SoC-Now

Mini SoC Generator by MERL-UIT

Live at: socnow.software

An Open-Source Web Based RISC-V SoC Generator | WOSET’22

Agenda

1. Introduction
2. Features
3. Verification Paradigm
4. Architecture
5. Frameworks
6. Solution to the Industry Problems
7. Applications
8. Achievements
9. Demonstration
10. Contributors
11. Acknowledgement

Introduction

- In modern technological world every aspect of life is reliant on technology or "smart devices." And these devices are called “smart” because they contain a computing system inside called as processor or system on chip (SoC).
- Development of SoC is a quite long process. It can take months to build and develop a SoC from scratch.
- Following the design stage, testing, and verification of it adds another layer of stress.
- Designers like to reuse a verified component in their design(IPs) to save time.
- The open-source semiconductor industry is therefore in need of software that can quickly produce or design a SoC rather than taking the designer months to complete the build or design.

Introduction (cont.)

- Alternately, any designer can also generate a component with plug and play support.

- SoC-Now is an open-source web-based RISC-V ISA standardised SoC design solution in terms of generator that enables anyone to create a SoC with their own customised specifications and then further process that SoC to the Field Programmable Gate Array (FPGA) Emulation.

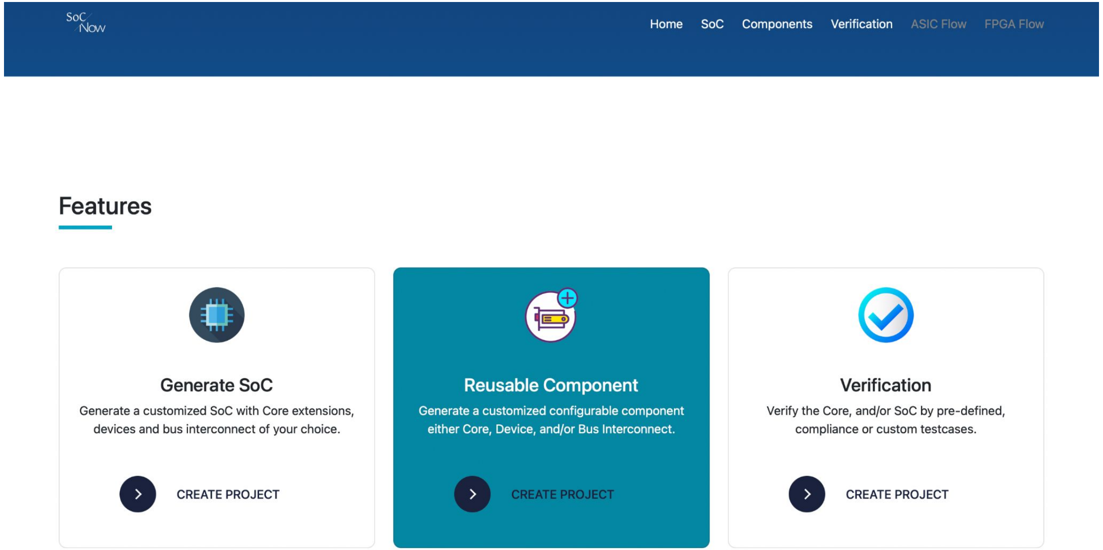
- Additionally, it provides the capability to produce any standalone, verified, and reusable SoC component (Core, Device, Bus) by means of generic interfaces.

- A completely automated Verification system is also included.

Features

SoCNow is roam around three main features which are.

1. Generate SoC
2. Reusable Component
3. Verification



Features

Generate SoC

Generate a customized SoC with Core extensions, devices and bus interconnect of your choice.

Reusable Component

Generate a customized configurable component either Core, Device, and/or Bus Interconnect.

Verification

Verify the Core, and/or SoC by pre-defined, compliance or custom testcases.

Generate SoC

This is the key feature of SoC-Now. User can easily generate a SoC by just select some checkboxes. User need to select the following:

- Core Extensions
- Devices
- Bus

After set the configurations, user can:

- Generate RTL
- Verify SoC
- Generate Bitstream

Reusable Component

- SoC-Now is designed in such a way so that user can easily reuse the components.
- Here user can generate RTL of each component and Integrate it in their own created SoC.
- We design all the components like core, Buses and Devices in a way so that it can be easily reusable.

