Using IEEE 1149.1-2013 (JTAG) with ATEASY®

CJ Clark
Intellitech Corp.
cclark@intellitech.com

Introduction

ATEasy® is a flexible integrated Test Executive development environment for board and system-level test. This white paper provides an overview of how ATEasy can be used to support JTAG/IEEE 1149.1 based instrument control, programming and test. The reader may be somewhat familiar with boundary-scan test using the 2001 version of the IEEE® 1149.1 standard. This white paper discusses the newer capabilities of boundary-scan defined in IEEE 1149.1-2013[1] for controlling on-chip instruments. Overall test program flow is basically the same for the older boundary-scan interconnect tests or FPGA programming via JTAG as it is for the new on-chip register support. This white paper first provides background information about IEEE 1149.1-2013 and then discusses how the user can integrate ATEasy with boundary-scan tools to read temperature and on-board voltages, without a DMM, using just an on-board Xilinx® FPGA. The reader is encouraged to download the free NEBULA™ software which includes the examples depicted in this white paper.[2,3]

IEEE 1149.1-2013 Silicon Instruments

IEEE 1149.1-2013 added substantially to the original IEEE 1149.1-2001 standard for test structures inside the IC. It includes definitions of hierarchically-accessible Intellectual Property (IP) blocks, standalone descriptions of IP block functions, and support for a standardized procedural language which allows communication to all things in the silicon via the Test Access Port (TAP).

Intellitech calls these IP blocks "Silicon Instruments™," as they perform functions just like external instrumentation, but they're not limited by the need for physical access to the I/O of the integrated circuit, nor do they impact the measurement like traditional probing.

Figure 1 shows an example of the types of Silicon Instruments which can be useful in modern System on a Chip (SoC) designs. Serializer/Deserializer (SerDes) BIST via an algorithmic Bit-Error-Rate test is a common on-chip function available from the TAP. The SerDes associated functions and their control such as differential swing, AC/DC coupling, and pre-emphasis can be accessed via the TAP interface during system verification and debug. This capability is particularly useful during the early stages of development when the mission mode firmware is not ready and the design engineer requires a non-intrusive method of evaluating the system/device without adding unwieldy variables to the debug analysis - such as mission mode firmware bugs. Other instruments such as Process-Voltage-Temperature (PVT) monitors are accessible in the SoC via the TAP, providing the means to monitor the device in-situ.

