DEBUGGING I3C PROTOCOL DESIGN ISSUES

Application Note-1

Introduction

The MIPI I3C Bus interface is an evolutionary specification that builds upon the legacy I2C standard. The aim is to reduce the number of physical pins used in sensor system integration, and supports low-power, high-speed digital communication typically associated with UART and SPI interfaces, so that I3C becomes a single interface combining all the capabilities of the legacy interfaces.

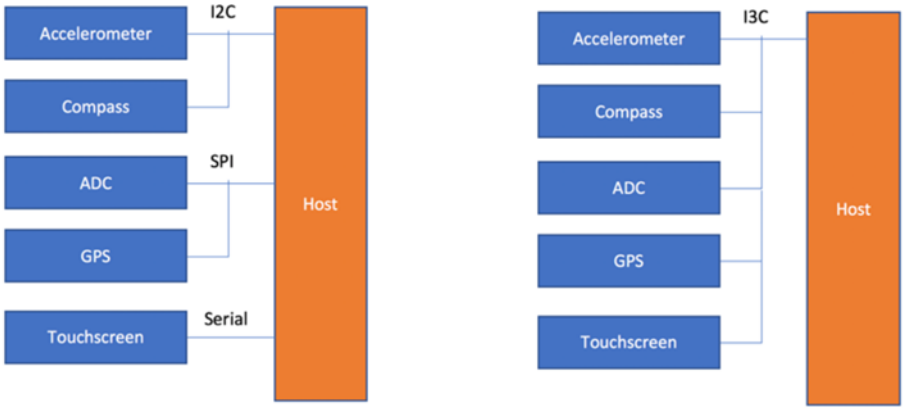


Fig1: Advantages of I3C

I3C has a multi-drop bus which, at 12.5 MHz, is over 12 times faster than what I2C supports while using significantly less power.

I3C's main features include:

- Backward compatibility with legacy I2C
- Multi-Master and Multi-drop capabilities-handles complex architecture and also enables faster through put.
- Dynamic Addressing-Helps in prioritizing Slave devices, helps minimize pin counts, which is key for meeting small form factors.
- In-Band Interrupts- helps eliminates the need for a separate general-purpose input/output (GPIO) for each slave, thus reducing system cost and complexity.
- Hot-Join support

The I3C interface is expected to play a fundamental role in streamlining sensor integration in smartphones, Internet-of-Things (IoT) devices and wearables. I3C can also be used to manage complex systems as and when designers migrate to a common management transport which drastically reduces cost and latency in such complex systems while also enabling new capabilities.

This Application Note is intended to help users understand how the I3C interface works and will bring out a few typical test cases faced by designers and how Prodigy solution helps address them effectively.

I3C Basics

I3C protocol supports several communication formats, all sharing a two-wire interface. The two wires are designated as SCL and SDA. SDA is a bi-directional data pin, whereas SCL can either be a clock-pin or a Bi-directional data pin while in certain HDR modes.

I3C Protocol Overview:

All I3C communications occurs within a frame. The frame begins with a START, followed by one or more transfers, and a STOP.

I3C interface supports Single Data Rate (SDR) messages which are similar to I2C Messages. Maximum clock speed is 12.5MHz. It also supports high data rate (HDR) messages. In HDR, data transfer is

equivalent to clock cycles. There are two types of messages Broadcast and Direct common command code (CCC) messages which allows the master to communicate with all or specific slave on the bus.

I3C protocol is based on frame encapsulation approach. The I3C frame always include START, the Header, the data and the STOP. I3C bus always initialized in SDR mode and never in HDR modes.

Common Command Code (CCC) commands protocol is formatted using only in SDR. CCC are transmitted to specific to slave or all slaves in I3C bus. CCC General format as shown in this figure.