Verification Paradigm

- The core was verified through compliance test. A set of tests that are executed on the core and the result is compared against the golden model.
- The SoC was verified by running directed tests written in riscv assembly. These tests were executed by simulating the SoC and on FPGA.
- The directed assembly tests verify the behaviour of different peripherals, buses the SoC as a whole.

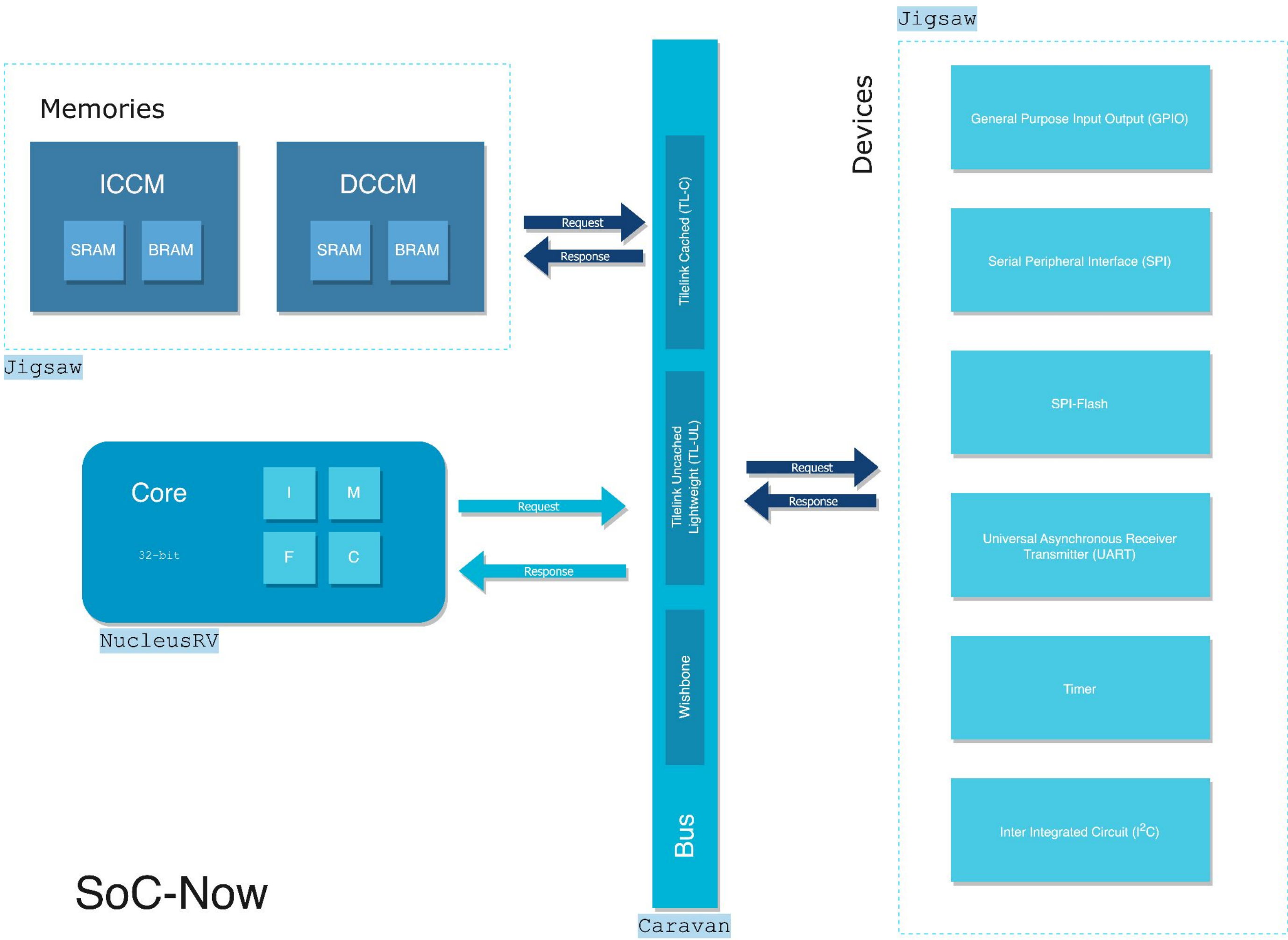
Architecture

- All the components including core, device(s), and bus interconnect are connected together by plug n play approach.
- This functionality is achieved by Generic Interfaces of Request and Response.
- This way any of the component can be picked up and easily connected with the System on Chip (SoC)
- Further on, after SoC Configuration and Generation, the Bitstream of SoC is also generated after asking the IOs mapping and program to be burned inside SoC.

