 NL*)?;
instr1                          :       'A.' STRING;
instr2                          :       'B.' STRING;
instr3                          :       'C.' STRING;

// BootStrap Sequence -----------------------------------------------------------
bootstrap                       :       startboot instr_block? NL* optionalBoot? instr_block? NL* endboot break?;
startboot                       :       BOOTSTRAP 'SEGMENT' '(' PRN deci ')'NL* ;
endboot                         :       EXITBOOT NL*;
optionalBoot            :       (bootStatements+);
bootStatements          :       printStat NL*   #Boot_Print
                                |       PAUSE NL*               #Boot_Pause
                                ;

// Segment Sequence -------------------------------------------------------------
segment                         :       segmentStart (testSection+) shutdown? segmentEnd;
segmentStart            :       partNoStat break?;
```

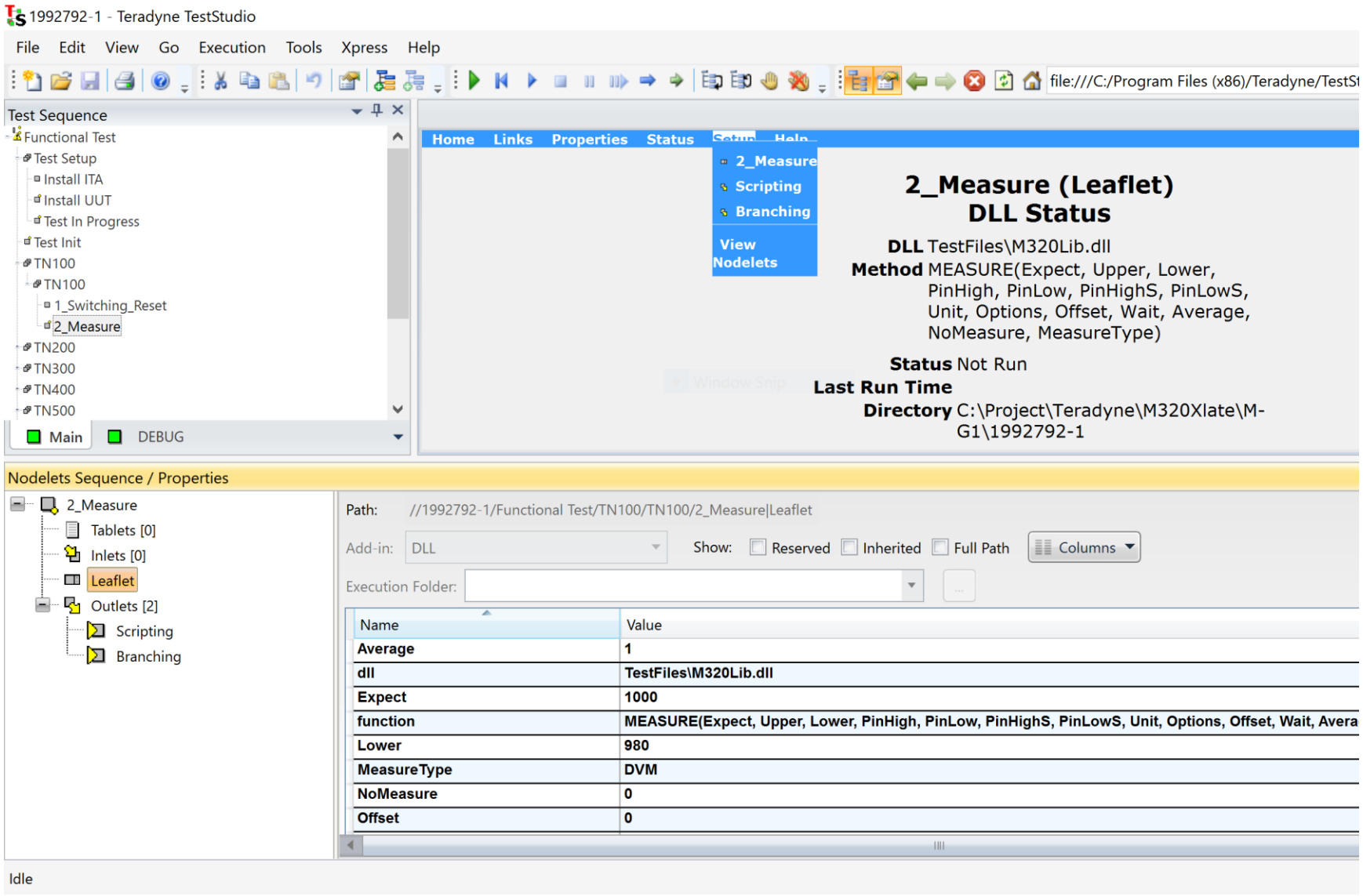
EXAMPLE – ORIGINAL TPS