S **Sr**	**7'h7E** / W / ACK	**Command Code** / T	**Data** (Optional) (Broadcast CCC only) / T	**Sr** **P**

For the HDR modes:

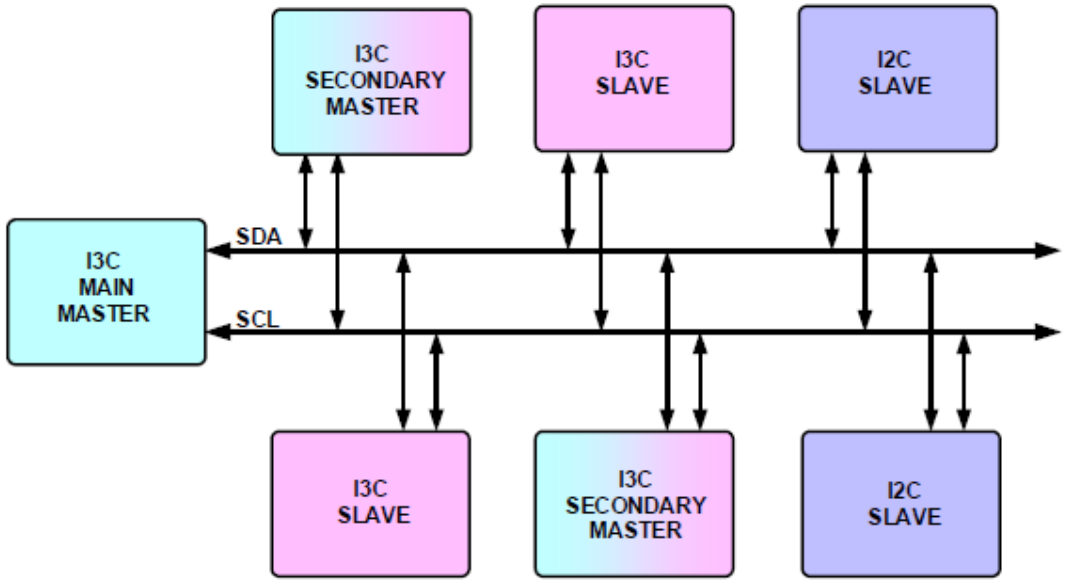
- First the dedicated Broadcast I3C address(7'h7E) is issued to all slaves on the I3C bus.
- Then one of the EnterHDR CCC's is issued, indicating that the Master is entering the HDR mode. Each HDR mode has its own EnterHDR CCC.
- This is followed by one or more HDR transfers.
- HDR mode is ended by using the HDR exit pattern protocol.

For more details on I3C protocols, please refer to MIPI® I3C specifications.

I3C devices have Bus and Device characteristic registers which will hold information about the capabilities of I3C device.

I3C Bus Configuration:

I3C bus can be configured with multiple devices. These devices are I3C Main Master, I3C secondary Master, I3C Slave and I2C Slave.



I3C devices many have many features as appropriate for their function in the I3C bus. Depending on the I3C Bus system need, it may not be necessary that all features are enabled for any particular bus instantiation. Enabled features of every I3C device shall be described in characteristic register of the device. Master obtains the characteristic register info during the power ip state of I3C bus.

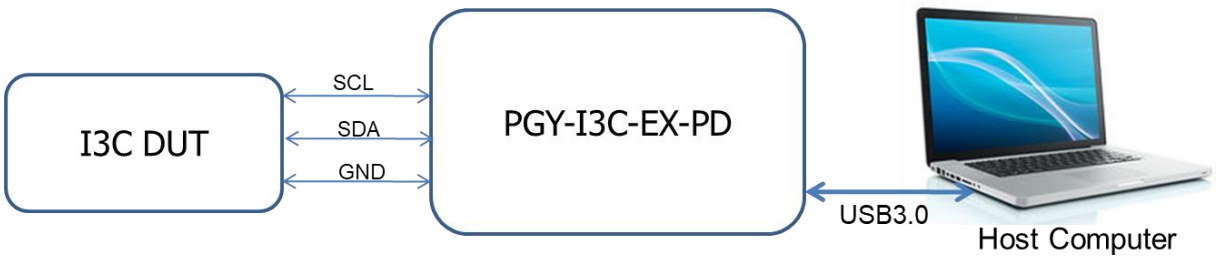
At start up stage from power-down, main master shall assign unique dynamic address to every device on the bus including itself. Dynamic address creates a priority ranking of device interrupts. If any secondary master present in the I3C bus shall be made aware of dynamic address and characteristic registers.