Frameworks

The architecture of SoC-Now is composed of 3 In house CHISEL Frameworks .

1. Caravan
2. Jigsaw
3. NucleusRV


Memories
ICCM
SRAM
BRAM
DCCM
SRAM
BRAM
Jigsaw
Request
Response
Core
32-bit
I
M
F
C
NucleusRV
Request
Response
Tilelink Cached (TL-C)
Tilelink Uncached Lightweight (TL-UL)
Wishbone
Bus
Caravan
Request
Response
Devices
Jigsaw
General Purpose Input Output (GPIO)
Serial Peripheral Interface (SPI)
SPI-Flash
Universal Asynchronous Receiver Transmitter (UART)
Timer
Inter Integrated Circuit (I2C)

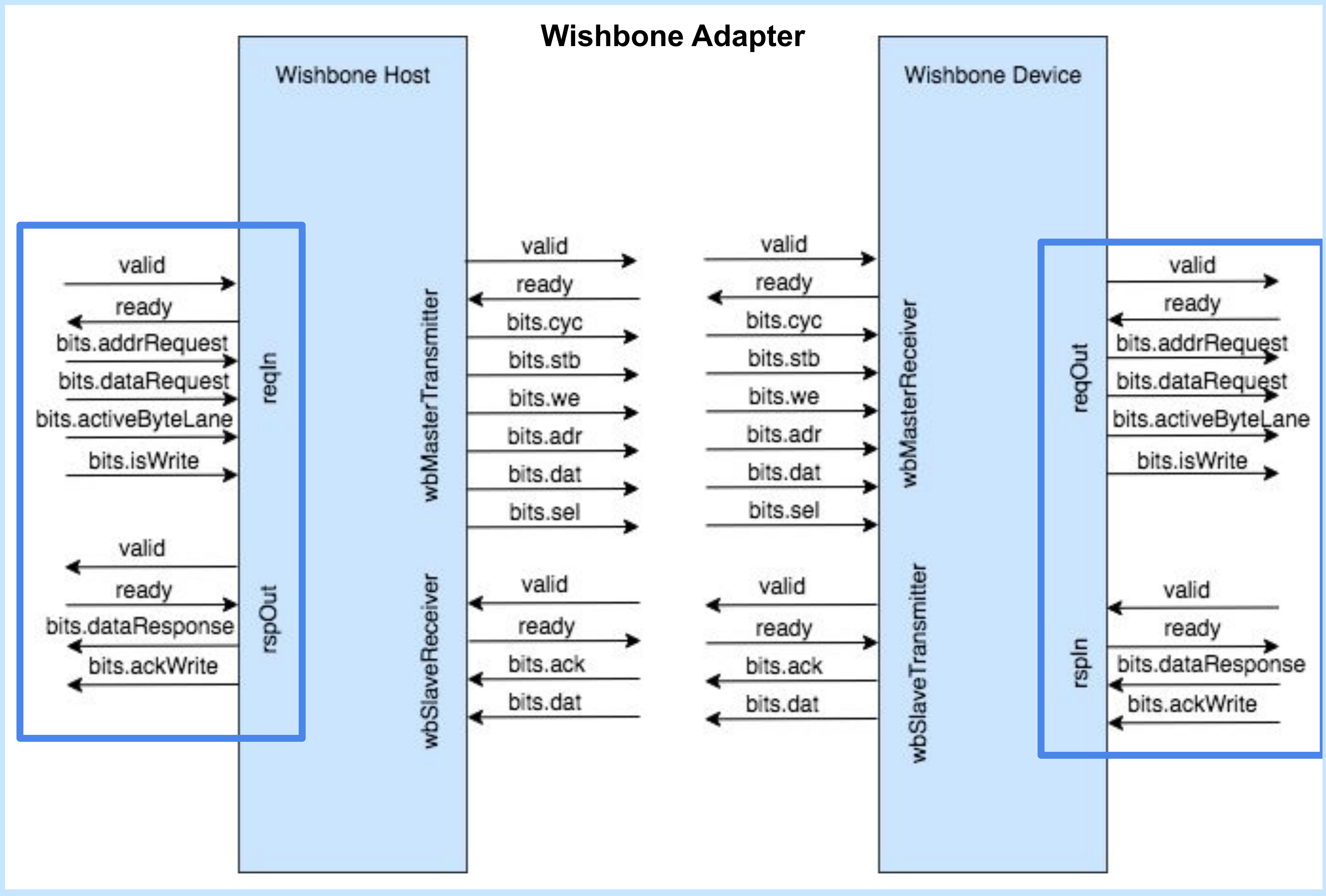
SoC-Now

Caravan

- A Framework developed in CHISEL for plug and play connection of Bus Interconnects.
- Bus Interconnects implemented in Caravan are:
  1. Wishbone
  2. Tilelink - Uncached Lightweight
  3. Tilelink - Cached
- Future Plans to have implementation of:
  1. AXI
  2. CXL