```
TN100
[TEST PACKAGE ID RESISTOR TEST]
MEASURE 1000+-20 OHMS AT CH 28 UL = 1.02E03 LL = 9.8E02
IF NOGO THEN STOP
--->
    PRINT "            TN 000100"
    PRINT "ENSURE THAT TEST PACKAGE IS M-G1"
--->
```

EXAMPLE – TRANSLATED TS NODE

```
Leaf "2_Measure"
 {
        "TestNumber"="TN100";
        Leaflet "SimpleDLL.Leaflet"
        {
            "_PassByProperty_Run" = "True";
            "dll" = "TestFiles\M320Lib.dll";
            "function" = "MEASURE(Expect, Upper, Lower, PinHigh, PinLow, PinHighS, PinLowS, Unit, Options, Offset, Wait, Average, NoMeasure, MeasureType)";
            "Unit" = "OHM";
            "Expect" = "1000";
            "Upper" = "1020";
            "Lower" = "980";
            "Offset" = "0";
            "Wait" = "0";
            "Average" = "1";
            "NoMeasure" = "0";
            "PinHigh" = "PIN101";
            "PinLow" = "PIN102";
            "PinHighS" = "";
            "PinLowS" = "";
            "MeasureType" = "DVM";
        }
        Outlet Always "Scripting" "Script.Leaflet"
        {
            "FailHighMessage" = "          TN 000100<br>ENSURE THAT TEST PACKAGE IS M-G1<br>";
            "FailLowMessage" = "          TN 000100<br>ENSURE THAT TEST PACKAGE IS M-G1<br>";
            "PauseAfterFailHighMessage" = "0";
            "PauseAfterFailLowMessage" = "0";
            "PauseAfterPassMessage" = "0";
            "scripttype" = "java";
            "source" = "file";
            "file" = "supportfiles\printtoconsole.js";
            }
        Outlet Always "Branching" "Script.Leaflet"
        {
                "Condition" = "(IsFail(""//1992792-1/Functional Test/TN100/TN100/2_Measure""))";
                "TargetPathStatus" = "//1992792-1/Functional Test/Test End";
                "TargetPathTrue" = "//1992792-1/Functional Test/Test End";
            }
    }
```

EXAMPLE – TS NODE



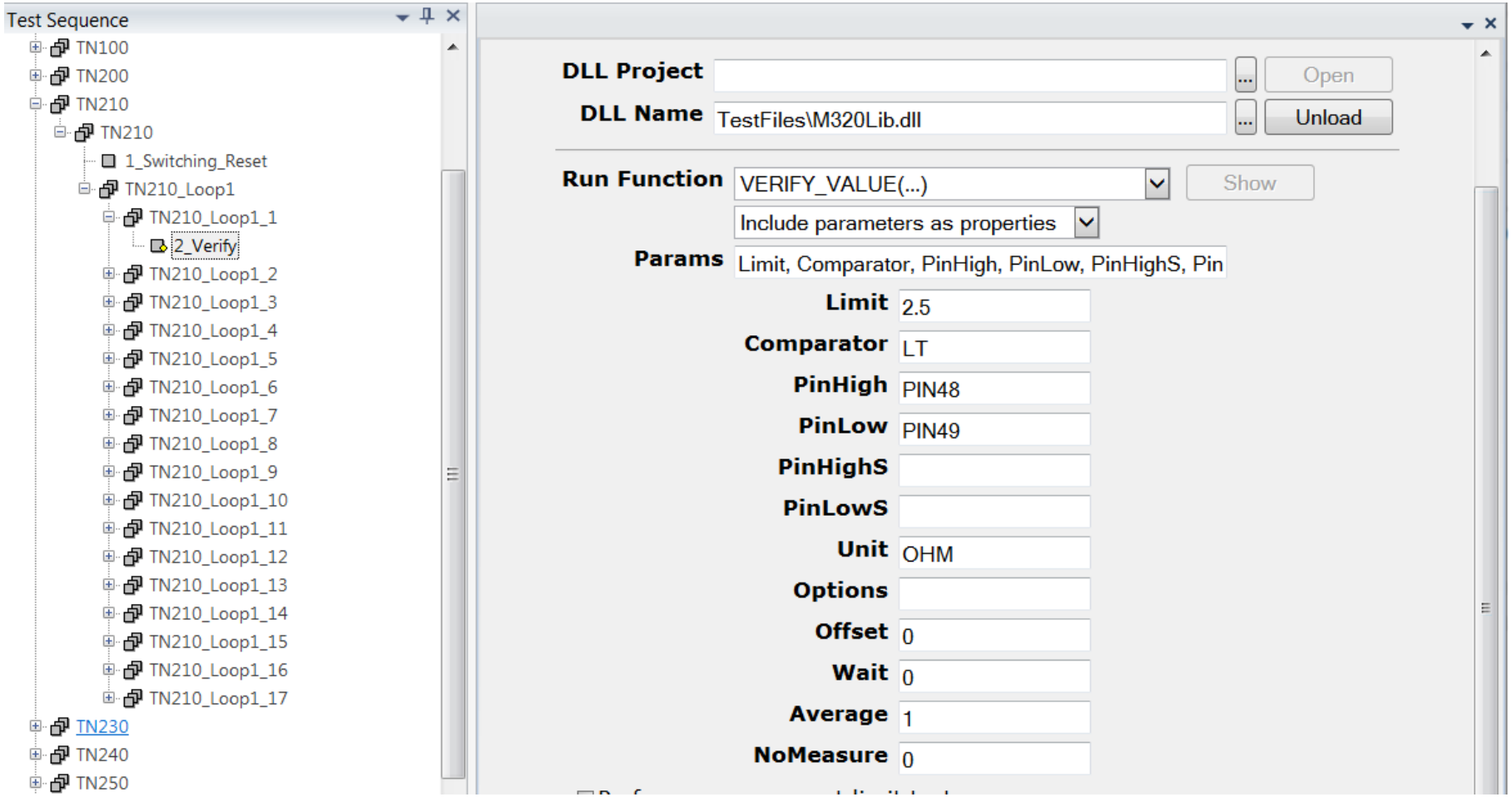
WHAT ABOUT FOR LOOP WITH VARIABLES

- FOR LOOP is used to read DATA and execute same tests at different test points
- Translator expand the for loop (i.e. flatten) into individual tests with the variables replaced by actual value

EXAMPLE – FOR LOOP