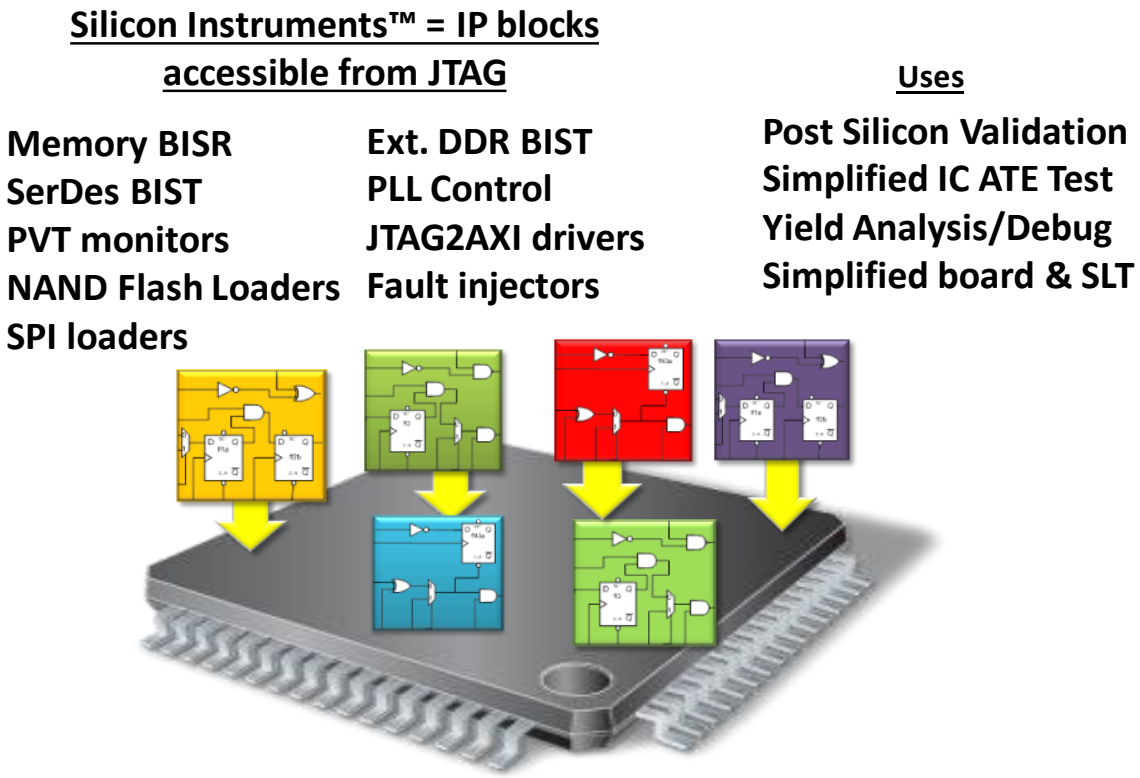

Figure 1. Silicon Instruments for SoCs

The concept used in IEEE 1149.1-2013 provides for standalone descriptions of TAP-accessible Silicon Instruments, allowing them to be pre-verified and then instantiated in the SoC hierarchy.

Readers using FPGAs may already be unknowingly using a JTAG accessible instrument. Whenever external SPI and NOR Flash are programmed through an FPGA JTAG tool, an instrument design loaded into the FPGA facilitates writing and reading of data to the attached non-volatile memory. This capability may be one of the most widely used Silicon Instruments in the industry and three major FPGA vendors license the technology from Intellitech's US 6,594,802 patent. [4]

For this white paper, the integration example uses the Xilinx system monitor instrument.[5,6] The Xilinx system monitor (SYSMON) is an appealing choice, as it is readily available in Xilinx Virtex 5 FPGAs and later generations, including UltraScale™ FPGAs. SYSMON is publicly documented, so the reader can access the Xilinx documentation directly to better understand its functionality.

Figure 2 shows a basic block diagram of the Xilinx SYSMON. SYSMON is essentially a TAP-accessible 10-bit Analog to Digital Converter (ADC) connected to an addressable differential analog mux. An 1149.1-compliant instruction register (IR) and test data register (TDR) is present. The TDR is segmented in the figure into a field for read/write (R/W), a field named DATA for receiving the value from the ADC, and an address field named ADDR which controls the analog mux. The analog mux allows selection of 16 different differential voltages, four FPGA core voltages, status, calibration, alarms and on-chip temperature.