I3C based electronic hardware design involves design of I3C Master, I3C Slave and I3C system designs. These designs offer different challenges at different state of the design cycle. Design engineers need a tool to debug the I3C device for reliable operations. Some of the important requirements are

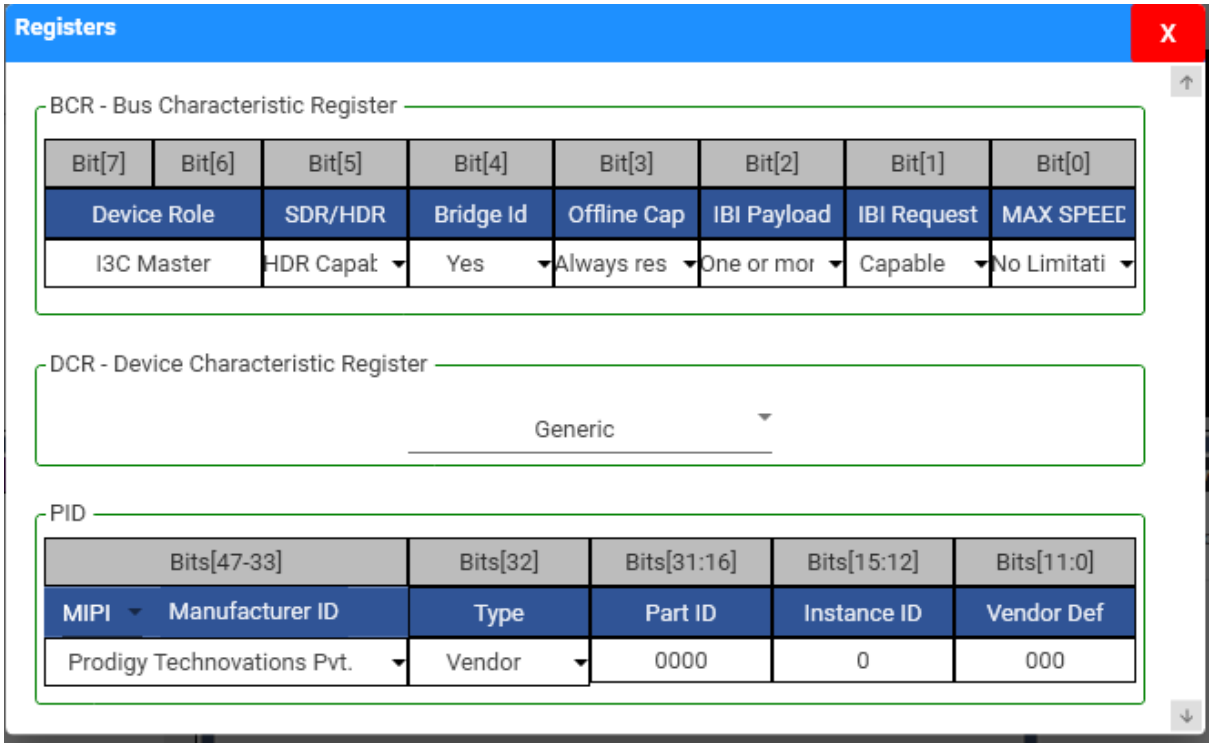
- Designers who are developing master or slave, they need fully working Master or Slave device which designers can configure and test their designs.
- Make sure I2C devices can co-exist with I3C bus
- They need a tool which will capture the communication between Master and slave and analyze the protocol traffic for protocol errors
- Need a tool to capture protocol traffic at a specific event. These events could be anticipated error condition or protocol activity
- Need to capture protocol activity over long period of time (may be days) and ensure design is reliable
- Emulate I3C bus using the designed Master or Slave device in I3C network

Testing a Master or Slave:

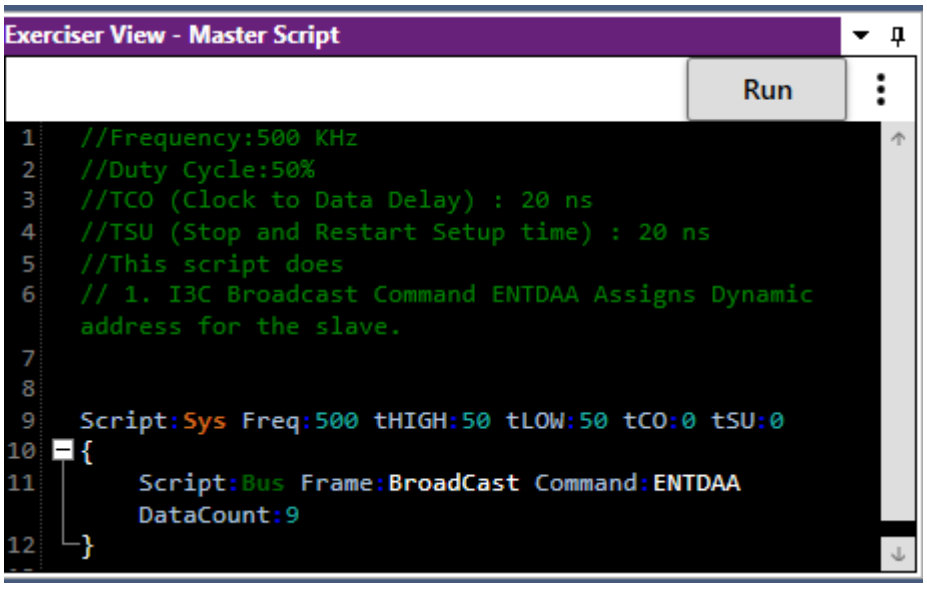
PGY-I3C-EX-PD is a I3C Exerciser and Protocol Analyser. This product is developed as per I3C V1.0 specification document. Typical test setup for Testing designers master or slave is as below



