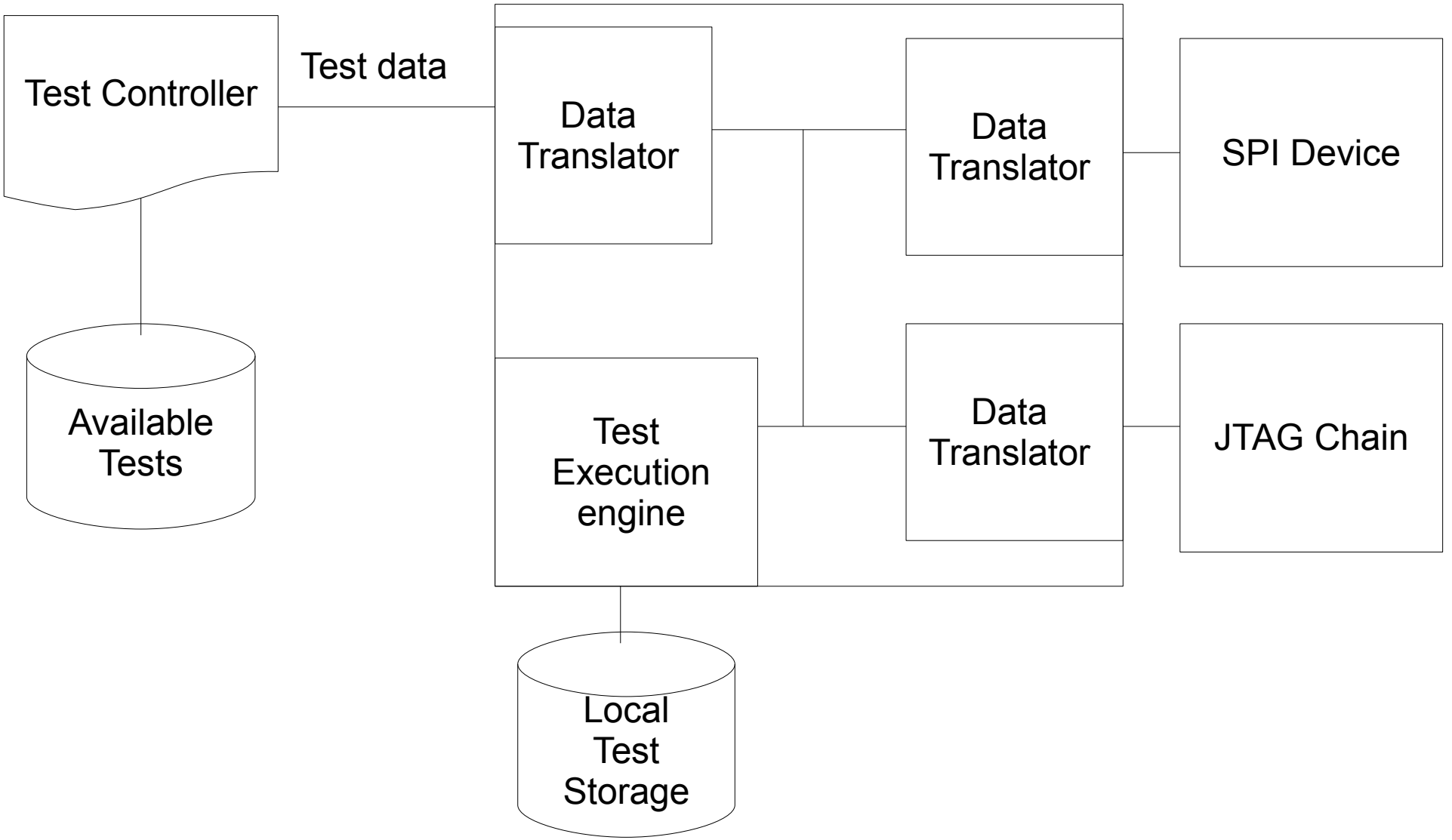
Demonstration of P2654 transformation concepts using the UTAP solution