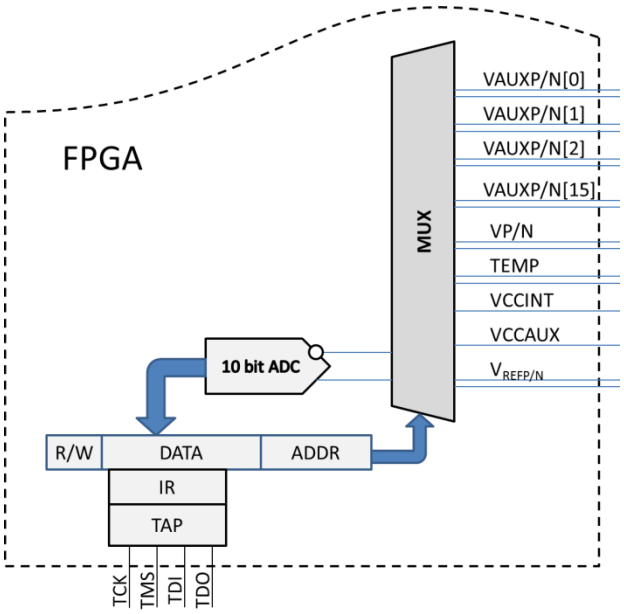

Figure 2. SYSMON Block Diagram

Using the Xilinx SYSMON, users can monitor voltages and current consumption. Figure 3 shows an example of how to use a small-value resistor to measure the current on the 3.3V voltage rail of the DC/DC converter. The differential voltage present at VAUXP[1] and VAUXN[1] can be used to calculate the current through the resistor using Ohm's law. The current, I, is Current(I) = (VAUXP(1) – VAUXN(1)) / .010. Other configurations are possible to measure other PCB currents and voltages without any external instruments.

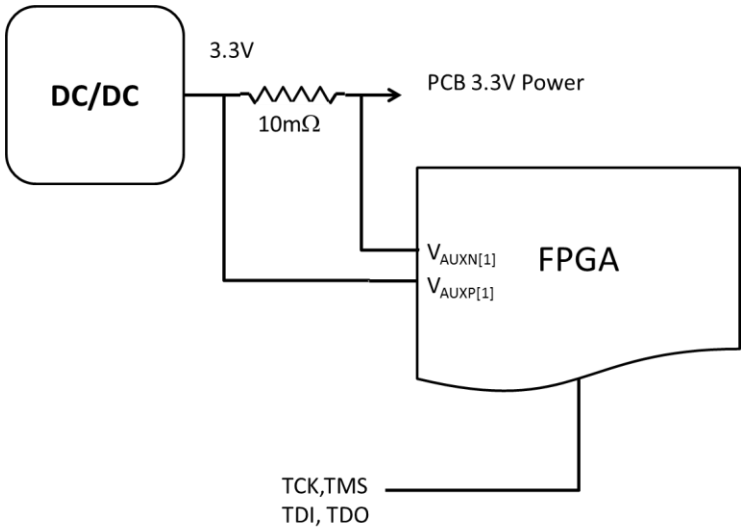

Figure 3. Measuring PCB Current via JTAG

The reader may find the information presented on SYSMON useful but may be unsure how to accomplish all of the work necessary to get SYSMON to take a measurement. One of the benefits of IEEE 1149.1-2013 is that it provides a method of describing instruments separately from the IC. The descriptions and operation of SYSMON can be pre-defined and validated such that the application-specific operation is straightforward, without the test engineer performing extra work to become an expert on each and every instrument. In the case of SYSMON, Intellitech has already done this work for you.

```
Package Sysmon is
  use Std_1149_1_2013.all;
end Sysmon;

package body Sysmon is

  use STD_1149_1_2013.all;

  attribute REGISTER_MNEMONICS of SYSMON : package is
  "CmdVal       (            " &
  "  No_Operation (0b0000), " &
  "  DRP_Read     (0b0001), " &
  "  DRP_Write    (0b0010)  " &
  "             ),           " &
  "AddrValue    ( " &
  "  Temperature       (    0), " &
  "  Vccint            (    1), " &
  "  Vccaux            (    2), " &
  "  Vp_Vn             (    3), " &
  "  Vrefp             (    4), " &
  "  Vrefn             (    5), " &

  "  Supply_Offset     ( 0X08), " &
  "  Vauxp_Vauxn_0     (0B0000010000), " &
```

```
  "  Vauxp_Vauxn_1     (0B0000010001), " &
  "  Vauxp_Vauxn_2     (0B0000010010), " &
  "  Vauxp_Vauxn_3     (0B0000010011), " &
  "  Vauxp_Vauxn_4     (0B0000010100), " &
  "  Vauxp_Vauxn_5     (0B0000010101), " &
  "  Vauxp_Vauxn_6     (0B0000010110), " &
  "  Vauxp_Vauxn_7     (0B0000010111), " &
  "  Vauxp_Vauxn_8     (0B0000011000), " &
…
  "  Max_Temp          (0X20), " &
  "  Max_Vccint        (0X21), " &
  "  Max_Vccaux        (0X22), " &
  "  Min_Temp          (0X24), " &
  "  Min_Vccint        (0X25), " &
  "  Min_Vccaux        (0X26), " &
 …
 "             ) " ;

attribute REGISTER_FIELDS of Sysmon : package is
"MONITOR_DRP_REG[32]  ( " &
"(Unused[2]       is ( 31 downto 30) TAPRESET NOPI
RESETVAL(0B00)), " &
"(DRP_Command[4]  is ( 29 downto 26) TAPRESET
CAPTURES(CmdVal(-)) RESETVAL(CmdVal(No_Operation))), " &
"(DRP_Address[10] is ( 25 downto 16) TAPRESET
CAPTURES(AddrValue(-)) RESETVAL(AddrValue(Temperature)) ), " &
"(DRP_Data[16]    is ( 15 downto  0) TAPRESET CAPTURES(-)
RESETVAL(0X0000) )" &
"                    ) " ;
end Sysmon;
```

