2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
INDIA

Overcoming the roadblocks in Display Port Automotive Extensions verification

Thenmozhy Kaliyamurthy, Principal Engineer, Synopsys India Pvt Ltd
(thenu@synopsys.com)

Meenakshy Ramachandran, Sr Staff Engineer, Synopsys India Pvt Ltd
(meenakshy.ramachandran@synopsys.com)

Pankaj Sharma, Sr Manager, Synopsys India Pvt Ltd
(pankajs@synopsys.com)

Nusrat Ali, Sr Director, Synopsys India Pvt Ltd
(nusrat@synopsys.com)

***Abstract*-Modern automotive displays use DisplayPort (DP) or Embedded DisplayPort (eDP) to carry video data from the central CPU to the displays. VESA's DisplayPort Automotive Extensions (DP AE) protocol adds automotive-grade functional safety and security to the existing DP and eDP standards. Chip manufactures have already started adopting DP AE for designing chipsets that will be part of future vehicles. Thus, there arises a need for an optimized testbench to ensure product quality and shorter time to market. This paper discusses the art of constructing a scalable testbench to mitigate the verification challenges in DP AE by leveraging existing DP testbench.**

***Keywords*-DisplayPort Automotive Extensions (DP AE), automotive displays, Functional Safety (FuSa), security, testbench, verification**

I. Introduction to DP AE

With the unprecedented advancements in digitalization, automotive displays have become the key interface between the driver and the vehicle. These displays provide a plethora of information including navigation, information, infotainment, driver assistance systems and others. With the demand for bigger and higher-resolution in-vehicle displays, Display Port (DP) is the undisputed choice for carrying video data to the displays, owing to its high bandwidth and support for Multi-Stream Transport (MST) feature (multiple displays to be connected to a single DP source port).

VESA's new DisplayPort Automotive Extensions (DP AE) specification protocol adds automotive-grade functional safety (FuSa) protocol and security protocol to the existing DisplayPort (DP) and embedded DisplayPort (eDP) standards. It follows ISO26262, an international standard for functional safety in the automotive industry. DP AE ensures the safety and security of the critical information from automotive displays. It supports four configuration profiles, ranging from basic profile to advanced profile as listed in Table 1. The basic functional safety profile performs cyclic redundancy check (CRC) for every video frame, which can detect frame drops or frame repetition. The more advanced profiles are optional but nevertheless highly recommended. They provide functional safety for the command-and-control data channel. Additionally, they support encryption of the command-and-control channel messages to prevent hackers from reading and tampering any sort of information pertaining to the displays.

To ensure backward compatibility with legacy DP and eDP systems, the changes brought in by DP AE are limited to a new secondary data packet (SDP) type called AE_SDP, new set of register definitions, new types of aux client messages and a security layer over the aux client messages. It uses CRC to detect data transfer errors, frame counter techniques to detect frame loss, and timeout monitoring to detect frozen or lost frames. It also introduces features such as Regions-of-Interest (ROI) and Superframes for enhanced functional safety of active frames. MAC generated and verified by AES-GCM is used for data integrity of both frames and the DP_AUX messages. It provides mechanisms for soft failure recovery and hard failure escalations. It supports Security Protocol & Data Model (SPDM) authentication support to establish a secure link between devices.

TABLE I
DP AE CONFIGURATION PROFILES

Configuration Profile	Features Supported	Purpose
Profile 0 **Basic FuSa**	Basic FuSa support via direct DPCD register access.	Performs cyclic redundancy check (CRC) for every video frame, which can detect frame drops or frame repetition.
Profile 1 **Basic FuSa**	Profile 0 + FuSa for DP_AUX message client 16-bit CRC validation.	Functional safety for the command-and-control data channel.
Profile 2 **Advanced FuSa (Basic Ctrl Plane Security)**	Profile 1 + advanced security for DP_AUX messages.	Encryption of the command-and-control channel messages to prevent hackers from reading and tampering display related information.
Profile 3 **FuSa with Enhanced Security**	Profile 2 + authentication and integrity for VESA data plane.	Prevents unwarranted tampering with the video data or display.