I3C DUT can be I3C Master or Slave or Secondary master. Software that runs in host computer which configures the PGY-I3C-EX-PD has a Master or Slave depending on DUT is Slave or Master. User can set up the Master or Slave characteristic registers of PGY-I3C-EX-PD based on the requirement in DUT by selecting appropriate in software as below



Write or use existing script or using GUI in PGY-I3C-EX-PD software designers can initiate the communication with DUT. For example, user can send ENTDAA command which is instantiation command during the power up state .



By running this script, Master (in this case PGY-I3C-EX-PD) sends the commands to DUT (in this case it is slave). Properly designed DUT is expected to give acknowledgement to this command. Analyser in the PGY-I3C-EX-PD captures this exchange of I3C packets between master and slave and list in protocol list and als0 timing diagram window. Designers write complex scripts and test DUT for its complete functionality as per I3C Version 1.0 specification.

I3C Design challenges:

Example : A design engineer needs to resolve issues facing with slave device working with the I3C Master device.

Potential issues:

The slave device may be an I2C device.

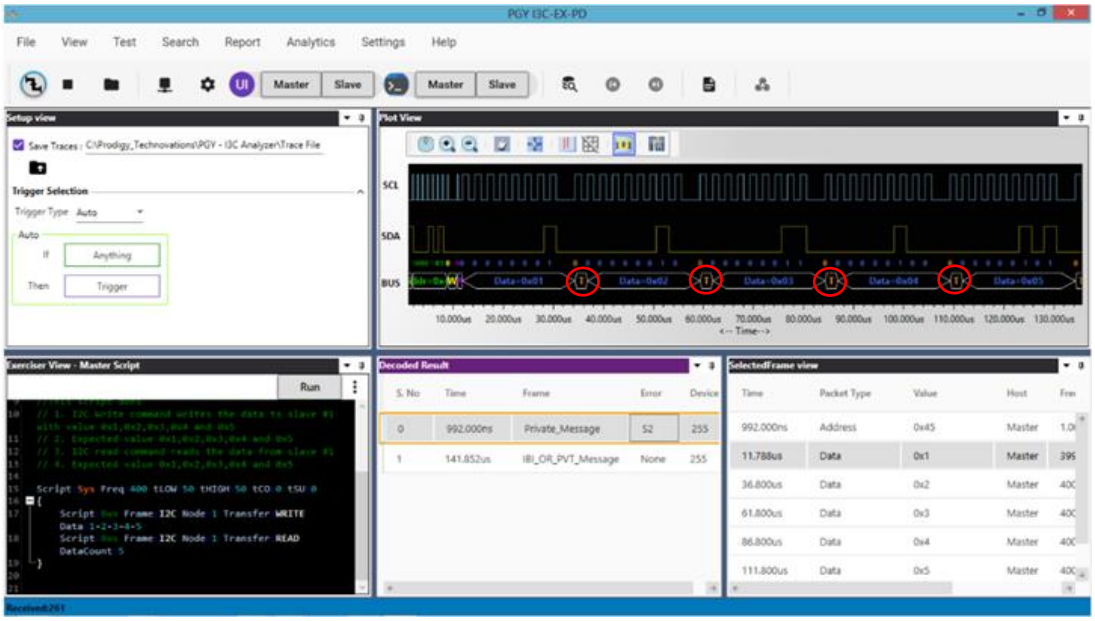
The slave may not be correctly configured.