Figure 4. Package File code listing for SYSMON

Figure 4 shows the IEEE 1149.1-2013 description of the SYSMON instrument. "Package Files" describe the test data registers in an instrument and describe the mnemonics and other attributes of the instrument. The REGISTER_FIELDS attribute provides a description of the fields detailed in the SYSMON documentation. The three main fields are DRP_Command, DRP_Address and DRP_Data. The package file contains a set of mnemonics which associates human-readable text with specific values needed for the SYSMON instrument. The mnemonics enhance the user experience with an interactive GUI such as that available in Intellitech's NEBULA software as shown in Figure 5. Their use for writing procedural scripts is described in the next section.

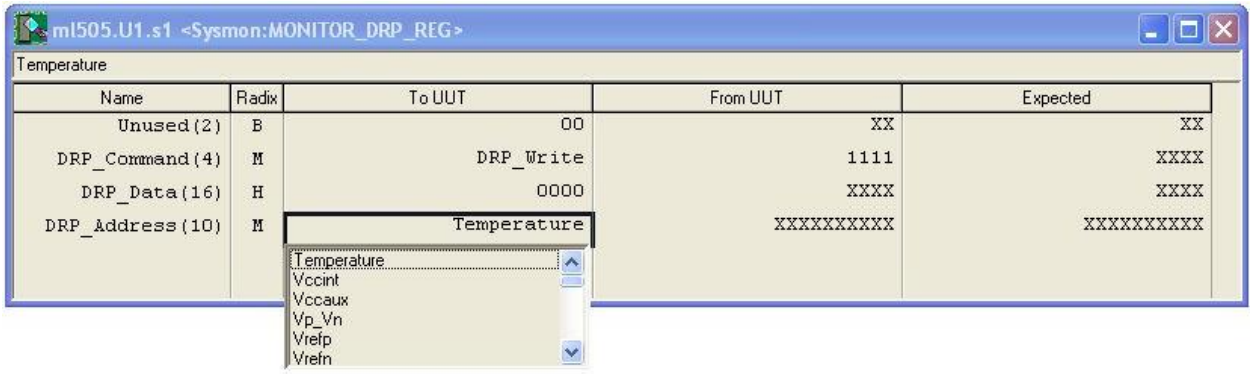

Figure 5. SYSMON instrument display

IEEE 1149.1-2013 Procedural Description Language

There's more to instruments than just JTAG-accessible test data registers. The IEEE 1149.1-2013 group also foresaw that a standardized language would allow an instrument provider to describe all of the functions associated with a specific instrument. This description, provided by the Procedural Description Language (PDL), can be used by anyone integrating the instrument into an IC. The result is one language for all vendors and customers, simplifying the use of these instruments and allowing re-usability.

```
#Read Target Temperature
iWrite DRP_Command DRP_Read
iWrite DRP_Address Temperature
iApply
#Acquire data from SYSMON read and return voltage
 iRead DRP_Data
iApply
set ADCVal [iGet -so -hex DRP_Data]
```