II. KEY VERIFICATION CHALLENGES

The verification of automotive displays is indispensable because they provide safety-critical information. Delayed, lost or frozen frames might compromise functional safety and can lead to catastrophic events. The display should also be secure and resistant to cyber-attacks and cyber breaches. A highly efficient testbench is required to inject different errors and simulate the attack scenarios to validate the robustness of the design. Since the DP AE changes are restricted to the high-level protocol layer of DP, it should be ensured that the integration of the new AE testbench framework into the existing DP testbench is smooth. Each scenario that is created needs to be validated for topologies such as Source, Sink and Branch devices. It also needs to be validated for both DP and eDP protocols.

The main challenge would be reducing the high simulation time for such scenarios because many frames are required to realize the scenarios. For example, a minimum of four frames are required to see a transition from non-FuSa protected state to FuSa protected state. Many such transitions are required to verify the system properly. Similarly, there are roll over frame counters of size 32 bits which can be verified only with certain hooks as it is highly unlikely to run a simulation with 2 ^32 frames. The specification also calls for computation intensive blocks like CRC and MAC calculators which perform bit wise operations for large data blocks leading to simulation speed degradation. Additionally, the specification has complex features such as ROI and Superframes. Moreover, the specification is being revised constantly to address the dynamic needs of industry.

The proposed testbench architecture handles these challenges effectively, allowing for faster and thorough verification. This paper also captures the results of deploying the proposed techniques to overcome the mentioned challenges.

III. ARCHITECTURE

The DP AE test suite is a plug and play environment with a lot of intricate and crucial features to help the users verify the safety and security of the design. Figure 1 depicts how the new DP AE blocks interact with the existing DP testbench. is for DP Main Link (ML). Similar changes are applicable for the DP AUX channel to encrypt the aux client messages. The blocks can be broadly classified as follows:

A. *DP AE Sequences:* The extensive DP AE sequences cover all possible scenarios of the DP AE specification. They are fully random smart sequences which achieve various desirable scenarios with minimum number of frames. The sequences are also highly configurable, allowing users to easily create scenarios of their choice.

B. *Testcases:* Scaled or custom frames are chosen to save simulation time as the number of frames required to realize DP AE scenarios are more. The testbench also has options to bypass the standard procedures like initialization, discovery, enumeration, authentication etc. which do not contribute to the validation of the core scenario. This eliminates the redundancy in validation, thus accelerating the simulation. The same scenario can be re-used across multiple topologies and modes via a run-time switch.

C. *DP AE Configuration:* This encapsulates all DP AE related configuration in a single configuration file.

D. *CRC generator:* Since the bit-by-bit algorithm is quite slow and inefficient for larger data, the lookup table-based CRC algorithm is used to speed up the execution.

E. *MAC generator:* The DP AE specification recommends the standard NIST SP800-38D for the computation of AES-GMAC, which is used to protect the frame information. The testbench re-uses a C code implementation of AES-GMAC using System Verilog DPI. AES-GMAC is a standard algorithm already in use in many protocols like PCIe, Ethernet etc. Since the DUT designers would possibly re-use these well-verified blocks and might want to skip the extensive verification of GMAC, the testbench provides an option for the user to bypass the MAC computations.

F. *ROI generator:* Certain regions of the display need additional protection. The ROI generator in the testbench configuration block generates these Regions of Interest (ROIs), of varying widths and heights. The intelligence in the code also takes care that the ROIs do not overlap with each other beyond the maximum of 4 ROI layers.

