International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 7– No.7, August 2014 – www.ijais.org

Design of SD/eMMC Protocol Compliance Solutions

Aparna N, Mamatha HR
Dept. of Information Science and Engineering
PES Institute of Technology
Bangalore India

Dheerendra Singh Tomar, Rakesh Kumar
SoC Post Silicon Validation
Intel Technology India Pvt. Ltd
Bangalore, India

ABSTRACT
Protocol Validation is an important step in validating the I/O interface of the System on Chip (SoC). In developing the protocol compliance solutions for validating the Secure Digital/Embedded Multi Media Controller (SD/eMMC) protocol, the main intention is to capture the behavior of protocol under various physical conditions using a Logic Analyzer (LA). LA facilitates more capture of data and offers more number of channels and is more cost effective compared to existing Digital Storage Oscilloscope (DSO) methodology. In addition to decode, the solution is capable of carrying out the analysis of the data captured and decoded from LA. Analysis functionality includes command-response analysis, cyclic redundancy check, integrity check and flow analysis that assess and reports the functionality and performance of the protocol under variations of temperature & voltage conditions. Analyzer helps in tracing out all kind of deviations from the specifications and standards defined for SD and eMMC.

General Terms
Protocol analysis, Secure Digital, Embedded Multi-Media Controller

Keywords
Digital Storage Oscilloscope Embedded Multi Media Controller, Logic Analyzer, Secure Digital, System on Chip.

1. INTRODUCTION
System-level post-silicon validation is a function that is both resource intensive and time consuming. This being the last stage of IC development cycle, has considerable impact on time-to-market, quality and yield. The interface of microprocessor with other components like secure digital card, embedded multi-media controller, inter-integrated circuit, serial peripheral interface etc. is been validated under different stress conditions of temperature and voltage. High valued equipment like digital storage oscilloscope is used to tap the communication between the host controller and the components resident in new generation tablets and smartphones. With no doubt DSO offers uncompromising accuracy that helps to trace out defects probable to happen with the processor interface to protocols. Although this benefit is very much helpful in case of electrical validation, as far as protocol validation is considered more weightage is given for duration of capture more than accuracy. The reason behind this is, with more duration of capture more packets can be captured to perform analysis which in turn facilitates more protocol errors to be uncovered. The investment for oscilloscopes for the dedicated performance of protocol validation compels to think about an alternative solution that is less expensive with some additional advantage to satisfy the requirement.

Obviously relevance of any equipment depends on the context in which it is used. As far as protocol validation is considered much focus is given to analysis of more protocol data than electrical measurements. Logic analyzer takes its role in this context. Protocols specifically SD/eMMC makes use of 4 to 8 data channels for its communication with host controller. One of the limitations of DSO is the number of analogue channels that it offers is minimal which restricts analysis to confine to 3-4 channels. Low cost handy LAs that offers around 16-32 analogue channels can be effectively used for the purpose. Record length of LA is more compared to DSO. Moreover the cost of LA is much cheaper compared than DSO (4GHz) [1-2]. In this paper we propose a better efficient solution that performs decode of the data tapped using LA and also the analysis of the data decoded to ensure the compliance of result with JEDEC Standard 4.5 (for eMMC)[3] and SD Physical Layer Specification (for SD) [4]. Apart from this, SD/eMMC analyzer is designed to work over the existing system that makes use of oscilloscope for the reason not to miss out the accuracy advantage offered by the oscilloscope can be considered as a case study.

Overall we summarize the contributions of this paper as follows:

- Decode of data using LA

  Data communicated between host controller and memory card is tapped using logic analyzer and decoded. The required fields are extracted and stored in excel and pytables.

- Analysis of protocol processed data

  Decoded (protocol processed data) is analyzed to check its compliance against integrity, command response analysis, timing analysis, cyclic redundancy checks with respect to specification standards

- Performance of flow analysis

  Sequence of packets is checked against the desired flow mentioned in JEDEC 4.5 specification (eMMC) and Physical layer specification (SD)

- Summarize and data & result in to Pytables [5]

  Result of analysis, flow analysis including processed and preprocessed data is entered in to Pytables for future reference.