UTAP P2654 Demonstration

- To provide a method to enable the delivery of test actions across multiple different interfaces within a system environment
- System can be a single card or multiple sub units connected via various interfaces
- Destination test vector bus sources from different input sources

UTAP Simplified Overview



Simplified Overview Contd

- Test controller can be external or embeded within the system. The controller may communicate over any avaliable iterface to the UTAP (RS232, SPI, JTAG, GPIO, I2C etc)
- Test Data : Test vector data being presented to the UTAP on one of its avaliable interfaces
- A SPI flash device is used to provide a sample SPI target
- Two devices are connected in JTAG chain to enable JTAG infra and inter testing

Test Data format

- The UTAP test data is the destination bus test data encapsulated with information required to support each transformation logic block
- The UTAP test data generation tools are supplied as Windows executables
- The UTAP test data can be generated on the fly or offline

UTAP Hardware

- The UTAP configuration is built from a family of IP blocks. It is fully customisable to supply the requirements of the end solution
- The UTAP can provide a standalone test system or be integrated along with the GD112 or GLD2 to provide system BIST
- The UTAP IP blocks are fully portable across different target platforms either FPGA's or ASIC

UTAP Demonstration

- The test actions for two target buses will be delivered from different bus interface compared to the target bus
- Provides vendor neutral platform for vector deliver and diagnosis
- Test actions may be executed dynamically or stored within the UTAP system
- Diagnostic information maybe stored locally or retrieved in real time during execution

UTAP Demonstration

- Storage of test action and result data can be accessed via multiple interfaces using native tools along with third party access. Third party JTAG access is performed using short chain IP using standard flash actions to write and read memory
- Full diagnostics can be performed with UTAP system or exported to the vector generation tool

UTAP Demonstration

- JTAG Target
  - Input format is the industry standard method of describing vectors at the IEEE1149.x interface STAPL or SVF files
  - All scan chain setup information independent of scan chain control method (Gateway, Linker, SIB) to be present in STAPL/SVF files

UTAP Demostration

- JTAG vector delivery using bus translation
  - From SPI bus to JTAG bus
  - From RS232 to JTAG bus
- For both executions its possible to achieve results in real time or store within the UTAP unit

UTAP Demonstration

- Third party JTAG tests converted into UTAP format (section information can also be added) and loaded into the UTAP or executed directly

```
Begin generating the BTF files
(test\Bravo_All_tests.btf).....

Writing SVF Test section : 0
Writing SVF Test section : 1
Writing SVF Test section : 2
Writing SVF Test section : 3
The file 'test\Bravo_All_tests.btf' opened
The file 'test\Bravo_All_tests.btf1' opened
The file 'test\Bravo_All_tests.btf2' opened
+=======+
| PASS! |
+=======+

Press any key to continue . . .
```

UTAP Demonstration

Diagnostic Sample Output Data

- Default report format HTML- other available on request
- Diaganostic report summary

Program Version	1.0.1
Program Executed	Tue Jan 28 02:13:52 PM
Number of Errors Detected	808
Device Summary's	Show Device Data
Vector Summary	Show Vector Summary Data
Pin Error List.	Show Pin Error Data

UTAP Demonstration

- Sample Vector by Vector Decode Summary

TAP STATE	IRSHIFT
TAP STATE	IREXIT1
TAP STATE	IRPAUSE
TAP STATE	IREXIT2
IRSHIFT Number	2
BITS in Error	2
I/O SSO High->Low	0
I/O SSO Low->High	0
Shift Summary	Show Shift Data
TAP STATE	IRUPDATE
TAP STATE	DRSELECT
TAP STATE	DRCAPTURE
TAP STATE	DRSHIFT
TAP STATE	DREXIT1
TAP STATE	DRPAUSE
TAP STATE	DREXIT2
DRSHIFT Number	2
BITS in Error	1
I/O SSO High->Low	0
I/O SSO Low->High	0