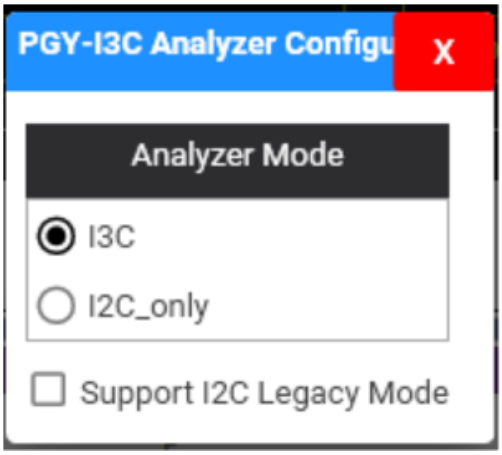
Likely solution:

(a) While it is expected for I2C devices to work in an I3C environment, the key thing designers need to note is that for I2C devices, 50ns spike filter must be present on scl line. If this is not handled there is possibility of the I2C device inadvertently take over the I3C bus communication causing major design issues.

(b) I3C Master, as per specification should decode 9th bit as Ack/Nack for I2C communication but can be seeing a T Bit which can lead to wrong behavior.

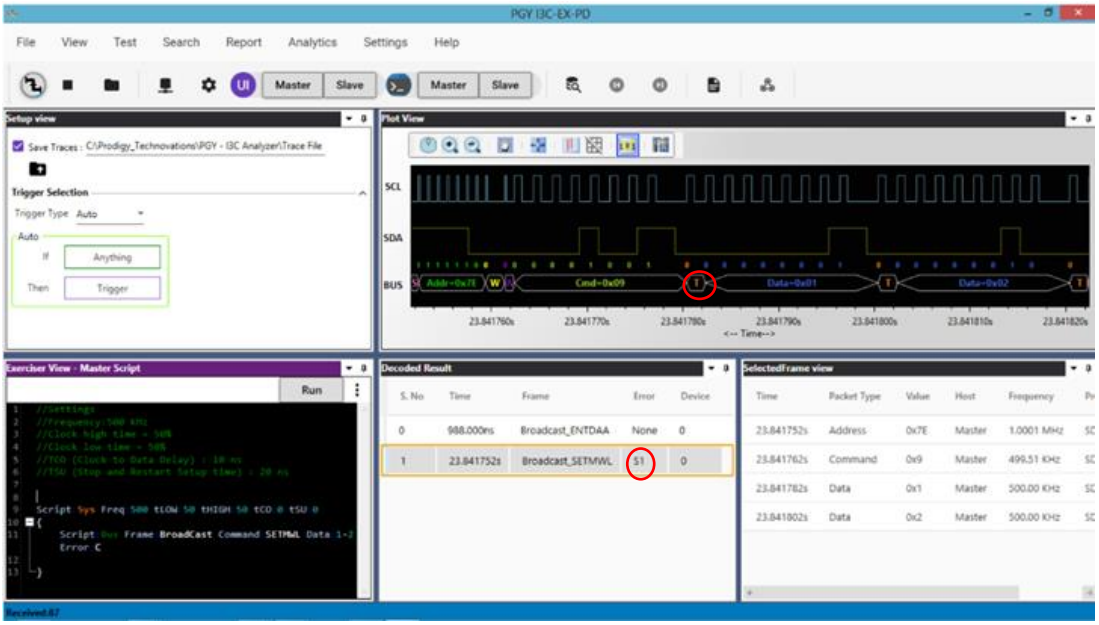
Also as per I3C specifications there are defined reserved address for I2C devices, designers can observe problems if their I2C slave device has been assigned a different address.(see red highlighted parts below)