The rest of this paper is structured as follows: Related work is discussed in Section II. Section III describes the system

overview. The implementation details are introduced in Section IV. Section V gives the result evaluation. Finally, the paper is concluded in the last section.

2. RELATED WORK

The simulation-based approach is one among the popular approach for verification. Protocol specification properties are represented in HDL (Hardware Description Language) [6] monitors. To perform verification HDL monitors are simulated along with the design under verification (DUV). All these are certain approaches to measure the quality of a simulation trace. But still the compliance cannot be guaranteed completely. In general, simulation-based approaches suffer from false positive problem.

Model checking techniques are used for Serial Peripheral Interface (SPI) protocol compliance verification in [7]. Here the Compliance Test Language (CTL) language is used to specify the protocol properties. Model checker performs the verification of the DUV against these properties. Success reported by the model checker implies 100% compliance with the properties However translation of protocol properties written in specification to CTL is complicated. Possibility to miss out some properties implied in the specification is more. When the design size increases there is possibility of memory explosion and long runtime. These issues prevent model checking techniques from being effectively and efficiently applied for compliance verification. Usage of verification protocol is another method but still its limitation is in the extent to which it can be customized to meet the specified requirements.

Assertion-based verification (ABV) methodology [8] is another approach for compliance verification. Many property specification languages (such as PSL 1121, OVL 1131, OVA) exists to provide different ways to specify properties in additional to CTL (Compliance Test Language). These approaches are implemented in pre-silicon validation. These languages are comparatively simpler than CTL at the semantic and syntactic level. However, whatever emerging property specification language is used, both dynamic ABV and static ABV suffers from the same problems that we mentioned above.

To overcome the draw backs an alternative methodology that uses high level languages that perform compliance checking in the user level is implemented in this paper. It can provide more control to the user in addition to ease of use.[9] Describes a protocol analysis platform based on open source software which provides a general, extensible, and easy-to-use framework. The implementation was based on functional prototype for TETRA trunking system, which can provide packet inspection at bit level, message flow analysis, and information extraction. The knowledge obtained from the architecture of protocol analysis platform is made use here. Along with the user interface which is easy for the end user to perform protocol analysis, the decode functionality implemented on the top of logic analyzer based capture methodology provides a more promising solution for the existing problem.

3. SYSTEM OVERVIEW

The overall picture of the system implemented is described here. The subsection 3.1 describes the existing system. The proposed system is described in subsection 3.2.

3.1 Existing System

Currently decode of protocol data communicated between the protocol and processor is captured using an oscilloscope and analysis been done manually. The input analogue signal is sampled and then converted into a digital record of the amplitude of the signal at each sample time. The displayed trace can be manipulated after acquisition; a portion of the display can be magnified to make fine detail more visible, or a long trace can be examined in a single display to identify areas of interest. Many instruments allow a stored trace to be annotated by the user. The limitations of the existing approach are:

- Record length

  Memory depth of DSO is limited restricting the record length. For an Agilent DSO with 4 ch x 10 GS/s (Standard) memory depth is 20Mpts on 4 channels [10]. As far as protocol validation is concerned, more protocol data is required. Hence more record length is expected

- No: of analogue channels

  Number of analogue channels is limited. Total number of analogue channels offered is limited up to 4 channels. So in case of validation of protocol interface like SD & eMMC, which requires up to 10 channels (1CLK, 1 CMD/RSP, 1-8 DAT channels), oscilloscope cannot solve the problem

- Size and space requirement

  Size and space required by the oscilloscope make it inconvenient to carry out the activities.

These limitations compel us to thinks for a better solution.

