Safety-Critical Extension

v1.1

Frank Brill (Cadence) & Jesse Villareal (TI)
Introduction

- **Objective:**
  - *Facilitating* implementation of OpenVX into safety-critical applications
    - Does not guarantee implementation is safe.

- Software written for safety-critical applications requires rigorous demands on the deployment software and development processes (e.g., ISO 26262)

- V-Model for safety-critical development processes
Requirements

- Annotated OpenVX spec with Functional Requirement tag numbers
  - Each requirement is identified with a [R#####] tag

\[
\text{vxCreateRemap()}
\]

\[
\text{vx\_remap VX\_API\_CALL vxCreateRemap (}
\begin{align*}
& \text{vx\_context context}, \\
& \text{vx\_uint32 src\_width}, \\
& \text{vx\_uint32 src\_height}, \\
& \text{vx\_uint32 dst\_width}, \\
& \text{vx\_uint32 dst\_height })
\end{align*}
\]

Creates a remap table object \([R01166]\).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>The reference to the overall context ([R01167]).</td>
</tr>
<tr>
<td>src_width</td>
<td>Width of the source image in pixel ([R01168]).</td>
</tr>
<tr>
<td>src_height</td>
<td>Height of the source image in pixels ([R01169]).</td>
</tr>
<tr>
<td>dst_width</td>
<td>Width of the destination image in pixels ([R01170]).</td>
</tr>
<tr>
<td>dst_height</td>
<td>Height of the destination image in pixels ([R01171]).</td>
</tr>
</tbody>
</table>

- Aids in traceability from requirements, to design, to implementation, to testing
- Each requirement must be tested
Implementation

• Updated OpenVX headers to be [MISRA-C](#) compliant
  • Change enums to #defines
  • Changes to `vx_pixel_value_t`
    • This is a union, with no variant tag—it is not portable across architectures, and requires a variant tag to specify the type of the contents

• Revision of some API’s
  • Those that pass pointers to objects with no defined size
Development and Deployment Feature Sets

### Development Environment
- Create and verify OpenVX graph
- Export all the objects that needs access during deployment
- Release all objects

### Deployment Environment
- Create context
- Import objects from binary
- Graph execution
- Release all objects
Development and Deployment Feature Sets

- **Needed for deployment:**
  - Import graphs
  - Provide input data objects
  - Process graphs
  - Access output data objects

- **Not needed for deployment (Development only):**
  - vxu immediate kernels
  - Graph construction
    - All node creation APIs for each of the vision kernels
  - Graph verification
    - System mapping and optimization decisions that typically occur during graph verification may be decided and fixed in the development feature set
  - Logging and performance features
  - Graph export functions
OpenVX Evolution

OpenVX 1.0 → OpenVX 1.1

OpenVX 1.1 → OpenVX SC

OpenVX SC → OpenVX 1.2

OpenVX 1.2 → OpenVX 1.3
Summary

- Annotated OpenVX spec with Functional Requirement tag numbers
- Made header files MISRA-C compliant
- API definition revisions
- Defined Development vs Deployment Feature sets
- Added Import Export Extension as part of SC specification
- Plan to merge OpenVX SC into main OpenVX 1.3 spec
  - No longer maintain a separate spec