UXL
Unified Acceleration Foundation

Safety Critical Special Interest Group

2nd session, Q2

May 29th 2024

This Meeting is Recorded

Any objections?

UXL Foundation Antitrust Policy

Members acknowledge that they may compete with one another in various lines of business and that it is therefore imperative that they and their respective representatives act in a manner that does not violate any applicable **antitrust laws** and **regulations**. Each Member may have similar agreements with others. Each Member may design, develop, manufacture, acquire or market competitive deliverables, products and services, and conduct its business, in whatever way it chooses.

No Member is obligated to announce or market any products or services associated with or based on the Project. Without limiting the generality of the foregoing, the Members agree not to have any discussion relating to any **product pricing**, **methods** or **channels of product distribution**, **division of markets**, **allocation of customers** or any other topic that should not be discussed among competitors.

If you have questions about these matters, please contact your company counsel or counsel to the Joint Development Foundation, DLA Piper.

Code of Conduct

All activities and working spaces of the UXL Foundation are subject to its code of conduct, located at https://www.contributor-covenant.org/version/1/0/0/code-of-conduct/

If you have a question about the CoC, please contact operations@uxlfoundation.org

For trademark and other policies, please visit https://jointdevelopment.org/policies/

Safety Critical SIG

Aim:

Enable/accelerate integration of UXL projects into safety critical systems

The Special Interest Groups are open to anyone and bring together industry experts to help guide the oneAPI specification and open-source projects.

SIG activities include the following:

- *Open technical discussions relevant to specific technologies and the oneAPI specification*
- *Helping review proposals*
- *Presentation of proposals*

Agenda

- UXL Foundation processes – update
- DPC++ upstreaming to LLVM
- Presentation by Verena: Integration of UXL Projects into SC System - Tasks
- Discussion: Next steps

Round of Introductions

- Why did you join the SIG?
- What part are you/your company interested in?
- What do you hope to get out of the SIG?

UXL Webinar

Recording: https://www.youtube.com/watch?v=OQ2OP32URLw

UXL Processes

Vincent Mendonca

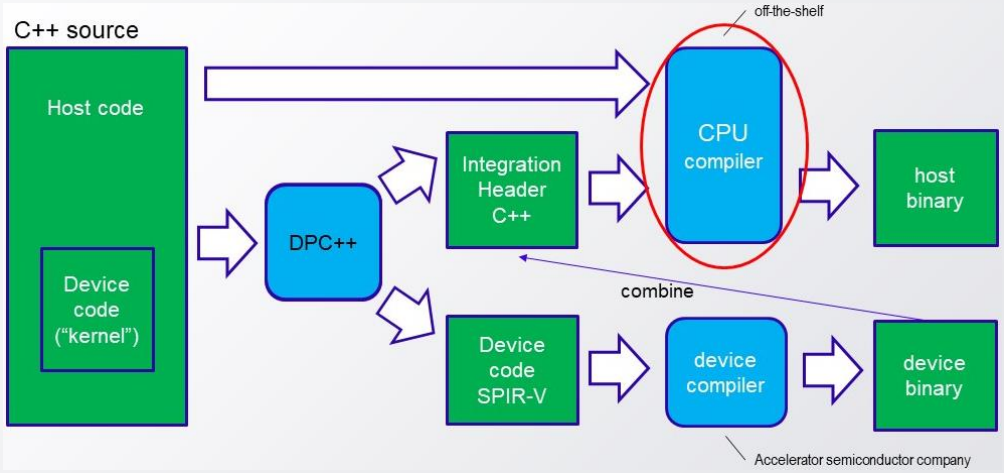
- Quality Manager at Codeplay
- Experience in automotive
- ISO 9001 auditor, ASPICE assessor, ISO 26262 trained

- Working with members to set up best-practice processes for UXL
- See open tickets: https://github.com/orgs/uxlfoundation/projects/5/views/1
- In the future: possible gap analysis for safety?

DPC++ in LLVM

- DPC++ being upstreamed into LLVM
- DPC++ relies on Integration Header to support arbitrary host compiler
- LLVM does not want to support Integration Header:

"Use of integration headers for support of third-party host compilers is ok if significant motivation for supporting such compilers becomes apparent in the future. Such support will not be added now and thus an integration header solution is not needed now."