Figure 6. SYSMON PDL snippet

Figure 6 shows an example PDL snippet created for the SYSMON instrument. The general form for writing data to an instrument in PDL is "iWrite <register name> <value or mnemonic>". The command "iWrite DRP_Command DRP_Read" means to use the mnemonic DRP_Read from the package file previously described, look up the value (which is '0b0001') and write that value into a software model of the DRP_Command register. The iApply then moves the data from the software model to the target integrated circuit instrument via a JTAG/1149.1 scan operation. The user can work at a high-level English-like language that is isolated from the low-level details of the 1149.1 state-machine and the timing required for shifting test data per the IEEE 1149.1 standard. The iGet command has the format "iGet <source> <radix> <register name>" where <source> is the location of the data; in the case of –so, it means the "serial out" of the instrument. The <radix> is a hex, decimal or binary value and the <register name> refers to the

data read register. In this case DRP_Data is the data register connected to the ADC in SYSMON.

PDL also allows the instrument provider to define how to convert digital data from JTAG test data registers into real world voltages or temperature. For instance, Xilinx gives the temperature equation for the Virtex 5 SYSMON as:

$$Temperature\ (C) = \left(\frac{ADCVal * 503.975}{1024}\right) - 273.15$$

What the user wants to know or use is a temperature value that is in human-readable form, either in degrees Celsius or degrees Fahrenheit. PDL allows for linear and non-linear conversions of the digital data into meaningful human readable values, without requiring the user to be an expert on the silicon instrument details. In PDL, the ADCVal can be converted to temperature as shown in Figure 7.

```
#Convert hex ADC value to C or to F if necessary
set temperature [expr (($ADCVal/1024)* 503.975) - 273.15]
# if user passes "F" as a parameter, convert to Fahrenheit
if {$CorF == "F"} {
       set temperature[expr $temperature * 1.8 + 32]
}
```

Figure 7. PDL snippet for converting ADC to temperature

Intellitech has created a set of procedures for the SYSMON instrument such that the user can easily read voltage and temperature. These procedures are available in the NEBULA download.[3]

Linking to NEBULA from ATEasy

A set of Winsock procedures are available in the ATEasy programming language. The procedures provide support for communicating to 3rd party software using the TCP/IP protocol stack. TCP/IP has become the standard for data transmission over networks, including the Internet. TCP establishes a connection for data transmission and IP defines the method for sending data packets. Intellitech uses the ATEasy procedures to interface to the Intellitech NEBULA software. All of Intellitech's software has a capability called the Remote Command Interface (RCI). The RCI listens for connections and data from Test Executives like ATEasy via TCP/IP sockets. Figure 8 shows that within the ATEasy GUI, there are four procedures developed by Intellitech that interface between the ATEasy and the NEBULA software.