```
PN 1997031-1-2.txt

TN210

[CONTINUITY TEST OF P1 JUMPERS]


FOR K = 1 TO 17

READ A,D

VERIFY VALUE LT 2.5 OHMS AT PINS A,D

--->

    PRINT "            TN 000210"

    PRINT "CHECK FOR OPEN BETWEEN PINS ";A;;" AND ";D;;""

    PRINT "REPAIR DEFECTIVE CONNECTOR OR ASSOCIATED PC WIRING"

    PRINT ""

    PRINT "DEPRESS 'F5' TO CONTINUE"

    PAUSE

<---

NEXT K

DATA 48,49,48,50,48,57,48,51,42,12

DATA 42, 4,42,34,42, 7,42,37,38, 8

DATA 35, 5,36, 6,33, 3,31, 1,39, 9

DATA 41,11,40,10
```

EXAMPLE – FOR LOOP



MULTIPLE ITA

- Since there are multiple ITA involved, need to be able to handle different wire lists
- Use Excel file as input
- Parse into vectors to be used by the Test Library

TERADYNE TIM 2017

EXCEL PARSER FOR UUT PINMAP

Parse this to generate DI Pinmap

3	L/N	(CH)	DI UNIT	DI NAME	Di Channel pin numbers			Di Channel pin numbers				UUT CH
4	1	0	Di-025-12	UNIT1-0H	CH1	A1-P22	E2	CH0	A1-P22	D1		CH1
5	2		64CH	UNIT1-0L				GND	A1-P22	D2		
6	3	2		UNIT1-2H	CH3	A1-P22	E4	CH2	A1-P22	D3		CH2
7	4			UNIT1-2L				GND	A1-P22	D4		
8	5	4		UNIT1-4H	CH5	A1-P22	E6	CH4	A1-P22	D5		CH3
9	6			UNIT1-4L				GND	A1-P22	D6		
10	7	6		UNIT1-6H	CH7	A1-P22	E8	CH6	A1-P22	D7		CH4
11	8			UNIT1-6L				GND	A1-P22	D8		
12	9	8		UNIT1-8H	CH9	A1-P22	E10	CH8	A1-P22	D9		CH5
13	10			UNIT1-8L				GND	A1-P22	D10		
14	11	10		UNIT1-10H	CH11	A1-P22	E12	CH10	A1-P22	D11		CH6
15	12			UNIT1-10L				GND	A1-P22	D12		
16	13	12		UNIT1-12H	CH13	A1-P22	E14	CH12	A1-P22	D13		CH7
17	14			UNIT1-12L				GND	A1-P22	D14		
18	15	14		UNIT1-14H	CH15	A1-P22	E16	CH14	A1-P22	D15		CH8
19	16			UNIT1-14L				GND	A1-P22	D16		
20	17	16		UNIT1-16H	CH17	A1-P22	E18	CH16	A1-P22	D17		CH9
21	18			UNIT1-16L				GND	A1-P22	D18		
22	19	18		UNIT1-18H	CH19	A1-P22	E20	CH18	A1-P22	D19		CH10
23	20			UNIT1-18L				GND	A1-P22	D20		
24	21	20		UNIT1-20H	CH21	A1-P22	E22	CH20	A1-P22	D21		CH11
25	22			UNIT1-20L				GND	A1-P22	D22		
26	23	22		UNIT1-22H	CH23	A1-P22	E24	CH22	A1-P22	D23		CH12
27	24			UNIT1-22L				GND	A1-P22	D24		
28	25	24		UNIT1-24H	CH25	A1-P22	E26	CH24	A1-P22	D25		CH13
29	26			UNIT1-24L				GND	A1-P22	D26		

UUT CH Assignment | M-G1 ITA Wire Run List | Ai760_3172 Wirelist

Ready

TERADYNE TIM 2017

EXCEL PARSER FOR ITA WIRE LIST

Parse this table to get connection path from one resource to another

MG1 ITA Route Path Table

		VPC Connector					SMP5001 Relay Module VPC Connector					SMP7600 VPC Connector					SMP5001 Relay Module VPC Connector					
		Signal Name					SMP5001 Relay Module VPC Connector					SMP7600 VPC Connector					SMP5001 Relay Module VPC Connector					Signal Name
		NETNAME1	Slot #	Pin #	NET	Pin #	Slot #	Pin #	RELAYA	RELAYB	Pin #	Slot #	Pin #	LOAD-IN	LOAD-OUT	Pin #	Slot #	Pin #	RELAYAL	RELAYBL	Pin #	NETNAME2