If you believe this use case is important for you, join the LLVM discussion!

https://discourse.llvm.org/t/rfc-sycl-host-compiler-integration-header-and-footer/74081/68

Integration of UXL Projects into SC System - Tasks

Verena Beckham, VP of Safety Engineering

29th May 2024

Background

- Results of an analysis I performed
- Entirely hypothetical

Assumptions

- Based on a target of ISO 26262 ASIL B
- Reuse DPC++ & UXL Foundation libraries
- Qualified C++17 CPU compiler available
- Optional: Qualified SPIR-V device compiler available
- Qualified (subset of) C++ standard library available
- Just a starting point

Steps Towards SC System Integration

Compiler

- Translates application into host/device assembly
- Runs on development system (cross-compiling)
- Used during development - qualified

Runtime

- Runs alongside application
- Schedules kernels on accelerator(s)
- Runs on device - certified

Libraries

- Optimised for device
- Runs on device
- Qualified or certified? Unclear

Compiler – DPC++ (or LLVM, once upstreamed)

- Qualify as C++ compiler
  - Test suite available from 3rd party
  - Qualification framework available from 3rd party
- Define requirements (improve SYCL SC spec)
- Map conformance tests to requirements
- Extend conformance testsuite
  - 100% requirements
- Draft safety manual

SYCL SC runtime (DPC++) – Code & Testing

- Branch from mainline
- Add SYCL SC features
- Remove/#ifdef out removed SYCL features
- Adapt to MISRA C++:2023
  - Draft exclusions document
- Measure code quality metrics (e.g. complexity)
  - Improve code quality
- Run additional static analysis tools?
- Document/Improve dev processes
- Collect/Write design documents
- Improve testing
  - Requirements coverage
  - Source coverage
  - Fault injection

Open-source proof-of-concept?

SYCL SC runtime – Integration & Safety

- Integrate with RTOS
- Integrate with scheduler
- Integrate with HW safety features?
- Perform safety analysis sessions
- Draft Safety Argument fragment

Low-level runtime

- Vulkan SC?
  - Add a new DPC++ backend
- Level 0?
  - Potentially make changes to support SYCL SC changes

UXL Foundation Libraries

- Port to SYCL SC
  - Different error management?
  - Different memory allocation?
- Port to device & optimise
- Define requirements (improve spec)
- Map testing to requirements
- Extend testing
  - 100% requirements
  - 100% source coverage
- Extend documentation (e.g. resource usage, error behaviour)
- Qualify? Certify?
  - Document dev processes

Extras

- Kernel coverage tool
- Tool for schedule amplification
- Random SYCL code generation

Disclaimers

A wee bit of legal

No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Task – Next Steps

- What do we want to achieve?
- What would be useful?
- Could be deliverables, could be discussion

Call to Action

What could you contribute?

- Use cases?
- Samples?
- Code?
- Experts?
  - E.g., FuSa experts to do analysis
  - E.g., Open-source experts to give advice
- Experience reports?
- Lessons Learnt from other groups (e.g. ELISA)?

Who are we missing?

- Companies/Individuals we should invite?
  - Participate
  - Speak
- Collaborations we should initiate?
- Conferences we should attend?

Thank you for attending!

- Minutes will be uploaded to https://github.com/uxlfoundation/foundation/tree/main/safety-critical
- Recording available at https://openprofile.dev/
- Contact me with agenda suggestions: verena@codeplay.com
- Forward invitations to others to join the SIG
- Continue discussion via mailing list: Safety-Critical-SIG@lists.uxlfoundation.org and Slack
- Join UXL on LinkedIn: https://www.linkedin.com/groups/14241252/

Join The UXL Foundation

UXL
Unified Acceleration Foundation

Steering Member
$20k*

- Seat on the Steering Committee
- Voting Rights
- Define the direction of the foundation

General Member
$5k*

- Working Group Voting Rights
- Influence Working Group direction
- Co-marketing

Contributor Member
$0

- Participate in Working Groups
- Contribute to the specification
- Contribute to the projects

membership@uxlfoundation.org

* plus Linux Foundation membership

www.uxlfoundation.org