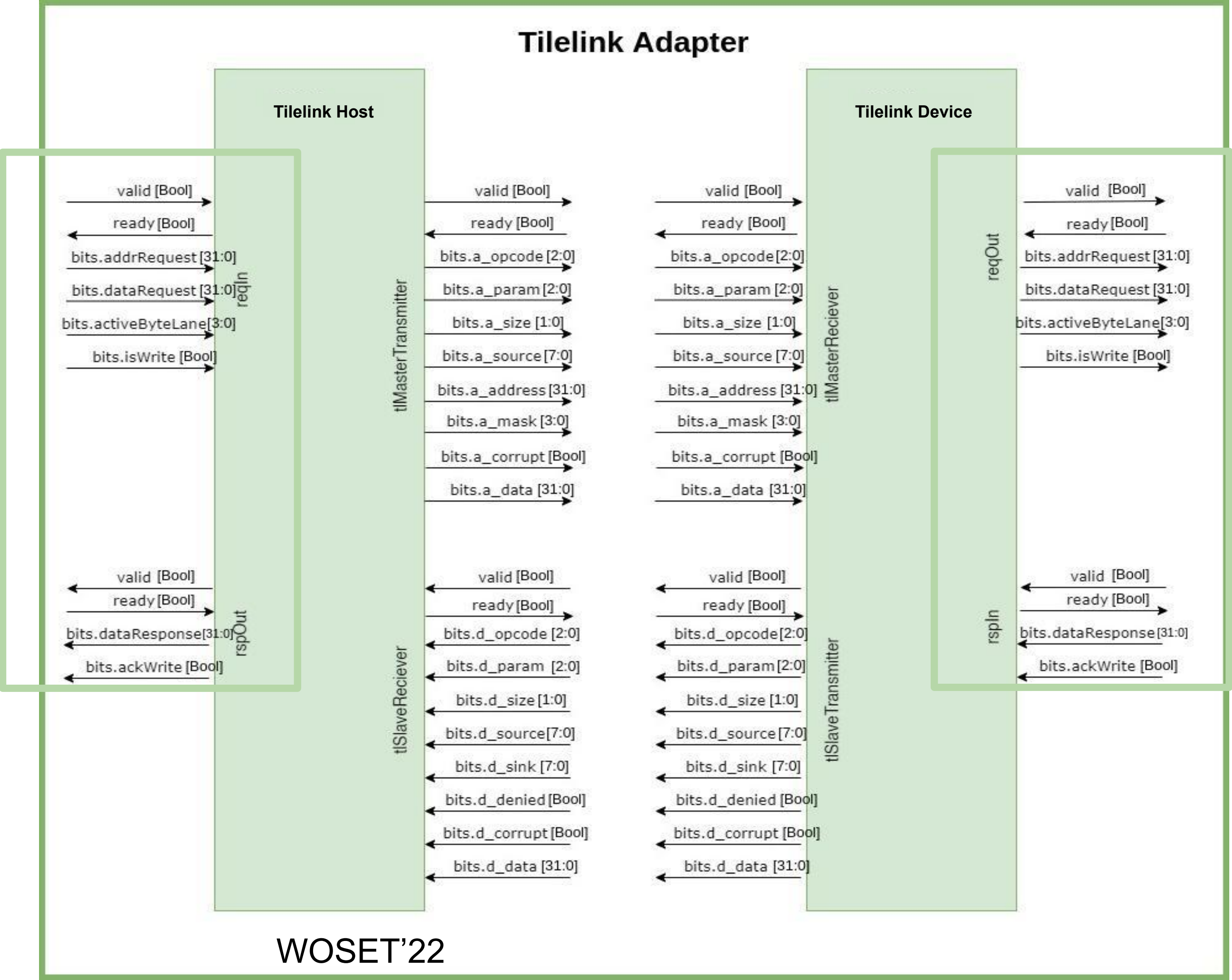
Caravan

Caravan provides generic Request and Response Interfaces to connect Buses



Caravan

Caravan provides generic Request and Response Interfaces to connect Buses



WOSET’22

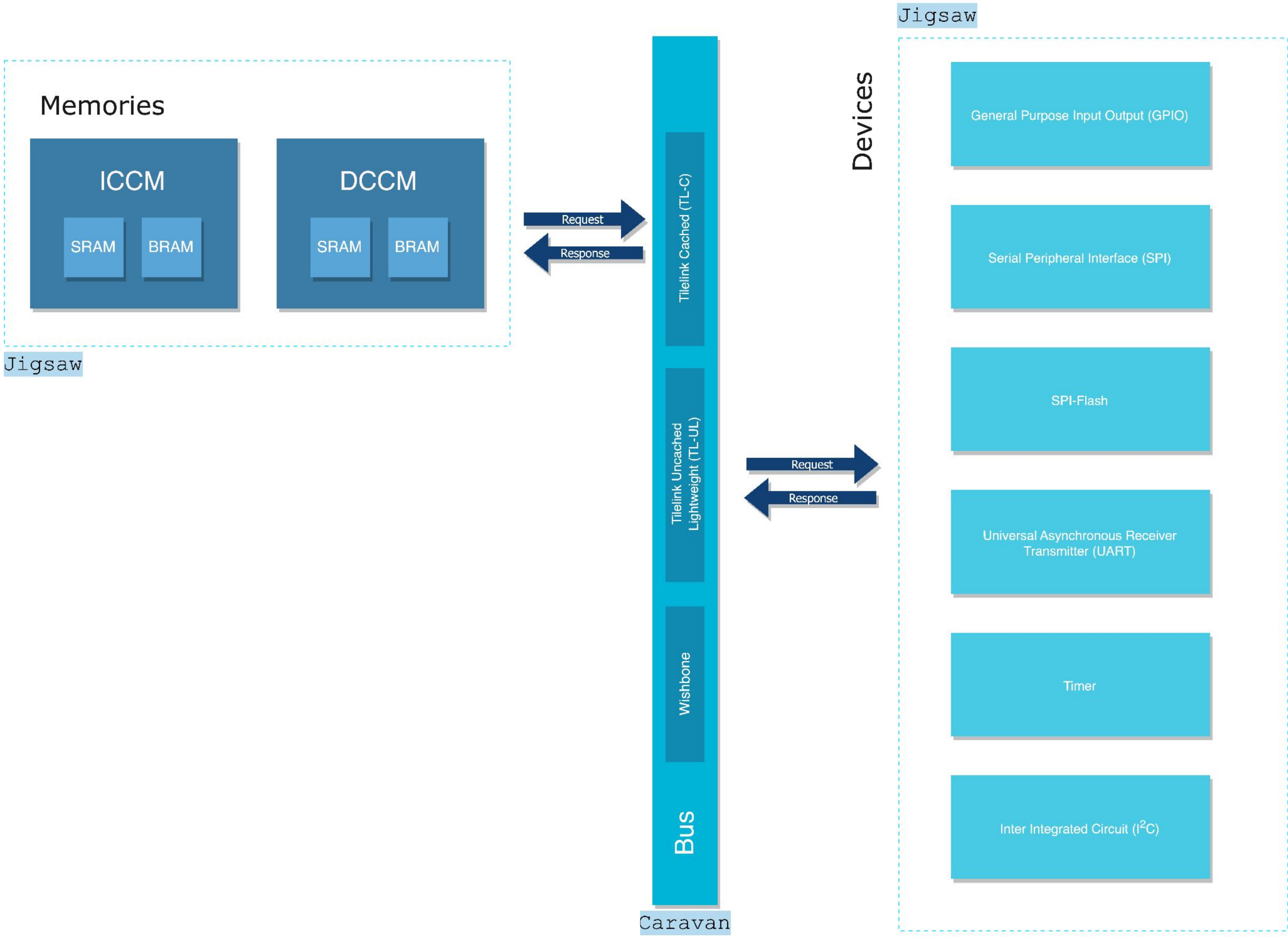
Jigsaw

- Jigsaw contains the implementation of devices and memories such as:

    1. Block RAMs
    2. Serial Peripheral Interface (SPI)
    3. SPI-Flash
    4. Universal Asynchronous Receiver Transmitter (UART)
    5. Inter Integrated Circuit (I2C)
    6. Timer

Jigsaw

Jigsaw has Generic Request and Response bundles with which devices are connected to the Bus Interconnect (Caravan)