490		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	H37	3CH28-A	3CH28-B	H38	$MCH047
491		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	J37	3CH29-A	3CH29-B	J38	$MCH048
492		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	K37	3CH30-A	3CH30-B	K38	$MCH052
493		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	A39	3CH31-A	3CH31-B	A40	$MCH053
494		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	B39	3CH32-A	3CH32-B	B40	$MCH054
495		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	C39	3CH33-A	3CH33-B	C40	$MCH056
496		$GND(PC1)					A1-P6	F23	2CH36-A	2CH36-B	F24	A1-P29	A1	$LOAD-01-IN	$LOAD-01-RTN	B1	A1-P6	D39	3CH34-A	3CH34-B	D40	$MCH063
497																						
498		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	E39	3CH35-A	3CH35-B	E40	$MCH006
499		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	F39	3CH36-A	3CH36-B	F40	$MCH012
500		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	G39	3CH37-A	3CH37-B	G40	$MCH014
501		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	H39	3CH38-A	3CH38-B	H40	$MCH016
502		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	J39	3CH39-A	3CH39-B	J40	$MCH017
503		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	K39	3CH40-A	3CH40-B	K40	$MCH018
504		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	A41	3CH41-A	3CH41-B	A42	$MCH019
505		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	B41	3CH42-A	3CH42-B	B42	$MCH027
506		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	C41	3CH43-A	3CH43-B	C42	$MCH039
507		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	D41	3CH44-A	3CH44-B	D42	$MCH040
508		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	E41	3CH45-A	3CH45-B	E42	$MCH049
509		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	F41	3CH46-A	3CH46-B	F42	$MCH055
510		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	G41	3CH47-A	3CH47-B	G42	$MCH058
511		$GND(PC1)					A1-P6	G23	2CH37-A	2CH37-B	G24	A1-P29	A9	$LOAD-02-IN	$LOAD-02-RTN	B9	A1-P6	H41	3CH48-A	3CH48-B	H42	$MCH059
512																						
513		$GND(PC1)					A1-P6	H23	2CH38-A	2CH38-B	H24	A1-P29	A17	$LOAD-03-IN	$LOAD-03-RTN	B17	A1-P6	J41	3CH49-A	3CH49-B	J42	$MCH024
514		$GND(PC1)					A1-P6	H23	2CH38-A	2CH38-B	H24	A1-P29	A17	$LOAD-03-IN	$LOAD-03-RTN	B17	A1-P6	K41	3CH50-A	3CH50-B	K42	$MCH050
515																						
516		$GND(PC1)					A1-P6	J23	2CH39-A	2CH39-B	J24	A1-P29	A25	$LOAD-04-IN	$LOAD-04-RTN	B25	A1-P6	A43	3CH51-A	3CH51-B	A44	$MCH021
517		$GND(PC1)					A1-P6	J23	2CH39-A	2CH39-B	J24	A1-P29	A25	$LOAD-04-IN	$LOAD-04-RTN	B25	A1-P6	B43	3CH52-A	3CH52-B	B44	$MCH057
518																						

UUT CH Assignment | **M-G1 ITA Wire Run List** | Ai760_3172 Wirelist

Ready


Q
&
A