3.2 Proposed System

Considering the limitations mentioned in the previous section an alternative methodology that replaces the existing system and also that can provide some added advantage has been thought for. Using logic analyzers to capture the protocol data will solve the problem to a great extent. Logic analyzer is very handy equipment that offers 16-32 channels and more memory depth. It clearly fills the limitations of existing system.

The proposed system make use of low cost handy LA and a three layered wrapper is developed for decode, analysis and ease of use. The external interface is highly customized to perform decode and analysis activity using LA pertaining simplicity. The decoder proved to parse up to 50,00000 bits (binary) dumped using LA in to the CSV file. Decoder extracts respective fields of each CMD/RSP packets as specified in the Specification standards and store the decoded packets in excel/pytables. In addition to decode, the software carries out the analysis of the decoded from LA which is also compatible with protocol processed data from oscilloscope carried out as a part of post silicon validation of SoC [11-12].

Analysis includes command-response analysis, timing analysis, cyclic redundancy check, integrity check and flow analysis that assess and reports the functionality and performance of the protocol under different physical conditions. Analyzer helps in tracing out all kind of deviations from the specifications and standard which greatly helps in producing high quality products before being released to the market. The solution is extremely unique because of its

methodology as well as features that it offers facilitating in-depth analysis.

4. IMPLEMENTATION

We have designed the logic analyzer decoder whose functionality is to perform decode of protocol data communicated between SD/eMMC card and the host controller. It is responsible for decoding command and response packets and store the decoded result in Pytables.

The architecture of Decode/Analysis platform using LA is shown in Fig 1.

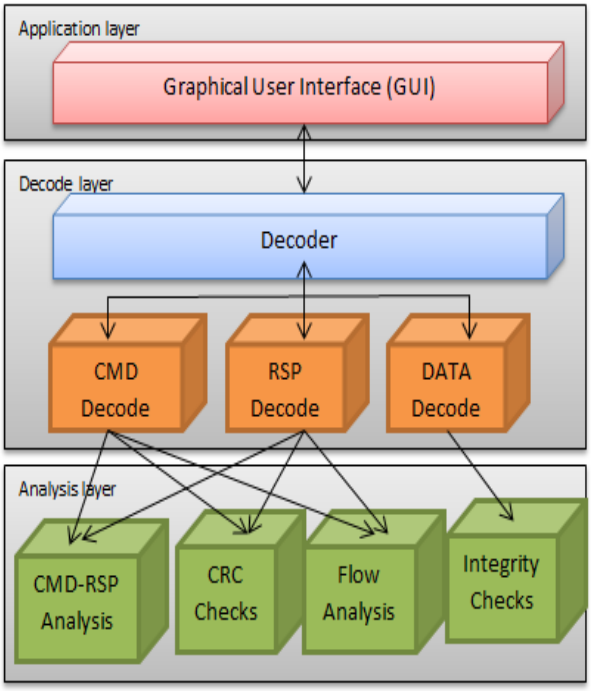

Fig 1 Architecture of Decode/Analysis Platform

The system is designed in such a manner that it receives inputs from user which is specific to post silicon validation and also the location pointing to the raw data captured from LA. The decoder takes its role after it gets the user inputs which dump the required data. Command decode and response decode happens in the same channel with reference to the clock data. If the protocol is eMMC, data can span is across1 or 4 channels. Analysis layer manages to perform command-response analysis, cyclic redundancy check and integrity check and flow analysis as per JEDEC 4.5 Electrical standard for eMMC and Physical layer simplified specification 4.10 for SD. As a case study, analysis layer is integrated with existing oscilloscope solution which performs analysis on the top of oscilloscope result.

LA Decoder dumps the data taking input from CSV file captured using logic analyzer. Depending up on mode of operation dumps the data with respect to clock, CMD/RSP and data. The extracted data with respect to clock, command and data channel is given as input to the parser. Parser module parses the command response and data bits in correspondence to clock and determines the type of packet. A decision is taken depending on the type of the packet and decides to decode either 136 bits (in case of R2 responses) or 48 bits (in case of commands and other responses). Protocol processed data is stored in pytables. The decoder manages small fluctuations in clock by keeping a tolerance value which it uses to keep synchronization.