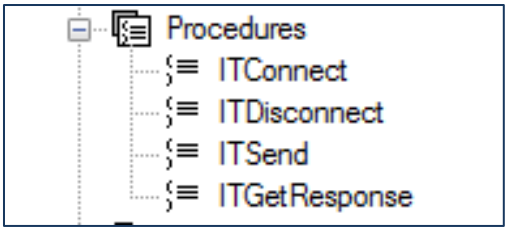

Figure 8. Four Procedures for use with ATEasy

A connection to a computer running NEBULA can be done in the ATEasy programming language as shown in Figure 9. NEBULA can be running locally with ATEasy (localhost) or on an entirely different computer.

```
!The Intellitech RCI (Remote Command Interface) allows third party tools to send text
!based commands over a TCPIP socket to the Intellitech software.
!The RCI can be enabled through the options menu in Eclipse, Nebula, and Scan
!Executive.

!Specify the IP address or hostname of the computer running the Intellitech RCI
!The computer does not have to be the same computer as the one you're currently on.
!You can also connect to a remote computer by name or IP.
!Note that when connecting to a remote computer you may have to configure the Windows
Firewall to allow remote connections to that PC.
!127.0.0.1 = local computer (the computer you're on now)
sHostName = "127.0.0.1"

!Specify the RCI port to connect to. The default value is 2000
lPort = 2000

!Call the Intellitech Connect procedure with the specified HostName and Port
!This creates a TCPIP connection between ATEasy and the Intellitech software

ITConnect(sHostName, lPort)
```

Figure 9. ATEasy code snippet for connecting to Intellitech's IEEE 1149.1 software

Putting it all together

Figure 10 shows an overview diagram of the ATEasy and NEBULA integration. The NEBULA software is used to manage all details of IEEE 1149.1-2013/JTAG. NEBULA can read and interpret compliant Boundary-Scan Description Language (BSDL) description files associated with all of the Integrated Circuits (ICs) on the PCB. It also can read in all of the Silicon Instrument package files which describe the registers of the instruments. NEBULA also has a compliant reader for IEEE 1149.1-2013 Procedural Description Language. This PDL is used by the instrument provider to describe the operation of the instrument and to convert digital values into real world values such as voltage, current and temperature. The BSDL and Silicon Instrument package files are compiled by NEBULA into a scan database. The scan database provides a machine-readable lookup of register positions and mnemonics that may be called out

by PDL commands. Overall control of NEBULA is performed by ATEasy via the TCP/IP socket interface.