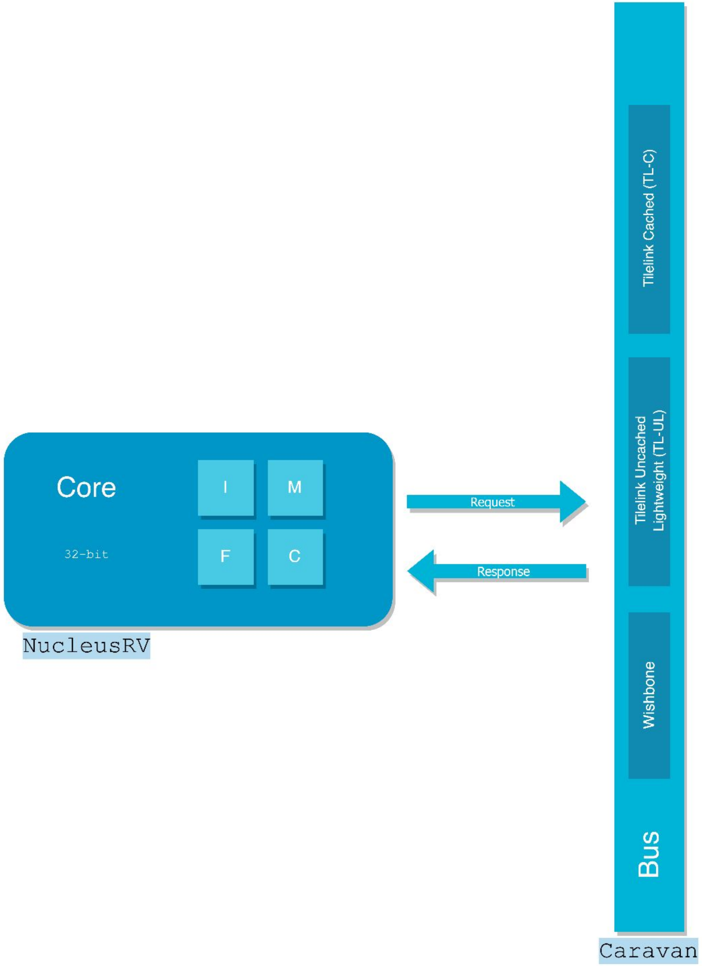

Memories
ICCM
SRAM
BRAM
DCCM
SRAM
BRAM
Jigsaw
Request
Response
Tilelink Cached (TL-C)
Tilelink Uncached Lightweight (TL-UL)
Wishbone
Bus
Caravan
Request
Response
Devices
Jigsaw
General Purpose Input Output (GPIO)
Serial Peripheral Interface (SPI)
SPI-Flash
Universal Asynchronous Receiver Transmitter (UART)
Timer
Inter Integrated Circuit (I2C)

NucleusRV

- NucleusRV (NRV) is our In-House RV32-IMC and partial F Core.
- It is completely designed in CHISEL HDL.
- It contains parameters to either include the M, F and C extensions.