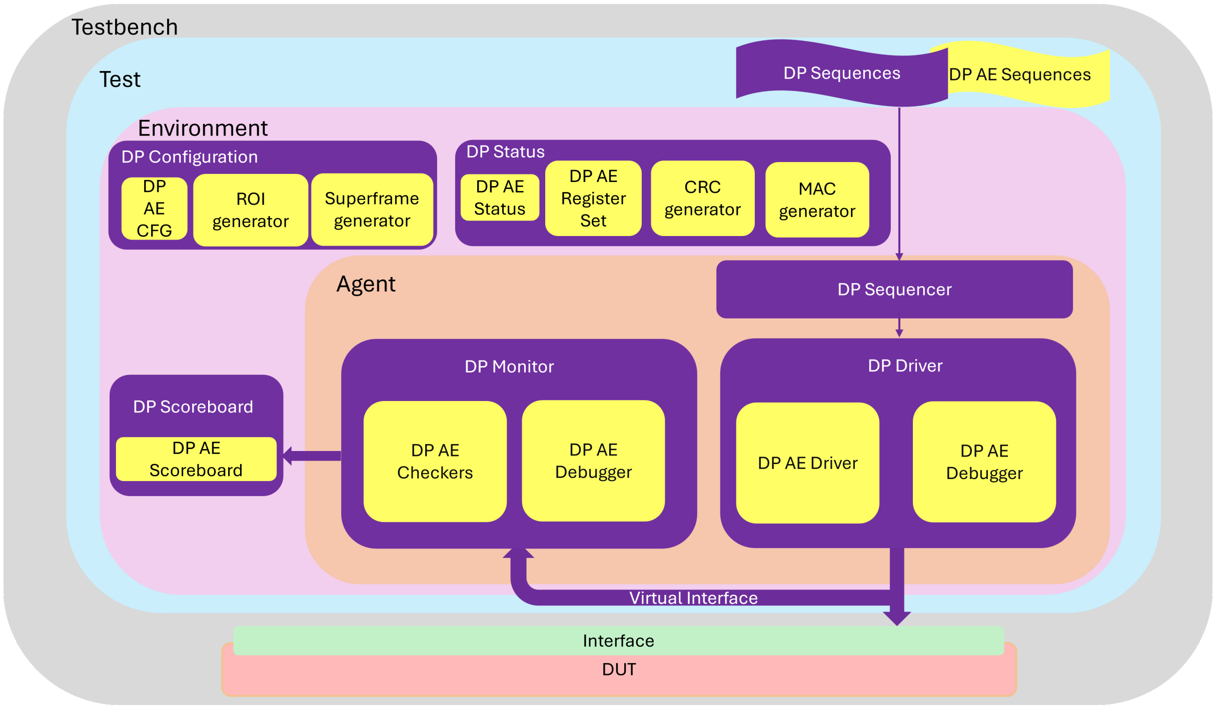


Figure 1. Architecture Diagram

🟪 Existing DP blocks in the testbench

🟨 New DP AE blocks in the testbench

G. *Superframe generator:* Supertram supports the transmission of multiple subframes. Abiding by all the rules of the specification, the Superframe generator prepares different subframes within a single frame, which shall be sent to different Sink devices.

H. *DP AE status:* The status variables related to DP AE feature can be accessed by various modules in the testbench.

I. *DP AE driver:* The DP AE driver is responsible for the generation and transmission of the newly added AE SDP, which carries the FuSa and security information.

J. *DP AE checkers:* The monitor performs extensive checks on the received AE SDP, with respect to both packet information and timing.

K. *DP AE debugger:* The testbench is equipped with advanced debugging techniques like dedicated interface signals, log messages and trace files. The detailed log messages can be enabled for any specific feature by passing a run-time switch. For example, in case of a CRC mismatch error, detailed debug messages at every stage of CRC computation can be obtained by just enabling a run-time switch.

L. *DP AE Scoreboard:* The DP AE scoreboard ensures that the transmitted AE_SDP transmitted matches the received AE_SDP.

IV. APPLICATION

The DP AE verification testbench will be a crucial tool in the verification of the functional safety and security of automotive displays. It will help to identify that the displays meet functional safety standards and work as expected, thus preventing accidents and saving lives. It will help in the early identification of the possible weaknesses in the security of the design that could be utilized by the hackers.

IV. SUMMARY WITH RESULTS, KEY TAKEAWAYS

The proposed updates in the DP testbench helps validate the DP AE specification thoroughly and quickly as users who have validated DP protocol with the DP testbench can validate DP AE protocol without any change in the integration.

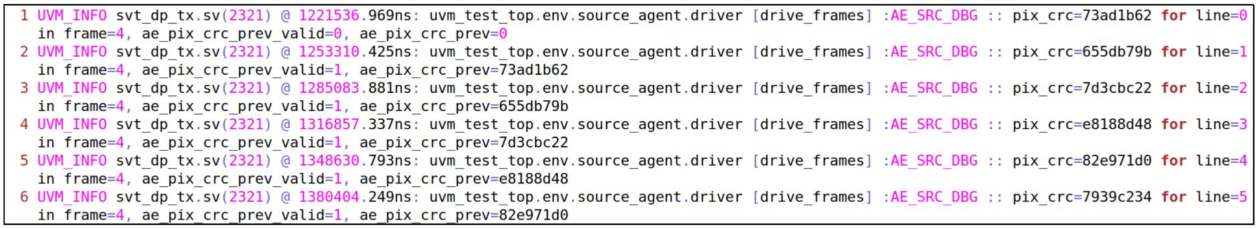
Table 2 illustrates the difference in the simulation time with different performance improving configurations provided by the DP AE testbench:

TABLE 2
PERFORMANCE IMPROVING CONFIGURATIONS

Configuration	Simulation Time (Config Disabled)	Simulation Time (Config Enabled)	Gain
Scaled down frame	16797s	553s	96%
Bypass initialization and standard procedures	487s	425s	12%

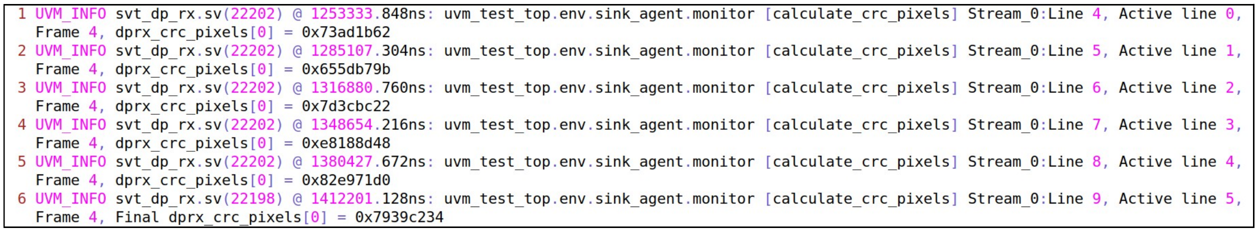
The DP AE debugger is equipped with impressive debug options as demonstrated below:

A. *Detailed debug messages:* Figure 2 and Figure 3 show the detailed log messages from the DP AE CRC calculator from the Source and Sink respectively. These messages are enabled by passing a command line argument. These command line options are available for all the major DP AE features.



```
UVM_INFO svt_dp_tx.sv(2321) @ 1221536.969ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=73ad1b62 for line=0
  in frame=4, ae_pix_crc_prev_valid=0, ae_pix_crc_prev=0
UVM_INFO svt_dp_tx.sv(2321) @ 1253310.425ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=655db79b for line=1
  in frame=4, ae_pix_crc_prev_valid=1, ae_pix_crc_prev=73ad1b62
UVM_INFO svt_dp_tx.sv(2321) @ 1285083.881ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=7d3cbc22 for line=2
  in frame=4, ae_pix_crc_prev_valid=1, ae_pix_crc_prev=655db79b
UVM_INFO svt_dp_tx.sv(2321) @ 1316857.337ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=e8188d48 for line=3
  in frame=4, ae_pix_crc_prev_valid=1, ae_pix_crc_prev=7d3cbc22
UVM_INFO svt_dp_tx.sv(2321) @ 1348630.793ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=82e971d0 for line=4
  in frame=4, ae_pix_crc_prev_valid=1, ae_pix_crc_prev=e8188d48
UVM_INFO svt_dp_tx.sv(2321) @ 1380404.249ns: uvm_test_top.env.source_agent.driver [drive_frames] :AE_SRC_DBG :: pix_crc=7939c234 for line=5
  in frame=4, ae_pix_crc_prev_valid=1, ae_pix_crc_prev=82e971d0
```

Figure 2. Debug messages from DP AE Source



```
UVM_INFO svt_dp_rx.sv(22202) @ 1253333.848ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 4, Active line 0,
  Frame 4, dprx_crc_pixels[0] = 0x73ad1b62
UVM_INFO svt_dp_rx.sv(22202) @ 1285107.304ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 5, Active line 1,
  Frame 4, dprx_crc_pixels[0] = 0x655db79b
UVM_INFO svt_dp_rx.sv(22202) @ 1316880.760ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 6, Active line 2,
  Frame 4, dprx_crc_pixels[0] = 0x7d3cbc22
UVM_INFO svt_dp_rx.sv(22202) @ 1348654.216ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 7, Active line 3,
  Frame 4, dprx_crc_pixels[0] = 0xe8188d48
UVM_INFO svt_dp_rx.sv(22202) @ 1380427.672ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 8, Active line 4,
  Frame 4, dprx_crc_pixels[0] = 0x82e971d0
UVM_INFO svt_dp_rx.sv(22198) @ 1412201.128ns: uvm_test_top.env.sink_agent.monitor [calculate_crc_pixels] Stream_0:Line 9, Active line 5,
  Frame 4, Final dprx_crc_pixels[0] = 0x7939c234
```

Figure 3. Debug messages from DP AE Sink

B. *Debug ports:* Figure 4 shows a few debug ports from a simulation where both FuSa and security are enabled. The debug ports for both CRC and MAC are shown per frame.

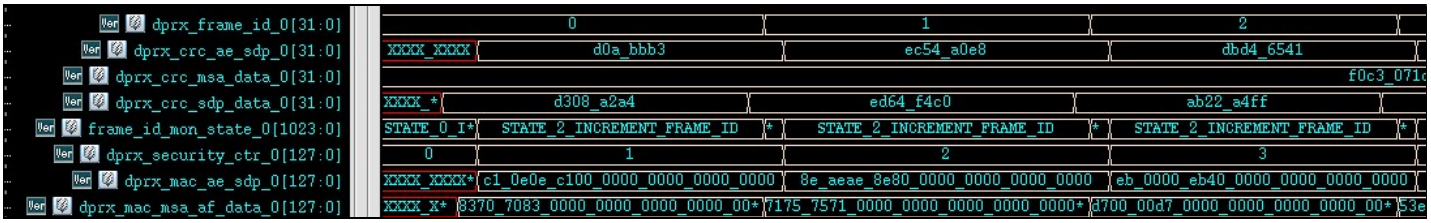
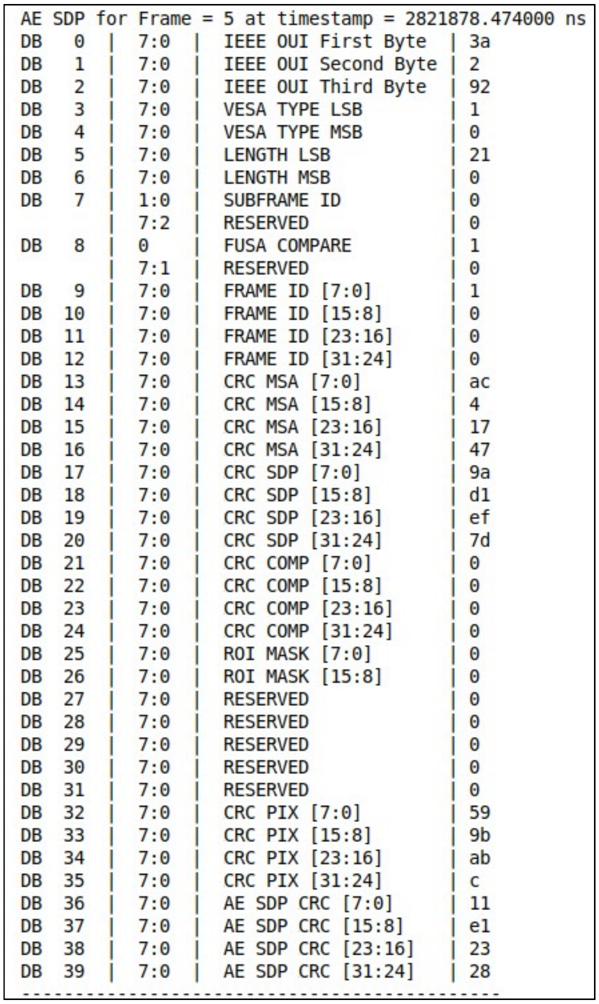

dprx_frame_id_0[31:0]
dprx_crc_ae_sdp_0[31:0]
dprx_crc_msa_data_0[31:0]
dprx_crc_sdp_data_0[31:0]
frame_id_mon_state_0[1023:0]
dprx_security_ctr_0[127:0]
dprx_mac_ae_sdp_0[127:0]
dprx_mac_msa_af_data_0[127:0]
0
1
2
XXXX_XXXX
d0a_bbb3
ec54_a0e8
dbd4_6541
f0c3_071c
XXXX_*
d308_a2a4
ed64_f4c0
ab22_a4ff
STATE_0_I*
STATE_2_INCREMENT_FRAME_ID
*
STATE_2_INCREMENT_FRAME_ID
*
STATE_2_INCREMENT_FRAME_ID
*
0
1
2
3
XXXX_XXXX*
c1_0e0e_c100_0000_0000_0000_0000
8e_aeae_8e80_0000_0000_0000_0000
eb_0000_eb40_0000_0000_0000_0000
XXXX_X*
8370_7083_0000_0000_0000_0000_00*
7175_7571_0000_0000_0000_0000_0000*
d700_00d7_0000_0000_0000_0000_00*
53e

Figure 4. Debug ports

C. *Trace files:* Figure 5 shows trace files that are generated for debugging any AE_SDP packet related issues.



```
AE SDP for Frame = 5 at timestamp = 2821878.474000 ns
DB   0  |  7:0  |  IEEE OUI First Byte  | 3a
DB   1  |  7:0  |  IEEE OUI Second Byte | 2
DB   2  |  7:0  |  IEEE OUI Third Byte  | 92
DB   3  |  7:0  |  VESA TYPE LSB        | 1
DB   4  |  7:0  |  VESA TYPE MSB        | 0
DB   5  |  7:0  |  LENGTH LSB           | 21
DB   6  |  7:0  |  LENGTH MSB           | 0
DB   7  |  1:0  |  SUBFRAME ID          | 0
        |  7:2  |  RESERVED             | 0
DB   8  |  0    |  FUSA COMPARE         | 1
        |  7:1  |  RESERVED             | 0
DB   9  |  7:0  |  FRAME ID [7:0]       | 1
DB  10  |  7:0  |  FRAME ID [15:8]      | 0
DB  11  |  7:0  |  FRAME ID [23:16]     | 0
DB  12  |  7:0  |  FRAME ID [31:24]     | 0
DB  13  |  7:0  |  CRC MSA [7:0]        | ac
DB  14  |  7:0  |  CRC MSA [15:8]       | 4
DB  15  |  7:0  |  CRC MSA [23:16]      | 17
DB  16  |  7:0  |  CRC MSA [31:24]      | 47
DB  17  |  7:0  |  CRC SDP [7:0]        | 9a
DB  18  |  7:0  |  CRC SDP [15:8]       | d1
DB  19  |  7:0  |  CRC SDP [23:16]      | ef
DB  20  |  7:0  |  CRC SDP [31:24]      | 7d
DB  21  |  7:0  |  CRC COMP [7:0]       | 0
DB  22  |  7:0  |  CRC COMP [15:8]      | 0
DB  23  |  7:0  |  CRC COMP [23:16]     | 0
DB  24  |  7:0  |  CRC COMP [31:24]     | 0
DB  25  |  7:0  |  ROI MASK [7:0]       | 0
DB  26  |  7:0  |  ROI MASK [15:8]      | 0
DB  27  |  7:0  |  RESERVED             | 0
DB  28  |  7:0  |  RESERVED             | 0
DB  29  |  7:0  |  RESERVED             | 0
DB  30  |  7:0  |  RESERVED             | 0
DB  31  |  7:0  |  RESERVED             | 0
DB  32  |  7:0  |  CRC PIX [7:0]        | 59
DB  33  |  7:0  |  CRC PIX [15:8]       | 9b
DB  34  |  7:0  |  CRC PIX [23:16]      | ab
DB  35  |  7:0  |  CRC PIX [31:24]      | c
DB  36  |  7:0  |  AE SDP CRC [7:0]     | 11
DB  37  |  7:0  |  AE SDP CRC [15:8]    | e1
DB  38  |  7:0  |  AE SDP CRC [23:16]   | 23
DB  39  |  7:0  |  AE SDP CRC [31:24]   | 28
---------------------------------------------
```

Figure 5. Trace files.

IV. CONCLUSION

In this paper, we have showcased a systematic approach to develop a DP AE testbench by enhancing the existing DP testbench. We have also demonstrated various strategies to accelerate the simulation. The advanced debugging capabilities helped us focus on the real aim of unearthing critical bugs in the design that could compromise safety and security. The run time switches used in the scenario generation helped us achieve faster coverage closure which in turn helped in the early closure of the DP AE verification activity.

References

[1] www.vesa.org, "VESA DisplayPort (DP) Automotive Extension Services," Version 1.0, July 24, 2025.
[2] www.vesa.org, "VESA DisplayPort (DP) Standard", Version 2.1a, 18 December 2023.
[3] www.vesa.org, "VESA Embedded DisplayPort (eDP) Standard", Version 1.5a, February 27, 2023.