UTAP Demonstration

- Shift operation devices and registers summary shown

PCB Ref.	U27
IR Value	11111111 (BYPASS)
BITS in Error	0
I/O SSO High->Low	0
I/O SSO Low->High	0
Full Device Report	Show Device Data
PCB Ref.	U28
IR Value	00000010
BITS in Error	1
I/O SSO High->Low	0
I/O SSO Low->High	0
Full Device Report	Show Device Data

UTAP Demonstration

- Data decoded for the register selected by the active shift operation

Boundary Scan Register data : Device U28 : ShiftDR Operation 2

Cell No.	PKG Loc.	Function	Cell Type	Expected	Received	Mask
0	S1	BIDIR	BC_7	0	0	0
1	-	CONTROL	BC_2	0	0	0
2	S2	BIDIR	BC_7	0	0	0
3	-	CONTROL	BC_2	0	0	0
4	S3	BIDIR	BC_7	0	0	0
5	-	CONTROL	BC_2	0	0	0
6	S4	BIDIR	BC_7	0	0	0
7	-	CONTROL	BC_2	0	0	0
8	S5	BIDIR	BC_7	0	0	0
9	-	CONTROL	BC_2	0	0	0
10	S6	BIDIR	BC_7	0	0	0
11	-	CONTROL	BC_2	0	0	0
12	S7	BIDIR	BC_7	0	0	0
13	-	CONTROL	BC_2	0	0	0
14	S8	BIDIR	BC_7	0	0	0
15	-	CONTROL	BC_2	0	0	0
16	S9	BIDIR	BC_7	0	0	0
17	-	CONTROL	BC_2	0	0	0
18	S10	BIDIR	BC_7	0	0	0

UTAP Demonstration

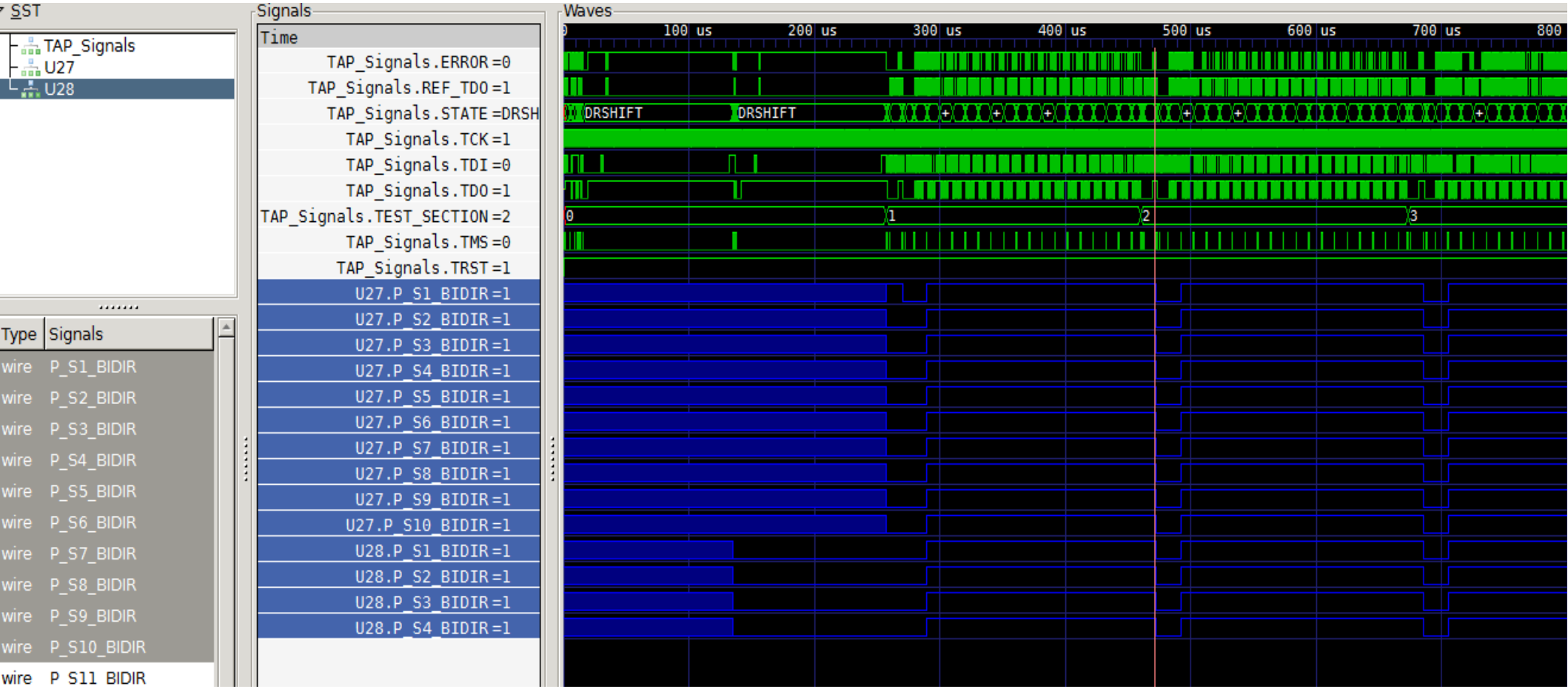
- Report all device pins in error at the completion of test execution