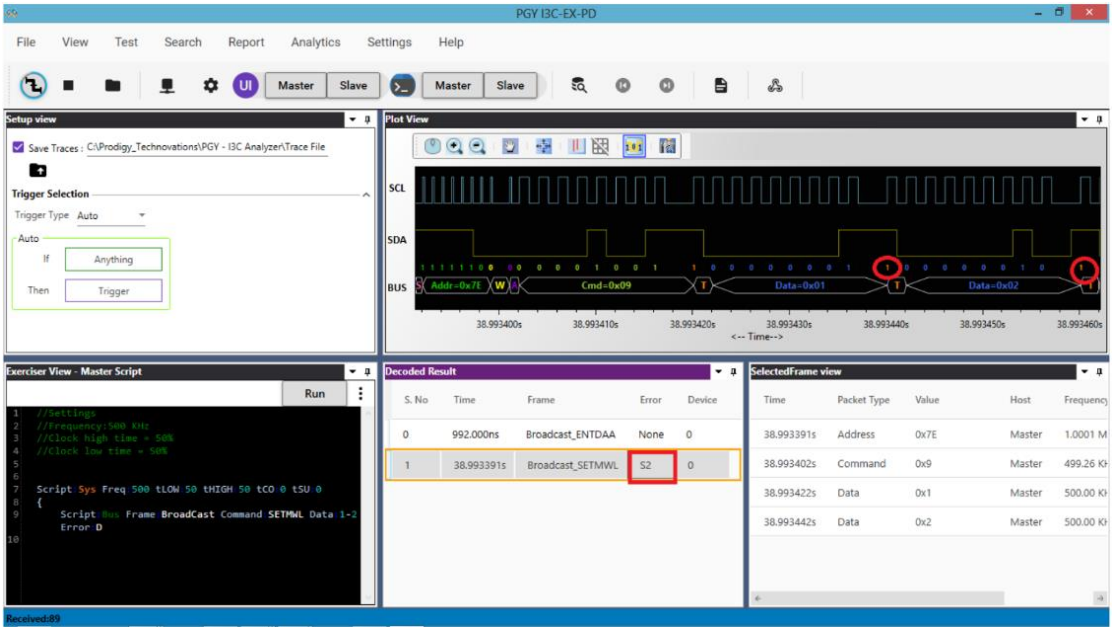


PGY analyser gives the ability to analyse commands either as per I2C or I3C specification thereby enabling designers to decode the issue effectively.

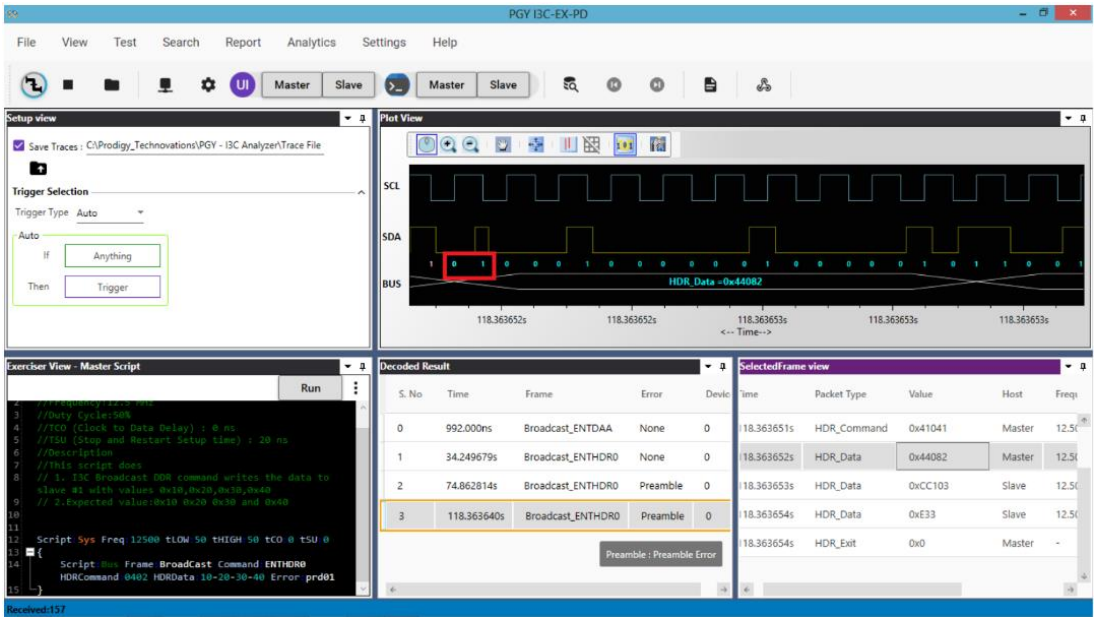
(c) Command Format (parity error) will be incorrect. If the implementation is incorrect, one can observe parity error as highlighted in the below screenshot leading to incorrect behaviour.



(d) Data parity error causing issues. If incorrect data is sent leading to data parity error as highlighted below, it leads to incorrect behavior. These needs to be debugged and corrected to ensure proper functioning of the design.