The analysis layer capable of performing command-response analysis, timing analysis, cyclic redundancy check based on the standards /specifications of Secure Digital & embedded Multi-Media Card. Integrity check can be carried out based on user requirement make use of Knuth-Morris-Pratt algorithm [13]. The idea behind implementing integrity check is to capture integrity errors that escape out of cyclic redundancy check which makes the analysis more robust. The data been written in to the memory card is the input pattern which is to be searched in the data dumped in the data line. This ensures the integrity of data been written in in to the card.

The flow chart representing execution flow is shown in fig 2

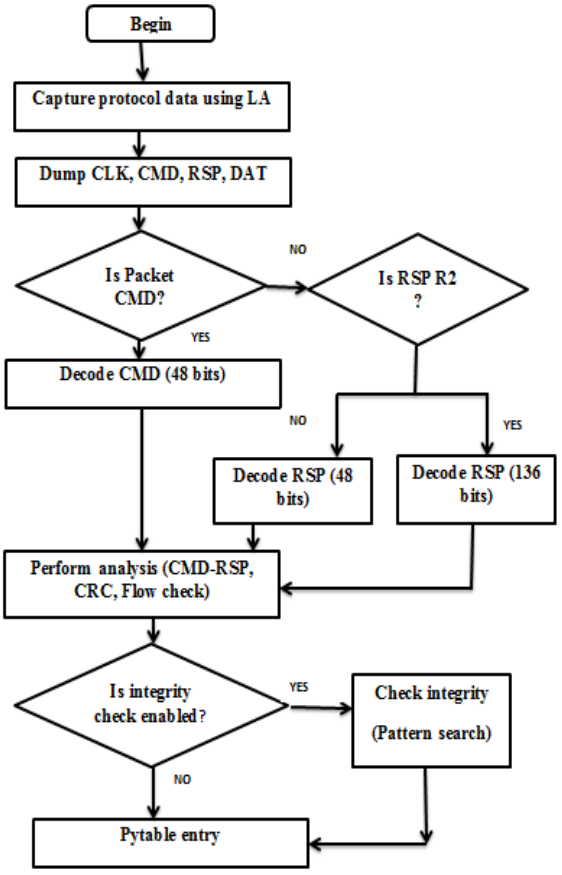

Fig 2 Flow of execution

Flow analysis checks for the sequence of packets described in boot mode, interrupt mode, data transfer mode and device identification mode appropriately defined in JEDEC 4.5 Electrical specification for eMMC. Flow verification ensures the conformance of order i.e. the sequence of packets been exchanged between memory card and the host controller with the respective specification. Any deviation from the expected flow is reported to be a failure. The solution is very much promising in terms of functionality that offers a great advantage over the existing solution.

The algorithm describing LA decoder is given in Fig 3 which specifies the logic applied to decode the data packets transmitted between the host controller and the card using LA.

Algorithm 1: LA Decoder

Input: Raw data (CLK, CMD/RSP, DAT) captured sing LA

Output: Decoded result

```
1.  CLK bits kept for reference (for SDR/DDR)
2.  ULimit : Predefined no of bits to be parsed (say 50,000)
3.  begin
4.      foreach bit( ↓ bit=1 (SDR), ↓ bit=1 and 0 (DDR)) ∈ CLK from start to ULimit do
5.          Dump bits ∈ CMD/RSP
6.          Probe for START_BIT (=0)
7.          if packet is CMD //CMD : TRANSMISSION_BIT =1 then
8.              Decode up to 48 bits
9.          elif packet is RSP //RSP : TRANSMISSION_BIT = 0
10.             if Response is R2 //Verify CMD to determine R2 RSP then
11.                 Decode up to 136 bits
12.             else
13.                 Decode up to 48 bits
14.         Redefine start position
15.         if START_BIT of packet not found after parsing ULimit then
16.             Probe sequentially till START_BIT
17.             If EoF reached then
18.                 break
19.             Go to step 4
20.     foreach packet decoded do
21.         Identify PACKET_NAME, PACKET_TYPE
22.         Extract START_BIT,TRANSMISSION_BIT,INDEX,ARGUMENT,CRC,END_BIT
23.         Store decoded result
24. end
```