Device Pins With Compare Errors

Comp. ID.	Cell No.	PKG Loc.	Function	Error Cnt.
U27	0	S1	BIDIR	26
U27	2	S2	BIDIR	25
U27	4	S3	BIDIR	25
U27	6	S4	BIDIR	25
U27	8	S5	BIDIR	25
U27	10	S6	BIDIR	25
U27	12	S7	BIDIR	25
U27	14	S8	BIDIR	25
U27	16	S9	BIDIR	25
U27	18	S10	BIDIR	25
U27	20	S11	BIDIR	25
U27	22	S12	BIDIR	25
U27	24	S13	BIDIR	25
U27	26	S14	BIDIR	25
U27	28	S15	BIDIR	25
U27	30	S16	BIDIR	25
U28	0	S1	BIDIR	25
U28	2	S2	BIDIR	25
U28	4	S3	BIDIR	25

UTAP Demonstration

- Waveforms generated showing signals in error

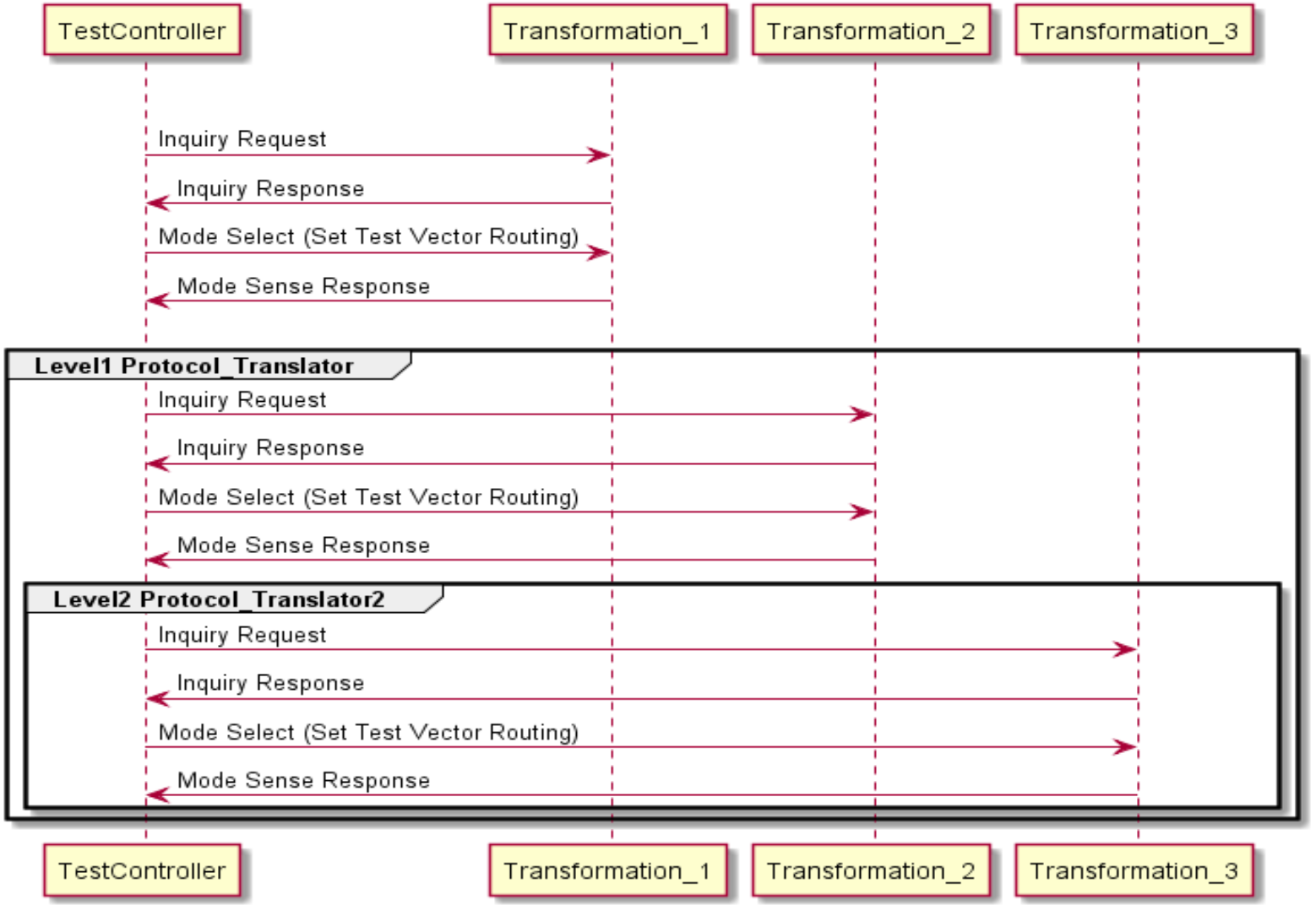


UTAP Demostration

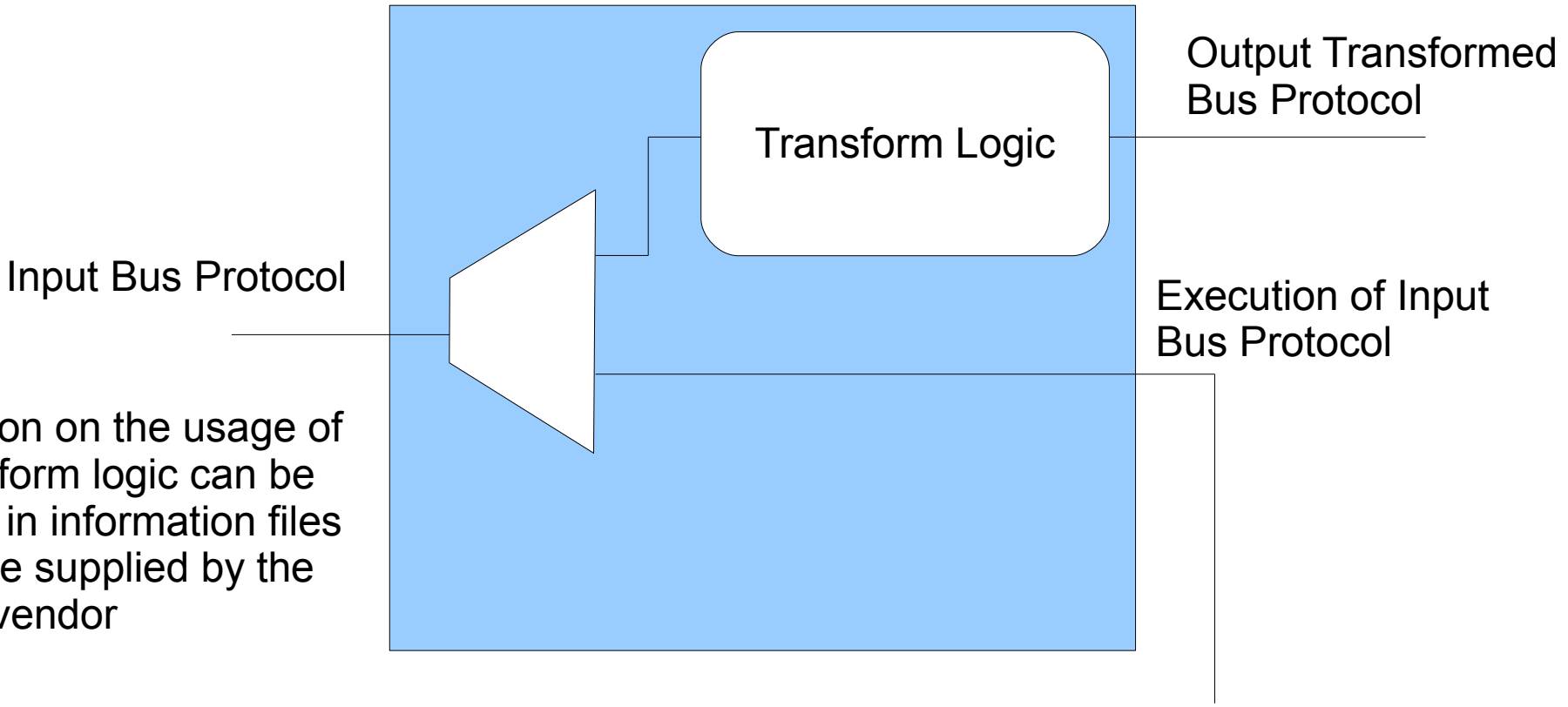
- SPI Target
  - Binary file for targeting SPI flash devices
  - Script of commands and responses to/from SPI device
  - Transformations added to the SPI data allowing access from other bus interfaces