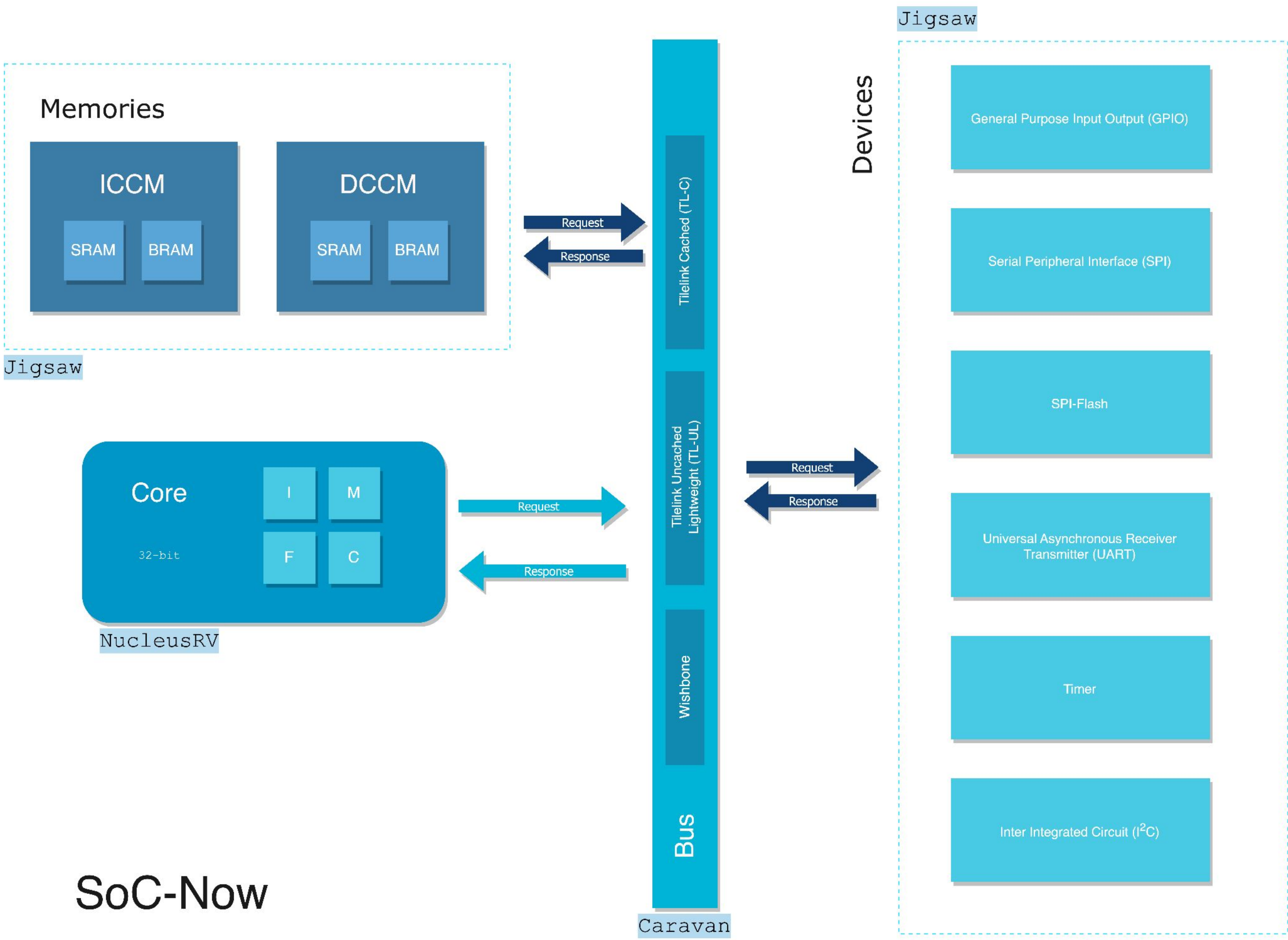
NucleusRV

NRV also have Generic Request and Response to be connected with the Bus Interconnect (Caravan) as Host



Complete Picture

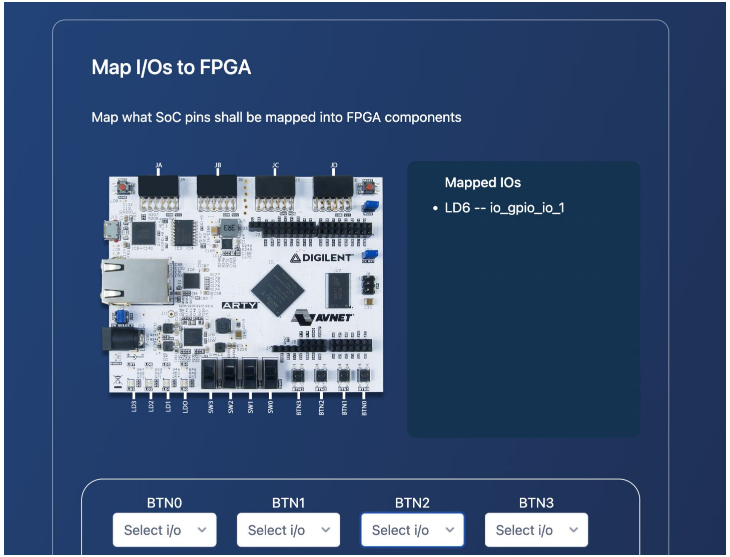
All these Frameworks connected together makes up SoC-Now


Memories
ICCM
SRAM
BRAM
DCCM
SRAM
BRAM
Jigsaw
Request
Response
Core
32-bit
I
M
F
C
NucleusRV
Request
Response
Tilelink Cached (TL-C)
Tilelink Uncached Lightweight (TL-UL)
Wishbone
Bus
Caravan
Request
Response
Devices
Jigsaw
General Purpose Input Output (GPIO)
Serial Peripheral Interface (SPI)
SPI-Flash
Universal Asynchronous Receiver Transmitter (UART)
Timer
Inter Integrated Circuit (I2C)

SoC-Now

FPGA Bitstream Generation

- Bitstream is generated by using the open source toolchain “F4PGA (FOSS for FPGA).