Fig 3 Algorithm of LA Decoder

5. RESULT EVALUATION

The performance of decode of data is using LA and the analysis of result can be performed in any mobile platform with sufficient hardware support. All the results will be stored in pytables which provides a consolidated media that also facilitates to compare the validation results captured using logic analyzer and oscilloscope.

The execution will perform decode and analysis functionality using the protocol data captured using LA. Protocol analyzer is compatible with DSO result, and hence it can be used to perform analysis of data captured using DSO. By default the analysis functionality includes CMD-RSP analysis, CRC checks. Timing analysis is performed over protocol processed data (PPD) from the oscilloscope. Integrity check is performed over LA data. All the flow analysis can be performed over PPD captured from both the oscilloscope and the logic analyzer up on user's requirement. The analysis result will be entered in to an excel file for each test. Apart from this, every time when the test is run, all the details including preprocessed data, processed data, and analysis result will be entered in to the Pytables. Pytables also keeps the reference flows for all the four flow analysis.

6. CONCLUSION AND FUTURE WORK

Logic analyzer based solution to decode the protocol data captured in the communication between host controller and SD/eMMC card works correctly. Analysis of protocol processed yield accurate results. The system proved to be a complete solution that performs processing as well as analysis of protocol data that verifies the compliance of SD/eMMC protocol with respective specifications. As a future enhancement the same system can be tried over pre-silicon validation that makes use of emulator to perform the execution. This left shift helps to capture more error before the manufacture of silicon will be really appreciated.

7. REFERENCES

[1] Digital Storage Oscilloscope specifications, Agilent http://www.home.agilent.com/en/pd-1632466-pn-MSO9404A/oscilloscope-4-ghz-4-analog-plus-16-digital-channels

[2] Logic Analyzer, Saleae, https://www.saleae.com/logic

[3] Embedded Multi-Media card, JEDEC eMMC Electrical Standard, Ver4.5, http://www.jedec.org/sites/default/files/docs/JESD84-A441.pdf

[4] Secure Digital card, SD Physical layer simplified specificationVer3.01, https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf

[5] Pytables user's guide, http://pytables.github.io/usersguide/tutorials.html

[6] Fey G, Drechsler R, "Improving simulation based verification based on formal methods" In Proc. Design Automation Conference (DAC) pp 640-643, January 2004

[7] Nguyen, M.D., Thalmaier, M. Wedler, M. , Stoffel, D. , Kunz, W. ; Bormann, J., "A re-use methodology for formal SoC protocol compliance verification" In Proc. IEEE International Symposium VLSI, pp.1-6, 2005

[8] Ping Yeung, Larsen K, "Practical Assertion based formal verification for SoC design" In Proc. IEEE International Symposium VLSI, pp.58-61,November 2005

[9] LI Hai, YU Yue-yue, LIU Chang, CUI Li-na, "A protocol Analysis Platform based on open source software" In Proc. IEEE Conference on Instrumentation & Measurement, Computer, Communication and Control, pp.846-849, June 2012

[10] Data sheet, Agilent Technologies, http://cp.literature.agilent.com/litweb/pdf/59903746EN.Pdf

[11] Intel documents: High level architectures, Board schematics, Equipment user guides

[12] Chipsets details, Intel documents, http://www.download.intel.com/design/chipsets/labtour/PVPT_WhitePaper.pdf

[13] Pattern search algorithm, Knuth Morris Pratt, Wikipedia, http://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm