Efficient Verification Framework for Audio/Video Interfaces

Noha Shaarawy
Agenda

- Introduction
- Proposed Verification Framework.
- HDMI 2.0 Verification Environment.
- Verification Environment for an HDMI TX and RX.
- Experimental Results.
Introduction
Target

Verifying such interfaces is a big Challenge.

Minimize Development Time
Target

Verifying such interfaces is a big Challenge.

Minimize Development Time

Reusability
Target

Verifying such interfaces is a big Challenge.

Minimize Development Time

Reusability

Configurable
Proposed Verification Framework
Proposed Verification Framework
Proposed Verification Framework

VIP components is divided into six regions:

Common Regions between Protocols:
- Hot Plug
- Data
- Video
- Streaming

Different Regions between Protocols:
- Auxiliary Channel
- Control
HDMI 2.0 Verification Environment

- HDMI 2.0 receiver and transmitter VIPs are developed based on:

![](sysverilog.png) ![UVM.png]
HDMI 2.0 Verification Environment

- HDMI 2.0 receiver and transmitter VIPs are developed based on FrameWork.
HDMI 2.0 Source VIP

DUT Source

Deframer

Pixel Decoder

Video Agent

Audio Depacketizer

I2S Tx

Audio Agent

Data Processor

Auxiliary Depacketizer

S/PDIF Tx

I2C Slave

HotPlug Responser

Configuration & Control

EDID

SCDC

Authentication

Auxiliary region

Stream region

HotPlug region

Video region

Data region

Control region

DUT
HDMI 2.0 Source UVM Environment
HDMI 2.0 Sink VIP

Audio Agent → I2S Rx
S/PDIF Rx → Audio Packetizer
Configuration & Control → Auxiliary Packetizer

Video Agent → Pixel Encoder

Streaming

I2C Master
HotPlug Stimulus

SCDC
Authentication

Auxiliary region
HotPlug region
Data region
Stream region
Video region
Control region

DUT Sink

DUT
HDMI 2.0 Sink UVM Environment
## Reusability

<table>
<thead>
<tr>
<th>Block Name</th>
<th>HDMI</th>
<th>DisplayPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Processor</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Video Agent</td>
<td></td>
<td>Reusable</td>
</tr>
<tr>
<td>Audio Processor</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Audio Agent</td>
<td></td>
<td>Reusable</td>
</tr>
<tr>
<td>I2S Audio Interface</td>
<td></td>
<td>Reusable</td>
</tr>
<tr>
<td>SPDIF Audio Interface</td>
<td></td>
<td>Reusable</td>
</tr>
<tr>
<td>I2C Interface</td>
<td></td>
<td>Reusable</td>
</tr>
<tr>
<td>Framer</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Deframer</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Authentication</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Auxiliary Processor</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>Controller</td>
<td>Customized</td>
<td>Customized</td>
</tr>
<tr>
<td>HotPlug Responser</td>
<td></td>
<td>Reusable</td>
</tr>
</tbody>
</table>
Experimental Results

HDMI Transmitter coverage result
- 115 test cases.
- Total coverage = 96.50%

HDMI Receiver coverage result
- 90 test cases.
- Total coverage = 94.00%
Experimental Results

The performance results for 10 frames sent to HDMI receiver.

<table>
<thead>
<tr>
<th>Video Format</th>
<th>CPU Time (in Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>720*480</td>
<td>349</td>
</tr>
<tr>
<td>2560*1080</td>
<td>2305</td>
</tr>
<tr>
<td>4096*2160 (4K)</td>
<td>8908</td>
</tr>
</tbody>
</table>
Questions