CORE-V IDE

Roisin O'Keeffe, Ashling Microsystems

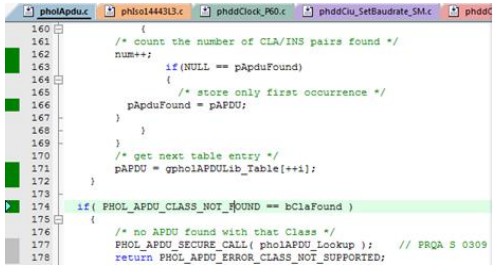
OSD Forum 2020
15th September 2020

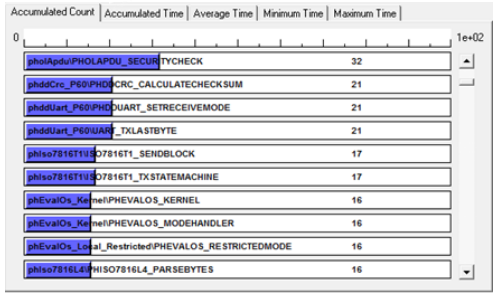
Agenda

- About Ashling
- Integrated Development Environments
  - CORE-V IDE
  - RiscFree IDE
- CORE-V Demo
  - How to build and debug a CORE-V example program within the IDE

Ashling Introduction

- Development tools for embedded processors and MCUs
- IDEs, Compilers, Debuggers, Hardware Debug and Trace Probes for a range of embedded architectures
- Engineering services for custom tools requirements
- Expertise in complex real-time debug tools including
  - high speed debug connections
  - high capacity real-time trace probes
  - code coverage analysis, profiling, performance analysis etc.