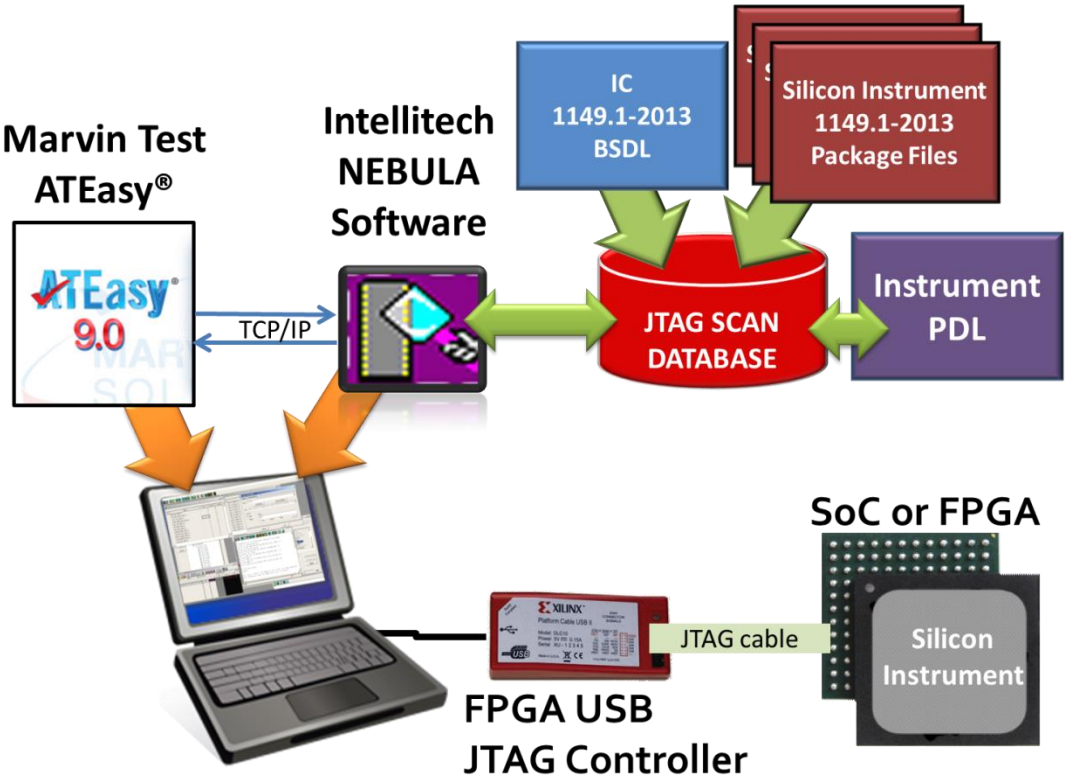

Figure 10. Overview diagram showing ATEasy and NEBULA integration

The commands that ATEasy sends to NEBULA are converted to JTAG accesses for communication with the various IC instruments. Figure 11 shows the ATEasy language necessary to read the on-die temperature from the Xilinx SYSMON. The ITSend command allows the ATEasy user to send any valid IEEE 1149.1-2013 command. NEBULA acts as a bridge and converts the command to the necessary JTAG sequences for accessing the on-chip instrument.

```
ITSend("[iCall U1.s1.readTemperature "C"]")

sretval = ITGetResponse()
print "Temperature is "; sretval
```

Figure 11. ATEasy programming language to read an FPGA on-die temperature

Figure 11 shows example ATEasy code to read the Xilinx on-die temperature. The iCall to U1 targets a Xilinx FPGA which is device reference designator U1 on the PCB. The SYSMON instrument inside U1 is s1. SYSMON has a number of pre-developed library procedures. One of these procedures is "readTemperature". The ATEasy user simply executes an ITSend to call this readTemperature procedure "in the context" of U1.s1. The "C" provided

after the readTemperature is a parameter passed to the procedure to return the temperature in degrees Celsius. "F" can be used to retrieve the temperature in Fahrenheit. The ATEasy ITGetResponse() procedure is then used to get responses back from NEBULA and the target instruments. In this case a variable "sretval" is used to hold the response from NEBULA and the temperature is simply printed in the third line as shown in Figure 11.

The ATEasy user should already be familiar with the ATEasy GUI. Figure 12 shows an example GUI with "Tests" under NEBULA_JTAG. The flow would be to first "Connect" to NEBULA and then load the PCB scan database as shown in "Load ML505 Database". This should always be followed by a check that the JTAG/1149.1 hardware is connected to the Unit Under Test (UUT) and that the 1149.1 TCK, TMS, TDI and TDO are all operational. Intellitech calls this the "scan path integrity test". Once this is tested, the next step shown is retrieving the on-chip temperature, followed by a check of the VCCAUX voltage and VCCInt voltages.