Vendor Portable Transformation Scheme

Test Data routing

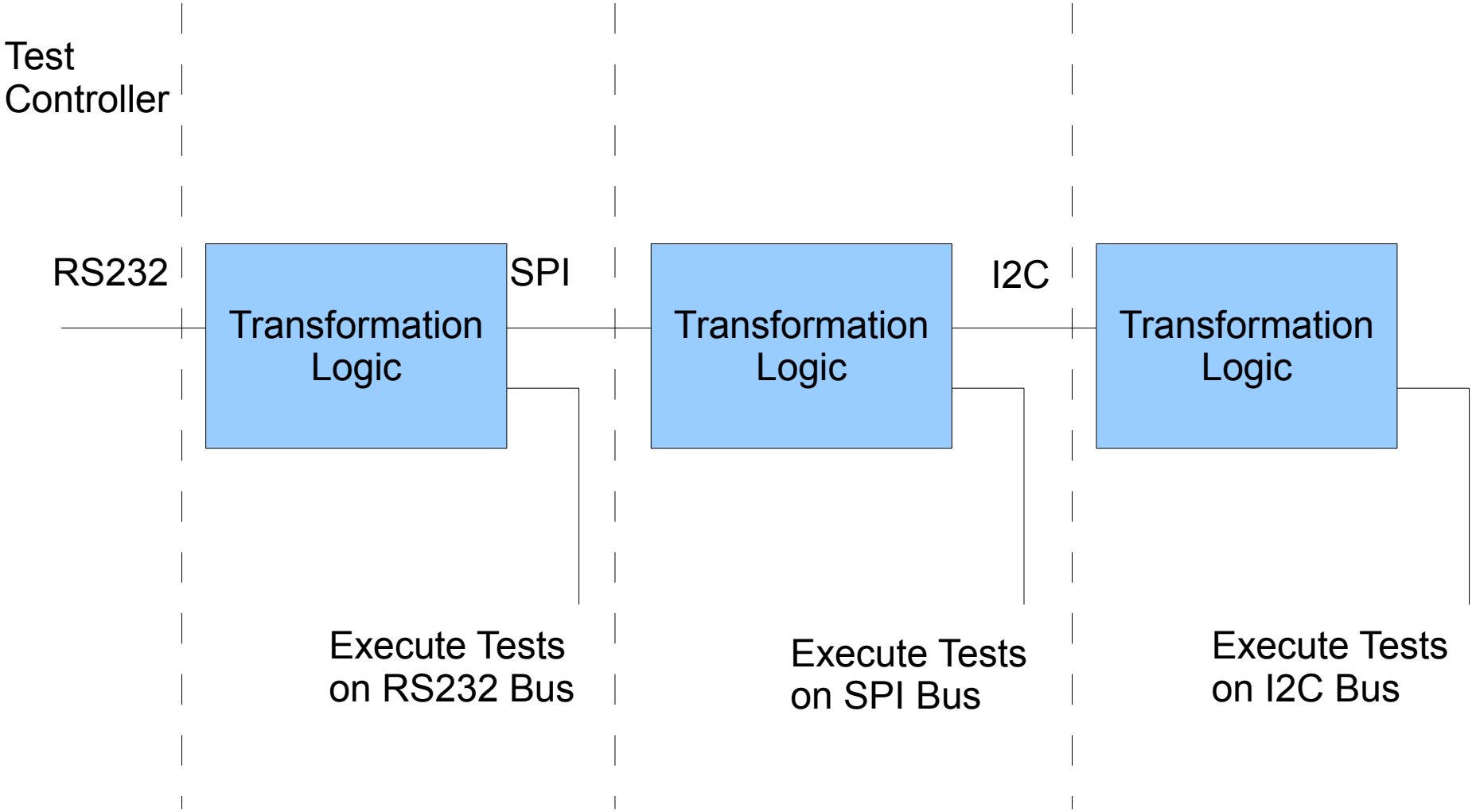


Transformation Logic



Information on the usage of the transform logic can be provided in information files (.inf) to be supplied by the solution vendor

Transformation Logic Contd



Test data encapsulation

- Test data encapsulation can be controlled using standard SCSI type block headers of a fixed number of bytes
- Inquiry command allows the test controller to identify the type of transformation logic present
- Data base of each encapsulation block can be maintained by the test controller
- Encapsulation tools/plugins or information can be supplied in multiple formats such as DLL or .inf files showing syntax for the required encapsulation function

Test data encapsulation

- Mode Select command allows the route the test data for delivery at current level of test hirearchy or to be delivered to the next bus protocol
- Mode Sense command allows the transformation logic setup to be indentified

Inquiry Command

Byte	Bit							
	7	6	5	4	3	2	1	0
0	Operation Code (12h)							
1	Logical Unit Number			Reserved (00h)			CmdDt (0)	EVPD
2	Page Code							
3	Reserved (00h)							
4	Allocation Length							
5	Control Byte (00h)							

Inquiry Response

- The response data for the Inquiry command can be specified to support defined fields such as vendor protocol translation type along with allowing vendor specfic data

Mode Select Command

Byte	Bit							
	7	6	5	4	3	2	1	0
0	Operation Code (15h)							
1	Logical Unit Number			PF (1)	Reserved (0)			SP
2	Reserved (00h)							
3	Reserved (00h)							
4	Parameter List Length							
5	Control Byte (00h)							

Mode Select Parameters

- The mode select parameter bytes can contain default fields such as routing of the test data for delivery at that level or be translated to the next protocol
- Additional data bytes in the mode select data fields may be used for vendor specfic functions

Mode Sense Command

Byte	Bit							
	7	6	5	4	3	2	1	0
0	Operation Code (1Ah)							
1	Logical Unit Number			Reserved (0)	DBD (1)	Reserved (0)		
2	Page Control		Page Code					
3	Reserved (00h)							
4	Allocation Length							
5	Control Byte (00h)							

Mode Sense Response data

- The mode sense response data bytes can contain default fields such as routing of the test data for delivery at that level or be translated to the next protocol
- Additional data bytes in the mode sense response data fields may be used for vendor specfic functions