(e) In case of incorrect implementation of the protocol, one can notice Preamble error which can lead to issues. Designers need to ensure these are corrected to ensure the design works as per I3C specification needs.

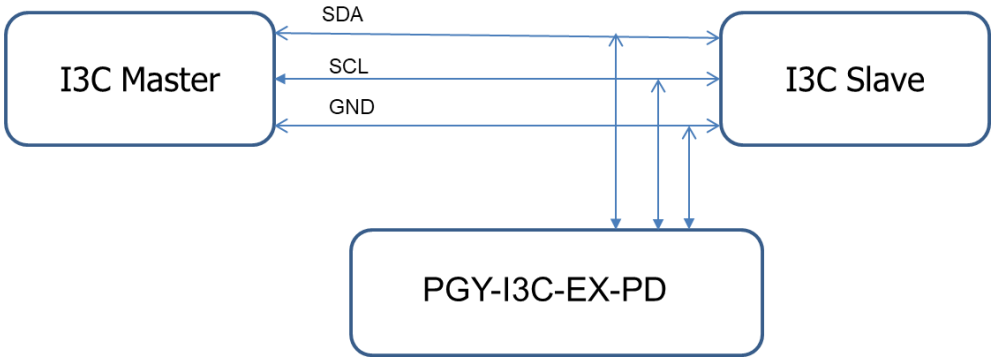


(f) Design issues can occur if the design is not able to handle full speed topology. Typically one can come across situations when the design does not work in higher speeds and this can be attributed to design timing flaws and it is very important to be able to look into these timing variations to ensure proper working.

PGY-I3C-EX-PD analyzer Timing and plot diagram helps identify timing issues. Designers can then fine tune their designs to meet the I3C timing specifications.

Protocol and Error Analysis of I3C messages on I3C Bus.

While testing interoperability between I3C master and Slave designers need a tool which can sniff the I3C bus and report the protocol activity and analysis of protocol activity for errors