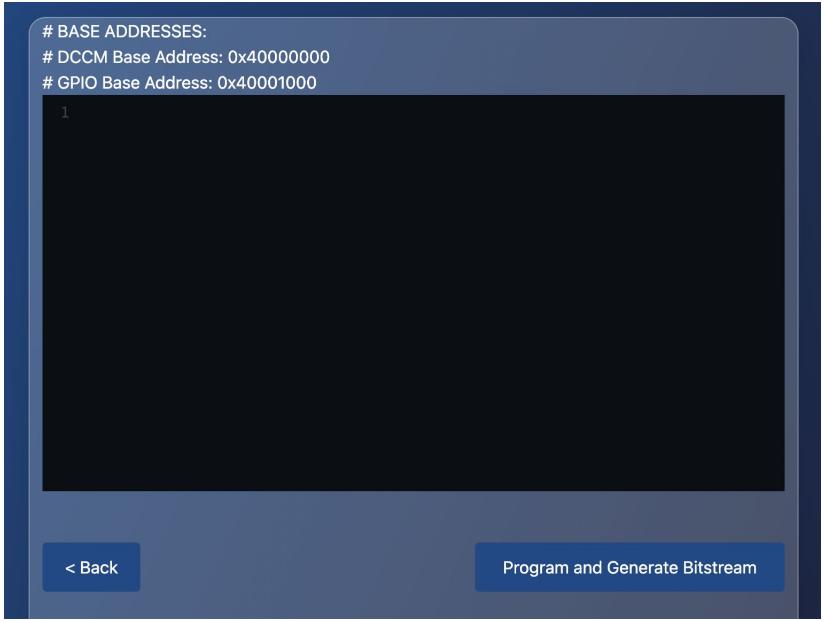
Map I/Os to FPGA

Map what SoC pins shall be mapped into FPGA components

Mapped IOs

- LD6 -- io_gpio_io_1

Mapping SoC IOs to FPGA Components



```
# BASE ADDRESSES:
# DCCM Base Address: 0x40000000
# GPIO Base Address: 0x40001000
```

Writing program to be burned in ICCM

Solution to the Industry Problem

- There is a rise in startups creating domain specific processors (DSPs) for solving problems in their niche.
- They require a base level system to integrate their DSP with.
- Usually the route is to pick up open-source core, attach memories and peripherals then integrate their DSP.
- With SoC Now our aim is to reduce the time spent in this effort.

Applications of SoC Now

- Generate the SoC with core + peripherals
- Verify the generated SoC
- Integrate your own DSP (in future)
- Generate bitstreams for FPGAs
- Generate GDS for ASIC (in future)

Achievements

- SoC Now was selected and funded by the National Technology Fund. The program aims to assist final year undergraduate students of ICT related disciplines studying in the Institutions by providing them financial assistance for developing prototypes / working models of their Final Year Projects (FYP)
- SoC Now was one of the 40 design selected for tapeout by Efabless Open MPW Program sponsored by Google. This program covers the cost of fabrication, packaging, evaluation boards and shipping.

Demonstration

Contributors

Hadir Khan
Computer Engineer
Research Associate MERL
Graduate Student UC-SC

Usman Zain
Computer Scientist
Research Associate MERL

Shahzaib Kashif
Software Engineer
Research Assistant MERL

Almas Ibrahim
Software Engineer
Research Intern MERL

Talha Ahmed
Software Engineer
Research Assistant MERL

Syeda Fizza
Software Engineer
Research Intern MERL

Muhammad Shahzaib
Software Engineer
Research Assistant MERL

Acknowledgement

- SoC-Now is a Final Year Project (FYP) of Undergraduate Software Engineering Students from **UIT University (UITU)** under the supervision of visionary mentors and with support and assistance of **Micro Electronics Research Lab (MERL).**
- We would also like to acknowledge the support of Open-Source Technologies and Community for providing with such revolutionary tools as **RISC-V**, **CHISEL**, **F4PGA**, **Python** and many more.

UIT UNIVERSITY

MERL

Accelerating Engineering Innovation

RISC-V

CHISEL

F4PGA