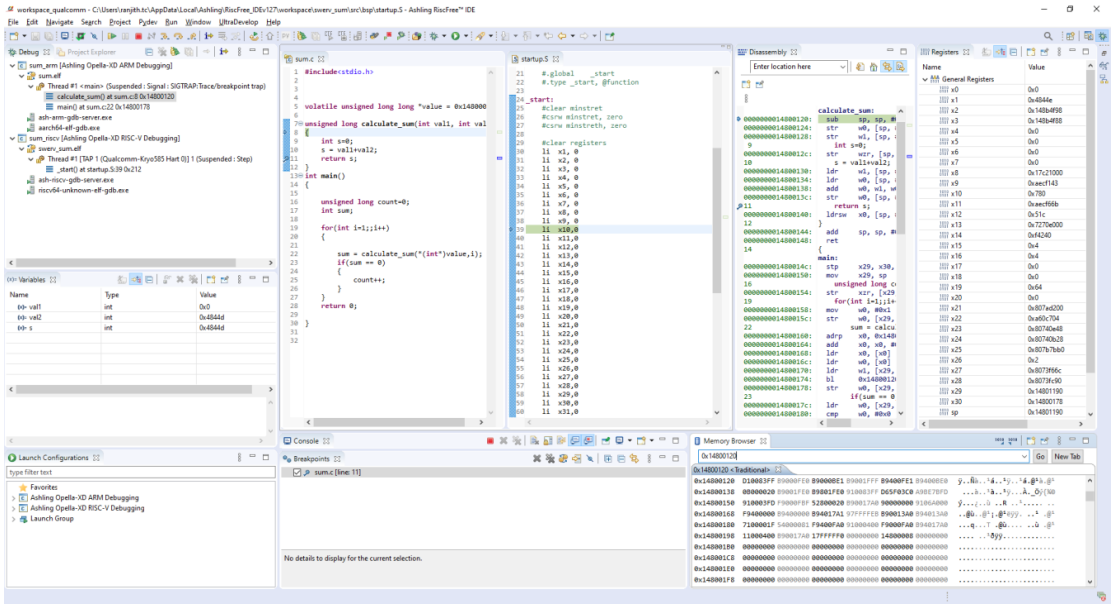




ASHLING
EMBEDDED TECHNOLOGY AND INNOVATION

CORE-V IDE

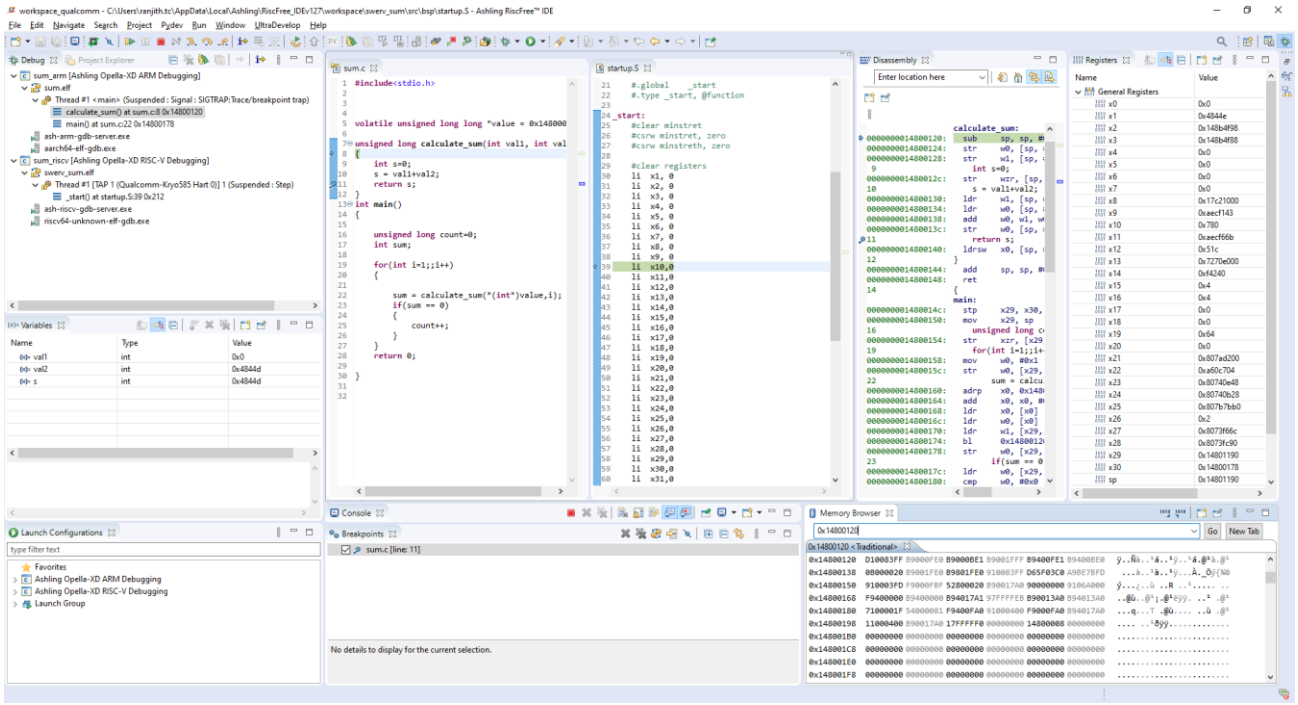
- ***CORE-V IDE*** is a freely available, open-source development environment created by the OpenHW group
- Eclipse based IDE for CORE-V development
- Includes the Compiler Toolchain for CORE-V provided by Embecosm
- OpenOCD Debug Support
- “Ready-to-run” examples for Digilent boards
- Getting started guide
- Available end September 2020



ASHLING
EMBEDDED TECHNOLOGY AND INNOVATION

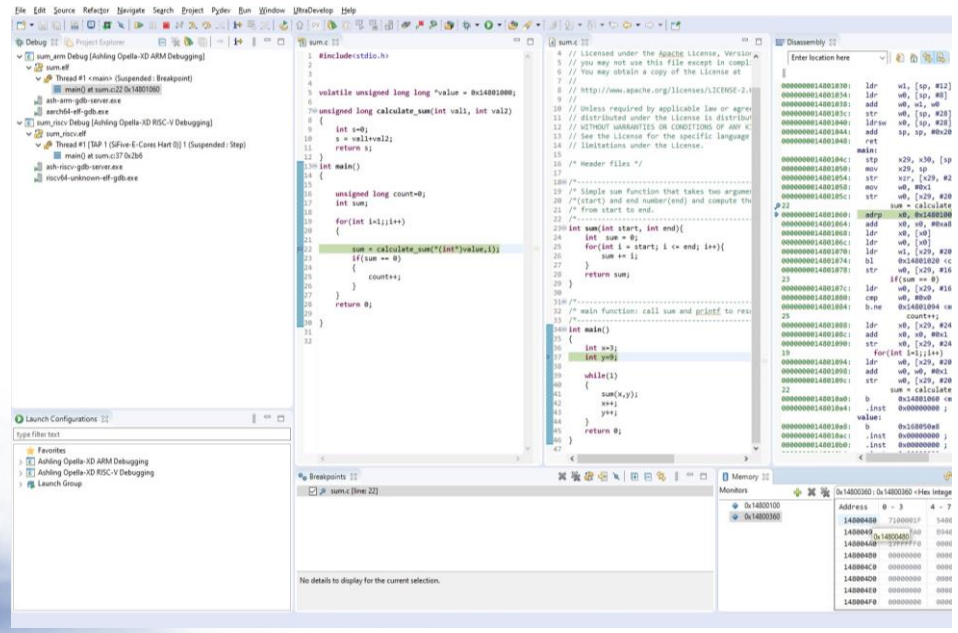
RiscFree IDE

- ***RiscFree™*** is Ashling's professional, commercial grade software development environment
- Supports a range of 32-bit and 64-bit devices
- Code-coverage analysis
- Profiling and Performance Analysis
- OS Debug support
- Multi-core Debug Support for homogeneous and heterogeneous debugging (e.g. RISC-V and non RISC-V)



ASHLING
EMBEDDED TECHNOLOGY AND INNOVATION

RiscFree support for OpenHW CORE-V



RiscFree Eclipse based IDE + Opella-XD JTAG Probe + Genesys2 FPGA Board

Professional, commercial grade software development and debug environment

CORE-V Demo by Rejeesh S.B.

- CORE-V Demo using Ashling's ***RiscFree*** + Opella-XD Probe + Genesys2 FPGA platform
- How to build a CORE-V example program
- How to launch a Debug session using Opella-XD connected to the Genesys2 board
- Run-time debug features including go/stop/step/breakpoints and target access