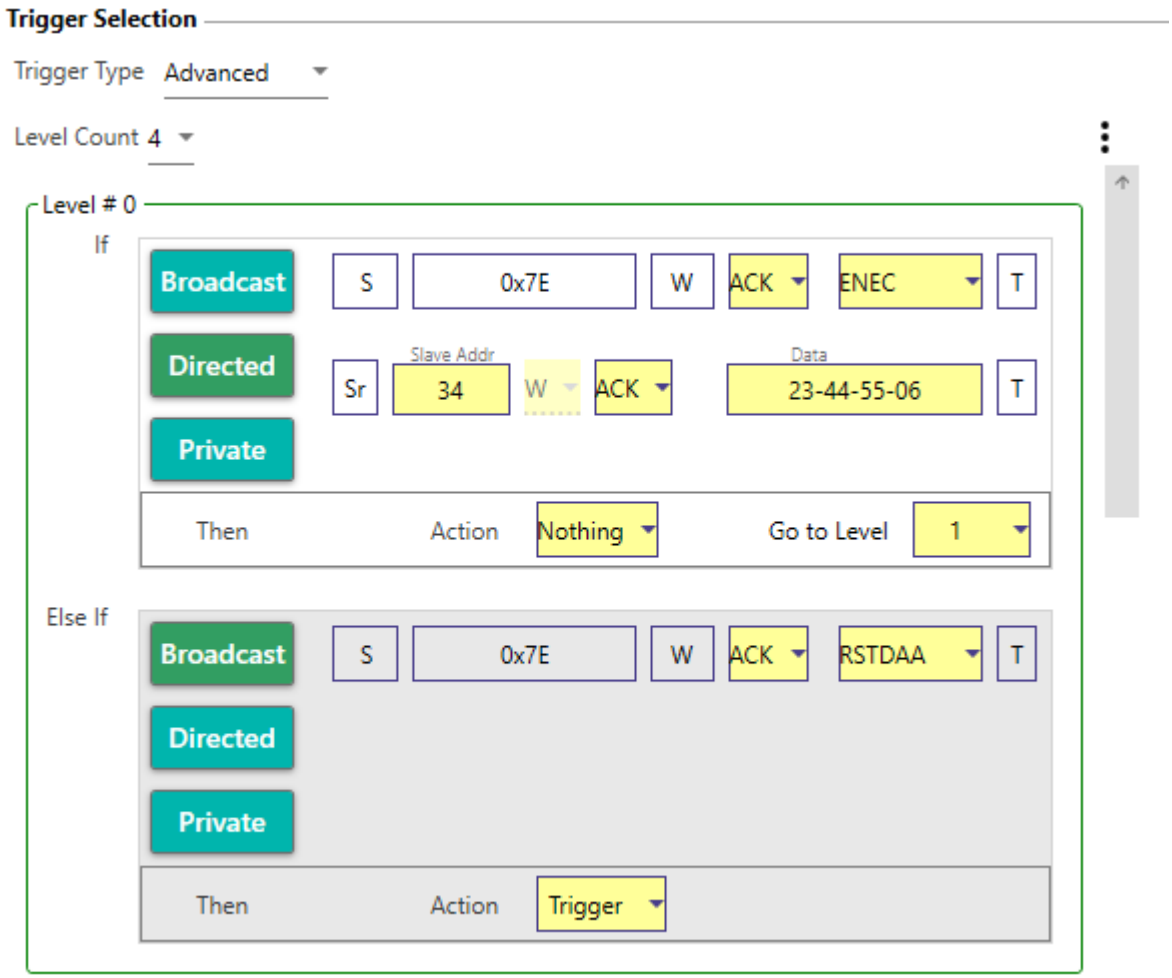


Analyser in PGY-I3C-EX-PD will continuously sniff the communication in I3C bus. The decoded results are displayed in Protocol listing and timing window of the software.

<Include Protocol listing results and timing plot>. summarize the different protocol errors in results and capability in PGY-I3C-EX-PD

PGY-I3C-EX-PD supports continuously streaming of Protocol activity to host computer hardware. This enables almost many gigabytes of storage buffer (theoretically limited by host computer storage space) to capture I3C Protocol traffic in I3C Bus.

PGY-I3C-EX-PD supports powerful trigger capability. This allows design and test engineer to capture Protocol activity at specific event and analyze elusive problems in I3C bus. PGY-I3C-EX –PD triggers on any I3C Protocol activity, specific protocol activity or If-then-else if trigger capabilities.



In this trigger condition PGY-I3C-EX-PD monitors for two trigger conditions. If Command ENEC with specific slave address 34 (user defined) and data 23-44-55-66 (user defined is) occurs on the I3C bus then goes to Level 1 and start monitoring the trigger condition defined in Level 1. If instead of ENEC command, broadcast RSTDAA is on the bus it will trigger on it and notify the user 'T' labelling in protocol list and timing plot.

Need to capture Protocol over long period of time and error analysis

I3C interface technology is targeted for Mobile, IOT and automotive applications. In this environment the electronic system expected to work round the clock with very high reliability.

This demands testing of I3C devices over few days i.e. continuously running different test cases over long period and monitor the performance of I3C protocol. PGY-I3C-EX-PD supports API support. API in PGY-I3C-PD allows test engineers to write automation script in Python or .net frame work and log the results. Using the API support test engineers can run different test cases and run these test cases over and over to locate at different sequences to locate any faults in designs.

Emulate I3C Bus Configuration:

In real world, I3C devices are expected to work in I3C bus configuration. Where there will be main master, secondary master and Multiple I3C Slaves. Hence it is important for design and test engineer test the I3C devices in I3C Bus configuration. PGY-I3C-EX-PD can be configured as single master, 3 slaves. User can setup the different characteristic registers of I3C devices in PGY-I3C-EX-PD and configure the I3C Bus along with DUT. Once the I3C bus is setup, the I3C bus diagram can pictorially viewed as below.

Conclusion

The MIPI I3C standardized sensor interface is a revolution for integrated sensor systems which is being extended for other interface needs as well. I3C has a superset of features over the existing I2C with additional high-speed data rate modes that satisfy the use cases where SPI is required. I3C is an evolutionary standard which combines the benefits of I2C & SPI with additional features such as in-band interrupts, dynamic address allocation & advance power management which benefits smartphones and wearable development.

Designers face many implementation challenges which can be due to incorrect hardware implementation, incorrect protocol implementation and signal integrity issues. All of these need to be debugged before the deign in considered final to ensure acceptance by the customers.

The Prodigy PGY-I3C-EX-PD can be used as an All-in-one tool to address the challenges a design engineer faces during design and testing of future innovative products.

For additional support and details, please leverage the below information:

(a) **contact@prodigytechno.com** (b) http://www.prodigytechno.com