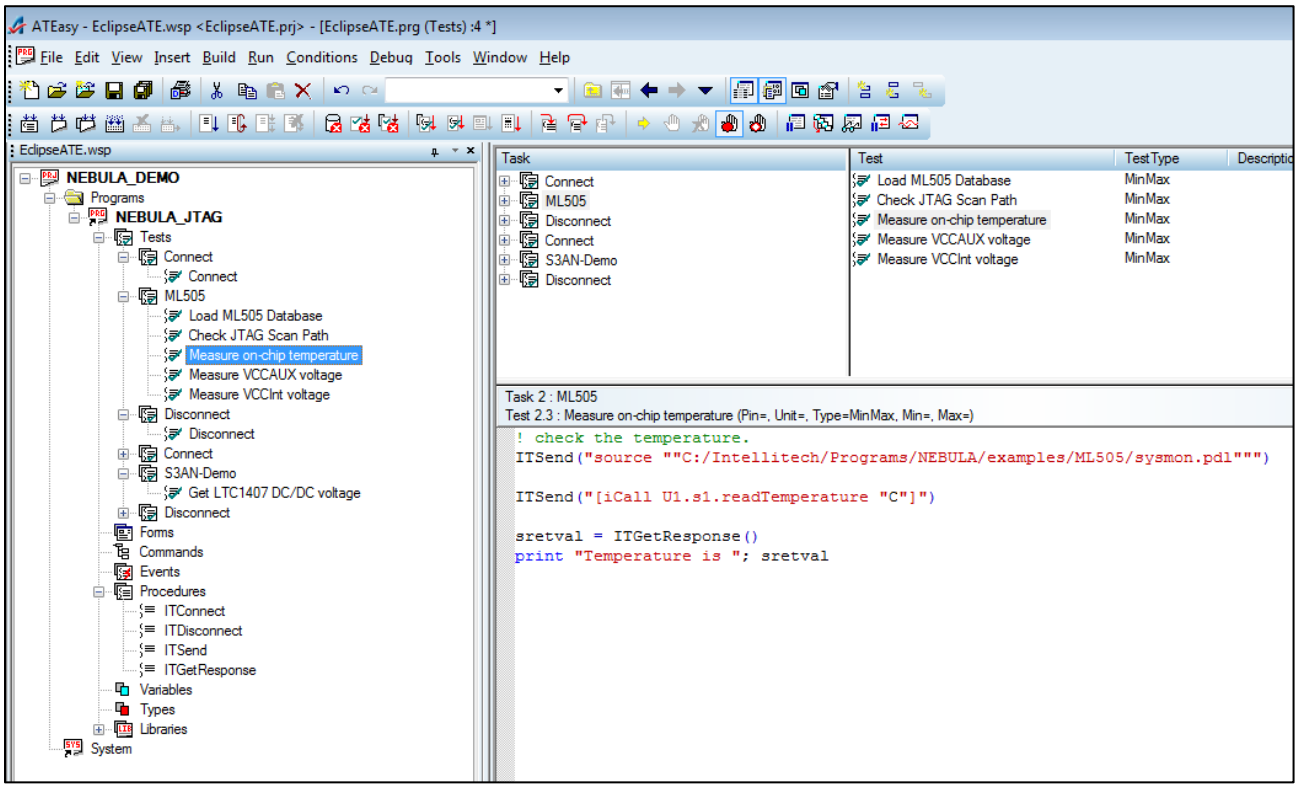

Figure 12. ATEasy program with JTAG instruments

Other voltages and currents can be measured within the limitation of the Xilinx SYSMON instrument. Also shown in the GUI is the use of an on-chip SPI instrument which can communicate to an external LTC1407 DC/DC converter for the programming and measurement of voltages associated with this IC. This is fully documented in the Spartan 3AN example in the NEBULA software download[3].

Conclusion

Silicon instruments can be pre-defined by IEEE 1149.1-2013 instrument providers and made available to board and system test engineers. This approach allows the expertise of the instrument maker to be fed forward to the test engineer. This flow reduces the test engineer's workload and improves test coverage and the overall quality of the product. Silicon Instruments are as useful as external GPIB/PXI/VXI/LXI based instruments and have the advantage over those instruments in that they don't require physical access.

Intellitech's NEBULA software provides an important bridge from ATEasy to JTAG accessible silicon instruments. The user can compile IEEE 1149.1-2013 instrument descriptions and operational libraries in PDL within Intellitech's NEBULA or Eclipse™ software and then access these procedures from ATEasy. While this white paper focuses on a simple SYSMON instrument available in Xilinx FPGAs, other instruments are available that support a range of test needs including at-speed testing of both SerDes and DDR memories. By incorporating tools such as NEBULA with ATEasy, board and system test engineers have additional test capabilities that offer robust and non-contact test capabilities.

1https://standards.ieee.org/findstds/standard/1149.1-2013.html

2 http://www.intellitech.com/ijtag/nebula-instruments-ieee-1149-1-2013.pdf
3 To download the examples and free NEBULA software, please register with a business email address on Intellitech's website at https://www.intellitech.com
4 https://www.intellitech.com/products/fac.asp
5 http://www.xilinx.com/support/documentation/user_guides/ug580-ultrascale-sysmon.pdf
6 http://www.xilinx.com/support/documentation/user_guides/ug192.pdf